Dissertation

submitted to the

Combined Faculties for the Natural Sciences and for Mathematics

of the Ruperto-Carola University of Heidelberg, Germany

for the degree of

Doctor of Natural Sciences

presented by

Sasithorn Chotewutmontri (M.Sc. Biotechnology)

born in Nakhon Phanom, Thailand

Oral-examination:

Genome integration structures and genotype variants of oncogenic human papillomavirus types HPV16 and HPV68 in cervical carcinoma-derived cell lines, cervical precursor lesions and carcinomas

Referees: **Prof. Dr. Elisabeth Schwarz**

Prof. Dr. Gabriele Petersen

First of all, I would like to express my utmost gratitude to my mentor, Prof. Dr. Elisabeth Schwarz, for giving me the opportunity to work in this interesting field, for her constant support and valuable advice in both scientific and personal matters, for her critical guidance, endless patience and the wealth of suggestions and comments in the completion of this dissertation, and for translating the Abstract into the German Zusammenfassung. At the same time, I would like to thank Prof. Dr. Gabriele Petersen for kindly being my second supervisor, and PD Dr. Stefan Wiemann for being a member of my advisory committee, and both of them for their attendance and discussions in my progress reports. Furthermore, I would like to thank PD Dr. Anne Régnier-Vigouroux and Prof. Dr. Rainer Zawatzky for kindly being my third and fourth examiners.

I would also like to thank our partners in the Cancéropôle du Grand-Est - DKFZ collaboration project, especially, Dr. Véronique Dalstein and Prof. Dr. Christine Clavel (Université de Reims Champagne-Ardenne, Reims, France), and Dr. Maëlle Saunier and Dr. Jean-Luc Prétet (Université de Franche-Comté, Besançon, France) for providing the clinical samples for this work.

I would like to specially thank Dr. Bo Xu for giving me the opportunity to collaborate in the development of the ASP16 strategy and for his advice and support. I also would like to give my thanks to Ursula Klos, Monika Frank-Stöhr, Ilona Braspenning-Wesch and Birgit Hub for laboratory advice and assistance, and for their friendship. Together, they created the most pleasant and friendly working environment I have ever known. Furthermore, I would like to thank Prof. Dr. Frank Rösl and the members of the Department of Viral Transformation Mechanisms at DKFZ for their support and friendliness.

I would like to thank Andreas Hunziker for Sanger sequencing, and Dr. Stephan Wolf for Roche/454 GS-FLX pyrosequencing and related discussions.

I would like to thank Dr. Kajohn Boonrod, Ria Kretzer and Jochen Kretzer for their constant support and encouragement in difficult times, and for being my second family.

Finally, I would like to dedicate my thesis to my beloved parents, sister Joy and brother Non. Without them, I would not have achieved this far.

Für die Entstehung von Gebärmutterhalskrebs (Zervixkarzinom) ist eine persistierende Infektion mit humanen Papillomviren (HPV) der Hochrisiko-Gruppe die entscheidende Voraussetzung. Häufig erfolgt später eine Integration der viralen DNA in das Genom der Wirtszellen. Durch die Integration wird meistens das virale E2-Gen zerstört oder deletiert, wodurch es zur Deregulation der in der „upstream regulatory region“ (URR) gestarteten Transkription der viralen Onkogene E6 und E7 kommt. Einen zusätzlichen Einfluss auf den Mehrstufenprozess der Zervixkarzinogenese kann die integrierte HPV-DNA ausüben, indem wichtige zelluläre Gene durch Insertionsmutagenese verändert werden. HPV16 ist der häufigste und HPV68 ein seltener Hochrisiko-HPV-Typ. HPV16 kommt in ungefähr 55 % der weltweit untersuchten Zervixkarzinome vor, HPV68 in weniger als 1 %. In dieser Arbeit wurden in Zervixkarzinom-Zelllinien und in klinischen Proben von Zervixabstrichen die DNA-Strukturen von HPV68 bestimmt sowie HPV16-Integration und Sequenzen des E1-E2-Genbereichs mittels der neuen ASP16-Strategie („amplification selection pyrosequencing of HPV16“) analysiert.

HPV68 wird in zwei Subtypen, a und b, unterteilt. Besonderes Merkmal von HPV68b ist das Vorkommen als integrierte DNA in der Zervixkarzinom-Zelllinie ME180. Die davon abgeleitete Linie ME180R, resistent gegenüber Wachstumsinhibition durch Tumor-Nekrose-Faktor alpha (TNFalpha), weist partielle Deletionen in der integrierten HPV68b-DNA auf. In dieser Arbeit wurden die kompletten Strukturen der integrierten HPV68b-DNA in ME180 und ME180R bestimmt. ME180-Zellen enthalten zwei unvollständige Kopien von HPV68b, die in einer einzigartigen Kopf-Kopf-Anordnung integriert vorliegen. Durch Selektion von zwei neuen TNFalpha-resistenten ME180-Sublinien konnte gezeigt werden, dass die Umlagerungen und partiellen Deletionen der integrierten HPV68b-DNA in ME180R nicht wesentlich für den TNFalpha-resistenten Phänotyp sind. Aus einer CIN2-Krebsvorstufe (zervikale intraepitheliale Neoplasie Grad 2) wurden ein komplettes und ein mutiertes HPV68b-Genom isoliert, kloniert und sequenziert. Das mutierte Genom, das eine 1,2 kb große Deletion im E1-Gen aufweist, liegt wahrscheinlich integriert vor. In elf HPV68-positiven klinischen Proben wurde das virale Genom durch partielle URR-Sequenzierung untersucht. Eine Probe enthielt HPV68a, die anderen zehn Proben enthielten HPV68b-Varianten, von denen neun vorher unbekannt waren. Die Ergebnisse zeigen, dass HPV68b häufiger als HPV68a und in vielen molekularen Varianten vorkommt.

Die ASP16-Strategie wurde entwickelt zur gleichzeitigen Bestimmung von HPV16-Integrationsstellen in einer Vielzahl von klinischen DNA-Proben. ASP16 besteht aus vier Hauptschritten: GenomePlex Gesamtgenom-Amplifikation, Anreicherung von HPV16 E1-E2-Sequenzen, Roche/454 GS-FLX Pyrosequenzierung, und Daten-Analyse. In dieser Arbeit wurden Computerprogramme zur ASP16-Datenanalyse entwickelt und eingesetzt. Die ASP16-Strategie wurde weiter optimiert und zur Analyse von 25 HPV16-positiven Proben eingesetzt. Es wurden längere Einzelsequenzen als vorher sowie eine Sequenzabdeckung von 89 % erreicht. HPV16-Integrationsstellen konnten in 3 von 4 Zelllinien und in 6 von 21 klinischen Proben identifiziert werden. Die in den klinischen Proben identifizierten HPV16-Integrationsstellen liegen alle in oder in der Nähe von zellulären Proto-Onkogenen oder Tumorsuppressorgenen. Diese Befunde unterstützen die Vermutung, dass die HPV-Integration durch Veränderungen krebsrelevanter zellulärer Gene zur Zervixkarzinogenese beitragen kann. Die hohe Sequenzabdeckung im E1-E2-Bereich ermöglichte auch die Bestimmung von HPV16-Varianten. Die ASP16-Strategie ist die erste Methode, die „next generation sequencing“-Technologien mit der Bestimmung von HPV-Integrationsstellen im Multiplex-Format kombiniert. ASP16 ermöglicht damit die serienmäßige Analyse von HPV16-Integrationsstellen in klinischen Proben und liefert gleichzeitig E1-E2-Sequenzen zur Bestimmung von Mutationen und Varianten.

Abstract

Persistent infection with high-risk human papillomavirus (hr-HPV) is essential for cervical carcinogenesis, and is frequently followed by integration of the viral DNA into the host genome. Upon integration, the viral E2 gene is usually disrupted or deleted leading to deregulated transcription of the E6/E7 oncogenes from the upstream regulatory region (URR). Integrated HPV DNA may also affect critical cellular genes through insertional mutagenesis, which can contribute to the multi-step process of cervical carcinogenesis. HPV16 is the most frequent and HPV68 is a rare hr-HPV type, present in about 55% and less than 1% of cervical carcinomas worldwide, respectively. In this work, HPV68 DNA structures in cervical carcinoma cell lines and clinical samples were analyzed. HPV16 integration and E1-E2 sequences were studied using the novel "amplification selection pyrosequencing of HPV16" (ASP16) strategy.

HPV68 is divided into two subtypes, a and b. A hallmark of HPV68b is its presence as integrated DNA in the cervical carcinoma cell line ME180. In the mutant cell line ME180R, selected for resistance to growth inhibition by tumor-necrosis-factor alpha (TNFalpha), partial deletions in the integrated HPV68b DNA had been detected. In this study, the complete structures of the integrated HPV68b in ME180 and ME180R have been determined. ME180 cells contain two disrupted HPV68b copies, integrated in a unique head-to-head arrangement into chromosome 18q21. By selection of new TNFalpha-resistant ME180 sub-lines, it was found that the rearrangements and partial deletions of HPV68b in ME180R are unnecessary for the TNFalpha-resistance phenotype. In addition, a full-length and a mutant HPV68b genome were isolated from a cervical intraepithelial neoplasia grade 2 (CIN2) precursor lesion, cloned and completely sequenced. The mutant genome carrying a 1.2-kb deletion in the E1 gene is probably integrated. Based on partial URR sequences, ten HPV68b variants, nine of them new, and one HPV68a variant have been identified in eleven clinical samples, suggesting that HPV68b is more widely distributed than HPV68a and is present in a multitude of molecular variants.

ASP16 was developed for simultaneous determination of HPV16 integration junctions in multiple clinical DNA samples. It consists of four main steps: GenomePlex whole genome amplification, HPV16 E1-E2 sequence enrichment, Roche/454 GS-FLX pyrosequencing, and data analysis. In this work, computer programs for ASP16 data analysis were developed and applied. The ASP16 strategy was further optimized and used for the analysis of 25 HPV16-positive samples. The optimized ASP16 delivered longer sequence read lengths and 89% average sequence coverage. HPV16 integration junctions were identified in 3 out of 4 cell lines, and 6 out of 21 clinical samples. The HPV16 integration sites identified in the clinical samples are all located near cellular proto-oncogenes or tumor suppressor genes, supporting the assumption that HPV integration contributes to cervical carcinogenesis by altering cancer-relevant cellular genes. The high E1-E2 sequence coverage also allowed HPV16 variant assignments. Altogether, the ASP16 strategy, which is the first method combining next generation sequencing technologies with HPV integration analysis in a multiplex format, shows the potential to identify HPV16 integration junctions in series of clinical samples in parallel and at the same time provides E1-E2 sequences suitable for mutation/variant analysis.

aa	amino acid
ASP16	amplification selection pyrosequencing of human papillomavirus type 16
bp	base pair
CA	cancer
CF	cutoff
CGE	Cancéropôle du Grand-Est
chrom.	chromosome
CIN	cervical intraepithelial neoplasia
DKFZ	Deutsches Krebsforschungszentrum (German Cancer Research Center)
emPCR	emulsion polymerase chain reaction
GPUA	GenomePlex universal adapter
HPV	human papillomavirus
hr-HPV	high-risk human papillomavirus
HSIL	high-grade squamous intraepithelial lesion
kb	kilo base pair
lr-HPV	low-risk human papillomavirus
LSIL	low-grade squamous intraepithelial lesion
MDA	multiple displacement amplification
nt	nucleotide
ORF	open reading frame
pBS	pBluescript vector
PCR	polymerase chain reaction
pos.	position
qPCR	quantitative polymerase chain reaction
RT-qPCR	real-time quantitative polymerase chain reaction
RA	Roche-A
RB	Roche-B
RCA	rolling circle amplification
RS-PCR	restriction-site polymerase chain reaction
TNFalpha	tumor-necrosis-factor alpha
URR	upstream regulatory region
WGA	whole genome amplification

Table of contents

1. Introduction

1.1 Human papillomavirus and cervical cancer

Cervical cancer is the third most frequent cancer in women worldwide; with the current number of new cases annually over 500,000 and the number of death over 270,000 (de Sanjose et al, 2010; WHO/ICO Information Centre on HPV and Cervical Cancer (HPV Information Centre), 2010). Human papillomaviruses (HPVs) are detected in 99.7% of cervical cancer worldwide (Walboomers et al, 1999). It was established that infection with HPV is a necessary cause of cervical cancer (Trottier & Franco, 2006; Walboomers et al, 1999; zur Hausen, 2002), and Prof. Harald zur Hausen was awarded the 2008 Nobel Prize in Physiology and Medicine for his discovery of HPV causing cervical cancer. HPVs are classified into "types", "subtypes" and "variants" based on the difference in nucleotide sequence of the L1 gene, with at least 10%, between 2-10% and maximally 2%, respectively (de Villiers et al, 2004). There are over 100 HPV types identified (Bernard et al, 2010; de Villiers et al, 2004), infecting cutaneous epithelia and mucosal (anogenital) epithelia. The genital HPVs, which cause benign and malignant lesions in the anogenital tract, are classified according to their potential to induce malignant transformation into high-risk and low-risk types (hr-HPV and lr-HPV). Fifteen HPV types are considered as high-risk: 16, 18, 31, 33, 35, 39, 45, 51, 52, 56, 58, 59, 68, 73 and 82 (Jacobs et al, 1995; Munoz et al, 2003). HPV16 and HPV18 are the two most prevalent genital HPVs, detected in ~50% and ~15% of cervical cancer worldwide, respectively (Castellsague, 2008; Li et al, 2010).

There are two major cervical cancer types, squamous cell carcinomas (SSCs) and adenocarcinomas. SSCs, accounting for almost 90% of cervical carcinomas (Vizcaino et al, 1998), develop from non-invasive precursor lesions, called cervical intraepithelial neoplasias (CINs) or squamous intraepithelial lesions (SILs) (Schiffman et al, 2007; Snijders et al, 2006). CINs, classified based on tissue histology, are divided into three grades, CIN1-3. SILs are classified into low and high grade, LSIL and HSIL, according to the Bethesda classification (Solomon et al, 2002). Figure 1.1 illustrates the different

developmental stages of squamous epithelium from normal to cancer, in combination with CINs and SILs grades.

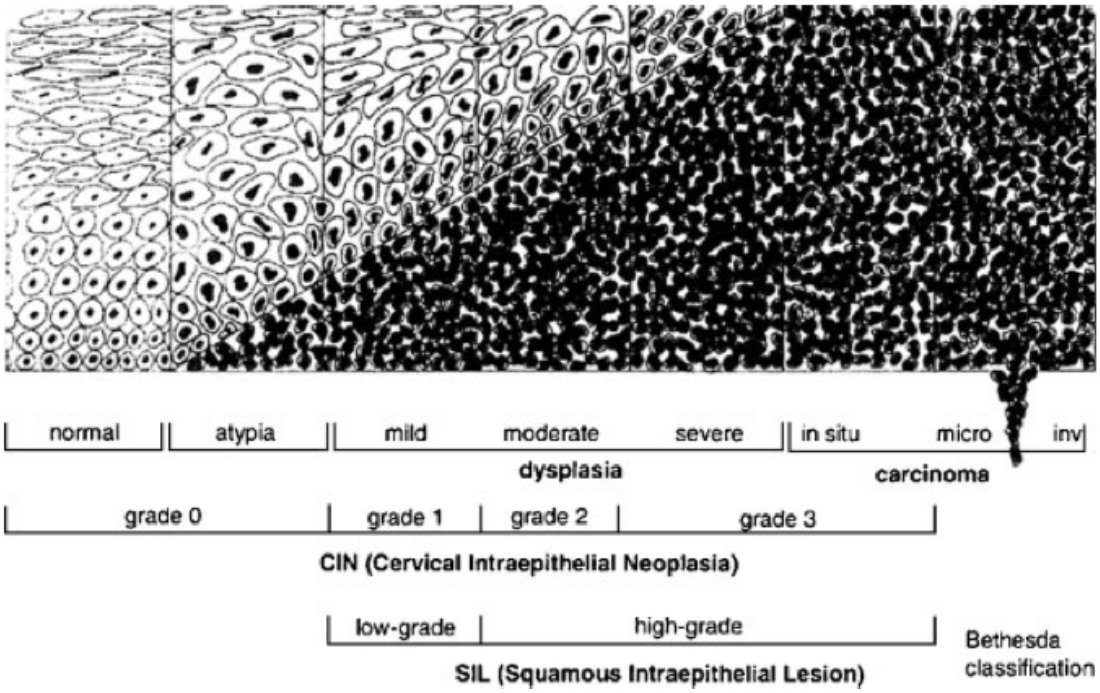


Figure 1.1: Model of histologically classified cervical squamous lesions. Taken from (Snijders et al, 2006). Morphological alterations of different development stages of cervical epithelial lesions are shown in comparisons with CIN and SIL classifications.

1.2 HPV genome organization

HPVs are small, non-enveloped DNA viruses. Their genomes are double-stranded DNA of about 8 kb encoding 8 genes, which are transcribed as polycistronic mRNAs from one DNA strand (Figure 1.2). These circular genomes are also called episomes. The HPV genome is composed of three regions: the non-coding upstream regulatory region (URR), and the protein-coding early and late regions.

The URR contains the origin of viral DNA replication, enhancer elements with binding sites for many cellular and viral transcriptional activators and repressors, and the early promoter (P97 in HPV16) at which transcription of the early genes is initiated (Park et al, 1995). The HPV16 P97 promoter (equivalent to HPV31 P99 and HPV18 P105), located upstream of ORF E6, is responsible for the expression of almost all early genes, whereas

the HPV16 P670 promoter, located within ORF E7, is responsible for the late gene expression (Zheng & Baker, 2006). The early promoter is controlled primarily by cis-elements in the URR.

The early region contains six early genes E6, E7, E1, E2, E4 and E5. The E1 and E2 genes encode proteins involved in the initiation of viral DNA replication (McBride, 2008). E1 encodes the primary viral DNA replication initiator protein, consisting of an N-terminal domain, a DNA-binding domain, an oligomerization domain and a C-terminal helicase domain (McBride, 2008; Stenlund, 2003). In the process of initiation of viral DNA replication, a dimer of the E1 protein cooperates with a dimer of the E2 protein and bind to the adjacent E1 and E2 binding sites (E1BS and E2BS) on the replication origin in the URR. The complex with E2 allows E1 to bind to the origin with higher specificity (Stenlund, 2003). When the E2 protein dimer is cut off from the complex through ATP hydrolysis by E1, additional E1 molecules are incorporated into the existing E1 dimer to form double E1 hexamers (Stenlund, 2003). The E1 hexamers melt and unwind the origin through ATP-dependent helicase activity, and when associated with the cellular DNA synthesis machinery, such as DNA polymerase alpha-primase, new viral DNA can be synthesized (McBride, 2008; Stenlund, 2003). The E2 multi-functional protein consists of three domains: the N-terminal transactivation domain (TAD), the non-conserved hinge region, and the C-terminal DNA binding domain (Ham et al, 1991). Apart from its involvement in the viral DNA replication initiation, E2 protein plays an important role in the regulation of viral promoter activity through several E2BSs in the URR (McBride, 2008). It had been demonstrated that the E2 of the bovine papillomavirus type 1 (BPV1) can activate early gene transcription (Spalholz et al, 1985; Thierry & Yaniv, 1987). When E2 binds to the E2BSs overlapping key promoter elements, such as TATA box and SP1 binding site, it represses viral transcription (McBride, 2008; Thierry, 2009). Another essential function of E2 is partitioning and maintaining the replicating viral episomes. The E2 protein binds to E2BSs in the URR of the episomes and attaches them to mitotic chromosomes, ensuring that the episomes are maintained within the nuclear envelope after mitotic and segregated evenly between daughter cells (McBride et al, 2006).

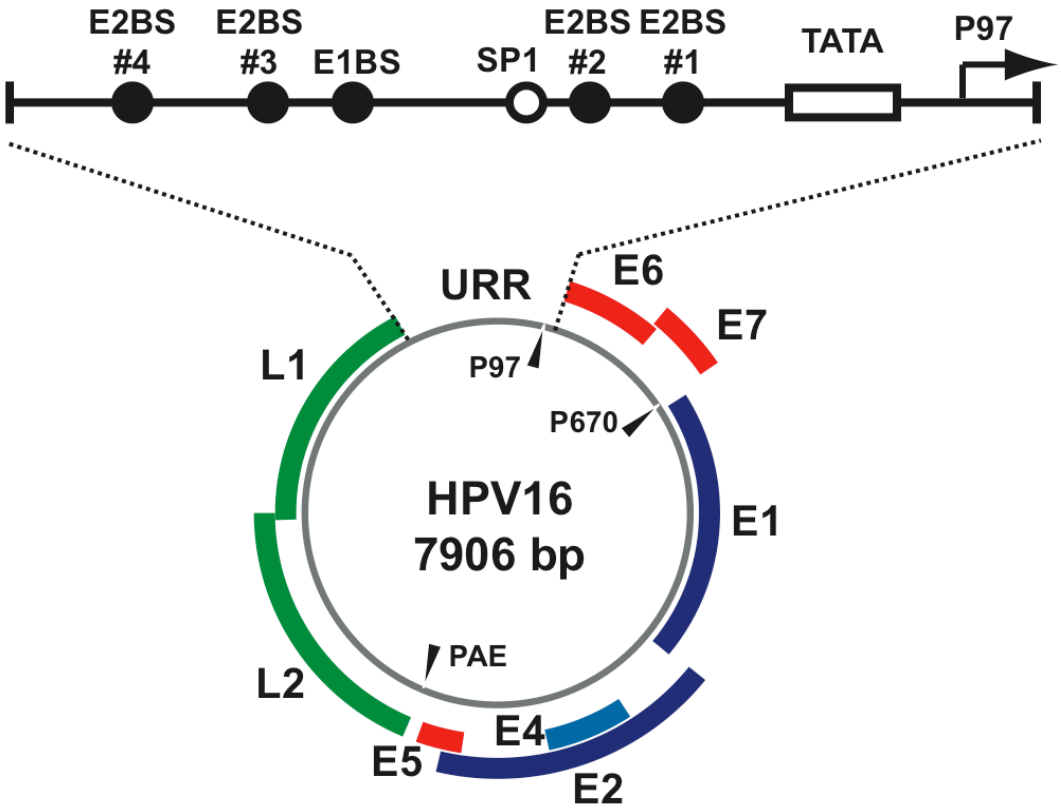


Figure 1.2: HPV16 genome organization. Adapted from Fig. 2 of (Doorbar, 2006). HPV16 genome is shown in circular form. The open reading frames (start to stop codon) of early genes (red and blue bars) and late genes (green bars) are indicated. Locations of the E1 binding site (E1BS), four E2 binding sites (E2BS #1-4), the SP1 binding site (SP1), TATA box and P97 in the non-coding upstream regulatory region (URR) are shown on top. P97 and P670: promoters. PAE: early polyadenylation site.

ORF E4 is located in the early region, embedded into the E2 region, but encodes a late gene product. Since ORF E4 does not contain a start codon, the E4 protein is translated from a transcript E1^E4. The E4 protein, also called E1^E4, therefore, carries five N-terminal amino acid residues from ORF E1. E4 protein is associated with the intermediate filament network of upper epithelial cells resulting in keratin reorganization and keratin depletion, and it may be involved in the release of mature virions from the infected cells (Doorbar et al, 1991; McIntosh et al, 2010; Wang et al, 2004). The E6, E7 and E5 genes encode viral oncoproteins (Moody & Laimins, 2010), with E6 and E7 playing the major roles in HPV-induced carcinogenesis. E6 protein binds to and facilitates the degradation of the tumor suppressor protein p53 (Werness et al, 1990). E7 protein binds to and facilitates the degradation of the tumor suppressor retinoblastoma protein (pRb) (Boyer et al, 1996; Chellappan et al, 1992). E5 protein supports cell cycle progression, and has weak transforming activity (Fehrmann et al, 2003; Maufort et al, 2010).

The late region contains two late genes L1 and L2, encoding the viral major and minor capsid proteins which encapsidate newly synthesized viral episomes (Moody & Laimins, 2010).

1.3 HPV integration and cervical carcinogenesis

Persistent infection with hr-HPV is an essential prerequisite for cervical carcinogenesis (zur Hausen, 2002). Expression of E6 and E7 genes is necessary but not sufficient to cause cervical carcinomas, and additional viral and cellular genetic alterations are required (zur Hausen, 1999). The progression from normal histology to invasive cancer is a multi-step process, as shown in Figure 1.3 (reviewed in details by Schiffman et al, 2007; Snijders et al, 2006; zur Hausen, 2000). After infection with hr-HPV, persistent expression of viral DNA combined with other factors leads to the development of precursor lesions, LSIL and HSIL (or CIN1-3). During these stages, the majority of the lesions can spontaneously regress (Chan et al, 2003; Ho et al, 1998; Nasiell et al, 1986). In some high-grade lesions, integration of hr-HPV into the host genome takes place. The integration, combined with additional cellular genetic alterations, induces genomic instability leading to progression toward invasive cancer.

In productive HPV infection (reviewed in details by Doorbar, 2005), the HPV DNA is maintained as episome of about 50-100 copies/cell (Stanley et al, 1989) in differentiating cells arising from the basal epithelial layer. During this stage, the expression of E6/E7 oncogenes is highly regulated and does not pose any risk toward cancer progression. This is because the viral oncogenes are expressed in the cells that are migrating from the inner layer to the outer epithelium, and eventually, these cells will be shed from the body. As the lesions progress, integrated HPV DNA can be detected: 5% in HSIL and 81% in cervical cancers (Cullen et al, 1991). These numbers suggested the importance of HPV integration as a major event before progression into cancer, and the integration of hr-HPV has been shown to be an important indication of progression to invasive cancer (Hopman et al, 2004).

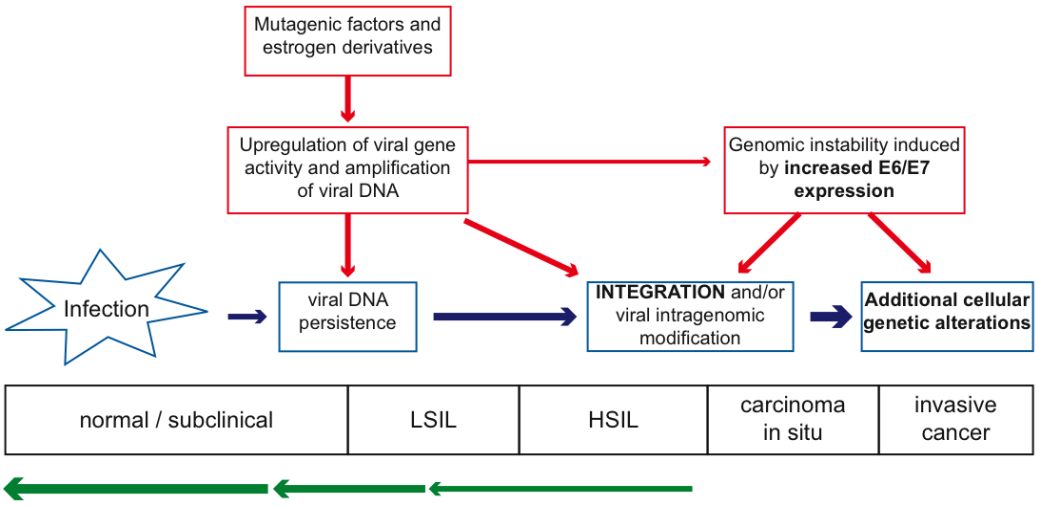


Figure 1.3: Multi-step progression to cervical cancer. Adapted from (zur Hausen, 2000). The scheme shows the progression from infection to invasive cancer. The factors contributing to progression (red and blue boxes) are shown in parallel with the histological stages (black boxes). Most of the lesions spontaneously regress back to normal as indicated by green arrows.

Upon integration, the episomal HPV DNA is disrupted and inserted into human chromosomes. The HPV breakpoints usually locate in the E1-E2 and L1-L2 regions. In integrated HPV DNA, E2 and/or E1 genes are often disrupted or deleted while URR, E6 and E7 genes always remain intact (Schwarz et al, 1985) (Figure 1.4). Since E2 represses the early promoter activity, the loss of E2 due to HPV integration allows transcriptional activation of E6/E7 oncogenes from the early promoter. HPV integration leads to increased level and stability of transcripts encoding the E6/E7 oncogenes (Jeon & Lambert, 1995), and to growth advantage of the cells (Jeon et al, 1995). Furthermore, it has been shown that disruption of E1 or E2 increases the capacity of HPV16 to immortalize cells (Romanczuk & Howley, 1992). Since the viral polyadenylation signal sequence (PAE) is missing due to integration, transcription of the E6/E7 oncogenes from the integrated HPV DNA continues until the next polyadenylation signal in the cellular DNA. This results in viral-cellular fusion transcripts. From the spliced viral-cellular fusion transcripts, the E6/E7 oncoproteins are expressed (Schneider-Gadicke & Schwarz, 1986; Smotkin & Wettstein, 1986).

HPV integration may lead to alterations of cellular genes through insertional mutagenesis. These alterations, especially the activation of cellular proto-oncogenes or inactivation of cellular tumor suppressor genes, could also contribute to cervical carcinogenesis. The integration of HPV DNA into human chromosomes occurs randomly, with preference for

genomic fragile sites and transcriptionally active regions (Kraus et al, 2008; Thorland et al, 2003; Wentzensen et al, 2004). Recurrent HPV integration sites have also been reported, in particular, integration within or close to the proto-oncogene c-myc (Couturier et al, 1991; Durst et al, 1987; Ferber et al, 2003; Peter et al, 2006; Wentzensen et al, 2002; Xu, 2010). MYC gene expression is activated when integrated hr-HPV DNA is located within chromosome 8q24 in the myc locus (Couturier et al, 1991; Peter et al, 2006; Xu, 2010), thus demonstrating activation of a cellular proto-oncogene by insertional mutagenesis. An example of the inactivation of a tumor suppressor gene by HPV integration is shown in the cervical cell line ME180 where hr-HPV68 is integrated into an intron of the potential tumor suppressor gene ZBTB7C (APM1) (Reuter et al, 1998).

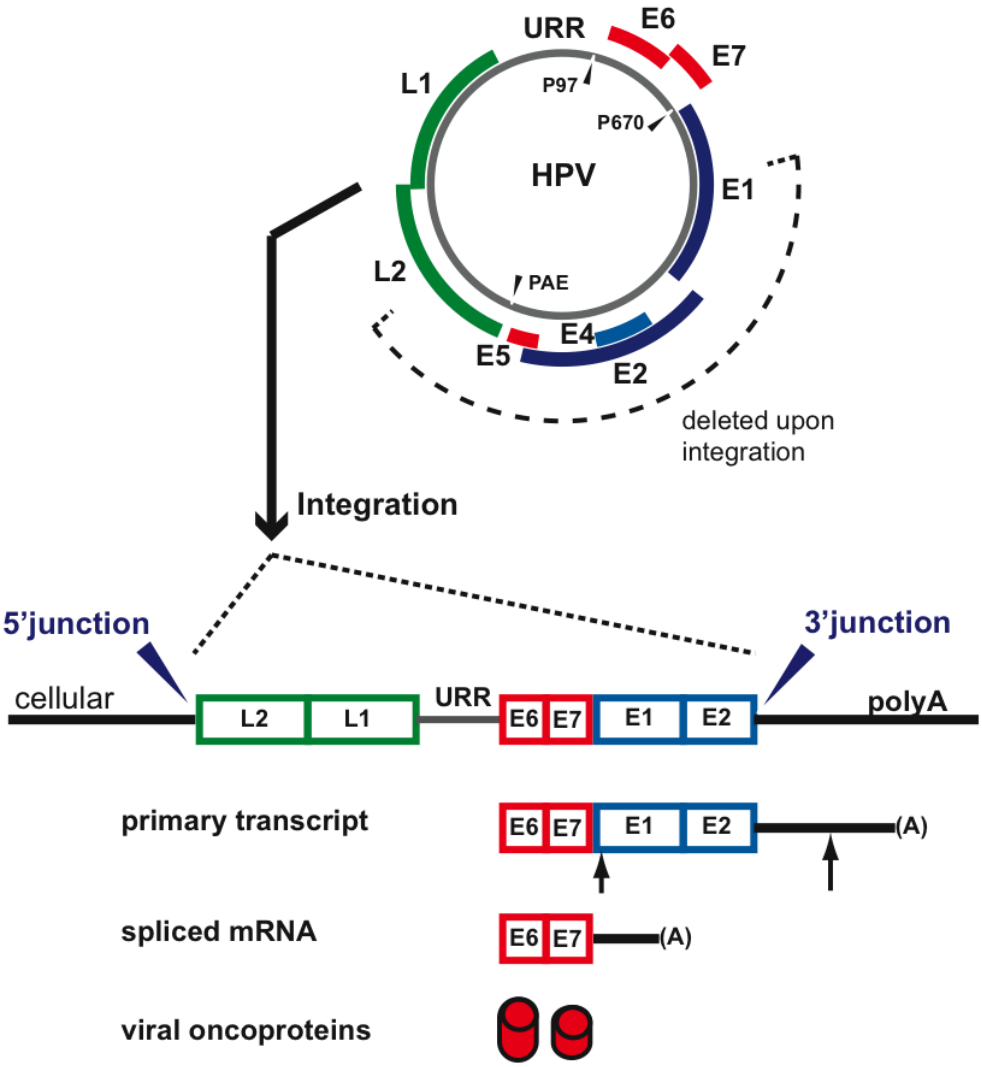


Figure 1.4: Viral oncogenes E6/E7 expression from integrated HPV DNA. An intact HPV16 episome is shown on top. The integrated DNA, flanked by cellular DNA (black lines), is shown in the lower portion. Splice donor and splice acceptor signals located in the primary transcript are indicated with small vertical black arrows. P97 and P670: early and late promoters. PAE: early polyadenylation signal.

1.4 Identification of HPV integration sites with the novel ASP16 strategy

Several methods have been developed for determining the location and structure of integrated HPV DNA. Fluorescence in situ hybridization (FISH) has been used to locate integrated HPV in the chromosomes and also to estimate the copy number of the integrated HPV (Callahan et al, 1992; Hopman et al, 2004; Mincheva et al, 1987). Based on the presence of viral-cellular transcripts produced from integrated HPV (Schneider-Gadicke & Schwarz, 1986), an RT-PCR method, named "amplification of papillomavirus oncogenes transcripts" (APOT) has been established which allows identification of integrated HPV and the chromosomal location from the cDNA sequences of the hybrid mRNA transcripts (Klaes et al, 1999). PCR-based methods for genomic DNA analysis were also developed, allowing the exact sequences of the HPV integration junctions to be determined (Kalantari et al, 2001; Luft et al, 2001). These methods require extensive laboratory works and the analysis is performed sample by sample.

In our group, another PhD student, Bo Xu, had started developing a novel PCR-based strategy for simultaneous determination of the 3'junction of integrated HPV (see Figure 1.4) of multiple DNA samples in one experiment. The strategy is called "Amplification Selection Pyrosequencing of HPV16", abbreviated ASP16 (Xu, 2010). The principle of the ASP16 analysis includes four main steps: (1) amplification of the genomic DNA from HPV16-positive cervical lesions using GenomePlex whole genome amplification, (2) enrichment of HPV16 DNA by linear amplification and HPV16-specific multiplex PCR, (3) Roche/454 GS-FLX pyrosequencing, and (4) data analysis. The strategy is described in details in (Xu, 2010), and illustrated in Figure 1.5. Briefly, 10-30 ng total DNA of each DNA sample is amplified using GenomePlex Whole Genome Amplification (Sigma Aldrich) where the genomic DNAs are chemically fragmented, ligated with the GenomePlex universal adapter (GPUA) at both ends, and amplified by PCR. The amplified DNA product of each DNA sample, called OmniPlex library, is composed of amplified randomly fragmented total DNA with the GPUA attached at both ends. To enrich HPV16 DNA, a set of biotin-labelled HPV16-specific forward primers is used to linearly amplify HPV16 DNA-containing molecules from each OmniPlex library in two multiplex reactions, each reaction composed of eight HPV16 primers. These primers span the HPV16 E1-E2 area where the 3'junction between viral and cellular DNA is located

(see Figure 1.4). The linear amplification products are purified using streptavidin-coated magnetic beads. They are then used as templates for semi-nested HPV16 multiplex PCR using corresponding bipartite nested forward primer and a tripartite reverse primer. The forward primers are composed of the Roche-A sequence and different HPV16 sequences. The reverse primers contain the Roche-B sequence, followed by a 4-nt barcode sequence used as identification tag for each DNA sample, and the GPUA sequence (see Figure 1.5). Roche-A and Roche-B are adapter sequences required in the sequencing step. The HPV16 multiplex PCR products are sequenced using the next generation sequencing Roche/454 GS-FLX pyrosequencing system, where individual amplicon molecules are amplified by emulsion PCR (emPCR) before being sequenced (http:/www.454.com/) (Metzker, 2010; Shendure & Ji, 2008).

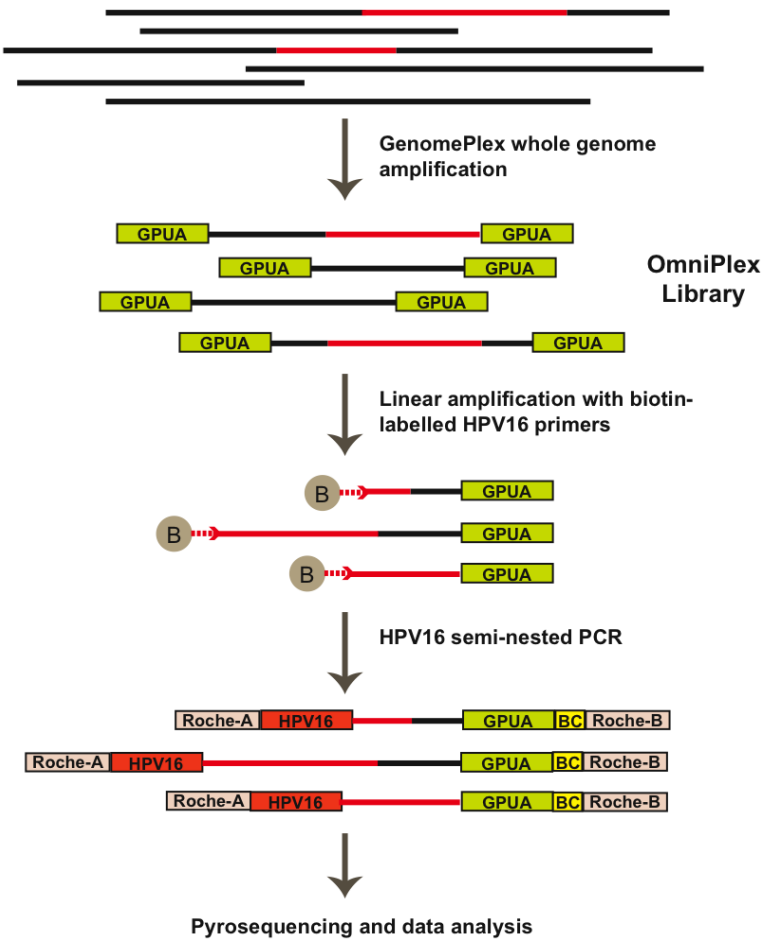


Figure 1.5: ASP16 strategy. Total DNA molecules are shown at the top. Black and red lines represent cellular DNA and HPV16 DNA, respectively. Red dashed arrows represent the biotin-labelled (B) HPV16 primers. Rectangular boxes represent the bipartite nested forward primers (Roche-A sequence fused with different HPV16 primer sequences) and the tripartite reverse primers (Roche-B sequence fused with 4-nt unique barcode sequence and GPUA sequence). GPUA: GenomePlex universal adapter. Roche-A and Roche-B: Roche adapter sequences. HPV16-boxes represent HPV16 nested primer sequences. BC: barcode.

1.5 CGE-DKFZ collaboration program

This PhD work is part of the joint program "Human papillomaviruses and cancer" between Cancéropôle du Grand-Est (CGE) and DKFZ. The main goals of the program are to unravel mechanisms of HPV-induced carcinogenesis, to identify novel progression markers, and to devise innovative therapies for virus-related cancers. We collaborate with CGE groups in Reims and Besancon working on HPV integration analysis in clinical DNA samples.

In Reims and Besancon, clinical cervical cell samples are collected from women participating in routine cervical cancer screening programs. The collected cells are tested cytologically using liquid-based ThinPrep system (Bory et al, 2002; Briolat et al, 2007; Clavel et al, 2001; Saunier et al, 2008). In the ThinPrep system, cervical cells are collected with cytobrushes, then resuspended and fixed in PreservCyt® solution. Cells are examined cytologically and classified based on Bethesda classification (Clavel et al, 2001; Solomon et al, 2002), see Figure 1.1. To determine the presence of hr-HPVs (types 16, 18, 31, 33, 35, 39, 45, 51, 52, 56, 58, 59 and 68), each cell sample is tested with the commercial Hybrid Capture 2 (HC-II) test (Digene). In the samples with hr-HPV, the HPV genotypes are determined using the commercial Linear Array (LA) HPV genotyping test (Roche) which tests for 15 hr-HPVs, 3 potentially hr-HPVs, 15 lr-HPVs and 4 HPVs with undetermined risk (Briolat et al, 2007). Additionally, cell samples are examined for DNA aneuploidy using DNA image cytometry because DNA aneuploidy has been recognized as a marker for lesions at risk for progression to cancer (Lorenzato et al, 2002; Lorenzato et al, 2001). From hr-HPV-positive cell samples, DNA and RNA are isolated for further molecular analyses.

For HPV16-positive cell samples, aliquots of the extracted DNAs are used to determine the physical status of HPV16 DNA by real-time quantitative PCR (RT-qPCR) of E2 and E6 genes (Briolat et al, 2007). The E2/E6 RT-qPCR method has been developed based on the gathered reports that the E2 hinge region is usually deleted, while the E6 gene is always intact in the integrated HPV16 (Peitsaro et al, 2002). E2 and E6 copy numbers are determined by RT-qPCR using E2- and E6-specific primer pairs and probes. The E2/E6 ratios indicate the HPV16 physical status. An E2/E6 ratio of 1 indicates pure episomal

HPV16 DNA, while values <1 indicate the presence of integrated HPV16. Figure 1.6 shows four examples of HPV16 physical status together with their expected E2/E6 ratio values and the percent integration values. Samples with pure episomal HPV16 should have 0% integration, samples with only integrated HPV16 100% integration, and samples with mixtures of episomal and integrated HPV16 DNA should have integration values of >0% to 100%. Cervical DNA samples with high percent integration values were selected for ASP16 analysis in this PhD work, because they should contain integrated HPV16 DNA.

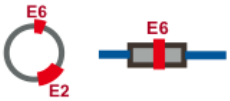
Physical status	Illustrated model	E2/E6 ratio	% Integration = (1 - E2/E6)*100%
Episomal HPV16		1/1 = 1	(1-1)*100% = 0 %
Integrated HPV16 (1 copy)		0/1= 0	(1-0)*100% = 100 %
Integrated HPV16 (2 copies)		1/2 = 0.5	(1-0.5)*100% = 50 %
Mixture: 1 episome 1 integrate		1/2 = 0.5	(1-0.5)*100% = 50 %

Figure 1.6: HPV physical status and E2/E6 ratio determined by RT-qPCR. The copy numbers of E6 and E2 genes (red boxes) are determined by RT-qPCR (Peitsaro et al, 2002). The E2/E6 ratios are indicated. Gray boxes represent individual HPV16 copies. Blue lines represent cellular DNA.

1.6 Goals of this work

This PhD work covers the integration analysis of two different hr-HPV types, HPV16 and HPV68b. While HPV16 is the most frequently detected type in cervical cancer, HPV68b is a rare hr-HPV. Previously in our group, HPV68 subtype b (HPV68b) was identified as integrated DNA in the cervical cell line ME180 (Reuter et al, 1991). The reported sequence of this integrated HPV68b was determined based on a cloned restriction fragment. Further analyses of this cell line, however, showed evidence of a probable

cloning-induced deletion in this restriction fragment (Reuter et al, 1991). Therefore, in the beginning of this PhD work, the complete and correct sequence of the integrated HPV68b DNA in the cell line ME180 was determined. Additionally, the mutant cell line ME180R, which is resistant to growth-inhibition by tumor-necrosis-factor-alpha (TNFalpha), was found to contain largely altered integrated HPV68b DNA compared to the parental cells ME180 (Elisabeth Schwarz, unpublished data). The question arose whether these alterations contribute to the resistance to TNFalpha. Thus, in this work, the complete integrated HPV68b DNA sequence in the cell line ME180R was determined, and it was investigated whether alterations in the integrated HPV68b are associated with resistance to TNFalpha by selecting new TNFalpha-resistant ME180 variants. Since no full-length HPV68b genome sequence was available, a complete HPV68b genome was isolated from a CIN2 DNA sample, and was sequenced.

In the HPV16 analysis using ASP16 strategy, the Roche/454 GS-FLX pyrosequencing system has the capacity to deliver about 210,000 sequence reads per sequencing round for the selected format. Such massive amount of data requires computer analysis software for data management and interpretation. With my knowledge in computer programming, I collaborated with Bo Xu in the ASP16 project and developed computer programs for automatically processing and analyzing ASP16 sequence data. Two ASP16 experiments had been performed previously (Xu, 2010). Because the strategy required further optimization, additional ASP16 experiments were performed as part of this PhD work.

2. Results

2.1 Analysis of HPV68-positive cell lines and cervical scrapes

In this study, the complete sequences of the integrated HPV68b in cell lines ME180 and ME180R were determined, and a complete genome of HPV68b was cloned from a CIN2 lesion and sequenced. By determination of sequence polymorphisms in the non-coding upstream regulatory region (URR) of HPV68 DNA in different cervical scrapes, new HPV68 variants were identified.

2.1.1 Determination of the complete integrated HPV68b sequence in ME180

The cell line ME180 was established from an omental metastasis of a rapidly spreading cervical carcinoma (Sykes et al, 1970). ME180 cells contain HPV68b DNA, integrated into chromosome 18q21 (Reuter et al, 1991). The integrated HPV68b in ME180 was first cloned and sequenced by Reuter et al (Reuter et al, 1991). The clone contains a SacI restriction fragment of 13.1 kb as insert AA13.1. Sequence analysis showed that the 13.1 kb fragment contains two incomplete copies of HPV68b DNA, named 5'copy and 3'copy, flanked by cellular sequences (Figure 2.1). Southern hybridization of SacI-digested genomic DNA showed a 20-kb restriction fragment instead of 13 kb (Reuter, 1995). This size discrepancy indicated a probable cloning-induced deletion of about 7 kb. According to the sequence organization, it was suspected that this deletion occurred in the area of the 5' copy of the integrated HPV68b (Figure 2.1). In the previous analyses, it was shown by Southern and PCR experiments of ME180 DNA that the full 5'copy probably contains ORFs E5, L2, L1, the non-coding URR, ORFs E6, E7 and E1 before the breakpoint in ORF E2 is reached (Reuter, 1995; Marco Springer and Elisabeth Schwarz, unpublished data). However, the exact structure and nucleotide sequences had not been determined. This was performed as part of this PhD thesis.

To obtain a complete sequence of the integrated HPV68b in ME180, both 5'copy and 3'copy were amplified by PCR, cloned into plasmid vectors and sequenced. Roche's Expand Long Template PCR System was used for amplification because of its high fidelity enzyme and long-range amplification capability. The amplification strategy for

both copies is illustrated in Figure 2.2. The 7.2-kb amplified fragment of the 5'copy was cloned into vector pCR4-TOPO (Invitrogen, Karlsruhe), the 6.2-kb fragment of the 3'copy into vector pSC-A (Stratagene, USA). The two clones were sequenced in both strands with Big-Dye terminator chemistry on an ABI model sequencer (Andreas Hunziker at the DKFZ Core Facilities). The sequences obtained from both clones (7123 bp and 6196 bp) were assembled at the overlapping area.

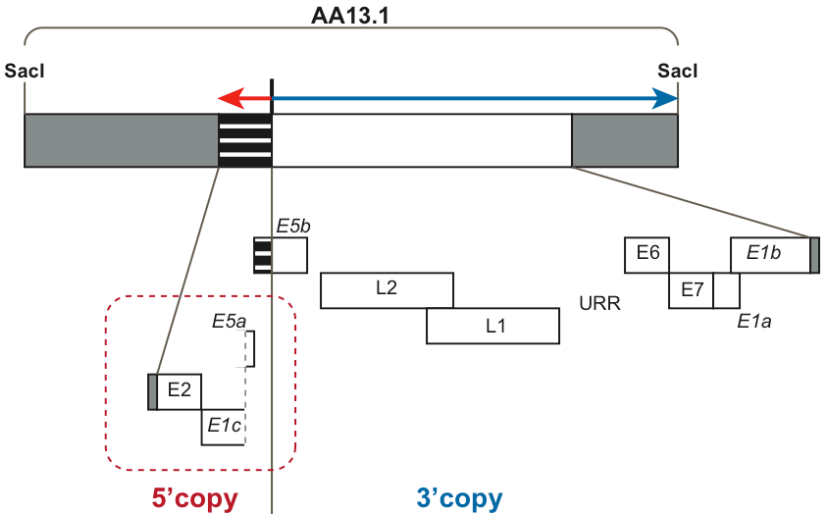


Figure 2.1: Previously cloned and sequenced SacI fragment AA13.1 containing integrated HPV68b in ME180. The 13.1-kb fragment contains two integrated incomplete copies of HPV68b in head-to-head orientation. The 5'copy is indicated by a horizontally striped box, the 3'copy by a white box, and the flanking cellular sequences by gray boxes. Red and blue arrows indicate the 5' to 3' polarity of the coding DNA strand. The open reading frames (ORFs) are shown in the enlarged segment below. The 5'copy was suspected to contain a cloning-induced deletion (circled in dashed red line). This figure is adapted from Fig. 2 of Reuter et al (Reuter et al, 1991).

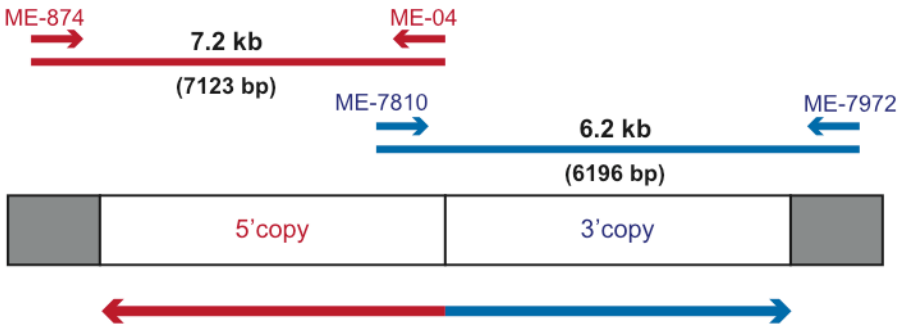


Figure 2.2: Amplification of the integrated HPV68b in ME180. Two fragments (red and blue lines), covering the integrated 5'copy (long red arrow) and 3'copy (long blue arrow) of HPV68b DNA were amplified from the genomic DNA of cell line ME180. The two primer pairs are indicated (short red and blue arrows) as well as the exact amplicon sizes determined by sequencing. Primers ME-874 and ME-7972 bind to the cellular DNA of chromosome 18 upstream and downstream of the HPV68b integration site. Primers ME-04 and ME-7810 bind to HPV68b DNA. Primer sequences are indicated in Materials and Methods.

The assembled nucleotide sequence, named HPV68b(int)-ME180, contains 13286 nucleotides. The sequence is shown in Appendix A1. The genetic organization of HPV68b(int)-ME180 is shown in Figure 2.3 and Table 2.1, and the following characteristics are emphasized:

1. The two HPV68b copies are integrated in head-to-head orientation into chromosome 18 between positions 45578341 (5' junction) and 45578364 (3' junction) based on sequence accession NC_000018.9, with 22 bp of cellular DNA deleted.
2. The 5' copy contains 6993 bp of HPV68b DNA. The 5'copy in AA13.1 has a size of only 870 bp (Reuter et al, 1991). This confirms the previous assumption of a cloning-induced deletion. Viewed in the 5'-to-3' direction of the coding strand, the 5'copy contains 41-bp non-coding sequence, intact ORFs E5, L2 and L1, the 809-bp URR, intact ORFs E6 and E7, followed by incomplete ORF E1 and truncated ORF E2. The identification of the 41-bp sequence as HPV68b DNA was possible because the sequence of a complete HPV68b genome has also been determined in this PhD work (see section 2.1.4). This sequence is non-coding, located between the E2 stop codon and the E5 start codon. The ORF E5 was considered to be complete because it contains the same 222 nucleotides from the start codon to the stop codon as in the published ORF E5 of HPV68a (accession DQ080079). The URR contains a 2-bp gap compared to the 3'copy (Figure 2.19). The ORF E1 harbors a 64-bp deletion causing a frameshift and a disruption into ORFs E1a and E1b (Figure 2.4). The E2 gene carries the integration breakpoint located 359 bp downstream of the E2 ATG start codon.
3. The 3' copy contains 6071 bp of HPV68b DNA. Viewed in the 5'-to-3' direction of coding strand, the 3'copy contains a truncated ORF E5, intact ORFs L2 and L1, intact URR, followed by intact ORFs E6 and E7, and a truncated ORF E1. Compared to the complete E5 ORF in the 5'copy, the E5 ORF of the 3'copy is truncated, missing the first 46 bp. The E1 gene is truncated at 1289 bp downstream of the ATG start codon. It is joined with a 14-bp sequence derived from ORF E5 before the flanking cellular sequences start.
4. The two viral-cellular junctions in the 5' and 3' copies of HPV68b(int)-ME180 are identical to those of AA13.1 (Figure 2.3).
5. No intact ORF E1 is present in HPV68b(int)-ME180. The E1 of the 3'copy is truncated due to integration and the E1 of the 5'copy harbors a 64-bp deletion and frameshift (Figure 2.4). Surprisingly, the 64-bp deletion originally identified in AA13.1 in

the E1 part of the 3'copy is present in the complete sequence HPV68b(int)-ME180 in the E1 part of the 5'copy (Figure 2.3). This indicates complex cloning-induced DNA rearrangements in clone AA13.1. In the 5'copy, ORF E1a encodes 116 amino acids with 108 N-terminal translated from the E1 frame and 8 C-terminal amino acids from a shifted frame due to the 64-bp deletion. ORF E1b encodes 555 amino acids, with 43 N-terminal residues translated from a shifted frame of E1 due to the 64-nt deletion and 512 C-terminal residues from the normal frame. In the 3'copy, the truncated ORF E1 encodes 445 amino acids, with the 430 N-terminal residues translated from E1 sequence, 4 residues from the partial E5 sequence, and 11 C-terminal residues from cellular DNA. The sequence of the complete ORF E1 of HPV68b-ME180 was assembled through combination of the E1 sequences in 5' and 3' copies. The assembled intact ORF E1 contains 1923 bp encoding 640 amino acids, the same numbers as ORF E1 of the published HPV68a (DQ080079).

6. The HPV68b(int)-ME180 contains no intact ORF E2. The ORF E2 in the 5'copy is truncated due to integration and E2 is completely deleted in the 3'copy. The ORF E2 of the 5'copy encodes 134 amino acids, with 120 N-terminal residues translated from the E2 gene and 14 C-terminal residues from cellular DNA. By comparison with the complete ORF E2 of HPV68a (DQ080079), 754 bp are missing.

7. The URR and ORFs E6 and E7 are present in intact form in both 5' and 3' copies.

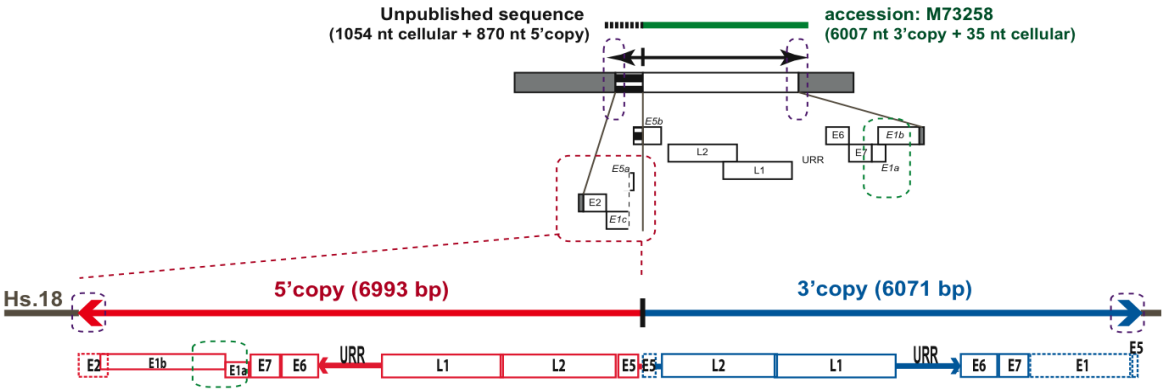


Figure 2.3: Complete structure of HPV68b(int)-ME180. The previously sequenced clone AA13.1 (Reuter et al, 1991) is shown at the top for comparison. The top green line represents the published sequence (accession M73258) containing the 3'copy of AA13.1, whereas the dashed black line represents the sequence of the 5'copy (Elisabeth Schwarz, unpublished). The locations of ORFs in the integrated HPV68b are shown at the bottom. Solid-lined red and blue boxes indicate the complete coding region (from ATG to stop codon) of each ORF, the dashed red and blue boxes the truncated ORFs. The red and blue arrows indicate the direction of transcription for the 5'copy and 3'copy, respectively. The viral-cellular junctions in the 5' and 3' copies in clone AA13.1 and in the new complete structure are identical (dashed purple circles). The E1a-E1b area of AA13.1 is present in the 5'copy of the complete structure (dashed green circle). AA13.1 contains 6007 bp HPV68b DNA in the 3'copy due to the presence of the 64-bp deletion. The positions of all ORFs are given in Table 2.1.

Table 2.1: Sequence organization of *HPV68b(int)-ME180*

Position	Length (bp)	Description [a]	Remark	HPV68b-CIN2 [b]
1-186	186	flanking cellular DNA of chrom.18, pos. 45578156-45578341 (NC_000018.9)	cellular DNA	
182-186		cellular-viral junction (5 bp overlap)		
136-540	405	truncated ORF E2 [c]	5'copy (6993 bp)	3031-2673
467-2002/2134	1536/1668	ORF E1b [c]		2746-1211
2001/2002		borders of 64-bp deletion		1212/1147
1975-2325	351	ORF E1a [c]		1238-824, (del 1211-1148)
2332-2664	333	complete ORF E7		817-485
2672-3148	477	complete ORF E6		477-1
3149-3957	809	URR		7836-7026
3958-5475	1518	complete ORF L1		7025-5508
5456-6865	1410	complete ORF L2		5527-4118
6912-7133	222	complete ORF E5		4071-3850
7134-7174	41			3849-3809
7174/7175		junction between 5'copy and 3'copy		
7175-7354	180	truncated ORF E5	3'copy (6071 bp)	3892-4071
7401-8810	1410	complete ORF L2		4118-5527
8791-10308	1518	complete ORF L1		5508-7025
10309-11119	811	URR		7026-7836
11120-11596	477	complete ORF E6		1-477
11604-11936	333	complete ORF E7		485-817
11943-13280	1338	truncated ORF E1		824-2112
13231		3'end of truncated E1		2112
13232-13245	14	part of ORF E5		3927-3914
13245		viral-cellular junction (1 bp overlap)		
13245-13286	42	flanking cellular DNA of chrom.18, pos. 45578364-45578405 (NC_000018.9)	cellular DNA	

(a) Because the 5'copy is in opposite orientation, position numbering of the ORFs is from stop codon to start codon. In the 3'copy, positions are opposite.
(b) Complete HPV68b genome isolated from a CIN2 lesion, described in section 2.1.4.
(c) See text and Figure 2.4 for explanation.

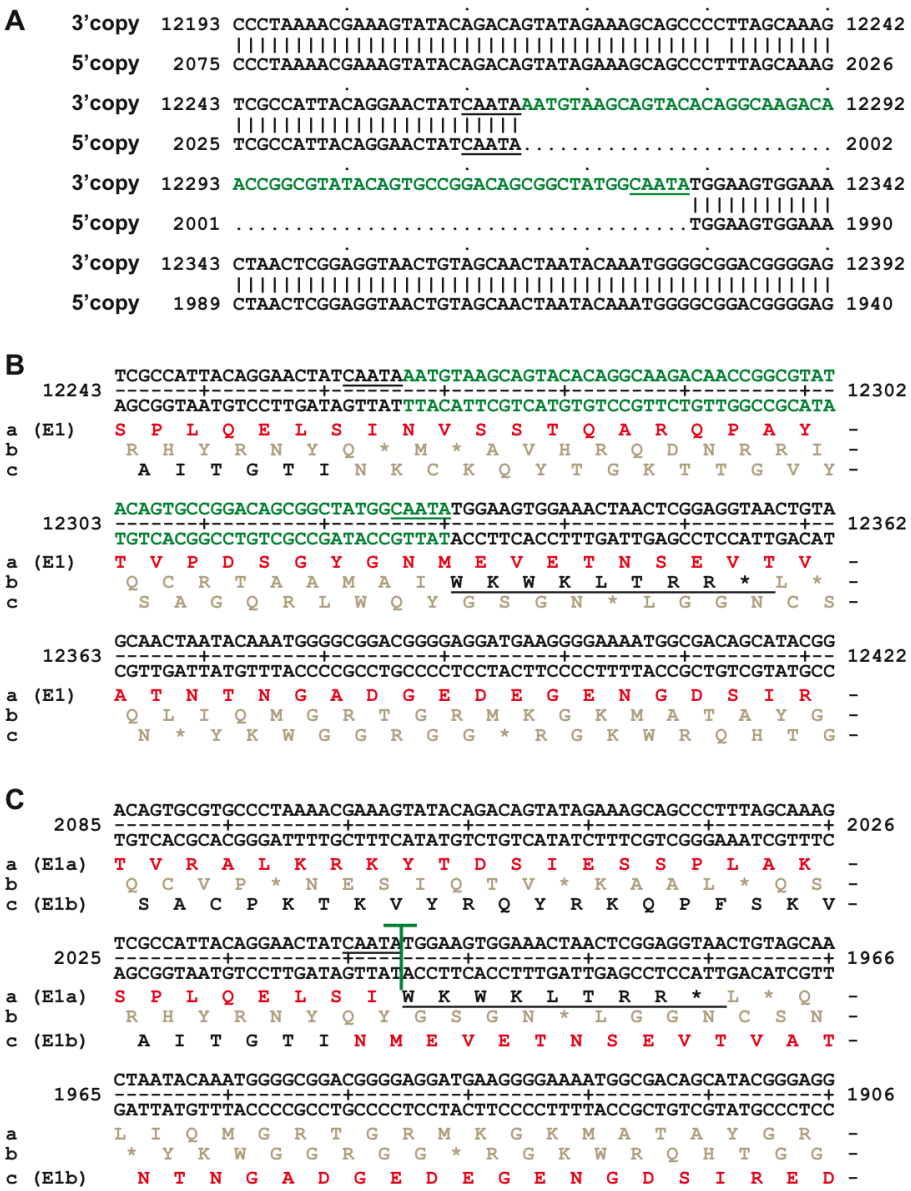


Figure 2.4: Internal 64-bp deletion and frameshift in ORF E1 of HPV68(int)-ME180. Panel A: Partial sequence comparison of E1 gene in the 5'copy and 3'copy. Green characters indicate the 64 bp deleted in the 5'copy. The repeated CAATA sequence at the border of the 64 bp deletion is underlined. **Panel B**: Partial nucleotide sequence of ORF E1 in the 3'copy, covering pos. 12243-12422. The 64-bp sequence deleted in the 5'copy is shown in green. The E1 protein is translated from frame a. The underlined amino acids in frame b indicate the frameshifted residues in ORF E1a of the 5'copy. **Panel C**: Partial nucleotide sequence of ORFs E1a and E1b in the 5'copy covering pos. 2085-1906, shown with translated amino acids. The green T-marker indicates the position of the 64-bp deletion. ORF E1a encodes 116 amino acid translated from frame-a with 8 C-terminal residues (underlined) originated from another frame in intact ORF E1 (see Panel C). ORF E1b is translated from frame-c. Red-color indicates amino acids originating from the E1 reading frame.

2.1.2 Determination of the complete integrated HPV68b sequence in ME180R

The cell line ME180R was derived from ME180 cells by selection of subclones resistant to growth inhibition by tumor-necrosis-factor-alpha (TNFalpha) (Pfreundschuh et al, 1989). Comparing the genomic organization of integrated HPV68b in ME180 and ME180R by Southern hybridization, large deletions of the integrated HPV68b DNA in ME180R became apparent (Figure 2.5; Schwarz, E, Reuter, S and Springer, M, unpublished data). Additional mapping and hybridization experiments had been performed in the past, but the exact structure of the integrated HPV68b in ME180R had not been determined. This task was performed as part of this PhD work.

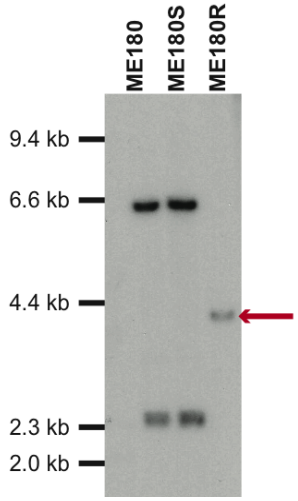


Figure 2.5: Genomic Southern hybridization of ME180, ME180R and ME180S. BamHI-digested genomic DNA from cell lines ME180, ME180R (TNFalpha resistant) and ME180S (TNFalpha sensitive) were hybridized with an HPV68b DNA probe (L1-L2-URR-E6-E7). The hybridization pattern of ME180R is different from the other two cell lines, showing a single fragment smaller than 4.2 kb (red arrow).

Based on the differences in the Southern hybridization patterns (Figure 2.5) together with the known locations of BamHI sites in the complete integrated HPV68b(int)-ME180, it was deduced that about 6200 bp of HPV68b DNA were deleted in ME180R. To determine the exact structure of the integrated HPV68b sequence in ME180R, five fragments (R_PCR_1 to R_PCR_5) were amplified using Roche's Expand Long Template PCR System. The primers were selected based on the previously determined HPV68b(int)-ME180 sequence. Table 2.2 shows the primer positions and the product sizes. The PCR products were cloned into vector pCR4-TOPO (Invitrogen, Karlsruhe), and completely sequenced. The sequences of the five fragments were assembled based on the overlapping areas (Figure 2.6).

The assembled nucleotide sequence, named HPV68b(int)-ME180R, consists of 6544 nucleotides and is shown in Appendix A1. The genetic organization of the integrated HPV68b in ME180R in comparison to ME180 is shown in Figure 2.6 and Table 2.3. Important features are explained in the following:

1. Two large deletions of HPV68b are present, one in the 5'copy (2316-bp deletion) and another in the 3'copy (4679-bp deletion). The sequences at the deletion boundaries are shown in Figure 2.7.
2. In the 5'copy, ORF L1 is completely deleted and ORF L2 partially deleted. In the 3'copy, the complete ORFs L2, L1, URR and E6, and partial E7 ORF (covering 273 bp from the ATG start codon) were deleted. The intact URR, together with the complete ORFs E6 and E7 are present in the 5'copy. Therefore, the oncogenes E6/E7 can only be expressed from the 5'copy.
3. Sequence comparison demonstrated that the region containing the head-to-head viral junction is reversed (Figure 2.6). The complete ORF E5 is present. However, in comparison to HPV68b(int)-ME180, it contains two point mutations, one of them changing a TGG into a TAG premature stop codon (Figure 2.8). Therefore, the E5 protein is shortened by 12 amino acids.
4. The URR in the 5'copy is shortened by four bp at the 5'end (Figure 2.7), and contains a 2-bp insertion (Figure 2.19), when compared with the URR of the HPV68b(int)-ME180 5'copy.

Table 2.2: Primers and products for PCR amplification of the integrated HPV68b in ME180R.

PCR name*	Primer**	5'end position in HPV68b(int)-ME180	ME180 PCR calculated product size	ME180R PCR product size***
R_PCR_1	ME-01	-247 (F)	2456 bp	~2.5 kb
	ME-6805	2209 (R) and 12059 (F)		
R_PCR_2	ME-7094	1924 (F) and 12408 (R)	5318 bp	~2.2 kb
	ME-7810	7091 (F)		
R_PCR_3	ME-1349	3775 (F) and 10491 (R)	9512 bp	~2.5 kb
	ME-7972	13286 (R)		
R_PCR_4	ME-7718	6999 (R) and 7267 (F)	5076 bp (5'copy) 5142 bp (3'copy)	~1.3 kb
	ME-7094	1924 (F) and 12408 (R)		
R_PCR_5	ME-6651	2367 (R) and 11901 (F)	1386 bp	~1.4 kb
	ME-7972	13286 (R)		

* Genomic DNA of ME180R was used as template.
** Primer sequences are given in Materials and Methods.
*** Estimated sizes from agarose gel.

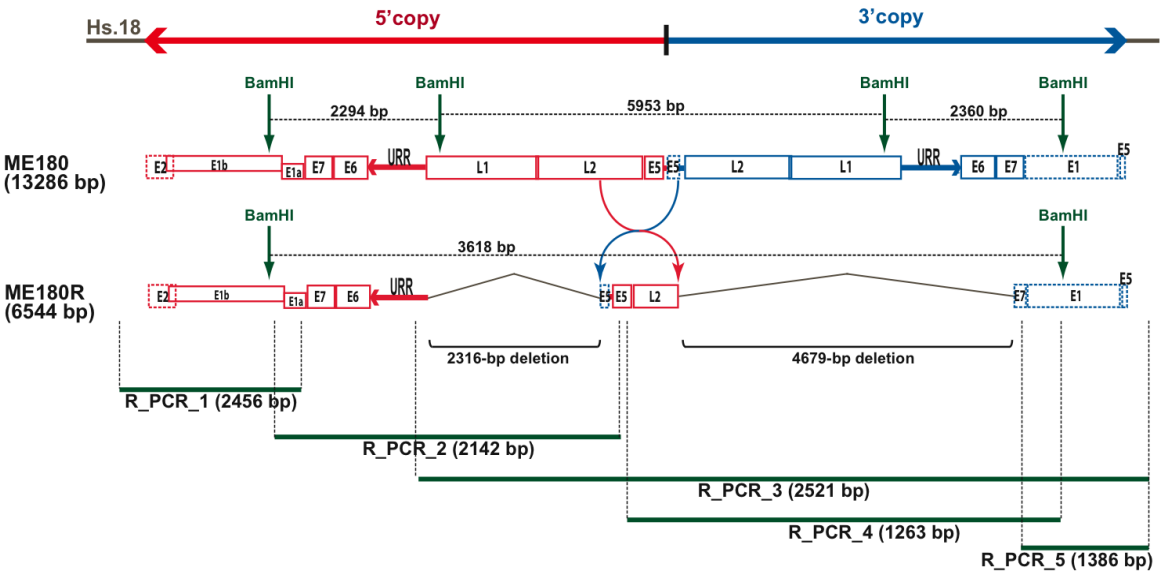


Figure 2.6: Structure of HPV68b(int)-ME180R in comparison with HPV68b(int)-ME180. The two large deletions in the 5'copy and 3'copy are shown. As indicated by the arrows, the viral-viral junction area (middle) is reversed in ME180R. BamHI restriction sites are indicated by green vertical arrows, and the sizes of the BamHI restriction fragments deduced from the nucleotide sequences are given. At the bottom part, regions covered by the five PCR products of ME180R are indicated together with the PCR sizes.

Table 2.3: Sequence organization of *HPV68b(int)-ME180R in comparison with HPV68b(int)-ME180.*

HPV68b(int)-ME180		HPV68b(int)-ME180R	Retained and deleted regions of ME180R*
Positions	Description [a]	Positions	Positions
1-186	flanking cellular DNA of chrom.18 pos. 45578156-341 (NC_000018.9)	248-433	retained: 3955 bp ME180R 248-4202 [b] ME180 1-3953
182-186	cellular-viral junction (5 bp overlap)	429-433	
136-540	truncated ORF E2	383-787	
467-2002/2134	ORF E1b	714-2249/2381	
2001/2002	borders of 64-bp deletion	2248/2249	
1975-2325	ORF E1a	2222-2572	
2332-2664	complete ORF E7	2579-2911	
2672-3148	complete ORF E6	2919-3395	
3149-3957	URR	3396-4202 [b]	
3958-5475	complete ORF L1	deleted: ME180 3954-6269 (2316 bp)	
5456-6865	complete ORF L2	5133-4538 (inversed and truncated)	retained: 928 bp ME180R 4206-5133 ME180 7197-6270
6912-7133	complete ORF E5	4491-4270 (inversed)	
7174/7175	junction between 5'copy and 3'copy	4229/4228 (inversed)	
7175-7354	truncated ORF E5	4228-4206 (inversed and truncated)	
7401-11596	complete ORFs L2, L1, URR, E6	deleted: ME180 7198-11876 (4679 bp)	
11604-11936	complete ORF E7	5135-5194 (truncated)	retained: 1410 bp ME180R 5135-6544 ME180 11877-13286
11943-13280	truncated ORF E1	5201-6538	
13231	3'end of truncated E1	6489	
13232-13245	part of ORF E5	6490-6503	
13245	viral-cellular junction (1 bp overlap)	6503	
13245-13286	flanking cellular DNA of chrom.18 pos. 45578364-405 (NC_000018.9)	6503-6544	

* Compared to HPV68b(int)-ME180.
(a) see footnote of Table 2.1.
(b) contains a 4-bp deletion at the 5'end of the URR and a 2-bp insertion (see text and Figure 2.7).

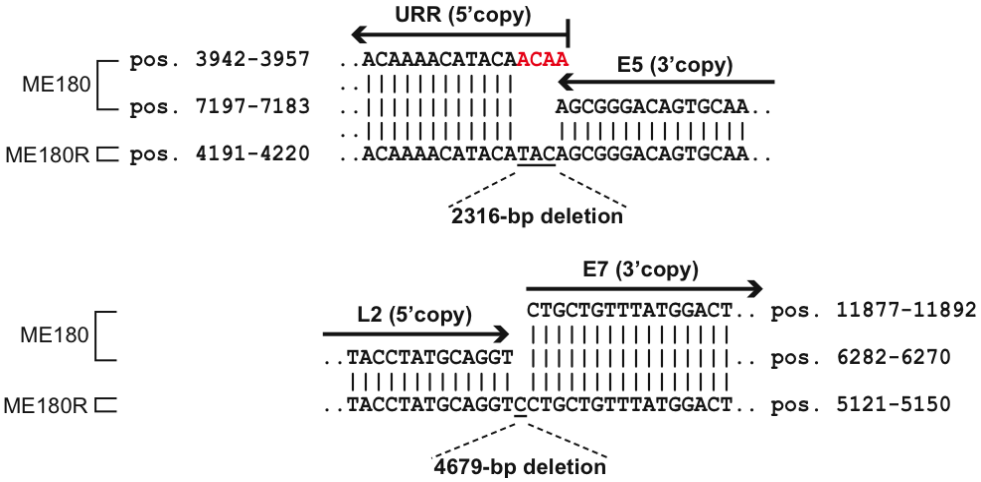


Figure 2.7: Nucleotide sequences at the deletion boundaries in ME180R. The top and bottom parts show the nucleotide sequence comparisons between the integrated HPV68b in ME180 and ME180R at the 5'copy and 3'copy deletion areas, respectively. The arrows represent the direction of transcription. The first four basepairs of the URR are deleted in ME180R (red). Three additional nucleotides (underlined TAC) are present in the 5'copy deletion area of ME180R, and a single additional nucleotide (underlined C) in the 3'copy deletion area.

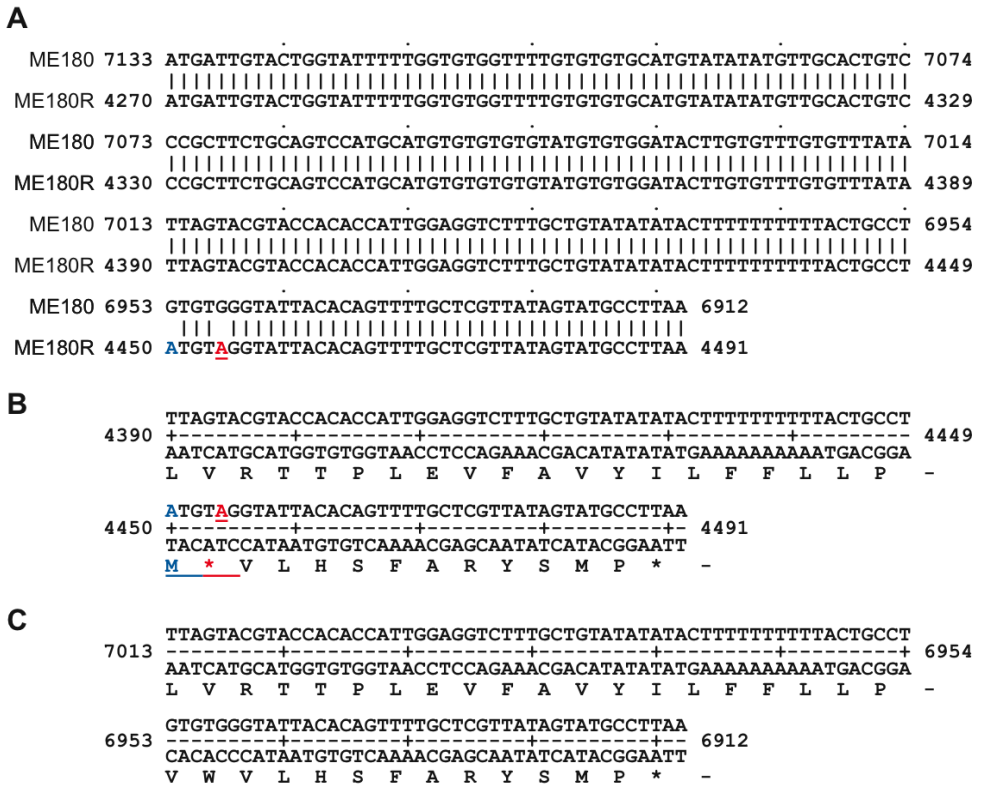


Figure 2.8: ORF E5 with premature stop codon in ME180R. Panel A: ORF E5 of ME180R is shown in comparison with ME180 5'copy. Point mutations in ME180R are colored in blue and red. **Panel B**: Partial translation of ORF E5 of ME180R containing missense (blue) and nonsense (red) mutations. **Panel C**: Partial translation of intact ORF E5 of HPV68b(int)-ME180 5'copy.

2.1.3 Isolation and analysis of TNFalpha-resistant cells from ME180

The complete sequence of the integrated HPV68b in the TNFalpha-resistant ME180R cells revealed that the HPV68b DNA is highly altered, especially by two large deletions, in comparison to the integrated HPV68b in the parental ME180 cells. Therefore, it was the question whether these two large deletions might contribute to the TNFalpha-resistant characteristics of ME180R cells. To investigate this issue, TNFalpha-resistant cells were independently isolated from the parental ME180 cells. The selection procedure is described in Figure 2.9. Two TNFalpha-resistant variants, ME180-2A and ME180-3A, were obtained.

The two newly derived variants were tested in a cytotoxicity assay with different concentrations of TNFalpha to determine their resistance to TNFalpha, compared with ME180 and ME180R cells. The surviving cells (cell viability) of the four cell populations after TNFalpha treatment were stained with crystal violet and their absorbance measured at wavelength 560 nm (Table 2.4). The calculated percentage values of cell viabilities are shown in a graphical form in Figure 2.10. Compared to the parental ME180 cells, the ME180R and the two isolated variants ME180-2A and ME180-3A are highly resistant to TNFalpha. Both variants showed comparable resistance to TNFalpha as ME180R.

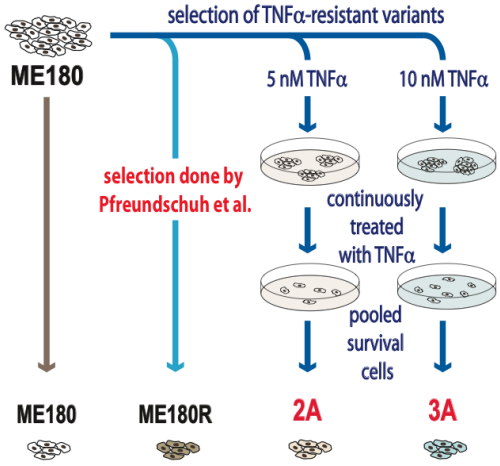


Figure 2.9: Isolation of TNFalpha-resistant variants from parental ME180 cells. ME180 cells were continuously treated with two different concentrations of TNFalpha for 18 weeks. Variants ME180-2A and ME180-3A were obtained from the pooled cells after the treatment with 5 nM and 10 nM TNFalpha, respectively. ME180R was previously selected (Pfreundschuh et al, 1989).

If the two large deletions in the integrated HPV68b in ME180R contribute to the TNFalpha-resistance phenotype, then the two newly isolated TNFalpha-resistant variants should contain some deletions as well. A genomic Southern hybridization was performed to compare the structures of integrated HPV68b in ME180, ME180R, M180-2A and ME180-3A. The results are shown in Figure 2.11. The HPV68b hybridization patterns of variants ME180-2A and ME180-3A were identical to that of ME180, demonstrating that deletions are not present. From these data, it can be concluded that the two large deletions in ME80R are not causally linked to the TNFalpha-resistant growth phenotype.

Table 2.4: Cell viability measurements for TNFalpha resistance of ME180, ME180R, ME180-2A and ME1803A.

A: Absorbance intensities of viable cells after treatment with TNFalpha. [(a)]

	ME180			ME180R			2A			3A			
0 nM	0,698	0,860	0,896	0,522	0,541	0,441	0,343	0,357	0,344	0,451	0,462	0,409	
2.5 nM	0,107	0,100	0,120	0,570	0,608	0,513	0,389	0,410	0,380	0,717	0,630	0,609	5000 cells/well
5 nM	0,107	0,111	0,109	0,574	0,622	0,506	0,422	0,409	0,430	0,742	0,649	0,685	
10 nM	0,100	0,106	0,104	0,638	0,623	0,517	0,389	0,403	0,394	0,709	0,639	0,609	
0 nM	1,018	1,021	0,990	0,583	0,581	0,477	0,390	0,394	0,372	0,703	0,578	0,485	
2.5 nM	0,127	0,109	0,113	0,569	0,518	0,395	0,327	0,288	0,349	0,512	0,774	0,000	10000 cells/well
5 nM	0,131	0,140	0,139	0,591	0,606	0,484	0,381	0,360	0,360	0,556	0,571	0,516	
10 nM	0,111	0,105	0,121	0,610	0,649	0,551	0,400	0,418	0,386	0,640	0,591	0,425	

B: Average absorbance intensities. [(b)]

[average]	ME180	ME180R	2A	3A
0 nM	0,914	0,524	0,367	0,515
2.5 nM	0,113	0,556	0,357	0,540
5 nM	0,123	0,564	0,394	0,620
10 nM	0,108	0,598	0,398	0,602

C: Percentage of absorbance intensities. [(c)]

[%]	ME180	ME180R	2A	3A
0 nM	100	100	100	100
2.5 nM	12	106	97	105
5 nM	13	108	107	120
10 nM	12	114	109	117

(a) The cells, seeded at densities of 5000 or 10000 cells/well, were treated in triplicates for each TNFalpha concentration. Gray highlighted value is excluded from the average and percentage calculation.
(b) Each value is an average of the same cell population, treated with the same TNFalpha concentration.
(c) Percentage based on cells in absense of TNFalpha (0 nM) for each cell population.

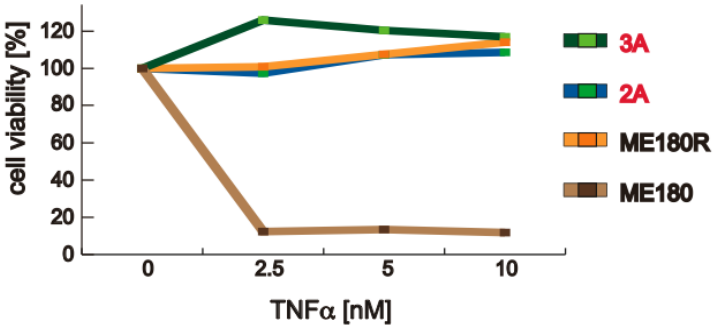


Figure 2.10: Cell viability for TNFalpha resistance of ME180, ME180R, ME180-2A and ME180-3A. The cell viabilities, determined as percentages of the number of cells after TNFalpha treatment (2.5, 5 and 10 nM) over the control (0 nM) of each cell population, are shown. The values of this plot are indicated in Table 2.4, Panel C.

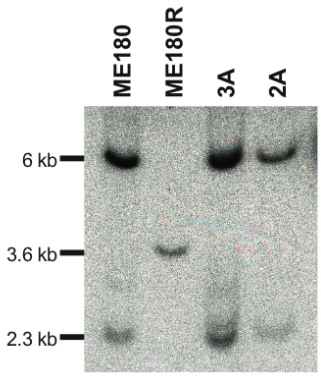


Figure 2.11: Southern hybridization analysis of integrated HPV68b in ME180, ME80R, ME180-2A and ME180-3A. BamHI-digested genomic DNA from ME180, ME180R, 2A and 3A were hybridized with an HPV68b DNA probe (L1-L2-URR-E6-E7). The sizes of the hybridized fragments are indicated.

2.1.4 Cloning and sequencing of a complete HPV68b genome from a CIN2 lesion

Because a complete sequence of HPV68b could not be assembled from the two incomplete copies of HPV68b(int)-ME180 and also was not available from the literature, it was aimed to clone and sequence a complete HPV68b genome from a cervical lesion. Through the DKFZ-CGE collaboration, an HPV68-positive DNA sample of a CIN2 lesion was provided. The DNA was amplified with phi29 DNA polymerase, in the process called multiple displacement amplification (MDA), because the original DNA amount was not enough for further analysis. The original DNA of this CIN2 sample is referred to as original-CIN2, and the amplified product as MDA-CIN2.

To determine whether the CIN2 sample contained HPV68b and not HPV68a, a partial sequence of ORF L1 was determined. Sequence differences between HPV subtypes are in the range of 2-10% in the L1 gene, whereas variants differ only by less than 2% (de Villiers et al, 2004). The 3064-bp region covering the HPV68 ORFs L2-L1 was amplified from MDA-CIN2 DNA (Figure 2.12). The product was cloned and sequenced from one end with the reverse primer as the sequencing primer. The sequence of 798 bp covering part of ORF L1 (Figure 2.12) was used for comparison with the corresponding region of HPV68b(int)-ME180 and HPV68a (DQ080079). The 798-bp sequence of the CIN2 sample showed 99% sequence identity to HPV68b(int)-ME180 3'copy (793 out of 798 nucleotides) and 91% to the HPV68a (732 out of 798 nucleotides). This result demonstrated that the CIN2 sample contains subtype HPV68b. In addition, analysis by the HPV Luminex assay (Schmitt et al, 2006) indicated that the CIN2 DNA contains HPV68b

as single HPV type. Therefore, the CIN2 DNA was used as source for isolation of the complete HPV68b genome.

To amplify the complete HPV68b genome from the CIN2 sample, two primer pairs were initially used (Figure 2.13, panel A) with MDA-CIN2 as template, using the Expand Long Template PCR System. The two primer pairs locate about 600 bp apart. From both primer pairs, products of ~8 kb were obtained (data not shown). Attempts to directly clone these two fragments into a TA vector failed. However, it was discovered during the cloning attempts that an EcoRI restriction site is present in the HPV68b-CIN2 genome (Figure 2.13, panel A). After EcoRI cleavage of the ~8 kb PCR products, the EcoRI site could be located in the L1 gene. An 851-bp area covering the suspected EcoRI site was amplified, cloned and sequenced. The sequence revealed an EcoRI site 390 bp downstream of the L1 start codon.

New primers flanking the EcoRI site were designed based on the 851-bp sequence, and were used to amplify the complete HPV68b-CIN2 genome from MDA-CIN2 DNA (Figure 2.13, panel B). The expected ~8 kb product was obtained, but also a smaller product of ~7 kb (Figure 2.13, panel B). The 8 kb and 7 kb fragments were digested with EcoRI for cloning into a pBluescript-KS-Plus (pBS) vector at the EcoRI site. The first attempt failed, probably due to the low concentration of the inserts. Therefore, the EcoRI-digested 8 kb and 7 kb fragments were re-ligated to produce circular DNA. These circular DNAs were amplified with phi29 DNA polymerase, digested with EcoRI and successfully cloned into the pBS vector at the EcoRI site.

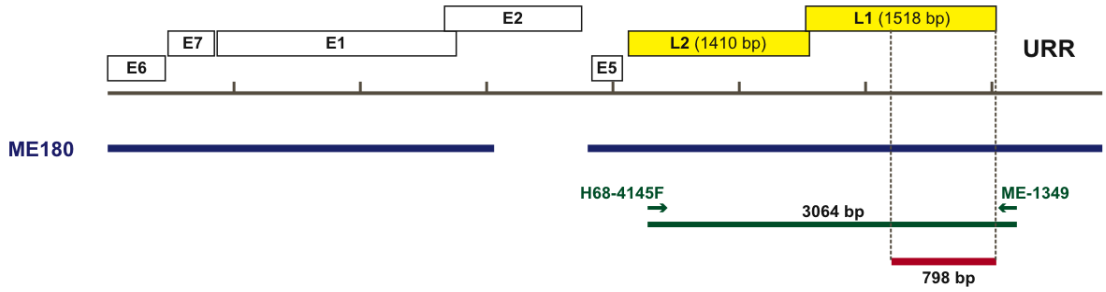


Figure 2.12: Determination of the HPV68 subtype in the CIN2 sample. The HPV68 genome is shown at the top. The genome parts present in ME180 are indicated by the blue lines. The L2-L1 PCR fragment amplified from MDA-CIN2 DNA is shown by the green line. The primers H68-4145F and ME-1349 bind at 1363 bp upstream from the L1 ATG start codon and at 189 bp downstream of the L1 stop codon, respectively. The reverse primer ME-1349 was also used as sequencing primer and a 798-bp partial L1 sequence was obtained (red). The 798-bp sequence of CIN2 was compared with HPV68b-ME180 and HPV68a for subtype determination. The primer sequences are shown in Materials and Methods.

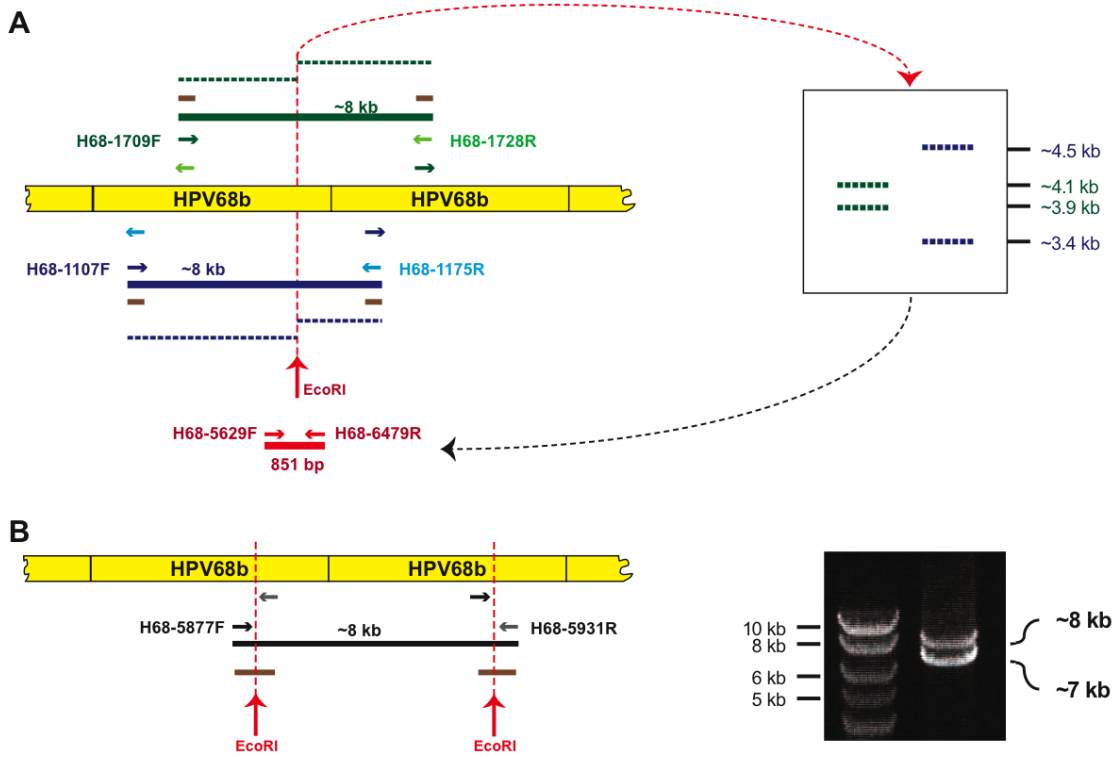


Figure 2.13: Amplification strategies for cloning the complete HPV68b genome from the CIN2 sample. Whole genomic amplification with phi29 DNA polymerase results in concatameric HPV68b DNA (yellow bar) which was use as template molecule for PCR of CIN2 DNA. **Panel A**: Two sets of primer pairs (green and blue arrows) were used for PCR, giving rise to two overlapping full-length genomes (green and blue bars). Primer pair H68-1709F and H68-1728R also produced a 20-bp product, and primer pair H68-1107F and H68-1175R a 69-bp product, indicated by brown bars. The EcoRI site is indicated by the red vertical arrow. On the right, the EcoRI restriction fragments of the two PCR products separated on an agarose gel are schematically shown with their estimated sizes. The 851-bp area (red bar) covering the suspected EcoRI site was amplified with a primer pair binding at 122 bp and 972 bp downstream of the ATG of L1 start codon. **Panel B**: A primer pair (black arrows) flanking the EcoRI site was used for PCR to amplify the full-length genome (black bar). The primers also produce a 54-bp product (brown bars). Agarose gel electrophoresis of the PCR product showed two products of ~8 kb and ~7kb (right lane). Primer sequences are available in Materials and Methods.

For both 8 kb and 7 kb HPV68b-CIN2 genomes, one positive clone was completely sequenced by Sanger method (A. Hunziker, DKFZ Core Facilities). The nucleotide sequences of the 8 kb and the 7 kb genomes, HPV68b-CIN2 and HPV68b-CIN2-Del, are shown in Appendix A2. The complete HPV68b-CIN2 genome contains 7836 bp. Figure 2.14 shows the HPV68b-CIN2 genome in a circular form, and Table 2.5 the genomic organization. The sequence of the HPV68b-CIN2-Del genome contains 6611 bp and carries a 1229-bp deletion in the E1 gene (Figure 2.15). This 1.2 kb deletion, located between nucleotide positions 1378 and 2608 corresponding to amino acid residues 186 to 595, leads to a frameshift and a premature stop codon at amino acid residue 186. In addition to the 1229-bp deletion, the HPV68b-CIN2-Del genome carries two 1-bp

insertions, one 2-bp insertion, and 11 mutations (Table 2.6). The two single nucleotide insertions in ORFs L1 and L2 would lead to frameshifts. However, these nucleotide differences were not verified by independent cloning/sequencing. The HPV68b-CIN2 and HPV68b-CIN2-Del genomes are illustrated in comparison with HPV68b-ME180 (Figure 2.16).

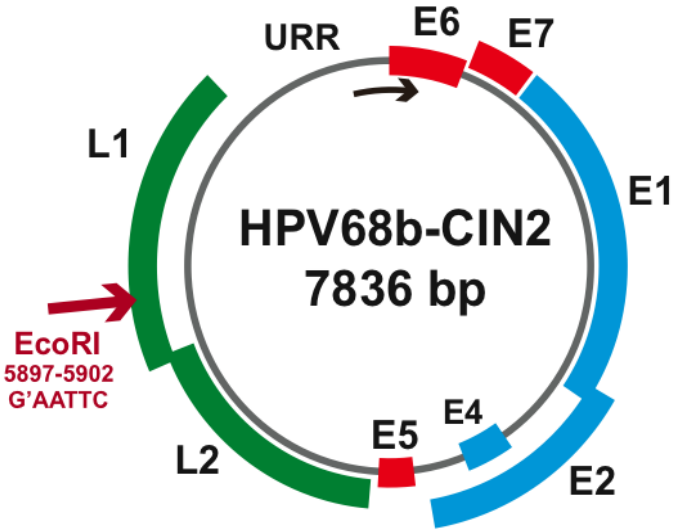


Figure 2.14: The complete HPV68b-CIN2 genome in circular form. The nucleotide positions of each gene and the encoded amino acid residues are given in **Table 2.5**. The ORFs are shown as red, blue and green bars. The single EcoRI site is indicated by the red arrow. The black arrow indicates the direction of transcription.

Table 2.5: Sequence organization of the complete HPV68b-CIN2 genome.

Gene	ORF* positions	Length (bp)	Protein (aa residues)
E6	1-477	477	158 aa
E7	485-817	333	110 aa
E1	824-2746	1923	640 aa
E2	2673-3785	1113	370 aa
E4	3268-3552 **	285	94 aa
E5	3850-4071	222	73 aa
L2	4118-5527	1410	469 aa
L1	5508-7025	1518	505 aa
URR	7026-7836	811	-

The first nucleotide of the E6 ATG start codon was taken as position 1 for sequence numbering.
* Positions from ATG to stop codon.
** ORF of E4 gene does not start with ATG. It was allocated by comparison with the ORF E4 of HPV68a (DQ080079), with the first position located after the upstream stop codon.

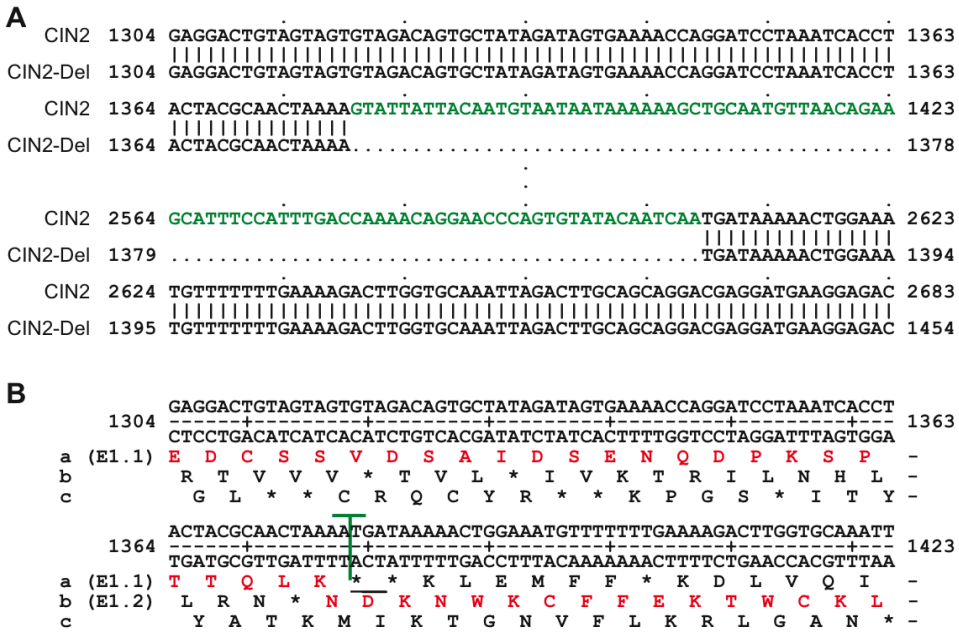


Figure 2.15: Deletion of 1229 bp in ORF E1 of HPV68b-CIN2-Del. Panel A: Nucleotide sequence comparison between HPV68b-CIN2 and HPV68b-CIN2-Del genomes in the left and right deletion boundaries. The nucleotides not present in HPV68b-CIN2-Del are in green. **Panel B**: Partial nucleotide sequence and translated amino acids of HPV68b-CIN2-Del genome. The 1229-bp deletion (green T-mark) leads to a premature stop codon (underlined) immediately following the deletion site. Red color indicates amino acids translated from normal ORF E1.

Table 2.6: Nucleotide alterations of HPV68b-CIN2-Del in comparison to HPV68b-CIN2.

Positions based on HPV68b-CIN2	Alterations in HPV68b-CIN2-Del	Locations	Effect on proteins
73	T>C	ORF E6	silent mutation
1024	A>G	ORF E1	silent mutation
1379-2607	1229 bp deleted	ORF E1	premature stop codon
3065	T>C	ORF E2	silent mutation
3378	C>T	ORF E2	missense mutation
4670	C>T	ORF L2	missense mutation
5022^23	A-insert	ORF L2	frameshift
5176	C>T	ORF L2	silent mutation
5837^38	G-insert	ORF L1	frameshift
6366	G>A	ORF L1	missense mutation
7307^08	TA-insert	URR	
7486	C>T		
7648	T>C		
7698	T>C		
7778	G>A		

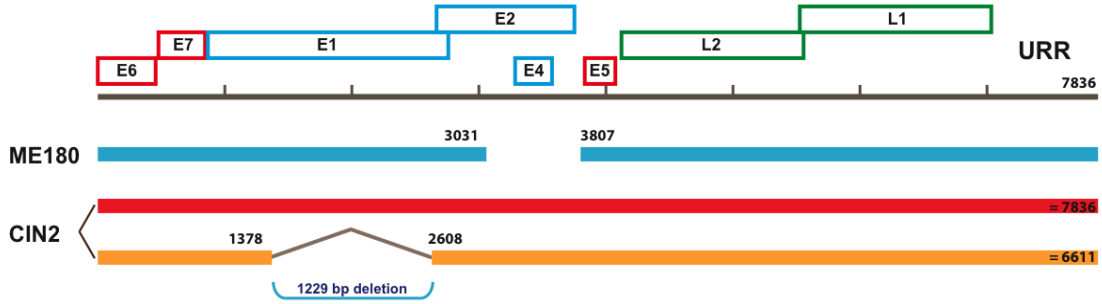


Figure 2.16: HPV68b genomes in ME180 and CIN2. The complete HPV68b genome with the ORFs based on the HPV68b-CIN2 sequence is shown at the top. The genome coverage of the integrated HPV68b in ME180 is shown by the blue lines. The red and orange lines depict the complete and partially deleted HPV68b-CIN2 genomes. The nucleotide positions flanking the deleted regions are given.

Because the HPV68b-CIN2 genomes were obtained through several steps of amplification, it was examined whether the HPV68b-CIN2-Del genome is an artifact or indeed present in the original CIN2 lesion. To clarify this, the area covering the 1.2-kb deletion was amplified from the original CIN2 DNA and from the MDA-CIN2 DNA. A 2.1-kb PCR product was expected from the complete genome, and a 0.9-kb product from the deleted genome. As shown in Figure 2.17, the 0.9-kb PCR product band could be identified in both MDA-CIN2 and original CIN2 DNA. Based on the product amounts of the 2.1-kb and 0.9-kb bands, it seemed possible that the complete HPV68b-CIN2 genome is present in much lower amount in the CIN2 DNA than the partially deleted HPV68b-CIN2-Del genome.

Previously, we had assumed that the HPV68b-CIN2 genome is present as episome in the CIN2 lesion. However, this assumption became disputable after isolation of the HPV68b-CIN2-Del genome. An episomal HPV genome relies on a functional E1 protein for replication and maintenance in the infected cells. Therefore, the question was addressed whether the HPV68b-CIN2-Del genome is present as integrated or episomal DNA. A genomic Southern hybridization was performed using the MDA-CIN2 DNA because of the limitation of the original DNA. EcoRI (single cut) and BamHI (double cut) were used for digestion, before hybridization with the complete HPV68b-CIN2 probe. The hybridization results are shown in Figure 2.18.

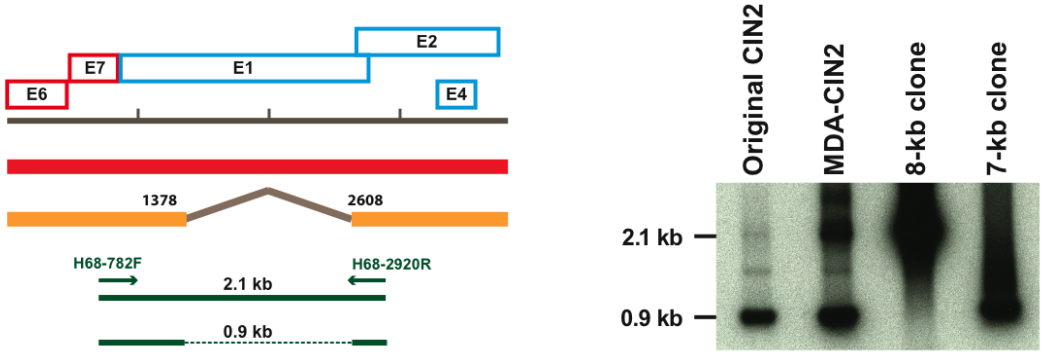


Figure 2.17: Proof of presence of the HPV68b-CIN2-Del genome in the original CIN2 DNA. The PCR covering the deleted region of the HPV68b-CIN2-Del genome is shown on the left. Red and orange lines represent the complete HPV68b-CIN2 and partially deleted HPV68b-CIN2-Del, respectively. The two expected PCR products (green lines) from both HPV68b genomes are indicated with their sizes. The primers are indicated by green arrows. The Southern hybridization of the PCR products from four different templates is shown on the right. Complete HPV68b-CIN2 DNA was used as probe. The 8 kb HPV68b-CIN2 ("8-kb clone") and 7 kb HPV68b-CIN2-Del ("7-kb clone") clones were used as positive controls for the 2.1-kb and 0.9-kb PCR products.

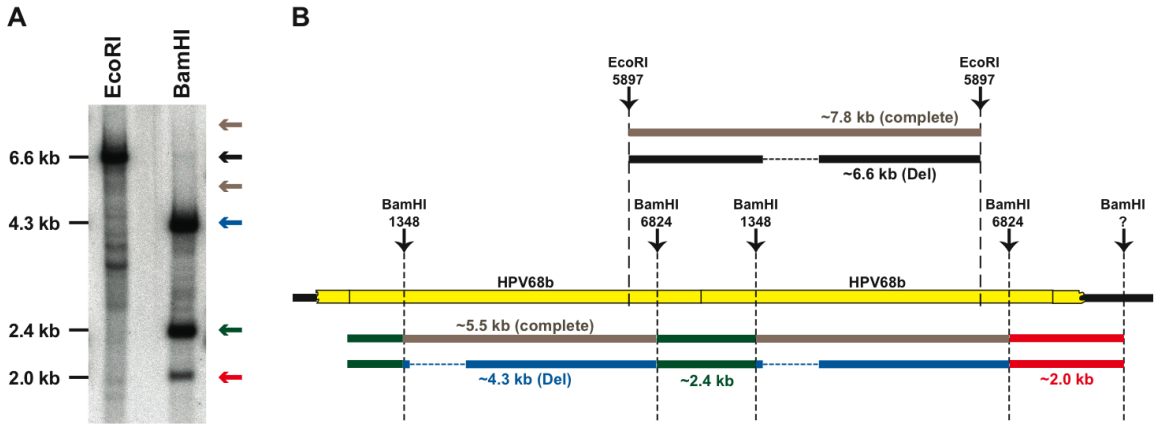


Figure 2.18: Genomic Southern hybridization of MDA-CIN2 DNA. MDA-CIN2 DNA was digested with EcoRI and BamHI, and the filter was hybridized with the complete HPV68b-CIN2 as radiolabelled probe (**panel A**). The ~6.6 kb (black arrow) fragment was expected from EcoRI-digested HPV68b-CIN2-Del genome, and the 4.3 kb (blue arrow) and 2.4 kb (green arrow) fragments from BamHI-digested HPV68b-CIN2-Del genome. The 2.0 kb fragment (red arrow) is assumed to represent a viral-cellular junction fragment. The brown arrows point to positions where the fragments of the complete HPV68b-CIN2 genome would be expected. **Panel B**: Restriction fragments obtained by BamHI and EcoRI digestion of the complete and partially deleted HPV68b genomes. If integrated, the HPV68b genome in CIN2 is assumed to be present in concatameric form (yellow bar). BamHI and EcoRI restriction sites are indicated by vertical black arrows. The color code for the restriction fragments is identical to that used for the horizonal arrows in panel A. Integration junctions are indicated at both ends of the concatameric fragment.

For the complete HPV68b-CIN2 genome, a ~7.8 kb EcoRI fragment, and two BamHI fragments of ~5.5 kb and ~2.4 kb were expected. Neither the 7.8 kb nor the 5.5 kb band were detected after hybridization. For the HPV68b-CIN2-Del genome, a ~6.6 kb EcoRI fragment, and two BamHI fragments of ~4.3 kb and ~2.4 kb were expected. All three bands were observed after hybridization (Figure 2.18). The results indicated that the complete HPV68b-CIN2 genome is present only in minute amount, compared to the HPV68b-CIN2-Del genome. An additional BamHI fragment of ~2.0 kb hybridized to the probe (Figure 2.18, panel A). This fragment could probably represent a viral-cellular junction fragment of integrated HPV68b DNA (Figure 2.18, panel B). Based on this assumption, the HPV68b-CIN2-Del genome is most likely integrated, and thus, the E1 gene is not necessary for genome maintenance. Despite best effort, experimental proof of integration could not be obtained.

Recently, the sequence of a complete HPV68b variant isolated from a patient in China has been published (Wu et al, 2009). Based on the accession number, this HPV68b variant will be named HPV68b-EU918769 in the following. To determine the sequence similarity of HPV68b-EU918769 to HPV68b-ME180 and HPV68b-CIN2, the sequences of these three genomes were compared. The complete genomes of HPV68b-EU918769 and HPV68b-CIN2 have 99% similarity. HPV68b-EU918769 carries a 2-bp deletion in the URR region compared to HPV68b-CIN2. The location of this 2-bp deletion is identical to the 2-bp deletion in the URR of HPV68b(int)-ME180 5'copy (Figure 2.19 and Table 2.7). The nucleotide sequence at this particular location is characterized by different numbers of repeats of the dinucleotide TA (4x, 5x and 6x) among the compared seven HPV68b genomes (Figure 2.19). This feature suggests that this TA-repeat maybe prone to synthesis error by DNA polymerase. Because sequence differences in ORF L1 are used to classify HPV types, subtypes and variants (de Villiers et al, 2004), the ORFs L1 of the three genomes were compared. They have identical length of 1518 bp and 99% sequence identity among each other (Table 2.8). Thus, they are three variants of subtype HPV68b.

Table 2.7: Sizes of genomes, URR and ORFs of HPV68b-ME180, HPV68b-CIN2 and HPV68b-EU918769.

	HPV68b-ME180	HPV68b-CIN2	HPV68b-EU918769
Complete length	7061*	7836	7834
E6 ORF	477	477	477
E7 ORF	333	333	333
E1 ORF	1923**	1923	1923
E2 ORF	incomplete	1113	1113
E5 ORF	222	222	222
L2 ORF	1410	1410	1410
L1 ORF	1518	1518	1518
URR	809 (5'copy) 811 (3'copy)	811	809

* 775 bp deletion in E2 (see Figure 2.16).
** Complete ORF E1 constructed by combination of the HPV68b(int)-ME180 5' and 3'copy sequences.

```
HPV68b-CIN2                 CATGTAATATATATATA..GTTCT pos.   7291-7312
                            |||||||||||||||   |||||
HPV68b-CIN2-Del             CATGTAATATATATATATAGTTCT pos.   6064-6087
                            |||||||||||||||   |||||
3'copy of AA13.1 (M73258)   CATGTAATATATATATA..GTTCT pos.   3400-3421
                            |||||||||||||||   |||||
HPV68b(int)-ME180 3'copy    CATGTAATATATATATA..GTTCT pos. 10574-10595
                            |||||||||||||||   |||||
HPV68b(int)-ME180 5'copy    CATGTAATATATATA....GTTCT pos.   3692-3673
                            |||||||||||||||   |||||
HPV68b(int)-ME180R          CATGTAATATATATATA..GTTCT pos.   3920-3941
                            |||||||||||||||   |||||
HPV68b-EU918769             CATGTAATATATATA....GTTCT pos.   7291-7310
```

Figure 2.19: Gap and insertion in URR of HPV68b genomes. Nucleotide sequences of partial URR of seven HPV68b genomes are shown in comparison. The names are indicated on the left and nucleotide positions on the right. The repeated TA is underlined.

Table 2.8: Nucleotide variations in ORF L1 of HPV68b.

	ORF L1 (positions based on HPV68b-CIN2)																No. of nucleotide differences			
Genome	5634	5811	5897	5915	6040	6188	6366	6489	6640	6662	6669	6709	6920	6929	6968		HPV68b-CIN2	HPV68b(int)-ME180 5'copy	HPV68b(int)-ME180 3'copy	HPV68b-EU918769
HPV68b-CIN2	T	T	G	C	G	A	G	C	A	C	G	A	A	C	C		-	5	8	9
HPV68b(int)-ME180 5'copy			A				A					G		T	T		5	-	5	8
HPV68b(int)-ME180 3'copy	C	C	A				A				A		G	T	T		8	5	-	11
HPV68b-EU918769			A	T	A	G	A	G	C	A				T			9	8	11	-

2.1.5 HPV68 variant analysis based on URR region

Through the DKFZ-CGE collaboration, eleven HPV68-positive DNA samples from cervical scrapes, including the CIN2 sample, were provided to us. Except HPV68b-CIN2, all other ten samples contain multiple HPV types (Table 2.9). Because the HPV genotyping cannot differentiate between HPV68 subtypes, it was the question whether the DNA samples contained HPV68b or HPV68a. Furthermore, the question was addressed whether the samples belong to already known or new HPV68b variants.

Typically, HPV variants are classified based on sequence differences of the L1 gene (de Villiers et al, 2004). Other regions of the HPV genome have also been analyzed for sequence variations. In case of HPV68, partial URR sequences of fourty-one HPV68-positive DNA samples were used to determine the phylogeny of HPV68, and sixteen variants of subtype HPV68b were identified based on the phylogenetic tree (Calleja-Macias et al, 2005).

Because the study of Calleja-Macias and coworkers provides the most extensive source for sequences of HPV68 variants, it was decided to determine the HPV68 sequences in the URR region of the 11 samples (Table 2.9) to compare them with the established 16 variants. The partial URR sequence corresponds to pos. 7279-7769 (491 bp) of HPV68b-CIN2. The sequences of the sixteen HPV68b variants were used as references. One of these variants contains sequences identical to HPV68b(int)-ME180 3'copy. In addition, the recently published complete sequence of an HPV68b variant isolated from China (accession EU918769) (Wu et al, 2009) was also included. For subtype HPV68a, the sequence of accession DQ080079 was used as reference. No further HPV68a sequence has yet been published. To obtain the URR sequences of the 10 additional samples, a region of ~773-bp (pos. 7279-7769 on HPV68-CIN2) covering this segment was amplified from all samples using primers H68-7179F and H68-128R. The PCR products were cloned and sequenced. The sequence of each sample was derived from three clones of identical sequences. The nucleotide sequences from pos. 7279-7769 of all 11 samples in Table 2.9 were compared with the seventeen HPV68b reference variants and the single HPV68a variant using ClustalW. The sequence alignment is shown in Figure 2.20. The nucleotide sequences are shown in Appendix A3.

Table 2.9: HPV68-positive cervical DNA samples.

Sample ID	Sample name	Cytology/ Histology	HPV genotyping*
Reims-06	44667	normal	16, 52, **68**, CP6108
Reims-09	46672	HSIL	16, 54, **68**, 70, IS39
Reims-12	52553	LSIL	39, 51, 52, 61, **68**, 83, CP6108
Reims-14	55178	normal	66, **68**
Reims-15	55670	LSIL	6, 16, 31, 35, 40, 53, 61, **68**, (52)
Reims-16	55954	HSIL	16, 58, **68**, (52)
Reims-18	56904	HSIL	16, 55, 56, **68**
Reims-28	64767	LSIL	16,51,66,**68**,69,73,83
Reims-31	66601	ASC-US	16,31,(52),58,59,67,**68**,72,CP6108
Reims-33	68374	LSIL	16,42,53,61,**68**
HPV68b-CIN2	CIN2	CIN2	**68b**

* HPV genotyping for the ten Reims samples was performed in Reims.

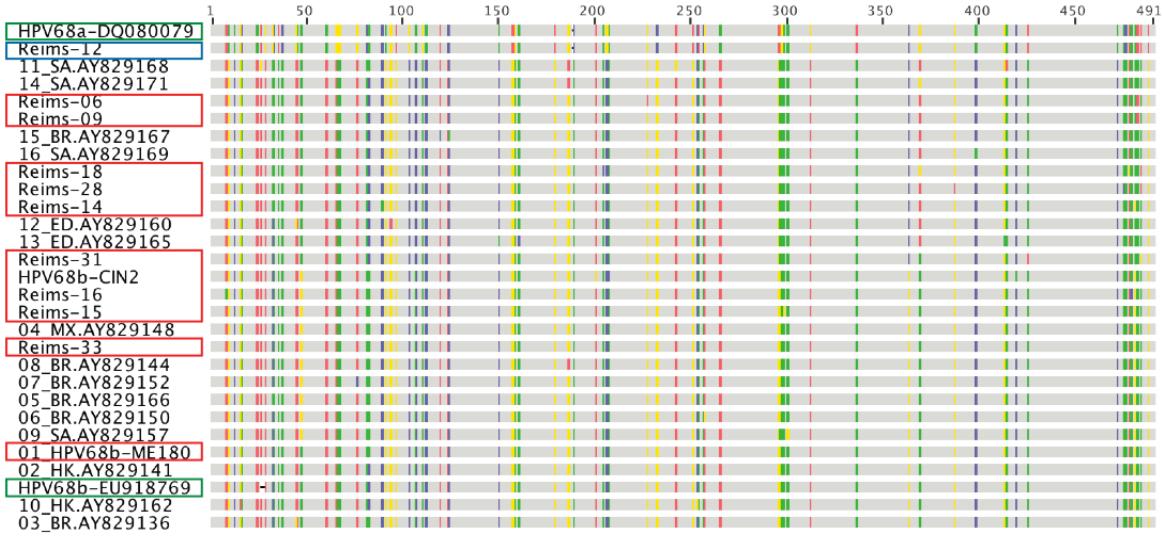


Figure 2.20: Alignment of partial URR sequences of different HPV68 isolates. The 491-bp URR segment (pos. 7279-7769 on HPV68b-CIN2 genome) of eleven clinical samples (Table 2.9) were aligned with the corresponding sequence of HPV68b-CIN2, HPV68b-EU918769 (Wu et al, 2009), HPV68a (accession DQ080079) and sixteen HPV68b variants (Calleja-Macias et al, 2005) using ClustalW. The nucleotide positions are indicated on top. The bases are highlighted at positions of sequence differences. Red represents A, blue C, yellow G, and green T. Gray area represents identical sequences. The sequence names are shown on the left. Sequences with blue or red boxed names were obtained in this PhD work. Except variant 01_HPV68b-ME180, the names of all reference HPV68b variants from Calleja-Macias et al start with a number (02-16), followed by the isolate origin (SA: South Africa, BR: Brazil, ED: Edinburg, MX: Mexico, HK: Hong Kong) and the accession number.

The 491-bp sequence of sample Reims-12 is identical to HPV68a-DQ080079, therefore it contains HPV68a. The other ten samples are HPV68b (Figure 2.20). Altogether, 44 nucleotide positions show sequence differences among the analyzed HPV68b samples (Table 2.10). Sample Reims-33 is identical to HPV68b 04_MX.AY829148. The other

samples seem to be new variants due to additional nucleotide polymorphisms compared to the most similar variants (Table 2.10).

To create a phylogenetic tree containing the new sequences with the 16 reference HPV68b variants, the Unweighted Pair Group Method with Arithmetic Mean Algorithm (UPGMA) algorithms was used. This algorithm was also applied in the earlier work (Calleja-Macias et al, 2005). The generated tree is shown in Figure 2.21. In agreement with the previous data (Calleja-Macias et al, 2005), the phylogenetic tree carries two deep dichotomic branches, representing HPV68a and HPV68b. In the HPV68b branch, the variants form two main clusters. The Chinese isolate HPV68b-EU918769 was grouped with the Hong Kong isolate 02_HK.AY829141, thus showing geographic correlation. Based on this tree, one known HPV68a variant, one known HPV68b variant and nine new HPV68b variants were identified among the eleven HPV68-positive samples from France.

Table 2.10: Nucleotide polymorphisms in the URR of HPV68b variants

DNA sample	Variant	Positions*																																									
		9	16	27	28	29	33	45	48	77	83	90	94	95	107	120	125	151	161	187	201	205	208	228	243	254	258	297	299	301	364	365	370	388	399	414	415	420	426	476	479	480	482	483	485
		7287	7294	7305	7306	7307	7311	7323	7326	7353	7361	7368	7372	7374	7385	7398	7403	7429	7439	7465	7479	7483	7486	7506	7521	7532	7536	7575	7577	7579	7642	7643	7648	7666	7677	7692	7693	7698	7704	7754	7757	7758	7760	7761	7763
HPV68b(int)-ME180 3'copy		A	G	A	T	A	C	A	T	A	T	C	G	G	T	A	C	C	T	G	A	T	T	G	A	C	A	G	T	T	G	T	T	G	C	G	T	C	T	T	A	T	G	T	T
HPV68b variants**	01_HPV68b-ME180																																												
	02_HK.AY829141										C																																		
	03_BR.AY829136							G																																					
	04_MX.AY829148								G																																				
	05_BR.AY829166						T		G																																				
	06_BR.AY829150								G																		G																		
	07_BR.AY829152								G	C																																			
	08_BR.AY829144								G											A																									
	09_SA.AY829157								G																			T		G															
	10_HK.AY829162		A								C															G																			
	11_SA.AY829168			G			T								C					A					G			T			C		A				A						T		
	12_ED.AY829160							G			C		A	A	C													T			C		A										T		
	13_ED.AY829165										C				C			T	C									T			C		A			T							T		
	14_SA.AY829171						T				C				C					A								T			C		G										T		
	15_BR.AY829167						T				C				C	C	T											T			C		A										T		
	16_SA.AY829169						T				C				C													T			C		A		T								T		
HPV68b(int)-ME180 5'copy					-	-																										C													
HPV68b-EU918769					-	-					C																																		
Reims-06	new (16)						T				C				C									A				T			C		A										T	A	
Reims-09	new (16)						T				C				C													T			C		A										T	A	
Reims-14	new (16)						T				C	T			C													T			C		A												
Reims-15	new (04)								G																				G																
Reims-16	new (04)	T							G																																	C			
Reims-18	new (14)						T				C				C							C						T			C		G								G		T		A
Reims-28	new (16)						T				C				C													T			C		A	A						C			T		A
Reims-31	new										C				C													T			C								A				T		G
Reims-33	04								G																																				
HPV68b-CIN2	new								G												G		C															T							

* Based on Figure 2.20 (first row) and the sequence of the complete HPV68b-CIN2 genome (second row).
** Obtained from (Calleja-Macias et al, 2005).
"-" indicates a gap.
Green highlight indicates variations not present in variants 01-16.
Variants assignment of the ten DNA samples (bottom part) is based on sequence similarity to one of the reference variants, where "new" indicates a new variant and the bracketed numbers refers to the most similar reference variant.

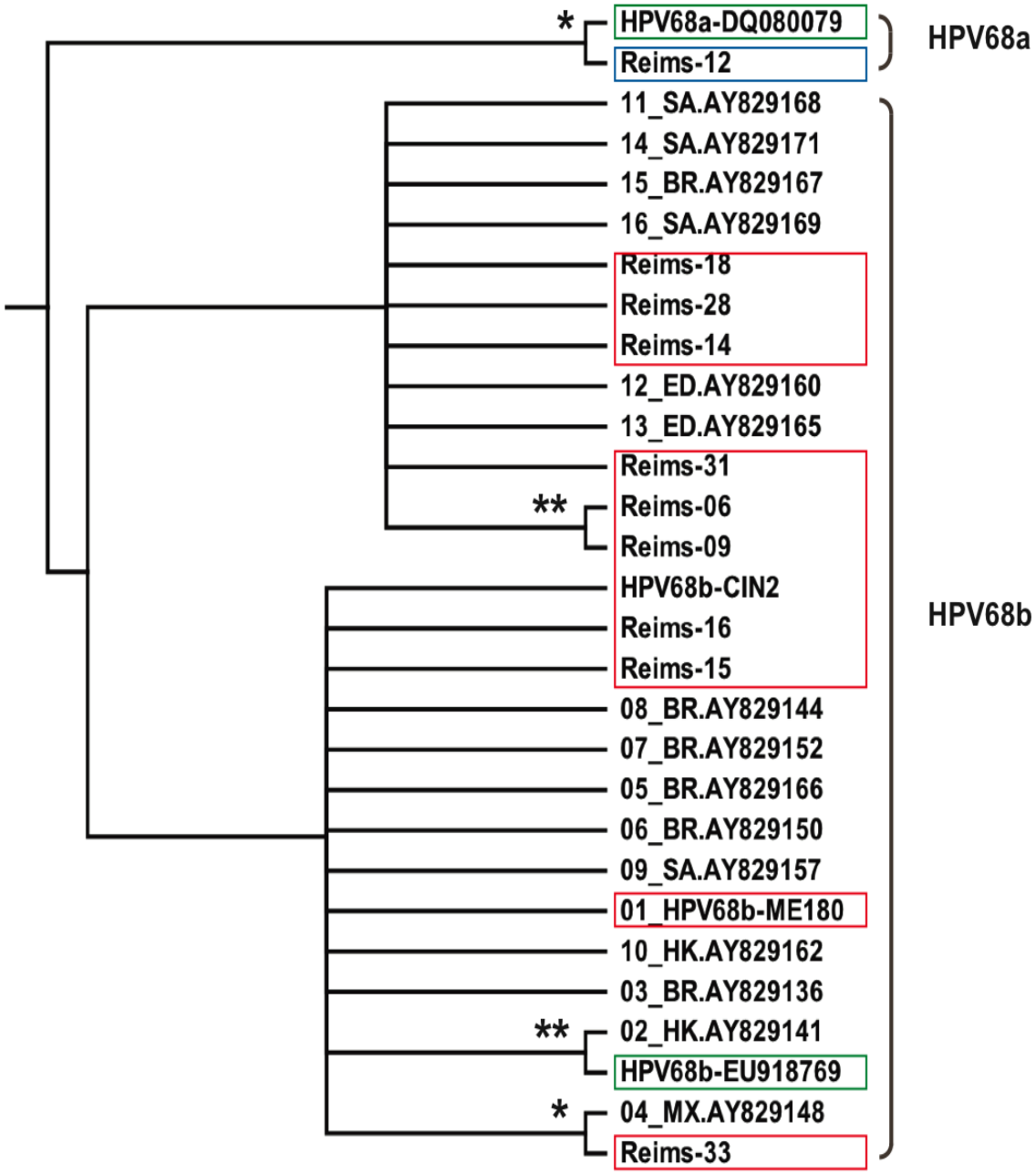


Figure 2.21: Phylogenetic tree created from partial HPV68 URR sequences. A rooted phylogenetic tree was created from the sequence alignment of HPV68 URR sequences (pos. 7279 to 7769 based on HPV68b-CIN2 genome), using UPGMA algorithm. The sequences obtained in this PhD work are bracketed in red and blue, sequences from the database in green, whereas the sixteen previously published HPV68b variants (Calleja-Macias et al, 2005) are shown without brackets. * indicates identical sequences. ** indicates sequences are not identical.

2.2 Analysis of HPV16 sequences generated by the ASP16 strategy

The ASP16 strategy was developed by Bo Xu as a novel strategy for determination of HPV16 integration sites in cervical lesions (Xu, 2010). The ASP16 strategy employs one of the next generation sequencing technologies, Roche/454 GS-FLX pyrosequencing (http://www.454.com/). With the Roche/454 GS-FLX format used, each ASP16 sequencing run can generate around 200,000 sequence reads. For this enormous amount of sequence data, it is necessary to apply computer power to assist in sequence data analysis. With my knowledge in computer programming, I collaborated with Bo Xu in the ASP16 project, and developed a set of computer programs for automatically analyzing sequence data generated by Roche/454 GS-FLX sequencing. The development of these programs is described in section 2.2.1. After Bo Xu had finished his PhD work with two ASP16 experiments completed, I continued to optimize the ASP16 strategy and conducted another two ASP16 experiments (ASP16-3 and ASP16-4). The results of these two experiments are described in section 2.2.2.

2.2.1 Development of computer programs for ASP16 analysis

2.2.1.1 Designing computer program strategies and platforms

In ASP16 strategy, the DNAs from cervical scrapes are amplified by GenomePlex whole genome amplification. The HPV16-containing DNAs are enriched by linear amplification using a set of HPV16-specific primers spanning the HPV16 E1-E2 region to produce single-stranded fragments, which are then amplified in multiplex PCR using bipartite forward primers and a tripartite reverse primer (Figure 2.22). There were 24 different 4-nt barcodes designed for the ASP16 experiments. Each barcode is unique for an individual DNA sample.

The primary aim of the ASP16 strategy is the identification of viral-cellular junction sequences. Considering the enormous number of sequence reads in each ASP16 experiment, however, it is not practical to perform human database blasting for all sequence reads. It was decided that the program sets would not perform this task, but narrow down the sequence reads with possible integration junctions by classifying the sequences into small groups using different criteria.

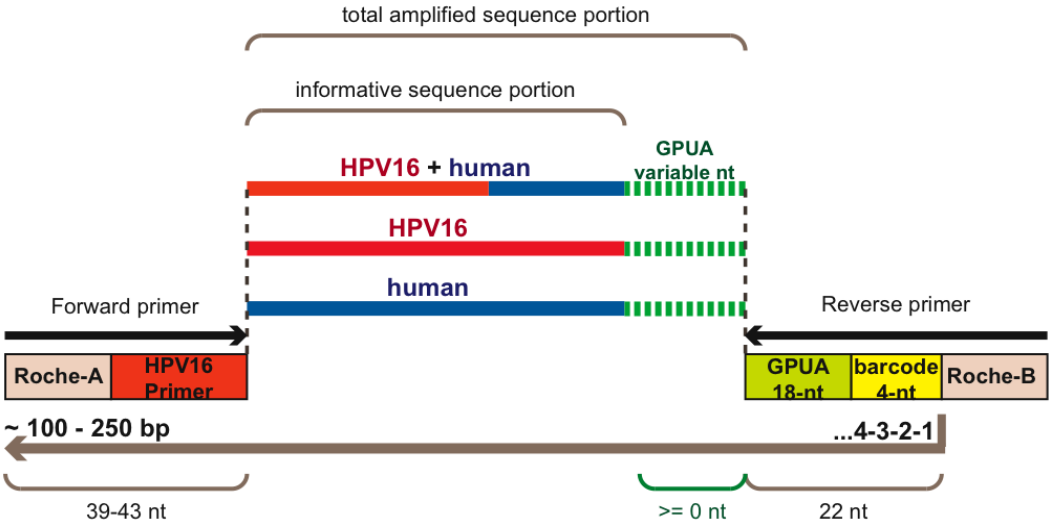


Figure 2.22: Principal structure of amplicon fragments for Roche/454 pyrosequencing in ASP16 experiments. The amplicon fragments are amplified using primer pairs (black arrows) whose compositions are indicated. Roche-B is used as the sequencing primer. The individual sequence read starts with the 4-nt barcode, followed by the minimum length of 18-nt GenomePlex universal adapter sequence (GPUA), then by variable amplicon sequence length (pure HPV16, pure human DNA, or HPV16 with human DNA), and ends with RA_HPV16 primer. The arrow at the bottom indicates the direction of sequencing. The dashed green line represents a variable length of variable GPUA sequence, ranging from 0 up to ~100 nt (see text).

The computer programs were designed to be able to

(1) sort sequence reads into sample groups where they originated from, according to the 4-nt barcode,

(2) eliminate sequence reads which do not contain HPV16 DNA sequence (pure human DNA),

(3) classify sequence reads into groups based on their HPV16 primers,

(3) generate alignments of grouped sequence reads,

(4) perform basic statistical analysis, and

(5) generate outputs in suitable file formats.

The program strategy is summarized in Figure 2.23. The input data is a text file, containing FASTA format sequence reads generated by the Roche/454 GS-FLX pyrosequencing. The program strategy starts by reading the sequence data from the input file. The sequence reads are analyzed one by one. Eventually, alignments of appropriately grouped sequence reads are generated and they may be viewed and edited with any sequence editor software. These alignments are helpful for sequence data visualization in order to search for integration junctions. The basic statistical outputs are generated as tab-delimited files and can be opened in Microsoft Excel or similar software.

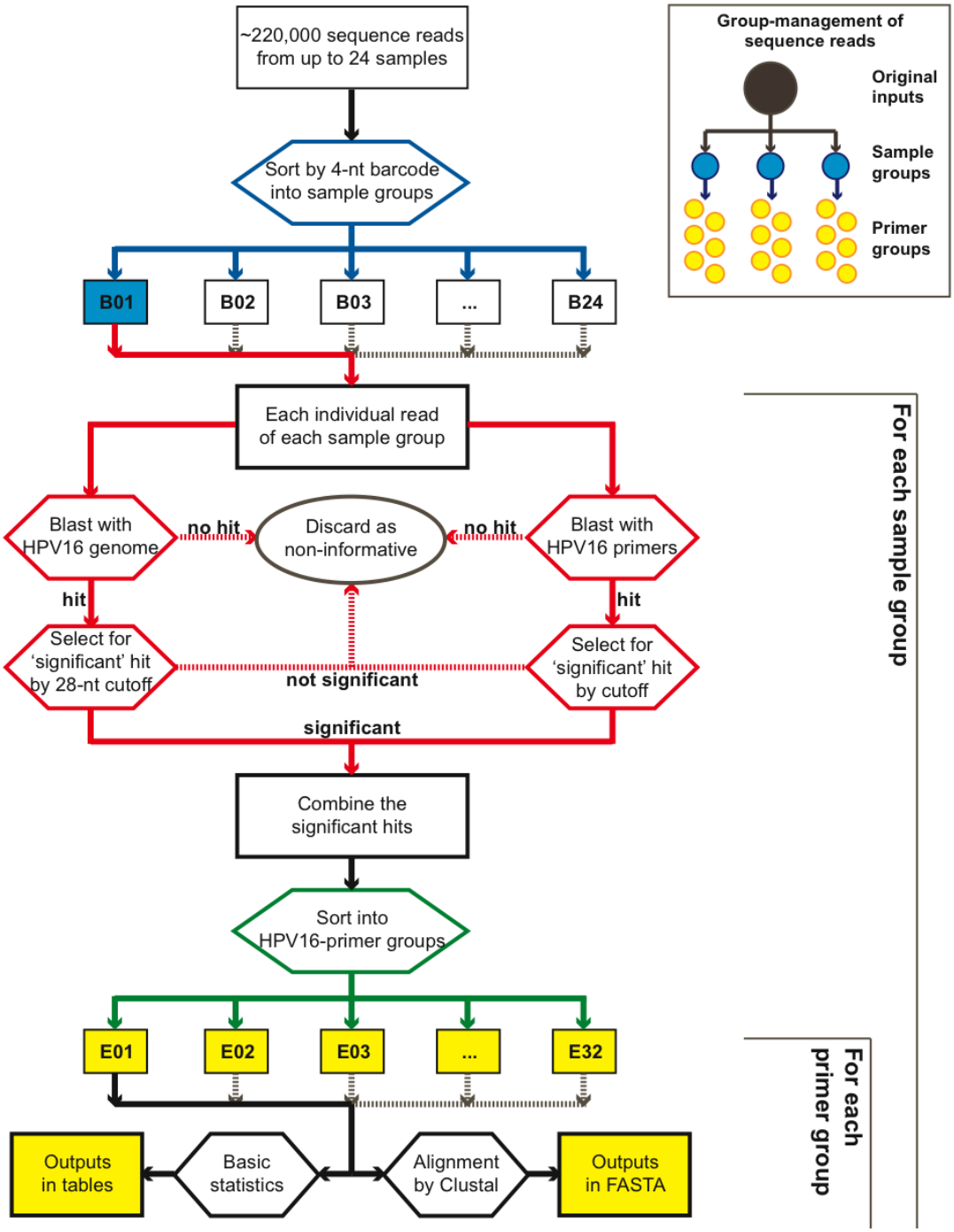


Figure 2.23: Program strategy for ASP16 sequence analysis. From the overall input of up to 220,000 sequence reads from Roche/454 GS-FLX, each sequence read is sorted into one of 24 sample groups according to the 4-nt barcodes (B01, …, B24). Each barcode is unique for an individual DNA sample. For each sample group, each sequence read is blasted with HPV16 reference genome and a set of complete Roche-A_HPV16 (RA_HPV16) primers. If there is no hit, the sequence is discarded. Otherwise, the hit sequence is filtered with appropriate cutoffs (see text). If the sequence meets the cutoff conditions, it is regarded as significant and processed further, otherwise discarded as non-informative. Significant sequences determined by both HPV16 genome and HPV16 primers are combined, and sorted into HPV16 primer groups (E01, …, E32). For each primer group, the sequence reads are aligned with ClustalW and the alignments are written as FASTA outputs. Basic statistics are also calculated and outputs are generated in table-format files.

This designed program strategy does not include automatic identification of human DNA in the sequence reads. The main reason originates from the complication of GPUA sequences. The GPUA sequence was incorporated into the reverse primer used to produce the amplicon and has the minimum length of 18 nt. However, as the sequence data of ASP16-1 and ASP16-2 showed, the GPUA can be longer with GT-rich unpredictable sequence variations of up to ~100 nt (Xu, 2010). Because of this unpredictability, it was not always possible to define whether the GT-rich variable sequence belongs to GPUA or human sequence (integration junction) without blasting it to the human sequence database. Blasting of such GT-rich sequence to the human database usually takes at least a few minutes and does not always come up with a satisfactory result. Therefore, this procedure is not integrated in the analysis process.

With the main computer program strategies decided, the next task was to decide on a suitable computer platform. Perl is a high-level computer programming language favored in the bioinformatics community (http://www.perl.org). For this study, it was chosen as the computer programming language because of its efficiency, simplicity, rapid prototyping, and ease-of-use for object-oriented programming. All computer programs were executed under Perl application version 5.8.8, and primarily designed to run under Mac operating system (Mac OS). With some modifications, they may also run under other Unix-based environments, such as Linux. The programs are not suitable for running under Windows operating system because of memory- and resource-handling problems, which will result in substantially slower program process.

2.2.1.2 Computer program algorithms and their tasks

Four sets of programs were written. Key algorithms of these program sets are described in the following. It is intended to show how the programs proceed and how they make decisions. The program source codes line-by-line will be explained in the Appendix A8.

Program set 1: Sort sequences into sample groups by barcodes

The first program set consists of a single program that sorts total sequence reads by their barcode sequences into sample groups (Figure 2.24). It consists of a single program named `PROG_SET_1_mac.pl`. A single text file, containing complete

sequence reads from the Roche/454 GS-FLX sequencing in FASTA format, is given as an input. The outputs of program set 1 are 24 groups of sequence reads, each group corresponding to an individual DNA sample.

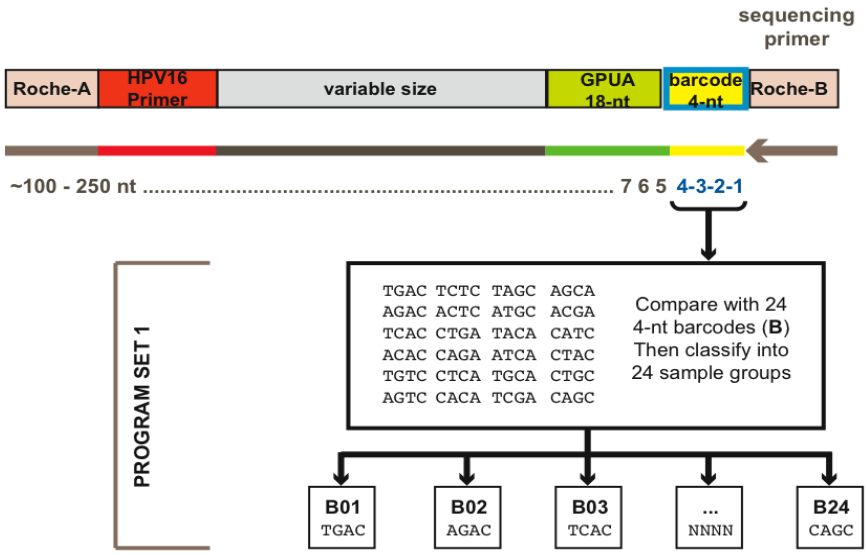


Figure 2.24: Scheme of program set 1. The principal structure of amplicon fragments for the Roche/454 GS-FLX sequencing is shown at the top. Roche-B is used as the sequencing primer. The first four nucleotides of the sequence read correspond to the 4-nt barcode. Each barcode represents an individual DNA sample. This program set compares the first 4 nucleotides of each sequence read with a collection of the twenty-four 4-nt barcodes used in ASP16. The sequences are sorted into 24 barcode groups, based on the matching 4-nt sequences.

Program set 2: Selection of informative sequences

The second set selects informative sequence reads by blasting and filtering steps, sorts the sequence reads into individual HPV16 primer groups, calculates basic statistical data, and delivers the outputs. It consists of nine programs, with the following names:

```
set2_p2mac.pl
set2_p3mac.pl
set2_p4mac.pl
set2_p5mac_a_noCutoff.pl
set2_p5mac_d_28bpCutoff.pl
set2_p5mac_edit.pl
set2_p6mac_before.pl
set2_p6mac_fasta.pl
set2_p7mac.pl.
```

The input to this program set is a group of sequence reads of an individual sample group (also called barcode group) from the outputs of program set 1. Set 2 will select informative sequence reads and group them into HPV16 primer groups. The tasks are divided into three steps: (1) blast with the complete HPV16 reference genome (HPV16R) database and filter with cutoffs, (2) blast with HPV16 primer database, sort into HPV16 primer groups and filter with cutoffs, and (3) combine results of the two database blasts, analyze possible HPV16 breakpoints and report data in Tab-delimited format. The scheme of program set 2 is shown in Figure 2.25.

Set 2 – step 1: Blasting with HPV16R, parsing information and filtering

With regard to the HPV16 genes E1 and E2, the pyrosequencing reads contain the sequence information of the minus (or antisense) strand. For conversion into plus (or sense) strand, the program set generates reverse-complementary sequences of all input sequences (Figure 2.25). This produces sequences starting from Roche-A sequence and ending with the 4-nt barcode sequence. Each reverse-complemented individual sequence is blasted with HPV16R database (`hpv16`), using `blastall` program with the following defined values: program=blastn, word-size=11, DUST-filter=off. Each blast result is saved as a text file. There are three possibilities of the blast results: (1) no hit, (2) one hit, (3) more than one hit. "Hit" refers to a match of a query (ASP16 sequence read) to a region on a sequence in the database. More-than-one-hit indicates matches on more than one location on the HPV16R genome.

After all the sequences have been blasted, the program reads information from each blast result file. Sequences with no hit are discarded. The program parses information from the blast result files of the sequences that have one or more hit. Figure 2.26 shows an example for information parsing.

For sequences with more than one hit, the programs check whether the second hit contains useful information for further analysis. The data of the second hit are considered useful and are collected only if the matched region is at least 94% similar (*%_match*) and covers at least 15 nt (*scoreBP*). Otherwise, the second hit is considered non-informative and only data from the first hit are collected.

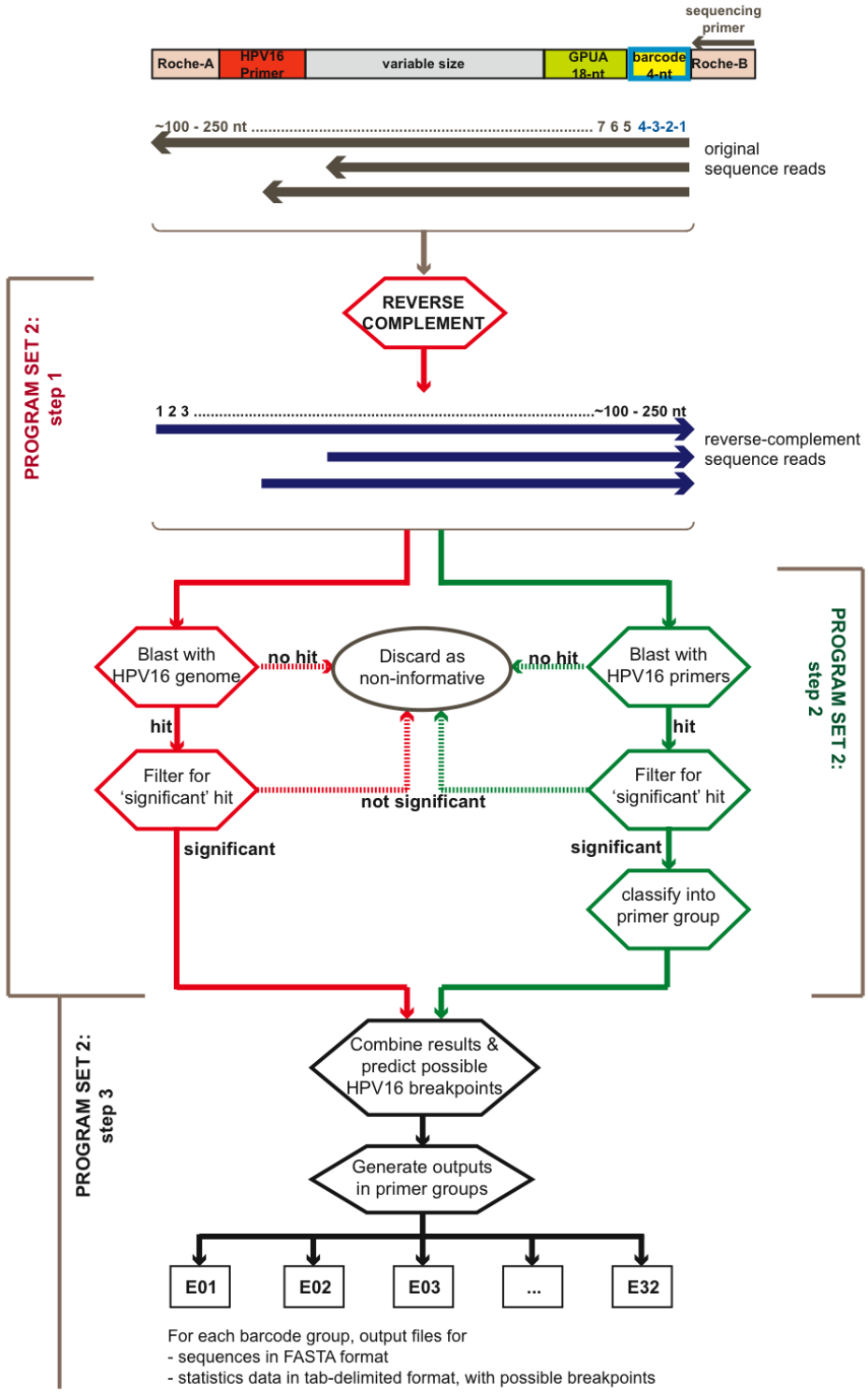


Figure 2.25: Scheme of program set 2. Program set 2 is divided into three steps. In step 1 (left side), the reverse complementary sequences of the original sequence reads are generated and used for further analysis. The reverse-complementary sequences are blasted with HPV16R genome, and sequences with significant HPV16R matches are selected through filtering processes. In step 2 (right side), the reverse-complementary sequences are blasted with RA_HPV16 primer database. The sequences with significant primer matches (see main text for details) are selected through filtering processes, and classified into primer groups. In step 3, significant sequence lists from step 1 and 2 are combined. For each primer group, the possible HPV16 breakpoints are predicted, and outputs are written in tab-delimited format. A file containing the complete sequences of each primer group is generated in FASTA format.

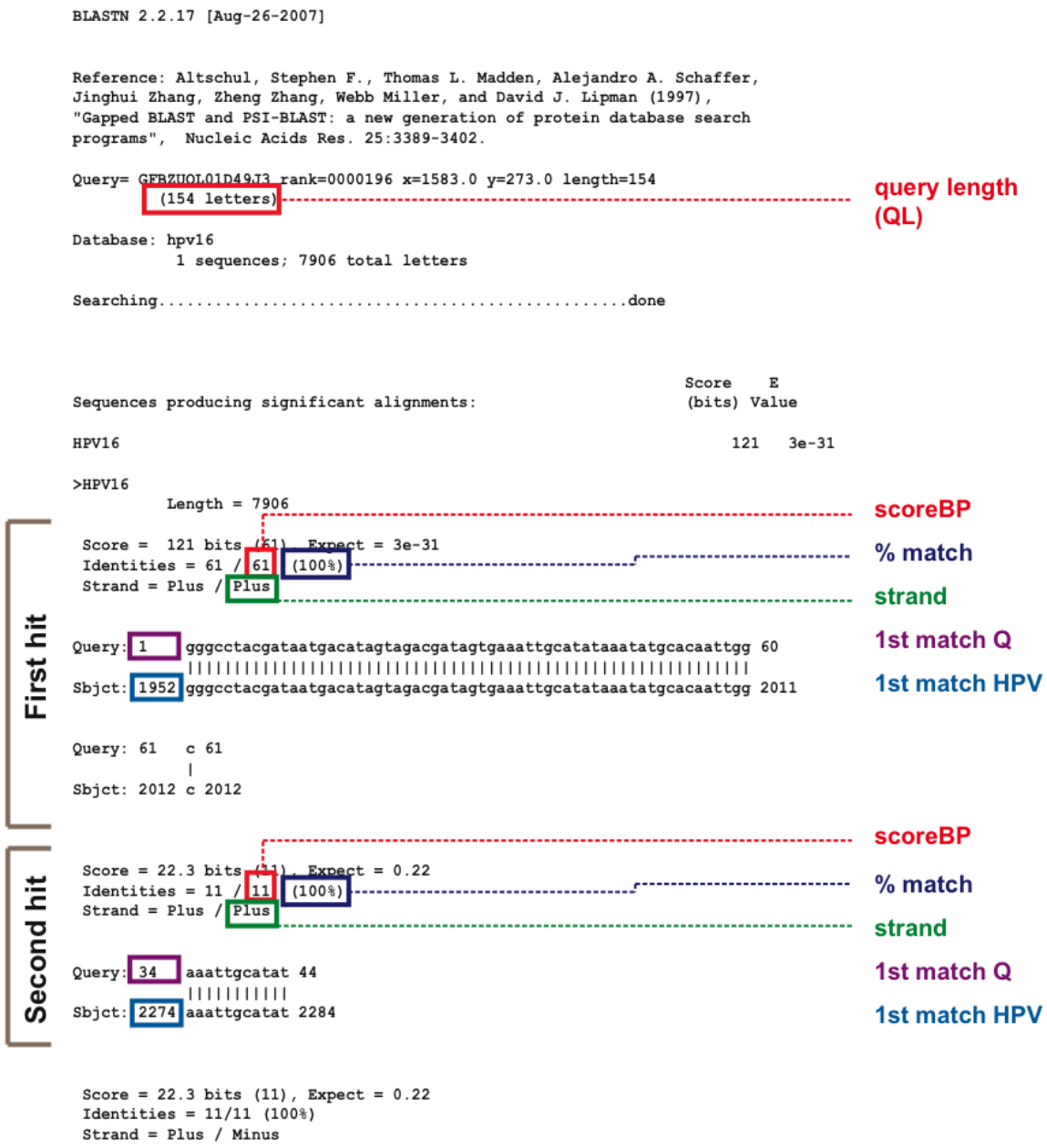


Figure 2.26: Parsing of information from HPV16R blast result. An example of a HPV16R blast result with more than one hit is shown. Query refers to the ASP16 sequence read and Sbjct is HPV16R. For HPV16R blast with one or more hit, six data are collected as marked in colored boxes from the first two hits. *Query length* (QL) is the length of the ASP16 sequence read. *ScoreBP* is the length (nt) of the area where query matches to HPV16R. *%_match* is the percentage of similarity between the query and HPV16R. *Strand* refers to the HPV16R matching strand. *1st_match_Q* is the position on the query strand where matching to HPV16R starts. *1st_match_HPV* is the position on HPV16R matching to 1st_match_Q position. ScoreBP of the first hit is used in the filtering step for significant HPV16R match decision.

The informative sequences should contain a significant length of HPV16R sequence to ascertain that the sequence read is not an unspecific amplicon. This is achieved by the filtering process. A cutoff value of 28 nt was defined for the filtering step. This means that the sequence read is considered significant if the *scoreBP* value of the first HPV16R hit is at least 28 nt. The list of significant sequences are collected and stored for further analysis in the third step.

Set 2 – step 2: Blasting with HPV16 primers, sorting to primer groups and filtering

Each reverse-complementary sequence, generated in step 1, is blasted with the RA_HPV16 primer database (`EEpyro3`), using `blastall` program with the following defined values: program=blastn, word-size=11, DUST-filter=off. Each blast result is saved as a text file. There are three possibilities of the blast results: (1) no hit, (2) one hit, (3) more than one hit. More-than-one-hit indicates matches to more than one HPV16 primer sequences.

Sequences with no hit to RA_HPV16 primer are discarded. For sequences with hit(s), there are six possibilities of how the query can match to a RA_HPV16 primer. These are outlined in Figure 2.27.

The program set determines, based on these six cases, whether each of the sequence reads has a significant primer hit. The parsed values from RA_HPV16 primer blast results are used to evaluate the conditions of the hit. The decisions are made according to different criteria that are listed in Table 2.11. There are three categories of significant RA_HPV16 primer matches. Category 1 includes matches as in case A1. Category 2 includes matches as in case A2. Category 3 includes matches as in cases B1, B2, C1 and C2. A sequence read is defined as having a significant RA_HPV16 primer match if it meets all the required conditions in one of these three categories. Otherwise, it is declared as having no match to any RA_HPV16 primer and excluded from further analysis. The sequences with significant RA_HPV16 primer hits are sorted into groups according to the RA_HPV16 primers, and stored for the next analyzing step.

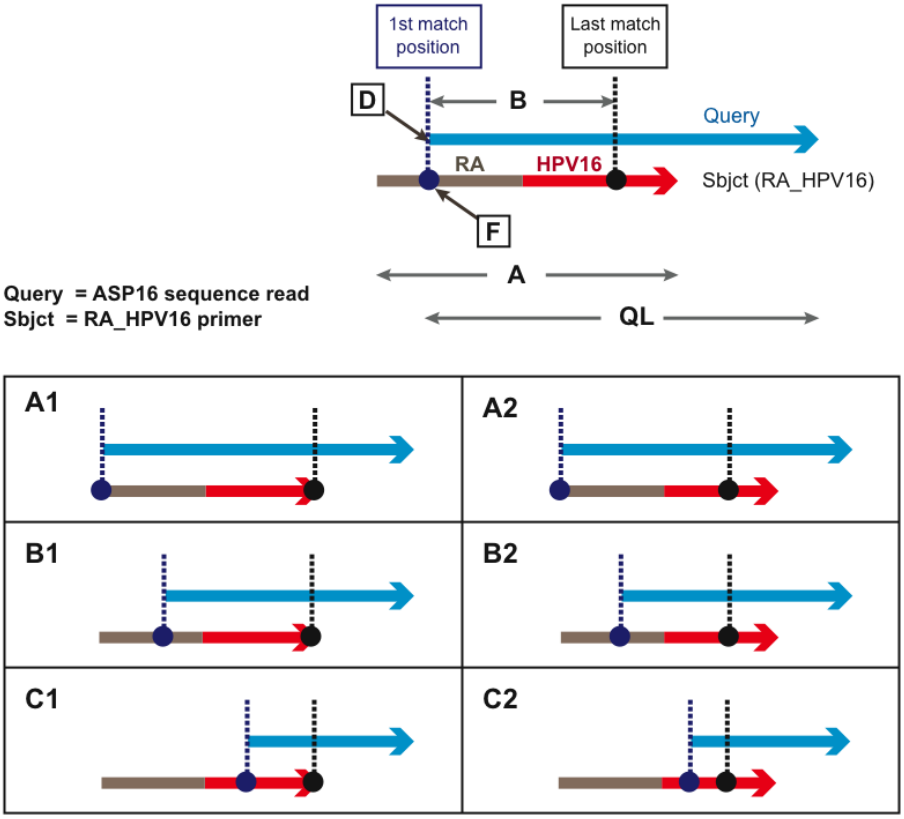


Figure 2.27: Matching possibilities between ASP16 sequence read and a RA_HPV16 primer. There are six possible matching patterns between an ASP16 sequence read and a RA_HPV16 primer. Cases A1 and A2 show ASP16 reads matching from the beginning of the primer. Cases B1 and B2 show ASP16 reads missing the 5'end of the primer. Cases C1 and C2 show ASP16 reads matching only in the HPV16 part of the primer. In cases A1, B1 and C1, ASP16 reads match until the 3'end of the primer. In cases A2, B2 and C2 the match terminates within the HPV16 part of the primer. QL: query length(nt). A: Sbjct length (nt). B: matching area (nt). D: first matched position on the query. F: first matched position on Sbjct. C: percent similarity (not illustrated). E: strand of Sbjct which the query matches to (not illustrated).

Table 2.11: Criteria for the selection of significant RA_HPV16 primer match

Criteria	Criteria*	significant RA_HPV16 primer match		
		Category 1 (A1)	Category 2 (A2)	Category 3 (B1,B2,C1,C2)
The matching strands of the query and the primer sequence are in the same orientation	E = Plus	+	+	+
The matching similarity is at least 90%	C >= 90%	+	+	+
The query matches the complete primer length	B >= A	+		
The query does not match the complete primer length	B < A		+	+
The query matches from the 5'end of the primer	F = 1	+	+	
The query does not match from the 5'end of the primer	F != 1			+
The query does not include the Roche-A part of the primer	(B+F) >20			+
The query does not match the 3'end of the primer, max. 4 nt	(A-(B+F)) > 5		+	+

* The criteria abbreviations are explained and illustrated in Figure 2.27.

Set 2 – step 3: combining blast results, predicting breakpoints and exporting the outputs

In this step, the lists of sequences with significant HPV16R matches and significant RA_HPV16 primer matches are combined and compared (see Figure 2.25). Only sequences that are present in both lists are processed further. Otherwise, they are declared as non-informative and discarded. At the same time, the program counts the numbers of sequence reads in each primer groups for both significant-only sequence reads (filtered by 28-nt cutoff in step 1) and total sequence reads (without the 28-nt cutoff filtering). The resulting sequence numbers are exported in two tab-delimited format files, one belonging to no-cutoff and the other belonging to 28-nt cutoff.

In this step, the program estimates the longest position on the HPV16R genome that each sequence read is matched to. This position is informative for hinting at a possible HPV16 breakpoint, and called "last-HPV16-match position". Using parsed information from HPV16R blast results in step 1 (see Figure 2.26), the values of the first and second (if exists) HPV16R hits of each sequence read are analyzed.

For the sequence reads with a single HPV16R hit, the data of this single hit are used for estimation of the last-HPV16-match position. For sequence reads with more than one HPV16R hits, the program evaluates whether to use the information from both hits or only one of the hits. The decisions are made based on positions and lengths of matching HPV16 area of each hit. All categories for sequences with two hits are explained in Figure 2.28.

After all sequence reads are sorted into appropriate categories, the program reports the selected data in tab-delimited format files for each sample group. Figure 2.29 shows an example of these data, opened in MS Excel. The data are present in 13 columns as explained in Table 2.12. For each DNA sample, a possible HPV16 breakpoint may be predicted by looking at the data in column F. If HPV16 integration exists, a number of sequence reads should have an identical HPV16 position in column F.

Category	illustration	Explanation and usage
1		**The first hit ends before the second hit.**
1.1		The first hit starts at the same HPV16 position as the second hit. The data of the second hit are used.
1.2a		The first hit starts earlier than the second hit. Both hits overlap. The second hit is >=15 nt and has >=94% similarity to HPV16R. The data of both hits are combined as a single match.
1.2b		The first hit starts earlier than the second hit. Both hits overlap. The second hit is <15 nt or has <94% similarity to HPV16R. The data of the first hit are used.
1.2c		The first hit starts earlier than the second hit. Both hits do not overlap. The second hit is >=15 nt and has >=94% similarity to HPV16R. The data of the both hit are combined as a single match, with or without a gap.
1.2d		The first hit starts earlier than the second hit. Both hits do not overlap. The second hit is <15 nt or has <94% similarity to HPV16R. The data of the first hit are used.
1.3		The first hit starts after the second hit. The data of the second hit are used.
2		**The first hit ends at the same position as the second hit.**
2.1		The first hit starts at the same position as the second hit. The data of the first hit are used.
2.2		The first hit starts earlier than the second hit. The data of the first hit are used.
2.3a		The first hit starts after the second hit. The second hit is <15 nt or has <94% similarity to HPV16R. The data of the first hit are used.
2.3b		The first hit starts after the second hit. The second hit is >=15 nt and has >=94% similarity to HPV16R. The data of the second hit are used.
3		**The first hit ends after the second hit.**
3.1		The first hit starts at the same position as the second hit. The data of the first hit are used.
3.2		The first hit starts earlier than the second hit. The data of the first hit are used.
3.3a		The first hit and second hit overlap. The second hit is >=15 nt and has >=94% similarity to HPV16R. The data of both hits are combined as a single match.
3.3b		The first hit and second hit do not overlap. The second hit is >=15 nt and has >=94% similarity to HPV16R. The data of both hits are combined as a single match, with or without a gap.
3.3c		The first hit and second hit overlap. The second hit is <15 nt or has <94% similarity to HPV16R. The data of the first hit are used.
3.3d		The first hit and second hit do not overlap. The second hit is <15 nt or has <94% similarity to HPV16R. The data of the first hit are used.

Figure 2.28: Different cases of double HPV16R blast hits of a ASP16 sequence read. This figure shows different cases of two HPV16R hits from a ASP16 sequence read. Each line in the illustration column represents a matched area on the HPV16R genome. For sequences with only one HPV16R hit (not shown in this figure), the data of this hit are used for breakpoint analysis. For sequences with two hits, the program decides, based on 16 different cases, which data to be used. Blue represents the first hit, and green the second hit. Continuous lines indicate the data of the hit are used for breakpoint analysis. Dashed lines indicate the data of that hit are not used for the analysis. The lengths and positions of the blue and green lines imitate the relative HPV16 matching length and actual positions on HPV16 genome respectively. Each case is explained in the last column. The usage of hit data is indicated.

A	B	C	D	E	F	G	H	I	J	K	L	M
barcode	primer	SeqNo	Seq Length	Last hit Position Query	Last hit Position HPV	Match nt	Last 2nd-hit Position Query	Last 2nd-hit Position HPV	Match nt 2ndHit	Overlap 1st & 2nd hits	Original Seq Name	Reverse complement sequence
1	E02	38	67	36	1312	36	none	none	none	none	>GFBZUQL01	AAGTGGAAAC
1	E02	299	61	29	1303	29	none	none	none	none	>GFBZUQL01	TGAAGTGGAA
1	E02	328	77	58	1332	58	none	none	none	none	>GFBZUQL01	TGAAGTGGAA
1	E02	331	68	47	1322	41	none	none	none	none	>GFBZUQL01	TGAAGTGAAA
1	E02	439	60	28	1302	28	none	none	none	none	>GFBZUQL01	TGAAGTGGAA
1	E02	459	75	37	1311	37	none	none	none	none	>GFBZUQL01	TGAAGTGGAA
1	E02	619	72	36	1310	36	none	none	none	none	>GFBZUQL01	TGAAGTGGAA
1	E02	644	91	42	1316	42	none	none	none	none	>GFBZUQL01	TGAAGTGGAA
1	E02	676	72	42	1316	42	none	none	none	none	>GFBZUQL01	TGAAGTGGAA
1	E02	737	70	38	1312	38	none	none	none	none	>GFBZUQL01	TGAAGTGGAA
1	E02	807	68	36	1310	36	none	none	none	none	>GFBZUQL01	TGAAGTGGAA
1	E02	863	64	34	1308	34	none	none	none	none	>GFBZUQL01	TGAAGTGGAA
1	E02	924	69	42	1316	42	none	none	none	none	>GFBZUQL01	TGAAGTGGAA

Figure 2.29: Example of a tab-delimited output from program set 2. For each DNA sample group, a tab-delimited file is exported by program set 2. The file is opened in MS Excel. For further explanation, see Table 2.12.

Table 2.12: Explanation of tab-delimited output from program set 2 as shown in Figure 2.29.

Column	Title	Description
A	barcode	barcode number
B	primer	RA_HPV16 primer group, assigned to the sequence read
C	SeqNo	sequence ID, assigned by this program set
D	SeqLength	sequence read length (nt)
E	Last hit Position Query	position on the sequence read corresponding to the farthest HPV16R position to which the sequence read matches
F	Last hit Position HPV	position on the HPV16R corresponding to the farthest position to which the sequence read matches
G	Match nt	number of nucleotides covering the HPV16R hit area
H	Last 2nd-hit Position Query	for sequence with significant second hit, this column contains the position on the sequence read corresponding to the farthest HPV16R position to which the second hit matches
I	Last 2nd-hit Position HPV	for sequence with significant second hit, this column contains the farthest HPV16R position to which the second hit matches
J	Match nt 2ndHit	number of nucleotides covering the HPV16R second hit area
K	Overlap 1st & 2nd hits	for sequences in categories 1.2a and 3.3a, this column contains the number of overlapped nucleotides between the two hits
L	Original Seq Name	original sequence name
M	Reverse complement sequence	reverse complementary sequence of the original sequence read

Program set 3: Multiple-alignment of grouped sequences

This program set performs multiple alignments of sequences in an individual primer group. The program process is illustrated in Figure 2.30. This set consists of two programs, with the following names:

```
set3_p8mac.pl
set3_p9mac.pl
```

The inputs to this program set are FASTA sequence lists of each RA_HPV16 primer group of each DNA sample from program set 2. To obtain a good alignment, the sequences should be edited prior to performing an alignment so that they contain as little sequence variations as possible. The program considers only the HPV16 sequence part of each sequence read to generate multiple alignments for each RA_HPV16 primer group.

Program set 4: Writing different sequence outputs in FASTA format

Program set 4 re-edits the results of multiple alignments from program set 3. The program process is illustrated in Figure 2.30. It consists of a set of two programs, with the following names:

```
set4_p10mac.pl
set4_p11mac.pl
```

The alignments generated by program set 3 contain only the HPV16 portions of each sequence. This program set re-edits the alignments by restoring the HPV16-portion-only sequences to their original sequences. The edited alignments are then exported in FASTA format as outputs. The alignments can be viewed and edited in any sequence editor software or even a text editor. In this study, Geneious Pro version 4.8.5 was used.

2.2.1.3 Requirements prior to executing the programs

Because the program sets integrate the usages of other stand-alone programs such as BLAST and ClustalW, these stand-alone programs must be installed in the proper locations on the analyzing computer prior to executing the ASP16 analysis program sets. Databases for BLAST analysis must also be set up. The following describes six requirements before the ASP16 analysis programs can be executed. The website addresses where BLAST and ClustalW applications can be obtained are indicated in the Materials and Methods section.

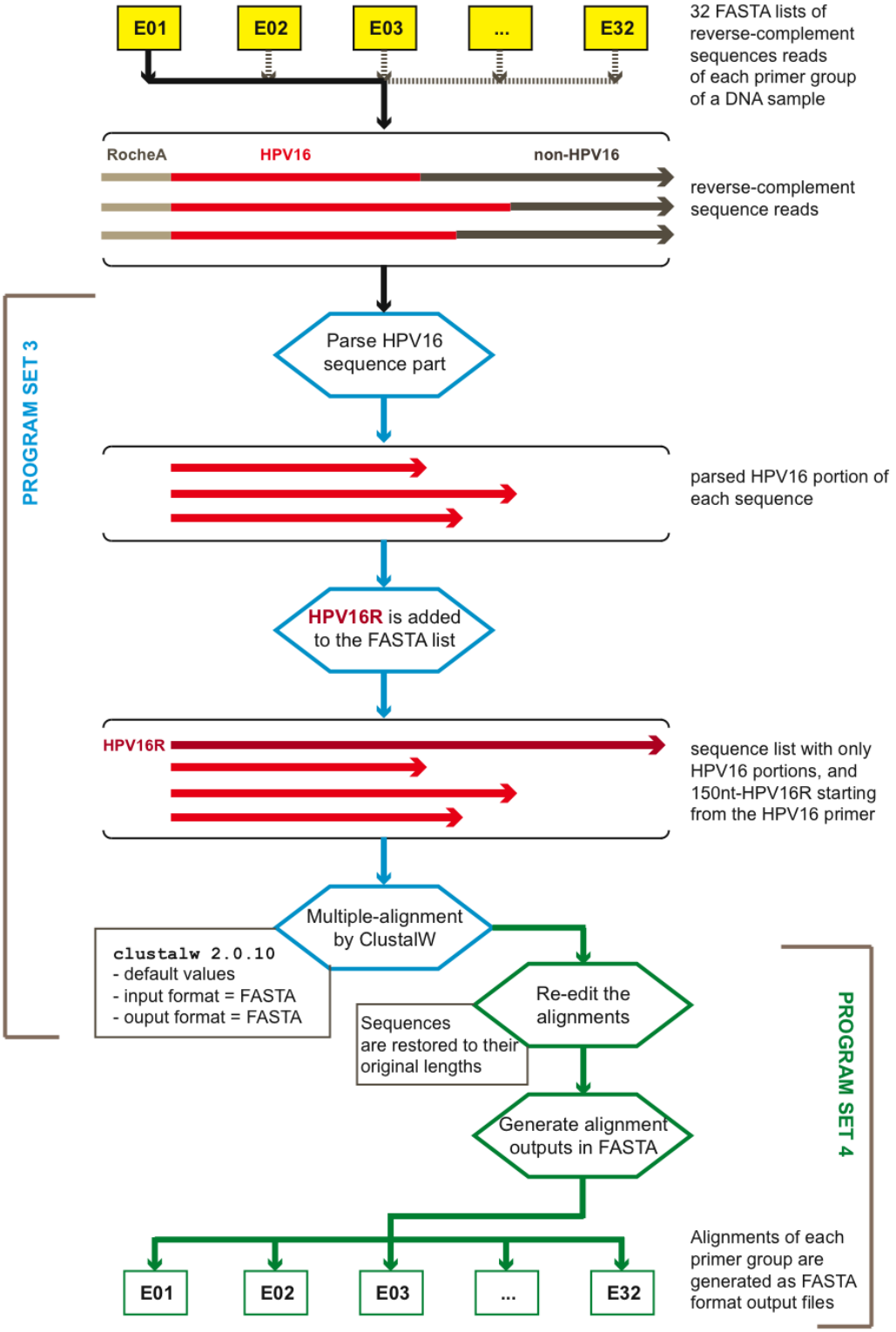


Figure 2.30: Scheme of program sets 3 and 4. In program set 3, the HPV16 portion of the significant sequences are extracted and put in a new list for each RA_HPV16 primer group. A 150-nt HPV16R sequence is then added to the list. The added portion of the HPV16R sequence starts at the same position as the respective RA_HPV16 primer. Multiple alignment is performed to the prepared sequence list, using ClustalW version 2.0.10 with default values. Program set 4 sorts and edits the resulting alignments for simpler sequence visualization. The edited alignments are exported in FASTA format. Each DNA sample group contains thirty-two alignment outputs because 32 primers are used in the multiplex PCR.

(1) Perl program version 5.8.8 or newer must be installed. In Mac OS and Linux OS, Perl is usually already installed. To check if Perl is installed, the command "`perl –v`" is given in the command-line. It should report the version of the Perl program.

(2) BLAST executables version 2.2.17 (binary for Unix) must be installed under a directory named "blast-2.2.17", located directly under root.

(3) ClustalW version 2.0.10 for Mac OS must be installed under a directory named "clustalw-2.0.10-macosx", located directly under root. The text file containing thirty-two 151-nt partial HPV16R nucleotide sequences, named "`hpv16R_EEprimer_151nt`", must be saved under the same directory. The file contents are given in Appendix A7.

(4) Databases of the HPV16R reference genome and RA_HPV16 primers must be formatted and placed under directory /blast-2.2.17/db/. For the HPV16R database, a text file containing FASTA format of the HPV16R sequence must be created with the name "hpv16". For the HPV16 primer database, a text file containing FASTA format of the thirty-two RA_HPV16 primer sequences (with primer names E01,…,E32) must be created with the name "EEpyro3". The primer sequences are listed in Materials and Methods section. To format the databases, the stand-alone program from the BLAST executables, `formatdb`, is used with the commands:

"`formatdb –i hpv16 –p F –o T`" for HPV16R database, and

"`formatdb –i EEpyro3 –p F –o T`" for primer database.

After the formatting is completed, five new files are created for each database. These files must be moved under directory /blast-2.2.17/db/.

(5) A text file input must contain total sequence reads from the Roche/454 GS-FLX pyrosequencing, in FASTA format. No fixed path is required for the input file.

(6) The path (or file location) where ASP16 analysis programs are located, and the path where the ASP16 analysis results should locate, must be specified. In each of the four ASP16 program sets (`PROG_SET_1_mac.pl`; `PROG_SET_2_mac.pl`; `PROG_SET_3_mac.pl`; `PROG_SET_4_mac.pl`), the default paths can be changed to user-specified paths by editing the Perl scripts of these programs in a text editor application.

2.2.1.4 Executing the programs

All programs in this study were written in Perl language, and were executed under Perl program version 5.8.8. They had not been tested with other versions of Perl program. The programs were designed for command-line interface, not graphical user interface (GUI).

The GUI is not necessary because the programs were written in such a way that only a few simple short commands are required. This means a user is required to "*enter*" one or more commands to the computer manually in order to run these ASP16 analysis programs, instead of clicking buttons. In Mac operating system, the command-line interface can be accessed through the Terminal application. To run the complete four program sets, two commands are used.

For program set 1: "`perl PROGRAM Input_A Input_C`"

For program sets 2 - 4: "`perl PROGRAM Input_B Input_C`"

where `PROGRAM` contains complete program name and path

`Input_A` is complete path of input FASTA file

`Input_B` is barcode number (01, 02, …, 24)

`Input_C` is the experimental number (1, 2, 3, 4, …)

(1) `PROG_SET_1_mac.pl` This program sorts sequences into sample groups according to the barcodes. It is required to run only once.

(2) `PROG_SET_2_mac.pl` This program automatically executes all programs in set 2 (programs 2-7) that perform blasts and statistic calculations. It works for an individual barcode, and is required to run once for each barcode.

(3) `PROG_SET_3_mac.pl` This program automatically executes all programs in set 3 (programs 8–9) that perform sequence alignments. It should be run only after program set 2 has been completed. It works for an individual barcode, and is required to run once for each barcode.

(4) `PROG_SET_4_mac.pl` This program automatically executes all programs in set 4 (programs 10-11) that generate full FASTA sequence end-results. `Input_B` can be any number. It is required to run only once after all barcodes have been analyzed by program set 2 and set 3.

2.2.1.5 Output files used for ASP16 analysis

There are several output files generated by the ASP16 analysis programs. Most output files exist for the benefits of cross-communications between programs. The following output files are informative for the users.

Output 1: Sequences and HPV16R-hit data

These files are generated by program set 2. There is a single file produced per DNA sample. An example of these files is shown in Figure 2.29. The information from

these outputs can be used as reference data sources for the sequence reads. It is also possible to predict a possible HPV16 breakpoint for each DNA sample by sorting/editing the data in Excel.

Output 2: Basic statistics for sequences with or without 28-nt cutoff

Two files are produced per DNA sample, one containing the data without the 28-nt cutoff filtering, and the other with the filtering. These files are generated by program set 2. They contain numbers of sequence reads of each RA_HPV16 primer group of each DNA sample. The information was used to perform further statistical analyses of the experimental results, such as calculation of histograms of sequence distribution among different primer group. See Table 2.16 and Table 2.17 for the results of ASP16-3 and ASP16-4.

Output 3: Multiple alignments of sequences in each RA_HPV16 primer group

Maximally thirty-two files are generated per DNA sample, corresponding to the 32 RA_HPV16 primer groups. The number of the output files depends on the number of the primer combinations used. These files are generated by program set 4. They contain the alignments of sequences in an individual RA_HPV16 primer group of a DNA sample in FASTA format. An example of these sequence alignments is shown in Figure 2.39. The alignments can be used to find HPV16R point mutations, to search for HPV16 integration junction sites, and to create an assembled HPV16 sequence for each DNA sample. The assembled sequences can be used to construct a phylogenetic tree. See section 2.2.2 for application and illustration.

2.2.2 Analysis of HPV16 sequences in ASP16-3 and ASP16-4

The first two ASP16 experiments, ASP16-1 and ASP16-2, had been conducted by Bo Xu (Xu, 2010). In these two experiments, the sequence reads of each DNA sample covered about 50% of the analyzed HPV16 E1-E2 gene region, in average. This means that the chance of finding any HPV16 integration site was only about 50%. Further optimization of the ASP16 was required to obtain better sequence coverage.

In the frame of this PhD work, two ASP16 experiments (ASP16-3 and ASP16-4) were performed and the computer programs described in the previous section were used for data analysis. The main purpose was to find HPV16 integration junctions in the analyzed

DNA samples. Another purpose was to continue the optimization of the ASP16 strategy and investigate the results of the optimizations, in particular whether complete sequence coverage could be achieved. In addition, the sequence data were used to identify the HPV16 variants present in the samples. The results of both ASP16 experiments will be described together.

2.2.2.1 Optimization of ASP16

The Roche/454 GS-FLX is capable of amplifying up to 500 bp DNA fragments during the emulsion PCR step, and delivering average sequence reads with length around 200 nt (http://www.454.com/). However, the average sequence read lengths obtained in ASP16-1 and ASP16-2 were only 116 nt and 92 nt, respectively (Xu, 2010). Due to the presence of RA_HPV16 primer (39-43 nt), GPUA (at least 18 nt) and 4-nt barcode (4 nt) sequences at the termini of each sequence read, the informative sequence of the amplified genomic DNA is at least 61 nt shorter than the total sequence read length (Figure 2.31). Thus, the average informative sequence lengths of ASP16-1 and ASP16-2 were only 55 nt and 31 nt, respectively. Shorter-than-expected average sequence read length is the major reason for low sequence coverage in ASP16-1 and ASP16-2. The problem of short read length is due to the general limitation of the Roche/454 GS-FLX system for amplicon sequencing, not for genomic DNA sequencing (S. Wolf, personal communication).

In ASP16-2, sixteen combinations of HPV16 forward primers were used, spanning ~3000 bp of the HPV16 E1-E2 area (Xu, 2010). Each combination includes a biotin-labeled HPV16 primer and the corresponding RA_HPV16 primer. The distance between the primers was about 200 bp. Due to the average sequence read length of ~100 nt obtained by pyrosequencing, about 50% of the E1-E2 sequence region was not covered by the ASP16 sequences (Figure 2.32).

In the ASP16 experiments (ASP16-3 and ASP16-4) performed in this PhD work, an essential aim was to try to obtain 100% sequence coverage in the HPV16 E1-E2 region. Toward this goal, the distance between HPV16 primers was reduced to about 100 bp by adding another sixteen combinations of HPV16 primers. These primer combinations were used since experiment ASP16-3. Two of the additional primer combinations, E17 and E21, replaced two previous combinations of ASP16-2, which were excluded due to their

low specificity. Altogether, 30 combinations of HPV16 primers were used in ASP16-3. The sequences and names of all HPV16 primers for ASP16-3 an ASP16-4 are given in Materials and Methods. The locations of the RA_HPV16 primers of these 30 pairs are shown in Figure 2.33.

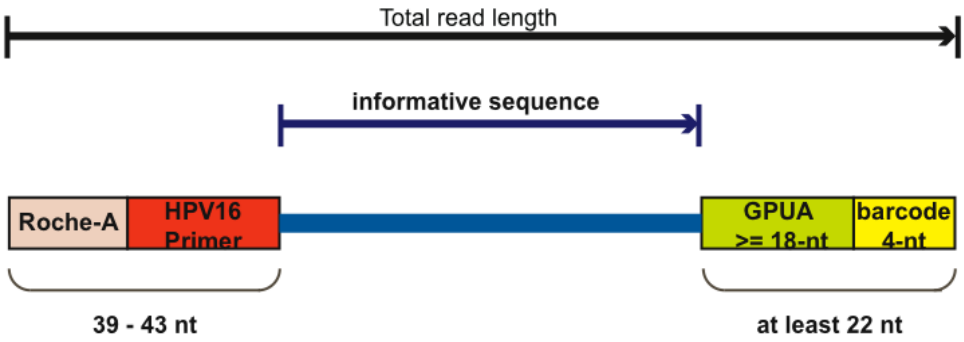


Figure 2.31: Components of an ASP16 sequence read. The total read length of each sequence read contains at least 61 nucleotides derived from the flanking primer sequences. The informative sequence corresponds to the remaining nucleotides of amplified sample DNA.

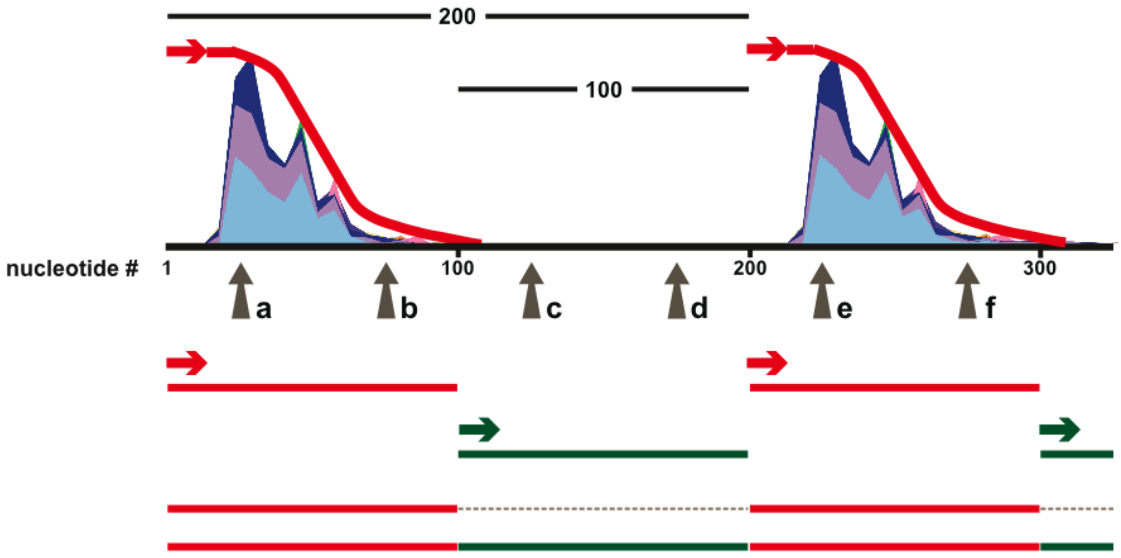


Figure 2.32: Model of primer locations and sequence read length in ASP16 experiments. The graphs represent histograms of sequence read length distribution obtained in ASP16-2. The graphs indicate that most sequence reads extend for 25-50 nt from the primer, and only a few sequences reach 100 nt. The HPV16 primers used in previous experiments (ASP16-1 and ASP16-2) are indicated by red arrows, additional HPV16 primers by green arrows. Red and green lines symbolize sequence reads of 100 nt. The gray vertical arrows (a-f) indicate the positions of possible HPV16 integration breakpoints. Since the sequence read length was ~100 nt but the primers (red arrows) were ~200 bp apart, gaps of about 100 nt with no sequence information were obtained. Addition of new HPV16 primers (green arrows) shortened the average distance between primers to 100 nt, thereby improving the probability for complete sequence coverage.

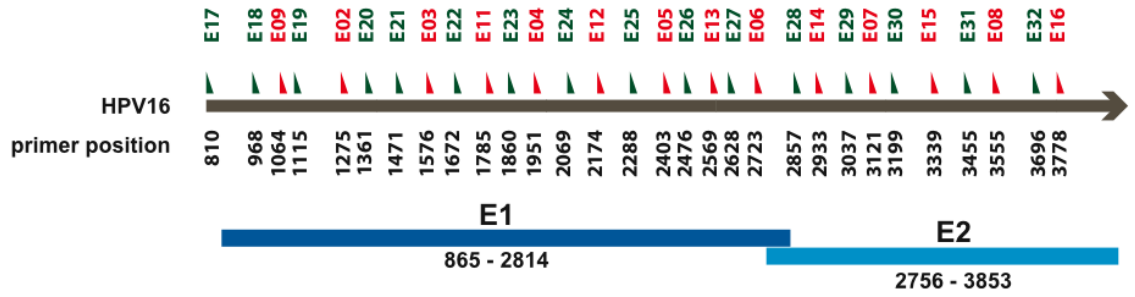


Figure 2.33: Locations of RA_HPV16 primers in ASP16-3 and ASP16-4. All thirty RA_HPV16 primers used in ASP16-3 and ASP16-4 are shown. All primers are in forward direction and cover almost the complete region of the HPV16 E1 and E2 genes. The red and green triangles indicate positions of previous primers (used in ASP16-2) and additional primers (added since ASP16-3), respectively. The primers were named with numbers according to their entries in the experiment. The positions of ORFs E1 and E2 are indicated at the bottom.

2.2.2.2 DNA samples in ASP16-3 and ASP16-4

Altogether 25 DNA samples were analyzed in ASP16-3 and ASP16-4 (Table 2.13). Four samples were HPV16-positive cervical carcinoma cell lines and 21 samples were cervical scrapes. The four cervical cell lines served as controls because their integration junctions are known. The cervical DNA samples were selected for the experiments based on their high probabilities of having integrated HPV16 (integration %). The integration percentage values were obtained through E2/E6 RT-qPCR (described in Introduction). From the clinical DNA samples, nine were analyzed in both experiments.

2.2.2.3 HPV16 amplicon preparation for ASP16-3 and ASP16-4 sequencing

To prepare HPV16 amplicons for pyrosequencing, the genomic DNA of each sample was subjected to several amplification steps, as described in Introduction and Materials and Methods. In ASP16-3 and ASP16-4, thirty biotin-labeled HPV16 forward primers were used during the linear amplification step, distributed into four primer mixes (Table 2.14). The 7 or 8 HPV16 primers of each primer mix locate at about 400 bp apart on the HPV16 genome. The HPV16 amplicons of each DNA sample were generated by HPV16 semi-nested multiplex PCR using the corresponding bipartite RA_HPV16 nested forward primers and a tripartite reverse primer (Roche-B_barcode_GPUA) (Table 2.14, see also Figure 2.33).

Table 2.13: DNA samples in ASP16-3 and ASP16-4.

Sample Name	ASP16-2 ID*	ASP16-3 ID	ASP16-4 ID	Integration % (a)	Cytology/ histology (b)	Origin (c)
MRI-H186	2B01	3B01	4B01	49	**cancer**	cell line
MRI-H196	2B02	3B02	4B02	32	**cancer**	cell line
SiHa	2B03	3B03	4B03		**cancer**	cell line
CaSki			4B04		**cancer**	cell line
1		3B04		76	CIN 2/3	Besancon
2		3B05		74	CIN 2/3	Besancon
5		3B06		95	CIN 2/3	Besancon
1503		3B07	4B12	74	CIN 2/3	Besancon
1511		3B08	4B13	87	CIN 2/3	Besancon
1801		3B09	4B14	72	CIN 2/3	Besancon
2219		3B10	4B15	72	CIN 2/3	Besancon
2227		3B11	4B16	99	CIN 2/3	Besancon
2229		3B12		76	CIN 2/3	Besancon
2237		3B13	4B17	98	CIN 2/3	Besancon
3009		3B14	4B18	81	CIN 2/3	Besancon
3035		3B15		74	CIN 2/3	Besancon
4242		3B16	4B20	79	CIN 2/3	Besancon
66019		3B18	4B06	**100**	HSIL/CIN3	Reims
07 C 381	2B06		4B05	**100**	**cancer**	Reims
61979			4B07	**100**	HSIL	Reims
75022			4B08	**100**	LSIL	Reims
75857			4B09	**100**	HSIL	Reims
4			4B11	82	CIN 2/3	Besancon
4238a			4B19	64	CIN 2/3	Besancon
07 C 368	2B05		4B21	84	**cancer**	Reims

* ASP16-2 was performed by Bo Xu (Xu, 2010).
(a) The integration percentage was determined by E2/E6 real-time quantitative PCR as described by (Briolat et al, 2007). See Introduction for further explanation.
(b) CIN: cervical intraepithelial neoplasia. HSIL: high-grade squamous intraepithelial lesion.
(c) The clinical DNA samples were obtained from Reims or Besancon.

Table 2.14: Primer combinations for HPV16 amplification in ASP16-3 and ASP16-4.

	HPV16 linear amplification	Semi-nested HPV16 multiplex PCR	
	(FORWARD) Biotin-labeled HPV16 primer [(a)]	(nested FORWARD) RA_HPV16 primer [(b)]	(REVERSE) Roche-B_barcode_GPUA primer [(c)]
Primer mix 1	5B-790F (BE-17)* 5B-1261F (BE-02) 5B-1562F (BE-03) 5B-1938F (BE-04) 5B-2389F (BE-05) 5B-2705F (BE-06) 5B-3101F (BE-07) 5B-3542F (BE-08)	E17 (810)* E02 (1275) E03 (1576) E04 (1951) E05 (2403) E06 (2723) E07 (3121) E08 (3555)	RB-B01, RB-B02, ... or RB-B24
Primer mix 2	5B-1046F (BE-09) 5B-1457F (BE-21) 5B-1760F (BE-11) 5B-2151F (BE-12) 5B-2539F (BE-13) 5B-2912F (BE-14) 5B-3318F (BE-15) 5B-3762F (BE-16)	E09 (1064) E21 (1471) E11 (1785) E12 (2174) E13 (2569) E14 (2933) E15 (3339) E16 (3778)	
Primer mix 3	5B-952F (BE-18) 5B-1336F (BE-20) 5B-1849F (BE-23) 5B-2277F (BE-25) 5B-2613F (BE-27) 5B-3021F (BE-29) 5B-3444F (BE-31)	E18 (968) E20 (1361) E23 (1860) E25 (2288) E27 (2628) E29 (3037) E31 (3455)	
Primer mix 4	5B-1105F (BE-19) 5B-1653F (BE-22) 5B-2053F (BE-24) 5B-2461F (BE-26) 5B-2842F (BE-28) 5B-3189F (BE-30) 5B-3680F (BE-32)	E19 (1115) E22 (1672) E24 (2069) E26 (2476) E28 (2857) E30 (3199) E32 (3696)	

* Primers BE-17 and E17 were not used in ASP16-4

(a) BE = biotin-labeled HPV16 primers.

(b) E = RA_HPV16 primers. The numbers in brackets are the positions (5'end) of the primers on HPV16R genome.

(c) RB = Roche-B_barcode_GPUA primers.

In ASP16-3, the HPV16 amplicons were analyzed by agarose gel electrophoresis and Southern hybridization, to determine whether the amplicons were HPV16-specific. The agarose gels and hybridization results are shown in Figure 2.34. The HPV16 PCR products were expected to contain fragments of variable sizes, producing smears on the gel. This expectation was fulfilled in the majority of cases. Some amplicons showed prominent bands (such as panel B, lane 9-3 or panel D, lane 13-2). These bands were not HPV16-specific, as no signal was observed after hybridization. The majority of the amplicons contained HPV16-specific fragments distributed between 100-600 bp. Many amplicons contained even larger fragments with sizes up to 1.5-2.0 kb. Aliquots of 100 ng from each amplicon were pooled together for Roche/454 GS-FLX sequencing. In ASP16-4, each HPV16 amplicon was size-selected to 200-300 bp using E-Gel SizeSelect system (Invitrogen), before being pooled together for sequencing, as described in Materials and Methods.

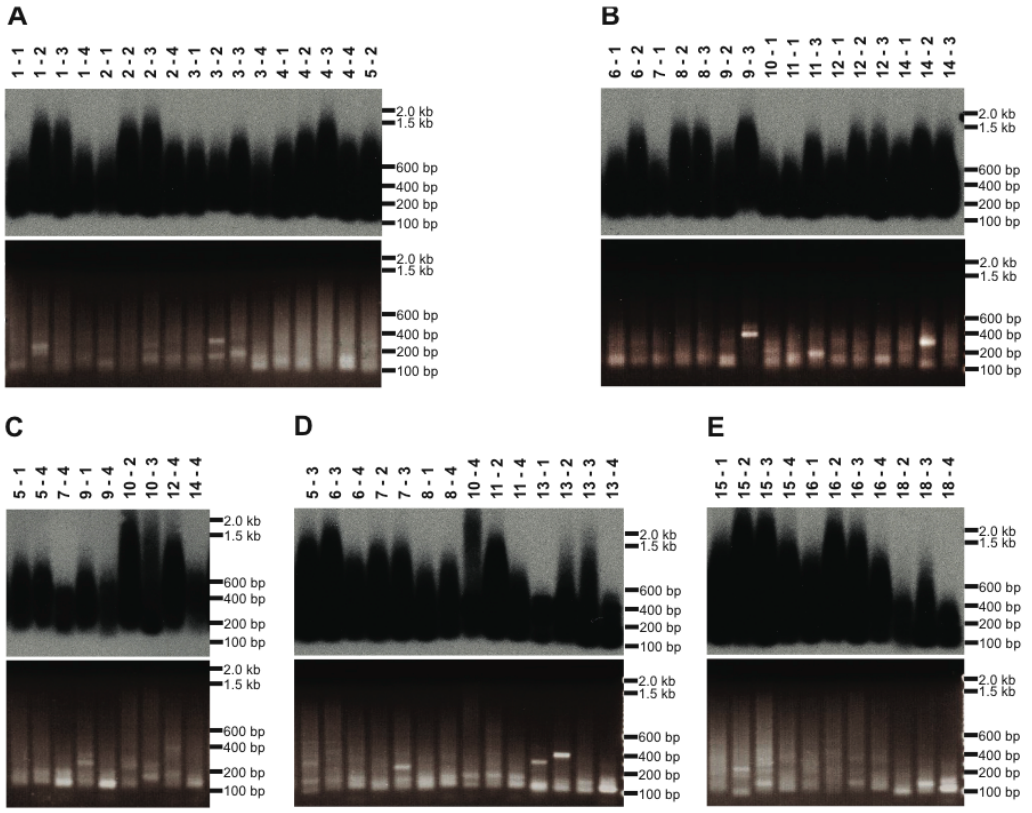


Figure 2.34: HPV16 multiplex PCR products in ASP16-3. Each panel (A-E) shows an agarose gel photo in the lower part and the Southern blot in the upper part. 50-ng aliquots of HPV16 multiplex PCR amplicons were loaded on the gel. The labeling of each lane indicates the barcode number (1, 2, 3,…, 24), followed by the multiplex reaction mix number (1, 2, 3, 4). The gels were blotted and hybridized with complete HPV16 genome as ^{32}P-labelled probe.

2.2.2.4 Amplicon sequencing by Roche/454 GS-FLX pyrosequencing

The pooled HPV16 amplicons were sequenced, using the Roche/454 GS-FLX standard system, at the DKFZ Genomics and Proteomics Core Facilities. GS emPCR kit III was used for emulsion PCR and GS LR70 as the sequencing kit, with one large region on a 70x75 picotiter plate used. Roche-B was used as the sequencing primer. The manufacturer's estimated number of sequence reads per sequencing run for the described sequencing format is 210000. The results were delivered as FASTA files, containing complete sequence reads of each experiment. The basic features of both experiments are shown in Table 2.15, in comparison with the results of the previous experiment ASP16-2 (Xu, 2010).

Table 2.15: Features of ASP16-2, ASP16-3 and ASP16-4

	ASP16-2*	ASP16-3	ASP16-4
Number of DNA samples	19	17	20
Number of HPV16 primer combinations	16	30	29
Number of total sequence reads	220001	152147	129150
Average sequence read length, excluding 19-nt Roche-B sequence (nt)	92	105	108
Average number of sequence reads per primer group per DNA sample	724	298	222

* Experiment ASP16-2 was conducted by Bo Xu (Xu, 2010).

2.2.2.5 Statistics of ASP16-3 and ASP16-4

The FASTA files containing the total sequence reads of each experiment, were initially analyzed with the four ASP16 analysis program sets, described in section 2.2.1.4. The numbers of sequence reads per primer group per sample group, before and after filtering with 28-nt cutoff, are shown in Table 2.16 for ASP16-3 and in Table 2.17 for ASP16-4.

To evaluate whether the HPV16 primers produced HPV16-specific sequences, the percentage of significant HPV16 sequence reads for each sample and each primer group was calculated with the following formula:

$$\frac{(number\_of\_sequence\_reads\_after\_28nt\_cutoff)}{(number\_of\_sequence\_reads\_before\_28nt\_cutoff)} \times 100\%$$

For each primer group, the distribution of the percentages of significant HPV16 sequences from all DNA sample are shown in boxplot diagrams (Figure 2.35). Twenty-eight of the thirty primers showed high specificity of more than 50%, which means that more than 50% of the sequence reads in the primer groups were HPV16-specific. Primer E17 (810) showed a reduced efficiency. Because it is located in the 5' part of the E1 gene where integration is unlikely to occur, it was not used in ASP16-4 and no alternative primer was added to replace primer E17 (810). Primer E21 (1471) also showed reduced efficiency, but was included also in ASP16-4.

Table 2.16: Number of sequence reads per primer group of ASP16-3 before and after 28-nt cutoff.

A

NO CF	810	968	1064	1115	1275	1361	1471	1576	1672	1785	1860	1951	2069	2174	2288	2403	2476	2569	2628	2723	2857	2933	3037	3121	3199	3339	3455	3555	3696	3778	Sum sigEE sigHPV	Total per sample
	E17	E18	E09	E19	E02	E20	E21	E03	E22	E11	E23	E04	E24	E12	E25	E05	E26	E13	E27	E06	E28	E14	E29	E07	E30	E15	E31	E08	E32	E16		
3B01	7	294	464	295	521	287	127	554	240	98	308	355	255	71	38	15	42	71	79	214	123	9	92	263	184	279	195	438	328	122	6368	6587
3B02	13	271	629	346	25	147	153	721	222	100	314	290	336	37	43	9	42	27	43	66	109	31	111	376	279	360	247	652	517	420	6936	7116
3B03	10	137	500	220	167	90	195	512	283	104	423	132	553	135	254	19	56	23	194	109	149	37	386	28	32	51	179	498	529	200	6205	6361
3B04	81	294	1045	378	444	165	389	910	486	201	391	408	554	284	237	30	81	86	140	342	233	99	378	538	495	713	266	784	694	681	11827	12149
3B05	12	189	1083	332	401	58	337	913	425	149	451	283	605	251	188	18	56	45	117	161	140	74	298	254	386	659	234	887	600	720	10326	10612
3B06	8	214	1250	418	885	69	157	1156	362	141	444	343	518	117	70	47	67	30	18	232	13	60	188	684	551	333	333	690	134	110	9642	9870
3B07	63	504	975	420	592	148	259	1101	387	173	611	345	913	130	206	71	121	67	142	234	225	33	218	336	380	834	374	1009	498	545	11914	12166
3B08	19	538	1395	392	325	160	195	923	328	196	820	177	470	172	289	72	173	86	75	255	9	18	433	406	602	434	408	434	36	94	9934	10173
3B09	17	61	829	197	71	16	163	780	373	111	396	70	740	117	234	16	75	47	141	92	194	27	254	184	170	650	236	772	136	455	7624	7856
3B10	55	423	980	176	453	78	23	1559	248	80	188	131	297	37	100	27	128	13	8	141	24	1	2	115	205	390	634	662	106	2	7286	7572
3B11	19	374	832	271	536	206	161	1136	277	125	811	306	649	100	148	23	73	26	138	266	218	11	370	471	400	477	457	1157	482	398	10918	11194
3B12	2	376	918	323	700	361	340	1135	450	207	596	223	554	337	101	28	146	38	176	230	144	24	218	330	492	676	561	1092	306	588	11672	12013
3B13	70	367	1210	342	90	50	44	1188	266	59	768	205	804	18	385	4	81	24	30	119	6	21	22	106	598	208	647	202	21	57	8012	8221
3B14	9	61	890	426	223	1	104	720	276	72	396	156	684	112	14	7	50	16	8	114	20	82	230	272	596	324	309	473	93	148	6886	7073
3B15	2	438	591	400	359	11	64	832	274	91	872	229	543	124	21	19	86	58	31	214	70	208	170	288	527	326	541	775	244	358	8766	9025
3B16	1	216	460	284	397	29	82	562	135	242	373	213	514	145	3	5	55	16	13	106	16	49	46	135	485	266	313	555	53	157	5926	6093
3B18	1	631	1202	167	3	160	193	2	280	32	376	0	1285	26	303	0	28	9	17	1	1	9	17	1	143	86	28	3	28	60	5092	5237

B

CF =28	810	968	1064	1115	1275	1361	1471	1576	1672	1785	1860	1951	2069	2174	2288	2403	2476	2569	2628	2723	2857	2933	3037	3121	3199	3339	3455	3555	3696	3778	Sum sigEE sigHPV	Total per sample
	E17	E18	E09	E19	E02	E20	E21	E03	E22	E11	E23	E04	E24	E12	E25	E05	E26	E13	E27	E06	E28	E14	E29	E07	E30	E15	E31	E08	E32	E16		
3B01	7	294	338	291	489	287	52	519	240	93	204	327	252	66	30	15	37	64	69	186	117	9	89	172	184	204	131	312	270	97	5445	6587
3B02	9	270	461	341	23	147	55	606	219	85	218	283	334	37	33	6	40	25	30	53	107	30	103	259	278	289	155	484	463	328	5771	7116
3B03	2	125	338	206	160	81	64	408	259	68	283	125	537	119	47	13	26	20	143	91	138	32	357	1	0	0	0	341	389	134	4507	6361
3B04	17	270	681	364	417	151	102	676	467	159	177	364	545	246	32	13	44	67	100	273	208	80	340	287	470	487	158	508	605	439	8747	12149
3B05	5	182	638	323	385	55	91	741	412	107	215	248	593	198	35	15	34	29	85	144	127	59	272	133	364	433	122	599	520	406	7570	10612
3B06	6	214	885	397	852	66	49	993	358	118	264	312	506	109	56	37	49	25	12	200	13	54	164	500	548	256	224	558	118	66	8009	9870
3B07	10	483	613	348	516	134	112	778	337	126	374	255	885	106	130	50	69	51	114	190	203	27	206	188	314	546	197	642	400	363	8767	12166
3B08	7	531	939	362	299	159	87	691	319	158	543	137	455	159	233	46	132	80	62	191	9	18	379	277	574	384	228	349	19	42	7869	10173
3B09	5	55	458	151	57	15	56	576	358	75	256	53	712	93	64	11	38	40	106	72	177	18	234	113	166	275	114	506	112	295	5261	7856
3B10	0	421	844	139	445	76	18	803	174	74	138	80	230	15	86	19	40	13	8	31	7	1	2	2	45	383	325	282	12	0	4713	7572
3B11	8	365	562	243	475	198	58	925	274	89	495	273	641	85	133	16	45	16	110	214	206	11	336	365	377	353	245	849	373	278	8618	11194
3B12	0	375	697	314	668	359	222	875	446	195	398	194	550	320	88	22	83	32	144	203	141	24	206	232	492	500	358	768	259	429	9594	12013
3B13	0	332	718	324	81	19	11	581	255	45	480	186	801	9	3	2	36	19	22	16	2	10	12	57	550	54	347	105	15	25	5117	8221
3B14	6	61	603	393	209	1	24	607	276	52	258	140	671	96	8	3	27	11	5	98	20	72	219	212	595	294	203	393	87	114	5758	7073
3B15	2	435	454	387	351	11	23	646	272	81	573	221	542	122	18	19	74	58	27	195	69	186	157	225	526	309	383	655	237	209	7467	9025
3B16	0	213	320	274	390	29	29	478	135	207	237	192	505	139	2	5	39	16	12	95	16	47	42	124	482	221	229	467	53	102	5100	6093
3B18	1	615	525	162	3	139	13	1	275	14	273	0	1198	12	84	0	14	0	8	1	0	2	6	1	1	0	3	3	0	19	3373	5237

Panel A shows the numbers before the 28-nt cutoff filtering was applied.
Panel B shows the numbers after the 28-nt cutoff filtering was applied.
The column Sum-sigEE-sigHPV contains sums of sequence reads per sample with significant primer hit.
The column Total-per-sample shows the sum of total sequence reads per sample before any selection process.
Gray areas indicate values excluded from analysis.

Table 2.17: Number of sequence reads per primer group of ASP16-4 before and after 28-nt cutoff.

A

NO CF	810	968	1064	1115	1275	1361	1471	1576	1672	1785	1860	1951	2069	2174	2288	2403	2476	2569	2628	2723	2857	2933	3037	3121	3199	3339	3455	3555	3696	3778	Sum sigEE sigHPV	Total per sample
	E17	E18	E09	E19	E02	E20	E21	E03	E22	E11	E23	E04	E24	E12	E25	E05	E26	E13	E27	E06	E28	E14	E29	E07	E30	E15	E31	E08	E32	E16		
4B01	0	208	404	495	89	77	39	645	259	36	441	329	271	119	45	39	78	19	59	540	67	107	95	159	161	163	360	382	408	132	6226	6620
4B02	0	259	732	399	13	58	90	936	303	113	542	259	715	118	96	27	44	26	83	206	115	97	231	226	463	652	855	403	535	227	8823	9456
4B03	0	410	657	631	189	219	34	400	303	116	609	383	546	241	248	16	64	31	94	239	93	110	258	1	44	6	6	312	507	93	6860	7411
4B04	0	255	602	1972	134	182	46	499	555	52	288	188	396	97	34	34	252	22	23	172	156	50	177	313	930	297	382	324	856	278	9566	10491
4B05	0	387	723	288	168	133	40	641	134	125	640	131	253	149	177	35	23	44	108	53	1	2	3	0	3	15	5	2	1	2	4286	4581
4B06	0	165	524	76	4	139	78	230	4	28	92	6	11	44	344	2	13	7	9	9	0	4	6	3	7	91	19	13	1	19	1948	2191
4B07	0	279	272	90	3	59	30	357	1	44	65	2	19	80	320	0	6	3	4	7	1	4	0	8	4	49	50	3	0	8	1768	1982
4B08	0	81	472	16	26	77	83	151	20	21	130	2	22	12	534	3	0	17	8	6	2	19	41	9	25	155	140	42	4	27	2145	2392
4B09	0	141	452	1479	51	18	21	455	598	57	369	213	368	185	52	33	123	62	48	219	112	91	258	279	606	670	516	756	987	247	9466	10049
4B11	0	195	572	720	164	77	44	677	372	66	428	206	363	97	45	19	51	13	40	150	48	86	157	209	458	564	556	509	612	238	7736	8291
4B12	2	423	688	535	126	175	107	629	340	122	670	140	321	158	209	39	97	24	122	106	69	36	174	192	432	675	525	357	411	216	8120	8755
4B13	0	298	817	1216	125	15	61	654	509	94	522	145	414	144	268	83	149	48	40	158	7	20	216	242	902	538	469	336	62	50	8602	9344
4B14	0	137	392	249	50	35	131	626	199	85	411	118	232	110	88	28	35	28	64	114	38	19	124	124	254	484	416	287	228	168	5274	5618
4B15	0	167	540	114	130	27	103	261	225	62	93	4	171	91	207	11	36	44	12	27	11	30	10	72	180	227	256	97	38	126	3372	3863
4B16	0	48	500	364	47	12	84	337	415	22	469	41	280	143	79	29	79	30	93	86	101	31	207	133	435	231	389	365	526	197	5773	6105
4B17	0	86	351	114	32	24	66	286	62	58	159	32	66	36	310	7	17	13	17	22	2	22	22	76	119	165	209	117	11	28	2529	2960
4B18	0	122	643	681	8	0	25	132	277	43	387	15	358	325	62	52	72	88	23	330	11	260	248	417	421	251	394	425	246	260	6576	7123
4B19	0	72	397	334	24	16	22	261	302	14	481	53	206	31	28	23	68	14	14	203	62	108	276	428	255	364	494	479	375	336	5740	6107
4B20	0	127	352	840	30	6	46	192	372	21	196	6	271	156	9	18	54	54	13	200	23	193	262	257	838	164	546	288	232	137	5903	6404
4B21	0	458	584	402	137	65	71	1017	232	96	648	219	207	57	17	8	8	25	38	59	36	26	125	103	310	558	238	385	290	101	6520	7074

B

CF =28	810	968	1064	1115	1275	1361	1471	1576	1672	1785	1860	1951	2069	2174	2288	2403	2476	2569	2628	2723	2857	2933	3037	3121	3199	3339	3455	3555	3696	3778	Sum sigEE sigHPV	Total per sample
	E17	E18	E09	E19	E02	E20	E21	E03	E22	E11	E23	E04	E24	E12	E25	E05	E26	E13	E27	E06	E28	E14	E29	E07	E30	E15	E31	E08	E32	E16		
4B01	0	207	264	495	84	77	14	615	258	33	309	324	264	115	27	39	69	18	55	493	66	102	90	121	161	151	290	311	384	118	5554	6620
4B02	0	259	422	398	10	57	65	859	299	100	368	244	713	105	74	25	37	23	81	201	109	91	228	171	463	607	718	283	479	202	7691	9456
4B03	0	410	435	626	185	217	15	372	302	110	489	380	542	229	226	15	61	31	94	238	84	103	234	1	42	0	4	249	460	87	6241	7411
4B04	0	255	378	1971	130	181	23	451	554	47	214	181	390	85	32	32	223	21	22	170	153	44	171	236	930	276	295	234	829	254	8782	10491
4B05	0	382	456	287	152	130	23	582	132	122	481	125	246	131	112	27	18	23	95	48	0	0	1	0	0	5	0	0	0	0	3578	4581
4B06	0	133	191	76	0	109	3	57	1	2	23	5	5	44	30	2	12	0	1	0	0	0	1	0	0	2	0	0	0	0	697	2191
4B07	0	248	37	90	0	52	5	157	1	9	36	0	17	45	55	0	6	0	0	0	1	0	0	0	0	0	1	0	0	0	760	1982
4B08	0	14	5	16	3	31	8	63	13	0	14	0	1	0	3	0	0	0	0	0	1	0	0	0	0	1	34	3	0	2	212	2392
4B09	0	139	278	1470	46	16	7	384	597	44	239	197	360	151	42	27	108	55	43	217	107	81	251	206	603	583	364	531	954	197	8297	10049
4B11	0	194	331	717	157	76	24	591	372	61	291	189	361	87	30	18	48	9	33	148	46	76	145	156	456	524	427	332	571	206	6676	8291
4B12	2	413	396	528	112	167	65	492	335	110	443	138	315	149	159	34	90	22	106	96	62	32	166	130	431	600	361	251	371	189	6765	8755
4B13	0	288	445	1206	92	14	43	579	504	78	333	129	405	111	241	68	133	46	36	152	7	13	208	187	900	519	351	261	60	36	7445	9344
4B14	0	133	185	245	43	31	83	550	191	74	275	112	227	95	64	25	29	24	57	102	35	19	121	87	247	348	281	185	223	130	4221	5618
4B15	0	142	136	74	111	12	21	160	159	23	51	4	144	11	45	4	16	12	6	23	7	3	2	4	106	96	118	21	2	0	1513	3863
4B16	0	48	289	362	45	12	56	288	410	17	265	29	272	126	52	26	41	29	83	76	96	25	194	98	426	164	275	272	481	166	4723	6105
4B17	0	70	85	113	9	6	3	77	36	10	66	20	59	24	5	0	6	3	2	15	2	3	5	6	75	72	161	4	3	1	941	2960
4B18	0	121	410	678	8	0	14	120	277	30	243	14	354	276	46	50	62	82	17	313	11	227	243	346	418	239	316	334	244	208	5701	7123
4B19	0	72	235	330	21	16	3	231	301	12	238	47	205	27	23	20	57	14	11	197	59	97	264	275	253	328	362	325	351	277	4651	6107
4B20	0	127	198	839	26	5	24	181	368	18	122	6	265	142	4	17	47	52	12	189	22	182	246	216	832	159	435	227	226	113	5300	6404
4B21	0	458	357	385	121	62	47	925	231	85	424	219	206	49	9	7	8	24	33	59	32	25	123	96	310	522	194	328	290	63	5692	7074

Panel A shows the numbers before the 28-nt cutoff filtering was applied.
Panel B shows the numbers after the 28-nt cutoff filtering was applied.
The column Sum-sigEE-sigHPV contains sums of sequence reads per sample with significant primer hit.
The column Total-per-sample shows the sum of total sequence reads per sample before any selection process.

To determine the distribution of sequence read lengths, histograms of sequence read lengths were plotted for ASP16-2, ASP16-3 and ASP16-4 (Figure 2.36). There were four apparent intervals with peaks at 55, 75, 95 and 150 nt, where most of the sequences fit. For ASP16-2, most sequence reads in the 55-nt peak belonged to primer-dimer products of HPV16 primer E10 (Xu, 2010). This primer was omitted in ASP16-3 and ASP16-4. Despite lower sequence read numbers in ASP16-3 and ASP16-4 compared to ASP16-2 (Table 2.15), the histograms show that higher numbers of long sequence reads (150 nt and longer) have been obtained in ASP16-3 and ASP16-4 compared to ASP16-2. This

indicates that the alterations of linear amplification reaction condition by increasing the elongation time from 30 to 40 seconds have been successful (see Materials and Methods).

To evaluate whether the sequence reads in each size-interval were HPV16-specific, histograms of sequence read lengths before and after 28-nt cutoff filtering were plotted (Figure 2.37). The histograms show that the 75-nt peaks of ASP16-3 and ASP16-4 contain large portions of sequences that are not HPV16-specific. Nevertheless, these sequences are too short to be informative. In contrast, the longer sequence reads (from 95 nt) of both ASP16-3 and ASP16-4 were almost 100% HPV16-specific, and therefore contained informative sequence data.

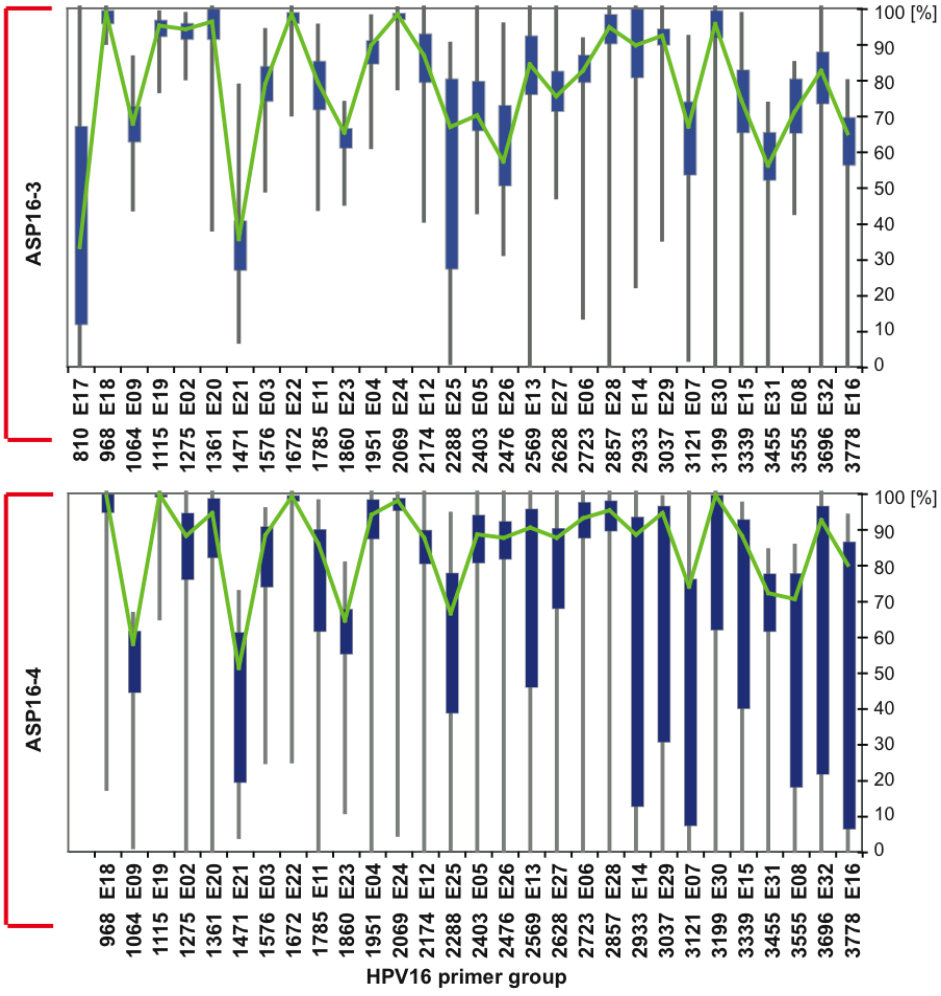


Figure 2.35: HPV16 primer efficiency in ASP16-3 and ASP16-4. Each boxplot shows the distribution (in percentage) of significant HPV16 sequence reads for each HPV16 primer group of all DNA samples. Each blue box contains values from the 25th to 75th percentile. The lines above and below the box represent values above the 75th and below the 25th percentile, respectively. Green lines were plotted from median percentage values of each primer group.

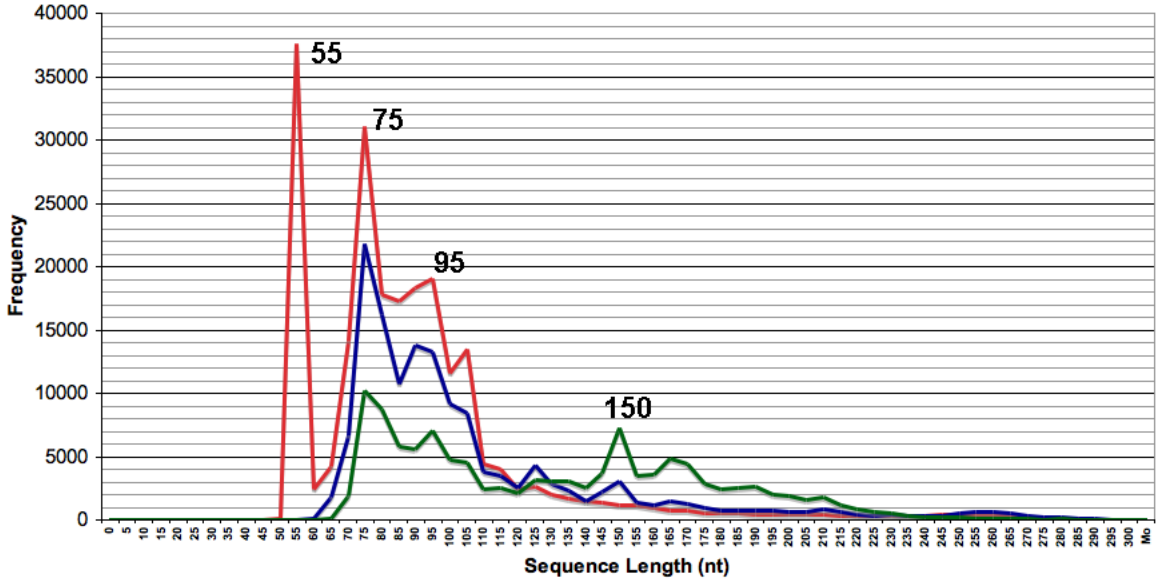


Figure 2.36: Sequence read lengths of ASP16-2, ASP16-3 and ASP16-4. The histograms of sequence read lengths before filtering by 28-nt cutoff are shown. X-axis represents 5-nt intervals of sequence read length. Y-axis indicates the frequency with which sequence read lengths fall into the interval categories. The red line represents ASP16-2, the blue line ASP16-3 and the green line ASP16-4.

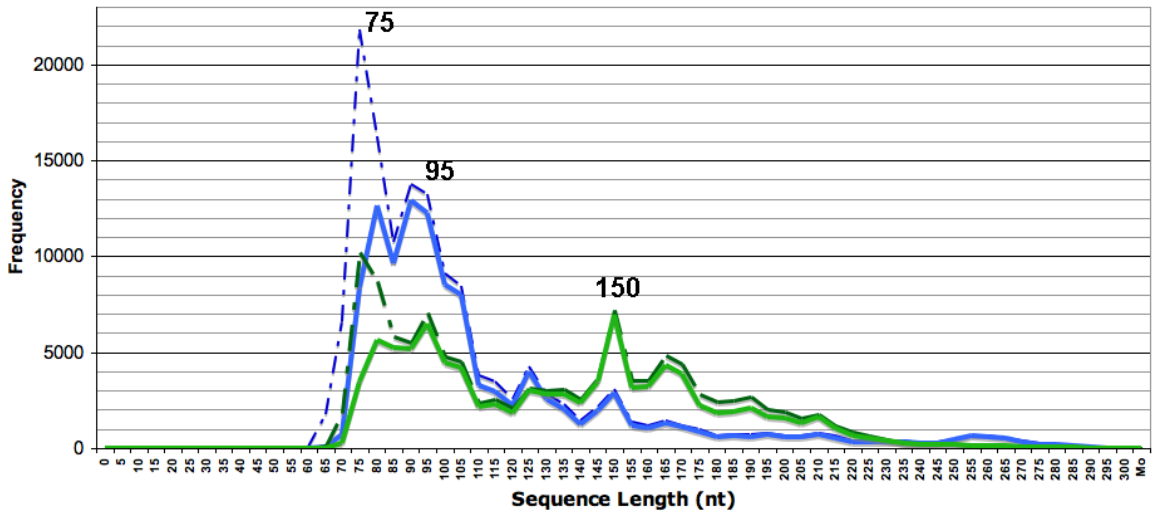


Figure 2.37: Sequence read lengths of ASP16-3 and ASP16-4 before and after filtering with a 28-nt cutoff. The histograms of sequence read lengths before and after filtering by 28-nt cutoff are shown by dashed and solid lines, respectively. X-axis represents 5-nt intervals of sequence read length. Y-axis indicates the frequency with which sequence read lengths fall into the interval categories. The blue lines represent ASP16-3 and the green lines ASP16-4.

2.2.2.6 HPV16 integration junctions

To determine whether the sequences contain possible HPV16 integration junctions, two output data after the ASP16 analysis programs were analyzed.

As first approximate method, the numbers of reads per each primer group of each sample after 28-nt cutoff (shown in Table 2.16 and Table 2.17, both in panel B) were inspected. If the sample contains exclusively integrated HPV16 DNA, the number of reads of the primer groups at and after the HPV16 breakpoint should drop to zero. As an example, the results for the cell line SiHa are shown in Figure 2.38. For the integrated HPV16 genome in SiHa, it is known that the 5' junction is located at pos. 3385 and the 3' junction at pos. 3133. Further more, a small region from pos. 3460-3512 is deleted (Meissner, 1999). In complete agreement with this structure, the read numbers are 1 or zero for primers E07 (3121), E30 (3199), E15 (3339) and E31 (3455) (Figure 2.38).

As the main method, the sequence alignments of each primer group for each DNA sample were analyzed. In this way, the nucleotide sequences at the HPV16 breakpoint location were determined. The alignments were viewed and edited in Geneious Pro version 4.8.5. For illustration, the results for SiHa are shown in Figure 2.39.

Applying these two methods, HPV16 integration junctions were identified in altogether 9 samples in ASP16-3 and ASP16-4: 3 of 4 cell lines, and 6 of 21 clinical samples (Table 2.18). In case of newly identified HPV16 integration junctions, they were examined by junction-specific PCR (Figure 2.40). The positive PCR products were cloned and sequenced. If the cloned sequences confirmed the ASP16 sequences, the junction was declared to be genuine. The next parts describe the determination of HPV16 integrations for all DNA samples listed in Table 2.13, compare the results of the different ASP16 experiments, and examine the ASP16 integration results in relation to those of E2/E6 qPCR.

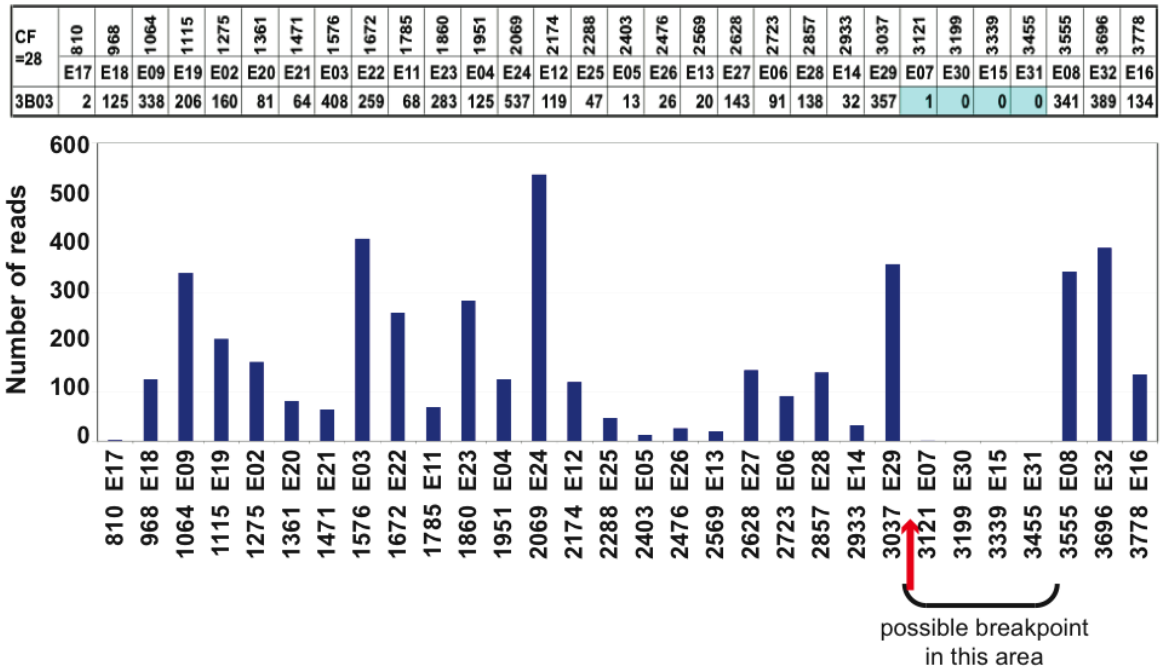


Figure 2.38: HPV16 breakpoint location in SiHa predicted by number of sequence reads. The numbers of reads of each primer group of SiHa in ASP16-3 after 28-nt cutoff, taken from Table 2.16, are shown at the top. A graph was plotted from these values, with X-axis showing the primer groups and Y-axis the number of reads. The possible breakpoint area is shown. The actual HPV16 breakpoint of SiHa is at position 3133 (red arrow).

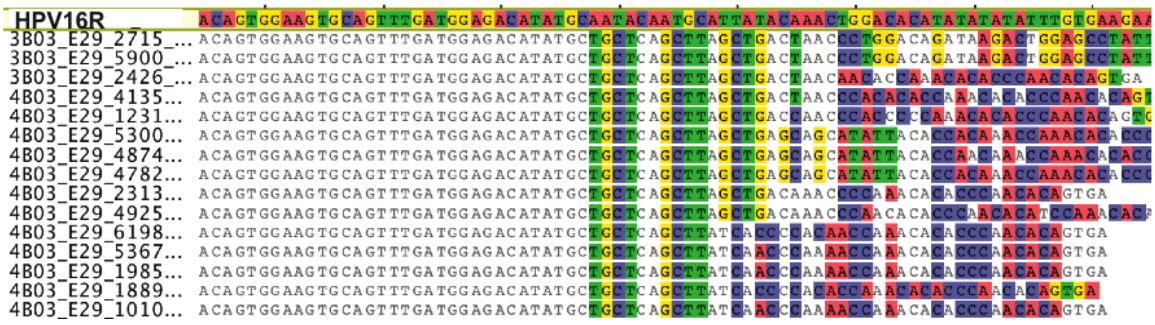


Figure 2.39: Sequence alignment of primer group E29 (3037) of SiHa with the HPV16 integration junction. Bases are color-coded: A in red, C in blue, G in yellow and T in green. The names of the sequences are indicated on the left. HPV16R (top) was used as the reference sequence and every base was highlighted. The sequences in primer group E29 of SiHa in this alignment were taken from both ASP16-3 (3B03) and ASP16-4 (4B03) as the names indicate. The bases in SiHa sequences are highlighted only if they are different from the reference. The highlighted region indicates cellular DNA downstream of the HPV16 integration breakpoint at pos. 3133.

Table 2.18: Identification of HPV16 integration junctions in ASP16 experiments.

Sample name [(a)]	Previous identification: Reference (Method)	Identification in ASP16-2 [(b)]	Identification in ASP16-3	Identification in ASP16-4	Integration percentage by E2/E6 qPCR[(c)]
Cell lines					
MRI-H186	Xu, 2010 (RS-PCR)	yes	yes	yes	49%
MRI-H196	Xu, 2010 (RS-PCR)	(yes) [(d)]	yes	yes	32%
SiHa	Baker, 1987 (Cloning, sequencing)	no	yes	yes	100% [(f)]
CaSki	Smits, 1991 (cDNA sequencing)	no		no	0% [(f)]
Clinical samples for which HPV16 integration breakpoints were identified by ASP16					
CA-07C381		yes		yes	100%
CA-07C368		yes [(e)]		yes	84%
HSIL-66019			yes	yes	100%
HSIL-61979				yes	100%
HSIL-75857				yes	100%
CIN2/3-1801			yes	yes	72%
Clinical samples for which HPV16 integration breakpoints were not identified by ASP16*					
CIN2/3-4242			false-positive	no	79%
CIN2/3-1503			false-positive	no	74%
LSIL-75022				no	100%
CIN2/3-2227			no	no	99%
CIN2/3-2237			no	no	98%
CIN2/3-0005			no		95%
CIN2/3-1511			no	no	87%
CIN2/3-0004				no	82%
CIN2/3-3009			no	no	81%
CIN2/3-0001			no		76%
CIN2/3-2229			no		76%
CIN2/3-0002			no		74%
CIN2/3-3035			no		74%
CIN2/3-2219			no	no	72%
CIN2/3-4238a				no	64%

(a) The sample names for cervical lesions start with the cytological/histological status. CA: cancer. CIN: cervical intraepithelial neoplasia. HSIL: high-grade squamous intraepithelial lesion. LSIL: low-grade squamous intraepithelial lesion.
(b) ASP16-2 was performed by Bo Xu (Xu, 2010).
(c) See Introduction for details of E2/E6 qPCR.
(d) One ASP16 read contained only 6 bp of cellular sequences, too short to be identified by Blast. The sequence was identified as integration junction by comparison with the RS-PCR sequence.
(e) Artifact integration junction was identified from the ASP16 reads. The genuine junction was identified via cloning and sequencing, see text for details.
(f) Estimated integration percentages.
* Ordering of samples by decreasing E2/E6 qPCR values for integration, except for samples CIN2/3-4242 and CIN2/3-1503.

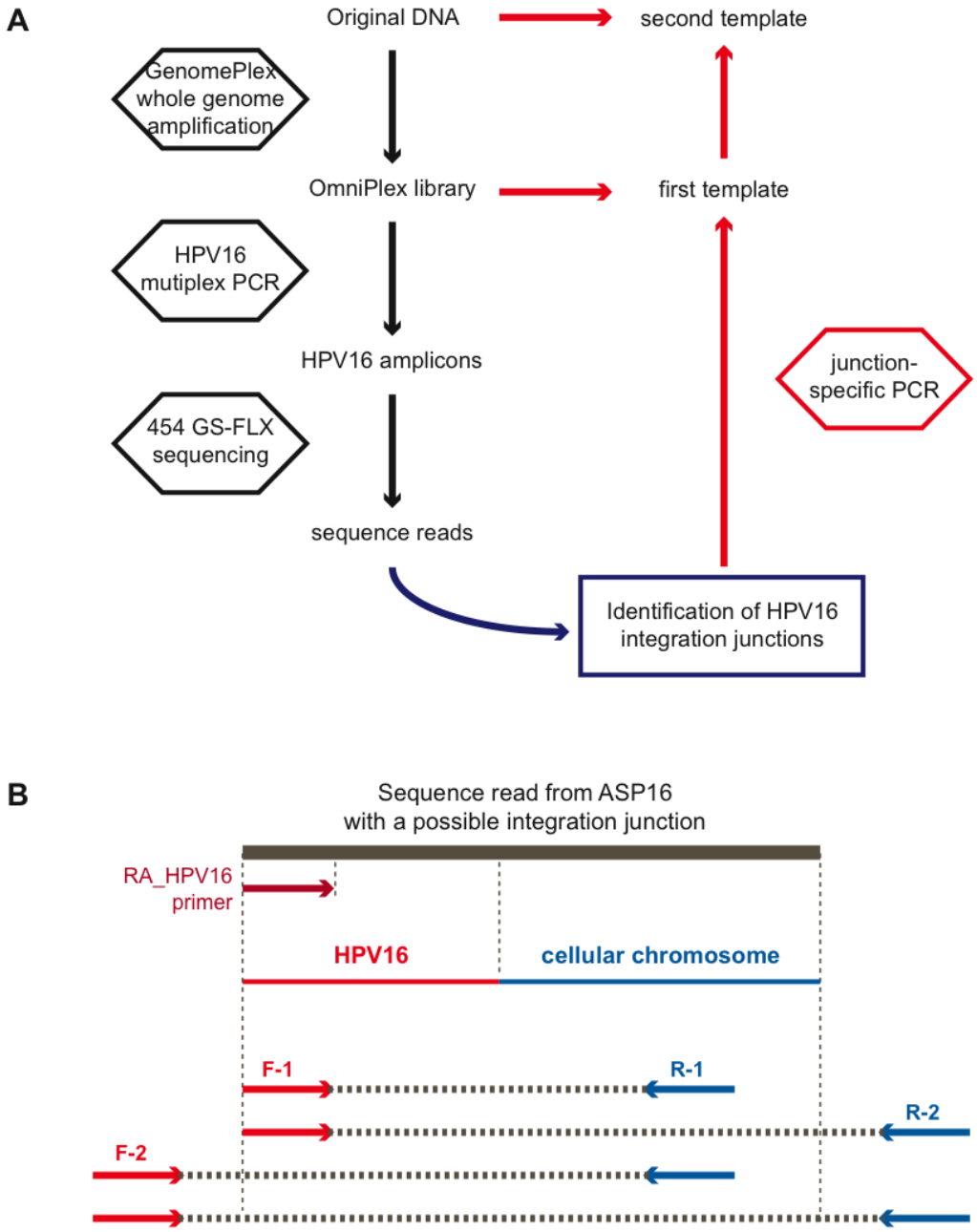


Figure 2.40: Junction-specific PCR for verification of HPV16 integration junctions identified in ASP16 sequence reads. Panel A shows the DNA amplification steps for HPV16 amplicon preparation and junction-specific PCR templates. The original DNA is amplified by GenomePlex whole genome amplification kit, to produce the OmniPlex library. HPV16 amplicons for Roche/454 GS-FLX sequencing were prepared from the OmniPlex libraries of each DNA. For the junction-specific PCR, the OmniPlex library was used first as template. After positive PCR results, the original DNA was used as template for PCR, if enough DNA was available. **Panel B** shows the primer design for junction-specific PCR. The top line represents an ASP16 sequence read containing a potential HPV16 integration junction. Two HPV16-specific forward primers, F-1 and F-2, were selected. F-1 binds to HPV16 inside the ASP16 sequence read, and F-2 binds upstream of the ASP16 read. Two reverse primers, R-1 and R-2, were selected in the flanking cellular sequences. R-1 binds the cellular DNA within the ASP16 read, while R-2 binds outside this area. The primers were combined for PCR in four combinations.

Cell line MRI-H186

MRI-H186 cells contain about 30-40 copies of HPV16 DNA integrated into chromosome 8q24 near the c-myc oncogene (Xu, 2010). Two HPV16 3' junctions have been identified, with the major junction at pos. 2754 and the minor junction at pos. 1224. The major junction is present as two integration variants, A and A+ (Xu, 2010). Variant A is a truncated HPV16 genome, whereas variant A+ includes a complete HPV16 genome (Figure 2.41). An integration percentage of 49% has been determined for this cell line by E2/E6 qPCR.

With the previous knowledge about the HPV16 breakpoint positions, the major and minor integration junctions were expected in the sequences in primer groups E06 (2723) and E19 (1115), respectively. The sequence alignments of primer group E06 in both ASP16-3 and ASP16-4 identified viral-cellular sequences with the major integration junction, together with purely viral sequences (Figure 2.42). The major integration was also detected in sequences of primer E27 (2628), located 95 bp upstream of primer E06. All informative sequences of primer E19, however, did not contain the minor integration junction. The results are summarized in Table 2.19.

Concerning the major 3'junction at pos. 2754, two aspects for junction determination by ASP16 can be noted. First, the distance between primer and breakpoint is short (i.e. 11 nt for sequences in primer group E06), the viral-cellular junction can be determined efficiently in a large number of sequence reads. Second, if the distance between primer and breakpoint is longer (i.e. 104 nt for sequences in primer group E27), a small number of long sequence reads is sufficient to allow an unequivocal determination of a viral-cellular junction. The minor 3'junction at pos. 1223 has been found by PCR experiments as two integration variants, B and A-B (Xu, 2010). In the ASP16 experiments, altogether 118 (24+94) informative sequence reads were obtained from primer E19. However, all reads contained only HPV16 sequences. The reason why ASP16 failed to identify this junction is unclear. However, these results may indicate that variants B and A-B constitute only a minor fraction of the integrated HPV16 genomes in MRI-H186.

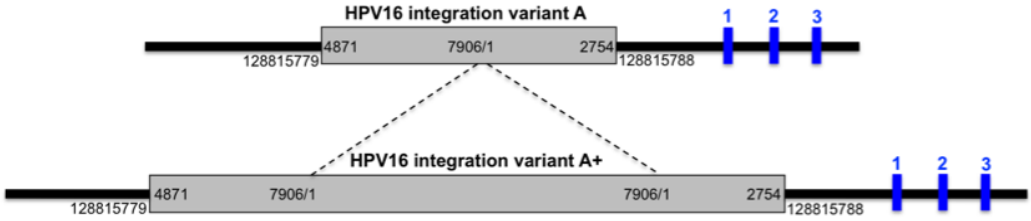


Figure 2.41: Two variants of the major HPV16 integration junctions in MRI-H186. Taken from (Xu, 2010). Variant A and A+ with the major integration junction at HPV16 pos. 2754 are shown. Gray boxes represent HPV16 DNA with positions indicated. Black lines represent cellular DNA. Blue boxes (1, 2, 3) represent the three exons of c-myc gene.

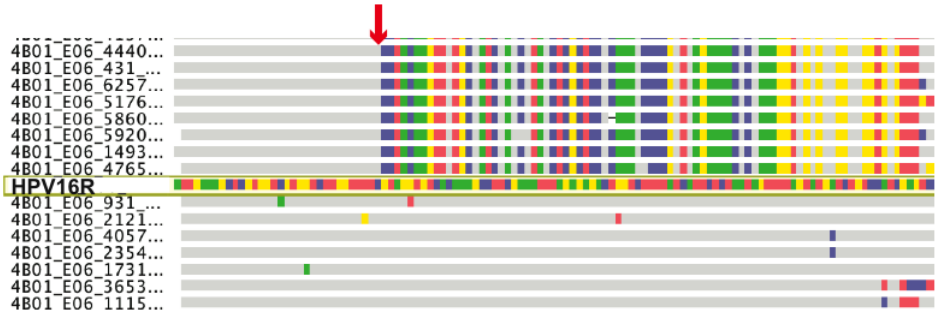


Figure 2.42: Sequence alignment of primer group E06 of MRI-H186 in ASP16-4. The sequences are color-coded. Sequence names are indicated on the left. The reference HPV16R sequence is indicated and all bases are highlighted. For other sequences, bases are highlighted only if they are different from the reference sequence. The sequences above the reference HPV16R contain the integration junction with the HPV16 breakpoint at pos. 2754 (red arrow).

Table 2.19: Features of the major and minor integration junctions of MRI-H186 in ASP16-3 and ASP16-4 sequence reads.

Sample name	MRI-H186						
Integration percentage by E2/E6 qPCR	49%						
Chromosome	8					8	
Cellular breakpoint position (NC_000008.10)	128746606					128675817	
Cellular DNA strand	plus					plus	
HPV16 break point position	(major) pos. 2754					(minor) pos. 1224	
Primer group for junction identification	E06 (2723)			E27 (2628)		E19 (1115)	
Primer positions	2723-2743			2628-2650		1115-1138	
Distance between 3'end of the primer and breakpoint	11 bp			104 bp		86 bp	
Sample ID	2B01*	3B01	4B01	3B01	4B01	3B01	4B01
Junction found?	yes	yes	yes	yes	yes	no	no
# of total reads in this primer group	671	186	485	66	44	283	478
# of reads extending over breakpoint	388	129	444	17	15	24	94
# of reads containing viral-cellular junction	320	112	343	15	14	0	0
# of reads containing HPV16 after breakpoint	68	17	101	2	1	24	94
% of reads containing viral-cellular junction[(a)]	82 %	87 %	77 %	88%	93%	0 %	0 %
Longest sequence read (nt) [(b)]	-	105	130	171	151	-	-

* Experiment ASP16-2 was performed by Bo Xu (Xu, 2010).
(a) The value was calculated from sequence reads extending over the breakpoint. The average value is 85%.
(b) The >=18 nt GPUA sequence and the 4-nt barcode were excluded.

Cell line MRI-H196

The HPV16 integration junction in cell line MRI-H196 was identified by RS-PCR (Xu, 2010). The HPV16 breakpoint is located at pos. 3858, integrated into chromosome 11 at pos. 47967861 (NC_000011.9) on plus strand. This leaves the E2 gene (pos.2756-3853) intact in MRI-H196. By E2/E6 qPCR, a low integration percentage of 32% had been determined. In clinical samples, this value would indicate episomal DNA. The E2 primer pair used in E2/E6 qPCR amplifies the segment from pos. 3362 to pos. 3443, which is present in the integrated HPV16 DNA in this cell line.

The sequence reads of primer E16 (3778) should contain the viral-cellular junction sequences. As shown in Table 2.20, all informative sequence reads from this primer group contained the integration junction sequence. In ASP16-3, the longest cellular part was only 8 nt, too short to be identified by Blast without prior knowledge of the integration junction. In ASP16-4, the longest cellular part was 102 nt, long enough to be mapped to chromosome 11.

Table 2.20: Features of the integration junctions of MRI-H196 in ASP16-3 and ASP16-4 sequence reads

Sample name	MRI-H196	
Integration percentage by E2/E6 qPCR	32%	
Chromosome	11	
Cellular breakpoint position (NC_000011.9)	47967861	
Cellular DNA strand	plus	
HPV16 break point position	pos. 3858	
Primer group for junction identification	E16 (3778)	
Primer positions	3778-3798	
Distance between 3'end of the primer and breakpoint	60 bp	
Sample ID	3B02	4B02
Junction found?	yes	yes
# of total reads in this primer group	324	184
# of reads extending over breakpoint	3	36
# of reads containing viral-cellular junction	3 (only 8 nt cellular)	36
# of reads containing HPV16 after breakpoint	0	0
% of reads containing viral-cellular junction [(a)]	100%	100%
Longest sequence read (nt) [(b)]	88	186

(a) The value was calculated from sequence reads extending over the breakpoint.
(b) The >=18 nt GPUA sequence and the 4-nt barcode were excluded.

Cell line SiHa

SiHa contains one HPV16 copy integrated into chromosome 13 (Mincheva et al, 1987). The HPV16 3' breakpoint is at pos. 3133 (Meissner, 1999). The sequences of primer E29 (3037) should contain the integration junctions. This was also evident from the pattern of sequence read numbers (Figure 2.38). The results were as expected and are summarized in Table 2.21. Since SiHa cells contain only one integrated HPV16 copy, all informative sequence reads contained the viral-cellular junction. Again, a higher number of long reads in ASP16-4 in comparison to ASP16-3 was observed.

Table 2.21: Features of the integration junction of SiHa in ASP16-3 and ASP16-4 sequences of primer E29 (3037).

Sample name	SiHa	
Chromosome	13	
Cellular breakpoint position (NC_000013.10)	74087558	
Cellular DNA strand	minus	
HPV16 break point position	pos. 3133	
Primer group for junction identification	E29 (3037)	
Primer positions	3037-3058	
Distance between 3'end of the primer and breakpoint	75 bp	
Sample ID	3B03	4B03
Junction found?	yes	yes
# of total reads in this primer group	355	228
# of reads extending over breakpoint	3	16
# of reads containing viral-cellular junction	3	16
# of reads containing HPV16 after breakpoint	0	0
% of reads containing viral-cellular junction [(a)]	100%	100%
Longest sequence read (nt) [(b)]	231	113

(a) The value was calculated from sequence reads extending over the breakpoint.
(b) The >=18 nt GPUA sequence and the 4-nt barcode were excluded.

Cell line CaSki

CaSki contains ~600 copies of full-length and rearranged HPV16 DNA (Callahan et al, 1992) integrated as head-to-tail repeats (Baker et al, 1987). More than 10 distinct integration sites have been mapped to different chromosomes (Van Tine et al, 2004). A single transcriptionally active HPV16 integrant has been located on a derivative chromosome 14 in conjunction with chromosome 6 DNA (Van Tine et al, 2004). The viral-cellular junction sequence, found in transcripts and genomic DNA, contains the HPV16 breakpoint at pos. 3728 flanked by chromosome 6 sequences (Jeon & Lambert, 1995; Smits et al, 1991; Van Tine et al, 2004).

Sequence reads of primer E32 (3696) may contain the integration junction. Due to the high number of about 600 HPV16 copies, the chance of finding sequences with the integration junction was estimated to be about 1:600. Among the 773 informative sequences in primer group E32, however, no sequence with integration junction was found. The features for sequence reads of primer E32 are shown in Table 2.22.

Table 2.22: Features of primer E32 sequence reads of CaSki in ASP16-4.

Sample name	CaSki
Chromosome	6
Cellular breakpoint position (NC_000006.11)	45659121
Cellular DNA strand	minus
HPV16 break point position	pos. 3728
Primer group for junction identification	E32 (3696)
Primer positions	3696-3717
Distance between 3'end of the primer and breakpoint	11 bp
Sample ID	4B04
Junction found?	no
# of total reads in this primer group	793
# of reads extending over breakpoint	773
# of reads containing viral-cellular junction	0
# of reads containing HPV16 after breakpoint	3728
% of reads containing viral-cellular junction [(a)]	0%
Longest sequence read (nt) [(b)]	172

(a) The value was calculated from sequence reads extending over the breakpoint.
(b) The >=18 nt GPUA sequence and the 4-nt barcode were excluded.

CA-07C381

The HPV16 integration junction in this DNA sample had previously been determined in ASP16-2 (Xu, 2010). This sample was re-analyzed in ASP16-4 (ID: 4B05). CA-07C381 contains exclusively integrated DNA (Xu, 2010), see also Table 2.23. In agreement with the known HPV16 breakpoint at pos. 2783, the pattern of sequence read numbers clearly indicated that the breakpoint is situated downstream of primer E06 (2723) (Figure 2.43). All 26 informative E06 sequence reads contained the viral-cellular junction sequence (Table 2.23), consistent with the fully integrated status of HPV16 DNA in this tumor. Comparing ASP16-2 and ASP16-4, the number of informative sequence reads has been significantly increased from 2 to 26.

Table 2.23: Features of primer E06 sequence reads of CA-07C381 in ASP16-4 in comparison with those of ASP16-2

Sample name	CA-07C381	
Integration percentage by E2/E6 qPCR	100%	
Chromosome	12	
Cellular breakpoint position (NC_000012.11)	57686270	
Cellular DNA strand	plus	
HPV16 break point position	pos.2783	
Primer group for junction identification	E06 (2723)	
Primer positions	2723-2742	
Distance between 3'end of the primer and breakpoint	41 bp	
Sample ID	4B05	2B06*
Junction found?	yes	yes
# of total reads in this primer group	42	
# of reads extending over breakpoint	26	2
# of reads containing viral-cellular junction	26	2
# of reads containing HPV16 after breakpoint	0	0
% of reads containing viral-cellular junction [(a)]	100%	100%
Longest sequence read (nt) [(b)]	198	171

* Experiment ASP16-2 was performed by Bo Xu (Xu, 2010).
(a) The value was calculated from sequence reads extending over the breakpoint.
(b) The >=18 nt GPUA sequence and the 4-nt barcode were excluded.

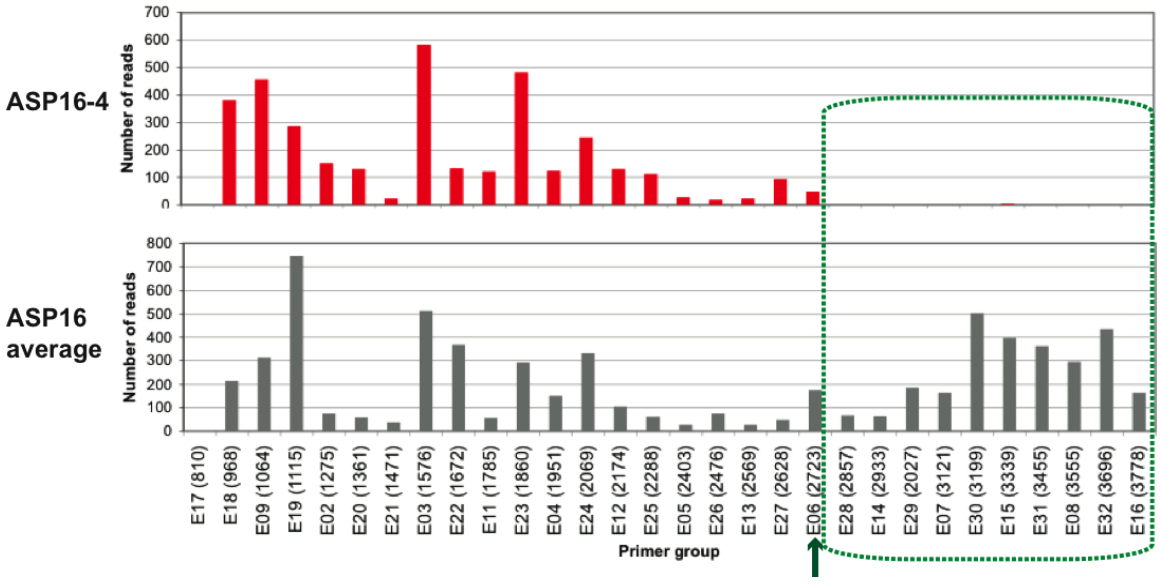


Figure 2.43: Patterns of sequence read numbers in ASP16-4 for CA-07C381. The graph (red) represents the number of reads in ASP16-4 after 28-nt cutoff for CA-07C381. The graph at the bottom (gray) represents the average number of reads in ASP16 after 28-nt cutoff. The data for these plots are shown in Table 2.17. X-axis represents the primer group (with primer name indicated) and Y-axis the number of reads. The possible breakpoint area is circled in green. The green arrow indicates the primer group E06 that contain the HPV16 integration junction sequences with the breakpoint at pos. 2783 (see Table 2.23).

CA-07C368

In ASP16-2, the HPV16 integration junction of sample CA-07C368 was identified in a curious manner (Xu, 2010). Three sequence reads of primer E03 (1576) contained a viral-cellular junction which could not be verified by junction-specific PCR and thus was an artifact. One of the PCR products hybridized strongly with HPV16. Cloning and sequencing revealed a novel viral-cellular junction sequence, which proved to be authentic according to junction-specific PCR. The authentic HPV16 breakpoint is located at pos. 1910, and the HPV16 was found to have integrated into chromosome 1 at pos. 240760698 (NC_000001.10) on plus strand.

CA-07C368 was re-analyzed in ASP16-4 (ID: 4B21). The authentic HPV16 junction was detected among the sequence reads in primer groups E23 (1860) and E11 (1785). The features of these sequence reads are summarized in Table 2.24. The artifact junction sequence of ASP16-2 was not found in the E03 sequence reads of 4B21.

Table 2.24: Features of sequence reads of CA-07C368 in ASP16-4.

Sample name	CA-07C368	
Integration percentage by E2/E6 qPCR	84%	
Chromosome	1	
Cellular breakpoint position (NC_000001.10)	240760698	
Cellular DNA strand	plus	
HPV16 break point position	pos.1910	
Primer group for junction identification	E23 (1860)	E11 (1785)
Primer positions	1860-1879	1785-1805
Distance between 3'end of the primer and breakpoint	32 bp	106 bp
Sample ID	4B21	
Junction found?	yes	yes
# of total reads in this primer group	265	72
# of reads extending over breakpoint	37	1
# of reads containing viral-cellular junction	36	1
# of reads containing HPV16 after breakpoint	1	0
% of reads containing viral-cellular junction [(a)]	97%	100%
Longest sequence read (nt) [(b)]	175	167

(a) The value was calculated from sequence reads extending over the breakpoint.
(b) The >=18 nt GPUA sequence and the 4-nt barcode were excluded.

HSIL-66019 and HSIL-61979

Both HSIL-66019 and HSIL-61979 were isolated from CIN3 lesions and have integration percentage of 100%. In previous experiments, the HPV16 breakpoint of HSIL-66019 was localized between pos. 2451 and 2722 by PCR screening (Steinmeyer, 2009). Because all efforts failed to identify the HPV16 junction by PCR-cloning-sequencing methods (Steinmeyer, 2009), the DNA of HSIL-66019 was analyzed in ASP16-3 (ID: 3B18) and ASP16-4 (ID: 4B06). The possible HPV16 integration junction was detected in sequence reads of 3B18 from primer group E26 (2476) (Figure 2.44, panel A). The junction contained the HPV16 breakpoint at pos. 2516, integrated into chromosome 7 (NC_000007.13) at three possible locations 72454568 (plus strand), 72818793 (plus strand) and 75010338 (minus strand). It was not possible to identify by junction-specific PCR which location is the correct one because these three locations on chromosome 7 contain almost 100% sequence similarity, each location covering a large area of 20-200 kb (Steinmeyer, 2009). The integration junction was confirmed by junction-specific PCR using the original DNA HSIL-66019 as template, and cloning-sequencing of the PCR products (Steinmeyer, 2009). In 4B06, the junction was also detected in sequence reads of primer E26.

For HSIL-61979 (ID: 4B07), an HPV16 integration junction was also detected in primer group E26. This junction sequence was identical to that of HSIL-66019. The original DNA HSIL-61979 was used as template for junction-specific PCR, with the two primer pairs previously used for DNA HSIL-66019 (Steinmeyer, 2009) (Figure 2.44, panel B). Positive products were obtained, cloned and sequenced. The sequences (shown in Appendix A4) were identical to the junction-specific PCR products of DNA HSIL-66019.

It is very unlikely that two unrelated DNA samples have exactly the same HPV16 breakpoint and chromosomal integration site. Review of the sample numbers revealed that both DNA samples were collected from the same woman at a six-month interval, in April 2007 (HSIL-61979) and October 2007 (HSIL-66019). The patterns of sequence read numbers for HSIL-66019 and HSIL-61979 are shown in Figure 2.45. All primer groups from E13 (2569) down to E16 (3778) lacked sequence reads. The primer group E26 (2476), located just upstream of this blank region, contained the viral-cellular junction sequences.

These results clearly demonstrates that the ASP16 strategy is suited both for an appropriate localization of HPV16 integration breakpoints by inspection of sequence read patterns and for exact nucleotide sequence identification of integration junctions. The ASP16 features for the integration junction of samples HSIL-66019 and HSIL-61979 are summarized in Table 2.25.

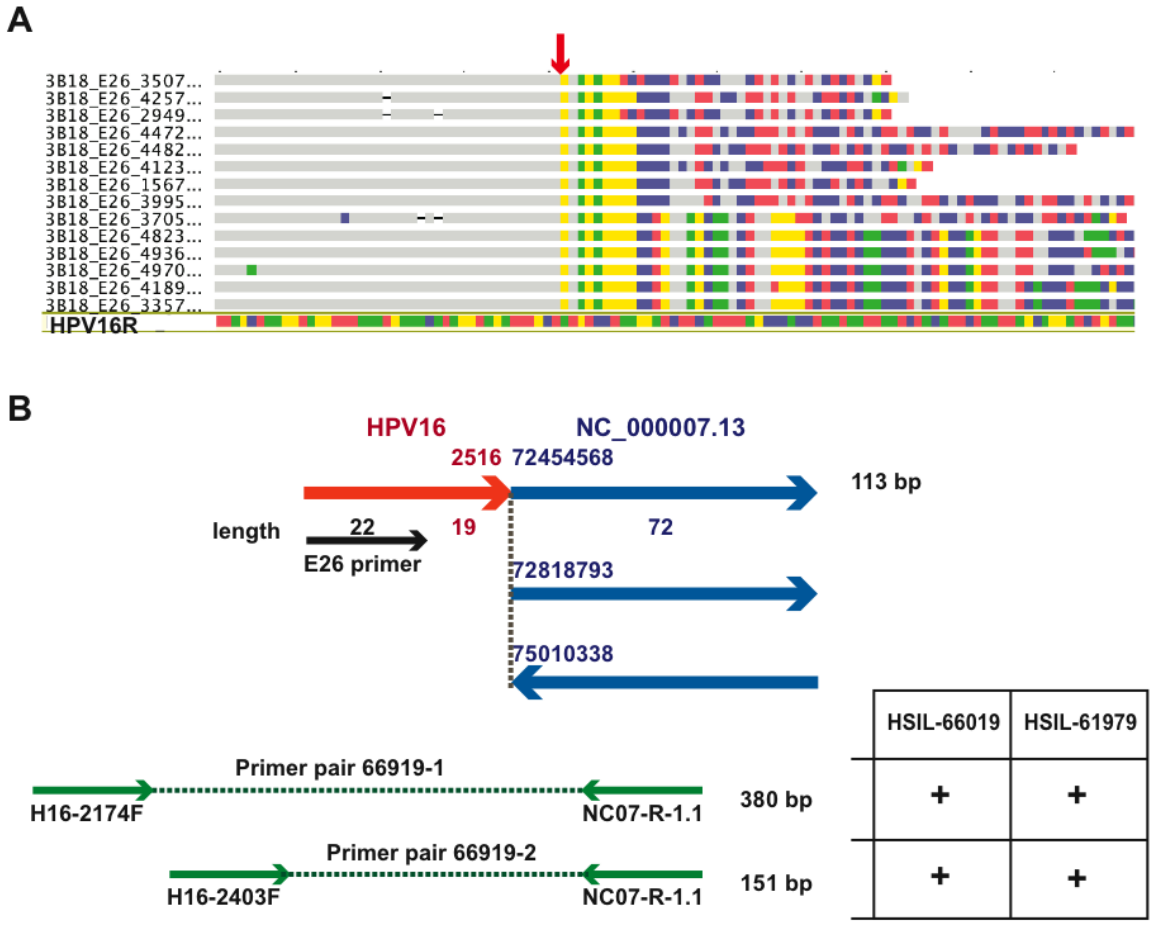


Figure 2.44: HPV16 integration junction in E26 sequences of HSIL-66019 and junction-specific PCR. **Panel A** shows the sequence alignment of E26 sequences of HSIL-66019 (3B18). The sequences are color-coded. Sequence names are indicated on the left. The reference HPV16R sequence is shown at the bottom and all bases are highlighted. For other sequences, bases are highlighted only if they are different from the reference sequence. All 14 reads above the reference HPV16R are composed of HPV16 and cellular sequences, with the HPV16 breakpoint at pos. 2516 (red arrow). **Panel B** shows the junction specific PCR of viral-cellular sequence detected in HSIL-66019 (3B18 and 4B06) and HSIL-61979 (4B07). The composition of the HPV16-cellular sequences is shown on top, with the three possible chromosomal integration sites indicated (blue arrows). The two primer pairs (green arrows) and the expected PCR products are shown below. The PCR results of the original DNAs of both samples are indicated on the right side. The junction-specific PCR for HSIL-66019 had also been performed previously (Steinmeyer, 2009). Primer sequences are given in Materials and Methods.

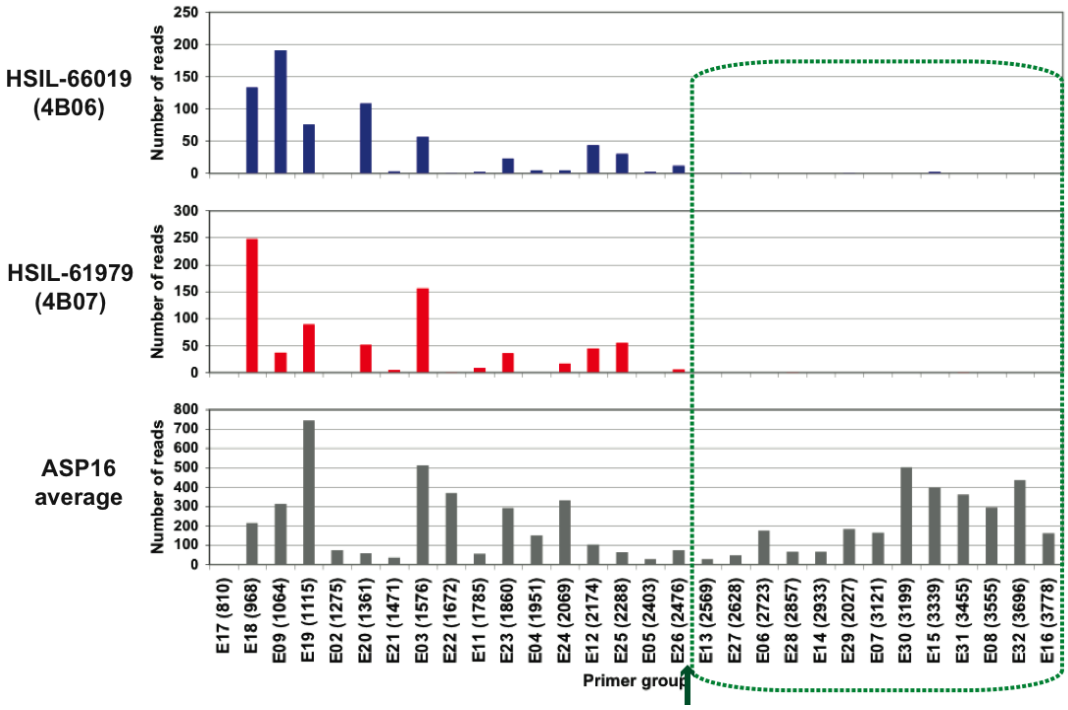


Figure 2.45: Patterns of sequence read numbers in ASP16-4 for HSIL-66019 and HSIL-61979. The graphs represent the number of reads in ASP16-4 after 28-nt cutoff for HSIL-66019 (blue) and HSIL-61979 (red), respectively. The graph at the bottom (gray) represents the average number of reads in ASP16 after 28-nt cutoff. The data for these plots are shown in Table 2.17. X-axis represents the primer group (with primer name indicated) and Y-axis the number of reads. The possible breakpoint area is circled in green. The green arrow indicates primer E26, whose sequences contain the HPV16 integration junction sequences (see Table 2.25).

Table 2.25: Features of the integration junctions of samples HSIL-66019 and HSIL-61979 in ASP16.

Sample name	HSIL-66019		HSIL-61979
Integration percentage by E2/E6 qPCR	100%		100%
Chromosome	7		
Cellular breakpoint position (NC_000007.13)	72454568 or 72818793 or 75010338		
Cellular DNA strand	plus or plus or minus		
HPV16 break point position	pos. 2516		
Primer group for junction identification	E26 (2476)		
Primer positions	2476-2497		
Distance between 3'end of the primer and breakpoint	19 bp		
Sample ID	3B18	4B06	4B07
Junction found?	yes	yes	yes
# of total reads in this primer group	14	11	6
# of reads extending over breakpoint	14	11	6
# of reads containing viral-cellular junction	14	4	6
# of reads containing HPV16 after breakpoint	0	(7) [c]	0
% of reads containing viral-cellular junction [a]	100%	(36%) [c]	100%
Longest sequence read (nt) [b]	113	111	117

(a) The value was calculated from sequence reads extending over the breakpoint.
(b) The >=18 nt GPUA sequence and the 4-nt barcode were excluded.
(c) Probably due to barcode error.

HSIL-75857

The pattern of sequence read numbers of sample HSIL-75857 (ID: 4B09) (Table 2.17), gave no hint at a possible HPV16 breakpoint location. Therefore, the alignments of all primer groups were studied. A candidate integration junction was detected in the sequence reads of primer E19 (1115) (Figure 2.46, panel A), with HPV16 breakpoint at pos. 1149. The suspected junction was verified by junction-specific PCR with two primer pairs, using the original DNA HSIL-75857 as template (Figure 2.46, panel B). Both PCR reactions were positive, and the products were cloned and sequenced. The sequences (shown in Appendix A4) confirmed the detected HPV16 integration. The features of the sequences of primer E19 of HSIL-75857 are shown in Table 2.26. Although the E2/E6 qPCR value of 100% integration indicated exclusively integrated HPV16 DNA, only 3% of the E16 sequence reads contained the viral-cellular junction, whereas 97% contained purely viral sequences. The reason for this discrepancy is unknown.

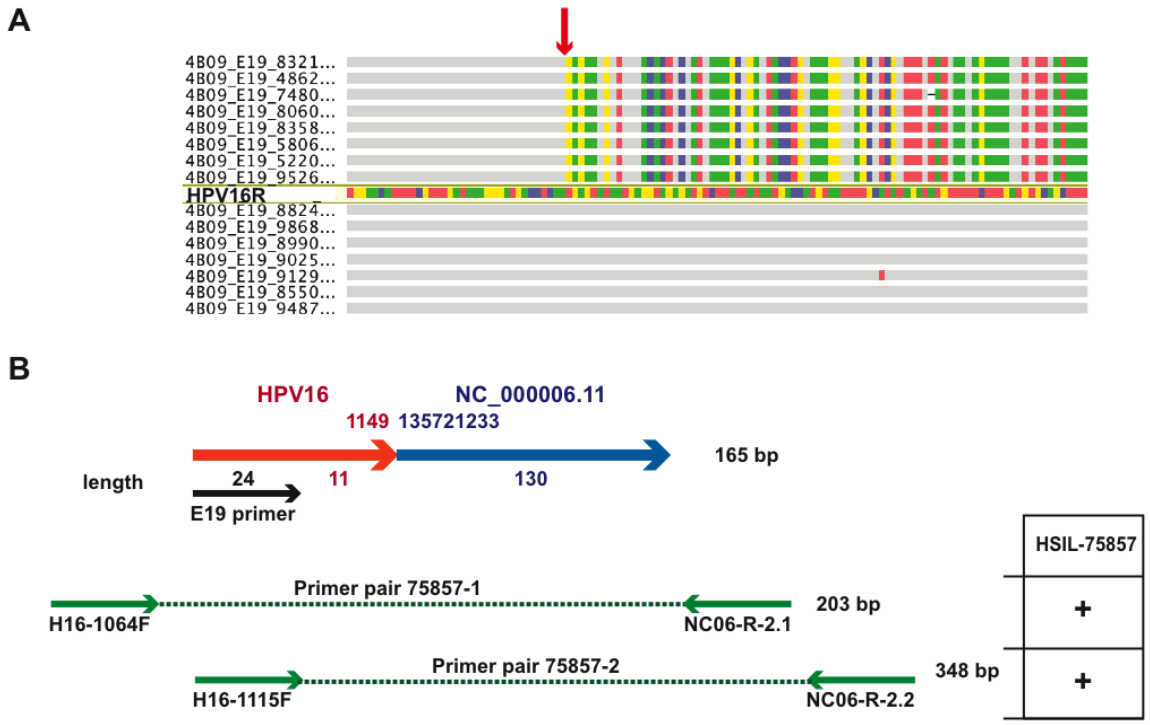


Figure 2.46: HPV16 integration junction in E19 sequences of HSIL-75857 and junction-specific PCR. **Panel A** shows the sequence alignment of E19 sequences of HSIL-75857 (4B09). The sequences are color-coded. Sequence names are indicated on the left. All bases of the reference HPV16R sequence are highlighted. In other sequences, bases are highlighted only if they are different from the reference sequence. All 8 reads above the reference HPV16R contain viral-cellular sequences, with the HPV16 breakpoint at pos. 1149 (red arrow). **Panel B** shows the junction specific PCR of viral-cellular sequence detected in 4B09. The composition of the HPV16-cellular sequences is shown on top. The two primer pairs (green arrows) and the expected PCR products are shown below. The PCR results of the original DNA HSIL-75857 as template are indicated on the right side. Primer sequences are given in Materials and Methods.

Table 2.26: Features of E19 sequence reads of DNA HSIL-75857 in ASP16-4.

Sample name	HSIL-75857
Integration percentage by E2/E6 qPCR	100%
Chromosome	6
Cellular breakpoint position (NC_000006.11)	135721233
Cellular DNA strand	plus
HPV16 break point position	pos. 1149
Primer group for junction identification	E19 (1115)
Primer positions	1115-1138
Distance between 3'end of the primer and breakpoint	11 bp
Sample ID	4B09
Junction found?	yes
# of total reads in this primer group	1453
# of reads extending over breakpoint	1420
# of reads containing viral-cellular junction	41
# of reads containing HPV16 after breakpoint	1379
% of reads containing viral-cellular junction [(a)]	3%
Longest sequence read (nt) [(b)]	165

(a) The value was calculated from sequence reads extending over the breakpoint.
(b) The >=18 nt GPUA sequence and the 4-nt barcode were excluded.

CIN2/3-1801

Sample CIN2/3-1801 was analyzed in both ASP16-3 (ID: 3B09) and ASP16-4 (ID: 4B14). The patterns of the sequence read numbers of each primer group in both ASP16-3 and ASP16-4 (Table 2.16 and Table 2.17) gave no hint at any HPV16 breakpoint location. Thus, the sequence alignments of all primer groups were analyzed.

In 3B09, five viral-cellular junction sequences were detected among 11 informative sequence reads of E23 (1860) with HPV16 breakpoint at pos. 1913 (Figure 2.47, panel A). In 4B14, the integration junction sequence was also detected, but only in one sequence read and with only 19 nt of flanking cellular sequence. The features of the detected integration junction in DNA CIN2/3-1801 are summarized in Table 2.27. The possible HPV16 integration junction was examined by junction-specific PCR with two primer pairs using the OminiPlex library and the original DNA as templates. Both templates resulted in positive PCR products for both of the primer pairs (Figure 2.47, panel B). The products from the original DNA template were cloned and sequenced. The sequences (shown in Appendix A4) confirmed the ASP16 sequence data.

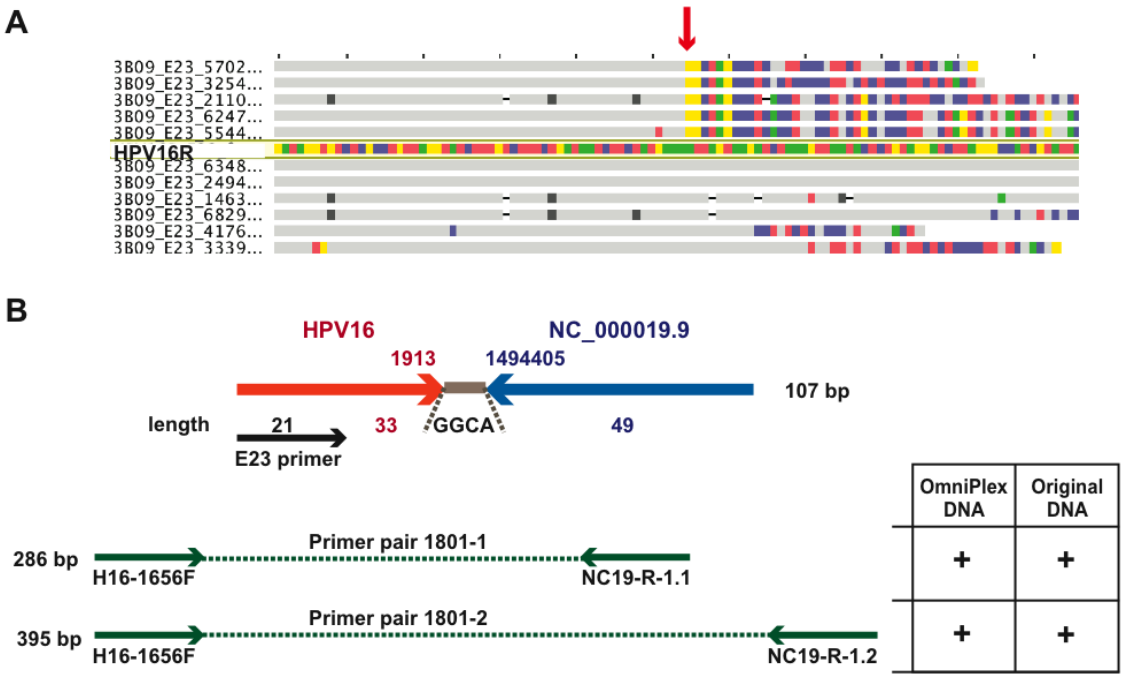


Figure 2.47: HPV16 integration junction in E23 sequences of CIN2/3-1801 from ASP16-3 and junction-specific PCR. Panel A shows the sequence alignment of E23 (1860) sequences of CIN2/3-1801 (3B09). The sequences are color-coded. Sequence names are indicated on the left. All bases of the reference HPV16R sequence are highlighted. In other sequences, bases are highlighted only if they are different from the reference sequence. All reads above the reference HPV16R contain viral-cellular sequences, with the HPV16 breakpoint at pos. 1913 (red arrow). **Panel B** shows the junction specific PCR of viral-cellular sequence detected in 3B09. The composition of the HPV16-cellular sequences is shown on top. The two primer pairs (green arrows) and the expected PCR products are shown below. The PCR results of the OmniPlex library and original DNA of CIN2/3-1801 as templates are indicated on the right side. Primer sequences are given in Materials and Methods.

Table 2.27: Features of the integration junction detected in E23 sequences of CIN2/3-1801 in ASP16.

Sample name	CIN2/3-1801	
Integration percentage by E2/E6 qPCR	32%	
Chromosome	19	
Cellular breakpoint position (NC_000019.9)	1494405	
Cellular DNA strand	minus	
HPV16 break point position	pos.1913	
Primer group for junction identification	E23 (1860)	
Primer positions	1860-1880	
Distance between 3'end of the primer and breakpoint	33 bp	
Sample ID	3B09	4B14
Junction found?	yes	yes
# of total reads in this primer group	252	176
# of reads extending over breakpoint	11	9
# of reads containing viral-cellular junction	5	1 (19 nt celular)
# of reads containing HPV16 after breakpoint	6	8
% of reads containing viral-cellular junction [(a)]	45%	11%
Longest sequence read (nt) [(b)]	107	77

(a) The value was calculated from sequence reads extending over the breakpoint.
(b) The >=18 nt GPUA sequence and the 4-nt barcode were excluded.

CIN2/3-4242

The clinical DNA CIN2/3-4242 was derived from a high-grade lesion (integration percentage 79%). The DNA was analyzed in both ASP16-3 (ID: 3B16) and ASP16-4 (ID: 4B20). The pattern of sequence read numbers of 3B16 gave no hints at a possible HPV16 breakpoint. Therefore, the sequence alignments of all primer groups for 3B16 were studied. A possible HPV16 integration junction was detected in one sequence read of primer E15 (3339).

Junction-specific PCRs with four primer pairs were performed using the OmniPlex library and the original DNA as templates (Figure 2.48). From the OmniPlex template, a PCR product was obtained only with primer pair 4242-1, while the original DNA was negative for all primer pairs. The E15 sequence reads of 4B20 did not contain this detected integration junction. The features of the E15 sequence reads of both 3B16 and 3B20 are summarized in Table 2.28. Altogether, the results indicate that the detected viral-cellular sequence was an artifact, and that an HPV16 integration junction for DNA CIN2/3-4242 could not be identified.

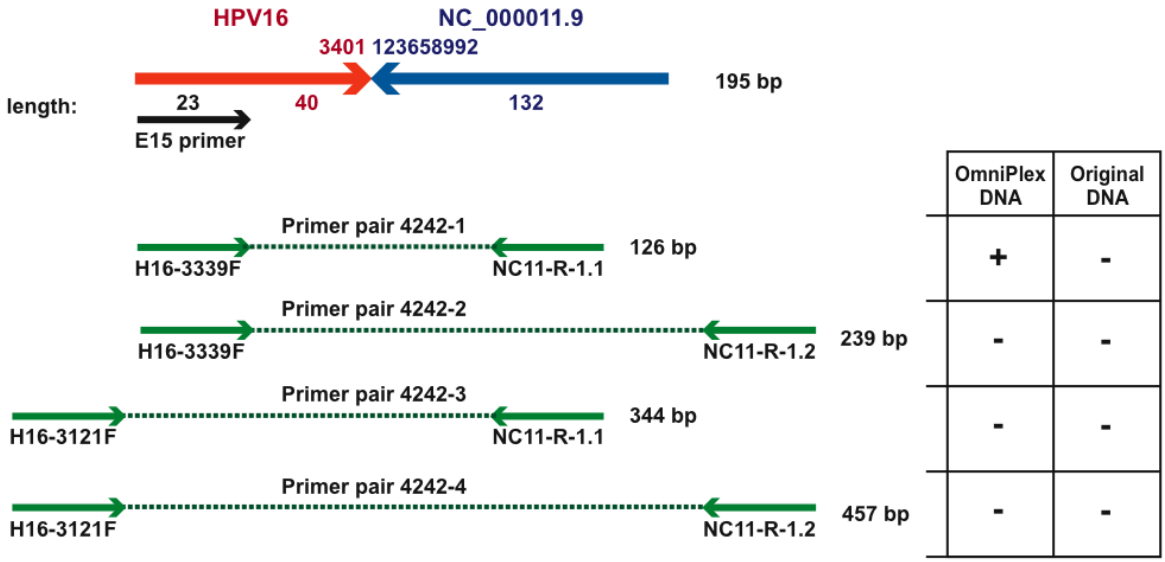


Figure 2.48: Junction-specific PCR of a viral-cellular sequence detected in 3B16. The composition of the HPV16-cellular sequences is shown on top. The four primer pairs (green arrows) and their expected PCR products are shown below. The PCR results of the OmniPlex library and original DNA of CIN2/3-4242 as templates are indicated on the right side. Primer sequences are given in Materials and Methods.

Table 2.28: Features of E15 sequence reads for sample CIN2/3-4242 in ASP16.

Sample name	CIN2/3-4242	
Integration percentage by E2/E6 qPCR	79%	
Chromosome	11	
Cellular breakpoint position (NC_000011.9)	123658992	
Cellular DNA strand	minus	
HPV16 break point position	pos. 3401	
Primer group for junction identification	E15 (3339)	
Primer positions	3339-3361	
Distance between 3'end of the primer and breakpoint	41 bp	
Sample ID	3B09	4B20
Junction found?	false-positive	no
# of total reads in this primer group	197	153
# of reads extending over breakpoint	112	124
# of reads containing viral-cellular junction	(1) [(a)]	0
# of reads containing HPV16 after breakpoint	111	124
% of reads containing viral-cellular junction		0%
Longest sequence read (nt) [(b)]	219	141

(a) False-positive, see text.
(b) The >=18 nt GPUA sequence and the 4-nt barcode were excluded.

CIN2/3-1503

DNA CIN2/3-1503 was analyzed in ASP16-3 (ID: 3B07) and ASP16-4 (ID: 4B12). The patterns of sequence read numbers gave no hint at a possible HPV16 breakpoint location. Therefore, the sequence alignments of all primer groups were analyzed.

In 3B07, a viral-cellular junction sequence was detected in two sequence reads of primer E19 (1115), with HPV16 breakpoint at pos. 1245 (Figure 2.49, panel A). To examine the authenticity of this junction, junction-specific PCRs with two primer pairs were performed (Figure 2.49, panel B). The OmniPlex library and the original DNA were used as templates. The OmniPlex template gave a positive product only with primer pair 1503-1, while the original DNA CIN2/3-1503 was negative for both primer pairs. The results indicated that the detected viral-cellular sequence is an artifact and not a genuine HPV16 integration junction.

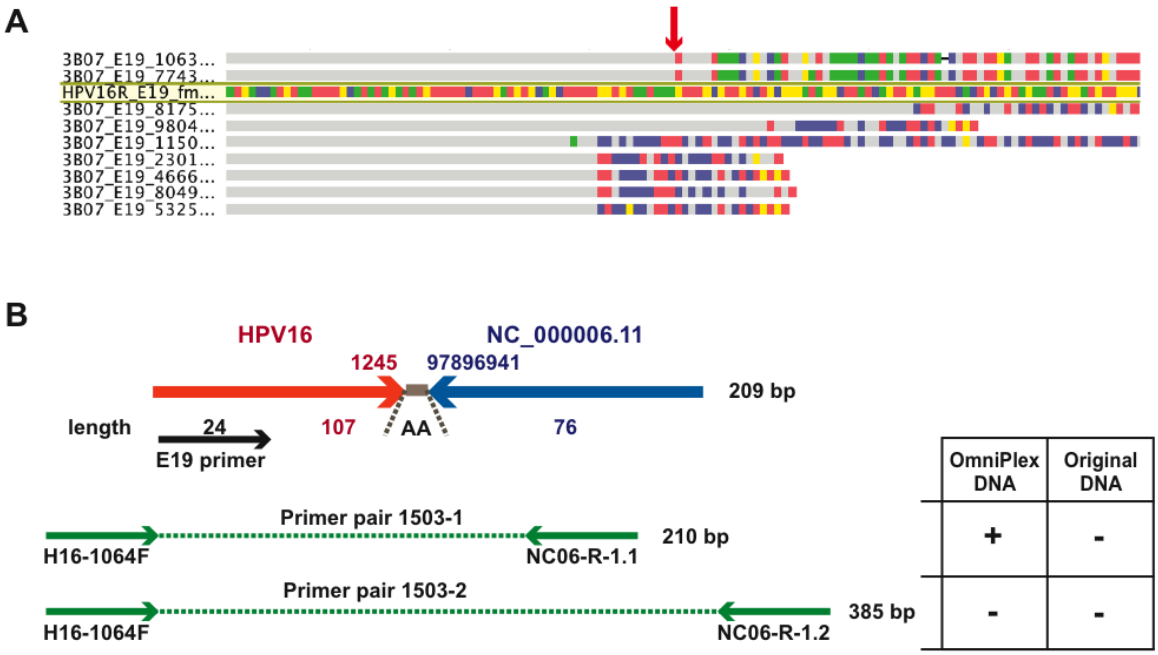


Figure 2.49: Analysis of viral-cellular junction sequences detected in DNA of CIN2/3-1503. Panel A shows the sequence alignment of E19 (1115) sequences of sample 3B07. The sequences are color-coded. Sequence names are indicated on the left. All bases of the reference HPV16R sequence are highlighted. In other sequences, bases are highlighted only if they are different from the reference sequence. The two reads above the reference HPV16R contain viral-cellular sequences, with the HPV16 breakpoint at pos. 1245 (red arrow). **Panel B** shows the junction-specific PCR of viral-cellular sequence detected in 3B07. The composition of the HPV16-cellular sequences is shown on top. The two primer pairs (green arrows) and the expected PCR products are shown below. The PCR results of the OmniPlex library and original DNA of CIN2/3-1503 as templates are indicated on the right side. Primer sequences are given in Materials and Methods.

LSIL- 75022

The DNA LSIL-75022 was derived from a low-grade lesion. Despite its low-grade status, the integration percentage of 100% by E2/E6 qPCR indicates that it should contain fully integrated HPV16 DNA. The viral load was very low (18 copies E7 per 50 ng DNA) compared to other samples (e.g. CA-07C368 with 71923 copies E7 per 50 ng DNA). Consequently, the total number of sequence reads of this DNA obtained in ASP16-4 was very low (see Table 2.17). Based on the pattern of sequence read numbers, possible HPV16 breakpoint positions were localized downstream of primer E23 (1860) and E31 (3455) (Figure 2.50). The sequence alignments of primer groups E23 and E31-E16 were studied. In primer group E23, the alignment shows a number of sequence reads with long CA-rich non-HPV16 sequence (Figure 2.51, panel A). Because the GPUA reverse sequence consists exclusively of C and A residues, it was not possible to figure out whether these sequences are of GPUA or cellular origin. In primer group E31, two out of 28 sequence reads contained a possible viral-cellular sequence of 23 nt (Figure 2.51, panel

B). Blasting these sequences to the human database was unsuccessful in mapping the sequences to a specific chromosome. The sequences of the remaining primer groups were also inspected, but no viral-cellular sequence could be detected. In summary, the breakpoint of the supposed integrated HPV16 DNA in LSIL-75022 could not be identified in experiment ASP16-4.

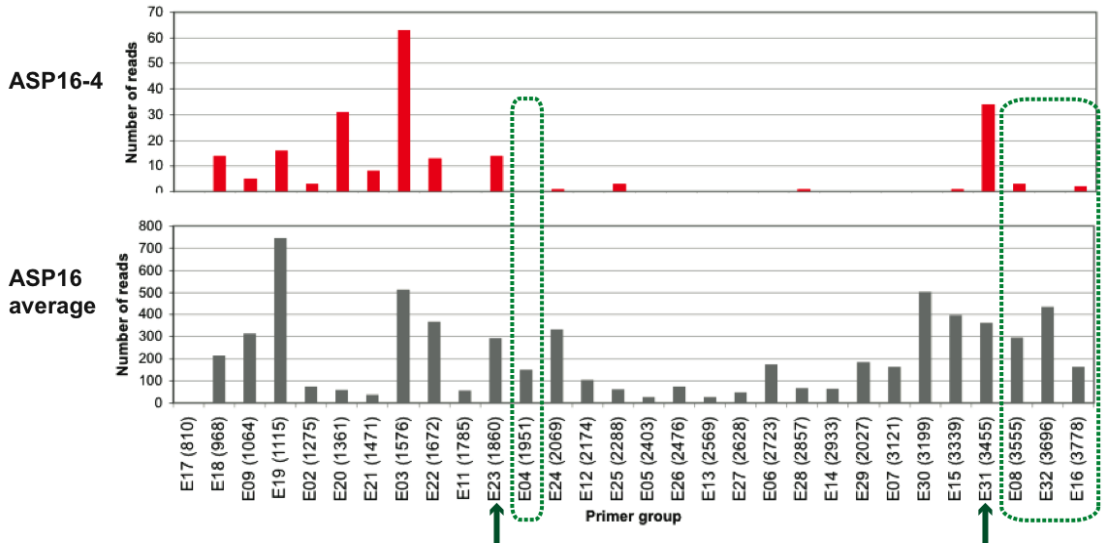


Figure 2.50: Prediction of HPV16 breakpoint locations in DNA LSIL-75022. The graph at the top (red) represents the number of reads for each primer group after 28-nt cutoff of DNA LSIL-75022 (4B08). The graph at the bottom (gray) represents the average number of reads in ASP16. The data for these plots are shown in Table 2.17. X-axis represents the primer group (with primer name indicated) and Y-axis the number of read. The possible breakpoint areas are circled in green. The green arrows indicate primer E23 and E31, whose sequences may contain the HPV16 integration junction sequences.

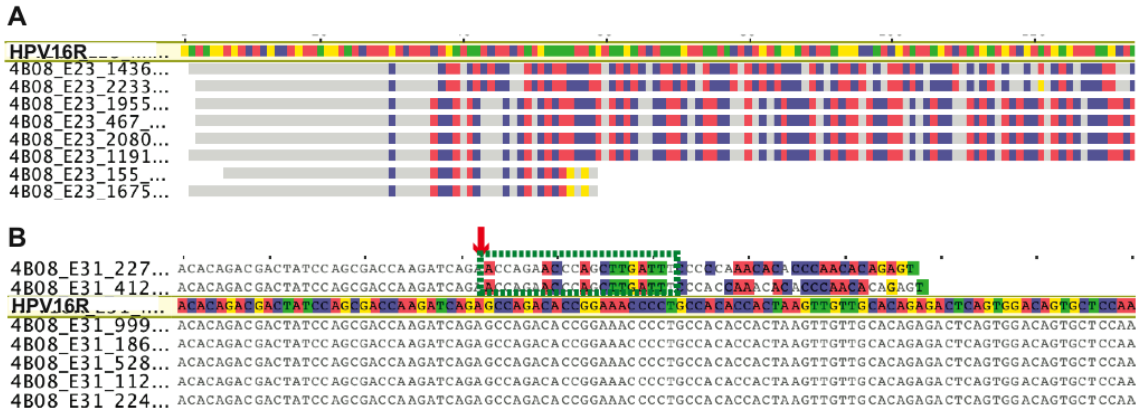


Figure 2.51: Sequence alignment of primer groups E23 and E31 of DNA LSIL-75022. Sequence names are indicated on the left. The sequences are color-coded. All bases of the reference HPV16R sequence are highlighted. In other sequences, bases are highlighted only if they are different from the reference sequence. In **panel A**, all E23 sequences below the reference HPV16R contain long CA-rich non-HPV16 DNA (A and C in red and blue colors). In **panel B**, the sequences above HPV16R contain a possible integration junction with short non-HPV16 sequence (circled in green) with a breakpoint at pos. 3586 (red arrow).

CIN2/3-2219

The DNA CIN2/3-2219 was derived from a high-grade lesion (integration percentage 72%). The patterns of sequence read numbers revealed possible HPV16 breakpoint locations downstream of primers E06 (2723) and E08 (3555) (Figure 2.52). The alignments of sequences from these two primer groups were studied. None of the sequences in these two primer groups was long enough to reach over the next downstream primers (E28 and E32, respectively). No viral-cellular sequence was identified among these sequences. Additionally, the alignments of all other primer groups were also inspected, but no possible integration junction was found.

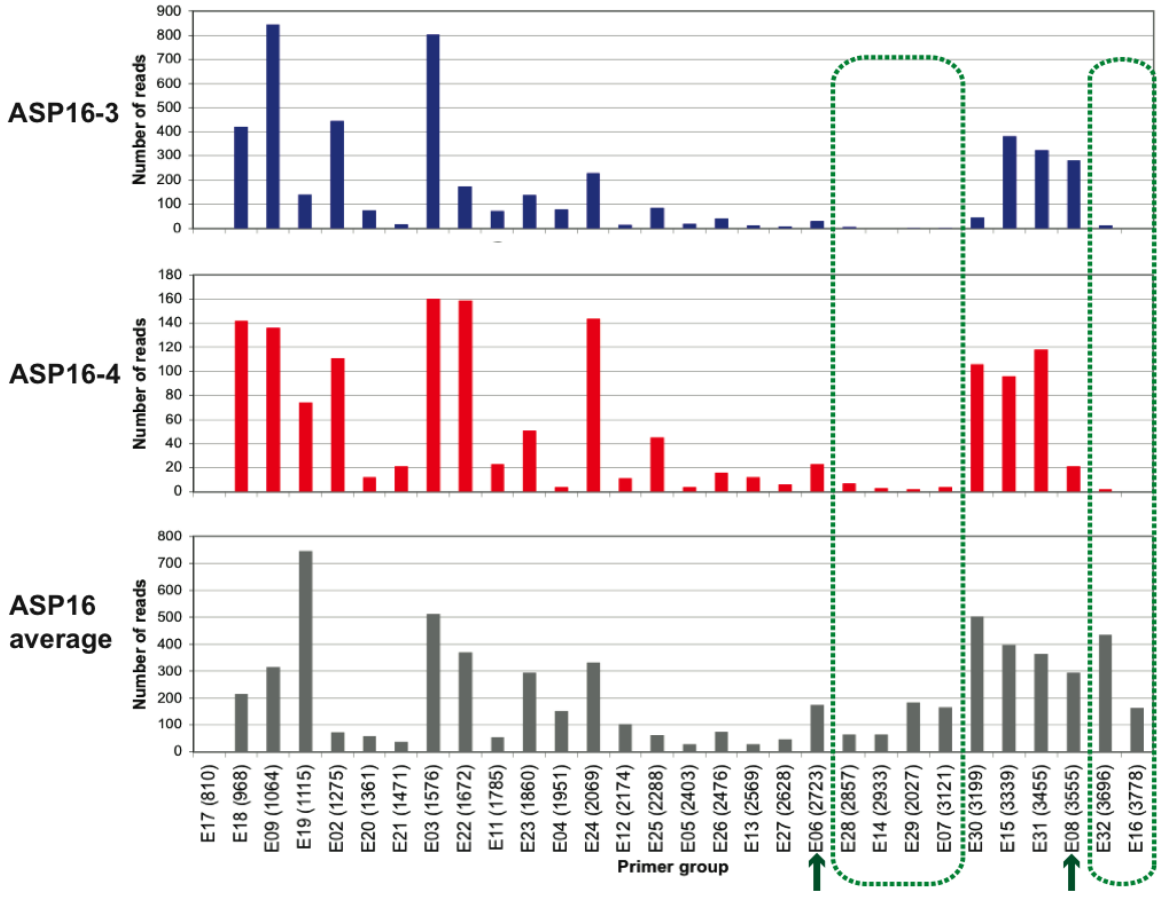


Figure 2.52: Patterns of sequence read numbers for DNA CIN2/3-2219. The graphs represent the number of reads in ASP16-3 (blue) and ASP16-4 (red) after 28-nt cutoff. The graph at the bottom (gray) represents the average number of reads in ASP16-4. The data for these plots are shown in Table 2.16 and Table 2.17. X-axis represents the primer group and Y-axis the number of read. The possible breakpoint areas are circled in green. The green arrows indicate the primer groups (E06 and E08) that may contain sequences with an integration junction.

Clinical DNA samples with no HPV16 breakpoint identified by ASP16

Concerning the remaining eleven DNA samples (Table 2.18), the patterns of sequence read numbers (Table 2.16 and Table 2.17) did not reveal any hint for HPV16 integration breakpoint locations. In addition, after examination of the sequence alignments of all primer groups for these samples, no sequence reads with possible viral-cellular sequences were found. Therefore, no HPV16 integration junctions for these samples could be identified by ASP16.

2.2.2.7 Locations of HPV16 integration junctions

Integration of hr-HPV is an important step toward the development of invasive cancer (Hopman et al, 2004; Schneider-Gadicke & Schwarz, 1986). It usually results in stable expression of the viral oncogenes E6/E7 (Jeon & Lambert, 1995), which is necessary to cause cervical carcinomas (zur Hausen, 1999). Furthermore, it is hypothesized that the integrated HPV DNA can also alter the expression of cellular genes through insertional mutagenesis, such as activation of proto-oncogenes or inactivation of tumor suppressor genes. Therefore, the knowledge of the integration junction locations as well as the cellular sequences directly affected by HPV integration or located in the proximity to the junctions is important to gain more insights into possible insertional mutagenesis effects. For this purpose, the locations of the HPV16 integration junctions and the cellular genes in the vicinity were reviewed by data mining in the NCBI databank. The integration junctions of four clinical DNA samples were newly identified in ASP16-3 and ASP16-4: HSIL-66019, HSIL-61979, HSIL-75857, and CIN2/3-1801. Samples HSIL-66019 and HSIL-61979 were collected from one woman as two independent samples (see previous section) and contain an identical HPV16 integration site, which has already been reviewed for HSIL-66019 (Steinmeyer, 2009). Therefore, in the following, the HPV16 integration junctions in samples HSIL-75857 and CIN2/3-1801 are examined.

For sample HSIL-75857, the HPV16 DNA is integrated into chromosome 6 within an intron of the Abelson helper integration site 1 (AHI1) gene (Figure 2.53). The integration site is located about 98 kb downstream from the 5'end of the first exon of the AH1 gene, which is transcribed from the opposite strand to the integrated HPV16 DNA. The integration site is located in the proximity of another five cellular genes. The MYB proto-oncogene locates about 181 kb upstream from the integration site, while the

phosphodiesterase 7B (PDE7B) gene is about 451 kb downstream. Another three cellular genes are the HBS1-like (HBS1L), the non-protein coding RNA 271 (NCRNA00271), and the glyceraldehyde-3-phosphate dehydrogenase pseudogene 73 (GAPDHP73).

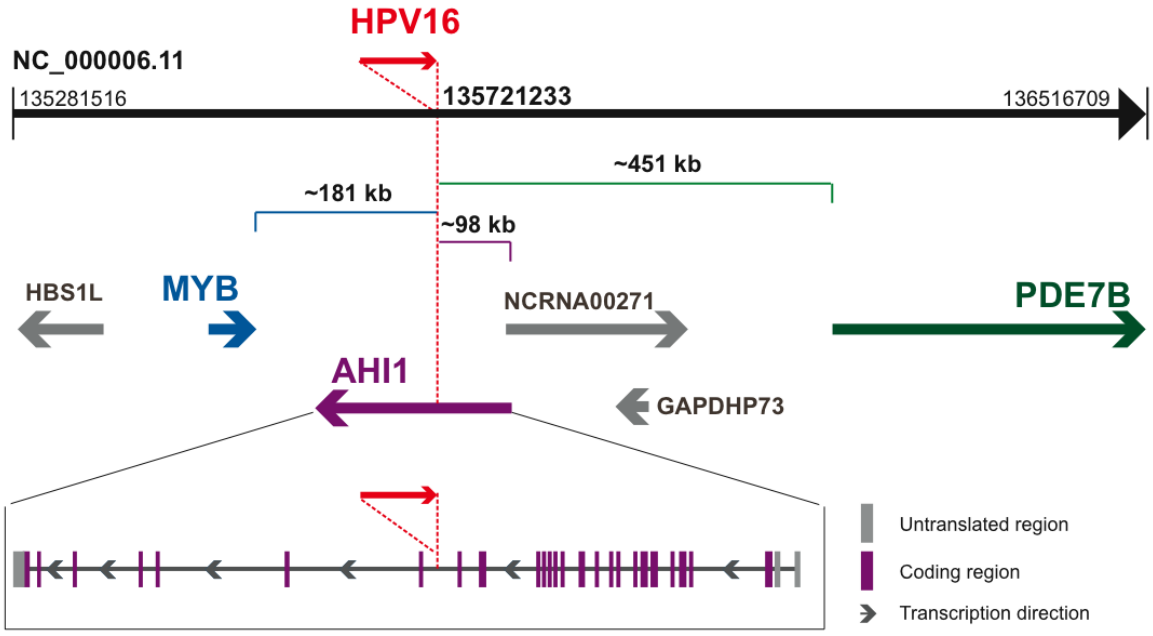


Figure 2.53: Cellular genes in close proximity to the integrated HPV16 of sample HSIL-75857. HPV16 DNA (red arrow) is integrated into chromosome 6 (top black arrow). Gray, blue, purple and green arrows represent coding regions and transcriptional directions of cellular genes, whose names are indicated above the arrows. The inlet at the bottom shows the exon-intron compositions of the AHI1 gene. AHI1: Abelson helper integration site 1. MYB: v-myb myeloblastosis viral oncogene homolog (avian). PDE7B: phosphodiesterase 7B. HBS1L: HBS1-like (S. cerevisiae). NCRNA00271: non-protein coding RNA 271. GAPDHP73: glyceraldehyde-3-phosphate dehydrogenase pseudogene 73.

For sample CIN2/3-1801, the HPV16 is integrated in chromosome 19 within an intron of the cellular gene receptor accessory protein 6 (REEP6) (Figure 2.54). The integration site is located 3240 bp downstream from the 5'end of the first exon of REEP6, which is transcribed from the opposite strand to the integrated HPV16 DNA. The proprotein convertase subtilisin/kexin type 4 (PCSK4) gene locates 3998 bp upstream from the integration site, while the adenomatosis polyposis coli 2 (APC2) gene is about 21 kb upstream. There are another three cellular genes in the proximity of the integration site: ADAMTS-like 5 (ADAMTSL5), pseudogene polo-like kinase 5 (PLK5P), and chromosome 19 open reading frame 25 (C19orf25).

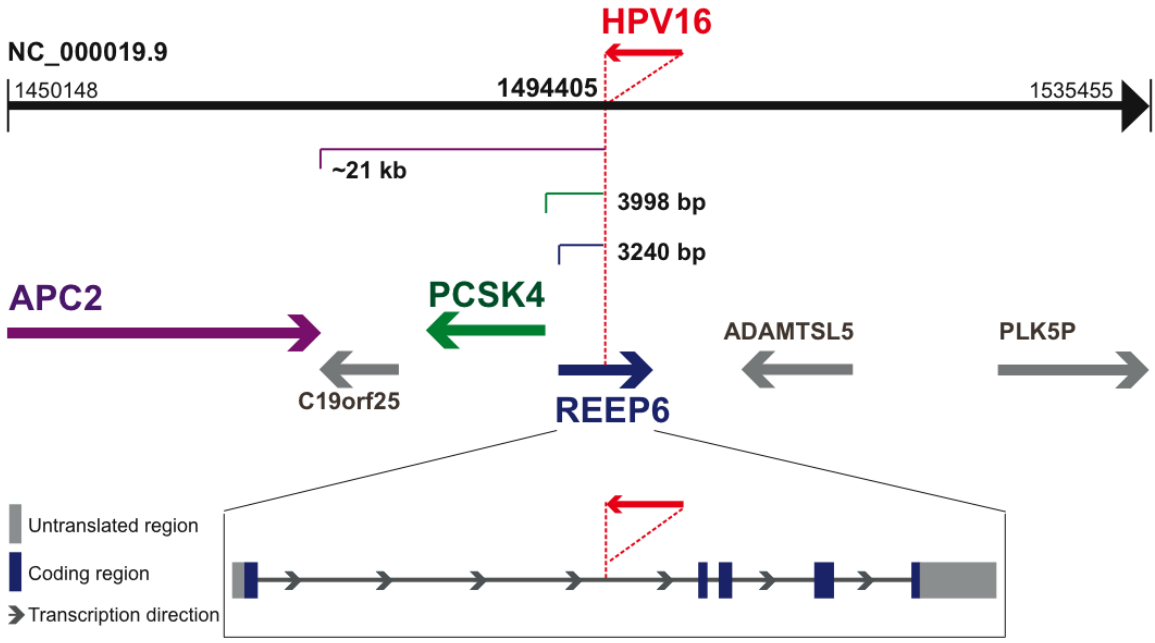


Figure 2.54: Cellular genes in close proximity to the integrated HPV16 of sample CIN2/3-1801. HPV16 DNA (red arrow) is integrated into chromosome 19 (top black arrow). Gray, green and blue arrows represent coding regions and the transcriptional directions of the cellular genes, whose names are indicated above the arrows. The inlet at the bottom shows the exon-intron composition of the REEP6 gene. REEP6: receptor accessory protein 6. ADAMTSL5: ADAMTS-like 5. APC2: adenomatosis polyposis coli 2. C19orf25: chromosome 19 open reading frame 25. PCSK4: proprotein convertase subtilisin/kexin type 4. PLK5P: polo-like kinase 5 (pseudogene).

2.2.2.8 Sequence coverage of the HPV16 E1-E2 region by ASP16 sequence reads

To determine whether the sequence reads from ASP16-3 and ASP16-4 cover the complete analyzed HPV16 E1-E2 area, a "representative E1-E2 sequence" of each DNA sample was created. First, a "representative primer group sequence" was selected from the alignment of each primer group of a DNA sample. Figure 2.55 shows an example. After the representative primer group sequences had been selected for all primer groups of a DNA sample, they were assembled to get the representative E1-E2 sequence of that DNA sample (Figure 2.56). The assembled sequences cover pos. 990-3853, starting with the first nucleotide after primer E18 (pos. 968-989) up to pos. 3853, which is the third nucleotide of the E2 stop codon TGA. Sequences from primer group E17 (810) were not included in the assembled E1-E2 sequences, because they were too short (about 50-60 nt) and never overlapped with sequences of the next primer E18 (968).

Figure 2.55: Example of a representative primer group sequence. The sequence alignment of primer group E08 of sample 3B11 is shown. HPV16R was used as reference sequence (top line). The four bases are highlighted in different color: A in red, C in blue, G in yellow and T in green. A representative primer group sequence (circled in blue) was selected as the sequence containing the most frequent nucleotide at every position, excluding the primer sequence. Red circles mark nucleotides that are different from the most frequent nucleotide at the given position.

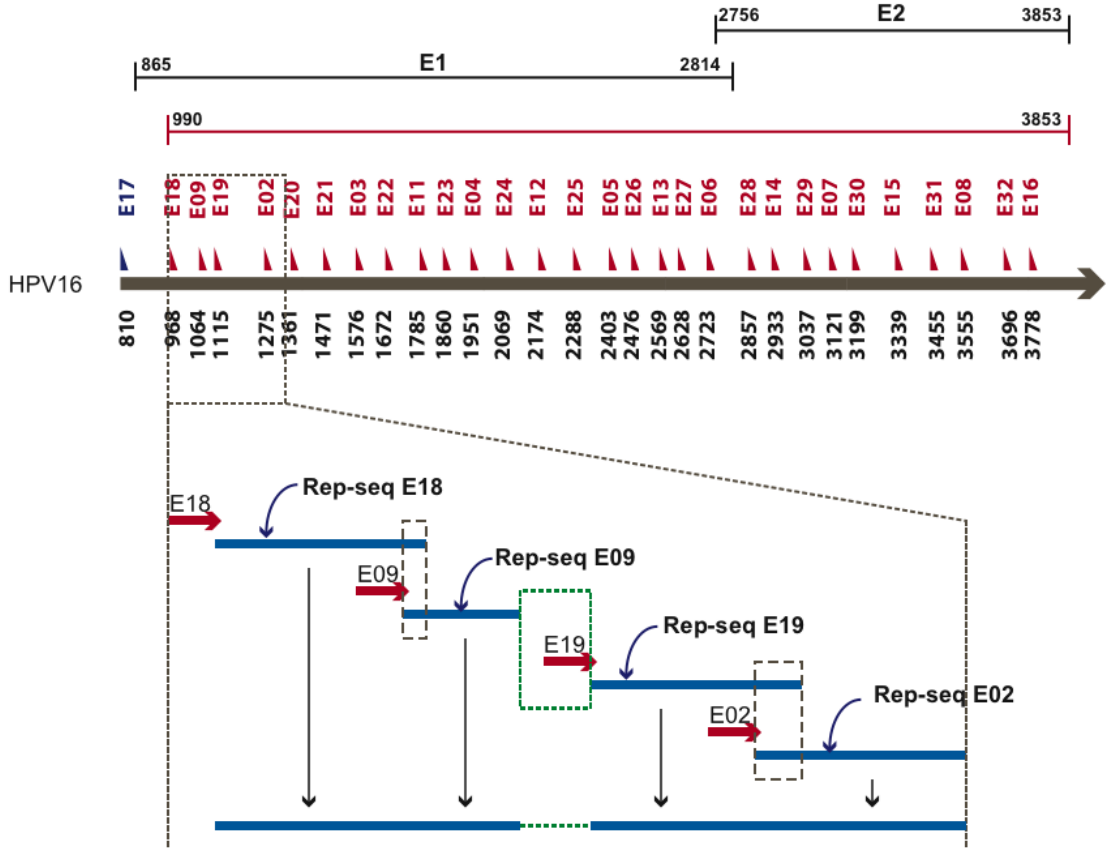


Figure 2.56: Model for assembly of a representative HPV16 E1-E2 sequence for a DNA sample. The locations of all RA_HPV16 primers are indicated along the black arrow. ORFs E1 and E2 are shown above as well as the area of assembled sequences (red line, top). In the lower part, the region from E18 to E02 is enlarged. The arrows represent four primers, for which representative primer group sequences (blue lines, "Rep-seq") are shown. The "representative E1-E2 sequence" of this DNA sample, assembled from the representative primer group sequences, is shown as blue line at the bottom. The dotted green line indicates a region where no sequence information is available from the ASP16 sequence reads.

From each assembled representative E1-E2 sequence, the sequence coverage was calculated by the following formula:

$$\frac{(\#\_of\_nucleotides\_in\_assembled\_seq\_from\_pos.\_990\_to\_3853)}{(\#\_of\_total\_nucleotides\_from\_pos.\_990\_to\_3853)} \times 100\%$$

The sequence coverage values for the 25 DNA samples are shown in Figure 2.57. The DNA samples HSIL-66019, HSIL-61979 and LSIL-75022 have very low sequence coverage, compared to others, due to the low number of sequence reads. Excluding these three DNA samples, the average sequence coverage in ASP16-3 and ASP16-4 was 89 %. Compared to the ASP16-2 in which only ~50% sequence coverage could be achieved, the optimized conditions in ASP16-3 and ASP16-4 gave much higher sequence coverage. The distribution of covered and uncovered parts in the assembled representative E1-E2 sequences are illustrated in Figure 2.58. The nucleotide sequences are given in the Appendix A5.

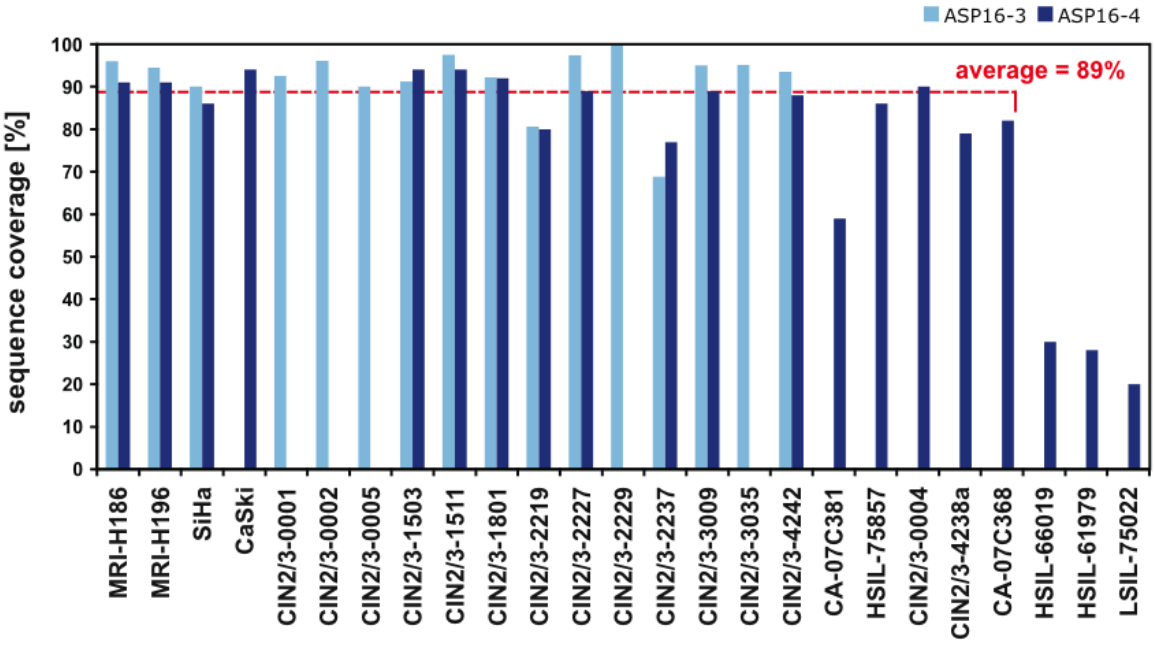


Figure 2.57: Sequence coverage in ASP16-3 and ASP16-4. Each column represents a sequence coverage value (in percent) of a DNA sample in each experiment. The sequence coverage covers the HPV16 E1-E2 area from pos. 990-3853. The columns are grouped by the DNA sample. The average sequence coverage from both experiments is 89%, excluding values from DNA samples HSIL-66019, HSIL-61979 and LSIL-75022.

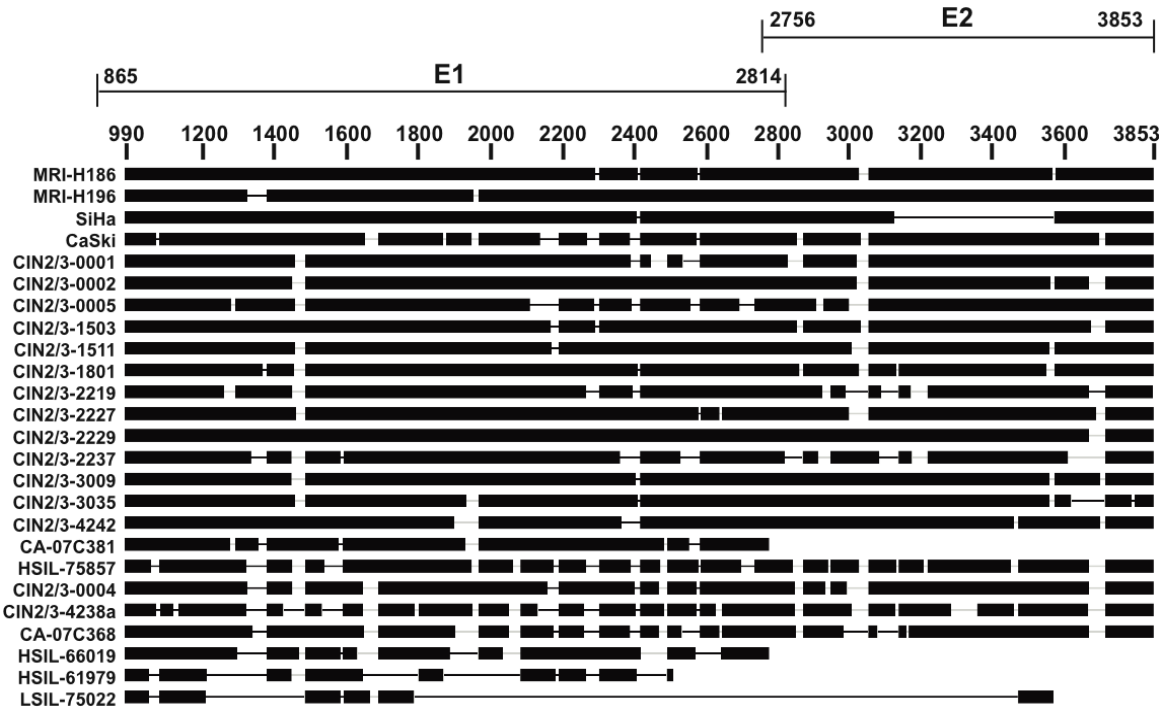


Figure 2.58: Assembled representative E1-E2 sequences of 25 DNA samples in ASP16-3 and ASP16-4. The HPV16R nucleotide positions are indicated on top, the DNA sample names on the left. ORFs E1 and E2 regions are indicated. Each black block represents a covered part for which sequence information could be assembled. Thin lines or blank represent uncovered parts for which no sequence information could be assembled from the ASP16 sequence reads.

2.2.2.9 Nucleotide mutation analysis in HPV16 E1-E2 area and HPV16 variant classification

HPV16 isolates are classified into variants, which are defined as HPVs that are maximally 2% different in their L1 nucleotide sequence (de Villiers et al, 2004). Many HPV16 variants had been identified not only by L1 sequence, but also by other regions such as E6, L2 and URR (Azizi et al, 2008; Casas et al, 1999; Chan et al, 1992; Ho et al, 1993; Kammer et al, 2002; Swan et al, 2005; Yamada et al, 1997; Yamada et al, 1995). Based on geographical distribution, five major phylogenetic branches (also called "lineages") and one minor branch were identified, designated European (E), Asian (As), African-1 (Af1), African-2 (Af2), Asian-American (AA) and North-American-1 (NA1), respectively (Chan et al, 1992; Ho et al, 1993; Yamada et al, 1995). Linkage of nucleotide polymorphisms between different regions of the same HPV16 isolates had been demonstrated (Casas et al, 1999; Eriksson et al, 1999; Swan et al, 2005). Some HPV16 variants were more often detected in high-grade cervical lesions than others, suggesting an important impact of HPV16 variants in cervical carcinogenesis (Kammer et al, 2000;

Matsumoto et al, 2000; Zehbe et al, 1998). HPV16 variants were shown to have different biochemical and biological properties and these may be one of the reasons for their different carcinogenic risks (Giannoudis & Herrington, 2001).

The assembled representative E1-E2 sequences were used for HPV16 variant classification of the 25 analyzed DNA samples. This was first done for E2. First, a collection of HPV16 variants gathered from the literature were sorted according to their lineage, as shown in Table 2.29, panel A. For each lineage, the nucleotide polymorphism patterns of the collected variants were manually aligned, and the variants were sub-grouped accordingly. As a result, a set of nine "reference nucleotide polymorphism pattern groups" covering the HPV16 E2 ORF (pos. 2756-3853), was created: four groups for the E lineage, two for the AA lineage and one for each As, Af1 and Af2 lineage. A reference nucleotide polymorphism pattern was created for each group, with mandatory and optional nucleotide mutations defined. The mandatory mutations are present in all variants within the same group, while the optional mutations are not. Because no E2 sequence has yet been described for the NA1 lineage, any variant of NA1 lineage could not be identified by this set of reference E2 polymorphism patterns.

Based on the available sequence information in the assembled representative E1-E2 sequences, the twenty-five DNA samples were assigned to the most similar or identical reference pattern, as shown in Table 2.29, panel B. All nucleotide mutations in the E2 ORF region were included in the table. Fifteen samples were successfully assigned to a single pattern, while it was not possible for the remaining 10 samples due to lack of adequate sequence information. Additional thirteen mutation positions, which are not present in the collected HPV16 variant sequences (Table 2.29, panel A), were found in nine DNA samples (Table 2.29, panel B). These mutations may either be variant-specific thus indicating new unknown HPV16 variants, or sample-specific indicating additional, spontaneous mutations occurred in the sample infected with known variants. For DNA samples CIN2/3-3009, CIN2/3-3035 and CIN2/3-4242, their patterns showed the highest similarity to those of AA (1) and AA (2), but do not fit all the mandatory mutations for them to be assigned with either pattern. These three samples may contain HPV16 variants of NA1 lineage because it is closely related to AA lineage (Yamada et al, 1995).

For DNA samples that could not be successfully assigned to a variant/lineage by the E2 sequence, assignment was accomplished by using the HPV16 E6 region. The nucleotide sequence of the NA1 lineage for the E6 region is available (Yamada et al, 1995), therefore all six lineages could now be assigned. Similar to E2, a set of reference nucleotide polymorphism patterns covering the E6 ORF (pos. 83 -559) was created from a collection of 35 HPV16 variants, as shown in Table 2.30, panel A. Out of these 35 variants, thirty-two had been used previously for the E2 reference patterns. Seven reference patterns were created for all six lineages. The HPV16 variants of the E lineage were divided into two pattern groups, Ep and E-T350G, whereas a single reference pattern was assigned to each of the other five lineages (Table 2.30, panel A). The sequences of HPV16 E6 ORF in the 25 DNA samples were determined by PCR amplification using the primers H16-44F and H16-691R (sequences shown in Materials and Methods), cloning and sequencing. The HPV16 E6 sequences of the DNA samples (shown in Appendix A6) were compared to the reference patterns, and the most similar or identical pattern was assigned to all DNA samples (Table 2.30, panel B).

All DNA samples could now be successfully assigned. As assumed from the E2 sequences, the DNA samples CIN2/3-3009, CIN2/3-3035 and CIN2/3-4242 were assigned to NA1 lineage by E6 sequences. No E2 sequence information had been obtained for samples HSIL-66019, HSIL-61979 and CA-07C381 because the breakpoints of the fully integrated HPV16 genomes are located upstream of E2. The E6 sequences showed that samples HSIL-66019 and HSIL-61979 belong to the Ep variant group, and sample CA-07C381 the E-T350G variant group.

The nucleotide polymorphisms in the E1 gene were obtained from the assembled representative HPV16 E1-E2 sequences and are shown in parallel with those of E2 and E6 genes (Table 2.31). Despite incomplete sequence information in the E1-E2 region, nucleotide co-variation in E1, E2 and E6 genes among the same lineage became apparent, especially for lineages NA1 and Af1. Though incomplete, these data provide for the first time the clear linkage of nucleotide variations in E1, E2 and E6 genes of the NA1 lineage.

The effects of the nucleotide variations on the amino acid sequences were investigated. The E6 protein sequences were translated from the complete E6 nucleotide sequences. For E1 and E2 proteins, the assembled nucleotide representative E1-E2 sequences were

translated into amino acid residues. Because the gaps in the assembled sequences prevent correct-frame translation, "N" nucleotides were added to these gaps to make continuous sequences. The amino acid changes of E6, E1 and E2 protein of all DNA samples are shown in Table 2.32. No mutation inducing premature stop codon (nonsense mutation) was observed. The most frequently detected non-synonymous mutations were E6 L83V (in 16 of 25 samples), E1 S220T (in 12 of 25 samples) and E2 P219S (in 14 of 20 samples). The number of non-synonymous mutations is similar to the number of silent mutations in most cases (European lineage ORFs E6, E1 and E2; NA1/Af1/Af2 lineages ORFs E6 and E1). In ORF E2 of NA1/Af1/Af2 lineages, however, the number of non-synonymous mutations is 3-4 fold higher than that of silent mutations.

Table 2.29: Nucleotide polymorphisms in HPV16 ORF E2 and variant assignment.

A: Reference patterns of nucleotide polymorphisms in ORF E2.

Pattern group (a)	Original isolate/variant name from the source	Source (b)	2828	2860	2938	2985	2990	3043	3068	3111	3159	3161	3162	3181	3182	3205	3224	3235	3243	3249	3262	3335	3377	3384	3387	3410	3411	3418	3422	3425	3431	3449	3465	3516	3524	3538	3551	3556	3566	3575	3603	3645	3664	3684	3694	3706	3778	3784	3807	3837
	HPV16R		G	C	A	T	T	C	G	T	C	C	A	A	G	T	T	T	G	G	A	G	C	T	T	C	G	G	C	A	G	G	C	T	T	A	C	C	T	T	G	C	T	C	T	T	G	T	C	T	G
E (1)	isolate Qv17722E (accession: AY686584)	Ref 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Ep	Ref 7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	isolate 995	Ref 8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	HPV16-E2-MTL27	Ref 1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-
	HPV16-E2-MTL14	Ref 1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	HPV16-E2-MTL24	Ref 1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
E (2)	clone 114/K (accession: EU118173)	Ref 5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	isolate Qv18158E (accession: AY686583)	Ref 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	isolate Qv16936E (accession: AY686580)	Ref 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	isolate 16W12E (accession: AF125673)	Ref 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	E-C109G	Ref 7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	E-G350	Ref 7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Ep-c	Ref 7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Grp1_8-10 (accession: AF193427)	Ref 4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	M2	Ref 9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	K2	Ref 9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	C1	Ref 9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	H2	Ref 9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	K1	Ref 9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	isolate 936	Ref 8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	H3	Ref 9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
E (3)	HPV16-E2-MTL26	Ref 1	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	HPV16-E2-MTL28	Ref 1	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Ep-a	Ref 7	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	K6	Ref 9	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	HPV16-E2-MTL29	Ref 1	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	isolate Qv15521E (accession: AY686581)	Ref 3	-	-	-	-	-	-	A	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-	-	C	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
E (4)	CaSki cell line (accession: U89348)	Ref 6	-	-	G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-
	E-G131G	Ref 7	-	-	G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-
	Ep-b	Ref 7	-	-	G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-
	Grp2_11-12 (accession: AF193428)	Ref 4	-	-	G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-
	T1	Ref 9	-	-	G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-
	isolate 59	Ref 8	-	-	G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-
	isolate European German (accession: AF536179)	Ref 3	-	-	G	C	C	-	-	C	-	-	-	-	-	-	-	-	-	-	-	-	C	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-
As	isolate LZcc11-16 (accession: EU918764)	Ref 10	-	-	-	-	-	A	-	A	-	-	-	-	-	-	-	-	-	-	-	-	C	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	T
	As	Ref 7	A	-	-	-	-	-	-	A	-	-	-	-	-	A	-	-	-	-	-	-	C	-	T	-	-	-	-	-	A	-	-	C	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-
	S1	Ref 9	A	-	-	-	-	-	-	A	-	-	-	-	-	A	-	-	-	-	-	-	C	-	T	-	-	-	-	-	A	-	-	C	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-
Af1	HPV16-E2-MTL33	Ref 1	-	A	-	-	-	-	-	A	T	-	-	A	-	-	C	-	A	G	-	G	-	-	-	-	-	C	-	A	A	-	-	-	-	G	-	-	-	-	A	-	T	G	A	-	-	-	-	-	-
	HPV16-E2-MTL35	Ref 1	-	A	-	-	-	-	-	A	T	G	-	A	-	-	C	-	A	G	-	G	-	-	-	-	-	C	-	A	A	-	-	-	-	G	-	-	-	-	A	-	T	G	A	-	-	-	-	-	-
	HPV16-E2-MTL34	Ref 1	-	A	-	-	-	-	-	A	T	-	-	A	-	-	C	-	A	G	-	G	-	-	-	-	-	T	-	A	A	-	-	-	-	G	-	-	T	-	-	A	-	T	G	A	-	-	-	-	-	-
	Af1 (accession: AF472508)	Ref 3	-	A	-	-	-	-	-	A	T	-	-	A	-	-	C	-	A	G	-	G	-	-	T	-	-	T	-	A	A	-	-	-	-	G	-	-	T	-	A	A	-	T	G	A	-	-	-	-	-	-
	Af1	Ref 7	-	A	-	-	-	-	-	A	T	-	C	A	-	-	-	-	A	G	-	G	-	-	T	-	-	C	-	A	A	-	-	-	-	G	-	-	-	C	A	A	-	T	G	A	-	-	-	-	-	-
Af2	HPV16-E2-MTL36	Ref 1	-	A	-	-	T	-	-	A	T	-	-	A	-	-	-	-	A	G	-	G	-	-	-	-	-	-	-	A	A	A	C	-	C	-	-	G	-	-	-	-	A	A	C	T	-	A	G	-	-	-
	Af2 (accession: AF472509)	Ref 3	-	A	-	-	T	-	-	A	T	-	-	A	-	-	-	-	A	G	-	G	-	-	T	-	-	-	-	A	A	A	C	-	C	-	-	G	-	-	-	-	A	A	C	T	-	A	G	-	-	-
	Af2	Ref 7	-	A	-	-	T	-	-	A	T	-	-	A	-	-	-	-	A	G	-	G	-	-	T	-	-	-	-	A	A	A	C	-	C	-	-	G	-	-	-	-	A	A	C	T	-	A	G	-	-	-
	Grp3_13-15 (accession: AF193429)	Ref 4	-	A	-	-	-	-	-	A	T	-	-	A	-	-	-	-	A	G	-	G	-	-	T	-	-	-	-	A	A	A	C	-	C	-	-	G	-	-	-	-	A	A	C	T	-	A	G	-	-	-
AA (1)	HPV16-E2-MTL39	Ref 1	-	A	-	-	-	-	-	A	-	-	C	A	-	-	-	-	A	G	-	G	-	C	-	-	-	-	-	-	A	A	C	-	C	-	-	G	-	-	-	C	A	A	C	T	-	A	G	-	-	-
	AA	Ref 7	-	A	-	-	-	-	-	A	-	-	C	A	-	A	-	-	A	G	-	G	-	C	T	-	-	-	-	-	A	A	C	-	C	-	-	G	-	-	-	C	A	A	C	T	-	A	G	-	-	-
	AA (accession: AF402678)	Ref 3	-	A	-	-	-	-	-	A	-	-	C	A	-	A	-	-	A	G	-	G	-	C	T	-	-	-	-	-	A	A	C	-	C	-	-	G	-	-	-	C	A	A	C	T	-	A	G	-	-	-
	isolate 33	Ref 8	-	A	-	-	-	-	-	A	-	-	C	A	-	A	-	-	A	G	-	G	-	C	T	-	-	-	-	-	A	A	C	-	C	-	-	G	-	-	-	C	A	A	C	T	-	A	G	-	-	-
	AA-a__MX68/121/133 (accession: AF115831)	Ref 2	-	A	-	-	-	-	-	A	-	-	C	A	-	A	-	-	A	G	-	G	-	C	T	-	-	-	-	-	A	A	C	-	C	-	-	G	-	-	-	C	A	A	C	T	-	A	G	-	-	-
	AA-a__MX69	Ref 2	-	A	-	-	-	-	-	A	-	-	C	A	-	A	-	-	A	G	-	G	-	C	T	-	-	-	-	-	A	A	C	-	C	-	-	G	-	-	-	C	A	A	C	T	-	A	G	-	-	-
	AA-a__MX109	Ref 2	-	A	-	-	-	-	-	A	-	-	C	A	-	A	-	-	A	G	-	G	-	C	T	-	-	-	-	-	A	A	C	-	C	-	-	G	-	-	-	C	A	A	C	T	-	A	G	-	-	-
AA (2)	K4	Ref 9	-	A	-	-	-	-	-	A	-	-	-	A	-	A	-	-	A	G	-	G	-	-	T	-	-	-	-	-	A	A	C	-	C	-	-	G	-	-	-	-	A	A	C	T	-	A	G	-	-	-
	AA-c__MX70/55/97/119 (accession: AF115832)	Ref 2	-	A	-	-	-	-	-	A	T	-	-	A	-	A	-	-	A	G	-	G	-	-	T	A	-	-	-	-	A	A	C	-	C	-	-	G	-	-	-	-	A	A	C	T	-	A	G	-	-	-
Reference patterns for E2 polymorphisms (c)		E (1)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	-	-	A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		E (2)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		E (3)	-	-	-	-	-	-	A	-	-	-	-	-	A	-	-	-	-	-	A	-	-	-	-	C	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		E (4)	-	-	G	C	C	-	-	C	-	-	-	-	-	-	-	-	-	-	-	-	C	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-
		As	A	-	-	-	-	A	-	A	-	-	-	-	-	A	-	-	-	-	-	-	C	-	T	-	-	-	-	-	A	-	-	C	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	T
		Af1	-	A	-	-	-	-	-	A	T	G	C	A	-	-	C	-	A	G	-	G	-	-	T	-	-	C/T	-	A	A	-	-	-	-	G	-	-	T	C	A	A	-	T	G	A	-	-	-	-	-	-
		Af2	-	A	-	-	T	-	-	A	T	-	-	A	-	-	-	-	A	G	-	G	-	-	T	-	-	-	-	A	A	A	C	-	C	-	-	G	-	-	-	-	A	A	C	T	-	A	G	-	-	-
		AA (1)	-	A	-	-	-	-	-	A	-	-	C	A	-	A	-	-	A	G	-	G	-	C	T	-	-	-	-	-	A	A	C	-	C	-	-	G	-	-	-	C	A	A	C	T	-	A	G	-	-	-
		AA (2)	-	A	-	-	-	-	-	A	T	-	-	A	-	A	-	-	A	G	-	G	-	-	T	A	-	-	-	-	A	A	C	-	C	-	-	G	-	-	-	-	A	A	C	T	-	A	G	-	-	-

B (Table 2.29 continued): Nucleotide polymorphisms in ORF E2 of 25 DNA samples in comparison with the reference patterns.

DNA sample	Nucleotide pos. (d)	2828	2860	2938	2985	2990	3043	3061	3068	3118	3159	3161	3162	3181	3182	3186	3205	3208	3217	3224	3229	3235	3243	3249	3286	3317	3362	3365	3377	3384	3387	3410
	E2 amino acid residue	25	35	61	77	79	96	102	105	121	135	136	136	142	143	144	150	151	154	157	158	160	163	165	177	188	203	204	208	210	211	219
	E2 aa change (e)		*		*				*		*	*	*		*	*				*			*	*		*	*	*	*	*		*
HPV16R		G	C	A	T	T	C	T	G	T	C	C	A	A	G	C	T	A	T	T	T	T	G	G	A	G	A	G	C	T	T	C
Reference patterns (c)	E (1)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	-	-	-	-	A	-	-	-	-
	E (2)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T
	E (3)	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-
	E (4)	-	-	G	C	C	-	-	-	C	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	-	T
	As	A	-	-	-	-	-	-	A	-	A	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	C	-	T
	Af1	-	A	-	-	-	-	-	-	-	A	T	G	C	A	-	-	-	-	-	-	C	-	A	-	-	G	-	G	-	-	T
	Af2	-	A	-	-	-	T	-	-	-	A	T	-	-	A	-	-	-	-	-	-	-	-	A	-	-	G	-	G	-	-	T
	AA (1)	-	A	-	-	-	-	-	-	-	A	-	-	C	A	-	-	-	-	A	-	-	-	A	-	-	G	-	G	-	C	T
	AA (2)	-	A	-	-	-	-	-	-	-	A	T	-	-	A	-	-	-	-	A	-	-	-	A	-	-	G	-	G	-	-	T
CIN2/3-0004	E (1)	-	#	-	-	-	#	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CA-07C368	E (1)	-	#	-	-	-	#	-	-	#	-	-	-	-	-	-	-	-	C	-	-	-	-	-	-	-	-	-	-	-	-	-
CIN2/3-0001	E (2)	-	#	-	-	-	#	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T
CIN2/3-0002	E (2)	-	-	-	-	-	#	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T
CIN2/3-1503	E (2)	-	-	-	-	-	#	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T
CIN2/3-1801	E (2)	-	-	-	-	-	#	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	T
CIN2/3-2227	E (2)	-	-	-	-	-	#	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T
HSIL-75857	E (2)	-	#	-	-	-	#	-	-	-	-	-	-	-	-	-	-	-	#	-	-	-	-	-	-	-	-	-	-	-	-	T
MRI-H196	E (2)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	T
SiHa	E (1 or 2)	-	-	-	-	-	-	-	A	-	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
MRI-H186	E (3)	-	-	-	-	-	#	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CaSki	E (4)	-	-	G	-	-	#	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	-	T
CIN2/3-2229	E (4)	-	-	-	-	-	-	-	-	C	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	-	T
CIN2/3-4238a	E (4)	-	#	-	-	-	#	-	-	C	-	-	-	-	-	-	-	G	-	-	C	-	-	-	-	#	-	-	-	C	-	T
LSIL-75022	E (1 - 4)	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
HSIL-66019	not possible	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
HSIL-61979	not possible	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
CA-07C381	not possible	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
CIN2/3-0005	Af1 or Af2	-	A	-	-	-	#	C	-	-	A	T	-	-	A	T	-	-	-	-	-	-	-	A	-	-	G	-	G	-	-	T
CIN2/3-1511	Af1 or Af2	-	A	-	-	-	#	C	-	-	A	T	-	-	A	T	-	-	-	-	-	-	-	A	-	-	G	-	G	-	-	T
CIN2/3-2219	Af1	-	A	#	-	-	#	-	-	#	A	T	-	#	#	#	#	#	#	-	-	C	-	A	-	-	G	-	G	-	-	T
CIN2/3-2237	Af2	#	#	#	-	-	T	-	-	#	A	T	-	-	#	#	#	#	#	-	-	-	-	A	G	A	G	-	G	-	-	T
CIN2/3-3009	(similar to AA)	-	A	-	-	-	-	-	-	-	G	T	-	-	A	-	-	-	-	A	-	-	-	A	-	-	G	-	G	-	-	-
CIN2/3-3035	(similar to AA)	-	A	-	-	-	-	-	-	-	G	T	-	-	A	-	-	-	-	A	-	-	-	A	-	-	G	-	G	-	-	-
CIN2/3-4242	(similar to AA)	-	A	-	-	-	-	-	-	-	G	T	-	-	A	-	-	-	-	A	-	-	-	A	-	-	G	-	G	-	-	-

DNA sample	Nucleotide pos. (d)	3416	3418	3422	3425	3428	3431	3449	3471	3491	3516	3517	3524	3538	3551	3556	3566	3575	3603	3629	3645	3664	3684	3694	3706	3778	3784	3787	3790	3805	3837	3846
	E2 amino acid residue	221	221	223	224	225	226	232	239	246	254	254	257	261	266	267	271	274	283	292	297	303	310	313	317	341	343	344	345	350	361	364
	E2 aa change (e)			*	*	*	*	*	*	*	*				*		*		*	*	*		*			*		*	*		*	*
HPV16R		G	G	C	A	A	G	G	A	G	C	T	T	A	C	C	T	T	G	A	C	T	C	T	T	G	T	C	A	T	G	C
Reference patterns (c)	E (1)	-	-	-	-	-	-	-	-	-	-	-	-	-	G	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	E (2)	-	-	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-	-	-	-	-	-
	E (3)	-	C	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	E (4)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-	-
	As	-	-	-	-	-	-	A	-	-	-	-	C	-	-	-	-	-	-	-	-	-	A	-	-	-	-	A	-	-	T	-
	Af1	-	-	-	C/T	-	-	A	-	-	A	-	-	-	-	-	G	-	-	-	T	C	A	A	-	T	G	A	-	-	-	-
	Af2	-	-	-	-	-	A	A	-	-	A	C	-	C	-	-	G	-	-	-	-	-	A	A	C	T	-	A	-	G	-	-
	AA (1)	-	-	-	-	-	-	A	-	-	A	C	-	C	-	-	G	-	-	-	-	C	A	A	C	T	-	A	-	G	-	-
	AA (2)	A	-	-	-	-	-	A	-	-	A	C	-	C	-	-	G	-	-	-	-	-	A	A	C	T	-	A	-	G	-	-
CIN2/3-0004	E (1)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	#	#	#	-	-	-	-	-	-	-
CA-07C368	E (1)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	#	#	#	-	-	-	-	-	-	-
CIN2/3-0001	E (2)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CIN2/3-0002	E (2)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	#	-	-	-	-	#	#	#	-	-	-	-	-	-	-
CIN2/3-1503	E (2)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	#	#	#	-	-	-	-	-	-	-
CIN2/3-1801	E (2)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	#	#	#	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CIN2/3-2227	E (2)	-	-	-	-	C	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	#	#	-	-	-	-	-	-	-
HSIL-75857	E (2)	-	-	T	-	-	-	-	#	-	-	-	-	-	-	-	-	-	-	-	-	-	#	#	#	-	-	-	T	-	-	-
MRI-H196	E (2)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	-	-	-	-	-	-	-	-	-	-	-	-
SiHa	E (1 or 2)	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	-	-	-	-	A	-	-	-	-	-	-	-	-	-
MRI-H186	E (3)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	#	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CaSki	E (4)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	#	-	-	-	-	-	-	-
CIN2/3-2229	E (4)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	#	#	#	-	-	-	-	-	-	-
CIN2/3-4238a	E (4)	-	-	-	-	-	-	-	#	-	-	-	-	-	-	-	-	-	-	-	-	-	#	#	#	-	-	-	-	-	-	-
LSIL-75022	E (1 - 4)	#	#	#	#	#	#	#	#	A	-	-	-	-	-	-	-	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
HSIL-66019	not possible	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
HSIL-61979	not possible	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
CA-07C381	not possible	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#
CIN2/3-0005	Af1 or Af2	-	-	-	-	-	-	A	-	-	A	-	-	-	-	-	G	-	-	-	-	-	A	A	-	T	-	A	-	C	-	G
CIN2/3-1511	Af1 or Af2	-	-	-	-	-	-	A	-	-	A	-	-	-	-	-	#	#	-	-	-	-	A	A	-	T	-	A	-	C	-	G
CIN2/3-2219	Af1	-	-	-	T	-	-	A	-	-	A	-	-	-	-	-	G	-	-	-	-	-	#	#	#	T	G	A	-	-	-	-
CIN2/3-2237	Af2	-	-	-	-	-	A	A	T	-	A	C	-	C	-	-	G	-	-	#	#	#	#	#	#	T	-	A	-	G	-	-
CIN2/3-3009	(similar to AA)	-	-	-	-	-	-	A	-	-	A	C	-	C	-	-	#	#	-	-	-	C	A	A	#	T	-	A	-	G	-	-
CIN2/3-3035	(similar to AA)	-	-	-	-	-	-	A	-	-	A	C	-	C	-	-	#	#	-	#	#	#	#	#	#	T	-	A	-	G	-	-
CIN2/3-4242	(similar to AA)	-	-	-	-	-	-	A	#	-	A	C	-	C	-	-	G	-	-	-	-	C	A	A	#	T	-	A	-	G	-	-

(a) Created according to groups of similar variants. (b) Ref 1 (Azizi et al, 2008), Ref 2 (Casas et al, 1999), Ref 3 (Chen et al, 2005), Ref 4 (Graham & Herrington, 2000), Ref 5 (Kirnbauer et al, 1993), Ref 6 (Seedorf et al, 1985), Ref 7 (Swan et al, 2005), Ref 8 (Veress et al, 1999), Ref 9 (Watts et al, 2002), and Ref 10 (Wu et al, 2009). (c) Summary of reference patterns derived based on the pattern groups. Green highlight indicates mandatory polymorphisms within each pattern group, yellow highlight optional ones. "-" indicates no change in nucleotide, and "#" no sequence information. (d) Blue colored positions indicate no mutation of that position present in the DNA samples. (e) Non-synonymous mutations inducing an amino acid change are indicated with "*". Gray highlight indicates mutations not present in the reference patterns.

Table 2.30: Nucleotide polymorphisms in HPV16 ORF E6 and variant assignment.

A: Reference patterns of nucleotide polymorphisms in ORF E6.

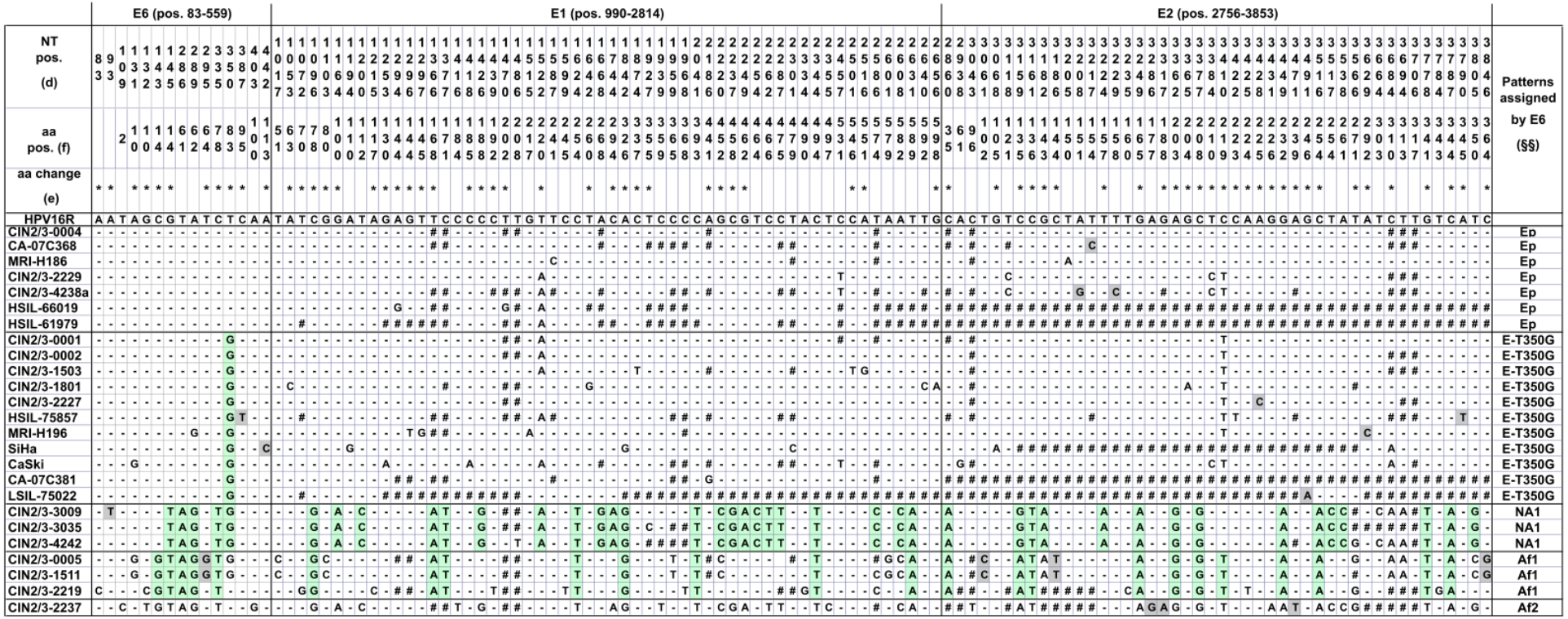
Pattern group (a)	Pattern group by E2 ORF (§)	Original isolate/variant name from the source	Source (b)	83	109	131	132	143	145	172	176	178	183	256	286	289	335	350	362	403	433	457	505	532
HPV16R				A	T	A	G	C	G	A	G	T	T	C	T	A	C	T	A	A	G	T	T	A
Ep	E (1)	Ep	Ref 7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ep	E (1)	isolate Qv17722E (accession: AY686584)	Ref 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	-	-
Ep	E (2)	Ep-c	Ref 7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ep	E (2)	H3	Ref 9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-
Ep	E (2)	K1	Ref 9	-	-	-	-	-	#	C	A	-	-	-	-	-	-	-	-	-	-	-	-	-
Ep	E (3)	K6	Ref 9	-	-	-	-	-	#	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ep	E (3)	Ep-a	Ref 7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ep	E (3)	isolate Qv15521E (accession: AY686581)	Ref 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ep	E (4)	Ep-b	Ref 7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ep	E (4)	T1	Ref 9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	C	-
E-T350G	E (2)	E-G350	Ref 7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-
E-T350G	E (2)	clone 114/K (accession: EU118173)	Ref 5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-
E-T350G	E (2)	isolate Qv16936E (accession: AY686580)	Ref 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-
E-T350G	E (2)	H2	Ref 9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-
E-T350G	E (2)	C1	Ref 9	-	-	-	-	-	#	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-
E-T350G	E (2)	E-C109G	Ref 7	-	C	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-
E-T350G	E (2)	isolate Qv18158E (accession: AY686583)	Ref 3	-	-	-	-	-	-	-	A	-	-	-	-	-	-	G	-	-	-	-	-	-
E-T350G	E (2)	isolate 16W12E (accession: AF125673)	Ref 3	-	-	-	-	-	-	-	-	-	-	T	-	-	-	G	-	-	-	-	-	-
E-T350G	E (2)	K2	Ref 9	-	-	-	-	-	#	C	-	-	-	-	-	-	-	G	-	-	-	-	-	-
E-T350G	E (2)	M2	Ref 9	-	-	-	-	-	#	C	A	-	-	-	-	-	-	G	-	-	-	-	-	-
E-T350G	E (4)	E-G131G	Ref 7	-	-	G	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-
E-T350G	E (4)	CaSki cell line (accession: U89348)	Ref 6	-	-	G	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-
E-T350G	E (4)	isolate European German (AF536179)	Ref 3	-	-	G	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-
As	As	As	Ref 7	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	-	-	-	-
As	As	S1	Ref 9	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	-	-	-	-
Af1	Af1	Af1	Ref 7	C	-	-	C	G	T	-	-	-	-	-	A	G	T	-	-	-	-	-	-	-
Af1	Af1	Af1 (accession: AF472508)	Ref 3	C	-	-	C	G	T	-	-	-	-	-	A	G	T	-	-	-	-	-	-	-
Af1	-	IS.398	Ref 12	-	-	G	-	G	T	-	-	-	-	-	A	G	T	G	-	-	-	-	-	-
Af1	-	IS.347	Ref 12	-	-	-	-	G	T	-	-	-	-	-	A	G	T	G	-	-	-	-	-	-
Af2	Af2	Af2 (accession: AF472509)	Ref 3	-	C	-	T	G	T	-	-	-	-	-	A	G	T	-	-	G	-	-	-	-
Af2	Af2	Af2	Ref 7	-	C	-	T	G	T	-	-	-	-	-	A	G	T	-	-	G	-	-	-	-
NA1	-	OR.3136	Ref 11	-	-	-	-	-	T			-			A	G	T	G	-	-	-	-	-	-
AA	AA (1)	AA	Ref 7	-	-	-	-	-	T	-	-	-	-	-	A	G	T	G	-	-	-	-	-	G
AA	AA (1)	AA (accession: AF402678)	Ref 3	-	-	-	-	-	T	-	-	-	-	-	A	G	T	G	-	-	A	-	-	G
AA	AA (2)	K4	Ref 9	-	-	-	-	-	T	-	-	-	G	-	A	G	T	G	-	-	-	-	C	G
Reference patterns for E6 polymorphisms (c)			Ep	-	-	-	-	-	-	C	A	-	-	-	-	-	-	-	G	-	-	C	C	-
			E-T350G	-	C	G	-	-	-	C	A	-	-	T	-	-	-	G	-	-	-	-	-	-
			As	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	-	-	-	-
			Af1	C	-	G	C	G	T	-	-	-	-	-	A	G	T	G	-	-	-	-	-	-
			Af2	-	C	-	T	G	T	-	-	-	-	-	A	G	T	-	-	G	-	-	-	-
			NA1	-	-	-	-	-	T	-	-	-	-	-	A	G	T	G	-	-	-	-	-	-
			AA	-	-	-	-	-	T	-	-	-	G	-	A	G	T	G	-	-	A	-	C	G

B: Nucleotide polymorphisms in ORF E6 of 25 DNA samples in comparison with the reference patterns.

DNA sample	Nucleotide pos. (d)	83	93	109	131	132	144	145	172	176	178	183	256	286	289	295	335	350	362	387	400	403	442	450	502	532	Patterns assigned by E2 (§)
	E6 amino acid residue (f)			2	10	10	14	14	23	25	25	27	51	61	62	64	78	83	87	95	100	110	113	118	134	143	
	E6 aa change (e)	*	*		*	*	*	*		*	*	*	*			*	*	*	*	*			*				*	
HPV16R		A	A	T	A	G	C	G	A	G	T	T	C	T	A	T	C	T	A	C	A	G	A	T	T	A	
Reference patterns (c)	Ep	-	-	-	-	-	-	-	C	A	-	-	-	-	-	-	-	G	-	-	-	-	-	C	C	-	
	E-T350G	-	-	C	G	-	-	-	C	A	-	-	T	-	-	-	-	G	-	-	-	-	-	-	-	-	
	As	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	Af1	C	-	-	G	C	G	T	-	-	-	-	-	A	G	-	T	G	-	-	-	-	-	-	-	-	
	Af2	-	-	C	-	T	G	T	-	-	-	-	-	A	G	-	T	-	-	-	G	-	-	-	-	-	
	NA1	-	-	-	-	-	-	T	-	-	-	-	-	A	G	-	T	G	-	-	-	-	-	-	-	-	
	AA	-	-	-	-	-	-	T	-	-	-	G	-	A	G	-	T	G	-	-	A	-	-	C	G		
CIN2/3-0004	Ep	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	E (1)
CA-07C368	Ep	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	E (1)
MRI-H186	Ep	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	E (3)
CIN2/3-2229	Ep	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	E (4)
CIN2/3-4238a	Ep	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	E (4)
HSIL-66019	Ep	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	not possible
HSIL-61979	Ep	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	not possible
CIN2/3-0001	E-T350G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	E (2)
CIN2/3-0002	E-T350G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	E (2)
CIN2/3-1503	E-T350G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	E (2)
CIN2/3-1801	E-T350G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	E (2)
CIN2/3-2227	E-T350G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	E (2)
HSIL-75857	E-T350G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	T	-	-	-	-	-	-	E (2)
MRI-H196	E-T350G	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	G	-	-	-	-	-	-	-	-	E (2)
SiHa	E-T350G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	C	-	-	-	E (1 or 2)
CaSki	E-T350G	-	-	-	G	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	E (4)
CA-07C381	E-T350G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	not possible
LSIL-75022	E-T350G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	E (1 - 4)
CIN2/3-3009	NA1	-	T	-	-	-	T	-	-	-	-	A	G	-	T	G	-	-	-	-	-	-	-	-	-	-	(similar to AA)
CIN2/3-3035	NA1	-	-	-	-	-	T	-	-	-	-	A	G	-	T	G	-	-	-	-	-	-	-	-	-	-	(similar to AA)
CIN2/3-4242	NA1	-	-	-	-	-	T	-	-	-	-	A	G	-	T	G	-	-	-	-	-	-	-	-	-	-	(similar to AA)
CIN2/3-0005	Af1	-	-	-	G	-	G	T	-	-	-	-	A	G	G	T	G	-	-	-	-	-	-	-	-	-	Af1 or Af2
CIN2/3-1511	Af1	-	-	-	G	-	G	T	-	-	-	-	A	G	G	T	G	-	-	-	-	-	-	-	-	-	Af1 or Af2
CIN2/3-2219	Af1	C	-	-	-	C	G	T	-	-	-	-	A	G	-	T	-	-	-	-	-	-	-	-	-	-	Af1
CIN2/3-2237	Af2	-	-	C	-	T	G	T	-	-	-	-	A	G	-	T	-	-	-	G	-	-	-	-	-	-	Af2

(a)-(e) See footnotes of Table 2.29. (b) additional references: Ref 11 (Yamada et al, 1995), and Ref 12 (Eriksson et al, 1999). (f) Because the early promoter is located at nucleotide 97, ORF E6 is translated from the second ATG codon (nucleotide 104). (§) Referred to Table 2.29.

Table 2.31: Nucleotide polymorphisms in HPV16 ORF E1 of 25 DNA samples in comparison with E2 and E6.



(d)-(e) See Table 2.29 for further explanation. (f) The protein of ORF E6 is translated from nucleotide 104. See Table 2.30 for further explanation. (§§) Referred to Table 2.30. Green highlight indicates common mutations within the same pattern group, as defined in the last column. Gray highlight indicates mutations not present in the reference patterns shown in Table 2.29 and Table 2.30.

Table 2.32: Amino acid residue variations in HPV16 E6, E1 and E2 proteins in 25 DNA samples.

aa pos.	E6 (158 residues)								
	E6 No. of silent mutations (§)	E6 No. of aa changes	10	14	64	78	83	95	113
HPV16R			R	Q	D	H	L	P	E
CIN2/3-0004	0	0							
CA-07C368	0	0							
MRI-H186	0	0							
CIN2/3-2229	0	0							
CIN2/3-4238a	0	0							
HSIL-66019	0	0							
HSIL-61979	0	0							
CIN2/3-0001	0	1					V		
CIN2/3-0002	0	1					V		
CIN2/3-1503	0	1					V		
CIN2/3-1801	0	1					V		
CIN2/3-2227	0	1					V		
HSIL-75857	0	2					V	L	
MRI-H196	1	1					V		
SiHa	0	2					V		D
CaSki	0	2	G				V		
CA-07C381	0	1					V		
LSIL-75022	0	1					V		
CIN2/3-3009	2	4		H		Y	V		
CIN2/3-3035	2	3		H		Y	V		
CIN2/3-4242	2	3		H		Y	V		
CIN2/3-0005	2	5	G	D	E	Y	V		
CIN2/3-1511	2	5	G	D	E	Y	V		
CIN2/3-2219	2	4	T	D		Y			
CIN2/3-2237	4	3	I	D		Y			

aa pos.	E1 (residues 43-645)																												
	E1 No. of silent mutations (§)	E1 No. of aa changes	63	70	78	80	100	117	130	144	145	168	185	188	192	202	220	260	294	326	337	355	441	452	458	462	546	551	598
HPV16R			E	L	Q	R	G	K	E	M	L	S	T	Q	T	N	S	L	L	I	P	F	Q	E	T	R	S	H	G
CIN2/3-0004	0	0										#				#							#						
CA-07C368	0	0										#										#	#						
MRI-H186	1	0																											
CIN2/3-2229	1	1															T												
CIN2/3-4238a	1	1										#			#	#	T						#						
HSIL-66019	0	3								V		#				K	T	#				#							
HSIL-61979	0	1		#					#	#	#	#				#	T		#			#						#	#
CIN2/3-0001	0	1														#	T												
CIN2/3-0002	0	1														#	T												
CIN2/3-1503	0	4															T				L		#				F	R	
CIN2/3-1801	1	3	D													#		V											R
CIN2/3-2227	0	0														#													
HSIL-75857	0	1		#								#				#	T						#						
MRI-H196	1	2								I	V	#																	
SiHa	2	1																		M									
CaSki	1	3							K				N				T						#						
CA-07C381	0	1								#		#											R						
LSIL-75022	0	0		#					#	#	#	#	#	#	#	#				#	#	#	#	#	#	#	#	#	#
CIN2/3-3009	12	10			E		E					S		E		#	T		M	M				D	S	K			
CIN2/3-3035	12	11			E		E					S		E		#	T		M	M		L		D	S	K			
CIN2/3-4242	13	10			E		E					S		E			T		M	M		#		D	S	K			
CIN2/3-0005	9	5			E	T				#		S				#				M			#	D					
CIN2/3-1511	10	5			E	T						S				#				M			#	D					
CIN2/3-2219	9	6		V	E			N		#		S			I	#				M									
CIN2/3-2237	11	8			E		E					#		E		#			M	M				D	S	K			

aa pos.	E2 (365 residues)																															Patterns assigned by E6 (§§)
	E2 No. of silent mutations (§)	E2 No. of aa changes	35	105	135	136	143	144	157	165	188	203	204	208	210	219	223	224	225	226	232	239	246	254	271	292	310	314	344	345	364	
HPV16R			H	A	T	H	A	S	L	R	A	N	E	P	I	P	H	T	K	A	E	Q	D	T	F	K	T	W	D	Q	S	
CIN2/3-0004	1	0	#																								#					Ep
CA-07C368	2	0	#																								#					Ep
MRI-H186	2	0																														Ep
CIN2/3-2229	2	2													T	S											#					Ep
CIN2/3-4238a	4	2	#								#				T	S						#					#					Ep
HSIL-66019	0	0	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	Ep
HSIL-61979	0	0	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	Ep
CIN2/3-0001	1	1	#													S																E-T350G
CIN2/3-0002	1	1														S											#					E-T350G
CIN2/3-1503	1	1														S											#					E-T350G
CIN2/3-1801	1	2											K			S									#							E-T350G
CIN2/3-2227	1	2														S			Q													E-T350G
HSIL-75857	1	3	#													S	Y					#					#			H		E-T350G
MRI-H196	1	2														S										Q						E-T350G
SiHa	1	2		T	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#		K					E-T350G
CaSki	2	3													T	S											K					E-T350G
CA-07C381	0	0	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	E-T350G
LSIL-75022	0	1	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	N			#	#	#	#	#	#	E-T350G
CIN2/3-3009	5	13	Q		R	Y	T		I	Q		D		A							K			N	#		K	C	E			NA1
CIN2/3-3035	3	12	Q		R	Y	T		I	Q		D		A							K			N	#	#	#	C	E			NA1
CIN2/3-4242	5	14	Q		R	Y	T		I	Q		D		A							K	#		N	V		K	C	E			NA1
CIN2/3-0005	4	16	Q		K	Y	T	S		Q		D		A		S					K			N	V		K	C	E		C	Af1
CIN2/3-1511	4	15	Q		K	Y	T	S		Q		D		A		S					K			N	#		K	C	E		C	Af1
CIN2/3-2219	3	13	Q		K	Y	#	#		Q		D		A		S		S			K			N	V		#	C	E			Af1
CIN2/3-2237	4	14	#		K	Y	#	#		Q	T	D		A		S				T	K	L		N	V	#	#	C	E			Af2

(§) The positions of the silent mutations are given in Table 2.29 panel B, Table 2.30 panel B and Table 2.31.
(§§) Referred to Table 2.30. "#" indicates no sequence information.

2.2.2.10 Summary of ASP16 results for HPV16 integration and mutation analysis

The integration junctions identified in ASP16-3 and ASP16-4, the sequence coverage and the assigned variants/lineages of each sample are shown in Table 2.33. All four cell lines belong to the European lineage. The 21 clinical DNA samples were obtained from 20 women, with samples HSIL-66019 and HSIL-61979 collected from the same woman. Among these 20 individuals, 13 contain HPV16 variants of European lineage, 4 of African and 3 of North-American.

Table 2.33: Summary of the integration and sequence analyses of ASP16-3 and ASP16-4.

Sample name [(a)]	Integration %	HPV16 breakpoint	Chromosome - position - strand - NC.version	HPV16 variant [(b)]	Sequence coverage
Cell lines					
MRI-H186	**49**	**2754**	**8 - 128746606 - plus - NC_000008.10**	**Ep**	**96%**
MRI-H196	**32**	**3858**	**11 - 47967861 - plus - NC_000011.9**	**E-T350G**	**95%**
SiHa		**3133**	**13 - 74087558 - minus - NC_000013.10**	**E-T350G**	**90%**
CaSki				**E-T350G**	**94%**
*Clinical samples for which HPV16 integration breakpoints were identified by ASP16 **					
CA-07C381	**100**	**2783**	**12 - 57686270 - plus - NC000012.11**	**E-T350G**	**59%**
CA-07C368	**84**	**1910**	**1 - 240760698 - plus - NC000001.10**	**Ep**	**82%**
HSIL-66019	**100**	**2516**	**7 - 72454568 - plus - NC_000007.13, or 7 - 72818793 - plus - NC_000007.13, or 7 - 75010338 - minus - NC_000007.13**	**Ep**	**30%**
HSIL-61979	**100**				**28%**
HSIL-75857	**100**	**1149**	**6 - 135721233 - plus - NC_000006.11**	**E-T350G**	**86%**
CIN2/3-1801	**72**	**1913**	**19 - 1494405 - minus - NC_000019.9**	**E-T350G**	**92%**
*Clinical samples for which HPV16 integration breakpoints were not identified by ASP16 ***					
CIN2/3-0004	**82**			**Ep**	**90%**
CIN2/3-2229	**76**			**Ep**	**100%**
CIN2/3-4238a	**64**			**Ep**	**79%**
CIN2/3-0001	**76**			**E-T350G**	**93%**
CIN2/3-0002	**74**			**E-T350G**	**96%**
CIN2/3-1503	**74**		**false-positive**	**E-T350G**	**94%**
CIN2/3-2227	**99**			**E-T350G**	**97%**
LSIL-75022	**100**			**E-T350G**	**20%**
CIN2/3-3009	**81**			**NA1**	**95%**
CIN2/3-3035	**74**			**NA1**	**95%**
CIN2/3-4242	**79**		**false-positive**	**NA1**	**94%**
CIN2/3-0005	**95**			**Af1**	**90%**
CIN2/3-1511	**87**			**Af1**	**98%**
CIN2/3-2219	**72**			**Af1**	**81%**
CIN2/3-2237	**98**			**Af2**	**77%**

(a) The sample names include the histological/cytological status as a prefix. CA: cervical carcinoma CIN: cervical intraepithelial neoplasia. LSIL/HSIL: low/high-grade squamous intraepithelial lesion.
(b) The variant assignment is based on E6 polymorphisms. See also Table 2.30.
* Sorted by histological/cytological status.
** Sorted by lineage/variants.

3. Discussion

3.1 Complete sequence of integrated HPV68b in ME180 and ME180R

The complete sequences of the integrated HPV68b DNA in cell lines ME180 and ME180R have been determined in this study. In comparison to the previous sequence of cloned restriction fragment AA13.1 (Reuter et al, 1991), the newly determined HPV68b(int)-ME180 is about 6.1 kb larger. The size difference corresponds to the probably cloning-induced rearrangement of AA13.1 where almost the complete 5'copy was deleted (Figure 2.3). The new sequence HPV68b(int)-ME180 contains two incomplete HPV68b copies, integrated in head-to-head arrangement. In cervical carcinoma cell lines, HPVs were found to be integrated as a single incomplete copy such as in SiHa (Meissner, 1999), or as a disrupted copy flanking an additional complete copy as in MRI-H186 (Xu, 2010) (see Figure 2.41) or flanking multiple complete copies as in CaSki (Baker et al, 1987; Callahan et al, 1992). These multiple integrated HPV copies were found in tandem head-to-tail arrangement. To our knowledge, the integrated HPV68b in ME180 is the only known case of head-to-head arrangement of integrated HPV in cervical carcinoma cell lines. Because the URR, and ORFs E6 and E7 in both 5'copy and 3'copy are intact, the viral oncogenes E6/E7 can be expressed from both copies.

In the sub-line ME180R which had been selected for resistance to growth-inhibition by TNFalpha (Pfreundschuh et al, 1989), the sequence HPV68b(int)-ME180R contains two large deletions, compared to ME180 (Figure 2.6). As the result, only URR, and ORFs E6 and E7 in the 5'copy are intact. It was investigated in this study whether alterations of the integrated HPV68b DNA are a recurrent phenomenon associated with selection of ME180 cells for resistance to TNFalpha. Two new TNFalpha-resistant variants, ME180-2A and ME180-3A, were selected which showed no alteration in the integrated HPV68b DNA. Thus, it was concluded that the two large deletions in ME180R are not necessary for TNFalpha resistance phenotype. Other cellular variations may contribute to this phenotype (Manchester et al, 1993; Nishikawa et al, 1992).

3.2 Full-length HPV68b genomes

In this study, a full-length (complete) HPV68b genome, HPV68b-CIN2, and a partially deleted genome, HPV68b-CIN2-Del, have been isolated from a CIN2 DNA sample. Compared to the full-length genome HPV68b-CIN2, the HPV68b-CIN2-Del genome carries a 1229-bp deletion in ORF E1, resulting in a premature stop codon of E1 immediately downstream of the deletion site (Figure 2.15). Since functional E1 protein is necessary for viral episomal genome maintenance but unnecessary for integrated HPV, it was hypothesized that the HPV68b-CIN2-Del genome was likely integrated rather than episomal. This is supported by genomic Southern blot analysis, which indicated a probably integrated status. Because no additional CIN2 genomic DNA for further studies was available, it was impossible to proof this assumption and to determine whether the full-length HPV68b-CIN2 genome is integrated or episomal. Furthermore, it remained unclear whether the two genomes were present in the same cell or originated from different cells of the CIN2 sample.

The presence of intact genome and E1-disrupted genome of HPV62 in a single cervical sample of normal cytology had been reported (Fu et al, 2004), where a single nucleotide in the 5' region of E1 was deleted in the disrupted genome resulting in a premature stop codon. Single nucleotide deletions in E1 ORF that caused premature stop codons were also detected in the reference genome sequences of HPV16, HPV53, HPV56, and HPV72 (Fu et al, 2004). Although the HPV68b-CIN2-Del carries a much larger deletion in ORF E1, it resulted in the same consequence of a premature stop codon. Altogether, these data show that the disruption in ORF E1 maybe rather common among genital HPVs.

The nucleotide sequence of HPV68b-CIN2 shows 99% similarity to HPV68b-ME180, 93% similarity to HPV68a subtype (accession DQ080079), and 99% similarity to the recently published complete nucleotide sequence of the HPV68b variant HPV68b-EU918769 (Wu et al, 2009). The HPV68b-CIN2 and HPV68b-EU918769 genomes are the only two HPV68b variants whose complete nucleotide sequences are available up to now.

3.3 HPV68 subtypes and variants

In an effort to determine HPV68 subtypes/variants among the eleven HPV68-positive clinical samples analyzed in this study, the nucleotide sequences (491 bp) of URR have been determined and compared with other published sequences (Table 2.10). The phylogenetic tree, constructed from these URR sequences, including HPV68b-CIN2, HPV68b-ME180, HPV68b-EU918769 and other HPV68b variants, clearly shows two deep dichotomic branches for HPV68a and HPV68b subtypes (Figure 2.21).

Sample Reims-12 contains subtype HPV68a. The analyzed URR sequence of sample Reims-12 is identical to that of HPV68a-DQ080079. Since both HPV68a genomes were isolated from cervical lesions of French women, it is likely that these two HPV68a genomes belong to the same HPV68a variant present in France. DQ080079 is the only published HPV68a sequence so far (Longuet et al, 1996).

The 41 samples in the study of (Calleja-Macias et al, 2005) had been collected from Brazil, Mexico, the United States, South Africa, Hong Kong and Scotland, while all eleven samples analyzed in this study were collected in France. All 41 samples (100%) in the study of (Calleja-Macias et al, 2005) contain subtype HPV68b, whereas 10 out of 11 samples (91%) analyzed in this study contain subtype HPV68b. It is clear from these observations that subtype HPV68b is more widely distributed and more frequently found than subtype HPV68a. HPV68a may be more prevalent in France than in other countries. It will be interesting to analyze whether HPV68a and HPV68b differ in their propensity to establish persistent infections, as has been shown for HPV16 variants (Schiffman et al, 2010).

3.4 Optimized ASP16 strategy and computer analysis programs

The ASP16 strategy was developed for massively parallel sequencing of HPV16 DNA in clinical samples with the main purpose of determining HPV16 integration junctions (Xu, 2010). It employs one of the next generation sequencing technologies, Roche/454 GS-FLX pyrosequencing (http://www.454.com/).

Because of the high output volumes for each sequencing run by Roche/454 GS-FLX, computer analysis programs are required to assist in initial sequence data management. The data analysis softwares provided by the manufacturer of Roche/454 GS-FLX system, such as GS Reference Mapper Software or GS Amplicon Variant Analyzer Software, are not directly suitable for our purposes. Therefore, in this study, four sets of ASP16 analysis computer programs were specifically created and configured for basic analysis of HPV16 sequence reads from ASP16 experiments. The programs can deliver sorted and edited sequence alignments in FASTA format, according to their sample IDs and primer groups. The output alignments provide a direct and simple mean to investigate HPV16 sequences and to detect HPV16 integration junctions. Furthermore, the programs also deliver basic statistical data such as sequence read counts for each sorted category. The programs had been used successfully to analyze the sequence data in ASP16 experiments.

Four ASP16 experiments had been performed until now. The first two experiments, ASP16-1 and ASP16-2, were conducted by Bo Xu (Xu, 2010). It had been demonstrated that the ASP16 strategy could successfully identify HPV16 integration junctions in cell lines and clinical samples. The average sequence read lengths of these two experiments were 116 nt and 92 nt, respectively. In ASP16-2, 16 combinations of HPV16 primers were used, each located about 200 bp apart, resulting in only about 50% sequence coverage of the analyzed HPV16 E1-E2 region. This reduced the chance of finding integration junction by 50%. Therefore, to obtain better performance, the ASP16 strategy was further optimized during the two experiments ASP16-3 and ASP16-4, performed in this study. The HPV16 primer combinations were optimized to locate about 100 bp apart by increasing the number of primer combinations (Figure 2.33). The average sequence read lengths were increased from 92 nt in ASP16-2 to 105 nt and 108 nt in ASP16-3 and ASP16-4 (Table 2.15). Closer inspection into the sequence data revealed that more longer reads were obtained in ASP16-3 and ASP16-4 (Figure 2.36 and Figure 2.37). The reduction of distance between primers to 100 nt and the increase of average read length to 105-108 nt, together, contributed to 89% average sequence coverage in both experiments.

The previously determined HPV16 integration junctions of cell lines MRI-H186, MRI-H196 and SiHa were identified in the sequence reads of ASP16-3 and ASP16-4 (Table 2.18). In ASP16-2, it was not possible to identify the junctions in SiHa due to unsuitable primer locations, and the identified junction in MRI-H196 was too short to be mapped to

chromosome 11 by Blast (Xu, 2010). The verified HPV16 integration junctions in two cervical carcinoma samples CA-07C381 and CA-07C368, previously analyzed in ASP16-2, were redetected in ASP16-4. Altogether, these results demonstrate higher efficiency of the optimized ASP16 strategy in identifying HPV16 integration junctions. Cell line CaSki was also analyzed in ASP16-4. It is known to contain ~600 integrated HPV16 copies in tandem repeats, but only one 3'junction. No HPV16 integration junction could be detected in any sequence reads of CaSki in ASP16-4. This indicates that the ASP16 strategy does not allow detection of an HPV16 integration junction in a background of about 600 full-length HPV16 genomes. This might become possible if the number of sequence reads per primer is increased by two or three folds.

In ASP16-3, false-positive HPV16 integration junctions were identified in 1 and 2 sequence reads of samples CIN2/3-4242 and CIN2/3-1503, respectively. False-positive integration junctions were also identified in 12 clinical samples analyzed in ASP16-1 and ASP16-2 (Xu, 2010). These false-positive sequence reads are artifacts that may be produced during the multiple amplification steps for preparation of HPV16 amplicons. However, the number of false viral-cellular sequence reads in ASP16-3 and ASP16-4 is very low, and has decreased from the first two ASP16 experiments, possibly by optimization of the ASP16 conditions. This improvement indicates that the optimized ASP16 strategy can deliver highly specific sequence reads.

Among 25 DNA samples analyzed in ASP16-3 and ASP16-4, genuine HPV16 integration junctions have been identified in 3 out of 4 cell lines (75%) and in 6 out of 21 clinical samples (29%) (Table 2.33). All clinical samples were expected to contain integrated HPV16 DNA based on their percent integration values determined by E2/E6 qPCR. However, integration junctions could only be identified in 29% of these samples. This low determination rate might be due to a low efficiency of the ASP16 strategy, but other reasons are more likely. Closer inspection of the 21 clinical samples showed that the integration junctions were identified in 6 out of 14 samples of European lineage (43%), 0 out of 3 samples of NA1 lineage, and 0 out of 4 samples of Af1/Af2 lineages. It had been demonstrated that nucleotide variations in non-European lineages at the binding regions for E2/E6 qPCR primers and probes could result in faulty estimation of HPV16 genome status (Jiang et al, 2009). Thus, the percent integration values determined by E2/E6 qPCR of the 7 clinical samples of NA1 and Af1/Af2 lineages may be too high misrepresenting

HPV16 integration. Several studies have been performed in the past aimed at determining the proportion of CIN2/3 lesions and carcinomas harboring integrated HPV16 DNA (Briolat et al, 2007; Hudelist et al, 2004; Li et al, 2008; Saunier et al, 2008; Woodman et al, 2007). The results were different, ranging from about 20% to almost 80%, giving an average value of about 50%. The proportion of samples with integrated HPV16 DNA in our collection, 29% for all and 43% for the European lineage, fits perfectly into this range. Therefore, it seems possible that the samples, for which ASP16 failed to identify any integration junction, indeed harbor episomal HPV16 DNA.

Disruption or deletion of E2 caused by HPV16 DNA integration is thought to be a central step in cervical cancer development. Nevertheless, it is evident that some cervical carcinomas definitely contain only episomal HPV16 DNA (Hudelist et al, 2004; Park et al, 1997; Vinokurova et al, 2008). It is not yet clear whether two pathways of HPV16-induced carcinogenesis exist, one with HPV16 DNA integration as an essential step, and the other for which the integration is not obligatory because E2 inactivation is achieved by other means. It was suggested that the E2 variations of AA lineage where E2 is retained may be an alternative mechanism for increasing E6/E7 oncogene expression (Casas et al, 1999). This is supported by another study in which E2 of AA lineage was found to be impaired for repression of E6/E7 oncogene transcription (Ordonez et al, 2004). Because the nucleotide variations in ORF E2 of NA1, Af1 and Af2 lineages are highly similar to that of AA lineage (Table 2.29 and 2.31), it is likely that the E2 of these three lineages may possess similar properties as the AA-E2, regarding the regulation of E6/E7 expression.

Overall, these results demonstrate that the optimized ASP16 strategy has the potential to analyze series of HPV16-positive clinical samples in parallel, detect HPV16 integration sites, and provide high coverage of HPV16 E1-E2 nucleotide sequences. The sequences of E1-E2 region can be used to identify mutations and assign HPV16 variants as demonstrated in this study. The integration junction analysis of the four cell lines shows that the ASP16 has a great potential to identify existent HPV16 integration junctions with the exception of samples containing a high background of full-length HPV16. The ASP16 strategy is the first method that combines the next generation sequencing technologies with HPV sequence analysis. In future development of the ASP16 strategy, it is planned to apply this strategy to other hr-HPV types, in particular HPV18 which is detected as

integrated DNA in almost 100% cervical carcinomas (Hudelist et al, 2004; Vinokurova et al, 2008). Furthermore, the Roche/454 GS-FLX will be replaced by Illumina/Solexa sequencing system, which probably has the advantage in delivering higher output volumes (~600 folds) with comparable costs. The increased output volumes will allow more samples to be analyzed per sequencing round, e.g. increasing from 20 samples to 40-50 samples, and possibly will allow the detection of integration junction in samples with high full-length HPV backgrounds, such as CaSki.

3.5 Possible consequences of HPV16 integration in samples HSIL-75857 and CIN2/3-1801

The integration of hr-HPV DNA into cellular DNA results in stable expression of viral oncogenes E6/E7 (Jeon & Lambert, 1995), which could result in transformation of the infected cells and progression into cancer (Munger et al, 1989). Furthermore, the integrated hr-HPV DNA may activate cellular proto-oncogenes or inactivate cellular tumor suppressor genes via insertional mutagenesis as suggested and demonstrated in earlier studies (Couturier et al, 1991; Peter et al, 2006; Reuter et al, 1998; Xu, 2010). When HPV DNA is integrated upstream or within and in the same orientation as a cellular gene, viral-cellular fusion transcripts can be produced which are initiated at the early viral promoter and contain the viral E6/E7 oncogenes followed by complete or truncated cellular genes (Kraus et al, 2008; Reuter et al, 1998; Wentzensen et al, 2004; Xu, 2010; Ziegert et al, 2003). The flanking cellular DNA may have an enhancer effect on viral oncogene expression, and vice versa, the HPV enhancer elements can lead to activation of cellular genes in the integration locus (von Knebel Doeberitz et al, 1991).

In sample HSIL-75857, HPV16 was found integrated into chromosome 6 in an intron of the AHI1 gene in opposite direction, and in proximity to two other cancer-relevant cellular genes, the MYB proto-oncogene and the PDE7B gene (Figure 2.53). AHI1 was suggested to be involved in tumor formation by interaction with other oncogenes such as c-myc or the tumor suppressor gene NF1 (Jiang et al, 2002). Mutations of AHI1 were reported to cause Joubert syndrome (JS) related disorders (Valente et al, 2006). The integrated HPV16 DNA may activate AHI1 gene expression. The presence of the integrated HPV16 DNA may also interfere with correct splicing of AHI1 transcripts.

MYB is a cellular proto-oncogene that can be activated by several mechanisms, including overexpression (Ramsay & Gonda, 2008). MYB was found to be overexpressed in cervical cancers and can transactivate HPV16 E6/E7 oncoprotein expression (Nurnberg et al, 1995). Integration of HPV16 DNA in the vicinity of the c-myc proto-oncogene has been shown to increase myc expression (Couturier et al, 1991; Peter et al, 2006; Xu, 2010). Thus, it is possible that the integrated HPV16 DNA, located about 181 kb downstream of myb gene, could activate the myb proto-oncogene expression, and thereby, contributing to carcinogenesis. The integrated HPV16 DNA is also located ~451 kb upstream of the cellular PDE7B gene. PDE7B was reported to be upregulated in chronic lymphocytic leukemia (CLL) (Zhang et al, 2008). Despite the relatively long distance, the integrated HPV16 DNA may affect also PDE7B expression.

In sample CIN2/3-1801, HPV16 was found integrated into chromosome 19 into an intron of the cellular REEP6 gene in opposite direction, and in proximity of two other cancer-relevant cellular genes, the APC2 and PCSK4 genes (Figure 2.54). REEP6 was proposed to be a tumor suppressor gene, and its polymorphisms are associated with colon cancer and inflammatory bowel disease (IBD) (Wellmann et al). As in the case of AHI1 gene described above, the integrated HPV16 may interfere with REEP6 expression. In addition, flanking cellular DNA may affect viral E6/E7 oncogene expression. PCSK4 is a multifunctional protein, which can activate p53 function (Batta & Kundu, 2007) as well as nonhomologous end joining and double strand break (DSB) repair activity (Batta et al, 2009). Since the integrated HPV16 DNA is located just 3998 bp upstream of PCSK4 gene in the same orientation, it is possible that viral-cellular fusion transcripts may be produced. APC2 is homologous to the tumor suppressor gene APC (van Es et al, 1999), which is associated with colon cancer (Munemitsu et al, 1995), and therefore may share functions with APC. APC2 was also suggested to be a potential tumor suppressor in ovarian cancer (Jarrett et al, 2001). The integrated HPV16 DNA may act as cis-acting regulatory sequence.

The HPV16 integration sites of both samples HSIL-75856 and CIN2/3-1801 locate in the vicinity of cellular proto-oncogenes or tumor suppressor genes. The hypothesized consequences of HPV16 integration upon the cellular genes can be investigated on RNA and protein levels, if the respective materials are available from the clinical samples. The HPV16 integration sites of the two samples support the assumption that HPV integration

is important for cervical carcinomas not only by deregulating/activating E6/E7 oncogene expression, but also by altering cancer-relevant cellular genes.

Up to the present, more than 200 HPV integration loci had been identified (Kraus et al, 2008; Wentzensen et al, 2004; Yu et al, 2005). However, the consequences of viral integration have not been investigated for these identified integration loci. With the vast resources of human genome sequences, genes and proteins now available through many database providers, such as NCBI, it is possible to create computer algorithms that can predict the outcomes of HPV integrations. Many tools are already available, such as gene structure and alternative splicing prediction (Coward et al, 2002), regulatory element prediction (Robertson et al, 2006) and protein interaction prediction (Jansen et al, 2003). In combining several available tools, different aspects of the effects of HPV integration could be virtually investigated.

4. Materials and methods

4.1 Chemicals, commercial media/solutions and antibiotics

Acetic acid, glacial	Sigma Aldrich, Munich
Agarose	Sigma Aldrich, Munich
Ampicillin, sodium salt, powder	AppliChem, Darmstadt
Bacto agar	Becton Dickinson, Heidelberg
Bovine serum albumin (BSA)	New England Biolabs, USA
Bromophenol blue, sodium salt	Serva, Heidelberg
Chloroform	Roth, Karlsruhe
deoxyadenosine triphosphate (dATP)	Roche Applied Science, Mannheim
deoxycytidine triphosphate (dCTP)	Roche Applied Science, Mannheim
deoxyguanosine triphosphate (dGTP)	Roche Applied Science, Mannheim
deoxythimidine triphosphate (dTTP)	Roche Applied Science, Mannheim
Diethylpyrocarbonate (DEPC)	AppliChem, Darmstadt
Dulbecco's Modified Eagle's Medium (DMEM)	Sigma Aldrich, Munich
Dimethyl sulfoxide (DMSO)	AppliChem, Darmstadt
Ethylenediaminetetraacetic acid (EDTA), disodium salt dihydrate	Sigma Aldrich, Munich
Ethanol, absolute	Sigma Aldrich, Munich
Fetal bovine serum (FBS), superior	Biochrom, Berlin
Ficoll 400	Serva, Heidelber
Formaldehyde	Merck, Darmstadt
Formamide	Merck, Darmstadt
Glucose	Sigma Aldrich, Munich
Glycerol	Sigma Aldrich, Munich
Hydrochloric acid (HCl), 37% (w/v)	Sigma Aldrich, Munich
Isoamyl alcohol	Sigma Aldrich, Munich
Isopropanol	Sigma Aldrich, Munich
Kanamycin sulfate, powder	AppliChem, Darmstadt
Magnesium chloride ($MgCl_2$)	Sigma Aldrich, Munich
Magnesium sulfate ($MgSO_4$)	Sigma Aldrich, Munich
NEN [α-^{32}P] dCTP	PerkinElmer, Rodgau-Jügesheim
Penicillin/streptomycin solution	Invitrogen, Karlsruhe
Phenol	Roth, Karlsruhe
Polyvinylpyrrolidone	Sigma Aldrich, Munich

Potassium chloride (KCl), powder	Sigma Aldrich, Munich
Potassium phosphate monobasic (KH_2PO_4)	Sigma Aldrich, Munich
Random hexamer - 5'NNNN*N*N3' (phosphothioate modified)	Thermo Fisher Scientific, Ulm
Redivue [α-^{32}P] dCTP	GE Healthcare, Munich
Sarkosyl, sodium salt	Sigma Aldrich, Munich
Sodium acetate, powder	Sigma Aldrich, Munich
Sodium chloride (NaCl), powder	Sigma Aldrich, Munich
Sodium citrate dihydrate	Sigma Aldrich, Munich
Sodium dodecyl sulfate (SDS)	Sigma Aldrich, Munich
Sodium hydroxide (NaOH)	Sigma Aldrich, Munich
Sodium phosphate dibasic (Na_2HPO_4)	Sigma Aldrich, Munich
t-RNA	Sigma Aldrich, Munich
Trizma base, powder	Sigma Aldrich, Munich
Trypsin/EDTA solution	Invitrogen, Karlsruhe
Tumor necrosis factor alpha (TNFalpha), recombinant	Strathmann Biotec, Hamburg
UltraPure water, DNase/RNase-free	Invitrogen, Karlsruhe
Tryptone	Becton Dickinson, Heidelberg
Yeast extract	Becton Dickinson, Heidelberg
Xylene cyanol FF, sodium salt	Serva, Heidelberg

4.2 Buffers, stock solutions and media

6xBPB DNA loading buffer

0.25% (w/v)	bromophenol blue
30% (v/v)	glycerol

6xXC DNA loading buffer

0.25% (w/v)	xylene cyanol
30% (v/v)	glycerol

50xTAE electrophoresis buffer (for 1 L)

242 g	Trizma
57.1 ml	glacial acetic acid
100 ml	0.5 M EDTA, pH 8.0

10xTE stock solution (for 1 L)

100 ml	1 M Tris-HCl, pH 8.0
20 ml	0.5 M EDTA, pH 8.0
880 ml	deionized water

1xTE/RNase buffer (for 100 ml)

10 ml	10xTE
1 ml	10 mg/ml RNase A
89 ml	deionized water

TEG buffer

50 mM	glucose (filtered)
10 mM	EDTA, pH 8.0
25 mM	Tris-HCl

NaOH/SDS solution

0.2 M	NaOH
1%	SDS

20xSSC stock solution, pH 7.0

3 M	NaCl
0.3 M	Sodium citrate

10xTNE stock solution

100 mM	Trizma base, pH 8.0
1 M	NaCl
100 mM	EDTA, pH 8.0
2%	SDS

50xDenhardt's solution

1% (w/v)	Ficoll 400
1% (w/v)	polyvinylpyrrolidone
1% (w/v)	BSA

Hybridization solution

50 mM	sodium phosphaste, pH 6.4
1%	SDS
1x	Denhardt's solution
5x	SSC
0.1 mg/ml	t-RNA

LB medium with selective antibiotics (for 1 L)

10 g	NaCl
10 g	tryptone
5 g	yeast extract

Adjusted to pH 7.0. After autoclaved, one of the following antibiotics was added:

	1 ml	100 mg/ml ampicillin
or	5 ml	10 mg/ml kanamycin

LB agar (1 L) with selective antibiotics

1 L	LB medium without antibiotics
14 g	bacto agar

After autoclaved, one of the following antibiotics was added:

	1 ml	100 mg/ml ampicillin
or	5 ml	10 mg/ml kanamycin

SOC medium (1 L)

0.5 g	NaCl
20 g	tryptone
5 g	yeast extract
20 ml	1 M glucose

Filter-sterilized.

10xPBS stock solution (1 L), pH 7.4

80 g	NaCl
2 g	KCl
14.4 g	Na_2HPO_4
2.4 g	KH_2PO_4

PBS/EDTA solution

1x	PBS
2 M	EDTA

3x lysis buffer

2x	TE
3% (w/v)	sarkosyl

Complete DMEM medium (for cell culture) with antibiotics

500 ml	DMEM
50 ml	FBS
5 ml	penicillin/streptomycin (50000 units each)

Cell freezing medium (for 10 ml)

8 ml	complete DMEM medium
1 ml	FBS
1 ml	DMSO

CIA (Chloroform:Isoamyl alcohol = 24:1)

24 part (v/v)	Chloroform
1 part (v/v)	Isoamyl alcohol

Phenol* (Phenol:CIA = 1:1)

1 part (v/v)	CIA
1 part (v/v)	Phenol

5xAnnealing buffer (for phi29 DNA polymerase)

85 mM	Tris-HCl, pH 8.0
85 mM	$MgCl_2$

4.3 Commercial kits and enzymes

Calf Intestine Alkaline Phosphatase (CIAP)	Fermentas Life Sciences, Canada
Expand long template PCR system	Roche Applied Science, Mannheim
GenomePlex WGA2 kit	Sigma Aldrich, Munich
Pfu DNA Polymerase	Promega, USA
Phi29 DNA Polymerase	Fermentas Life Sciences, Canada
Proteinase K	Merck, Darmstadt
Qiagen Multiplex PCR kit	Qiagen, Hilden
Qiagen Plasmid Maxi kit	Qiagen, Hilden
QIAquick Gel Extraction kit	Qiagen, Hilden
QIAquick PCR Purification kit	Qiagen, Hilden
QIAprep Spin Miniprep kit	Qiagen, Hilden
Rediprime II DNA Labeling System	GE Healthcare, Munich
RNase A	Qiagen, Hilden
StrataClone PCR Cloning kit	Stratagene, La Jolla, USA
StrataClone Blunt PCR Cloning kit	Stratagene, La Jolla, USA
T4 DNA Ligase (2000 U/ml)	New England Biolabs, USA
Taq DNA Polymerase kit	Invitrogen, Karlsruhe
TOPO TA PCR Cloning kit	Invitrogen, Karlsruhe
TOPO Blunt PCR Cloning kit	Invitrogen, Karlsruhe

4.4 Laboratory equipments and commercial materials

BioMag streptavidin beads	Polysciences, Heidelberg
Biomax MR film	GE Healthcare, Munich
E-Gel iBase Power System	Invitrogen, Karlsruhe
E-Gel Safe Imager Transilluminator	Invitrogen, Karlsruhe
E-Gel SizeSelect	Invitrogen, Karlsruhe
Hybond-N+ nylon membrane	GE Healthcare, Munich
MicroSpin G-25 column	GE Healthcare, Munich
Sterile filter (0.45 µm)	Millipore, Schwalbach/Ts.
Thermocycler PTC-200	MJ Research, Waltham, USA
Whatman 3MM paper	Schleicher & Schuell, Dassel

4.5 Computer software and Internet resources

BLAST version 2.2.17

ftp://ftp.ncbi.nlm.nih.gov/blast/executables/release/

BLAST executables are based on NCBI C Toolkit and consist of several stand-alone executable programs, including *blastall* and *formatdb*. In this study, *formatdb* was used to create sequence databases on the analyzing computer, and *blastall* was used for blasting sequence reads of ASP16 (as queries) to the sequences in the created databases (as subjects).

ClustalW version 2.0.10

http://www.clustal.org/

ClustalW is the command-line version of Clustal, a multiple sequence alignment program. It accepts and delivers several input/output sequence formats. FASTA format was used in this study for both inputs and outputs.

Geneious Pro version 4.8.5

http://www.geneious.com/

Geneious Pro is an integrated, cross-platform bioinformatics software suited for manipulating, finding, sharing, and exploring biological data. It was used mainly for sequence visualization and edition.

HUSAR: Heidelberg Unix Sequence Analysis Resources

http://genome.dkfz-heidelberg.de/

HUSAR is a program package providing access to more than 200 databases and offering more than 260 applications for DNA and protein analysis, including many widely known programs such as *clustal* (for sequence alignment), *bl2Seq* (for sequence comparison) and *primer* (for primer design).

NCBI: National Center for Biotechnology Information

http://www.ncbi.nlm.nih.gov/

NCBI contains various databases of biotechnology information, including nucleotide and protein sequence databases of different organisms. It was used for searching, comparing and retrieving sequences.

For nucleotide sequences of the twenty-four human chromosomes, the reference sequences with accession numbers NC_000001, NC_000002,…, NC_000024 are used in this work. Each accession number comes with a version number, identified with suffix "**.**(version-number)".

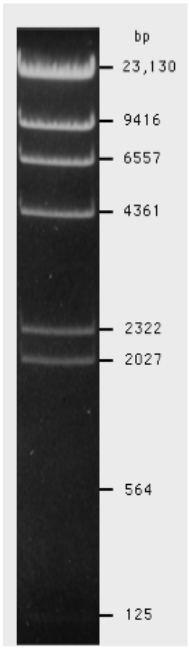
Perl application version 5.8.8

http://www.perl.org/

Perl version 5.8.8 is a program that reads a Perl language program, translates it into instructions the computer can understand, and runs (or executes) it. The developed ASP16 data analysis programs (see Results section 2.2.1) were executed in Perl 5.8.8 under Mac operating system.

4.6 DNA molecular markers

Lambda DNA/HindIII marker



This DNA ladder is frequently used when running gDNA because of its high molecular weight bands

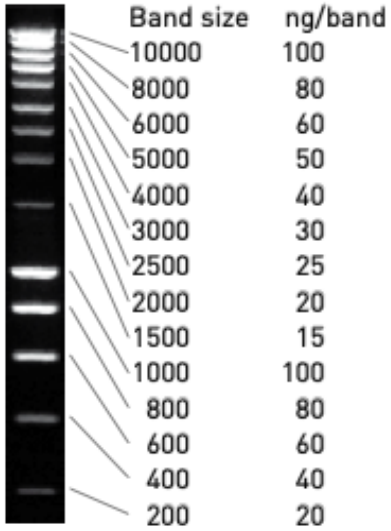
Practical size range: 2 – 23 kb

0.5 µg/lane

1.2% agarose gel, stained with ethidium bromide

(Invitrogen, Karlsruhe)

SmartLadder

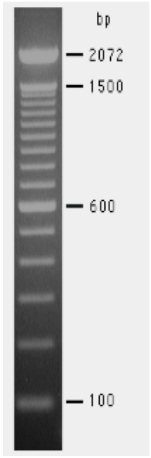


Size range: 200 bp – 10 kb

5 µl/lane (720 ng)

(Eurogentec, Köln)

100bp Ladder



Size range: 100 bp – 2 kb

0.5 µg/lane

2% agarose gel, stained with ethidium bromide

(Invitrogen, Karlsruhe)

4.7 Oligonucleotide primers

Oligonucleotide primers for routine PCRs were designed so that they have optimal melting temperature about 60°C, and are 18-24 nt in length.

4.7.1 Primers for cellular genes

The primers, shown in Table 4.1, were used to amplify parts of cellular genes after DNA preparation to determine the integrity of the DNA template. ß-globin and GAPDH are house-keeping genes, while c-myc is a proto-oncogene.

Table 4.1: Oligonucleotide primers for cellular genes.

Primer name*	Sequence (5' to 3')	Target cellular gene	Expected product size
ß-globin-F	ACACAACTGTGTTCACTAGC	ß-globin	110 bp
ß-globin-R	CAACTTCATCCACGTTCACC		
GAPDH-F	CACCACCAACTGCTTAGCAC	GAPDH	367 bp
GAPDH-R	GAGGCAGGGATGATGTTCTG		
c-myc-F	CTTTATAATGCGAGGGTCTGG	c-Myc	591 bp
c-myc-R	GCTTACCTGGTTTTCCACTACC		

* Binding directions are indicated. F: forward. R: reverse.

4.7.2 Primers for HPV68 analysis

The primers used for amplification of HPV68 DNA described in Results section 2.1 are shown in Table 4.2. The primers with names starting with "ME" were used mainly for amplification of the integrated HPV68b in cell lines ME180 and ME180R. The primers with names starting with "H68" were used mainly for analysis of HPV68b DNA in cervical samples.

Table 4.2: Oligonucleotide primers for HPV68.

Primer name	Sequence (5' to 3')	Primer binding sites *
ME-01 §	GGAATTATGCCTATTTGACA	pos. 45577909-F (NC_000018.9), upstream of the integrated HPV68b in ME180/ME180R
ME-04	TGGTATTTTTGGTGTGGTTTTGTG	pos. 3860-F of HPV68b-CIN2 genome
ME-874 §	ACACACCAATGGAGAGGAGTG	pos. 45578156-F of NC_000018.9, upstream of the integrated HPV68b in ME180/ME180R
ME-1349	TAGTCACAGGGTGCAACCAC	pos. 7208-R of HPV68b-CIN2 genome
ME-6651	TTTGTGTGTCCGTGGTGTG	pos. 782-F of HPV68b-CIN2 genome
ME-6805	CGCGACAGATACAGGTTCAG	pos. 940-F of HPV68b-CIN2 genome
ME-7094	CATTTTCCCCTTCATCCTCC	pos. 1289-R of HPV68b-CIN2 genome
ME-7718	ACCATTGGAGGTCTTTGCTG	pos. 3984-F of HPV68b-CIN2 genome
ME-7810	ACATGCACACACAAAACCAC	pos. 3892-R of HPV68b-CIN2 genome
ME-7972 §	TCTGGCTCAGGAAATCGC	pos. 45578405-R of NC_000018.9, downstream of the integrated HPV68b in ME180/ME180R
H68-128R	TCTGTCCGTTGTAGTTGCCTTC	Primer positions and binding directions on CIN2-HPV68b** genome are indicated in the primer names, after the prefix "H68-".
H68-377R	CTTCGTTTTGAATTTAGGTGCC	
H68-782F	TTTGTGTGTCCGTGGTGTG	
H68-830R	TGGCCATTGCAGATTACTGG	
H68-1107F	GCAGCCCTTTAGCAAAGTC	
H68-1175R	GTTGTCTTGCCTGTGTACTGC	
H68-1709F	CTTTTGCAGCCACCAAAATT	
H68-1728R	AATTTTGGTGGCTGCAAAAG	
H68-2915F	GATGGCACTAGAGAGCATTGC	
H68-2920F	GCCATCTGCAGTTCAATGG	
H68-4145F	AAGCGTGCATCTGCAACTG	
H68-4572R	ACTTGCACAGACCCAGACGAAG	
H68-5629F	GGTTATTAACTGTAGGCCATCC	
H68-5877F	AATAGGCTAGATGATACTGAG	
H68-5931R	CCTTAGGATTTTTGTTGGAGGAAAAC	
H68-6479R	ACAAATACCATTGTTGTGTCCC	
H68-7192F	GTTGCACCCTGTGACTAACATATG	
H68-7954F	TGCCTAATAGCATAGTTGGCC	
MY11	GCMCAGGGWCATAAYAATGG M=A/C, W=A/T, Y=C/T, R=A/G	pos. 6453-F of HPV68b-CIN2 genome

* Binding positions (5'end) and binding directions are indicated. F: forward. R: reverse.
** See Results section 2.1.4. The sequence of HPV68b-CIN2 is shown in Appendix A2.
§ Primers bind to human chromosome 18, flanking the integrated HPV68b in ME180 and ME180R.

4.7.3 Primers for HPV16 and HPV16 integration junctions

The primers used for amplification of HPV16 DNA and HPV16 integration junctions described in Results section 2.2 are listed in Table 4.3.

Table 4.3: Oligonucleotide primers for HPV16 and HPV16 integration junctions.

A: Primers specific for HPV16 DNA.

Primer name *	Sequence (5' to 3')	Remarks
H16-44F	GGTTGAACCGAAACCGGTTAG	
H16-691R	GTCCAGCTGGACCATCTATTTC	
H16-1064F	CACATGCGTTGTTTACTGCAC	binds at the same pos. as primer E09 (1064) **
H16-1105F	GATGCAGTACAGGTTCTAAAACG	binds at the same pos. as primer BE-19 **
H16-1115F	AGGTTCTAAAACGAAAGTATTTGG	binds at the same pos. as primer E19 (1115) **
H16-1656F	TCAAAGTTTAGCATGTTCATGGG	
H16-2174F	GTGATTGGAAGCAAATTGTTATG	binds at the same pos. as primer E12 (2174) **
H16-2403F	AGCAGATGCCAAAATAGGTATG	binds at the same pos. as primer E05 (2403) **
H16-3121F	TGGAGACATATGCAATACAATGC	binds at the same pos. as primer E07 (3121) **
H16-3339F	GTCCTACATCTGTGTTTAGCAGC	

* Binding positions (5'end) and binding directions are indicated after prefix "H16-". F: forward. R: reverse.
** Primers are shown in Table 4.4 and Table 4.5.

(Table 4.3, continued)

B: Primers specific for human chromosomes.

Primer name	Sequence (5' to 3')	Primer binding sites ***
NC06-R-1.1	AGAAAACTTCCCTAGTCTGAAATC	chrom.6 pos. 097896916-F (NC_000006.11)
NC06-R-1.2	TCAACTAGGATTACCTACCCTCC	chrom.6 pos. 097896741-F (NC_000006.11)
NC06-R-2.1	GTGTTTTTCAATTATTTGGAAAGG	chrom.6 pos. 135721349-R (NC_000006.11)
NC06-R-2.2	ATTGTAGTCGAATCCACCTCGG	chrom.6 pos. 135721494-R (NC_000006.11)
NC07-R-1.1	AGTGGTGTCCCCGCTGTAAG	chrom.7 pos. 72454604-R, 72818829-R and 75010302-F (NC_000007.13)
NC11-R-1.1	GAGGAGTGACTTTGAATAGAATGG	chrom.11 pos. 123658930-F (NC_000011.9)
NC11-R-1.2	AGTGGGGAGGTAGCTTCTGG	chrom.11 pos. 123658817-F (NC_000011.9)
NC19-R-1.1	GGGGCTGATGGCATGGAGTG	chrom.19 pos. 01494382-F (NC_000019.9)
NC19-R-1.2	GTTTGCCTCCAAGACAG	chrom.19 pos. 01494273-F (NC_000019.9)

*** Binding positions (5'end) and binding directions are indicated. F: forward. R: reverse.

4.7.4 HPV16 primers for ASP16 strategy

In the ASP16 strategy, two sets of HPV16 forward primers and one set of reverse primers are used in the HPV16 DNA enrichment step (see section 4.22.2). The first set of HPV16 forward primers are shown in Table 4.4. These primers are labeled with biotin at the 5'end, and were used in linear amplification step. The second set of HPV16 forward primers are shown in Table 4.5. They are bipartite nested HPV16 primers that contain the 19-nt Roche-A sequence (RA) at the 5'end. The reverse primers, shown in Table 4.6, are tripartite. They contain the 19-nt Roche-B sequence (RB), followed by a 4-nt barcode and the 18-nt GenomePlex universal adapter sequence (GPUA). These reverse primers were used in combination with the bipartite nested HPV16 primers for the nested HPV16 PCR, following the linear amplification step in ASP16 strategy.

Table 4.4: Biotin-labelled HPV16 forward primers for linear amplification in ASP16 strategy.

Short name	Full name*	Sequence (5' to 3')
BE-01	5B-810F	AATGGGCACACTAGGAATTGTG
BE-02	5B-1261F	GGGTATGGCAATACTGAAGTGG
BE-03	5B-1562F	GTTGCGATTGGTGTATTGCTG
BE-04	5B-1938F	ACAGATGGTACAATGGGCCTAC
BE-05	5B-2389F	TGGTTACAACCATTAGCAGATGC
BE-06	5B-2705F	CAAGGACGTGGTCCAGATTAAG
BE-07	5B-3101F	ACAGTGGAAGTGCAGTTTGATG
BE-08	5B-3542F	GACAGTGCTCCAATCCTCACTG
BE-09	5B-1046F	AGGCAGAAACAGAGACAGCAC
BE-10	5B-1394F	GAGAGGGTGTTAGTGAAAGACAC
BE-11	5B-1760F	GTGTGTCTCCAATGTGTATGATG
BE-12	5B-2151F	ATGTGATAGGGTAGATGATGGAG
BE-13	5B-2539F	TGCCCTCCATTATTAATTACATC
BE-14	5B-2912F	CATATTAACCACCAAGTGGTGC
BE-15	5B-3318F	CGGGTGGTCAGGTAATATTATG
BE-16	5B-3762F	CATATGATAGTGAATGGCAACG
BE-17	5B-790F	CGTACTTTGGAAGACCTGTTAATG
BE-18	5B-952F	GGGGATGCTATATCAGATGACG
BE-19	5B-1105F	GATGCAGTACAGGTTCTAAAACG
BE-20	5B-1336F	AGTCAGTATAGTGGTGGAAGTGG
BE-21	5B-1457F	TAAAAACTAGTAATGCAAAGGCAG
BE-22	5B-1653F	CATTCAAAGTTTAGCATGTTCATG
BE-23	5B-1849F	ATTAGTGAAGTGTATGGAGACACG

(Table 4.4, continued)

Short name	Full name*	Sequence (5' to 3')
BE-24	5B-2053F	TCACAGGCAAAAATTGTAAAGG
BE-25	5B-2277F	TTGCATATTACTATATGGTGCAGC
BE-26	5B-2461F	GATGACAATTTAAGAAATGCATTG
BE-27	5B-2613F	GGTGGTGTTTACATTTCCTAATG
BE-28	5B-2842F	TATAGACTATTGGAAACACATGCG
BE-29	5B-3021F	ATGAAAAGTGGACATTACAAGACG
BE-30	5B-3189F	TAACTGTGGTAGAGGGTCAAGTTG
BE-31	5B-3444F	GCACCGAAGAAACACAGACG
BE-32	5B-3680F	TGTACATTGTATACTGCAGTGTCG

* Binding positions (5'end) and binding direction are indicated after prefix "5B-". F: forward.

Table 4.5: Nested HPV16 forward primers for semi-nested PCR in ASP16 strategy.

Short name	Full name*	Sequence (5' to 3')
E01	RA-857F	GCCTCCCTCGCGCCATCAG AATCTACCATGGCTGATCCTG
E02	RA-1275F	GCCTCCCTCGCGCCATCAG TGAAGTGGAAACTCAGCAGATG
E03	RA-1576F	GCCTCCCTCGCGCCATCAG ATTGCTGCATTTGGACTTACAC
E04	RA-1951F	GCCTCCCTCGCGCCATCAG TGGGCCTACGATAATGACATAG
E05	RA-2403F	GCCTCCCTCGCGCCATCAG AGCAGATGCCAAAATAGGTATG
E06	RA-2723F	GCCTCCCTCGCGCCATCAG TAAGTTTGCACGAGGACGAG
E07	RA-3121F	GCCTCCCTCGCGCCATCAG TGGAGACATATGCAATACAATGC
E08	RA-3555F	GCCTCCCTCGCGCCATCAG TCCTCACTGCATTTAACAGCTC
E09	RA-1064F	GCCTCCCTCGCGCCATCAG CACATGCGTTGTTTACTGCAC
E10	RA-1411F	GCCTCCCTCGCGCCATCAG AGACACACTATATGCCAAACACC
E11	RA-1785F	GCCTCCCTCGCGCCATCAG AGAGCCTCCAAAATTGCGTAG
E12	RA-2174F	GCCTCCCTCGCGCCATCAG GTGATTGGAAGCAAATTGTTATG
E13	RA-2569F	GCCTCCCTCGCGCCATCAG AATGCTGGTACAGATTCTAGGTG
E14	RA-2933F	GCCTCCCTCGCGCCATCAG CCAACACTGGCTGTATCAAAG
E15	RA-3339F	GCCTCCCTCGCGCCATCAG GTCCTACATCTGTGTTTAGCAGC
E16	RA-3778F	GCCTCCCTCGCGCCATCAG GCAACGTGACCAATTTTTGTC
E17	RA-810F	GCCTCCCTCGCGCCATCAG AATGGGCACACTAGGAATTGTG
E18	RA-968F	GCCTCCCTCGCGCCATCAG ATGACGAGAACGAAAATGACAG
E19	RA-1115F	GCCTCCCTCGCGCCATCAG AGGTTCTAAAACGAAAGTATTTGG
E20	RA-1361F	GCCTCCCTCGCGCCATCAG GTGGTTGCAGTCAGTACAGTAGTG
E21	RA-1471F	GCCTCCCTCGCGCCATCAG GCAAAGGCAGCAATGTTAGC
E22	RA-1672F	GCCTCCCTCGCGCCATCAG TCATGGGGAATGGTTGTGTTAC
E23	RA-1860F	GCCTCCCTCGCGCCATCAG GTATGGAGACACGCCAGAATG
E24	RA-2069F	GCCTCCCTCGCGCCATCAG TAAAGGATTGTGCAACAATGTG
E25	RA-2288F	GCCTCCCTCGCGCCATCAG TATATGGTGCAGCTAACACAGG
E26	RA-2476F	GCCTCCCTCGCGCCATCAG AATGCATTGGATGGAAATTTAG
E27	RA-2628F	GCCTCCCTCGCGCCATCAG TCCTAATGAGTTTCCATTTGACG
E28	RA-2857F	GCCTCCCTCGCGCCATCAG ACACATGCGCCTAGAATGTG
E29	RA-3037F	GCCTCCCTCGCGCCATCAG ACAAGACGTTAGCCTTGAAGTG
E30	RA-3199F	GCCTCCCTCGCGCCATCAG AGAGGGTCAAGTTGACTATTATGG
E31	RA-3455F	GCCTCCCTCGCGCCATCAG ACACAGACGACTATCCAGCGAC
E32	RA-3696F	GCCTCCCTCGCGCCATCAG CAGTGTCGTCTACATGGCATTG

* Binding positions (5'end) and binding direction are indicated after prefix "RA-". F: forward.

Table 4.6: Reverse barcode primers for semi-nested PCR in ASP16 strategy.

Primer name	Sequence (5' to 3')
RB-B01	GCCTTGCCAGCCCGCTCAG TGAC TGTGTTGGGTGTGTTTGG
RB-B02	GCCTTGCCAGCCCGCTCAG AGAC TGTGTTGGGTGTGTTTGG
RB-B03	GCCTTGCCAGCCCGCTCAG TCAC TGTGTTGGGTGTGTTTGG
RB-B04	GCCTTGCCAGCCCGCTCAG ACAC TGTGTTGGGTGTGTTTGG
RB-B05	GCCTTGCCAGCCCGCTCAG TGTC TGTGTTGGGTGTGTTTGG
RB-B06	GCCTTGCCAGCCCGCTCAG AGTC TGTGTTGGGTGTGTTTGG
RB-B07	GCCTTGCCAGCCCGCTCAG TCTC TGTGTTGGGTGTGTTTGG

(Table 4.6, continued)

Primer name	Sequence (5' to 3')
RB-B08	GCCTTGCCAGCCCGCTCAG ACTC TGTGTTGGGTGTGTTTGG
RB-B09	GCCTTGCCAGCCCGCTCAG CTGA TGTGTTGGGTGTGTTTGG
RB-B10	GCCTTGCCAGCCCGCTCAG CAGA TGTGTTGGGTGTGTTTGG
RB-B11	GCCTTGCCAGCCCGCTCAG CTGA TGTGTTGGGTGTGTTTGG
RB-B12	GCCTTGCCAGCCCGCTCAG CACA TGTGTTGGGTGTGTTTGG
RB-B13	GCCTTGCCAGCCCGCTCAG TAGC TGTGTTGGGTGTGTTTGG
RB-B14	GCCTTGCCAGCCCGCTCAG ATGC TGTGTTGGGTGTGTTTGG
RB-B15	GCCTTGCCAGCCCGCTCAG TACA TGTGTTGGGTGTGTTTGG
RB-B16	GCCTTGCCAGCCCGCTCAG ATCA TGTGTTGGGTGTGTTTGG
RB-B17	GCCTTGCCAGCCCGCTCAG TGCA TGTGTTGGGTGTGTTTGG
RB-B18	GCCTTGCCAGCCCGCTCAG TCGA TGTGTTGGGTGTGTTTGG
RB-B19	GCCTTGCCAGCCCGCTCAG AGCA TGTGTTGGGTGTGTTTGG
RB-B20	GCCTTGCCAGCCCGCTCAG ACGA TGTGTTGGGTGTGTTTGG
RB-B21	GCCTTGCCAGCCCGCTCAG CATC TGTGTTGGGTGTGTTTGG
RB-B22	GCCTTGCCAGCCCGCTCAG CTAC TGTGTTGGGTGTGTTTGG
RB-B23	GCCTTGCCAGCCCGCTCAG CTGC TGTGTTGGGTGTGTTTGG
RB-B24	GCCTTGCCAGCCCGCTCAG CAGC TGTGTTGGGTGTGTTTGG

4.8 Cervical carcinoma cell lines

Frozen stocks of the following cervical carcinoma cell lines were provided by Elisabeth Schwarz. The integrated HPV types and the cell lines are indicated in Table 4.7.

Table 4.7: Cervical carcinoma cell lines.

Cell line name	Integrated HPV type	Source
CaSki	HPV16	American Type Culture Collection
ME180	HPV68b	American Type Culture Collection
ME180R	HPV68b	M. Schaadt. Medizinische Klinik Universität Köln, Germany
MRI-H186	HPV16	American Type Culture Collection
MRI-H196	HPV16	American Type Culture Collection
SiHa	HPV16	American Type Culture Collection

4.9 Clinical DNA samples

Our collaboration partners in Besancon and Reims provided us with the clinical DNA samples from cervical scrapes or lesions. Through the routine cervical screening programs in Besancon and Reims, cervical scrapes were collected from women of different ages using the ThinPrep system. In brief, the cervical cells were collected by a cyto-brush and suspended in the ThinPrep PreserveCyt solution where the cells are fixed. This allows long-term storage of the samples. All samples underwent cytology analysis and HPV genotyping. For selected samples, DNA was extracted using Qiagen EZ1 DNA tissue kit (Qiagen, Hilden). Selected HPV16- and HPV68-positive DNA samples were sent to us. For an example, see (Briolat et al, 2007) for a description of the sample preparation and analysis.

4.10 *In vitro* cultivation of cervical carcinoma cell lines

All procedures took place under a laminar flow hood. All culture media and solutions used were prewarmed at 37°C. Cell cultures were incubated in a cell incubator at 37°C with 5% CO_2 and 95% humidity.

The cells were taken from 1-ml cell line aliquot stocks stored in liquid nitrogen. After rapidly thawing to room temperature, the cells were transferred to 14-ml complete DMEM medium in a 75cm^2 T-flask, and suspended by repeat pipetting. As soon as the cells adhered to the bottom of the flask, the medium was renewed to clear the residual DMSO from the freezing medium. Medium was changed every 2-3 day. When the cells had grown to 90% confluent, they were subcultured.

For subculturing (splitting), confluent cells were rinsed with 5 ml trypsin/EDTA, then incubated with new 3 ml trypsin/EDTA at 37°C for 3-10 minutes until the adherent cells rounded up. Cell suspensions were then diluted 1:10 or 1:20 with complete DMEM medium and put in a new 75cm^2 T-flask. After the dilution, the cells were suspended by repeat pipetting. The cultures were cultivated in the incubator.

For long-term storage of the cells, they were cultured in 175cm^2 T-flasks until reaching 80% confluency. After trypsinization, the cell pellets were collected by centrifugation at 800 rpm for 10 minutes at 4°C. Cell pellets were washed with 10 ml complete DMEM medium and collected by centrifugation again. Medium was discarded. Cell pellets were resuspended in 5 ml freezing medium and transferred to five cryogenic vials. The vials were cooled down slowly to -80°C, and then stored in liquid nitrogen.

4.11 Isolation of TNFalpha-resistant cells from ME180

The cervical carcinoma cell line ME180 was cultured in complete DMEM with 5 nM and 10 nM TNFalpha (Strathmann Biotec, Hamburg) continuously for 18 weeks. During this time, the culturing media with fresh TNFalpha were changed every 2-3 days. After 18 weeks, the survival cells were pooled and cultured in complete DMEM medium without TNFalpha for another 4 weeks. The two pooled cell populations, obtained from 5 nM and 10 mM TNFalpha treatments, were designated ME180-2A and ME180-3A, respectively.

4.12 TNFalpha cytotoxicity assay

The ME180, ME180R, ME180-2A and ME180-3A cells (see Results section 2.1.3) were seeded on a 96-well culture plate at density of 5000 and 10000 cells/well; 12 wells per cell concentration per cell population. The cells were cultured in complete DMEM medium overnight, allowing them to attach to the plate. In the next day, four different concentrations of TNFalpha (0, 2.5, 5 and 10 nM) were given; 3 wells per TNFalpha concentration per cell density per cell population. The cultures were incubated for 72 hours. Afterward, the media were discarded and the cells were fixed in 4% glutaraldehyde for 3 minutes and then washed in tab water. The fixed cells were stained with 1% crystal violet for 3 minutes and then washed in tab water. The plate was left to dry completely at room temperature. Prior to the measurement, 100 µl of 33% acetic acid was added to each well. The absorbance of the crystal violet dye, directly proportional to the cell amount in each well, was measured at wavelength 560 nm. The average absorbance values of each triplicate (three wells of the same TNFalpha concentration, seeding density and cell source) were determined. The cell viability was determined, in percentage, as the absorbance value after treatments with TNFalpha (2.5, 5 and 10 nM) over the value in absence of TNFalpha (0 nM) for each cell population.

4.13 Isolation of genomic DNA from monolayer cell cultures

Genomic DNA (gDNA) was isolated from cultured cells using a standard procedure (Strauss, 2001), which involves proteinase K digestion for protein degradation, repeated extractions with phenol/chloroform/isoamylalcohol for deproteinization, ethanol precipitation, and resuspension of DNA.

Cells were cultivated until 90-95% confluent in 75cm^2 T-flasks. The cell monolayers were rinsed twice with cold PBS/EDTA solution. To lyse the cells, 3 ml cold PBS/EDTA and 1.5 ml 3x lysis buffer were added to the flask. The lysis buffer was applied to the monolayer, and the flask was swirled slowly until the solution became viscous. The cell lysates were transferred to a 50-ml Falcon tube and 90 µl Proteinase K (final concentration 0.2 mg/ml) was added. The lysates were incubated at 55°C overnight, allowing histones to be digested and cellular nucleases to be inactivated by the enzyme. Next day, the reactions were cooled down to room temperature before 5 ml Phenol* was added. The reaction was suspended by inverting the tubes for 30 minutes, and the aqueous and phenol phases were separated by centrifugation at 2000 rpm for 10 minutes at 20°C. The aqueous layer (4.5 ml) was transferred to a new 14-ml Falcon tube, and 4.5 ml CIA solution was added. The solutions were mixed by inverting the tube for 30 minutes, and the aqueous and phenol phases were separated by centrifugation at 2000 rpm for 10 minutes at 20°C.

The aqueous layer (4.2 ml) was transferred to a new 14-ml Falcon tube, and the gDNA was precipitated by adding 0.1 volume of 3 M sodium acetate (pH 5.2) and 2 volumes ice-cold absolute ethanol. gDNA was collected by centrifugation at 5000 rpm for 30 minutes at 4°C, then washed with 5 ml ice-cold 75% ethanol. The DNA pellet was left at room temperature for 30 minutes to clear the residue ethanol. gDNA was suspended in 2 ml 1xTE buffer and stored at 4°C.

Due to the high molecular weight of genomic DNA, their quality and concentration were determined by gel electrophoresis, in comparison with Lambda DNA/HindIII DNA marker.

4.14 Polymerase chain reaction (PCR)

Polymerase chain reaction is a molecular technique used to amplify specific piece of DNA from trace amount of target molecules (Kuslich et al, 2008). It can produce millions copies of the target DNA. In this work, three PCR systems had been used: PCR with Taq DNA polymerase, with Pfu DNA polymerase, and with the Expand Long Template PCR System kit (Roche). The PCR reactions were set up in 250-µl PCR tubes and run in a Thermocycler PTC-200, which can handle PCR reaction volume up to 50 µl. The specificity of the PCR products was analyzed by agarose gel electrophoresis where 1/10 of each PCR products were run together with a suitable DNA marker

PCR system 1: PCR with Taq DNA polymerase

This was used for routine PCR. There is no proof-reading activity by Taq DNA polymerase and the amplified DNA products contain nontemplated A overhangs at their 3'end (http://www.invitrogen.com/). The enzyme, buffer and $MgCl_2$ were included in the Taq DNA Polymerase kit (Invitrogen).

Reaction set up		**Final concentration**
DNA template		250 ng genomic DNA
	or	0.5-2 µl GenomePlex library
	or	100-500 pg plasmid DNA
10xPCR buffer without $MgCl_2$		1x
50 mM $MgCl_2$		3 mM
10 mM dNTP (10mM each)		0.2 mM each
10 pmol/µl forward primer		0.2 µM
10 pmol/µl reverse primer		0.2 µM
Taq DNA polymerase (5 U/µl)		2 U per reaction

UltraPure water was added to the intended final volume.

PCR system 2: PCR with Pfu DNA polymerase

The Pfu DNA polymerase has a proof-reading activity (http://www.promega.com/). This PCR system was used to obtain accurate sequences of the DNA targets. The products are blunt-ended. The Pfu DNA Polymerase and Pfu buffer were from Promega.

Reaction set up		**Final concentration**
DNA template		250 ng genomic DNA
	or	0.5-2 µl GenomePlex library
	or	100-500 pg plasmid DNA
10xPfu buffer		1x
10 mM dNTP (10mM each)		0.2 mM each
10 pmol/µl forward primer		0.2 µM
10 pmol/µl reverse primer		0.2 µM
Pfu DNA polymerase (5 U/µl)		2 U per reaction

UltraPure water was added to the intended final volume.

PCR system 3: PCR with Expand Long Template PCR System

This system can amplify large targets up to 20 kb long. The enzyme mix consists of Taq DNA polymerase and Tgo DNA polymerase, thus possessing a proof-reading activity (http://www.roche-applied-science.com/). It was used for amplification of up to 8 kb DNA targets in this work, where the accuracy of sequences was considered. The products contain nontemplated A overhang at their 3'end. The enzyme mix and buffer were included in the Expand long template PCR system kit (Roche Applied Science). Reaction and cycling conditions were setup as recommended by the supplier.

Reaction set up		**Final concentration**
DNA template		250 ng genomic DNA
	or	0.5-2 µl GenomePlex library
	or	100-500 pg plasmid DNA
10xBuffer 3		1x
10 mM dNTP (10mM each)		0.2 mM each
10 pmol/µl forward primer		0.2 µM
10 pmol/µl reverse primer		0.2 µM
Enzyme mix		0.75 µl per reaction

UltraPure water was added to the intended final volume.

PCR cycle program for Taq and Pfu system

1. Initial denaturation	95°C, 3 min
2. Denaturation	94°C, 30 s

3. Annealing	60°C, 30 s
4. Elongation	72°C, 1 min per 1 kb target
5. Repeat from step 2 for another 34 cycles	
6. Final elongation	72°C, 10 min
7. Cooling	4°C, forever

PCR cycle program for Expand Long Template system

1. Initial denaturation	94°C, 3 min
2. Denaturation	94°C, 30 s
3. Annealing	60°C, 30 s
4. Elongation	68°C, 8 min
5. Repeat from step 2 for another 9 cycles	
6. Denaturation	94°C, 30 s
7. Annealing	60°C, 30 s
8. Elongation	68°C, 8 min
	+20 s for each successive cycle
9. Repeat from step 6 for another 24 cycles	
10. Final elongation	68°C, 10 min
11. Cooling	4°C, forever

4.15 Purification of PCR products

In some occasions, the PCR products needed to be purified to get rid of unspecific product bands or to get rid of salts in the PCR solution. This was accomplished by two alternative methods: direct PCR purification and gel extraction.

For direct PCR purification, the total PCR products were purified through columns using QIAquick PCR Purification kit according to the manufacturer's manual. This kit eliminates primer-dimer products (DNA size below 100 bp) and salts from the original PCR products.

For gel extraction, total PCR products were loaded on 1.0-1.5% TAE agarose gel and run until the band of interest was clearly separated from other bands. The target band was excised from the gel and the DNA was purified from the gel using QIAquick Gel Extraction kit according to the manufacturer's manual.

Routinely, an aliquot of the purified PCR band/product was run in agarose gel electrophoresis to ascertain its quality, before it was used for further experiments.

4.16 Cloning of PCR products

The PCR products from Taq DNA polymerase and Expand Long Template PCR systems contain nontemplated A overhangs at the 3'end, while those from Pfu DNA polymerase PCR system have blunt end. The PCR products with sticky end were cloned into TA vectors. Two commercial cloning systems for TA cloning were used: Invitrogen's TOPO TA PCR cloning kit (TopoPCR4 vector) and Stratagene's StrataClone PCR cloning kit (pSC-A vector). The blunt-end PCR products were cloned into blunt vectors with either of the two commercial kits: Invitrogen's TOPO Blunt PCR cloning kit (TopoPCR4 vector) and Stratagene's StrataClone Blunt PCR cloning kit (pSC-B vector).

The cloning procedures were performed according to the manufacturer's manual, with some modifications. For all four kits used, the cloning reaction volume was reduced to half of the recommended volume from the manufacturer. Briefly, 1 µl PCR product was mixed with 1.5 µl cloning buffer and 0.5 µl vector mix in 3-µl reaction volume, then incubated at room temperature for 15-30 minutes. The reaction was transferred into thawed competent cells (Top10 for TOPO kits, and SoloPack for StrataClone kits), mixed and incubated on ice for 30 minutes. The reaction was heat shocked at 42°C for 1 minute and then immediately put on ice. After 2 minutes on ice, 250 µl prewarmed SOC medium were added. The reaction was incubated at 37°C, with horizontal shaking at 220 rpm, for 1 hour before spread on LB agar plates containing selective antibiotics (100 µg/ml ampicillin). The plates were incubated overnight at 37°C to allow colonies to grow.

4.17 Plasmid DNA preparation

Quick and dirty plasmid DNA mini preparation method (quick-miniprep) was used as a rapid plasmid DNA isolation method for screening of positive clones. The plasmid DNA obtained by this method is suitable for restriction enzyme digestions, but not for direct sequencing.

To screen for positive clones, single bacterial colonies were picked from the LB plates spread after cloning. Each of them were cultured in 5 ml LB with 100 µg/ml ampicillin at 37°C with shaking at 220 rpm overnight. Next morning, bacterial cells were collected from 1 ml of each overnight-cultures by centrifugation at 13000 rpm for 4 minutes. Cell pellets were resuspended in 100 µl TEG buffer. To lyse the cells, 200 µl NaOH/SDS solution was added to the cell suspension and mixed by inverting the tubes 10 times. The lysates were incubated at room temperature for 3-5 minutes until the solution became viscous. 150 µl 3 M sodium acetate (pH 5.2) was added to each reaction, and the tubes were shortly vortexed before put on ice. After 5 minutes on ice, cell debris was separated by centrifugation at 13000 rpm for 7-10 minutes. The supernatant was

transferred to a new tube containing 800 µl ice-cold absolute ethanol and the reactions were incubated on ice for 2 minutes. The precipitated plasmid DNA in the supernatant was collected by centrifugation at 13000 rpm for 5 minutes, and then washed with 1 ml ice-cold 75% ethanol. Air-dry plasmid DNA pellets were resuspended with 30 µl TE/RNase solution. RNase was added to eliminate the co-isolated RNA. 3-µl aliquots of each plasmid DNA were digested with EcoRI, which cut at two locations on the vector adjacent to the cloning site. The digestion reactions were incubated at 37°C for 1 hour before loading on agarose gel for electrophoresis. Plasmid DNA containing correct inserts can be detected by the presence of the vector band and the insert band.

For sequencing and for long-term plasmid DNA storage, the plasmid DNAs were isolated from the bacterial cultures using commercial kits. Two formats of commercial kits were use in this study: QIAprep Spin Miniprep kit (for small scale) and Qiagen Plasmid Maxi kit (for large scale). For small scale, bacterial cell pellets were collected from 4 ml of bacterial cultures (with positive inserts). For large scale, bacterial cell pellets were collected from 100 ml of bacterial cultures (with positive inserts). The plasmid DNA isolation procedures were performed as described by the manufacturer's manuals.

4.18 DNA sequencing

Plasmid DNA containing inserts of interest were sent to Andreas Hunziker (DKFZ Genomics and Proteomics Core Facilities) for sequencing by the Sanger DNA sequencing method. The DNAs were sequenced with Big-Dye terminator chemistry on an ABI model sequencer.

4.19 Southern hybridization

Southern hybridization is a method for detection of a specific DNA sequence in a DNA population (such as genomic DNA or PCR products) by hybridizing the DNA population with a specific DNA probe (Mays Hoopes, 2008). It is divided into 4 steps: DNA transfer from agarose gel to nylon membrane (blotting), labeling of probes, hybridization and signal visualization.

Transfer of DNA from agarose gel to nylon membrane

Genomic DNA was first digested by appropriate restriction endonulease(s) before loaded on 0.8% agarose gel. For PCR products, restriction digestion may be omitted. After separation was complete, selected DNA molecular marker positions were marked on the gel using the scalpel. The gel was soaked in 0.25 M HCl for 10 minutes to partially depurinate DNA, allowing more efficient transfer of large DNA fragments (larger than 4 kb). This step was not necessary for PCR products. Then the gel was immersed with agitation in denaturation buffer (0.4 M NaOH) twice,

each for 15 minutes, to denature the DNA. Afterward the gel was ready for transfer. Vertical downward capillary transfer was used in this study.

In preparation for transfer, one Hybond N+ nylon membrane was cut to the size slightly larger than the gel and four Whatman 3 MM papers were cut to the size slightly larger than the nylon membrane. The system was setup by placing a stack of tissue paper on the table first, followed by two dry Whatman paper. Another two Whatman papers were soaked in 0.4 M NaOH before placed on top of the dry Whatman papers. The nylon membrane was shortly soaked in 0.4 M NaOH and then placed on top of the Whatman papers. The gel from previous denaturation step was placed atop the nylon membrane. For efficient transfer, it is important to avoid any air bubbles during the transfer setup. The whole setup was wrapped with plastic foil. A small plastic board (only large enough to cover the gel) was placed on top of the setup and then 250 g weight was placed on top of the board. The DNA transferring process took approximately 4-6 hours. During the transfer, DNA was driven from the gel to the membrane by capillary force. Once arrived at the membrane, the DNA binds to the positively charged nylon membrane, and is fixed to the membrane by the alkaline solvent. After the transfer was complete, the marks of the selected DNA markers, as well as the wells on the gel, were marked on the nylon membrane with colored pencil. The membrane was then soaked in the neutralizing buffer (3 M NaCl, 0.3 M sodium citrate, 0.5 M Tris-HCl, pH 7.0) for 15 minutes before let dry on a clean Whatman paper at room temperature.

Radioactive labeling of DNA probes

For hybridization, the DNA probes were labeled with [α-^{32}P] dCTP, using Rediprime II DNA Labeling System (GE Healthcare) as described in the manufacturer's manual. Briefly, 25-30 ng linear DNA probe was diluted in UltraPure water to a volume of 45 µl. The DNA was denatured at 95°C for 5 minutes, then put immediately on ice. The DNA was spun down shortly, and transferred to a Rediprime II reaction mixture tube. 5 µl of 5 µCi [α-^{32}P] dCTP was added to the mixture and mixed by pipetting. The reaction was incubated at 37°C for 30 minutes and stopped by adding 20 µl stop-solution (2xTNE, 0.1% BPB, 0.5% blue dextran). The labeling reactions were later purified through pre-equilibrated columns. To equilibrate a purification column, the resin particles in a MicroSpin G-25 column were resuspended by vigorous vortex. The tip of the column was cut off, and the column was placed in a 2 ml collecting tube. The tubes were centrifuged at 300 rpm for 1 minute and the flow-through was discarded. To equilibrate the resin, 200 µl 1xTNE was added to the column, and the tubes were centrifuged at 300 rpm for 1 minute. The column was transferred into a new collecting tube. 75 µl 1xTNE were added to the complete labeling reaction. The solution was mixed by pipetting and transferred to the equilibrated column. The tubes were centrifuged at 300 rpm for 1 minute. The small DNA and dNTP remained in the

matrix of the column, while larger DNA molecules passed through. The eluate was stored at -20°C or directly used for hybridization.

DNA hybridization with a radioactive labeled probe

The hybridization of labeled DNA probe with the complementary sequences on the membrane were performed with high-stringency condition, at 68°C. The hybridization oven and hybridization solution were prewarmed to 68°C before use. The membrane was prehybridized at 68°C in the oven with constant rotation for at least 1 hour before it was ready for hybridization. Prior to adding the labeled probe, the prehybridization solution was discarded and 10 µl of fresh prewarmed hybridization solution was added to the tube. 70 µl of the labeled probe were used per 10 ml hybridization volume. The labeled probe was denatured by the addition of 0.2 M NaOH (final concentration) and incubated at room temperature for 10 minutes. The denatured probe was added to the hybridization tube, and the membrane was hybridized overnight at 68°C with constant rotation. Next day, the membrane was washed 3x 30 minutes with prewarmed washing buffer (2xSSC, 0.1% SDS) at 68°C with constant rotation to reduce unspecific binding background. Afterward, the membrane was dried at room temperature and wrapped with plastic foil.

Detection of hybridization signal by autoradiography

The P-32 emits high-energy ß-particles that can easily be detected by autoradiography. To set up the exposure, the membrane was taped to a piece of Whatman 3MM paper and placed in a cassette. An autoradiography film, MR film, was placed over the membrane and covered with an intensifying screen. The cassette was closed and stored at -80°C. Genomic DNA hybridization requires longer exposure time than PCR product hybridization. The usual exposition times were 7 days and 30-120 minutes for genomic DNA and PCR products, respectively. After the exposure was complete, the film was taken out and developed by Curix 60 film processor machine.

4.20 DNA amplification with phi29 DNA polymerase

Phi29 DNA polymerase is a replicative polymerase isolated from *Bacillus subtilis* phage phi29. The enzyme exhibits exceptional strand-displacement and processive synthesis properties. It has also inherent 3'-5' proofreading exonuclease activity (Blanco et al, 1989; Esteban et al, 1993). The enzyme is able to amplify from both circular and linear DNA templates at 30°C, and the primers are required to be exonuclease-resistant (Dean et al, 2002; Hutchison et al, 2005).

In this work, phi29 DNA polymerase, in combination with random hexamer, was used to amplified linear DNA template (called multiple displacement amplification, MDA) and circular

DNA template (called rolling circle amplification, RCA). To make the random hexamer resistant to exonuclease activity, it was modified with phosphothioate at the 3' end (Hutchison et al, 2005). The reaction conditions of both MDA and RCA were identical except for the nature of template DNA. The products of MDA and RCA are high molecular weight linear DNA and high molecular weight linear concatameric DNA respectively. The DNA may reach up to 70 kb long.

The reactions were carried out at 20 µl final volume. The maximum volume of template DNA was limited to 3 µl. The volume of the template DNA was brought to 3 µl by adding UltraPure water. First, the template DNA was mixed with 1 µl 500 pmol/µl random hexamer (phosphothioate modified at 3' end) and 1 µl 5xAnnealing buffer. The reaction was mixed by pipetting and denatured at 95°C for 3 minutes and immediately left on ice for 2 minutes. The polymerase started by adding 2 µl 10xBSA, 2 µl 10xPhi29 buffer, 5 µl 10 mM dNTP, 1 µl phi29 DNA polymerase (Fermentas) and 5 µl UltraPure water. The reaction was incubated at 30°C for 16 hours, then inactivated at 65°C for 10 minutes. The quality of the amplified DNA products was observed by running a 1-µl aliquot on 1% agarose gel together with lamda-DNA/HindIII marker. The products should distribute mostly between 23 kb band and the 2 kb band of the DNA marker.

4.21 Preparation of pBluescript (pBS) vector for cloning at EcoRI

To prepare the pBS vector for cloning at the EcoRI site, 12 µg pBS was digested with 140 U EcoRI in 100 µl reaction volume. The linearized vector was purified by phenol/chloroform-chloroform extraction and later precipitated with sodium acetate and ethanol. In brief, 1 volume of phenol/chloroform solution was added to the digestion reaction. The mixture was vigorously shaken for 15 seconds and centrifuged at 13000 for 5 minutes. The upper aqueous layer was collected in a new tube, and 1 volume of chloroform was added. The reaction tube was vigorously shaken for 15 seconds and centrifuged at 13000 rpm for 5 minutes. The upper aqueous phase was collected in a new tube and the plasmid DNA was precipitated by addition of 0.1 volume 3 M sodium acetate (pH 5.2) and 2.5 volumes of ice-cold absolute ethanol. The tube was incubated at -20°C for 1 hour before centrifuged at 13000 rpm for 20 minutes. The DNA pellet was washed with 300 µl ice-cold 75% ethanol. The air-dried DNA pellet was resuspended in UltraPure water.

The linearized pBS was dephosphorylated to prevent self-ligation. Briefly, a 4 µg aliquot of the linearized pBS was added with 5 µl 10xCIAP buffer, 5 µl CIAP (1 U/µl) and UltraPure water to final volume of 50 µl. The reaction was incubated at 37°C for 2 hours before inactivated at 85°C for 20 minutes. The reaction was purified by phenol/chloroform-chloroform extraction and later precipitated with sodium acetate and ethanol, as described above. The dephosphorylated EcoRI-

linearized pBS vector was resuspended with 12 µl UltraPure water and stored at -20°C prior to use.

4.22 ASP16 strategy for HPV16 integration analysis

The newly developed ASP16 strategy for HPV16 integration analysis in cervical scrapes (Xu, 2010) consists of four major steps: (1) whole genome amplification, (2) HPV16 DNA enrichment, (3) high-throughput DNA sequencing using the Roche/454 Genome Sequencer FLX, and (4) data analysis.

4.22.1 GenomePlex whole genome amplification

Since DNA of clinical samples was only available in nanogram amounts, the genomic DNA samples were amplified with GenomePlex whole genome amplification kit (GenomePlex WGA2 kit, Sigma Aldrich) to provide a library of amplified DNA fragments. In the GenomePlex WGA method, the DNA is first chemically fragmented, then GenomePlex universal adapters are added to both ends of the fragmented DNA molecules to create template molecules for library amplification, and the complete library is obtained by PCR amplification of these template molecules. Such library is called OmniPlex library. As the result of fragmentation, the OmniPlex library contained amplified DNA molecules with size distribution between 200 bp - 1.5 kb. The GenomePlex amplification was carried out according to the manufacturer's manual. The minimum template DNA was 10 ng. Generally for clinical DNA samples, 10-50 ng DNA were used for templates. The entire incubation process was carried out on a Thermocycler PTC-200 machine.

4.22.2 HPV16 enrichment from GenomePlex DNA libraries

The enrichment of HPV16 DNA was carried out in three steps: linear amplification of HPV16 DNA, isolation of HPV16 DNA via biotin-streptavidin, and HPV16 semi-nested PCR. In this study, the HPV16-specific linear amplification and semi-nested PCR were performed as multiplex amplification reactions. Four amplification sets were performed for each DNA sample during both of these steps. Each amplification set consisted of a combination of 7-8 HPV16 primers. Primers are listed in section 4.7.4 (Table 4.4, Table 4.5 and Table 4.6).

HPV16 linear amplification by primer extension

Thirty-two 5'-biotin-labeled HPV16-specific forward primers, located in the E1-E2 region (see Figure 2.33), were used in this step, dividing into four groups of primer mixes. See Table 2.14 in

Results section 2.2.2.3 for primer combinations of the four mixes. Four linear amplification reactions were setup for each DNA sample, each reaction containing one of the four primer mixes, respectively. In the primer mixes, each biotin-labeled HPV16 primer had a concentration of 0.1 µM. Linear amplification was carried out in 50 µl reaction volume, using Taq DNA polymerase (Invitrogen). In brief, 2.5 µl GenomePlex library was mixed with 5 µl 10xreaction buffer without $MgCl_2$, 3 µl 50mM $MgCl_2$, 1 µl dNTP (10 mM each), 8 µl biotin-labeled HPV16 primer mix, 0.5 µl Taq DNA polymerase and 30 µl UltraPure water. The polymerization reaction was carried out for 45 cycles. The amplification products were single-stranded HPV16 DNA with biotin-labeled at the 5'end of the DNA strand. The cycle conditions are described below:

1. Initial denaturation	95°C, 2 min
2. Denaturation	94°C, 30 s
3. Annealing	60°C, 75 s
4. Elongation	72°C, 40 s
5. Repeat steps 2-4 for another 44 cycles	
6. Cooling	4°C, forever

Isolation of HPV16 DNA with biotin-streptavidin system

The biotin at the 5'end of each HPV16 single-stranded DNA molecules allowed easy isolation of these molecules with streptavidin-coated magnetic beads. BioMag streptavidin beads (Polysciences) were used. Briefly, a 75-µl aliquot of suspended BioMag streptavidin beads (1 mg/ml) was transferred to a 1.5-ml tube. The tube was placed in a magnet and incubated for 1 minute, allowing the beads to concentrate at one side of the tube. The supernatant was discarded and the tube was removed from the magnet. To wash the beads, they were resuspended in 100 µl 2xBinding buffer (40 mM Tris-HCl, 2 mM EDTA, 2 M NaCl, 0.1% Tween20, pH 7.8 in DEPC water), separated from supernatant on the magnet, and resuspended again in 50 µl 2xBinding buffer. A total 50 µl reaction from the linear amplification step was added to the bead suspension, mixed by pipetting, and incubated for 15-30 minutes at room temperature with occasional agitation. The biotin-labeled HPV16 DNA molecules bound to the beads, while the background genomic DNA did not. The HPV16-DNA-bound BioMag beads were placed on the magnet, incubated for 1 minute and the supernatant was discarded. The beads were washed twice with 150 µl 1xWash buffer (20 mM Tris-HCl, 1 mM EDTA, 1 M NaCl, 0.05% Tween20, pH 7.8 in DEPC water), then three times with 150 µl 1xTE buffer (in DEPC water). The beads were resuspended in 30 µl 1xTE buffer and stored at -20°C.

HPV16 semi-nested multiplex PCR

The isolated single-stranded HPV16 DNA molecules were amplified by multiplex PCR, generating amplified double-stranded HPV16 DNA as the products. To make the PCR products

compatible for GS-FLX sequencing, Roche-A and Roche-B adapters were required at both ends of each DNA molecule. As forward primers, nested HPV16 primers, fused with the Roche-A sequence at the 5'end, were used (RA_HPV16, see Table 4.5). The reverse primers contained the Roche-B sequence at the 5' end, followed by 4-nt barcode and then the GenomePlex universal adapter sequence (RB-B01 to -B24, see Table 4.6). For each DNA sample, the nested HPV16 primers were assembled in four groups and used in four separate multiplex PCR reactions. For each DNA sample, one reverse primer with a unique barcode was used. See Table 2.14 in Results section 2.2.2.3 for primer combinations of the four reactions. The multiplex PCR was carried out in 50 µl reaction volume, using Qiagen multiplex PCR kit, according to the manufacturer's manual. Briefly, a 2.5 µl aliquot of each purified single-stranded HPV16 DNA product from previous step was mixed with 25 µl 2xQiagen multiplex PCR master mix, 2.5 µl of corresponding 20xRA_HPV16 primer mix (2 µM each), 1 µl 10 pmol/µl RB-Barcode-GPUA primer (unique for each DNA sample) and 19 µl UltraPure water. The cycle conditions are described below:

1. Initial denaturation	95°C, 15 min
2. Denaturation	94°C, 30 s
3. Annealing	60°C, 75 s
4. Elongation	72°C, 1 min
5. Repeat from step 2 for another 34 cycles	
6. Cooling	4°C, forever

The quality of the multiplex PCR products was determined by running 3-µl aliquots on agarose gels. By blotting the gel and hybridizing with an HPV16-specific probe, the specificity of the multiplex PCR products was determined.

4.22.3 Size selection of HPV16 multiplex PCR products

In experiment ASP16-4, it was attempted to limit the size distribution of the multiplex PCR product to 200-300 bp, using the E-Gel SizeSelect system (Invitrogen). The procedure was carried out according to the manufacturer's manual. Each multiplex PCR product was loaded on a single well of E-Gel SizeSelect. The E-Gel contained 2% agarose with dye that allowed real-time visulization of DNA migration. Smear portions of the desired DNA size were collected from the collecting wells by directly pipetting the solution out. The quality and specificity of the smear portion of each multiplex PCR products were determined by agarose gel electrophoresis and Southern hybridization with HPV16 probe. The four size-selected aliquots of each DNA sample were pooled in equal amounts, approximately 100 ng each. The mixture was concentrated by precipitation with 3 M sodium acetate (pH 5.2) and ethanol, and dissolved in 100 µl TE. The concentration and quality of the pooled amplicons was measured with NanoDrop, before it was sent to DKFZ Genomics and Proteomics Core Facilities for sequencing.

4.22.4 Roche/454 GS-FLX pyrosequencing

The amplicons were sequenced by the Roche/454 GS-FLX pyrosequencing. One of the two regions on 70x75 picotiter plate was chosen for sequencing platform. According to the manufacturer, the selected sequencing format can generate, per each sequencing round, up to 210000 sequence reads of up to 250 nt in length. Individual amplicon molecules were amplified by emulsion PCR (emPCR) before they were sequenced on the sequencing platform (more details can be found at http://www.454.com/). In this work, Roche-B was used as the sequencing primer. This resulted in sequence reads starting from the 4-nt barcode sequence, followed by GenomePlex universal adapter, then the amplicon sequence, and then the nested RA-HPV16 primer sequence at the end. The outputs were generated as a list of FASTA sequence reads.

4.22.5 Data analysis

Each Roche/454 GS-FLX run generated up to 210000 single sequence reads. This required initial automatic data analysis and manipulation. Writing a suitable set of programs to analyze ASP16 sequence data using Perl programming language was an essential part of this PhD work. The details are presented in Results and Discussion sections. Shortly, the output FASTA sequence reads from GS-FLX sequencing were sorted into sample groups according to the 4-nt barcodes. Then for each sample group, the programs analyzed and selected only informative sequence reads into output lists. These selected sequences were aligned with Clustal algorithms, and then re-edited for easier visualization. The sequence and sequence alignment outputs can be viewed and manipulated with any sequence editor software. In this work, HUSAR and Geneious Pro were employed. The programs then proceeded to calculate and generate basic statistical data, of which the outputs were written in tab-delimited format that may be open in Microsoft Excel or similar software.

References

Azizi N, Brazete J, Hankins C, Money D, Fontaine J, Koushik A, Rachlis A, Pourreaux K, Ferenczy A, Franco E, Coutlee F (2008) Influence of human papillomavirus type 16 (HPV-16) E2 polymorphism on quantification of HPV-16 episomal and integrated DNA in cervicovaginal lavages from women with cervical intraepithelial neoplasia. *The Journal of general virology* **89:** 1716-1728

Baker CC, Phelps WC, Lindgren V, Braun MJ, Gonda MA, Howley PM (1987) Structural and transcriptional analysis of human papillomavirus type 16 sequences in cervical carcinoma cell lines. *Journal of virology* **61:** 962-971

Batta K, Kundu TK (2007) Activation of p53 function by human transcriptional coactivator PC4: role of protein-protein interaction, DNA bending, and posttranslational modifications. *Molecular and cellular biology* **27:** 7603-7614

Batta K, Yokokawa M, Takeyasu K, Kundu TK (2009) Human transcriptional coactivator PC4 stimulates DNA end joining and activates DSB repair activity. *Journal of molecular biology* **385:** 788-799

Bernard HU, Burk RD, Chen Z, van Doorslaer K, Hausen H, de Villiers EM (2010) Classification of papillomaviruses (PVs) based on 189 PV types and proposal of taxonomic amendments. *Virology* **401:** 70-79

Blanco L, Bernad A, Lazaro JM, Martin G, Garmendia C, Salas M (1989) Highly efficient DNA synthesis by the phage phi 29 DNA polymerase. Symmetrical mode of DNA replication. *The Journal of biological chemistry* **264:** 8935-8940

Bory JP, Cucherousset J, Lorenzato M, Gabriel R, Quereux C, Birembaut P, Clavel C (2002) Recurrent human papillomavirus infection detected with the hybrid capture II assay selects women with normal cervical smears at risk for developing high grade cervical lesions: a longitudinal study of 3,091 women. *International journal of cancer* **102:** 519-525

Boyer SN, Wazer DE, Band V (1996) E7 protein of human papilloma virus-16 induces degradation of retinoblastoma protein through the ubiquitin-proteasome pathway. *Cancer research* **56:** 4620-4624

Briolat J, Dalstein V, Saunier M, Joseph K, Caudroy S, Pretet JL, Birembaut P, Clavel C (2007) HPV prevalence, viral load and physical state of HPV-16 in cervical smears of patients with different grades of CIN. *International journal of cancer* **121:** 2198-2204

Callahan DE, Karim A, Zheng G, Tso PO, Lesko SA (1992) Quantitation and mapping of integrated human papillomavirus on human metaphase chromosomes using a fluorescence microscope imaging system. *Cytometry* **13:** 453-461

Calleja-Macias IE, Kalantari M, Allan B, Williamson AL, Chung LP, Collins RJ, Zuna RE, Dunn ST, Ortiz-Lopez R, Barrera-Saldana HA, Cubie HA, Cuschieri K, Villa LL, Bernard HU (2005) Papillomavirus subtypes are natural and old taxa: phylogeny of human papillomavirus types 44 and 55 and 68a and -b. *Journal of virology* **79:** 6565-6569

Casas L, Galvan SC, Ordonez RM, Lopez N, Guido M, Berumen J (1999) Asian-american variants of human papillomavirus type 16 have extensive mutations in the E2 gene and are highly amplified in cervical carcinomas. *International journal of cancer* **83:** 449-455

Castellsague X (2008) Natural history and epidemiology of HPV infection and cervical cancer. *Gynecologic oncology* **110:** S4-7

Chan JK, Monk BJ, Brewer C, Keefe KA, Osann K, McMeekin S, Rose GS, Youssef M, Wilczynski SP, Meyskens FL, Berman ML (2003) HPV infection and number of lifetime sexual partners are strong predictors for 'natural' regression of CIN 2 and 3. *British journal of cancer* **89:** 1062-1066

Chan SY, Ho L, Ong CK, Chow V, Drescher B, Durst M, ter Meulen J, Villa L, Luande J, Mgaya HN, et al. (1992) Molecular variants of human papillomavirus type 16 from four continents suggest ancient pandemic spread of the virus and its coevolution with humankind. *Journal of virology* **66:** 2057-2066

Chellappan S, Kraus VB, Kroger B, Munger K, Howley PM, Phelps WC, Nevins JR (1992) Adenovirus E1A, simian virus 40 tumor antigen, and human papillomavirus E7 protein share the capacity to disrupt the interaction between transcription factor E2F and the retinoblastoma gene product. *Proceedings of the National Academy of Sciences of the United States of America* **89:** 4549-4553

Chen Z, Terai M, Fu L, Herrero R, DeSalle R, Burk RD (2005) Diversifying selection in human papillomavirus type 16 lineages based on complete genome analyses. *Journal of virology* **79:** 7014-7023

Clavel C, Masure M, Bory JP, Putaud I, Mangeonjean C, Lorenzato M, Nazeyrollas P, Gabriel R, Quereux C, Birembaut P (2001) Human papillomavirus testing in primary screening for the detection of high-grade cervical lesions: a study of 7932 women. *British journal of cancer* **84:** 1616-1623

Couturier J, Sastre-Garau X, Schneider-Maunoury S, Labib A, Orth G (1991) Integration of papillomavirus DNA near myc genes in genital carcinomas and its consequences for proto-oncogene expression. *Journal of virology* **65:** 4534-4538

Coward E, Haas SA, Vingron M (2002) SpliceNest: visualization of gene structure and alternative splicing based on EST clusters. *Trends Genet* **18:** 53-55

Cullen AP, Reid R, Campion M, Lorincz AT (1991) Analysis of the physical state of different human papillomavirus DNAs in intraepithelial and invasive cervical neoplasm. *Journal of virology* **65:** 606-612

de Sanjose S, Quint WG, Alemany L, Geraets DT, Klaustermeier JE, Lloveras B, Tous S, Felix A, Bravo LE, Shin HR, Vallejos CS, de Ruiz PA, Lima MA, Guimera N, Clavero O, Alejo M, Llombart-Bosch A, Cheng-Yang C, Tatti SA, Kasamatsu E, Iljazovic E, Odida M, Prado R, Seoud M, Grce M, Usubutun A, Jain A, Suarez GA, Lombardi LE, Banjo A, Menendez C, Domingo EJ, Velasco J, Nessa A, Chichareon SC, Qiao YL, Lerma E, Garland SM, Sasagawa T, Ferrera A, Hammouda D, Mariani L, Pelayo A, Steiner I, Oliva E, Meijer CJ, Al-Jassar WF, Cruz E, Wright TC, Puras A, Llave CL, Tzardi M, Agorastos T, Garcia-Barriola V, Clavel C, Ordi J, Andujar M, Castellsague X, Sanchez GI, Nowakowski AM, Bornstein J, Munoz N, Bosch FX (2010) Human papillomavirus genotype attribution in invasive cervical cancer: a retrospective cross-sectional worldwide study. *The lancet oncology* **11:** 1048-1056

de Villiers EM, Fauquet C, Broker TR, Bernard HU, zur Hausen H (2004) Classification of papillomaviruses. *Virology* **324:** 17-27

Dean FB, Hosono S, Fang L, Wu X, Faruqi AF, Bray-Ward P, Sun Z, Zong Q, Du Y, Du J, Driscoll M, Song W, Kingsmore SF, Egholm M, Lasken RS (2002) Comprehensive human genome amplification using multiple displacement amplification. *Proceedings of the National Academy of Sciences of the United States of America* **99:** 5261-5266

Doorbar J (2005) The papillomavirus life cycle. *J Clin Virol* **32 Suppl 1:** S7-15

Doorbar J (2006) Molecular biology of human papillomavirus infection and cervical cancer. *Clin Sci (Lond)* **110:** 525-541

Doorbar J, Ely S, Sterling J, McLean C, Crawford L (1991) Specific interaction between HPV-16 E1-E4 and cytokeratins results in collapse of the epithelial cell intermediate filament network. *Nature* **352:** 824-827

Durst M, Croce CM, Gissmann L, Schwarz E, Huebner K (1987) Papillomavirus sequences integrate near cellular oncogenes in some cervical carcinomas. *Proceedings of the National Academy of Sciences of the United States of America* **84:** 1070-1074

Eriksson A, Herron JR, Yamada T, Wheeler CM (1999) Human papillomavirus type 16 variant lineages characterized by nucleotide sequence analysis of the E5 coding segment and the E2 hinge region. *The Journal of general virology* **80 (Pt 3):** 595-600

Esteban JA, Salas M, Blanco L (1993) Fidelity of phi 29 DNA polymerase. Comparison between protein-primed initiation and DNA polymerization. *The Journal of biological chemistry* **268:** 2719-2726

Fehrmann F, Klumpp DJ, Laimins LA (2003) Human papillomavirus type 31 E5 protein supports cell cycle progression and activates late viral functions upon epithelial differentiation. *Journal of virology* **77:** 2819-2831

Ferber MJ, Thorland EC, Brink AA, Rapp AK, Phillips LA, McGovern R, Gostout BS, Cheung TH, Chung TK, Fu WY, Smith DI (2003) Preferential integration of human papillomavirus type 18 near the c-myc locus in cervical carcinoma. *Oncogene* **22:** 7233-7242

Fu L, Terai M, Matsukura T, Herrero R, Burk RD (2004) Codetection of a mixed population of candHPV62 containing wild-type and disrupted E1 open-reading frame in a 45-year-old woman with normal cytology. *The Journal of infectious diseases* **190:** 1303-1309

Giannoudis A, Herrington CS (2001) Human papillomavirus variants and squamous neoplasia of the cervix. *The Journal of pathology* **193:** 295-302

Graham DA, Herrington CS (2000) HPV-16 E2 gene disruption and sequence variation in CIN 3 lesions and invasive squamous cell carcinomas of the cervix: relation to numerical chromosome abnormalities. *Mol Pathol* **53:** 201-206

Ham J, Dostatni N, Gauthier JM, Yaniv M (1991) The papillomavirus E2 protein: a factor with many talents. *Trends in biochemical sciences* **16:** 440-444

Ho GY, Bierman R, Beardsley L, Chang CJ, Burk RD (1998) Natural history of cervicovaginal papillomavirus infection in young women. *The New England journal of medicine* **338:** 423-428

Ho L, Chan SY, Burk RD, Das BC, Fujinaga K, Icenogle JP, Kahn T, Kiviat N, Lancaster W, Mavromara-Nazos P, et al. (1993) The genetic drift of human papillomavirus type 16 is a means of reconstructing prehistoric viral spread and the movement of ancient human populations. *Journal of virology* **67:** 6413-6423

Hopman AH, Smedts F, Dignef W, Ummelen M, Sonke G, Mravunac M, Vooijs GP, Speel EJ, Ramaekers FC (2004) Transition of high-grade cervical intraepithelial neoplasia to micro-invasive carcinoma is characterized by integration of HPV 16/18 and numerical chromosome abnormalities. *The Journal of pathology* **202:** 23-33

Hudelist G, Manavi M, Pischinger KI, Watkins-Riedel T, Singer CF, Kubista E, Czerwenka KF (2004) Physical state and expression of HPV DNA in benign and dysplastic cervical tissue: different levels of viral integration are correlated with lesion grade. *Gynecologic oncology* **92:** 873-880

Hutchison CA, 3rd, Smith HO, Pfannkoch C, Venter JC (2005) Cell-free cloning using phi29 DNA polymerase. *Proceedings of the National Academy of Sciences of the United States of America* **102:** 17332-17336

Jacobs MV, de Roda Husman AM, van den Brule AJ, Snijders PJ, Meijer CJ, Walboomers JM (1995) Group-specific differentiation between high- and low-risk human papillomavirus genotypes by general primer-mediated PCR and two cocktails of oligonucleotide probes. *Journal of clinical microbiology* **33:** 901-905

Jansen R, Yu H, Greenbaum D, Kluger Y, Krogan NJ, Chung S, Emili A, Snyder M, Greenblatt JF, Gerstein M (2003) A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science* **302:** 449-453

Jarrett CR, Blancato J, Cao T, Bressette DS, Cepeda M, Young PE, King CR, Byers SW (2001) Human APC2 localization and allelic imbalance. *Cancer research* **61:** 7978-7984

Jeon S, Allen-Hoffmann BL, Lambert PF (1995) Integration of human papillomavirus type 16 into the human genome correlates with a selective growth advantage of cells. *Journal of virology* **69:** 2989-2997

Jeon S, Lambert PF (1995) Integration of human papillomavirus type 16 DNA into the human genome leads to increased stability of E6 and E7 mRNAs: implications for cervical carcinogenesis. *Proceedings of the National Academy of Sciences of the United States of America* **92:** 1654-1658

Jiang M, Baseman JG, Koutsky LA, Feng Q, Mao C, Kiviat NB, Xi LF (2009) Sequence variation of human papillomavirus type 16 and measurement of viral integration by quantitative PCR. *Journal of clinical microbiology* **47:** 521-526

Jiang X, Hanna Z, Kaouass M, Girard L, Jolicoeur P (2002) Ahi-1, a novel gene encoding a modular protein with WD40-repeat and SH3 domains, is targeted by the Ahi-1 and Mis-2 provirus integrations. *Journal of virology* **76:** 9046-9059

Kalantari M, Blennow E, Hagmar B, Johansson B (2001) Physical state of HPV16 and chromosomal mapping of the integrated form in cervical carcinomas. *Diagn Mol Pathol* **10:** 46-54

Kammer C, Tommasino M, Syrjanen S, Delius H, Hebling U, Warthorst U, Pfister H, Zehbe I (2002) Variants of the long control region and the E6 oncogene in European human papillomavirus type 16 isolates: implications for cervical disease. *British journal of cancer* **86:** 269-273

Kammer C, Warthorst U, Torrez-Martinez N, Wheeler CM, Pfister H (2000) Sequence analysis of the long control region of human papillomavirus type 16 variants and functional consequences for P97 promoter activity. *The Journal of general virology* **81:** 1975-1981

Kirnbauer R, Taub J, Greenstone H, Roden R, Durst M, Gissmann L, Lowy DR, Schiller JT (1993) Efficient self-assembly of human papillomavirus type 16 L1 and L1-L2 into virus-like particles. *Journal of virology* **67:** 6929-6936

Klaes R, Woerner SM, Ridder R, Wentzensen N, Duerst M, Schneider A, Lotz B, Melsheimer P, von Knebel Doeberitz M (1999) Detection of high-risk cervical intraepithelial neoplasia and cervical cancer by amplification of transcripts derived from integrated papillomavirus oncogenes. *Cancer research* **59:** 6132-6136

Kraus I, Driesch C, Vinokurova S, Hovig E, Schneider A, von Knebel Doeberitz M, Durst M (2008) The majority of viral-cellular fusion transcripts in cervical carcinomas cotranscribe cellular sequences of known or predicted genes. *Cancer research* **68:** 2514-2522

Kuslich CD, Chui B, Yamashiro CT (2008) Overview of PCR. *Curr Protoc Essential Lab Tech* **00:** 10.12.11-31

Li N, Franceschi S, Howell-Jones R, Snijders PJ, Clifford GM (2010) Human papillomavirus type distribution in 30,848 invasive cervical cancers worldwide: Variation by geographical region, histological type and year of publication. *International journal of cancer*

Li W, Wang W, Si M, Han L, Gao Q, Luo A, Li Y, Lu Y, Wang S, Ma D (2008) The physical state of HPV16 infection and its clinical significance in cancer precursor lesion and cervical carcinoma. *Journal of cancer research and clinical oncology* **134:** 1355-1361

Longuet M, Beaudenon S, Orth G (1996) Two novel genital human papillomavirus (HPV) types, HPV68 and HPV70, related to the potentially oncogenic HPV39. *Journal of clinical microbiology* **34:** 738-744

Lorenzato M, Bory JP, Cucherousset J, Nou JM, Bouttens D, Thil C, Dez F, Evrard G, Quereux C, Birembaut P, Clavel C (2002) Usefulness of DNA ploidy measurement on liquid-based smears showing conflicting results between cytology and high-risk human papillomavirus typing. *American journal of clinical pathology* **118:** 708-713

Lorenzato M, Clavel C, Masure M, Nou JM, Bouttens D, Evrard G, Bory JP, Maugard B, Quereux C, Birembaut P (2001) DNA image cytometry and human papillomavirus (HPV) detection help to select smears at high risk of high-grade cervical lesions. *The Journal of pathology* **194:** 171-176

Luft F, Klaes R, Nees M, Durst M, Heilmann V, Melsheimer P, von Knebel Doeberitz M (2001) Detection of integrated papillomavirus sequences by ligation-mediated PCR (DIPS-PCR) and molecular characterization in cervical cancer cells. *International journal of cancer* **92:** 9-17

Manchester KM, Heston WD, Donner DB (1993) Tumour necrosis factor-induced cytotoxicity is accompanied by intracellular mitogenic signals in ME-180 human cervical carcinoma cells. *The Biochemical journal* **290 (Pt 1):** 185-190

Matsumoto K, Yoshikawa H, Nakagawa S, Tang X, Yasugi T, Kawana K, Sekiya S, Hirai Y, Kukimoto I, Kanda T, Taketani Y (2000) Enhanced oncogenicity of human papillomavirus type 16 (HPV16) variants in Japanese population. *Cancer letters* **156:** 159-165

Maufort JP, Shai A, Pitot HC, Lambert PF (2010) A role for HPV16 E5 in cervical carcinogenesis. *Cancer research* **70:** 2924-2931

Mays Hoopes LL (2008) Nucleic acid blotting: Southern and Northern. *Curr Protoc Essential Lab Tech* **00:** 8.2.1-24

McBride AA (2008) Replication and partitioning of papillomavirus genomes. *Advances in virus research* **72:** 155-205

McBride AA, Oliveira JG, McPhillips MG (2006) Partitioning viral genomes in mitosis: same idea, different targets. *Cell cycle* **5:** 1499-1502

McIntosh PB, Laskey P, Sullivan K, Davy C, Wang Q, Jackson DJ, Griffin HM, Doorbar J (2010) E1--E4-mediated keratin phosphorylation and ubiquitylation: a mechanism for keratin depletion in HPV16-infected epithelium. *Journal of cell science* **123:** 2810-2822

Meissner JD (1999) Nucleotide sequences and further characterization of human papillomavirus DNA present in the CaSki, SiHa and HeLa cervical carcinoma cell lines. *The Journal of general virology* **80 (Pt 7):** 1725-1733

Metzker ML (2010) Sequencing technologies - the next generation. *Nature reviews genetics* **11:** 31-46

Mincheva A, Gissmann L, zur Hausen H (1987) Chromosomal integration sites of human papillomavirus DNA in three cervical cancer cell lines mapped by in situ hybridization. *Medical microbiology and immunology* **176:** 245-256

Moody CA, Laimins LA (2010) Human papillomavirus oncoproteins: pathways to transformation. *Nature reviews cancer* **10:** 550-560

Munemitsu S, Albert I, Souza B, Rubinfeld B, Polakis P (1995) Regulation of intracellular beta-catenin levels by the adenomatous polyposis coli (APC) tumor-suppressor protein. *Proceedings of the National Academy of Sciences of the United States of America* **92:** 3046-3050

Munger K, Phelps WC, Bubb V, Howley PM, Schlegel R (1989) The E6 and E7 genes of the human papillomavirus type 16 together are necessary and sufficient for transformation of primary human keratinocytes. *Journal of virology* **63:** 4417-4421

Munoz N, Bosch FX, de Sanjose S, Herrero R, Castellsague X, Shah KV, Snijders PJ, Meijer CJ (2003) Epidemiologic classification of human papillomavirus types associated with cervical cancer *The New England journal of medicine* **348:** 518-527

Nasiell K, Roger V, Nasiell M (1986) Behavior of mild cervical dysplasia during long-term follow-up. *Obstetrics and gynecology* **67:** 665-669

Nishikawa K, Rosenblum MG, Newman RA, Pandita TK, Hittelman WN, Donato NJ (1992) Resistance of human cervical carcinoma cells to tumor necrosis factor correlates with their increased sensitivity to cisplatin: evidence of a role for DNA repair and epidermal growth factor receptor. *Cancer research* **52:** 4758-4765

Nurnberg W, Artuc M, Nawrath M, Lovric J, Stuting S, Moelling K, Czarnetzki BM, Schadendorf D (1995) Human c-myb is expressed in cervical carcinomas and transactivates the HPV-16 promoter. *Cancer research* **55:** 4432-4437

Ordonez RM, Espinosa AM, Sanchez-Gonzalez DJ, Armendariz-Borunda J, Berumen J (2004) Enhanced oncogenicity of Asian-American human papillomavirus 16 is associated with impaired E2 repression of E6/E7 oncogene transcription. *The Journal of general virology* **85:** 1433-1444

Park JS, Hwang ES, Park SN, Ahn HK, Um SJ, Kim CJ, Kim SJ, Namkoong SE (1997) Physical status and expression of HPV genes in cervical cancers. *Gynecologic oncology* **65:** 121-129

Park TW, Fujiwara H, Wright TC (1995) Molecular biology of cervical cancer and its precursors. *Cancer* **76:** 1902-1913

Peitsaro P, Johansson B, Syrjanen S (2002) Integrated human papillomavirus type 16 is frequently found in cervical cancer precursors as demonstrated by a novel quantitative real-time PCR technique. *Journal of clinical microbiology* **40:** 886-891

Peter M, Rosty C, Couturier J, Radvanyi F, Teshima H, Sastre-Garau X (2006) MYC activation associated with the integration of HPV DNA at the MYC locus in genital tumors. *Oncogene* **25:** 5985-5993

Pfreundschuh M, Genth B, Kirchner H, Scheurich P, Lindemann A, Steinmetz HT, Schaadt M, Diehl V (1989) [Mechanism of resistance to tumor necrosis factor]. *Onkologie* **12:** 128-130

Ramsay RG, Gonda TJ (2008) MYB function in normal and cancer cells. *Nature reviews cancer* **8:** 523-534

Reuter S (1995) Identifizierung eines neuen Zinkfingergens im Integrationsbereich von Papillomavirus-DNA im Genom der Zervixkarzinomzellinie ME180 und Charakterisierung der viralen Integratstruktur. Dr. rer. nat. Thesis, Naturwissenschaftlich-Mathematischen Gesamtfakultät, Ruprecht-Karls-Universität Heidelberg, Heidelberg

Reuter S, Bartelmann M, Vogt M, Geisen C, Napierski I, Kahn T, Delius H, Lichter P, Weitz S, Korn B, Schwarz E (1998) APM-1, a novel human gene, identified by aberrant co-transcription with papillomavirus oncogenes in a cervical carcinoma cell line, encodes a BTB/POZ-zinc finger protein with growth inhibitory activity. *The EMBO journal* **17:** 215-222

Reuter S, Delius H, Kahn T, Hofmann B, zur Hausen H, Schwarz E (1991) Characterization of a novel human papillomavirus DNA in the cervical carcinoma cell line ME180. *Journal of virology* **65:** 5564-5568

Robertson G, Bilenky M, Lin K, He A, Yuen W, Dagpinar M, Varhol R, Teague K, Griffith OL, Zhang X, Pan Y, Hassel M, Sleumer MC, Pan W, Pleasance ED, Chuang M, Hao H, Li YY, Robertson N, Fjell C, Li B, Montgomery SB, Astakhova T, Zhou J, Sander J, Siddiqui AS, Jones SJ (2006) cisRED: a database system for genome-scale computational discovery of regulatory elements. *Nucleic acids research* **34:** D68-73

Romanczuk H, Howley PM (1992) Disruption of either the E1 or the E2 regulatory gene of human papillomavirus type 16 increases viral immortalization capacity. *Proceedings of the National Academy of Sciences of the United States of America* **89:** 3159-3163

Saunier M, Monnier-Benoit S, Mauny F, Dalstein V, Briolat J, Riethmuller D, Kantelip B, Schwarz E, Mougin C, Pretet JL (2008) Analysis of human papillomavirus type 16 (HPV16) DNA load and physical state for identification of HPV16-infected women with high-grade lesions or cervical carcinoma. *Journal of clinical microbiology* **46:** 3678-3685

Schiffman M, Castle PE, Jeronimo J, Rodriguez AC, Wacholder S (2007) Human papillomavirus and cervical cancer. *Lancet* **370:** 890-907

Schiffman M, Rodriguez AC, Chen Z, Wacholder S, Herrero R, Hildesheim A, Desalle R, Befano B, Yu K, Safaeian M, Sherman ME, Morales J, Guillen D, Alfaro M, Hutchinson M, Solomon D, Castle PE, Burk RD (2010) A population-based prospective study of carcinogenic human papillomavirus variant lineages, viral persistence, and cervical neoplasia. *Cancer research* **70:** 3159-3169

Schmitt M, Bravo IG, Snijders PJ, Gissmann L, Pawlita M, Waterboer T (2006) Bead-based multiplex genotyping of human papillomaviruses. *Journal of clinical microbiology* **44:** 504-512

Schneider-Gadicke A, Schwarz E (1986) Different human cervical carcinoma cell lines show similar transcription patterns of human papillomavirus type 18 early genes. *The EMBO journal* **5:** 2285-2292

Schwarz E, Freese UK, Gissmann L, Mayer W, Roggenbuck B, Stremlau A, zur Hausen H (1985) Structure and transcription of human papillomavirus sequences in cervical carcinoma cells. *Nature* **314:** 111-114

Seedorf K, Krammer G, Durst M, Suhai S, Rowekamp WG (1985) Human papillomavirus type 16 DNA sequence. *Virology* **145:** 181-185

Shendure J, Ji H (2008) Next-generation DNA sequencing. *Nature biotechnology* **26:** 1135-1145

Smits HL, Cornelissen MT, Jebbink MF, van den Tweel JG, Struyk AP, Briet M, ter Schegget J (1991) Human papillomavirus type 16 transcripts expressed from viral-cellular junctions and full-length viral copies in CaSki cells and in a cervical carcinoma. *Virology* **182:** 870-873

Smotkin D, Wettstein FO (1986) Transcription of human papillomavirus type 16 early genes in a cervical cancer and a cancer-derived cell line and identification of the E7 protein. *Proceedings of the National Academy of Sciences of the United States of America* **83:** 4680-4684

Snijders PJ, Steenbergen RD, Heideman DA, Meijer CJ (2006) HPV-mediated cervical carcinogenesis: concepts and clinical implications. *The Journal of pathology* **208:** 152-164

Solomon D, Davey D, Kurman R, Moriarty A, O'Connor D, Prey M, Raab S, Sherman M, Wilbur D, Wright T, Jr., Young N (2002) The 2001 Bethesda System: terminology for reporting results of cervical cytology. *Jama* **287:** 2114-2119

Spalholz BA, Yang YC, Howley PM (1985) Transactivation of a bovine papilloma virus transcriptional regulatory element by the E2 gene product. *Cell* **42:** 183-191

Stanley MA, Browne HM, Appleby M, Minson AC (1989) Properties of a non-tumorigenic human cervical keratinocyte cell line. *International journal of cancer* **43:** 672-676

Steinmeyer N (2009) Mutationen und Integration von DNA humaner Hochrisiko-Papillomaviren in anogenitalen Krebsvorstufen und Karzinomen. Diplom Thesis, Fachbereich Chemie- und Biotechnologie im Studiengang Biotechnologie, Hochschule Darmstadt, Darmstadt

Stenlund A (2003) Initiation of DNA replication: lessons from viral initiator proteins. *Nat Rev Mol Cell Biol* **4:** 777-785

Strauss WM (2001) Preparation of genomic DNA from mammalian tissue. *Curr Protoc Mol Biol 00***:**2.2.1-3

Swan DC, Rajeevan M, Tortolero-Luna G, Follen M, Tucker RA, Unger ER (2005) Human papillomavirus type 16 E2 and E6/E7 variants. *Gynecologic oncology* **96:** 695-700

Sykes JA, Whitescarver J, Jernstrom P, Nolan JF, Byatt P (1970) Some properties of a new epithelial cell line of human origin. *Journal of the National Cancer Institute* **45:** 107-122

Thierry F (2009) Transcriptional regulation of the papillomavirus oncogenes by cellular and viral transcription factors in cervical carcinoma. *Virology* **384:** 375-379

Thierry F, Yaniv M (1987) The BPV1-E2 trans-acting protein can be either an activator or a repressor of the HPV18 regulatory region. *The EMBO journal* **6:** 3391-3397

Thorland EC, Myers SL, Gostout BS, Smith DI (2003) Common fragile sites are preferential targets for HPV16 integrations in cervical tumors. *Oncogene* **22:** 1225-1237

Trottier H, Franco EL (2006) The epidemiology of genital human papillomavirus infection. *Vaccine* **24 Suppl 1:** S1-15

Valente EM, Brancati F, Silhavy JL, Castori M, Marsh SE, Barrano G, Bertini E, Boltshauser E, Zaki MS, Abdel-Aleem A, Abdel-Salam GM, Bellacchio E, Battini R, Cruse RP, Dobyns WB, Krishnamoorthy KS, Lagier-Tourenne C, Magee A, Pascual-Castroviejo I, Salpietro CD, Sarco D, Dallapiccola B, Gleeson JG (2006) AHI1 gene mutations cause specific forms of Joubert syndrome-related disorders. *Annals of neurology* **59:** 527-534

van Es JH, Kirkpatrick C, van de Wetering M, Molenaar M, Miles A, Kuipers J, Destree O, Peifer M, Clevers H (1999) Identification of APC2, a homologue of the adenomatous polyposis coli tumour suppressor. *Curr Biol* **9:** 105-108

Van Tine BA, Kappes JC, Banerjee NS, Knops J, Lai L, Steenbergen RD, Meijer CL, Snijders PJ, Chatis P, Broker TR, Moen PT, Jr., Chow LT (2004) Clonal selection for transcriptionally active viral oncogenes during progression to cancer. *Journal of virology* **78:** 11172-11186

Veress G, Szarka K, Dong XP, Gergely L, Pfister H (1999) Functional significance of sequence variation in the E2 gene and the long control region of human papillomavirus type 16. *The Journal of general virology* **80 (Pt 4):** 1035-1043

Vinokurova S, Wentzensen N, Kraus I, Klaes R, Driesch C, Melsheimer P, Kisseljov F, Durst M, Schneider A, von Knebel Doeberitz M (2008) Type-dependent integration frequency of human papillomavirus genomes in cervical lesions. *Cancer research* **68:** 307-313

Vizcaino AP, Moreno V, Bosch FX, Munoz N, Barros-Dios XM, Parkin DM (1998) International trends in the incidence of cervical cancer: I. Adenocarcinoma and adenosquamous cell carcinomas. *International journal of cancer* **75:** 536-545

von Knebel Doeberitz M, Bauknecht T, Bartsch D, zur Hausen H (1991) Influence of chromosomal integration on glucocorticoid-regulated transcription of growth-stimulating papillomavirus genes E6 and E7 in cervical carcinoma cells. *Proceedings of the National Academy of Sciences of the United States of America* **88:** 1411-1415

Walboomers JM, Jacobs MV, Manos MM, Bosch FX, Kummer JA, Shah KV, Snijders PJ, Peto J, Meijer CJ, Munoz N (1999) Human papillomavirus is a necessary cause of invasive cervical cancer worldwide. *The Journal of pathology* **189:** 12-19

Wang Q, Griffin H, Southern S, Jackson D, Martin A, McIntosh P, Davy C, Masterson PJ, Walker PA, Laskey P, Omary MB, Doorbar J (2004) Functional analysis of the human papillomavirus type 16 E1=E4 protein provides a mechanism for in vivo and in vitro keratin filament reorganization. *Journal of virology* **78:** 821-833

Watts KJ, Thompson CH, Cossart YE, Rose BR (2002) Sequence variation and physical state of human papillomavirus type 16 cervical cancer isolates from Australia and New Caledonia. *International journal of cancer* **97:** 868-874

Wellmann A, Fogt F, Hollerbach S, Hahne J, Koenig-Hoffmann K, Smeets D, Brinkmann U (2010) Polymorphisms of the apoptosis-associated gene DP1L1 (deleted in polyposis 1-like 1) in colon cancer and inflammatory bowel disease. *Journal of cancer research and clinical oncology* **136:** 795-802

Wentzensen N, Ridder R, Klaes R, Vinokurova S, Schaefer U, Doeberitz MK (2002) Characterization of viral-cellular fusion transcripts in a large series of HPV16 and 18 positive anogenital lesions. *Oncogene* **21:** 419-426

Wentzensen N, Vinokurova S, von Knebel Doeberitz M (2004) Systematic review of genomic integration sites of human papillomavirus genomes in epithelial dysplasia and invasive cancer of the female lower genital tract. *Cancer research* **64:** 3878-3884

Werness BA, Levine AJ, Howley PM (1990) Association of human papillomavirus types 16 and 18 E6 proteins with p53. *Science* **248:** 76-79

WHO/ICO Information Centre on HPV and Cervical Cancer (HPV Information Centre). (2010) Human Papillomavirus and Related Cancers in World. Summary Report Update 2010.

Woodman CB, Collins SI, Young LS (2007) The natural history of cervical HPV infection: unresolved issues. *Nature reviews* **7:** 11-22

Wu X, Zhang C, Feng S, Liu C, Li Y, Yang Y, Gao J, Li H, Meng S, Li L, Zhang Y, Hu X, Wu X, Lin L, Li X, Wang Y (2009) Detection of HPV types and neutralizing antibodies in Gansu province, China. *Journal of medical virology* **81:** 693-702

Xu B (2010) Integration of Human Papillomavirus Type 16 DNA in Cervical Carcinogenesis: Design of a novel strategy for HPV16 integration site determination in cervical scrapes and analysis of HPV16-induced c-myc insertional mutagenesis. Dr. rer. nat. Thesis, Fakultät für Biowissenschaften, Ruprecht-Karls-Universität Heidelberg, Heidelberg

Yamada T, Manos MM, Peto J, Greer CE, Munoz N, Bosch FX, Wheeler CM (1997) Human papillomavirus type 16 sequence variation in cervical cancers: a worldwide perspective. *Journal of virology* **71:** 2463-2472

Yamada T, Wheeler CM, Halpern AL, Stewart AC, Hildesheim A, Jenison SA (1995) Human papillomavirus type 16 variant lineages in United States populations characterized by nucleotide sequence analysis of the E6, L2, and L1 coding segments. *Journal of virology* **69:** 7743-7753

Yu T, Ferber MJ, Cheung TH, Chung TK, Wong YF, Smith DI (2005) The role of viral integration in the development of cervical cancer. *Cancer genetics and cytogenetics* **158:** 27-34

Zehbe I, Wilander E, Delius H, Tommasino M (1998) Human papillomavirus 16 E6 variants are more prevalent in invasive cervical carcinoma than the prototype. *Cancer research* **58:** 829-833

Zhang L, Murray F, Zahno A, Kanter JR, Chou D, Suda R, Fenlon M, Rassenti L, Cottam H, Kipps TJ, Insel PA (2008) Cyclic nucleotide phosphodiesterase profiling reveals increased expression of phosphodiesterase 7B in chronic lymphocytic leukemia. *Proceedings of the National Academy of Sciences of the United States of America* **105:** 19532-19537

Zheng ZM, Baker CC (2006) Papillomavirus genome structure, expression, and post-transcriptional regulation. *Front Biosci* **11:** 2286-2302

Ziegert C, Wentzensen N, Vinokurova S, Kisseljov F, Einenkel J, Hoeckel M, von Knebel Doeberitz M (2003) A comprehensive analysis of HPV integration loci in anogenital lesions combining transcript and genome-based amplification techniques. *Oncogene* **22:** 3977-3984

zur Hausen H (1999) Immortalization of human cells and their malignant conversion by high risk human papillomavirus genotypes. *Seminars in cancer biology* **9:** 405-411

zur Hausen H (2000) Papillomaviruses causing cancer: evasion from host-cell control in early events in carcinogenesis. *Journal of the National Cancer Institute* **92:** 690-698

zur Hausen H (2002) Papillomaviruses and cancer: from basic studies to clinical application. *Nature reviews cancer* **2:** 342-350

Appendix

A1. Nucleotide sequences of integrated HPV68b in ME180 and ME180R

HPV68b(int)-ME180

The sequence is described in Results section 2.1.1, including Figures 2.3-2.4, and Table 2.1. Black color represents cellular sequence of chromosome 18. Red color and blue color represent the 5'copy and 3'copy of the integrated HPV68b respectively. Underlined nucleotides indicate overlapping areas between viral and cellular DNA.

```
   1 ACACACCAAT GGAGAGGAGT GCCACCGCAG ATGGCTGGCA CCCTGGCCGG GTTGGCCATG
  61 GAAGTCTGCG GCAGGCAAAT CAAATGGTGC CCCTTCAAGC TGTTGTTCTT TAATTATTTT
 121 TTCATCCTTT ATTTTCTAGG GGGCAGTGGG GGGAGACTTG TTTTCTTTTA AGAAAAGTTG
 181 CACACCATGT TTTTTAAAAC ATTGCTTTGG CTTTGTATGC CATAGTTCAT TACTTGTGTC
 241 CCTTAATGTC CACTCCTCTG CACTATATGC AGTTTGAGCA ATGCTCTCTA GTGCCATCTG
 301 CAGTTCAATG GCTTGATATG CTTTAGTTTT TGAAATGTTT ACAGGAGGCA CCACCTGGTG
 361 GTCAATATTA TGCATACCAC GTTCTCGTGC TGCATAATAT ATTGCATTTT CCAGTCGTAT
 421 ACAGTTCCAA TAGTTAATAT GGTCCTTTAT ACATTTACTG TCCTGTTCAT AATGTTCTAA
 481 TATTTTCTCC TGTAACACAT TTAAACGTTG GGAAAGTGTT TTCATCATTG TCTCCTTCAT
 541 CCTCGTCCTG CTGCAAGTCT AATTTGCACC AAGTCTTTTC AAAAAAACAT TTCCAGTTTT
 601 TATCATTGAT TGTATACACT GGGTTCCTGT TTTGGTCAAA TGGAAATGCA TTAGGAAATT
 661 TAGACACAGT TAGTCTGCTA TGTAAATACG GCCACCTATT GTCTTCCACA GGGTTGGTAT
 721 TTGATGTTAT TAGCATTGGT GGACATTTTA TTTGTATTAG GTGTCTGTGT TTTCTATCTA
 781 AACTTATTGG GTTACCATCT AATGCATTTC TCATGTAATT ATCTAAATAT GACCAACATG
 841 TACCTGTTGC GTCATCTAAC ATGGCTATTT TTGCATCTGC AAGTGGCTCT AACCAAAAGT
 901 GACTAGCTGA ATTTACATAT GAAATTATTG TGCCTTGCAA GAAATGTATA AGACTCATGC
 961 AAAAATATGA CTTGCCTGTA TTTGGTGGCC CATGTATAAC TATACAATTA CGTTTTGGCG
1021 TGCCTTTTAA AAAATCCTTT AATGCACATA AAAATGTTAT AAATTCTAGC CCCTGATATC
1081 TTAAAAACTG TACAATTGGT CGCCAATCAC CGCCTTCATC ACATTTACTG CATCTAAATT
1141 TAATCCATTG CGGCATTGAC ATTTGTCGTT TTTGTGCCCG TTTGTAATGT CTACACATTG
1201 TTGCACAATC TTTTACATAT TTTGCTTGAC AGTTGCTTTT TAAAAACGCT GCAGCATTAC
1261 TATTACAATC TGCCAACATA GCATATGAAA ATGCTATATC ACTTTCATCT GTTAACTCAT
1321 TATCAAATGC CCATTGTACC ATGTCTGATA GATCAAATAC ACTATCATCT ATTCCATGTT
1381 GTATTATAGT TAATATTTTT ATCCATTCTG GCGTGTTTTC ACACACCTCA CTAATATTAG
1441 ATATTCCTGT TCTATACCAA TACAATGCTG CAACAGGGCT ACGCAATTTT GGTGGCTGCA
1501 AAAGCATACA GCTGTCTGGA ACATGCAACA ATGTACTTAA TCCTTTTCCT ACTGTTATTC
1561 TATTTTTCCC ACATTTGTAT CTTATTAACA TTAATATTAA TATTCCGTTT TTTGTATCTA
1621 AACATTGTAT ATGGGTATAT AATGCATATT GTTTAATTAG TGTTTTAAAC CCTTCGGCAA
1681 TGGTTGGATT TACTCCGAAT ATTGCTGCTA CCCAGTCCGT GCATGTGGTC TTATCACTTT
1741 TAAATGTACG TACTAGGTCA TTAAAGGGCA ATCCATATAC TTTTTTAAAT TCTGTTAACA
1801 TTGCAGCTTT TTTATTATTA CATTGTAATA ATACTTTTAG TTGCGTAGTA GGTGATTTAG
1861 GATCCTGGTT TTCACTATCT ATAGCACTGT CTACACTACT ACAGTCCTCC CGTATGCTGT
1921 CGCCATTTTC CCCTTCATCC TCCCCGTCCG CCCCATTTGT ATTAGTTGCT ACAGTTACCT
1981 CCGAGTTAGT TTCCACTTCC ATATTGATAG TTCCTGTAAT GGCGACTTTG CTAAAGGGCT
2041 GCTTTCTATA CTGTCTGTAT ACTTTCGTTT TAGGGCACGC ACTGTTTGTG CATCCCTTTG
2101 GGCCTGTTGC ATATTTAACA GTACCTGCGC TGTCTCACGC TCTGCCTGTA TACAAATATC
2161 TGTAGCATCA TCAATGAAAT CTACCATGTC TGAACCTGTA TCTGTCGCGT TTTCATCCTC
2221 ATCCTCTGAG ACTGTGTCAC CTGTTTGTTT ATCTACTATT GCTTGTACAA AAAACCATCC
2281 GTTACACCCC GTCCCGTCCC CATCTGTACC TTCACAATTG GCCATTGCAG ATTACTGGGT
2341 TTCCGTTGCA CACCACGGAC ACACAAAATT TAGTGAGTCC ATAAACAGCA GTTCTAGCTT
2401 CCGCAGGTTC TCCCGCGACG CTTCTACTAC TAGTTGCAGT AGGTTGTTAC ACTTACAACA
2461 CGTACACTGA ATTGTGTGAC GCTGTTGTTC GTCCCGTCTG GCTAGTAGTT GATGTTGGTG
2521 GTGATTAACT GCATGGTCGG GTTCATCTAT TTCGTCGTCT GAATCTCCTA ATTGCTCGTG
2581 ACATACAAGG TCGACCGGCT CTATTTCATT GCATGGACAT AACTCTAACA CAATTTCCTG
2641 CACGGTGGGC TTTGGTCCAT GCATAGTTAG TTTATACTTG TGTTTCCTGC CGTGTGCGTC
2701 TGCGGTCCTC TCGTTTACTG GTCCAGCAGT GGCGACACTG TCCTGTAAAG TTTCCTGCTA
2761 TTTTATGAAA TCTTCGTTTT GAATTTAGGT GCCTTAGTTT TTCAGCAGGA CTCAATGGTT
2821 TCAGGCAACA CATGCACCTT ATTGATAAAT CATATAACTT TGTATTAGTT ATGGTTTCTA
2881 ATGTTGTTGC ATACACCGAT TCTGAGTAAT ATCGTAGTTC CCGCATTTTC GCATAAAATT
2941 TAATACATGA TTGGCATGCA GCTAATGGTA CCCCGTCCCT ATATACTACA TTTAAGTCAC
3001 CAAAGGCAAA TTCATATACC TCTGTCCGTT GTAGTTGCCT TCTGCAATAG ACACAGTCTA
3061 TTGTAACGTC ATGCAATGTG GTGTCCAATG TCCTGCACAG GTCTGGCAAT TTGTATGGCC
3121 GTTCCTCAGG GTTGTGAAAT AGCGCCATTG GTATAGACAA CTGCTGTGTT CAGCTTTATA
3181 TACACCGTTT TCGGTCATGA CCGTTTTCGG TCACTCCCTT TATACTATAA AATGATTGTA
3241 AAATGTTATA AGTAAAAGTA TTGGTATGTG TGCCCAACCT ATTTCGGTTG CACAAATTTA
3301 TGGATGTGGT ATTAGTCACT GTATAACTTA CACTACAAGA CGCATTCTTG AATGCAAAAG
3361 TAGTTATACA GGCCAACTAT GCTATTAGGC AAGTAAAAAC AGTTAAAAAC CTATGCCTAA
3421 AAGCAGTTTT ATTACAAGGG TGGAGATATA TAGGCTGCCA AACTATTGTT GCACTGCACC
3481 TGGACAGGAT GATGACTAAG TAAGGTGCGC CAGTACATAC TGTTACCGGT GCCAACTATG
3541 TGGGTAAACC AAGGTGAAAC ATGTTTTTGC TGAAACACAT AGTTTTTAAT ACTGTATAGA
3601 AGGAAATAGC ACCACCGACC GAAAACGGTT GCACACAAAA TGGCCTTACA AAATGGCCAC
3661 TTGGTATAAT ATAGAACTAT ATATATTACA TGTACTGTAA CACACCTTAG GGTAGGGCTA
3721 CAAATTATGT AGGAAATGTT GCAGTCCTAT GATATGTAAA ACAAGGACAT ATGTTAGTCA
3781 CAGGGTGCAA CCACAAAGTA AACTGACATA CATACTTTAT TAACAAAAAC AGATTATACA
3841 AACATACCCG CAAACATACA CATATACATA TACACATACA TGCACAACAT ACCAACAACA
3901 ACATGACACG TATACAACCA CACATACAAC CAACATACAA AACAAAACAT ACAACAATTA
3961 CTTTGACACA CGTTTACGTT TGTGCTTAGA GGTAGATGCA GTAGTTGCTG TGGCAGGGCG
4021 TTTACGGGGG CCTATAGTGG GTCGTCGGCG GACACCTGCC TGTAAAAGAA ATTTGCGTCC
4081 TAAAGGAAAC TGGTCCAGTT CAGAACTAAA CTTTTCCTTT AAATTTACAT TCCAAAAGTT
4141 TAAGCCATCA TATGGATCCT TTTTAGTAGG TGCAGGGGCG TCTTTTTGAC ATGTAATTGC
4201 TGCTGATTGC AGATAGCGGT ATGTATCTAC AAGACTAGCA GATGGTGGAG GGGCAACACC
4261 AAAATTCCAA TCACCCAAAA TAGCAGGATT CATAGTATGT ATATAGGACA TTACATCAGT
4321 GGACAATGTT ATAGTACACA ACTGAAATAT AAATTGCAAA TCATATTCCT CAACATGCCT
4381 AATATATTCC TTAAATTTAT TAGGATCATA AATATTTGGT ACAGCTGATT CAGTAGTAGT
4441 AGACAAAGTA AAATTGGTAC TGCGAGTGGT ATCCACAACA GTAAGAAATA ATTGATTATG
4501 CCAACAAATA CCATTGTTGT GTCCCTGTGC CTTGTGCAGC CAATAGGGCT TGTTAAATAA
4561 CTGGGAGTCT GAGGATACCA TAGACCCACT AGGCGAGGGG GCATATACAT AACTACTAGG
4621 ACTGTCACGT ATGTCAGTGC CCTTAATATA CAATTCAGTA GGTATAGTGT CCCCTACCAT
4681 GCCCCCTCTA TTCCAAAAAT GCCTAGCAAA TAACTGTTCC CTACGTAAAC AAAAGAACAT
4741 ACTGTCTCCA TATACATCTG CAGACATTTG TAAATAGTCA GGATATTTGC AGACTGATTG
4801 ACATATATCT AAAGGCACCT CGCTTTTTGT TTCTTGTAAT GTACTAAAGT CCATAGCACC
4861 ATATCCTGTA TCAATCATAT CGCCATCCTG AATAGGTGTA TTTACTAATT CCAATGGTGG
4921 ACAGTCCCCG GGCTGCACAT TGCTAGGCTT ACAAGATTTA CCTTTGGCCC AGTGCTCCCC
4981 AATGGCAGGA ACACAGCCTA TAATACATAG TTGCGTTTGT TTATAGTCCA CTGAAACATT
5041 GTCCCTACTG TCCTTAGGAT TTTTGTTGGA GGAAAACGGG GAATTTTCAG TATCATCTAG
5101 CCTATTATAT AATGGATGCC CACTAAGGCC AACACCTAAT GGCTGCCCCC TACCTATTTC
5161 AACACCAACA CAGGCCCATA CCAATCGCTG CGTATCAGGG TTATATAATG TAGACTCAGG
5221 AAGACTAAAT TTATTAGGAT CAGGTAGGGA AATCCTAAAC ACCCTGTATT GATATGCAGA
5281 CACCTTAGGA ATGTCCTGCT TGCGGCCCCC AGACATAGGG ACCTTAAAAT ATGGATGGCC
5341 TACAGTTAAT AACCTAGATG TACCAGCATA GTAATAAATG CCAGTGCGTG TTACGTAATC
5401 ATCTGTATTG ACAACCTTCG CCACTGAGGG GGGAGGCAAA TACACCATGT TGTCGCTAGA
5461 GCGCCACAAT GCCATCTGCA AAAAAATAAG GAAGGCGTTT ACGTTTTTTT AATAAAAAGA
5521 ACAATAATGG TAATAAATAA TAATTGGTGC CATATATAGT TATGGCATAG GTTGTATCAA
5581 TTGGTGTAGA GGGTGTTAAA GGCAACTGTG GAGACGTTGC TGGTAACACA ACATCAGGAC
5641 CAGTATTTAC AGGCGTGTTC CAAGCAGTAC CAATAGGAAT AGTAGTGTTA GCATATGTAG
5701 TAGATGCTGT AGACGCTAAT GAAGGAACAG ATATATGGGA ACGGGAGGTA AATGTAGCAT
5761 TATGGAATGC AGTATCCAAT ACTGTAGTAT TGTCAGTATC TGATGCATAT ATATCATATA
5821 AAGTATCCAT AGGGTCAGAC TGCTCTGGGG CAACCAAAGG TTGTAGTTCA ATGCTGTCAG
5881 CAGGAGCAAT GCCACTAATA TCATGATAAT AGTGCACCTG TGCCCCAATT TGTGTACCCC
5941 GGCGTGTAAA CATAGTTGCC TTTTTGCCTA CTCTGCTAAA ACGTACTGTG CCTCTTCGGG
6001 AAGTTAAGGC AGGCCTATGT AAACGAACAA TGTCCAGAAA ATCCGGATCA GGAGCTATGT
6061 CAGCAGGTTC ATATGTAAGT GTAGTATCAA CAGGCTCAAA AGCAGGATTA TCAAATGTTA
6121 CAAATGATGA AGGGTGAGTT ACAAAATCAA AATTACTAAC ACGAACCTGT TGATGTGCCC
6181 TACTATATAA ACGTGGCCCT GCCACACGAC TAACCCCAGG TATAGGTGTA CTACTAATAG
6241 GTTCTGTACC AGTGCCATGT GTTGCAAATA CCTGCATAGG TATTTCTTCA TATCCATGTG
6301 TTCCCGATGT GGGGGTACTT ACAAACACAT TACCAGAGAC TTCACCTGTT TGAGGCACTT
6361 CTATAATAGT GGGGTCTGCA AATGCAGGGT TAGTAAAACT AGTACTGCTT ACTTGCACAG
6421 ACCCAGACGA AGGGGTAATG TCTAACACAG CAGGTGTAGT GGTAGAAGAA GATGTAATTT
6481 CAAACCCAGA AGTGTCTGTA AATGTTGGTA CCGGTGTGCC AGATGTAATA ACACTGGAAT
6541 CTTCCACCAA TTGCACAATG GAGGGTTCTG TAGGACCCAC AGGTTCAATA ACCACAGGTG
6601 GACGTGCAGG CGAAACATCT ACAACAGTAT TAGGTTTACC ACCTAAAGGA ATGTACCCAG
6661 TACGACCCCC GGTTCCTGAC CCAGTACCAA TGCCTAGGCC ACCCAAAAAA ATACCTAAAC
6721 TGGTCCATTG CAATAGTTTG TCTGCAAGTG TGGTGCCTTC AACCTTATTT ATAACATCAG
6781 GAGGACATGT GCCTGATTGT TTGCATGTTT TATATAATTC AGTTGCAGAT GCACGCTTGC
6841 GCCTGGCAGC ACGGTGTGAT ACCATATTTA TTTACAAAAA TATACACCAA TACAAATGCA
6901 CAATACAAAA CTTAAGGCAT ACTATAACGA GCAAAACTGT GTAATACCCA CACAGGCAGT
6961 AAAAAAAAAA GTATATATAC AGCAAAGACC TCCAATGGTG TGGTACGTAC TAATATAAAC
```

```
 7021 ACAAACACAA GTATCCACAC ATACACACAC ACATGCATGG ACTGCAGAAG CGGGACAGTG
 7081 CAACATATAT ACATGCACAC ACAAAACCAC ACCAAAAATA CCAGTACAAT CATATGTACA
 7141 TGTATGGTGG TATGTATCCA ATATACATAC AGTATATGTA TGTTGCACTG TCCCGCTTCT
 7201 GCAGTCCATG CATGTGTGTG TGTATGTGTG GATACTTGTG TTTGTGTTTA TATTAGTACG
 7261 TACCACACCA TTGGAGGTCT TTGCTGTATA TATACTTTTT TTTTTACTGC CTACGTGGGT
 7321 ATTACACAGT TTTGCTCGTT ATAGTATGCC TTAAGTTTTG TATTGTGCAT TTGTATTGGT
 7381 GTATATTTTT ATAAATAAAT ATGGTATCAC ACCGTGCTGC CAGGCGCAAG CGTGCATCTG
 7441 CAACTGAATT ATATAAAACA TGCAAACAAT CAGGCACATG TCCTCCTGAT GTTATAAATA
 7501 AGGTTGAAGG CACCACACTT GCAGACAAAC TATTGCAATG GACCAGTTTA GGTATTTTTT
 7561 TGGGTGGCCT AGGCATTGGT ACTGGGTCAG GAACCGGGGG TCGTACTGGG TACATTCCTT
 7621 TAGGTGGTAA ACCTAATACT GTTGTAGATG TTTCGCCTGC ACGTCCACCT GTGGTTATTG
 7681 AACCTGTGGG TCCTACGGAA CCCTCCATTG TGCAATTGGT GGAAGATTCC AGTGTTATTA
 7741 CATCTGGCAC ACCGGTACCA ACATTTACAG GCACTTCTGG GTTTGAAATT ACATCTTCTT
 7801 CTACCACTAC ACCTGCTGTG TTAGACATTA CCCCTTCGTC TGGGTCTGTG CAAGTAAGCA
 7861 GTACTAGTTT TACTAACCCT GCATTTGCAG ACCCCACTAT TATAGAAGTG CCTCAAACAG
 7921 GTGAAGTCTC TGGTAATGTG TTTGTAAGTA CCCCCACATC GGGAACACAT GGATATGAAG
 7981 AAATACCTAT GCAGGTATTT GCAACACATG GCACTGGTAC AGAACCTATT AGTAGTACAC
 8041 CTATACCTGG GGTTAGTCGT GTGGCAGGGC CACGTTTATA TAGTAGGGCA CATCAACAGG
 8101 TTCGTGTTAG TAATTTTGAT TTTGTAACTC ACCCTTCATC ATTTGTAACA TTTGATAATC
 8161 CTGCTTTTGA GCCTGTTGAT ACTACACTTA CATATGAACC TGCTGACATA GCTCCTGATC
 8221 CGGATTTTCT GGACATTGTT CGTTTACATA GGCCTGCCTT AACTTCCCGA AGAGGCACAG
 8281 TACGTTTTAG CAGAGTAGGC AAAAAGGCAA CTATGTTTAC ACGCCGGGGT ACACAAATTG
 8341 GGGCACAGGT GCACTATTAT CATGATATTA GTGGCATTGC TCCTGCTGAC AGCATTGAAC
 8401 TACAACCTTT GGTTGCCCCA GAGCAGTCTG ACCCTATGGA TACTTTATAT GATATATATG
 8461 CACCAGATAC TGACAATACT ACAGTATTGG ATACTGCATT CCATAATGCT ACATTTACCT
 8521 CCCGTTCCCA TATATCTGTT CCTTCGTTAG CGTCTACAGC ATCTACTACA TATGCTAACA
 8581 CTACTATTCC TATTGGTACT GCTTGGAACA CGCCTGTAAA TACTGGTCCT GATGTTGTGT
 8641 TACCAGCAAC GTCTCCACAG TTGCCTTTAA CACCCTCTAC ACCAATTGAT ACAACCTATG
 8701 CCATAACTAT ATATGGCACC AATTATTATT TATTACCATT ATTGTTCTTT TTATTAAAAA
 8761 AACGTAAACG CCTTCCTTAT TCTTTTGCAG ATGGCATTGT GGCGCTCTAG CGACAACATG
 8821 GTGTATTTGC CTCCCCCCTC AGTGGCGAAG GTTGTCAATA CAGATGATTA CGTAACACGC
 8881 ACTGGCATTT ATTACTATGC TGGTACATCT AGGTTACTAA CTGTAGGCCA TCCATATTTT
 8941 AAGGTCCCTA TGTCTGGGGG CCGCAAGCAG GACATTCCTA AGGTGTCTGC ATATCAATAC
 9001 AGGGTGTTTA GGATTTCCCT ACCTGATCCT AATAAATTTA GTCTTCCTGA GTCTACATTA
 9061 TATAACCCTG ATACGCAGCG ATTGGTATGG GCCCGTGTTG GTGTTGAAAT AGGTAGGGGG
 9121 CAGCCATTAG GTGTTGGCCT TAGTGGGCAT CCATTATATA ATAGGCTAGA TGATACTGAA
 9181 AATTCCCCGT TTTCCTCCAA CAAAAATCCT AAGGACAGTA GGGACAATGT TTCAGTGGAC
 9241 TATAAACAAA CGCAACTATG TATTATAGGC TGTGTTCCTG CCATTGGGGA GCACTGGGCC
 9301 AAAGGTAAAT CTTGTAAGCC TAGCAATGTG CAGCCCGGGG ACTGTCCACC ATTGGAATTA
 9361 GTAAATACAC CTATTCAGGA TGGCGATATG ATTGATACAG GATATGGTGC TATGGACTTT
 9421 AGTACATTAC AAGAAACAAA AAGCGAGGTG CCTTTAGATA TATGTCAATC AGTCTGCAAA
 9481 TATCCTGACT ATTTACAAAT GTCTGCAGAT GTATATGGAG ACAGTATGTT CTTTTGTTTA
 9541 CGTAGGGAAC AGTTATTTGC TAGGCATTTT TGGAATAGAG GGGGCATGGT AGGGGACACT
 9601 ATACCTACTG AATTGTATAT TAAGGGCACT GACATACGTG ACAGTCCTAG TAGTTATGTA
 9661 TATGCCCCCT CGCCTAGTGG GTCTATGGTA TCCTCAGACT CCCAGTTATT TAACAAGCCC
 9721 TATTGGCTGC ACAAGGCACA GGGACACAAC AATGGTATTT GTTGGCATAA TCAATTATTT
 9781 CTTACTGTTG TGGATACCAC TCGCAGTACC AATTTTACTT TGTCTACTAC TACTGAATCA
 9841 GCTGTACCAA ATATTTATGA TCCTAATAAA TTTAAGGAAT ATATTAGGCA TGTTGAGGAA
 9901 TATGATTTGC AATTTATATT TCAGTTGTGT ACTATAACAT TGTCCACTGA TATAATGTCC
 9961 TATATACATA CTATGAATCC TGCTATTTTG GATGATTGGA ATTTTGGTGT TGCCCCTCCA
10021 CCATCTGCTA GTCTTGTAGA TACATACCGC TATCTGCAAT CAGCAGCAAT TACATGTCAA
10081 AAAGACGCCC CTGCACCTAC TAAAAAGGAT CCATATGATG GCTTAAACTT TTGGAATGTA
10141 AATTTAAAGG AAAAGTTTAG TTCTGAACTG GACCAGTTTC CTTTAGGACG CAAATTTCTT
10201 TTGCAGGCAG GTGTCCGCCG ACGACCCACT ATAGGCCCCC GTAAACGCCC TGCCACAGCA
10261 ACTACTGCAT CTACCTCTAA GCACAAACGT AAACGTGTGT CAAAGTAATT GTTGTATGTT
10321 TTGTTTTGTA TGTTGGTTGT ATGTGTGGTT GTATATGTGT CATGTTGTTG TTGGTATGTT
10381 GTGCATGTAT GTGTATATGT ATATGTGTAT GTTTGCAGGT ATGTTTGTAT AATCTGTTTT
10441 TGTTAATAAA GTATGTATGT CAGTTTACTT TGTGGTTGTA CCCTGTGACT AACATATGTC
10501 CTTGTTTTAC ATATCATAGG ACTGCAACAT TTCCTACATA ATTTGTAGCC CTACCCTAAG
10561 GTGTGTTACA GTACATGTAA TATATATATA GTTCTATATT ATACCAAGTG GCCATTTTGT
10621 AAGGCCATTT TGTGTGCAAC CGTTTTCGGT CGGTGGTGCT ATTTCCTTCT ATACAGTATT
10681 AAAAACTATG TGTTTCAGCA AAAACATGTT TCACCTTGGT TTACCCACAT AGTTGGCACC
10741 GGTAACAGTA TGTACTGGCG CACCTTACTT AGTCATCATC CTGTCCAGGT GCAGTGCAAC
10801 AATAGTTTGG CAGCCTATAT ATCTCCACCC TTGTAATAAA ACTGCTTTTA GGCATAGGTT
10861 TTTAACTGTT TTTACTTGCC TAATAGCATA GTTGGCCTGT ATAACTACTT TTGCATTCAA
10921 GAATGTGTCT TGTAGTGTAA GTTATACAGT GACTAATACC ACATCCATAA ATTTGTGCAA
10981 CCGAAATAGG TTGGGCACAC ATACCAATAC TTTTACTTAT AACATTTTAC AATCATTTTA
11041 TAGTATAAAG GGAGTGACCG AAAACGGTCA TGACCGAAAA CGGTGTATAT AAAGCTGAAC
11101 ACAGCAGTTG TCTATACCAA TGGCGCTATT TCACAACCCT GAGGAACGGC CATACAAATT
11161 GCCAGACCTG TGCAGGACAT TGGACACCAC ATTGCATGAC GTTACAATAG ACTGTGTCTA
```

```
11221 TTGCAGAAGG CAACTACAAC GGACAGAGGT ATATGAATTT GCCTTTGGTG ACTTAAATGT
11281 AGTATATAGG GACGGGGTAC CATTAGCTGC ATGCCAATCA TGTATTAAAT TTTATGCGAA
11341 AATACGGGAA CTACGATATT ACTCAGAATC GGTGTATGCA ACAACATTAG AAACCATAAC
11401 TAATACAAAG TTATATGATT TATCAATAAG GTGCATGTGT TGCCTGAAAC CATTGAGTCC
11461 TGCTGAAAAA CTAAGGCACC TAAATTCAAA ACGAAGATTT CATAAAATAG CAGGAAACTT
11521 TACAGGACAG TGTCGCCACT GCTGGACCAG TAAACGAGAG GACCGCAGAC GCACACGGCA
11581 GGAAACACAA GTATAAACTA ACTATGCATG GACCAAAGCC CACCGTGCAG GAAATTGCGT
11641 TAGAGTTATG TCCATGCAAT GAAATAGAGC CGGTCGACCT TGTATGTCAC GAGCAATTAG
11701 GAGATTCAGA CGATGAAATA GATGAACCCG ACCATGCAGT TAATCACCAC CAACATCAAC
11761 TACTAGCCAG ACGGGACGAA CAACAGCGTC ACACAATTCA GTGTACGTGT TGTAAGTGTA
11821 ACAACCTACT GCAACTAGTA GTAGAAGCGT CGCGGGAGAA CCTGCGGAAG CTAGAACTGC
11881 TGTTTATGGA CTCACTAAAT TTTGTGTGTC CGTGGTGTGC AACGGAAACC CAGTAATCTG
11941 CAATGGCCAA TTGTGAAGGT ACAGATGGGG ACGGGACGGG GTGTAACGGA TGGTTTTTTG
12001 TACAAGCAAT AGTAGATAAA CAAACAGGTG ACACAGTCTC AGAGGATGAG GATGAAAACG
12061 CGACAGATAC AGGTTCAGAC ATGGTAGATT TCATTGATGA TGCTACAGAT ATTTGTATAC
12121 AGGCAGAGCG TGAGACAGCA CAGGTACTGT TAAATATGCA ACAGGCCCAA AGGGATGCAC
12181 AAACAGTGCG TGCCCTAAAA CGAAAGTATA CAGACAGTAT AGAAAGCAGC CCCTTAGCAA
12241 AGTCGCCATT ACAGGAACTA TCAATAAATG TAAGCAGTAC ACAGGCAAGA CAACCGGCGT
12301 ATACAGTGCC GGACAGCGGC TATGGCAATA TGGAAGTGGA AACTAACTCG GAGGTAACTG
12361 TAGCAACTAA TACAAATGGG GCGGACGGGG AGGATGAAGG GGAAAATGGC GACAGCATAC
12421 GGGAGGACTG TAGTAGTGTA GACAGTGCTA TAGATAGTGA AAACCAGGAT CCTAAATCAC
12481 CTACTACGCA ACTAAAAGTA TTATTACAAT GTAATAATAA AAAAGCTGCA ATGTTAACAG
12541 AATTTAAAAA AGTATATGGA TTGTCCTTTA ATGACCTAGT ACGTACATTT AAAAGTGATA
12601 AGACCACATG TACGGACTGG GTAGCAGCAA TATTCGGAGT AAATCCAACC ATTGCCGAAG
12661 GGTCTAAAAC ACTAATTAAA CAATATGCAT TATATACCCA TATACAATGT TTAGATACAA
12721 AAAACGGAAT ATTAATATTA ATGTTAATAA GATACAAATG TGGGAAAAAT AGAATAACAG
12781 TAGGAAAAGG ATTAAGTACA TTGTTGCATG TTCCAGACAG CTGTATGCTT TTGCAGCCAC
12841 CAAAATTGCG TAGCCCTGTT GCAGCATTGT ATTGGTATAG AACAGGAATA TCTAATATTA
12901 GTGAGGTGTG TGGAGACACG CCAGAATGGA TAAAAAGATT AACTATAATA CAACATGGAA
12961 TAGATGATAG TGTATTTGAT CTATCAGACA TGGTACAATG GGCATTTGAT AATGAGTTAA
13021 CAGATGAAAG TGATATAGCA TTTTCATATG CTATGTTGGC AGATTGTAAT AGTAATGCTG
13081 CAGCGTTTTT AAAAAGCAAC TGTCAAGCAA AATATGTAAA AGATTGTGCA ACAATGTGTA
13141 GACATTACAA ACGGGCACAA AAACGACAAA TGTCAATGCC GCAATGGATT AAATTTAGAT
13201 GCAGTAAATG TGATGGAGGC GGTGATTGGC GCATGGACTG CAGAATTTAC TACTTGTGTT
13261 TATTACAGGC GATTTCCTGA GCCAGA
```

HPV68b(int)-ME180R

The sequence is described in Results section 2.1.2, including Figures 2.6-2.8, and Table 2.3. Black color represents cellular sequence of chromosome 18. Blue color represents the integrated HPV68b DNA. Underlined nucleotides indicate overlapping areas between viral and cellular DNA.

```
   1 GGAATTATGC CTATTTGACA GATGAGGAAA CTAAAGTGAA GGTGAGGCTA AGGTCACAAA
  61 GTGATGGGAT GAGAGTGGGG GCGCTCGCAT CTTCCCTGAG GCCATTGCTT TTGCTAATGC
 121 ACCATGCAGG GGACTTGCAA GAAAGACAGA AAAAGTGGGT TCCCTACGGG GCTGCCTGGC
 181 CTCAGTCACA AAGTGAGGGC TCGTTTCAGT TTGCTAGGGT GCAACTGCAG GGTCAAGAGC
 241 TGGCGGGACA CACCAATGGA GAGGAGTGCC ACCGCAGATG GCTGGCACCC TGGCCGGGTT
 301 GGCCATGGAA GTCTGCGGCA GGCAAATCAA ATGGTGCCCC TTCAAGCTGT TGTTCTTTAA
 361 TTATTTTTTC ATCCTTTATT TTCTAGGGGG CAGTGGGGGG AGACTTGTTT TCTTTTAAGA
 421 AAAGTTGCAC ACCATGTTTT TTAAAACATT GCTTTGGCTT TGTATGCCAT AGTTCATTAC
 481 TTGTGTCCCT TAATGTCCAC TCCTCTGCAC TATATGCAGT TTGAGCAATG CTCTCTAGTG
 541 CCATCTGCAG TTCAATGGCT TGATATGCTT TAGTTTTTGA AATGTTTACA GGAGGCACCA
 601 CCTGGTGGTC AATATTATGC ATACCACGTT CTCGTGCTGC ATAATATATT GCATTTTCCA
 661 GTCGTATACA GTTCCAATAG TTAATATGGT CCTTTATACA TTTACTGTCC TGTTCATAAT
 721 GTTCTAATAT TTTCTCCTGT AACACATTTA AACGTTGGGA AAGTGTTTTC ATCATTGTCT
 781 CCTTCATCCT CGTCCTGCTG CAAGTCTAAT TTGCACCAAG TCTTTTCAAA AAAACATTTC
 841 CAGTTTTTAT CATTGATTGT ATACACTGGG TTCCTGTTTT GGTCAAATGG AAATGCATTA
 901 GGAAATTTAA ACACAGTTAG TCTGCTATGT AAATACGGCC ACCTATTGTC TTCCACAGGG
 961 TTGGTATTTG ATGTTATTAG CATTGGTGGA CATTTTATTT GTATTAGGTG TCTGTGTTTT
1021 CTATCTAAAC TTATTGGGTT ACCATCTAAT GCATTTCTCA TGTAATTATC AAAATATGAC
1081 CAACATGTAC CTGTTGCGTC ATCTAACATG GCTATTTTTG CATCTGCAAG TGGCTCTAAC
1141 CAAAAGTGAC TAGCTGAATT TACATATGAA ATTATTGTGC CTTGCAAGAA ATGTATAAGA
1201 CTCATGCAAA AATATGACTT GCCTGTATTT GGTGGCCCAT GTATAACTAT ACAATTACGT
1261 TTTGGCGTGC CTTTTAAAAA ATCCTTTAAT GCACATAAAA ATGTTATAAA TTCTAGCCCC
1321 TGATATCTTA AAAACTGTAC AATTGGTCGC CAATCACCGC CTTCATCACA TTTACTGCAT
1381 CTAAATTTAA TCCATTGCGG CATTGACATT TGTCGTTTTT GTGCCCGTTT GTAATGTCTA
1441 CACATTGTTG CACAATCTTT TACATATTTT GCTTGACAGT TGCTTTTTAA AAACGCTGCA
1501 GCATTACTAT TACAATCTGC CAACATAGCA TATGAAGATG CTATATCACT TTCATCTGTT
1561 AACTCATTAT CAAATGCCCA TTGTACCATG TCTGGTAGAT CAAATACACT ATCATCTATT
1621 CCATGTTGTA TTATAGTTAA TATTTTTATC CATTCTGGCG TGTTTTCACA CACCTCACTA
1681 ATATTAGATA TTCCTGTTCT ATACCAATAC AATGCTGCAA CAGGGCTACG CAATTTTGGT
1741 GGCTGCAAAA GCATACAGCT GTCTGGAACA TGCAACAATG TACTTAATCC TTTTCCTACT
1801 GTTATTCTAT TTTTCCCACA TTTGTATCTT ATTAACATTA ATATTAATAT TCCGTTTTTT
1861 GTATCTAAAC ATTGTATATG GGTATATAAT GCATATTGTT TAATTAGTGT TTTAAACCCT
1921 TCGGCAATGG TTGGATTTAC TCCGAATATT GCTGCTACCC AGTCCGTACA TGTGGTCTTA
1981 TCACTTTTAA ATGTACGTAC TAGGTCATTA AAGGACAATC CATATACTTT TTTAAATTCT
2041 GTTAACATTG CAGCTTTTTT ATTATTACAT TGTAATAATA CTTTTAGTTG CGTAGTAGGT
2101 GATTTAGGAT CCTGGTTTTC ACTATCTATA GCACTGTCTA CACTACTACA GTCCTCCCGT
2161 ATGCTGTCGC CATTTTCCCC TTCATCCTCC CCGTCCGCCC CATTTGTATT AGTTGCTACA
2221 GTTACCTCCG AGTTAGTTTC CACTTCCATA TTGATAGTTC CTGTAATGGC GACTTTGCTA
2281 AAGGGCTGCT TTCTATACTG TCTGTATACT TTCGTTTTAG GGCACGCACT GTTTGTGCAT
2341 CCCTTTGGGC CTGTTGCATA TTTAACAGTA CCTGTGCTGT CTCACGCTCT GCCTGTATAC
2401 AAATATCTGT AGCATCATCA ATGAAATCTA CCATGTCTGA ACCTGTATCT GTCGCGTTTT
2461 CATCCTCATC CTCTGAGACT GTGTCACCTG TTTGTTTATC TACTATTGCT TGTACAAAAA
2521 ACCATCCGTT ACACCCCGTC CCGTCCCCAT CTGTACCTTC ACAATTGGCC ATTGCAGATT
2581 ACTGGGTTTC CGTTGCACAC CACGGACACA CAAAATTTAG TGAGTCCATA AACAGCAGTT
2641 CTAGCTTCCG CAGGTTCTCC CGCGACGCTT CTACTACTAG TTGCAGTAGG TTGTTACACT
2701 TACAACACGT ACACTGAATT GTGTGACGCT GTTGTTCGTC CCGTCTGGCT AGTAGTTGAT
2761 GTTGGTGGTG ATTAACTGCA TGGTCGGGTT CATCTATTTC ATCGTCTGAA TCTCCTAATT
2821 GCTCGTGACA TACAAGGTCG ACCGGCTCTA TTTCATTGCA TGGACATAAC TCTAACACAA
2881 TTTCCTGCAC GGTGGGCTTT GGTCCATGCA TAGTTAGTTT ATACTTGTGT TTCCTGCCGT
2941 GTGCGTCTGC GGTCCTCTCG TTTACTGGTC CAGCAGTGGC GACACTGTCC TGTAAAGTTT
3001 CCTGCTATTT TATGAAATCT TCGTTTTGAA TTTAGGTGCC TTAGTTTTTC AGCAGGACTC
3061 AATGGTTTCA GGCAACACAT GCACCTTATT GATAAATCAT ATAACTTTGT ATTAGTTATG
3121 GTTTCTAATG TTGTTGCATA CACCGATTCT GAGTAATATC GTAGTTCCCG TATTTTCGCA
3181 TAAAATTTAA TACATGATTG GCATGCAGCT AATGGTACCC CGTCCCTATA TACTACATTT
3241 AAGTCACCAA AGGCAAATTC ATATACCTCT GTCCGTTGTA GTTGCCTTCT GCAATAGACA
3301 CAGTCTATTG TAACGTCATG CAATGTGGTG TCCAATGTCC TGCACAGGTC TGGCAATTTG
3361 TATGGCCGTT CCTCAGGGTT GTGAAATAGC GCCATTGGTA TAGACAACTG CTGTGTTCAG
3421 CTTTATATAC ACCGTTTTCG GTCATGACCG TTTTCGGTCA CTCCCTTTAT ACTATAAAAT
3481 GATTGTAAAA TGTTATAAGT AAAAGTATTG GTATGTGTGC CCAACCTATT TCGGTTGCAC
3541 AAATTTATGG ATGTGGTATT AGTCACTGTA TAACTTACAC TACAAGACAC ATTCTTGAAT
3601 GCAAAAGTAG TTATACAGGC CAACTATGCT ATTAGGCAAG TAAAAACAGT TAAAAACCTA
3661 TGCCTAAAAG CAGTTTTATT ACAAGGGTGG AGATATATAG GCTGCCAAAC TATTGTTGCA
3721 CTGCACCTGG ACAGGATGAT GACTAAGTAA GGTGCGCCAG TACATACTGT TACCGGTGCC
3781 AACTATGTGG GTAAACCAAG GTGAAACATG TTTTTGCTGA AACACATAGT TTTTAATACT
3841 GTATAGAAGG AAATAGCACC ACCGACCGAA AACGGTTGCA CACAAAATGG CCTTACAAAA
3901 TGGCCACTTG GTATAATATA GAACTATATA TATATTACAT GTACTGTAAC ACACCTTAGG
3961 GTAGGGCTAC AAATTATGTA GGAAATGTTG CAGTCCTATG ATATGTAAAA CAAGGACATA
4021 TGTTAGTCAC AGGGTGCAAC CACAAAGTAA ACTGACATAC ATACTTTATT AACAAAAACA
4081 GATTATACAA ACATACCTGC AAACATACAC ATATACATAT ACACATACAT GCACAACATA
4141 CCAACAACAA CATGACACAT ATACAACCAC ACATACAACC AACATACAAA ACAAAACATA
4201 CATACAGCGG GACAGTGCAA CATACATATA CTGTATGTAT ATTGGATACA TACCACCATA
4261 CATGTACATA TGATTGTACT GGTATTTTTG GTGTGGTTTT GTGTGTGCAT GTATATATGT
4321 TGCACTGTCC CGCTTCTGCA GTCCATGCAT GTGTGTGTGT ATGTGTGGAT ACTTGTGTTT
4381 GTGTTTATAT TAGTACGTAC CACACCATTG GAGGTCTTTG CTGTATATAT ACTTTTTTTT
4441 TTACTGCCTA TGTAGGTATT ACACAGTTTT GCTCGTTATA GTATGCCTTA AGTTTTGTAT
4501 TGTGCATTTG TATTGGTGTA TATTTTTATA AATAAATATG GTATCACACC GTGCTGCCAG
4561 GCGCAAGCGT GCATCTGCAA CTGAATTATA TAAAACATGC AAACAATCAG GCACATGTCC
4621 TCCTGATGTT ATAAATAAGG TTGAAGGCAC CACACTTGCA GACAAACTAT TGCAATGGAC
4681 CAGTTTAGGT ATTTTTTTGG GTGGCCTAGG CATTGGTACT GGGTCAGGAA CCGGGGGTCG
4741 TACTGGGTAC ATTCCTTTAG GTGGTAAACC TAATACTGTT GTAGATGTTT CGCCTGCACG
4801 TCCACCTGTG GTTATTGAAC CTGTGGGTCC TACAGAACCC TCCATTGTGC AATTGGTGGA
4861 AGATTCCAGT GTTATTACAT CTGGCACACC GGTACCAACA TTTACAGGCA CTTCTGGGTT
4921 TGAAATTACA TCTTCTTCTA CCACTACACC TGCTGTGTTA GACATTACCC CTTCGTCTGG
4981 GTCTGTGCAA GTAAGCAGTA CTAGTTTTAC TAACCCTGCA TTTGCAGACC CCACTATTAT
5041 AGAAGTGCCT CAAACAGGTG AAGTCTCTGG TAATGTGTTT GTAAGTACCC CCACATCGGG
5101 AACACATGGA TATGAAGAAA TACCTATGCA GGTCCTGCTG TTTATGGACT CACTAAATTT
5161 TGTGTGTCCG TGGTGTGCAA CGGAAACCCA GTAATCTGCA ATGGCCAATT GTGAAGGTAC
5221 AGATGGGGAC GGGACGGGGT GTAACGGATG GTTTTTTGTA CAAGCAATAG TAGATAAACA
5281 AACAGGTGAC ACAGTCTCAG AGGATGAGGA TGAAAACGCG ACAGATACAG GTTCAGACAT
5341 GGTAGATTTC ATTGATGATG CTACAGATAT TTGTATACAG GCAGAGCGTG AGACAGCACA
5401 GGTACTGTTA AATATGCAAC AGGCCCAAAG GGATGCACAA ACAGTGCGTG CCCTAAAACG
5461 AAAGTATACA GACAGTATAG AAAGCAGCCC TTTAGCAAAG TCGCCATTAC AGGAACTATC
5521 AATAAATGTA AGCAGTACAC AGGCAAGACA ACCGGCGTAT ACAGTGCCGG ACAGCGGCTA
5581 TGGCAATATG GAAGTGGAAA CTAACTCGGA GGTAACTGTA GCAACTAATA CAAATGGGGC
5641 GGACGGGGAG GATGAAGGGG AAAATGGCGA CAGCATACGG GAGGACTGTA GTAGTGTAGA
5701 CAGTGCTATA GATAGTGAAA ACCAGGATCC TAAATCACCT ACTACGCAAC TAAAAGTATT
5761 ATTACAATGT AATAATAAAA AAGCTGCAAT GTTAACAGAA TTTAAAAAAG TATATGGATT
5821 GTCCTTTAAT GACCTAGTAC GTACATTTAA AAGTGATAAG ACCACATGTA CGGACTGGGT
5881 AGCAGCAATA TTCGGAGTAA ATCCAACCAT TGCCGAAGGG TTTAAAACAC TAATTAAACA
5941 ATATGCATTA TATACCCATA TACAATGTTT AGATACAAAA AACGGAATAT TAATATTAAT
6001 GTTAATAAGA TACAAATGTG GGAAAAATAG AATAACAGTA GGAAAAGGAT TAAGTACATT
6061 GTTGCATGTT CCAGACAGCT GTATGCTTTT GCAGCCACCA AAATTGCGTA GCCCTGTTGC
6121 AGCATTGTAT TGGTATAGAA CAGGAATATC TAATATTAGT GAGGTGTGTG GAGACACGCC
6181 AGAATGGATA AAAAGATTAA CTATAATACA ACATGGAATA GATGATAGTG TATTTGATCT
6241 ATCAGACATG GTACAATGGG CATTTGATAA TGAGTTAACA GATGAAAGTG ATATAGCATT
6301 TTCATATGCT ATGTTGGCAG ATTGTAATAG TAATGCTGCA GCGTTTTTAA AAAGCAACTG
6361 TCAAGCAAAA TATGTAAAAG ATTGTGCAAC AATGTGTAGA CATTACAAAC GGGCACAAAA
6421 ACGACAAATG TCAATGCCGC AATGGATTAA ATTTAGATGC AGTAAATGTG ATGAAGGCGG
6481 TGATTGGCGC ATGGACTGCA GAATTTACTA CTTGTGTTTA TTACAGGCGA TTTCCTGAGC
6541 CAGA
```

A2. Nucleotide sequences of HPV68b genomes in CIN2 sample

HPV68b-CIN2

The sequence is described in Results section 2.1.4, including Figure 2.14, and Table 2.5.

```
   1 ATGGCGCTAT TTCACAACCC TGAGGAACGG CCATACAAAT TGCCAGACCT GTGCAGGACA
  61 TTGGACACCA CATTGCATGA CGTTACAATA GACTGTGTCT ATTGCAGAAG GCAACTACAA
 121 CGGACAGAGG TATATGAATT TGCCTTTGGT GACTTAAATG TAGTATATAG GGACGGGGTA
 181 CCATTAGCTG CATGCCAATC ATGTATTAAA TTTTATGCGA AAATACGGGA ACTACGATAT
 241 TACTCAGAAT CGGTGTATGC AACAACATTA GAAACCATAA CTAATACAAA GTTATATGAT
 301 TTATCAATAA GGTGCATGTG TTGCCTGAAA CCATTGAGTC CTGCTGAAAA ACTAAGGCAC
 361 CTAAATTCAA AACGAAGATT TCATAAAATA GCAGGAAACT TTACAGGACA GTGTCGCCAC
 421 TGCTGGACCA GTAAACGAGA GGACCGCAGA CGCACACGGC AGGAAACACA AGTATAAACT
 481 AACTATGCAT GGACCAAAGC CCACCGTGCA GGAAATTGTG TTAGAGTTAT GTCCATGCAA
 541 TGAAATAGAG CCGGTTGACC TTGTATGTCA CGAGCAATTA GGAGATTCAG ACGATGAAAT
 601 AGATGAACCC GACCATGCAG TTAATCACCA CCAACATCAA CTACTAGCCA GACGGGACGA
 661 ACAACAGCGT CACACAATTC AGTGTACGTG TTGTAAGTGT AACAACCTAC TGCAACTAGT
 721 AGTAGAAGCG TCGCGGGAGA ACCTGCGGAA GCTAGAACTG CTGTTTATGG ACTCACTAAA
 781 TTTTGTGTGT CCGTGGTGTG CAACGGAAAC CCAGTAATCT GCAATGGCCA ATTGTGAAGG
 841 TACAGATGGG GACGGGACGG GGTGTAACGG ATGGTTTTTT GTACAAGCAA TAGTAGATAA
 901 ACAAACAGGT GACACAGTCT CAGAGGATGA GGATGAAAAC GCGACAGATA CAGGTTCAGA
 961 CATGGTAGAT TTCATTGATG ATGCTACAGA TATTTGTATA CAGGCAGAGC GTGAGACAGC
1021 ACAAGTACTG TTAAATATGC AACAGGCCCA AAGGGATGCA CAAACAGTGC GTGCCCTAAA
1081 ACGAAAGTAT ACAGACAGTA TAGAAAGCAG CCCTTTAGCA AAGTCGCCAT TACAGGAACT
1141 ATCAATAAAT GTAAGCAGTA CACAGGCAAG ACAACCGGCG TATACAGTGC CGGACAGCGG
1201 CTATGGCAAT ATGGAAGTGG AAACTAACTC GGAGGTAACT GTAGCAACTA ATACAAATGG
1261 GGCGGACGGG GAGGATGAAG GGGAAAATGG CGACAGCATA CGGGAGGACT GTAGTAGTGT
1321 AGACAGTGCT ATAGATAGTG AAAACCAGGA TCCTAAATCA CCTACTACGC AACTAAAAGT
1381 ATTATTACAA TGTAATAATA AAAAAGCTGC AATGTTAACA GAATTTAAAA AAGTATATGG
1441 ATTGTCCTTT AATGACCTAG TACGTACATT TAAAAGTGAT AAGACCACAT GTACGGACTG
1501 GGTAGCAGCA ATATTTGGAG TAAATCCAAC CATTGCCGAA GGGTTTAAAA CACTAATTAA
1561 ACAATATGCA TTATATACCC ATATACAATG TTTAGATACA AAAAACGGAG TATTAATATT
1621 AATGTTAATA AGATACAAAT GTGGGAAAAA TAGAATAACA GTAGGAAAAG GATTAAGTAC
1681 ATTGTTGCAT GTTCCAGACA GCCGTATGCT TTTGCAGCCA CCAAAATTGC GTAGCCCTGT
1741 TGCAGCATTG TATTGGTATA GAACAGGAAT ATCTAATATT AGTGAGGTGT GTGGAGACAC
1801 GCCAGAATGG ATAAAAAGAT TAACTATAAT ACAACATGGA ATAGATGATA GTGTATTTGA
1861 TCTATCAGAC ATGGTACAAT GGGCATTTGA TAATGAGTTA ACAGATGAAA GTGATATAGC
1921 ATTTTCATAT GCTATGTTGG CAGATTGTAA TAGTAATGCT GCAGCGTTTT TAAAAAGCAA
1981 CTGTCAAGCA AAATATGTAA AAGATTGTGC AACAATGTGT AGACATTACA AACGGGCACA
2041 AAAACGACAA ATGTCAATGC CGCAATGGAT TAAATTTAGA TGCAGTAAAT GTGATGAAGG
2101 CGGTGATTGG CGACCAATTG TACAGTTTTT AAGATATCAG GGGCTAGAAT TTATAACATT
2161 TTTATGTGCA TTAAAGGATT TTTTAAAAGG CACGCCAAAA CGTAATTGTA TAGTTATACA
2221 TGGGCCACCA AATACAGGCA AGTCATATTT TTGCATGAGT CTTATACATT TCTTGCAAGG
2281 CACAATAATT TCACATGTAA ATTCAGCTAG TCACTTTTGG TTAGAGCCAC TTGCAGATGC
2341 AAAAATAGCC ATGTTAGATG ACGCAACAGG TACATGTTGG TCATATTTTG ATAATTACAT
2401 GAGAAATGCA TTAGATGGTA ACCCAATAAG TTTAGATAGA AAACACAGAC ACCTAATACA
2461 AATAAAATGT CCACCAATGC TAATAACATC AAATACCAAC CCTGTGGAAG ACAATAGGTG
2521 GCCTTATTTA CATAGCGGAC TAACTGTGTT TAAATTTCCT AATGCATTTC CATTTGACCA
2581 AAACAGGAAC CCAGTGTATA CAATCAATGA TAAAAACTGG AAATGTTTTT TTGAAAAGAC
2641 TTGGTGCAAA TTAGACTTGC AGCAGGACGA GGATGAAGGA GACAATGATG AAAACACTTT
2701 CCCAACGTTT AAATGTGTTA CAGGAGAAAA TATTAGAACA TTATGAACAG GACAGTAAAT
2761 GTATAAAGGA CCATATTAAC TATTGGAACT GTATACGACT GGAAAATGCA ATATATTATG
2821 CAGCACGAGA ACGTGGTATG CATAATATTG ACCACCAGGT GGTGCCTCCT GTAAACATTT
2881 CAAAAACTAA AGCATATCAA GCCATTGAAC TGCAGATGGC ACTAGAGAGC ATTGCTCAAA
2941 CTGCATATAG TGCAGAGGAG TGGACATTAA GGGACACAAG TAATGAACTA TGGCATACAA
3001 AGCCAAAGCA ATGTTTTAAA AAACATGGTG TTACAGTGGA AGTATGGTAT GACGGTGACA
3061 AAAGTAACTC AATGCACTAT GTAGTATGGG GTACAATTTA TTTTAAAAAC AGTACAGACA
3121 CATGGTGTAA AACGGAAGGG TGTGTGGATT ATTGGGGTGT ATATTATATG TATGAAAAAC
3181 AAAAAACGTA TTATGAAAGG TTTATGAATG ATGCACAACT ATATGGGACT AGTGGAAAAT
3241 GGGACGTGCA TTATAATGGC AACATAATTC ATTGTCCTGA CTCTATGTGC AGTACCACTG
3301 ACGGAACAGT ATCCACTTCT GAATCTATTG CCGAACTACA GAACACCACC GCGACCCATA
3361 CCACCGCAGG TGCCCCACGC ACCAAAAAAA CACGCTCGAC GTCGTCTTGG AAGCACCCCA
3421 GACAGTACGG AATCACTGAG CCCTCTGAGC CCAACGACGT GTCCGTGGAC GGTGTCAACC
3481 TCCCACTCCT CAGTAGAAGT GCAGGCCACA ACAAAAGAAG GAACGTGTGT TGTGGTGACA
3541 CTACACCTAT AGTGCATTTA AAAGGTGACA AAAATGGTTT AAAATGCCTT AGGTATAGGT
3601 TGCAAAAATA TAACGCTTTG TATGAAAATA TATCATGTAC ATGGCATTGG ATACGGGGTA
3661 GGGGATCAAC CAATACAGGA ATATTGACTG TAACATATAG TACTGAAGCA CAACGCCAGA
3721 AATTTTTGGA AACTGTTAAA ATACCTTCCA GTGTAACTGT TTCATTAGGA TATATGACAT
3781 TATAAGTGTT GTATTGTACC ACACTGTATA CTGTATGTAT ATTGGATACA TACCACCATA
3841 CATGTACATA TGATTGTACT GGTATTTTTG GTGTGGTTTT GTGTGTGCAT GTATATATGT
3901 TGCACTGTCC CGCTTCTGCA GTCCATGCAT GTGTGTGTGT ATGTGTGGAT ACTTGTGTTT
3961 GTGTTTATAT TAGTACGTAC CACACCATTG GAGGTCTTTG CTGTATATAT ACTTTTTTTT
4021 TTACTGCCTA TGTGGGTATT ACACAGTTTT GCTCGTTATA GTATGCCTTA AGTTTTGTAT
4081 TGTGCATTTG TATTGGTGTA TATTTGTATA AATAAATATG GTATCACACC GTGCTGCCAG
4141 GCGCAAGCGT GCATCTGCAA CTGAATTATA TAAAACATGC AAACAATCAG GCACATGTCC
4201 TCCTGATGTT ATAAATAAGG TTGAAGGCAC CACACTTGCA GACAAAATAT TGCAATGGAC
4261 CAGTTTAGGT ATTTTTTTGG GTGGCCTAGG CATTGGTACT GGGTCAGGAA CCGGGGGTCG
4321 TACTGGGTAC ATTCCTTTAG GTGGTAAACC TAATACTGTT GTAGATGTTT CGCCTGCACG
4381 TCCACCTGTG GTTATTGAAC CTGTGGGTCC TACAGAACCC TCCATTGTGC AATTGGTGGA
4441 AGATTCCAGT GTTATTACAT CTGGCACACC GGTACCAACA TTTACAGGCA CTTCTGGGTT
4501 TGAAATTACA TCTTCTTCTA CCACTACACC TGCTGTGTTA GACATTACCC CTTCGTCTGG
4561 GTCTGTGCAA GTAAGCAGTA CTAGTTTTAC TAACCCTGCA TTTGCAGACC CCACTATTAT
4621 AGAAGTGCCT CAAACAGGTG AAGTCTCTGG TAATGTGTTT GTAAGTACCC CCACATCGGG
4681 AACACATGGA TATGAAGAAA TACCTATGCA GGTATTTGCA ACACATGGCA CTGGTACAGA
4741 ACCTATTAGT AGTACACCTA TACCTGGGGT TAGTCGTGTG GCAGGGCCAC GTTTATATAG
4801 TAGGGCACAT CAACAGGTTC GTGTTAGTAA TTTTGATTTT GTAACTCACC CTTCATCATT
4861 TGTAACATTT GATAATCCTG CTTTTGAGCC TGTTGATACT ACACTTACAT ATGAACCTGC
4921 TGACATAGCT CCTGATCCGG ATTTTCTGGA CATTGTTCGT TTACATAGGC CTGCCTTAAC
4981 TTCCCGAAGA GGCACAGTAC GTTTTAGCAG AGTAGGCAAA AAGGCAACTA TGTTTACACG
5041 CCGGGGTACA CAAATCGGGG CACAGGTGCA CTATTATCAT GATATTAGTG GCATTGCTCC
5101 TGCTGACAGC ATTGAACTAC AACCTTTGGT TGCCCCAGAG CAGTCTGACC CTATGGATAC
5161 TTTATATGAT ATATACGCAC CAGATACTGA CAATACTACA GTATTGGATA CTGCATTCCA
5221 TAATGCTACA TTTACCTCCC GTTCCCATAT ATCTGTTCCT TCATTAGCGT CTACAGCATC
5281 TACTACATAT GCTAACACTA CTATTCCTAT TGGTACTGCT TGGAACACGC CTGTAAATAC
5341 TGGTCCTGAT GTTGTGTTAC CAGCAACGTC TCCACAGTTG CCTTTAACAC CCTCTACACC
5401 AATTGATACA ACCTATGCCA TAACTATATA TGGCACCAAT TATTATTTAT TACCATTATT
5461 GTTCTTTTTA TTAAAAAAAC GTAAACGCCT TCCTTATTTT TTTGCAGATG GCATTGTGGC
5521 GCTCTAGCGA CAACATGGTG TATTTGCCTC CCCCCTCAGT GGCGAAGGTT GTCAATACAG
5581 ATGATTACGT AACACGCACT GGCATTTATT ACTATGCTGG TACATCTAGG TTATTAACTG
5641 TAGGCCATCC ATATTTTAAG GTCCCTATGT CTGGGGGCCG CAAGCAGGAC ATTCCTAAGG
5701 TGTCTGCATA TCAATACAGG GTGTTTAGGA TTTCCCTACC TGATCCTAAT AAATTTAGTC
5761 TTCCTGAGTC TACATTATAT AACCCTGATA CGCAGCGATT GGTATGGGCC TGTGTTGGTG
5821 TTGAAATAGG TAGGGGGCAG CCATTAGGTG TTGGCCTTAG TGGGCATCCA TTATATAATA
5881 GGCTAGATGA TACTGAGAAT TCCCCGTTTT CCTCCAACAA AAATCCTAAG GACAGTAGGG
5941 ACAATGTTTC AGTGGACTAT AAACAAACGC AACTATGTAT TATAGGCTGT GTTCCTGCCA
6001 TTGGGGAGCA CTGGGCCAAA GGTAAATCTT GTAAGCCTAG CAATGTGCAG CCCGGGGACT
6061 GTCCACCATT GGAATTAGTA AATACACCTA TTCAGGATGG CGATATGATT GATACAGGAT
6121 ATGGTGCTAT GGACTTTAGT ACATTACAAG AAACAAAAAG CGAGGTGCCT TTAGATATAT
6181 GTCAATCAGT CTGCAAATAT CCTGACTATT TACAAATGTC TGCAGATGTA TATGGAGACA
6241 GTATGTTCTT TTGTTTACGT AGGGAACAGT TATTTGCTAG GCATTTTTGG AATAGAGGGG
6301 GCATGGTAGG GGACACTATA CCTACTGAAT TGTATATTAA GGGCACTGAC ATACGTGACA
6361 GTCCTGGTAG TTATGTATAT GCCCCCTCGC CTAGTGGGTC TATGGTATCC TCAGACTCCC
6421 AGTTATTTAA CAAGCCCTAT TGGCTGCACA AGGCACAGGG ACACAACAAT GGTATTTGTT
6481 GGCATAATCA ATTATTTCTT ACTGTTGTGG ATACCACTCG CAGTACCAAT TTTACTTTGT
6541 CTACTACTAC TGAATCAGCT GTACCAAATA TTTATGATCC TAATAAATTT AAGGAATATA
6601 TTAGGCATGT TGAGGAATAT GATTTGCAAT TTATATTTCA GTTGTGTACT ATAACATTGT
6661 CCACTGATGT AATGTCCTAT ATACATACTA TGAATCCTGC TATTTTGGAT GATTGGAATT
6721 TTGGTGTTGC CCCTCCACCA TCTGCTAGTC TTGTAGATAC ATACCGCTAT CTGCAATCAG
6781 CAGCAATTAC ATGTCAAAAA GACGCCCCTG CACCTACTAA AAAGGATCCA TATGATGGCT
6841 TAAACTTTTG GAATGTAAAT TTAAAGGAAA AGTTTAGTTC TGAACTGGAC CAGTTTCCTT
6901 TAGGACGCAA ATTTCTTTTA CAGGCAGGCG TCCGCCGACG ACCCACTATA GGCCCCCGTA
6961 AACGCCCCGC CACAGCAACT ACTGCATCTA CCTCTAAGCA CAAACGTAAA CGTGTGTCAA
7021 AGTAATTGTT GTATGTTTTG TTTTGTATGT TGGTTGTATG TGTGGTTGTA TATGTGTCAT
7081 GTTGTTGTTG GTATGTTGTG CATGTATGTG TATATGTATA TGTGTATGTT TGCAGGTATG
7141 TTTGTATAAT CTGTTTTTGT TAATAAAGTA TGTATGTCAG TTTACTTTGT GGTTGCACCC
7201 TGTGACTAAC ATATGTCCTT GTTTTACATA TCATAGGACT GCAACATTTC CTACATAATT
7261 TGTAGCCCTA CCCTAAGGTG TGTTACAGTA CATGTAATAT ATATATAGTT CTATATTATA
7321 CCAAGGGGCC ATTTTGTAAG GCCATTTTGT GTGCAACCGT TTTCGGTCGG TGGTGCTATT
7381 TCCTTCTATA CAGTATTAAA AACTATGTGT TTCAGCAAAA ACATGTTTCA CCTTGGTTTA
7441 CCCACATAGT TGGCACCGGT AACAGTATGT ACTGGCGCGC CTTACCTAGT CATCATCCTG
7501 TCCAGGTGCA GTGCAACAAT AGTTTGGCAG CCTATATATC TCCACCCTTG TAATAAAACT
7561 GCTTTTAGGC ATAGGTTTTT AACTGTTTTT ACTTGCCTAA TAGCATAGTT GGCCTGTATA
7621 ACTACTTTTG CATTCAAGAA TGTGTCTTGT AGTGTAAGTT ATACAGTGAC TAATACCACA
7681 TCCATAAATT TGTGCAATCG AAATAGGTTG GGCACACATA CCAATACTTT TACTTATAAC
7741 ATTTTACAAT CATTTTATAG TATAAAGGGA GTGACCGGAA ACGGTCATGA CCGAAAACGG
7801 TGTATATAAA GCTGAACACA GCAGTTGTCT ATACCA
```

HPV68b-CIN2-Del

The sequence is described in Results section 2.1.4, including Figures 2.15-2.16 and Table 2.6.

```
   1 ATGGCGCTAT TTCACAACCC TGAGGAACGG CCATACAAAT TGCCAGACCT GTGCAGGACA
  61 TTGGACACCA CACTGCATGA CGTTACAATA GACTGTGTCT ATTGCAGAAG GCAACTACAA
 121 CGGACAGAGG TATATGAATT TGCCTTTGGT GACTTAAATG TAGTATATAG GGACGGGGTA
 181 CCATTAGCTG CATGCCAATC ATGTATTAAA TTTTATGCGA AAATACGGGA ACTACGATAT
 241 TACTCAGAAT CGGTGTATGC AACAACATTA GAAACCATAA CTAATACAAA GTTATATGAT
 301 TTATCAATAA GGTGCATGTG TTGCCTGAAA CCATTGAGTC CTGCTGAAAA ACTAAGGCAC
 361 CTAAATTCAA AACGAAGATT TCATAAAATA GCAGGAAACT TTACAGGACA GTGTCGCCAC
 421 TGCTGGACCA GTAAACGAGA GGACCGCAGA CGCACACGGC AGGAAACACA AGTATAAACT
 481 AACTATGCAT GGACCAAAGC CCACCGTGCA GGAAATTGTG TTAGAGTTAT GTCCATGCAA
 541 TGAAATAGAG CCGGTTGACC TTGTATGTCA CGAGCAATTA GGAGATTCAG ACGATGAAAT
 601 AGATGAACCC GACCATGCAG TTAATCACCA CCAACATCAA CTACTAGCCA GACGGGACGA
 661 ACAACAGCGT CACACAATTC AGTGTACGTG TTGTAAGTGT AACAACCTAC TGCAACTAGT
 721 AGTAGAAGCG TCGCGGGAGA ACCTGCGGAA GCTAGAACTG CTGTTTATGG ACTCACTAAA
 781 TTTTGTGTGT CCGTGGTGTG CAACGGAAAC CCAGTAATCT GCAATGGCCA ATTGTGAAGG
 841 TACAGATGGG GACGGGACGG GGTGTAACGG ATGGTTTTTT GTACAAGCAA TAGTAGATAA
 901 ACAAACAGGT GACACAGTCT CAGAGGATGA GGATGAAAAC GCGACAGATA CAGGTTCAGA
 961 CATGGTAGAT TTCATTGATG ATGCTACAGA TATTTGTATA CAGGCAGAGC GTGAGACAGC
1021 ACAGGTACTG TTAAATATGC AACAGGCCCA AAGGGATGCA CAAACAGTGC GTGCCCTAAA
1081 ACGAAAGTAT ACAGACAGTA TAGAAAGCAG CCCTTTAGCA AAGTCGCCAT TACAGGAACT
1141 ATCAATAAAT GTAAGCAGTA CACAGGCAAG ACAACCGGCG TATACAGTGC CGGACAGCGG
1201 CTATGGCAAT ATGGAAGTGG AAACTAACTC GGAGGTAACT GTAGCAACTA ATACAAATGG
1261 GGCGGACGGG GAGGATGAAG GGGAAAATGG CGACAGCATA CGGGAGGACT GTAGTAGTGT
1321 AGACAGTGCT ATAGATAGTG AAAACCAGGA TCCTAAATCA CCTACTACGC AACTAAAATG
1381 ATAAAAACTG GAAATGTTTT TTTGAAAAGA CTTGGTGCAA ATTAGACTTG CAGCAGGACG
1441 AGGATGAAGG AGACAATGAT GAAAACACTT TCCCAACGTT TAAATGTGTT ACAGGAGAAA
1501 ATATTAGAAC ATTATGAACA GGACAGTAAA TGTATAAAGG ACCATATTAA CTATTGGAAC
1561 TGTATACGAC TGGAAAATGC AATATATTAT GCAGCACGAG AACGTGGTAT GCATAATATT
1621 GACCACCAGG TGGTGCCTCC TGTAAACATT TCAAAAACTA AAGCATATCA AGCCATTGAA
1681 CTGCAGATGG CACTAGAGAG CATTGCTCAA ACTGCATATA GTGCAGAGGA GTGGACATTA
1741 AGGGACACAA GTAATGAACT ATGGCATACA AAGCCAAAGC AATGTTTTAA AAAACATGGT
1801 GTTACAGTGG AAGTATGGTA TGACGGTGAC AAAAGCAACT CAATGCACTA TGTAGTATGG
1861 GGTACAATTT ATTTTAAAAA CAGTACAGAC ACATGGTGTA AAACGGAAGG GTGTGTGGAT
1921 TATTGGGGTG TATATTATAT GTATGAAAAA CAAAAAACGT ATTATGAAAG GTTTATGAAT
1981 GATGCACAAC TATATGGGAC TAGTGGAAAA TGGGACGTGC ATTATAATGG CAACATAATT
2041 CATTGTCCTG ACTCTATGTG CAGTACCACT GACGGAACAG TATCCACTTC TGAATCTATT
2101 GCCGAACTAC AGAACACCAC CGCGACCCAT ACCACCGCAG GTGCCCCATG CACCAAAAAA
2161 ACACGCTCGA CGTCGTCTTG GAAGCACCCC AGACAGTACG GAATCACTGA GCCCTCTGAG
2221 CCCAACGACG TGTCCGTGGA CGGTGTCAAC CTCCCACTCC TCAGTAGAAG TGCAGGCCAC
2281 AACAAAAGAA GGAACGTGTG TTGTGGTGAC ACTACACCTA TAGTGCATTT AAAAGGTGAC
2341 AAAAATGGTT TAAAATGCCT TAGGTATAGG TTGCAAAAAT ATAACGCTTT GTATGAAAAT
2401 ATATCATGTA CATGGCATTG GATACGGGGT AGGGGATCAA CCAATACAGG AATATTGACT
2461 GTAACATATA GTACTGAAGC ACAACGCCAG AAATTTTTGG AAACTGTTAA AATACCTTCC
2521 AGTGTAACTG TTTCATTAGG ATATATGACA TTATAAGTGT TGTATTGTAC CACACTGTAT
2581 ACTGTATGTA TATTGGATAC ATACCACCAT ACATGTACAT ATGATTGTAC TGGTATTTTT
2641 GGTGTGGTTT TGTGTGTGCA TGTATATATG TTGCACTGTC CCGCTTCTGC AGTCCATGCA
2701 TGTGTGTGTG TATGTGTGGA TACTTGTGTT TGTGTTTATA TTAGTACGTA CCACACCATT
2761 GGAGGTCTTT GCTGTATATA TACTTTTTTT TTTACTGCCT ATGTGGGTAT TACACAGTTT
2821 TGCTCGTTAT AGTATGCCTT AAGTTTTGTA TTGTGCATTT GTATTGGTGT ATATTTGTAT
2881 AAATAAATAT GGTATCACAC CGTGCTGCCA GGCGCAAGCG TGCATCTGCA ACTGAATTAT
2941 ATAAAACATG CAAACAATCA GGCACATGTC CTCCTGATGT TATAAATAAG GTTGAAGGCA
3001 CCACACTTGC AGACAAAATA TTGCAATGGA CCAGTTTAGG TATTTTTTTG GGTGGCCTAG
3061 GCATTGGTAC TGGGTCAGGA ACCGGGGGTC GTACTGGGTA CATTCCTTTA GGTGGTAAAC
3121 CTAATACTGT TGTAGATGTT TCGCCTGCAC GTCCACCTGT GGTTATTGAA CCTGTGGGTC
3181 CTACAGAACC CTCCATTGTG CAATTGGTGG AAGATTCCAG TGTTATTACA TCTGGCACAC
3241 CGGTACCAAC ATTTACAGGC ACTTCTGGGT TTGAAATTAC ATCTTCTTCT ACCACTACAC
3301 CTGCTGTGTT AGACATTACC CCTTCGTCTG GGTCTGTGCA AGTAAGCAGT ACTAGTTTTA
3361 CTAACCCTGC ATTTGCAGAC CCCACTATTA TAGAAGTGCC TCAAACAGGT GAAGTCTCTG
3421 GTAATGTGTT TGTAAGTACC TCCACATCGG GAACACATGG ATATGAAGAA ATACCTATGC
3481 AGGTATTTGC AACACATGGC ACTGGTACAG AACCTATTAG TAGTACACCT ATACCTGGGG
3541 TTAGTCGTGT GGCAGGGCCA CGTTTATATA GTAGGGCACA TCAACAGGTT CGTGTTAGTA
3601 ATTTTGATTT TGTAACTCAC CCTTCATCAT TTGTAACATT TGATAATCCT GCTTTTGAGC
3661 CTGTTGATAC TACACTTACA TATGAACCTG CTGACATAGC TCCTGATCCG GATTTTCTGG
3721 ACATTGTTCG TTTACATAGG CCTGCCTTAA CTTCCCGAAG AGGCACAGTA CGTTTTAGCA
3781 GAGTAGGCAA AAAAGGCAAC TATGTTTACA CGCCGGGGTA CACAAATCGG GGCACAGGTG
3841 CACTATTATC ATGATATTAG TGGCATTGCT CCTGCTGACA GCATTGAACT ACAACCTTTG
3901 GTTGCCCCAG AGCAGTCTGA CCCTATGGAT ACTTTATATG ATATATATGC ACCAGATACT
3961 GACAATACTA CAGTATTGGA TACTGCATTC CATAATGCTA CATTTACCTC CCGTTCCCAT
4021 ATATCTGTTC CTTCATTAGC GTCTACAGCA TCTACTACAT ATGCTAACAC TACTATTCCT
4081 ATTGGTACTG CTTGGAACAC GCCTGTAAAT ACTGGTCCTG ATGTTGTGTT ACCAGCAACG
4141 TCTCCACAGT TGCCTTTAAC ACCCTCTACA CCAATTGATA CAACCTATGC CATAACTATA
4201 TATGGCACCA ATTATTATTT ATTACCATTA TTGTTCTTTT TATTAAAAAA ACGTAAACGC
4261 CTTCCTTATT TTTTTGCAGA TGGCATTGTG GCGCTCTAGC GACAACATGG TGTATTTGCC
4321 TCCCCCCTCA GTGGCGAAGG TTGTCAATAC AGATGATTAC GTAACACGCA CTGGCATTTA
4381 TTACTATGCT GGTACATCTA GGTTATTAAC TGTAGGCCAT CCATATTTTA AGGTCCCTAT
4441 GTCTGGGGGC CGCAAGCAGG ACATTCCTAA GGTGTCTGCA TATCAATACA GGGTGTTTAG
4501 GATTTCCCTA CCTGATCCTA ATAAATTTAG TCTTCCTGAG TCTACATTAT ATAACCCTGA
4561 TACGCAGCGA TTGGTATGGG CCTGTGTTGG TGTTGAAATA GGTAGGGGGG CAGCCATTAG
4621 GTGTTGGCCT TAGTGGGCAT CCATTATATA ATAGGCTAGA TGATACTGAG AATTCCCCGT
4681 TTTCCTCCAA CAAAAATCCT AAGGACAGTA GGGACAATGT TTCAGTGGAC TATAAACAAA
4741 CGCAACTATG TATTATAGGC TGTGTTCCTG CCATTGGGGA GCACTGGGCC AAAGGTAAAT
4801 CTTGTAAGCC TAGCAATGTG CAGCCCGGGG ACTGTCCACC ATTGGAATTA GTAAATACAC
4861 CTATTCAGGA TGGCGATATG ATTGATACAG GATATGGTGC TATGGACTTT AGTACATTAC
4921 AAGAAACAAA AAGCGAGGTG CCTTTAGATA TATGTCAATC AGTCTGCAAA TATCCTGACT
4981 ATTTACAAAT GTCTGCAGAT GTATATGGAG ACAGTATGTT CTTTTGTTTA CGTAGGGAAC
5041 AGTTATTTGC TAGGCATTTT TGGAATAGAG GGGGCATGGT AGGGGACACT ATACCTACTG
5101 AATTGTATAT TAAGGGCACT GACATACGTG ACAGTCCTAG TAGTTATGTA TATGCCCCCT
5161 CGCCTAGTGG GTCTATGGTA TCCTCAGACT CCCAGTTATT TAACAAGCCC TATTGGCTGC
5221 ACAAGGCACA GGGACACAAC AATGGTATTT GTTGGCATAA TCAATTATTT CTTACTGTTG
5281 TGGATACCAC TCGCAGTACC AATTTTACTT TGTCTACTAC TACTGAATCA GCTGTACCAA
5341 ATATTTATGA TCCTAATAAA TTTAAGGAAT ATATTAGGCA TGTTGAGGAA TATGATTTGC
5401 AATTTATATT TCAGTTGTGT ACTATAACAT TGTCCACTGA TGTAATGTCC TATATACATA
5461 CTATGAATCC TGCTATTTTG GATGATTGGA ATTTTGGTGT TGCCCCTCCA CCATCTGCTA
5521 GTCTTGTAGA TACATACCGC TATCTGCAAT CAGCAGCAAT TACATGTCAA AAAGACGCCC
5581 CTGCACCTAC TAAAAAGGAT CCATATGATG GCTTAAACTT TTGGAATGTA AATTTAAAGG
5641 AAAAGTTTAG TTCTGAACTG GACCAGTTTC CTTTAGGACG CAAATTTCTT TTACAGGCAG
5701 GCGTCCGCCG ACGACCCACT ATAGGCCCCC GTAAACGCCC CGCCACAGCA ACTACTGCAT
5761 CTACCTCTAA GCACAAACGT AAACGTGTGT CAAAGTAATT GTTGTATGTT TTGTTTTGTA
5821 TGTTGGTTGT ATGTGTGGTT GTATATGTGT CATGTTGTTG TTGGTATGTT GTGCATGTAT
5881 GTGTATATGT ATATGTGTAT GTTTGCAGGT ATGTTTGTAT AATCTGTTTT TGTTAATAAA
5941 GTATGTATGT CAGTTTACTT TGTGGTTGCA CCCTGTGACT AACATATGTC CTTGTTTTAC
6001 ATATCATAGG ACTGCAACAT TTCCTACATA ATTTGTAGCC CTACCCTAAG GTGTGTTACA
6061 GTACATGTAA TATATATATA TAGTTCTATA TTATACCAAG GGGCCATTTT GTAAGGCCAT
6121 TTTGTGTGCA ACCGTTTTCG GTCGGTGGTG CTATTTCCTT CTATACAGTA TTAAAAACTA
6181 TGTGTTTCAG CAAAAACATG TTTCACCTTG GTTTACCCAC ATAGTTGGCA CCGGTAACAG
6241 TATGTACTGG CGCGCCTTAC TTAGTCATCA TCCTGTCCAG GTGCAGTGCA ACAATAGTTT
6301 GGCAGCCTAT ATATCTCCAC CCTTGTAATA AAACTGCTTT TAGGCATAGG TTTTTAACTG
6361 TTTTTACTTG CCTAATAGCA TAGTTGGCCT GTATAACTAC TTTTGCATTC AAGAATGTGT
6421 CTCGTAGTGT AAGTTATACA GTGACTAATA CCACATCCAT AAATTTGTGC AACCGAAATA
6481 GGTTGGGCAC ACATACCAAT ACTTTTACTT ATAACATTTT ACAATCATTT TATAGTATAA
6541 AGGGAGTGAC CGAAAACGGT CATGACCGAA AACGGTGTAT ATAAAGCTGA ACACAGCAGT
6601 TGTCTATACC A
```

A3. Partial URR sequences of eleven HPV68 positive samples

The nucleotide sequences of eleven clinical samples (Results, Table 2.9) covering partial HPV68 URR region, corresponding to pos. 7279-7769 of the HPV68b-CIN2, are shown as an alignment. The sequences are described in Results section 2.1.5, including Figure 2.20 and Table 2.10.

```
                .         .         .         .         .         .  60
Reims-06        TGTGTTACAGTACATGTAATATATATATAGTTTTATATTATACCAAGTGGCCATTTTGTA
Reims-09        TGTGTTACAGTACATGTAATATATATATAGTTTTATATTATACCAAGTGGCCATTTTGTA
Reims-12        TGTGTTACATTATATGCAATATATTTATGGTTGCAAACTATACCGTGTGGCCATTTTGTA
Reims-14        TGTGTTACAGTACATGTAATATATATATAGTTTTATATTATACCAAGTGGCCATTTTGTA
Reims-15        TGTGTTACAGTACATGTAATATATATATAGTTCTATATTATACCAAGGGGCCATTTTGTA
Reims-16        TGTGTTACTGTACATGTAATATATATATAGTTCTATATTATACCAAGGGGCCATTTTGTA
Reims-18        TGTGTTACAGTACATGTAATATATATATAGTTTTATATTATACCAAGTGGCCATTTTGTA
Reims-28        TGTGTTACAGTACATGTAATATATATATAGTTTTATATTATACCAAGTGGCCATTTTGTA
Reims-31        TGTGTTACAGTACATGTAATATATATATAGTTCTATATTATACCAAGTGGCCATTTTGTA
Reims-33        TGTGTTACAGTACATGTAATATATATATAGTTCTATATTATACCAAGGGGCCATTTTGTA
HPV68b-CIN2     TGTGTTACAGTACATGTAATATATATATAGTTCTATATTATACCAAGGGGCCATTTTGTA

                .         .         .         .         .         . 120
Reims-06        AGGCCATTTTGTGTGCAACCGTCTTCGGTCGGTGGTGCTATTTCCTCCTATACAGTATTA
Reims-09        AGGCCATTTTGTGTGCAACCGTCTTCGGTCGGTGGTGCTATTTCCTCCTATACAGTATTA
Reims-12        TGGCCGGGTTGTGTGCGACCGCTTTCGGTCGTTGGTACTATTTGCTTTTAGATAGTATTA
Reims-14        AGGCCATTTTGTGTGCAACCGTCTTCGGTTGGTGGTGCTATTTCCTCCTATACAGTATTA
Reims-15        AGGCCATTTTGTGTGCAACCGTTTTCGGTCGGTGGTGCTATTTCCTTCTATACAGTATTA
Reims-16        AGGCCATTTTGTGTGCAACCGTTTTCGGTCGGTGGTGCTATTTCCTTCTATACAGTATTA
Reims-18        AGGCCATTTTGTGTGCAACCGTCTTCGGTCGGTGGTGCTATTTCCTCCTATACAGTATTA
Reims-28        AGGCCATTTTGTGTGCAACCGTCTTCGGTCGGTGGTGCTATTTCCTCCTATACAGTATTA
Reims-31        AGGCCATTTTGTGTGCAACCGTCTTCGGTCGGTGGTGCTATTTCCTCCTATACAGTATTA
Reims-33        AGGCCATTTTGTGTGCAACCGTTTTCGGTCGGTGGTGCTATTTCCTTCTATACAGTATTA
HPV68b-CIN2     AGGCCATTTTGTGTGCAACCGTTTTCGGTCGGTGGTGCTATTTCCTTCTATACAGTATTA

                .         .         .         .         .         . 180
Reims-06        AAAACTATGTGTTTCAGCAAAAACATGTTTCACCTTGGTTTACCCACATAGTTGGCACCG
Reims-09        AAAACTATGTGTTTCAGCAAAAACATGTTTCACCTTGGTTTACCCACATAGTTGGCACCG
Reims-12        AAACCTATGTGTTTCAGCAAAAACATGTTTTACCTTGAGTTACCCACATAGTTGGCACCA
Reims-14        AAAACTATGTGTTTCAGCAAAAACATGTTTCACCTTGGTTTACCCACATAGTTGGCACCG
Reims-15        AAAACTATGTGTTTCAGCAAAAACATGTTTCACCTTGGTTTACCCACATAGTTGGCACCG
Reims-16        AAAACTATGTGTTTCAGCAAAAACATGTTTCACCTTGGTTTACCCACATAGTTGGCACCG
Reims-18        AAAACTATGTGTTTCAGCAAAAACATGTTTCACCTTGGTTTACCCACATAGTTGGCACCG
Reims-28        AAAACTATGTGTTTCAGCAAAAACATGTTTCACCTTGGTTTACCCACATAGTTGGCACCG
Reims-31        AAAACTATGTGTTTCAGCAAAAACATGTTTCACCTTGGTTTACCCACATAGTTGGCACCG
Reims-33        AAAACTATGTGTTTCAGCAAAAACATGTTTCACCTTGGTTTACCCACATAGTTGGCACCG
HPV68b-CIN2     AAAACTATGTGTTTCAGCAAAAACATGTTTCACCTTGGTTTACCCACATAGTTGGCACCG

                .         .         .         .         .         . 240
Reims-06        GTAACAGTATGTACTGGCGCACCTTACTTAGTCATCATCCTGTCCAGATGCAGTGCAACA
Reims-09        GTAACAGTATGTACTGGCGCACCTTACTTAGTCATCATCCTGTCCAGGTGCAGTGCAACA
Reims-12        GTAACAGT-CGTACTGGCGCACCTTAGTTAGTCATCATCCTGTCCAGGTGCACTGCAACA
Reims-14        GTAACAGTATGTACTGGCGCACCTTACTTAGTCATCATCCTGTCCAGGTGCAGTGCAACA
Reims-15        GTAACAGTATGTACTGGCGCACCTTACTTAGTCATCATCCTGTCCAGGTGCAGTGCAACA
Reims-16        GTAACAGTATGTACTGGCGCACCTTACTTAGTCATCATCCTGTCCAGGTGCAGTGCAACA
Reims-18        GTAACAGTATGTACTGGCGCACCTCACTTAGTCATCATCCTGTCCAGGTGCAGTGCAACA
Reims-28        GTAACAGTATGTACTGGCGCACCTTACTTAGTCATCATCCTGTCCAGGTGCAGTGCAACA
Reims-31        GTAACAGTATGTACTGGCGCACCTTACTTAGTCATCATCCTGTCCAGGTGCAGTGCAACA
Reims-33        GTAACAGTATGTACTGGCGCACCTTACTTAGTCATCATCCTGTCCAGGTGCAGTGCAACA
HPV68b-CIN2     GTAACAGTATGTACTGGCGCGCCTTACCTAGTCATCATCCTGTCCAGGTGCAGTGCAACA

                .         .         .         .         .         . 300
Reims-06        ATAGTTTGGCAGCCTATATATCTCCACCCTTGTAATAAAACTGCTTTTAGGCATAGTTTT
Reims-09        ATAGTTTGGCAGCCTATATATCTCCACCCTTGTAATAAAACTGCTTTTAGGCATAGTTTT
Reims-12        ATAGTTTGGCAACCAACGTATCTCCTCCCTTGTAATAAAACTGCTTTTAGGCATAAAGTT
Reims-14        ATAGTTTGGCAGCCTATATATCTCCACCCTTGTAATAAAACTGCTTTTAGGCATAGTTTT
Reims-15        ATAGTTTGGCAGCCTATATATCTCCACCCTTGTAATAAAACTGCTTTTAGGCATAGGTGT
Reims-16        ATAGTTTGGCAGCCTATATATCTCCACCCTTGTAATAAAACTGCTTTTAGGCATAGGTTT
Reims-18        ATAGTTTGGCAGCCTATATATCTCCACCCTTGTAATAAAACTGCTTTTAGGCATAGTTTT
Reims-28        ATAGTTTGGCAGCCTATATATCTCCACCCTTGTAATAAAACTGCTTTTAGGCATAGTTTT
Reims-31        ATAGTTTGGCAGCCTATATATCTCCACCCTTGTAATAAAACTGCTTTTAGGCATAGTTTT
Reims-33        ATAGTTTGGCAGCCTATATATCTCCACCCTTGTAATAAAACTGCTTTTAGGCATAGGTTT
HPV68b-CIN2     ATAGTTTGGCAGCCTATATATCTCCACCCTTGTAATAAAACTGCTTTTAGGCATAGGTTT

                .         .         .         .         .         . 360
Reims-06        TTAACTGTTTTTACTTGCCTAATAGCATAGTTGGCCTGTATAACTACTTTTGCATTCAAG
Reims-09        TTAACTGTTTTTACTTGCCTAATAGCATAGTTGGCCTGTATAACTACTTTTGCATTCAAG
Reims-12        TTAACTGTTTTTGCTTGCCTAATAGCATAGTTGGCCAGTATAACTACTTTTGCATTCAAG
Reims-14        TTAACTGTTTTTACTTGCCTAATAGCATAGTTGGCCTGTATAACTACTTTTGCATTCAAG
Reims-15        TTAACTGTTTTTACTTGCCTAATAGCATAGTTGGCCTGTATAACTACTTTTGCATTCAAG
Reims-16        TTAACTGTTTTTACTTGCCTAATAGCATAGTTGGCCTGTATAACTACTTTTGCATTCAAG
Reims-18        TTAACTGTTTTTACTTGCCTAATAGCATAGTTGGCCTGTATAACTACTTTTGCATTCAAG
Reims-28        TTAACTGTTTTTACTTGCCTAATAGCATAGTTGGCCTGTATAACTACTTTTGCATTCAAG
Reims-31        TTAACTGTTTTTACTTGCCTAATAGCATAGTTGGCCTGTATAACTACTTTTGCATTCAAG
Reims-33        TTAACTGTTTTTACTTGCCTAATAGCATAGTTGGCCTGTATAACTACTTTTGCATTCAAG
HPV68b-CIN2     TTAACTGTTTTTACTTGCCTAATAGCATAGTTGGCCTGTATAACTACTTTTGCATTCAAG

                .         .         .         .         .         . 420
Reims-06        AATCTGTCTAGTAGTGTAAGTTATACAGTGACTAATACCACATCCATAAATTTGTGCAAC
Reims-09        AATCTGTCTAGTAGTGTAAGTTATACAGTGACTAATACCACATCCATAAATTTGTGCAAC
Reims-12        AATCTGTCTGGTAGTGTAAGTTATACAGTGACTAATACTACATCCATAAATTTGTGCAAC
Reims-14        AATCTGTCTAGTAGTGTAAGTTATACAGTGACTAATACCACATCCATAAATTTGTGCAAC
Reims-15        AATGTGTCTTGTAGTGTAAGTTATACAGTGACTAATACCACATCCATAAATTTGTGCAAC
Reims-16        AATGTGTCTTGTAGTGTAAGTTATACAGTGACTAATACCACATCCATAAATTTGTGCAAC
Reims-18        AATCTGTCTGGTAGTGTAAGTTATACAGTGACTAATACCACATCCATAAATTTGTGCAAC
Reims-28        AATCTGTCTAGTAGTGTAAGTTATACAATGACTAATACCACATCCATAAATTTGTGCAAC
Reims-31        AATCTGTCTTGTAGTGTAAGTTATACAGTGACTAATACCACATCCATAAATTTGTGCAAC
Reims-33        AATGTGTCTTGTAGTGTAAGTTATACAGTGACTAATACCACATCCATAAATTTGTGCAAC
HPV68b-CIN2     AATGTGTCTTGTAGTGTAAGTTATACAGTGACTAATACCACATCCATAAATTTGTGCAAT

                .         .         .         .         .         . 480
Reims-06        CGAAATAGGTTGGGCACACATACCAATACTTTTACTTATAACATTTTACAATCATTTTAT
Reims-09        CGAAATAGGTTGGGCACACATACCAATACTTTTACTTATAACATTTTACAATCATTTTAT
Reims-12        CGAAAAAGGTTGGGCACACATACCAATACTTTTACTTATAACATTTTACAATTATTCTAT
Reims-14        CGAAATAGGTTGGGCACACATACCAATACTTTTACTTATAACATTTTACAATCATTTTAT
Reims-15        CGAAATAGGTTGGGCACACATACCAATACTTTTACTTATAACATTTTACAATCATTTTAT
Reims-16        CGAAATAGGTTGGGCACACATACCAATACTTTTACTTATAACATTTTACAATCATTTTAC
Reims-18        CGAAATAGGTTGGGCACACATACCAATACTTTTACTTATAACATTTTACAATCATTTTGT
Reims-28        CGAAATAGGTTGGGCACACATACCAATACTTTTACTTATAACATTTTACAATCATCTTAT
Reims-31        CGAAAAAGGTTGGGCACACATACCAATACTTTTACTTATAACATTTTACAATCATTTTAT
Reims-33        CGAAATAGGTTGGGCACACATACCAATACTTTTACTTATAACATTTTACAATCATTTTAT
HPV68b-CIN2     CGAAATAGGTTGGGCACACATACCAATACTTTTACTTATAACATTTTACAATCATTTTAT

                .  491
Reims-06        ATAATAAAGGG
Reims-09        ATAATAAAGGG
Reims-12        ATAAAAAAAGG
Reims-14        AGTATAAAGGG
Reims-15        AGTATAAAGGG
Reims-16        AGTATAAAGGG
Reims-18        ATTAAAAAGGG
Reims-28        ATTAAAAAGGG
Reims-31        ATTAGAAAGGG
Reims-33        AGTATAAAGGG
HPV68b-CIN2     AGTATAAAGGG
```

A4. Nucleotide sequences of identified HPV16 integration junctions in ASP16 experiments

Sample HSIL-61979

See Results section 2.2.2.6, pages 78-80. The nucleotide sequence shown here is composed of the junction-specific PCR product sequence and the ASP16 E26 sequence reads (Results, Figure 2.44). Red color indicates HPV16 DNA between pos. 2403-2516, and black color the flanking cellular DNA.

```
  1 AGCAGATGCC AAAATAGGTA TGTTGGATGA TGCTACAGTG CCCTGTTGGA ACTATATAGA
 61 TGACAATTTA AGAAATGCAT TGGATGGAAA TTTAGTTTCT ATGGATGTAA AGCAGATGTG
121 GGGCCAGCAT GCTTACAGCG GGGACACCAC TTCCCAACAG CCTGAAAAAC TCCCATTTCA
181 CCTTAAAAAC
```

Sample HSIL-75857

See Results section 2.2.2.6, pages 81-82. The nucleotide sequence shown here is assembled from the sequences of two junction-specific PCR products (Results, Figure 2.46). Red color indicates HPV16 DNA between pos. 1064-1149, and black color the flanking cellular DNA.

```
  1 CACATGCGTT GTTTACTGCA CAGGAAGCAA AACAACATAG AGATGCAGTA CAGGTTCTAA
 61 AACGAAAGTA TTTGGGTAGT CCACTTGTGT TTGTAAGTTC TCATCTTAAT TTGCAGTTAT
121 CCAGTTTTGG AATGCCACGA AAAAATAATT ATGTTTTTAA AAATTATTTT TTAAAAAAAC
181 CTTTCCAAAT AATTGAAAAA CACTATTCTA CTCAAATCTC AAAAATATCC AAATATACGT
241 TTTAAACAGA AAAATTAAAA AAATATTTTC TCTGGATTCT CTATTTTCTG AACCTTTAAA
301 AAAATGTTTT CAGTGAATAA TATGTTCCGA GGTGGATTCG ACTACAAT
```

Sample CIN2/3-1801

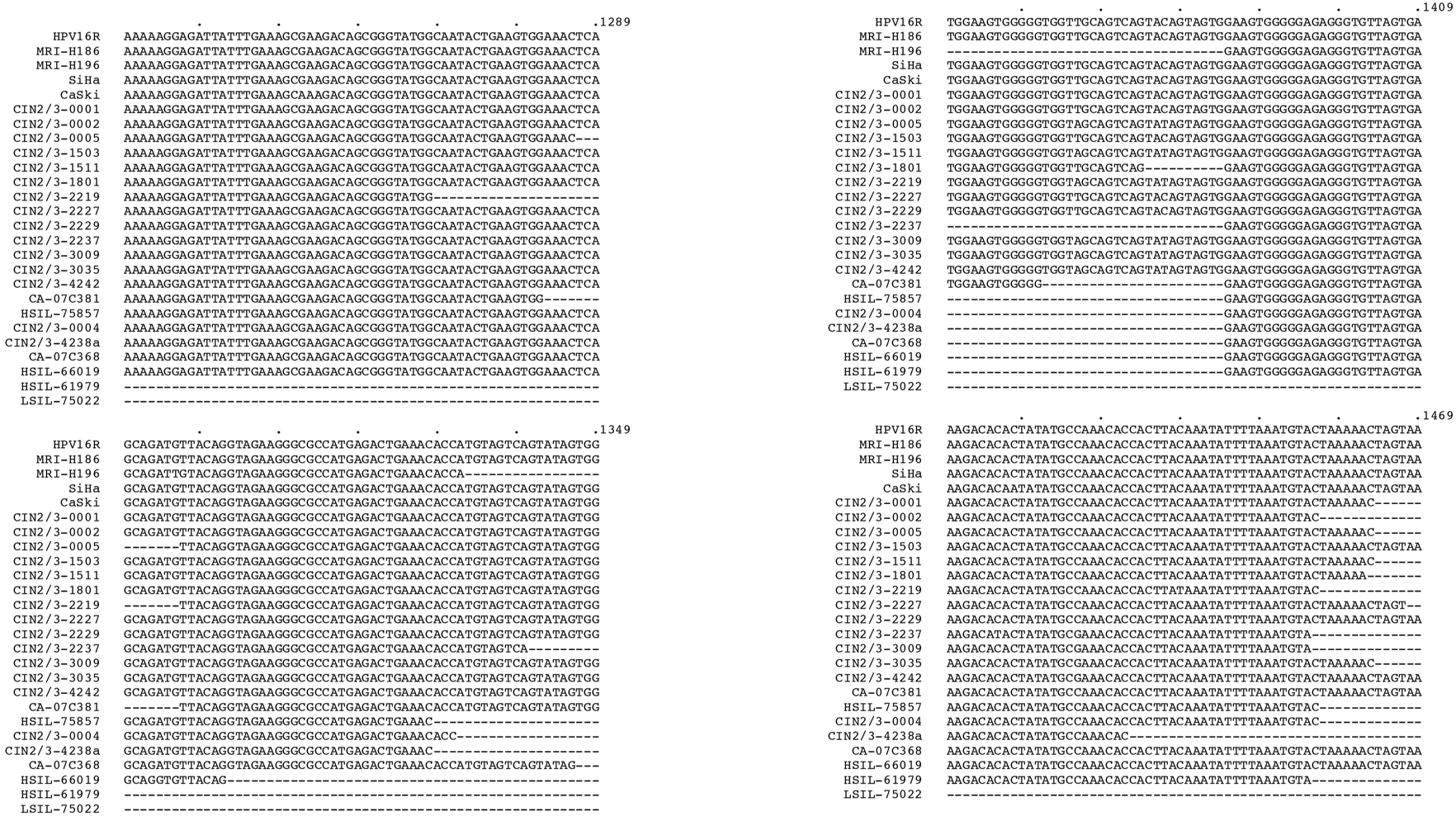
See Results section 2.2.2.6, pages 82-83. The nucleotide sequence shown here belongs to the junction-specific PCR product of primer pair 1801-2 (Results, Figure 2.47). Red color indicates HPV16 DNA between pos. 1656-1913, green color the 4 bp of unidentified origin, and black color the flanking cellular DNA.

```
  1 TCAAAGTTTA GCATGTTCAT GGGGAATGGT TGTGTTACTA TTAGTAAGAT ATAAATGTGG
 61 AAAAAATAGA GAAACAATTG AAAAATTGCT GTCTAAACTA TTATGTGTGT CTCCAATGTG
121 TATGATGATA GAGCCTCCAA AATTGCGTAG TACAGCAGCA GCATTATATT GGTATAAAAC
181 AGGTATATCA AATATTAGTG AAGTGTATGG AGACACGCCA GAATGGATAC AAAGACAAAC
241 AGTATTACAA CATAGTTTGG CATGCCCACT CCATGCCATC AGCCCCCAAG TCATGACGAC
301 TACAGGTATC TCCAGACATC ACCCAGTGTC CCCTGGGGCG GGGTGGGGGG CAGGGCTACC
361 CCCTTTGAGA ACCTCCTGTC TTGGAGGCAG CAAAC
```

A5. Representative HPV16 E1/E2 sequences of 25 DNA samples analyzed in ASP16 experiments

See Results section 2.2.2.8 and Figure 2.58. The assembled representative HPV16 E1-E2 nucleotide sequences of the 25 DNA samples (Results, Table 2.13 and Table 2.18), covering pos. 990-3853 are shown as an alignment. "**-**" indicates the regions where no sequence information is available.

```
                .         .         .         .         .        .1049
      HPV16R    TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
    MRI-H186    TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
    MRI-H196    TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
       SiHa     TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
      CaSki     TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
  CIN2/3-0001   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
  CIN2/3-0002   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
  CIN2/3-0005   TGATACAGGTGAAGATTTGGTAGATTTCATAGTAAATGATAATGATTATTTAACACAGGC
  CIN2/3-1503   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
  CIN2/3-1511   TGATACAGGTGAAGATTTGGTAGATTTCATAGTAAATGATAATGATTATTTAACACAGGC
  CIN2/3-1801   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
  CIN2/3-2219   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
  CIN2/3-2227   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
  CIN2/3-2229   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
  CIN2/3-2237   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
  CIN2/3-3009   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
  CIN2/3-3035   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
  CIN2/3-4242   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
    CA-07C381   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
   HSIL-75857   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
  CIN2/3-0004   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
 CIN2/3-4238a   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
    CA-07C368   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
   HSIL-66019   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
   HSIL-61979   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC
   LSIL-75022   TGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGC

                .         .         .         .         .        .1109
      HPV16R    AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGC
    MRI-H186    AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGC
    MRI-H196    AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGC
       SiHa     AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGC
      CaSki     AGAAACAGAGACAGCACATGCGTTGTTTAC-----AGGAAGCAAAACAACATAGAGATGC
  CIN2/3-0001   AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGC
  CIN2/3-0002   AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGC
  CIN2/3-0005   AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAAGAACATAGAGATGC
  CIN2/3-1503   AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGC
  CIN2/3-1511   AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAAGAACATACAGATGC
  CIN2/3-1801   AGACACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGC
  CIN2/3-2219   AGAAACAGAGACAGCACATGCGGTGTTTACTGCACAGGAAGCAAAAGAACATAGAGATGC
  CIN2/3-2227   AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGC
  CIN2/3-2229   AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGC
  CIN2/3-2237   AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAAGAACATAGAGATGC
  CIN2/3-3009   AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAAGAACATAGAGATGC
  CIN2/3-3035   AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAAGAACATAGAGATGC
  CIN2/3-4242   AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAAGAACATAGAGATGC
    CA-07C381   AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGC
   HSIL-75857   AGAAACAGAGAC----------------------AGGAAGCAAAACAACATAGAGATGC
  CIN2/3-0004   AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGC
 CIN2/3-4238a   AGAAACAGAGACAGCACATGCGTTGTTTA------AGGAAGCAAAACAACATAGAGATGC
    CA-07C368   AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGC
   HSIL-66019   AGAAACAGAGACAGCACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGC
   HSIL-61979   AGAAACA---------------------------AGGAAGCAAAACAACATAGAGATGC
   LSIL-75022   AGAAACA---------------------------AGGAAGCAAAACAACATAGAGATGC

                .         .         .         .         .        .1169
      HPV16R    AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
    MRI-H186    AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
    MRI-H196    AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
       SiHa     AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
      CaSki     AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
  CIN2/3-0001   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
  CIN2/3-0002   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
  CIN2/3-0005   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
  CIN2/3-1503   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
  CIN2/3-1511   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
  CIN2/3-1801   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
  CIN2/3-2219   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
  CIN2/3-2227   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
  CIN2/3-2229   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
  CIN2/3-2237   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGAATGTGT
  CIN2/3-3009   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGAATGTGT
  CIN2/3-3035   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGAATGTGT
  CIN2/3-4242   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGAATGTGT
    CA-07C381   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
   HSIL-75857   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
  CIN2/3-0004   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
 CIN2/3-4238a   AGTACAGGTTCTAAAAC----------GTAGTCCACTTAGTGATATTAGTGGATGTGT
    CA-07C368   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
   HSIL-66019   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
   HSIL-61979   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT
   LSIL-75022   AGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGT

                .         .         .         .         .        .1229
      HPV16R    AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
    MRI-H186    AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
    MRI-H196    AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
       SiHa     AGACAATAATATTAGTCCTAGATTGAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
      CaSki     AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
  CIN2/3-0001   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
  CIN2/3-0002   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
  CIN2/3-0005   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
  CIN2/3-1503   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
  CIN2/3-1511   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
  CIN2/3-1801   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
  CIN2/3-2219   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAACCAAAGTAGAGCTGC
  CIN2/3-2227   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
  CIN2/3-2229   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
  CIN2/3-2237   AGACAATAATATTAGTCCTAGATTAAAAGCCATATGTATAGAAAAACAAAGTAGAGCTGC
  CIN2/3-3009   AGACAATAATATTAGTCCTAGATTAAAAGCCATATGTATAGAAAAACAAAGTAGAGCTGC
  CIN2/3-3035   AGACAATAATATTAGTCCTAGATTAAAAGCCATATGTATAGAAAAACAAAGTAGAGCTGC
  CIN2/3-4242   AGACAATAATATTAGTCCTAGATTAAAAGCCATATGTATAGAAAAACAAAGTAGAGCTGC
    CA-07C381   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
   HSIL-75857   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
  CIN2/3-0004   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
 CIN2/3-4238a   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
    CA-07C368   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
   HSIL-66019   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAAAGTAGAGCTGC
   HSIL-61979   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAACAA-----------
   LSIL-75022   AGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAAA--------------
```

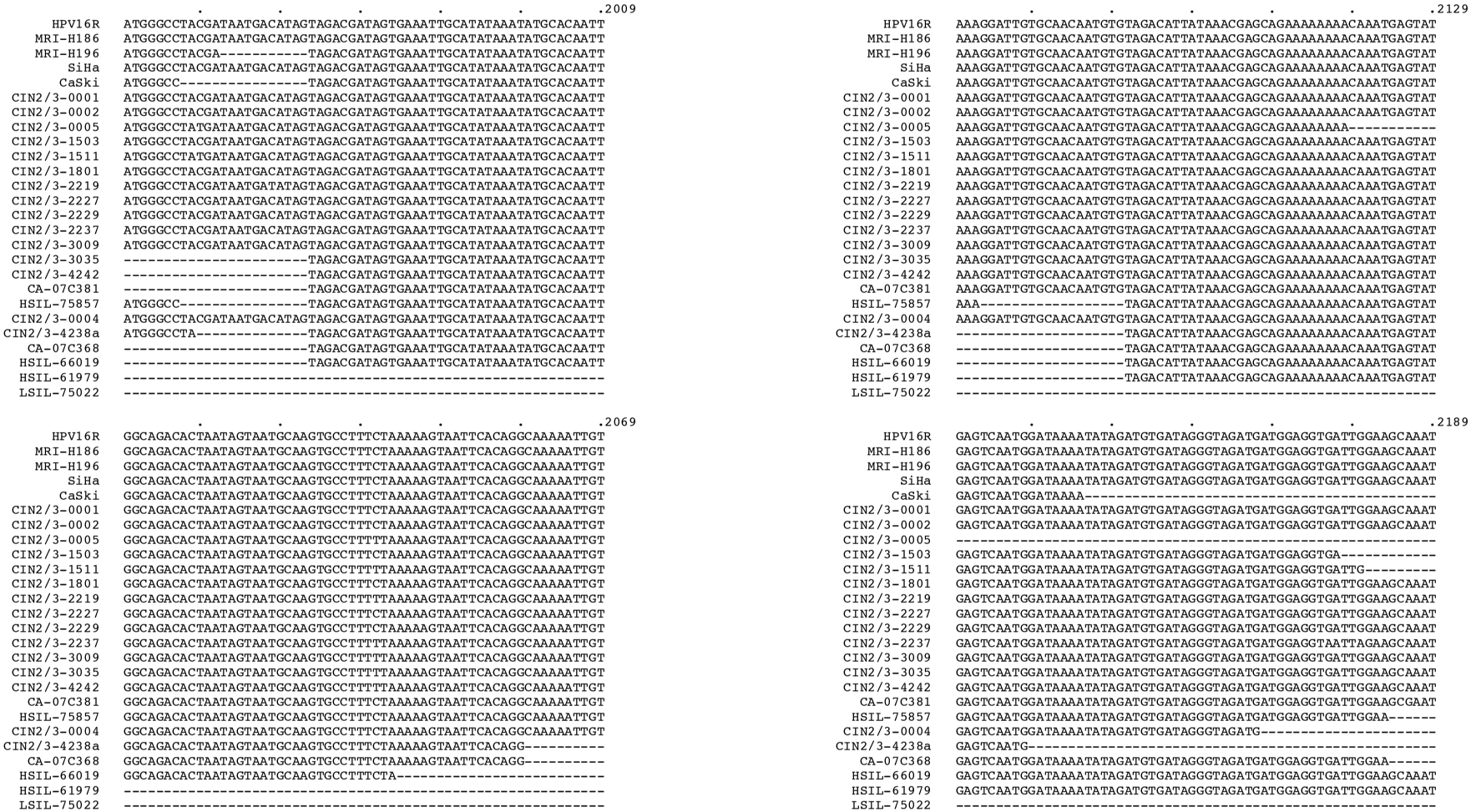

1289
HPV16R AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
MRI-H186 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
MRI-H196 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
SiHa AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CaSki AAAAAGGAGATTATTTGAAAGCAAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CIN2/3-0001 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CIN2/3-0002 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CIN2/3-0005 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAAC---
CIN2/3-1503 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CIN2/3-1511 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CIN2/3-1801 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CIN2/3-2219 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGG----------
CIN2/3-2227 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CIN2/3-2229 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CIN2/3-2237 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CIN2/3-3009 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CIN2/3-3035 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CIN2/3-4242 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CA-07C381 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGG-------
HSIL-75857 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CIN2/3-0004 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CIN2/3-4238a AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
CA-07C368 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
HSIL-66019 AAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTATGGCAATACTGAAGTGGAAACTCA
HSIL-61979 ----------
LSIL-75022 ----------
1349
HPV16R GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
MRI-H186 GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
MRI-H196 GCAGATTGTACAGGTAGAAGGGCGCCATGAGACTGAAACACCA----------
SiHa GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
CaSki GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
CIN2/3-0001 GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
CIN2/3-0002 GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
CIN2/3-0005 -------TTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
CIN2/3-1503 GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
CIN2/3-1511 GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
CIN2/3-1801 GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
CIN2/3-2219 -------TTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
CIN2/3-2227 GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
CIN2/3-2229 GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
CIN2/3-2237 GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCA---------
CIN2/3-3009 GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
CIN2/3-3035 GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
CIN2/3-4242 GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
CA-07C381 -------TTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGG
HSIL-75857 GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAAC----------
CIN2/3-0004 GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACC----------
CIN2/3-4238a GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAAC----------
CA-07C368 GCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAG---
HSIL-66019 GCAGGTGTTACAG----------
HSIL-61979 ----------
LSIL-75022 ----------
1409
HPV16R TGGAAGTGGGGGTGGTTGCAGTCAGTACAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGA
MRI-H186 TGGAAGTGGGGGTGGTTGCAGTCAGTACAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGA
MRI-H196 ----------GAAGTGGGGGAGAGGGTGTTAGTGA
SiHa TGGAAGTGGGGGTGGTTGCAGTCAGTACAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGA
CaSki TGGAAGTGGGGGTGGTTGCAGTCAGTACAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGA
CIN2/3-0001 TGGAAGTGGGGGTGGTTGCAGTCAGTACAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGA
CIN2/3-0002 TGGAAGTGGGGGTGGTTGCAGTCAGTACAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGA
CIN2/3-0005 TGGAAGTGGGGGTGGTAGCAGTCAGTATAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGA
CIN2/3-1503 TGGAAGTGGGGGTGGTTGCAGTCAGTACAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGA
CIN2/3-1511 TGGAAGTGGGGGTGGTAGCAGTCAGTATAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGA
CIN2/3-1801 TGGAAGTGGGGGTGGTTGCAGTCAG----------GAAGTGGGGGAGAGGGTGTTAGTGA
CIN2/3-2219 TGGAAGTGGGGGTGGTAGCAGTCAGTATAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGA
CIN2/3-2227 TGGAAGTGGGGGTGGTTGCAGTCAGTACAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGA
CIN2/3-2229 TGGAAGTGGGGGTGGTTGCAGTCAGTACAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGA
CIN2/3-2237 ----------GAAGTGGGGGAGAGGGTGTTAGTGA
CIN2/3-3009 TGGAAGTGGGGGTGGTAGCAGTCAGTATAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGA
CIN2/3-3035 TGGAAGTGGGGGTGGTAGCAGTCAGTATAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGA
CIN2/3-4242 TGGAAGTGGGGGTGGTAGCAGTCAGTATAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGA
CA-07C381 TGGAAGTGGGGG----------GAAGTGGGGGAGAGGGTGTTAGTGA
HSIL-75857 ----------GAAGTGGGGGAGAGGGTGTTAGTGA
CIN2/3-0004 ----------GAAGTGGGGGAGAGGGTGTTAGTGA
CIN2/3-4238a ----------GAAGTGGGGGAGAGGGTGTTAGTGA
CA-07C368 ----------GAAGTGGGGGAGAGGGTGTTAGTGA
HSIL-66019 ----------GAAGTGGGGGAGAGGGTGTTAGTGA
HSIL-61979 ----------GAAGTGGGGGAGAGGGTGTTAGTGA
LSIL-75022 ----------
1469
HPV16R AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTACTAAAAACTAGTAA
MRI-H186 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTACTAAAAACTAGTAA
MRI-H196 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTACTAAAAACTAGTAA
SiHa AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTACTAAAAACTAGTAA
CaSki AAGACACAATATATGCCAAACACCACTTACAAATATTTTAAATGTACTAAAAACTAGTAA
CIN2/3-0001 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTACTAAAAAC------
CIN2/3-0002 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTAC----------
CIN2/3-0005 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTACTAAAAAC------
CIN2/3-1503 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTACTAAAAACTAGTAA
CIN2/3-1511 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTACTAAAAAC------
CIN2/3-1801 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTACTAAAAA-------
CIN2/3-2219 AAGACACACTATATGCCAAACACCACTTATAAATATTTTAAATGTAC----------
CIN2/3-2227 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTACTAAAAACTAGT--
CIN2/3-2229 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTACTAAAAACTAGTAA
CIN2/3-2237 AAGACATACTATATGCGAAACACCACTTACAAATATTTTAAATGTA----------
CIN2/3-3009 AAGACACACTATATGCGAAACACCACTTACAAATATTTTAAATGTA----------
CIN2/3-3035 AAGACACACTATATGCGAAACACCACTTACAAATATTTTAAATGTACTAAAAAC------
CIN2/3-4242 AAGACACACTATATGCGAAACACCACTTACAAATATTTTAAATGTACTAAAAACTAGTAA
CA-07C381 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTACTAAAAACTAGTAA
HSIL-75857 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTAC----------
CIN2/3-0004 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTAC----------
CIN2/3-4238a AAGACACACTATATGCCAAACAC----------
CA-07C368 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTACTAAAAACTAGTAA
HSIL-66019 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTACTAAAAACTAGTAA
HSIL-61979 AAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTA----------
LSIL-75022 ----------

```
                .       .       .       .       .      .1529
HPV16R      TGCAAAGGCAGCAATGTTAGCAAAATTTAAAGAGTTATACGGGGTGAGTTTTTCAGAATT
MRI-H186    TGCAAAGGCAGCAATGTTAGCAAAATTTAAAGAGTTATACGGGGTGAGTTTTTCAGAATT
MRI-H196    TGCAAAGGCAGCAATGTTAGCAAAATTTAAAGAGTTATACGGGGTAAGTTTTTCAGAATT
SiHa        TGCAAAGGCAGCAATGTTAGCAAAATTTAAAGAGTTATACGGGGTGAGTTTTTCAGAATT
CaSki       TGCAAAGGCAGCAATGTTAGCAAAATTTAAAGAGTTATACGGGGTGAGTTTTACAGAATT
CIN2/3-0001 ----------------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTACAGAATT
CIN2/3-0002 ----------------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTACAGAATT
CIN2/3-0005 ----------------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTTCAGAATT
CIN2/3-1503 TGCAAAGGCAGCAATGTTAGCAAAATTTAAAGAGTTATACGGGGTGAGTTTTACAGAATT
CIN2/3-1511 ----------------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTTCAGAATT
CIN2/3-1801 ----------------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTTCAGAATT
CIN2/3-2219 ----------------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTTCAGAATT
CIN2/3-2227 ----------------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTTCAGAATT
CIN2/3-2229 TGCAAAGGCAGCAATGTTAGCAAAATTTAAAGAGTTATACGGGGTGAGTTTTACAGAATT
CIN2/3-2237 ----------------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTTCAGAATT
CIN2/3-3009 ----------------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTACAGAATT
CIN2/3-3035 ----------------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTACAGAATT
CIN2/3-4242 TGCAAAGGCAGCAATGCTAGCAAAATTTAAAGAGTTATACGGGGTGAGTTTTACAGAATT
CA-07C381   TGCAAAGGCAGCAATGTTAGCAAAATTTAAAGAGTTATACGGGGTGAGTTTTTCAGAATT
HSIL-75857  ----------------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTACAGAATT
CIN2/3-0004 ----------------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTTCAGAATT
CIN2/3-4238a ---------------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTACAGAATT
CA-07C368   TGCAAAGGCAGCAATGTTAGCAAAATTTAAAGAGTTATACGGGGTGAGTTTTTCAGAATT
HSIL-66019  GGCAAA----------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTACAGAATT
HSIL-61979  ----------------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTACAGAATT
LSIL-75022  ----------------------AAAATTTAAAGAGTTATACGGGGTGAGTTTTTCAGAATT

                .       .       .       .       .      .1589
HPV16R      AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
MRI-H186    AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTCGG
MRI-H196    AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
SiHa        AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CaSki       AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CIN2/3-0001 AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CIN2/3-0002 AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CIN2/3-0005 AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CIN2/3-1503 AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CIN2/3-1511 AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CIN2/3-1801 AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CIN2/3-2219 AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CIN2/3-2227 AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CIN2/3-2229 AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CIN2/3-2237 AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CIN2/3-3009 AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CIN2/3-3035 AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CIN2/3-4242 AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CA-07C381   AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCAT----
HSIL-75857  AGTAAGACCATTTAAA----------------------------------------------
CIN2/3-0004 AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
CIN2/3-4238a AGTAAGACCA---------------------------------------------------
CA-07C368   AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
HSIL-66019  AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
HSIL-61979  AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
LSIL-75022  AGTAAGACCATTTAAAAGTAATAAATCAACGTGTTGCGATTGGTGTATTGCTGCATTTGG
```

```
                .       .       .       .       .      .1649
HPV16R      ACTTACACCCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
MRI-H186    ACTTACACCCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
MRI-H196    ACTTACACCCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
SiHa        ACTTACACCCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
CaSki       ACTTACACCCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
CIN2/3-0001 ACTTACACCCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
CIN2/3-0002 ACTTACACCCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
CIN2/3-0005 ACTTACACCCAGTATAGCTGACAGTATAAAAACATTATTACAACAATATTGTTTATATTT
CIN2/3-1503 ACTTACACCCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
CIN2/3-1511 ACTTACACCCAGTATAGCTGACAGTATAAAAACATTATTACAACAATATTGTTTATATTT
CIN2/3-1801 ACTTACACCCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTGTATATTT
CIN2/3-2219 ACTTACACCCTAGTATAGCTGACAGTATAAAAACATTATTACAACAATATTGTTTATATTT
CIN2/3-2227 ACTTACACCCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
CIN2/3-2229 ACTTACACCCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
CIN2/3-2237 AC------CCAGTATAGCTGACAGTATAAAAACATTATTACAACAATATTGTTTATATTT
CIN2/3-3009 ACTTACACCCAGTATAGCTGACAGTATAAAAACATTATTACAACAATATTGTTTATATTT
CIN2/3-3035 ACTTACACCCAGTATAGCTGACAGTATAAAAACATTATTACAACAATATTGTTTATATTT
CIN2/3-4242 ACTTACACCCAGTATAGCTGACAGTATAAAAACATTATTACAACAATATTGTTTATATTT
CA-07C381   --------CCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
HSIL-75857  --------CCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
CIN2/3-0004 ACTTACACCCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
CIN2/3-4238a --------CCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
CA-07C368   ACTTACACCCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
HSIL-66019  ACT-----CCAGTATAGCTGACAGTATAAAAACACTATTACAACAAT-------------
HSIL-61979  ACTTACACCCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT
LSIL-75022  AC------CCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTT

                .       .       .       .       .      .1709
HPV16R      ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
MRI-H186    ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
MRI-H196    ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
SiHa        ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
CaSki       ACACATTCAA----------------------------------TATTAGTAAGATATAA
CIN2/3-0001 ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
CIN2/3-0002 ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
CIN2/3-0005 ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
CIN2/3-1503 ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
CIN2/3-1511 ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
CIN2/3-1801 ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
CIN2/3-2219 ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
CIN2/3-2227 ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
CIN2/3-2229 ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
CIN2/3-2237 ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
CIN2/3-3009 ACACATTCAAAGTTTAGCGTGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
CIN2/3-3035 ACACATTCAAAGTTTAGCGTGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
CIN2/3-4242 ACACATTCAAAGTTTAGCGTGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
CA-07C381   ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
HSIL-75857  ACACATTCAAAGTTTAGCATGTTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAA
CIN2/3-0004 ACACA----------------------------------------TATTAGTAAGATATAA
CIN2/3-4238a ACACA---------------------------------------TATTAGTAAGATATAA
CA-07C368   ACACATT----------------------------------TATTAGTAAGATATAA
HSIL-66019  ------------------------------------------TATTAGTAAGATATAA
HSIL-61979  ACACA-----------------------------------------------------
LSIL-75022  ACACATTCAAAGTTTAGCATGTT--------------------TATTAGTAAGATATAA
```

```
                .         .         .         .         .        .1769
HPV16R      ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
MRI-H186    ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
MRI-H196    ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
SiHa        ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
CaSki       ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
CIN2/3-0001 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
CIN2/3-0002 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
CIN2/3-0005 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
CIN2/3-1503 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
CIN2/3-1511 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
CIN2/3-1801 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
CIN2/3-2219 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
CIN2/3-2227 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
CIN2/3-2229 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
CIN2/3-2237 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGATGTCTAAACTATTATGTGTGTCTCC
CIN2/3-3009 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGATGTCTAAACTATTATGTGTGTCTCC
CIN2/3-3035 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGATGTCTAAACTATTATGTGTGTCTCC
CIN2/3-4242 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGATGTCTAAACTATTATGTGTGTCTCC
  CA-07C381 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
 HSIL-75857 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
CIN2/3-0004 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
CIN2/3-4238a ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
  CA-07C368 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
 HSIL-66019 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC
 HSIL-61979 ------------------------------------------------------------
 LSIL-75022 ATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTCTCC

                .         .         .         .         .        .1829
HPV16R      AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
MRI-H186    AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
MRI-H196    AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
SiHa        AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CaSki       AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CIN2/3-0001 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CIN2/3-0002 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CIN2/3-0005 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CIN2/3-1503 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CIN2/3-1511 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CIN2/3-1801 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CIN2/3-2219 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CIN2/3-2227 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CIN2/3-2229 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CIN2/3-2237 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CIN2/3-3009 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CIN2/3-3035 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CIN2/3-4242 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
  CA-07C381 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
 HSIL-75857 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CIN2/3-0004 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
CIN2/3-4238a AATGTGTATGATGATAGAGCCTCCAAA---------TACAGCAGCAGCATTATATTGGTA
  CA-07C368 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
 HSIL-66019 AATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTA
 HSIL-61979 ----------------------------------TACAGCAGCAGCATTATATTGGTA
 LSIL-75022 AATGTGTATGATGATAGAGCCTCCAAA--------------------------------
```

```
                .         .         .         .         .        .1889
HPV16R      TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
MRI-H186    TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
MRI-H196    TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
SiHa        TAAAACAGGTATGTCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
CaSki       TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAA--GATACAAAG
CIN2/3-0001 TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
CIN2/3-0002 TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
CIN2/3-0005 TAAAACAGGTATGTCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
CIN2/3-1503 TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCTAGAATGGATACAAAG
CIN2/3-1511 TAAAACAGGTATGTCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
CIN2/3-1801 TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
CIN2/3-2219 TAAAACAGGTATGTCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
CIN2/3-2227 TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
CIN2/3-2229 TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
CIN2/3-2237 TAAAACAGGTATGTCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
CIN2/3-3009 TAAAACAGGTATGTCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
CIN2/3-3035 TAAAACAGGTATGTCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
CIN2/3-4242 TAAAACAGGTATGTCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
  CA-07C381 TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
 HSIL-75857 TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
CIN2/3-0004 TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
CIN2/3-4238a TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
  CA-07C368 TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
 HSIL-66019 TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAATGGATACAAAG
 HSIL-61979 TAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAA-----------
 LSIL-75022 ------------------------------------------------------------

                .         .         .         .         .        .1949
HPV16R      ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
MRI-H186    ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
MRI-H196    ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
SiHa        ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
CaSki       ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
CIN2/3-0001 ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
CIN2/3-0002 ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
CIN2/3-0005 ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
CIN2/3-1503 ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
CIN2/3-1511 ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
CIN2/3-1801 ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
CIN2/3-2219 ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
CIN2/3-2227 ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
CIN2/3-2229 ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
CIN2/3-2237 ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCATAGATGGTACA
CIN2/3-3009 ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
CIN2/3-3035 ACAAACAGTATTACAACATAGTTTTAATGATTGTACACTTGAATTATCACA----------
CIN2/3-4242 ACAAACAGTATTACAACA------------------------------------------
  CA-07C381 ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCAC----------
 HSIL-75857 ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
CIN2/3-0004 ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
CIN2/3-4238a ACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACA
  CA-07C368 ACAAACAGTATTACAACATAG------------------------------------------
 HSIL-66019 ACAAACA-------------------------------------------------------
 HSIL-61979 ------------------------------------------------------------
 LSIL-75022 ------------------------------------------------------------
```


. 2009
HPV16R ATGGGCCTACGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATT
MRI-H186 ATGGGCCTACGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATT
MRI-H196 ATGGGCCTACGA----------TAGACGATAGTGAAATTGCATATAAATATGCACAATT
SiHa ATGGGCCTACGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATT
CaSki ATGGGCC---------------TAGACGATAGTGAAATTGCATATAAATATGCACAATT
CIN2/3-0001 ATGGGCCTACGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATT
CIN2/3-0002 ATGGGCCTACGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATT
CIN2/3-0005 ATGGGCCTATGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATT
CIN2/3-1503 ATGGGCCTACGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATT
CIN2/3-1511 ATGGGCCTATGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATT
CIN2/3-1801 ATGGGCCTACGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATT
CIN2/3-2219 ATGGGCCTACGATAATGATATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATT
CIN2/3-2227 ATGGGCCTACGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATT
CIN2/3-2229 ATGGGCCTACGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATT
CIN2/3-2237 ATGGGCCTACGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATT
CIN2/3-3009 ATGGGCCTACGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATT
CIN2/3-3035 ----------TAGACGATAGTGAAATTGCATATAAATATGCACAATT
CIN2/3-4242 ----------TAGACGATAGTGAAATTGCATATAAATATGCACAATT
CA-07C381 ----------TAGACGATAGTGAAATTGCATATAAATATGCACAATT
HSIL-75857 ATGGGCC---------------TAGACGATAGTGAAATTGCATATAAATATGCACAATT
CIN2/3-0004 ATGGGCCTACGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATT
CIN2/3-4238a ATGGGCCTA--------------TAGACGATAGTGAAATTGCATATAAATATGCACAATT
CA-07C368 ----------TAGACGATAGTGAAATTGCATATAAATATGCACAATT
HSIL-66019 ----------TAGACGATAGTGAAATTGCATATAAATATGCACAATT
HSIL-61979 ----------
LSIL-75022 ----------
. 2069
HPV16R GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGGCAAAAATTGT
MRI-H186 GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGGCAAAAATTGT
MRI-H196 GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGGCAAAAATTGT
SiHa GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGGCAAAAATTGT
CaSki GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGGCAAAAATTGT
CIN2/3-0001 GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGGCAAAAATTGT
CIN2/3-0002 GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGGCAAAAATTGT
CIN2/3-0005 GGCAGACACTAATAGTAATGCAAGTGCCTTTTTAAAAAGTAATTCACAGGCAAAAATTGT
CIN2/3-1503 GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGGCAAAAATTGT
CIN2/3-1511 GGCAGACACTAATAGTAATGCAAGTGCCTTTTTAAAAAGTAATTCACAGGCAAAAATTGT
CIN2/3-1801 GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGGCAAAAATTGT
CIN2/3-2219 GGCAGACACTAATAGTAATGCAAGTGCCTTTTTAAAAAGTAATTCACAGGCAAAAATTGT
CIN2/3-2227 GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGGCAAAAATTGT
CIN2/3-2229 GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGGCAAAAATTGT
CIN2/3-2237 GGCAGACACTAATAGTAATGCAAGTGCCTTTTTAAAAAGTAATTCACAGGCAAAAATTGT
CIN2/3-3009 GGCAGACACTAATAGTAATGCAAGTGCCTTTTTAAAAAGTAATTCACAGGCAAAAATTGT
CIN2/3-3035 GGCAGACACTAATAGTAATGCAAGTGCCTTTTTAAAAAGTAATTCACAGGCAAAAATTGT
CIN2/3-4242 GGCAGACACTAATAGTAATGCAAGTGCCTTTTTAAAAAGTAATTCACAGGCAAAAATTGT
CA-07C381 GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGGCAAAAATTGT
HSIL-75857 GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGGCAAAAATTGT
CIN2/3-0004 GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGGCAAAAATTGT
CIN2/3-4238a GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGG----------
CA-07C368 GGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAAAAGTAATTCACAGG----------
HSIL-66019 GGCAGACACTAATAGTAATGCAAGTGCCTTTCTA----------
HSIL-61979 ----------
LSIL-75022 ----------
. 2129
HPV16R AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAAACAAATGAGTAT
MRI-H186 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
MRI-H196 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
SiHa AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CaSki AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CIN2/3-0001 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CIN2/3-0002 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CIN2/3-0005 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAAA----------
CIN2/3-1503 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CIN2/3-1511 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CIN2/3-1801 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CIN2/3-2219 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CIN2/3-2227 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CIN2/3-2229 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CIN2/3-2237 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CIN2/3-3009 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CIN2/3-3035 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CIN2/3-4242 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CA-07C381 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
HSIL-75857 AAA----------TAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CIN2/3-0004 AAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CIN2/3-4238a ----------TAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
CA-07C368 ----------TAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
HSIL-66019 ----------TAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
HSIL-61979 ----------TAGACATTATAAACGAGCAGAAAAAAACAAATGAGTAT
LSIL-75022 ----------
. 2189
HPV16R GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCAAAT
MRI-H186 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCAAAT
MRI-H196 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCAAAT
SiHa GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCAAAT
CaSki GAGTCAATGGATAAAA----------
CIN2/3-0001 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCAAAT
CIN2/3-0002 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCAAAT
CIN2/3-0005 ----------
CIN2/3-1503 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGA----------
CIN2/3-1511 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTG---------
CIN2/3-1801 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCAAAT
CIN2/3-2219 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCAAAT
CIN2/3-2227 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCAAAT
CIN2/3-2229 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCAAAT
CIN2/3-2237 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTAATTAGAAGCAAAT
CIN2/3-3009 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCAAAT
CIN2/3-3035 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCAAAT
CIN2/3-4242 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCAAAT
CA-07C381 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCGAAT
HSIL-75857 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAA------
CIN2/3-0004 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATG----------
CIN2/3-4238a GAGTCAATG----------
CA-07C368 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAA------
HSIL-66019 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCAAAT
HSIL-61979 GAGTCAATGGATAAAATATAGATGTGATAGGGTAGATGATGGAGGTGATTGGAAGCAAAT
LSIL-75022 ----------

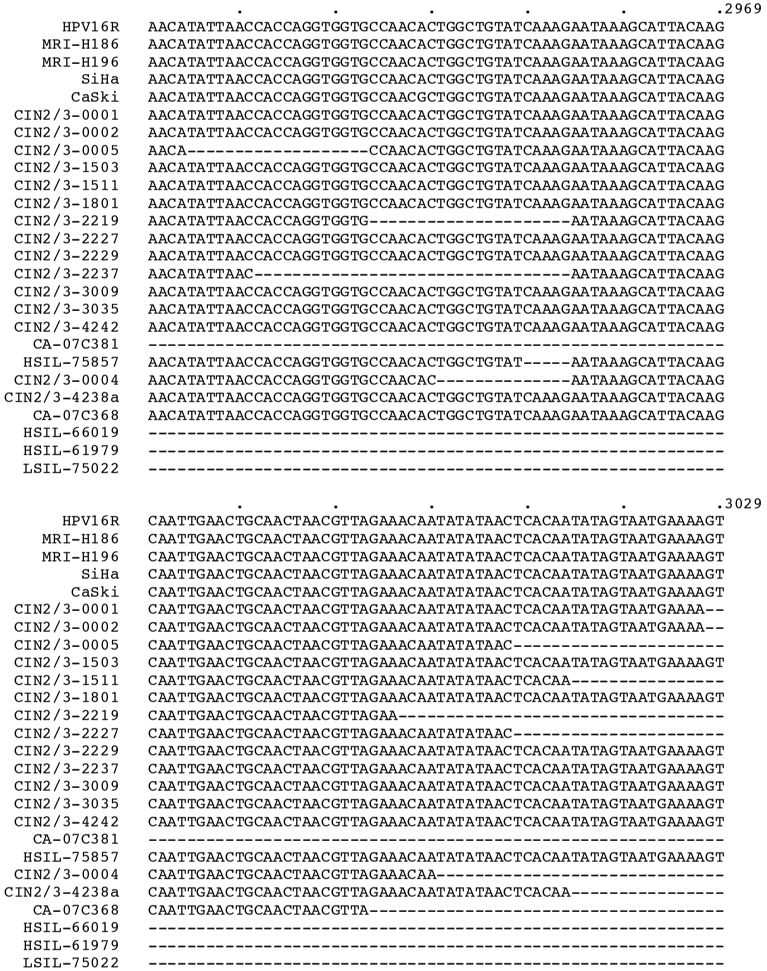
```
                .         .         .         .         .        .2249
HPV16R      TGTTATGTTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
MRI-H186    TGTTATGTTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
MRI-H196    TGTTATGTTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
SiHa        TGTTATGTTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
CaSki       -------TTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
CIN2/3-0001 TGTTATGTTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
CIN2/3-0002 TGTTATGTTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
CIN2/3-0005 -------TTTTTAAGGTATCAAGGTGTAGACTTTATGTCATTTTTAACTGCATTAAAAAG
CIN2/3-1503 -------TTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
CIN2/3-1511 -------TTTTTAAGGTATCAAGGTGTAGACTTTATGTCATTTTTAACTGCATTAAAAAG
CIN2/3-1801 TGTTATGTTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
CIN2/3-2219 TGTTATGTTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
CIN2/3-2227 TGTTATGTTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
CIN2/3-2229 TGTTATGTTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
CIN2/3-2237 TGTTATGTTTTTAAGGTATCAAGGTGTAGACTTTATGTCATTTTTAAGTGCATTAAAAAA
CIN2/3-3009 TGTTATGTTTTTAAGGTATCAAGGTGTAGACTTTATGTCATTTTTAAGTGCATTAAAAAA
CIN2/3-3035 TGTTATGTTTTTAAGGTATCAAGGTGTAGACTTTATGTCATTTTTAAGTGCATTAAAAAA
CIN2/3-4242 TGTTATGTTTTTAAGGTATCAAGGTGTAGACTTTATGTCATTTTTAAGTGCATTAAAAAA
  CA-07C381 TGTTATGTTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
 HSIL-75857 -------TTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
CIN2/3-0004 -------TTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
CIN2/3-4238a -------TTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
  CA-07C368 -------TTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
 HSIL-66019 TGTTATGTTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
 HSIL-61979 -------TTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAG
 LSIL-75022 ------------------------------------------------------------

                .         .         .         .         .        .2309
HPV16R      ATTTTTGCAAGGCATACCTAAAAAAAATTGCATATTACTATATGGTGCAGCTAACACAGG
MRI-H186    ATTTTTGCAAGGCATACCTAAAAAAAATTGCATATTACTATATGGTGCAGC---------
MRI-H196    ATTTTTGCAAGGCATACCTAAAAAAAATTGCATATTACTATATGGTGCAGCTAACACAGG
SiHa        ATTTTTGCAAGGCATACCTAAAAAAAATTGCATATTACTATATGGTGCAGCCAACACAGG
CaSki       ATTTTTGCAAGGCATACCTAAAAAAAA----------------------------------
CIN2/3-0001 ATTTTTGCAAGGCATACCTAAAAAAAATTGCATATTACTATATGGTGCAGCTAACACAGG
CIN2/3-0002 ATTTTTGCAAGGCATACCTAAAAAAAATTGCATATTACTATATGGTGCAGCTAACACAGG
CIN2/3-0005 ATTTTTGCAAGGCATACCTAAAAAAAATTGCATATTACTATATGGTGCA-----------
CIN2/3-1503 ATTTTTGCAAGGCATACCTAAAAAAAATTGCATATTACTATATGGTGCAGC---------
CIN2/3-1511 ATTTTTGCAAGGCATACCTAAAAAAAATTGCATATTACTATATGGTGCAGCTAACACAGG
CIN2/3-1801 ATTTTTGCAAGGCATACCTAAAAAAAATTGCATATTACTATATGGTGCAGCTAACACAGG
CIN2/3-2219 ATTTTTGCAAGGCATACCTAAAAAA----------------------------------
CIN2/3-2227 ATTTTTGCAAGGCATACCTAAAAAAAATTGCATATTACTATATGGTGCAGCTAACACAGG
CIN2/3-2229 ATTTTTGCAAGGCATACCTAAAAAAAATTGCATATTACTATATGGTGCAGCTAACACAGG
CIN2/3-2237 ATTTTTGCAAGGTATACCTAAAAAAAATTGCATATTATTATATGGTGCAGCTAACACAGG
CIN2/3-3009 ATTTCTGCAAGGTATACCTAAAAAAAATTGCATATTATTATATGGTGCAGCTAACACAGG
CIN2/3-3035 ATTTCTGCAAGGTATACCTAAAAAAAATTGCATATTATTATATGGTGCAGCTAACACAGG
CIN2/3-4242 ATTTCTGCAAGGTATACCTAAAAAAAATTGCATATTATTATATGGTGCAGCTAACACAGG
  CA-07C381 ATTTTTGCAAGGCATACCTAAAAAAAATTGCATATTACTATATGGTGCAGCTAACACAGG
 HSIL-75857 ATTTTTGCAAGGCATACCTAAAAA----------------------------------
CIN2/3-0004 ATTTTTGCAAGGCATACCTAAAAAAAATTGCATATTACTATATGGTGCAGCTAACACAGG
CIN2/3-4238a ATTTTTGCAAGGCATACCTA------------------------------------------
  CA-07C368 ATTTTTGCAAGGCATACCT-------------------------------------------
 HSIL-66019 ATTTTTGCAAGGCATACCTAAAAAAAATTGCATATTACTATATGGTGCAGCTAACACAGG
 HSIL-61979 ATTTTTGCAAGGCATACCTAAAAAAA----------------------------------
 LSIL-75022 ------------------------------------------------------------
```

```
                .         .         .         .         .        .2369
HPV16R      TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
MRI-H186    TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
MRI-H196    TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
SiHa        TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
CaSki       TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
CIN2/3-0001 TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
CIN2/3-0002 TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
CIN2/3-0005 TAAATCATTATTTGGTATGAGTTTAATGAAATTTTTGCAAGGGTCTGTAATATGTTTTGT
CIN2/3-1503 TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
CIN2/3-1511 TAAATCATTATTTGGTATGAGTTTAATGAAATTTTTGCAAGGGTCTGTAATATGTTTTGT
CIN2/3-1801 TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
CIN2/3-2219 TAAATCATTATTTGGTATGAGTTTGATGAAATTTTTGCAAGGGTCTGTAATATGTTTTGT
CIN2/3-2227 TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
CIN2/3-2229 TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
CIN2/3-2237 TAAATCATTATTTGGTATGAGTTTAATGAAATTTTTGCAAGGGTCCGTAATATGTTTTGT
CIN2/3-3009 TAAATCATTATTTGGTATGAGTTTAATGAAATTTTTGCAAGGGTCTGTAATATGTTTTGT
CIN2/3-3035 TAAATCATTATTTGGTATGAGTTTAATGAAATTTTTGCAAGGGTCTGTAATATGTTTTGT
CIN2/3-4242 TAAATCATTATTTGGTATGAGTTTAATGAAATTTTTGCAAGGGTCTGTAATATGTTTTGT
  CA-07C381 TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
 HSIL-75857 TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
CIN2/3-0004 TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
CIN2/3-4238a TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
  CA-07C368 TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
 HSIL-66019 TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
 HSIL-61979 TAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGT
 LSIL-75022 ------------------------------------------------------------

                .         .         .         .         .        .2429
HPV16R      AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGCCAAAATAGGTATGTTAGA
MRI-H186    AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGCCAAAATAG-----TTAGA
MRI-H196    AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGCCAAAATAGGTATGTTAGA
SiHa        AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGCCAAA---------TTAGA
CaSki       AAATTCTAAAAGCCATTTTTGGTTACAA----------------------------TTAGA
CIN2/3-0001 AAATTCTAAAAGCCATTTTTGGTTACAACCA------------------------TTAGA
CIN2/3-0002 AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGCCAAAATAGGTATGTTAGA
CIN2/3-0005 AAATTCTAAAAGCCATTTTTGGTTACAACCAT-----------------------TTAGA
CIN2/3-1503 AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGCCAAAATAGGTATGTTAGA
CIN2/3-1511 AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGCCAAAATAGGTATGTTAGA
CIN2/3-1801 AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGCCAAAATAG-----TTAGA
CIN2/3-2219 AAATTCTAAAAGCCATTTTTGGTTACAACCATTAG--------------------TTAGA
CIN2/3-2227 AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGCCAAAATAGGTATGTTAGA
CIN2/3-2229 AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGCCAAAATAGGTATGTTAGA
CIN2/3-2237 ------------------------------------------------------TTAGA
CIN2/3-3009 AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGC-------------TTAGA
CIN2/3-3035 AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGCCAAAA--------TTAGA
CIN2/3-4242 AAATT-------------------------------------------------TTAGA
  CA-07C381 AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGCCAAAATAGGTATGTTAGA
 HSIL-75857 AAATTCTAAAAGCCATTTTTGGTTACAACCA------------------------TTAGA
CIN2/3-0004 AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGAT--------------TTAGA
CIN2/3-4238a AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGC------------TTAGA
  CA-07C368 AAATTCTAAAAGCCATTTTTGGTTACA----------------------------TTAGA
 HSIL-66019 AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGCCAAAATAGGTATGTTAG-
 HSIL-61979 AAATTCTAAAAGCCATTTTTGGTTACAACCATTAGCAGATGCCAAAA------------
 LSIL-75022 ------------------------------------------------------------
```

```
                .         .         .         .         .        .2489
HPV16R      TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
MRI-H186    TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
MRI-H196    TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
SiHa        TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
CaSki       TGATGCTACAGTGCCCTGTTGGAACTATATAGATGACAATTTAAGAAATGCATTGGATGG
CIN2/3-0001 TGATGCTACAGTGCCCTGTTGGAACT---------------------------------
CIN2/3-0002 TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
CIN2/3-0005 TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
CIN2/3-1503 TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
CIN2/3-1511 TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
CIN2/3-1801 TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
CIN2/3-2219 TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
CIN2/3-2227 TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
CIN2/3-2229 TGATGCTACAGTGCCCTGTTGGAACTATATAGATGACAATTTAAGAAATGCATTGGATGG
CIN2/3-2237 TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
CIN2/3-3009 TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
CIN2/3-3035 TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
CIN2/3-4242 TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
  CA-07C381 TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGG
 HSIL-75857 TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCAT-------
CIN2/3-0004 TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAA-------------
CIN2/3-4238a TGATGCTACAGTGCCCTGTTGGAACTATATAGATGACAATTTAAGAAATGCATTGGATGG
  CA-07C368 TGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAA------------
 HSIL-66019 ------------------------------------------------------------
 HSIL-61979 ------------------------------------------------------------
 LSIL-75022 ------------------------------------------------------------

                .         .         .         .         .        .2549
HPV16R      AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
MRI-H186    AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
MRI-H196    AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
SiHa        AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
CaSki       AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
CIN2/3-0001 ---------TTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCC------
CIN2/3-0002 AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
CIN2/3-0005 AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
CIN2/3-1503 AAATTTAGTTTTTATGGATGTAAAGCGTAGACCATTGGTACAACTAAAATGCCCTCCATT
CIN2/3-1511 AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
CIN2/3-1801 AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
CIN2/3-2219 AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
CIN2/3-2227 AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
CIN2/3-2229 AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
CIN2/3-2237 AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAA-----------
CIN2/3-3009 AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
CIN2/3-3035 AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
CIN2/3-4242 AAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
  CA-07C381 AAA-----TTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
 HSIL-75857 --------TTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
CIN2/3-0004 --------TTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
CIN2/3-4238a AAA-----TTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
  CA-07C368 --------TTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATG---------
 HSIL-66019 --------TTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATT
 HSIL-61979 --------TTTCTATGGATGTAAAGCA-------------------------------
 LSIL-75022 ------------------------------------------------------------
```

```
                .         .         .         .         .        .2609
HPV16R      ATTAATTACATCTAACATTAATGCTGGTACAGATTCTAGGTGGCCTTATTTACATAATAG
MRI-H186    ATTAATTACATCTAACATTAATGCTGGTACAGATTC------GCCTTATTTACATAATAG
MRI-H196    ATTAATTACATCTAACATTAATGCTGGTACAGATTCTAGGTGGCCTTATTTACATAATAG
SiHa        ATTAATTACATCTAACATTAATGCTGGTACAGATTCTAGGTGGCCTTATTTACATAATAG
CaSki       ATTAATTACATCTAACATTAATGCTGGTACAGA---------GCCTTATTTACATAATAG
CIN2/3-0001 ----------------------------------------GCCTTATTTACATAATAG
CIN2/3-0002 ATTAATTACATCTAACATTAATGCTGGTACAGATTCTAGGTGGCCTTATTTACATAATAG
CIN2/3-0005 ATTAATTACATCTAAC------------------------GCCTTATTTGCATAATCG
CIN2/3-1503 ATTAATTACATCTAACATTAATGCTGGTACAGATTCTAGGTGGCCTTATTTACATAATAG
CIN2/3-1511 ATTAATTACATCTAACATTAATGCTGGTACAGATTCCAGGTGGCCTTATTTGCATAATCG
CIN2/3-1801 ATTAATTACATCTAACATTAATGCTGGTACAGATTCTAGGTGGCCTTATTTACATAATAG
CIN2/3-2219 ATTAATTACATCTAACATTAATGCTGGTACAGATTCCAGGTGGCCTTATTTACATAATAG
CIN2/3-2227 ATTAATTACATCTAACATTAATGCTGGTACAGATTCTAG---GCCTTATTTACATAATAG
CIN2/3-2229 ATTAATTACATCTAACATTAATGCTGGTACAGATTCTAGGTGGCCTTATTTACATAATAG
CIN2/3-2237 ----------------------------------------GCCTTATTTACATAATCG
CIN2/3-3009 ATTAATTACATCTAACATTAATGCTGGTACAGATTCCAGGTGGCCTTATTTACATAATCG
CIN2/3-3035 ATTAATTACATCTAACATTAATGCTGGTACAGATTCCAGGTGGCCTTATTTACATAATCG
CIN2/3-4242 ATTAATTACATCTAACATTAATGCTGGTACAGATTCCAGGTGGCCTTATTTACATAATCG
  CA-07C381 ATTAATTACATC------------------------------GCCTTATTTACATAATAG
 HSIL-75857 ATTAATTACATCTAACATTAATGCTGGTACAGATTCTAG---GCCTTATTTACATAATAG
CIN2/3-0004 ATTAATTACATCTAACATTAATGCTGGTACA-----------GCCTTATTTACATAATAG
CIN2/3-4238a ATTAATTACATCTAACATTAATGCTGGTACA-----------GCCTTATTTACATAATAG
  CA-07C368 ----------------------------------------GCCTTATTTACATAATAG
 HSIL-66019 ATTAATTACATCTAACATTAATGCTGGTACA------------------------------
 HSIL-61979 ------------------------------------------------------------
 LSIL-75022 ------------------------------------------------------------

                .         .         .         .         .        .2669
HPV16R      ATTGGTGGTGTTTACATTTCCTAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
MRI-H186    ATTGGTGGTGTTTACATTTCCTAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
MRI-H196    ATTGGTGGTGTTTACATTTCCTAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
SiHa        ATTGGTGGTGTTTACATTTCCTAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
CaSki       ATTGGTGGTGTTTACATTTCCTAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
CIN2/3-0001 ATTGGTGGTGTTTACATTTCCTAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
CIN2/3-0002 ATTGGTGGTGTTTACATTTCCTAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
CIN2/3-0005 ATTGGTGGTGTTTACATTTCCAAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
CIN2/3-1503 ATTGGTGGTGTTTACATTTCCTAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
CIN2/3-1511 ATTGGTGGTGTTTACATTTCCAAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
CIN2/3-1801 ATTGGTGGTGTTTACATTTCCTAATGAGTTCCCATTTGACGAAAACAGAAATCCAGTGTA
CIN2/3-2219 ATTGGTGGTGTTTACATTTCCAAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
CIN2/3-2227 ATTGGTGGTGTTTACATTTCCTAATGAGTTTCCATTTGA--AAAACGGAAATCCAGTGTA
CIN2/3-2229 ATTGGTGGTGTTTACATTTCCTAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
CIN2/3-2237 ATTGGTGGTGTTTACATTTCCAAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
CIN2/3-3009 ATTGGTGGTGTTTACATTTCCAAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
CIN2/3-3035 ATTGGTGGTGTTTACATTTCCAAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
CIN2/3-4242 ATTGGTGGTGTTTACATTTCCAAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
  CA-07C381 ATTGGTGGTGTTTACATTTCCTAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
 HSIL-75857 ATTGGTGGTGTTTACATTTCCTAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
CIN2/3-0004 ATTGGTGGTGTTTACATTTCCTAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTA
CIN2/3-4238a ATTGGTGGTGTTTACATTTCCTAAT---------------AAAACGGAAATCCAGTGTA
  CA-07C368 ATTGGTGGTGTTTACATTTCCTAATGAGTTTCCATTTGA--AAAACGGAAATCCAGTGTA
 HSIL-66019 ----------------------------------------AAAACGGAAATCCAGTGTA
 HSIL-61979 ------------------------------------------------------------
 LSIL-75022 ------------------------------------------------------------
```

```
                .         .         .         .         .        .2729
HPV16R      TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
MRI-H186    TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
MRI-H196    TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
SiHa        TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
CaSki       TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
CIN2/3-0001 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
CIN2/3-0002 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
CIN2/3-0005 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTT--------------------------
CIN2/3-1503 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
CIN2/3-1511 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
CIN2/3-1801 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
CIN2/3-2219 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
CIN2/3-2227 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
CIN2/3-2229 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
CIN2/3-2237 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
CIN2/3-3009 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
CIN2/3-3035 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
CIN2/3-4242 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
  CA-07C381 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
 HSIL-75857 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAA----------------------
CIN2/3-0004 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
CIN2/3-4238a TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
  CA-07C368 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
 HSIL-66019 TGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGATTAAGTTT
 HSIL-61979 ------------------------------------------------------------
 LSIL-75022 ------------------------------------------------------------

                .         .         .         .         .        .2789
HPV16R      GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
MRI-H186    GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
MRI-H196    GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
SiHa        GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CaSki       GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CIN2/3-0001 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CIN2/3-0002 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CIN2/3-0005 -------------GACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CIN2/3-1503 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CIN2/3-1511 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CIN2/3-1801 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CIN2/3-2219 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CIN2/3-2227 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CIN2/3-2229 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CIN2/3-2237 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CIN2/3-3009 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CIN2/3-3035 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CIN2/3-4242 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
  CA-07C381 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATG------
 HSIL-75857 -------------GACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CIN2/3-0004 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
CIN2/3-4238a GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
  CA-07C368 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTC
 HSIL-66019 GCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATG------
 HSIL-61979 ------------------------------------------------------------
 LSIL-75022 ------------------------------------------------------------

                .         .         .         .         .        .2849
HPV16R      AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
MRI-H186    AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
MRI-H196    AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
SiHa        AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
CaSki       AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
CIN2/3-0001 AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTAC---------------
CIN2/3-0002 AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
CIN2/3-0005 AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
CIN2/3-1503 AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
CIN2/3-1511 AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
CIN2/3-1801 AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
CIN2/3-2219 AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
CIN2/3-2227 AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
CIN2/3-2229 AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
CIN2/3-2237 AGGACAAAATACTAACACATTATGAAAATGATAGTACA----------------------
CIN2/3-3009 AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
CIN2/3-3035 AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
CIN2/3-4242 AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
  CA-07C381 ------------------------------------------------------------
 HSIL-75857 AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
CIN2/3-0004 AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
CIN2/3-4238a AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
  CA-07C368 AGGACAAAATACTAACACATTATGAAAATGATAGTACAGACCTACGTGACCATATAGACT
 HSIL-66019 ------------------------------------------------------------
 HSIL-61979 ------------------------------------------------------------
 LSIL-75022 ------------------------------------------------------------

                .         .         .         .         .        .2909
HPV16R      ATTGGAAACACATGCGCCTAGAATGTGCTATTTATTACAAGGCCAGAGAAATGGGATTTA
MRI-H186    ATTGGAAACACATGCGCCTAGAATGTGCTATTTATTACAAGGCCAGAGAAATGGGATTTA
MRI-H196    ATTGGAAACACATGCGCCTAGAATGTGCTATTTATTACAAGGCCAGAGAAATGGGATTTA
SiHa        ATTGGAAACACATGCGCCTAGAATGTGCTATTTATTACAAGGCCAGAGAAATGGGATTTA
CaSki       ATTGGAAACAC----------------CTATTTATTACAAGGCCAGAGAAATGGGATTTA
CIN2/3-0001 ---------------------------CTATTTATTACAAGGCCAGAGAAATGGGATTTA
CIN2/3-0002 ATTGGAAACACATGCGCCTAGAATGTGCTATTTATTACAAGGCCAGAGAAATGGGATTTA
CIN2/3-0005 ATTGGAAACAAATGCGCCTAGAATGTGCTATTTATTACAAGGCCAGAGAAATGGGATTTA
CIN2/3-1503 ATTGGAAACACA---------------CTATTTATTACAAGGCCAGAGAAATGGGATTTA
CIN2/3-1511 ATTGGAAACAAATGCGCCTAGAATGTGCTATTTATTACAAGGCCAGAGAAATGGGATTTA
CIN2/3-1801 ATTGGAAACACATGCGCC---------CTATTTATTACAAGGCCAGAGAAATGGGATTTA
CIN2/3-2219 ATTGGAAACAAATGCGCCTAGAATGTGCTATTTATTACAAGGCCAGAGAAATGGGATTTA
CIN2/3-2227 ATTGGAAACACATGCGCCTAGAATGTGCTATTTATTACAAGGCCAGAGAAATGGGATTTA
CIN2/3-2229 ATTGGAAACACATGCGCCTAGAATGTGCTATTTATTACAAGGCCAGAGAAATGGGATTTA
CIN2/3-2237 ---------------------------CTATTTATTACAAGGCCAGAGAAATGGGATTTA
CIN2/3-3009 ATTGGAAACAAATGCGCCTAGAATGTGCTATTTATTACAAGGCCAGAGAAATGGGATTTA
CIN2/3-3035 ATTGGAAACAAATGCGCCTAGAATGTGCTATTTATTACAAGGCCAGAGAAATGGGATTTA
CIN2/3-4242 ATTGGAAACAAATGCGCCTAGAATGTGCTATTTATTACAAGGCCAGAGAAATGGGATTTA
  CA-07C381 ------------------------------------------------------------
 HSIL-75857 AT-------------------------CTATTTATTACAAGGCCAGAGAAATGGGATTTA
CIN2/3-0004 ATTGG----------------------CTATTTATTACAAGGCCAGAGAAATGGGATTTA
CIN2/3-4238a ATTGGAA--------------------CTATTTATTACAAGGCCAGAGAAATGGGATTTA
  CA-07C368 ATTGGAAACA----------------CTATTTATTACAAGGCCAGAGAAATGGGATTTA
 HSIL-66019 ------------------------------------------------------------
 HSIL-61979 ------------------------------------------------------------
 LSIL-75022 ------------------------------------------------------------
```


2969
HPV16R AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
MRI-H186 AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
MRI-H196 AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
SiHa AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
CaSki AACATATTAACCACCAGGTGGTGCCAACGCTGGCTGTATCAAAGAATAAAGCATTACAAG
CIN2/3-0001 AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
CIN2/3-0002 AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
CIN2/3-0005 AACA----------CCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
CIN2/3-1503 AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
CIN2/3-1511 AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
CIN2/3-1801 AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
CIN2/3-2219 AACATATTAACCACCAGGTGGTG----------AATAAAGCATTACAAG
CIN2/3-2227 AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
CIN2/3-2229 AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
CIN2/3-2237 AACATATTAAC----------AATAAAGCATTACAAG
CIN2/3-3009 AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
CIN2/3-3035 AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
CIN2/3-4242 AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
CA-07C381 ----------
HSIL-75857 AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTAT-----AATAAAGCATTACAAG
CIN2/3-0004 AACATATTAACCACCAGGTGGTGCCAACAC----------AATAAAGCATTACAAG
CIN2/3-4238a AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
CA-07C368 AACATATTAACCACCAGGTGGTGCCAACACTGGCTGTATCAAAGAATAAAGCATTACAAG
HSIL-66019 ----------
HSIL-61979 ----------
LSIL-75022 ----------
3029
HPV16R CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAAGT
MRI-H186 CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAAGT
MRI-H196 CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAAGT
SiHa CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAAGT
CaSki CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAAGT
CIN2/3-0001 CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAA--
CIN2/3-0002 CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAA--
CIN2/3-0005 CAATTGAACTGCAACTAACGTTAGAAACAATATATAAC----------
CIN2/3-1503 CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAAGT
CIN2/3-1511 CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAA----------
CIN2/3-1801 CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAAGT
CIN2/3-2219 CAATTGAACTGCAACTAACGTTAGAA----------
CIN2/3-2227 CAATTGAACTGCAACTAACGTTAGAAACAATATATAAC----------
CIN2/3-2229 CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAAGT
CIN2/3-2237 CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAAGT
CIN2/3-3009 CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAAGT
CIN2/3-3035 CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAAGT
CIN2/3-4242 CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAAGT
CA-07C381 ----------
HSIL-75857 CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAAGT
CIN2/3-0004 CAATTGAACTGCAACTAACGTTAGAAACAA----------
CIN2/3-4238a CAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAA----------
CA-07C368 CAATTGAACTGCAACTAACGTTA----------
HSIL-66019 ----------
HSIL-61979 ----------
LSIL-75022 ----------
3089
HPV16R GGACATTACAAGACGTTAGCCTTGAAGTGTATTTAACTGCACCAACAGGATGTATAAAAA
MRI-H186 GGACA----------TATTTAACTGCACCAACAGGATGTATAAAAA
MRI-H196 GGACATTACAAGACGTTAGCCTTGAAGTGTATTTAACTGCACCAACAGGATGTATAAAAA
SiHa GGACATTACAAGACGTTAGCCTTGAAGTGTATTTAACTACACCAACAGGATGTATAAAAA
CaSki GGACATTACAA----------TATTTAACTGCACCAACAGGATGTATAAAAA
CIN2/3-0001 ----------TATTTAACTGCACCAACAGGATGTATAAAAA
CIN2/3-0002 ----------TATTTAACTGCACCAACAGGATGTATAAAAA
CIN2/3-0005 ----------TACTTAACTGCACCAACAGGATGTATAAAAA
CIN2/3-1503 GGACATTACAA----------TATTTAACTGCACCAACAGGATGTATAAAAA
CIN2/3-1511 ----------TACTTAACTGCACCAACAGGATGTATAAAAA
CIN2/3-1801 GGACA----------TATTTAACTGCACCAACAGGATGTATAAAAA
CIN2/3-2219 ----------TATTTAACTGCACCAACAGGATGTATAAAAA
CIN2/3-2227 ----------TATTTAACTGCACCAACAGGATGTATAAAAA
CIN2/3-2229 GGACATTACAAGACGTTAGCCTTGAAGTGTATTTAACTGCACCAACAGGATGTATAAAAA
CIN2/3-2237 GGACATTACAAGATGTTAGCCTTGAAGTGTATTTAACTGCACCAACAGGATGTATAAAAA
CIN2/3-3009 GGACATTACAAGACGTTAGCCTTGAAGTGTATTTAACTGCACCAACAGGATGTATAAAAA
CIN2/3-3035 GGACATTACAAGACGTTAGCCTTGAAGTGTATTTAACTGCACCAACAGGATGTATAAAAA
CIN2/3-4242 GGACATTACAAGACGTTAGCCTTGAAGTGTATTTAACTGCACCAACAGGATGTATAAAAA
CA-07C381 ----------
HSIL-75857 GGACA----------TATTTAACTGCACCAACAGGATGTATAAAAA
CIN2/3-0004 ----------TATTTAACTGCACCAACAGGATGTATAAAAA
CIN2/3-4238a ----------TATTTAACTGCACCAACAGGATGTATAAAAA
CA-07C368 ----------TATTTAACTGCACCAACAGGATGTAT-----
HSIL-66019 ----------
HSIL-61979 ----------
LSIL-75022 ----------
3149
HPV16R AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAATGCATTATA
MRI-H186 AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAATGCATTATA
MRI-H196 AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAATGCATTATA
SiHa AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGC----------
CaSki AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAATGCATTATA
CIN2/3-0001 AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAATGCATTATA
CIN2/3-0002 AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAATGCATTATA
CIN2/3-0005 AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAATGCATTATA
CIN2/3-1503 AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAATGCATTATA
CIN2/3-1511 AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAATGCATTATA
CIN2/3-1801 AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAA---ATTATA
CIN2/3-2219 AACATG----------ATTATA
CIN2/3-2227 AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAATGCATTATA
CIN2/3-2229 AACATGGATATACAGTGGAAGTGCAGTTCGATGGAGACATATGCAATACAATGCATTATA
CIN2/3-2237 AA----------ATTATA
CIN2/3-3009 AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAATGCATTATA
CIN2/3-3035 AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAATGCATTATA
CIN2/3-4242 AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAATGCATTATA
CA-07C381 ----------
HSIL-75857 AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAA---ATTATA
CIN2/3-0004 AACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATATGCAATACAATGCATTATA
CIN2/3-4238a AACATGGATATACAGTGGAAGTGCAGTTCGATGGAGACATATGCAA--------ATTATA
CA-07C368 ----------ATTATA
HSIL-66019 ----------
HSIL-61979 ----------
LSIL-75022 ----------

```
                        .         .         .         .         .        .3209
      HPV16R    CAAACTGGACACATATATATATTTGTGAAGAAGCATCAGTAACTGTGGTAGAGGGTCAAG
      MRI-H186  CAAACTGGACACATATATATATTTGTGAAGAAGCATCAGTAACTGTGGTAGAGGGACAAG
      MRI-H196  CAAACTGGACACATATATATATTTGTGAAGAAGCATCAGTAACTGTGGTAGAGGGTCAAG
          SiHa  ------------------------------------------------------------
         CaSki  CAAACTGGACACATATATATATTTGTGAAGAAGCATCAGTAACTGTGGTAGAGGGTCAAG
    CIN2/3-0001 CAAACTGGACACATATATATATTTGTGAAGAAGCATCAGTAACTGTGGTAGAGGGTCAAG
    CIN2/3-0002 CAAACTGGACACATATATATATTTGTGAAGAAGCATCAGTAACTGTGGTAGAGGGTCAAG
    CIN2/3-0005 CAAACTGGAAATATATATATATTTGTGAAGAAACATTAGTAACTGTGGTAGAGGGTCAAG
    CIN2/3-1503 CAAACTGGACACATATATATATTTGTGAAGAAGCATCAGTAACTGTGGTAGAGGGTCAAG
    CIN2/3-1511 CAAACTGGAAATATATATATATTTGTGAAGAAACATTAGTAACTGTGGTAGAGGGTCAAG
    CIN2/3-1801 CAAACTGGACACATATATATATTTGTGAAGAAGCATCAGTAACTGTGGTAGAGGGTCAAG
    CIN2/3-2219 CAAACTGGAAATATATATATATTTGTGAA----------------------------
    CIN2/3-2227 CAAACTGGACACATATATATATTTGTGAAGAAGCATCAGTAACTGTGGTAGAGGGTCAAG
    CIN2/3-2229 CAAACTGGACACATATATATATTTGTGAAGAAGCATCAGTAACTGTGGTAGAGGGTCAAG
    CIN2/3-2237 CAAACTGGAAATATATATATATTTGTGAAGAA----------------------------
    CIN2/3-3009 CAAACTGGAGATATATATATATTTGTGAAGAAACATCAGTAACTGTGGTAGAGGGTCAAG
    CIN2/3-3035 CAAACTGGAGATATATATATATTTGTGAAGAAACATCAGTAACTGTGGTAGAGGGTCAAG
    CIN2/3-4242 CAAACTGGAGATATATATATATTTGTGAAGAAACATCAGTAACTGTGGTAGAGGGTCAAG
      CA-07C381 ------------------------------------------------------------
     HSIL-75857 CAAACTGGACACATATATATATTTGTGAAGAAGCATCAGTAACTGTGGTAGAGGGTCAAG
    CIN2/3-0004 CAAACTGGACACATATATATATTTGTGAAGAAGCATCAGTAACTGTGGTAGAGGGTCAAG
   CIN2/3-4238a CAAACTGGACACATATATATATTTGTGAAGAAGCATCAGTAACTGTGGTAGAGGGTCAGG
      CA-07C368 CAAACTGGACACATATATAT--TTGTGAAGAAGCATCAGTAACTGTGGTAGAGGGTCAAG
     HSIL-66019 ------------------------------------------------------------
     HSIL-61979 ------------------------------------------------------------
     LSIL-75022 ------------------------------------------------------------

                        .         .         .         .         .        .3269
      HPV16R    TTGACTATTATGGTTTATATTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTA
      MRI-H186  TTGACTATTATGGTTTATATTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTA
      MRI-H196  TTGACTATTATGGTTTATATTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTA
          SiHa  ------------------------------------------------------------
         CaSki  TTGACTATTATGGTTTATATTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTA
    CIN2/3-0001 TTGACTATTATGGTTTATATTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTA
    CIN2/3-0002 TTGACTATTATGGTTTATATTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTA
    CIN2/3-0005 TTGACTATTATGGTTTATATTATGTTCATGAAGGAATACAAACATATTTTGTGCAGTTTA
    CIN2/3-1503 TTGACTATTATGGTTTATATTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTA
    CIN2/3-1511 TTGACTATTATGGTTTATATTATGTTCATGAAGGAATACAAACATATTTTGTGCAGTTTA
    CIN2/3-1801 TTGACTATTATGGTTTATATTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTA
    CIN2/3-2219 -------------TTTATATTATGTCCATGAAGGAATACAAACATATTTTGTGCAGTTTA
    CIN2/3-2227 TTGACTATTATGGTTTATATTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTA
    CIN2/3-2229 TTGACTATTATGGTTTATATTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTA
    CIN2/3-2237 -------------TTTATATTATGTTCATGAAGGAATACAAACATATTTTGTGCAGTTTA
    CIN2/3-3009 TTGACTATTATGGTATATATTATGTTCATGAAGGAATACAAACATATTTTGTGCAGTTTA
    CIN2/3-3035 TTGACTATTATGGTATATATTATGTTCATGAAGGAATACAAACATATTTTGTGCAGTTTA
    CIN2/3-4242 TTGACTATTATGGTATATATTATGTTCATGAAGGAATACAAACATATTTTGTGCAGTTTA
      CA-07C381 ------------------------------------------------------------
     HSIL-75857 TTGACTA------TTTATATTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTA
    CIN2/3-0004 TTGACTATTATGGTTTATATTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTA
   CIN2/3-4238a TTGACTATTATGGTTTATACTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTA
      CA-07C368 TTGACTACTATGGTTTATATTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTA
     HSIL-66019 ------------------------------------------------------------
     HSIL-61979 ------------------------------------------------------------
     LSIL-75022 ------------------------------------------------------------
```

```
                        .         .         .         .         .        .3329
      HPV16R    AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
      MRI-H186  AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
      MRI-H196  AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
          SiHa  ------------------------------------------------------------
         CaSki  AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
    CIN2/3-0001 AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
    CIN2/3-0002 AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
    CIN2/3-0005 AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
    CIN2/3-1503 AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
    CIN2/3-1511 AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
    CIN2/3-1801 AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
    CIN2/3-2219 AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
    CIN2/3-2227 AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
    CIN2/3-2229 AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
    CIN2/3-2237 AAGATGATGCAGAAAAGTATAGTAAAAATAAAGTATGGGAAGTTCATACGGGTGGTCAGG
    CIN2/3-3009 AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
    CIN2/3-3035 AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
    CIN2/3-4242 AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
      CA-07C381 ------------------------------------------------------------
     HSIL-75857 AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
    CIN2/3-0004 AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
   CIN2/3-4238a AAGATGATGCAGAAAAATATAG----------------------------------
      CA-07C368 AAGATGATGCAGAAAAATATAGTAAAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGG
     HSIL-66019 ------------------------------------------------------------
     HSIL-61979 ------------------------------------------------------------
     LSIL-75022 ------------------------------------------------------------

                        .         .         .         .         .        .3389
      HPV16R    TAATATTATGTCCTACATCTGTGTTTAGCAGCAACGAAGTATCCTCTCCTGAAATTATTA
      MRI-H186  TAATATTATGTCCTACATCTGTGTTTAGCAGCAACGAAGTATCCTCTCCTGAAATTATTA
      MRI-H196  TAATATTATGTCCTACATCTGTGTTTAGCAGCAACGAAGTATCCTCTCCTGAAATTATTA
          SiHa  ------------------------------------------------------------
         CaSki  TAATATTATGTCCTACATCTGTGTTTAGCAGCAACGAAGTATCCTCTCCTGAAACTATTA
    CIN2/3-0001 TAATATTATGTCCTACATCTGTGTTTAGCAGCAACGAAGTATCCTCTCCTGAAATTATTA
    CIN2/3-0002 TAATATTATGTCCTACATCTGTGTTTAGCAGCAACGAAGTATCCTCTCCTGAAATTATTA
    CIN2/3-0005 TAATATTATGTCCTACATCTGTGTTTAGCAGCGACGAAGTATCCTCTGCTGAAATTATTA
    CIN2/3-1503 TAATATTATGTCCTACATCTGTGTTTAGCAGCAACGAAGTATCCTCTCCTGAAATTATTA
    CIN2/3-1511 TAATATTATGTCCTACATCTGTGTTTAGCAGCGACGAAGTATCCTCTGCTGAAATTATTA
    CIN2/3-1801 TAATATTATGTCCTACATCTGTGTTTAGCAGCAACAAAGTATCCTCTCCTGAAATTATTA
    CIN2/3-2219 TAATATTATGTCCTACATCTGTGTTTAGCAGCGACGAAGTATCCTCTGCTGAAATTATTA
    CIN2/3-2227 TAATATTATGTCCTACATCTGTGTTTAGCAGCAACGAAGTATCCTCTCCTGAAATTATTA
    CIN2/3-2229 TAATATTATGTCCTACATCTGTGTTTAGCAGCAACGAAGTATCCTCTCCTGAAACTATTA
    CIN2/3-2237 TAATATTATGTCCTACATCTGTGTTTAGCAGCGACGAAGTATCCTCTGCTGAAATTATTA
    CIN2/3-3009 TAATATTATGTCCTACATCTGTGTTTAGCAGCGACGAAGTATCCTCTGCTGAAATTATTA
    CIN2/3-3035 TAATATTATGTCCTACATCTGTGTTTAGCAGCGACGAAGTATCCTCTGCTGAAATTATTA
    CIN2/3-4242 TAATATTATGTCCTACATCTGTGTTTAGCAGCGACGAAGTATCCTCTGCTGAAATTATTA
      CA-07C381 ------------------------------------------------------------
     HSIL-75857 TAATATTATGTCCTACATCTGTGTTTAGCAGCAACGAAGTATCCTCTCCTGAAATTATTA
    CIN2/3-0004 TAATATTATGTCCTACATCTGTGTTTAGCAGCAACGAAGTATCCTCTCCTGAAATTATTA
   CIN2/3-4238a ----------------------------AACGAAGTATCCTCTCCTGAAACTATTA
      CA-07C368 TAATATTATGTCCTACATCTGTGTTTAGCAGCAACGAAGTATCCTCTCCTGAAATTATTA
     HSIL-66019 ------------------------------------------------------------
     HSIL-61979 ------------------------------------------------------------
     LSIL-75022 ------------------------------------------------------------
```

```
                .         .         .         .         .        .3449
HPV16R      GGCAGCACTTGGCCAACCACCCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCG
MRI-H186    GGCAGCACTTGGCCAACCACCCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCG
MRI-H196    GGCAGCACTTGGCCAACCACTCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCG
SiHa        ------------------------------------------------------------
CaSki       GGCAGCACTTGGCCAACCACTCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCG
CIN2/3-0001 GGCAGCACTTGGCCAACCACTCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCG
CIN2/3-0002 GGCAGCACTTGGCCAACCACTCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCG
CIN2/3-0005 GGCAGCACTTGGCCAACCACTCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCA
CIN2/3-1503 GGCAGCACTTGGCCAACCACTCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCG
CIN2/3-1511 GGCAGCACTTGGCCAACCACTCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCA
CIN2/3-1801 GGCAGCACTTGGCCAACCACTCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCG
CIN2/3-2219 GGCAGCACTTGGCCAACCACTCCGCCGCGACCCATTCCAAAGCCGTCGCCTTGGGCACCA
CIN2/3-2227 GGCAGCACTTGGCCAACCACTCCGCCGCGACCCATACCCAAGCCGTCGCCTTGGGCACCG
CIN2/3-2229 GGCAGCACTTGGCCAACCACTCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCG
CIN2/3-2237 GGCAGCACTTGGCCAACCACTCCGCCGCGACCCATACCAAAACCGTCGCCTTGGGCACCA
CIN2/3-3009 GGCAGCACTTGGCCAACCACCCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCA
CIN2/3-3035 GGCAGCACTTGGCCAACCACCCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCA
CIN2/3-4242 GGCAGCACTTGGCCAACCACCCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCA
CA-07C381   ------------------------------------------------------------
HSIL-75857  GGCAGCACTTGGCCAACCACTCCGCCGCGACCTATACCAAAGCCGTCGCCTTGGGCACCG
CIN2/3-0004 GGCAGCACTTGGCCAACCACCCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCG
CIN2/3-4238a GGCAGCACTTGGCCAACCACTCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCG
CA-07C368   GGCAGCACTTGGCCAACCACCCCGCCGCGACCCATACCAAAGCCGTCGCCTTGGGCACCG
HSIL-66019  ------------------------------------------------------------
HSIL-61979  ------------------------------------------------------------
LSIL-75022  ------------------------------------------------------------

                .         .         .         .         .        .3509
HPV16R      AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
MRI-H186    AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
MRI-H196    AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
SiHa        ------------------------------------------------------------
CaSki       AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CIN2/3-0001 AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CIN2/3-0002 AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CIN2/3-0005 AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CIN2/3-1503 AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CIN2/3-1511 AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CIN2/3-1801 AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CIN2/3-2219 AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CIN2/3-2227 AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CIN2/3-2229 AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CIN2/3-2237 AAGAAACACAGACGACTATCCTGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CIN2/3-3009 AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CIN2/3-3035 AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CIN2/3-4242 AAGAAACACAGACGAC----------CAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CA-07C381   ------------------------------------------------------------
HSIL-75857  AAGAAACA------------------CAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CIN2/3-0004 AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CIN2/3-4238a AAGAAACACAGACGAC----------CAAGATCAGAGCCAGACACCGGAAACCCCTGCC
CA-07C368   AAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCC
HSIL-66019  ------------------------------------------------------------
HSIL-61979  ------------------------------------------------------------
LSIL-75022  -------------------------CAAGATCAGAGCCAAACACCGGAAACCCCTGCC
```

```
                .         .         .         .         .        .3569
HPV16R      ACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCATTTA
MRI-H186    ACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCATTTA
MRI-H196    ACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCATTTA
SiHa        ------------------------------------------------------------
CaSki       ACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCATTTA
CIN2/3-0001 ACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCATTTA
CIN2/3-0002 ACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCATTTA
CIN2/3-0005 ACACCAATAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCAGTTA
CIN2/3-1503 ACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCATTTA
CIN2/3-1511 ACACCAATAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCA----
CIN2/3-1801 ACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAA----------------
CIN2/3-2219 ACACCAATAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCAGTTA
CIN2/3-2227 ACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCATTTA
CIN2/3-2229 ACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCATTTA
CIN2/3-2237 ACACCAACAAGTTGTTGCACAGAGACTCCGTGGACAGTGCTCCAATCCTCACTGCAGTTA
CIN2/3-3009 ACACCAACAAGTTGTTGCACAGAGACTCCGTGGACAGTGCTCCAATCCTCACTGCA----
CIN2/3-3035 ACACCAACAAGTTGTTGCACAGAGACTCCGTGGACAGTGCTCCAATCCTCACTGCA----
CIN2/3-4242 ACACCAACAAGTTGTTGCACAGAGACTCCGTGGACAGTGCTCCAATCCTCACTGCAGTTA
CA-07C381   ------------------------------------------------------------
HSIL-75857  ACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCATTTA
CIN2/3-0004 ACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCATTTA
CIN2/3-4238a ACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCATTTA
CA-07C368   ACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCATTTA
HSIL-66019  ------------------------------------------------------------
HSIL-61979  ------------------------------------------------------------
LSIL-75022  ACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGCTCCAATCCTCACTGCATTTA

                .         .         .         .         .        .3629
HPV16R      ACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
MRI-H186    ACAGC--ACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
MRI-H196    ACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAC
SiHa        -------ACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CaSki       ACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CIN2/3-0001 ACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CIN2/3-0002 -------ACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CIN2/3-0005 ACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CIN2/3-1503 ACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CIN2/3-1511 -------ACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CIN2/3-1801 -------ACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CIN2/3-2219 ACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CIN2/3-2227 ACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CIN2/3-2229 ACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CIN2/3-2237 ACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCA------------
CIN2/3-3009 -------ACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CIN2/3-3035 -------ACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACA-----
CIN2/3-4242 ACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CA-07C381   ------------------------------------------------------------
HSIL-75857  ACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CIN2/3-0004 ACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CIN2/3-4238a ACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
CA-07C368   ACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAA
HSIL-66019  ------------------------------------------------------------
HSIL-61979  ------------------------------------------------------------
LSIL-75022  ACAGC-----------------------------------------------------------
```

```
                    .         .         .         .         .        .3689
      HPV16R  AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAAGCATTGTACATTGT
     MRI-H186 AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAAGCATTGTACATTGT
     MRI-H196 AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAAGCATTGTACATTGT
        SiHa  AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAAGCATTGTAAATTGT
       CaSki  AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAAGCATTGTAAATTGT
   CIN2/3-0001 AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAAGCATTGTACATTGT
   CIN2/3-0002 AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAA--------------
   CIN2/3-0005 AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAAGCATTGTAAATTGT
   CIN2/3-1503 AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAAGCATTG--------
   CIN2/3-1511 AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAAGCATTGTAAATTGT
   CIN2/3-1801 AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAAGCATTGTACATTGT
   CIN2/3-2219 AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAA--------------
   CIN2/3-2227 AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAAGCATTGTACATTGT
   CIN2/3-2229 AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAA--------------
   CIN2/3-2237 ------------------------------------------------------------
   CIN2/3-3009 AAGGTGATGCTAATACTTTAAAATGTTTAAGATACAGATTTAAAAAGCATTGTAAATTGT
   CIN2/3-3035 ------------------------------------------------------------
   CIN2/3-4242 AAGGTGATGCTAATACTTTAAAATGTTTAAGATACAGATTTAAAAAGCATTGTAAATTGT
     CA-07C381 ------------------------------------------------------------
    HSIL-75857 AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAA--------------
   CIN2/3-0004 AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAA--------------
  CIN2/3-4238a AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAA----------------
     CA-07C368 AAGGTGATGCTAATACTTTAAAATGTTTAAGATATAGATTTAAAAA--------------
    HSIL-66019 ------------------------------------------------------------
    HSIL-61979 ------------------------------------------------------------
    LSIL-75022 ------------------------------------------------------------

                    .         .         .         .         .        .3749
      HPV16R  ATACTGCAGTGTCGTCTACATGGCATTGGACAGGACATAATGTAAAACATAAAAGTGCAA
     MRI-H186 ATACTGCAGTGTCGTCTACATGGCATTGGACAGGACATAATGTAAAACATAAAAGTGCAA
     MRI-H196 ATACTGCAGTGTCGTCTACATGGCATTGGACAGGACATAATGTAAAACATAAAAGTGCAA
        SiHa  ATACTGCAGTGTCGTCTACATGGCATTGGACAGGACATAATGTAAAACATAAAAGTGCAA
       CaSki  ATACTGCAGTGTC----------------GACAGGACATAATGTAAAACATAAAAGTGCAA
   CIN2/3-0001 ATACTGCAGTGTCGTCTACATGGCATTGGACAGGACATAATGTAAAACATAAAAGTGCAA
   CIN2/3-0002 ----------------------------GACAGGACATAATGTAAAACATAAAAGTGCAA
   CIN2/3-0005 ATACAGCAGTGTCGTCTACATGGCATTGGACAGGACATAATGTAAAACATAAAAGTGCAA
   CIN2/3-1503 ----------------------------GACAGGACATAATGTAAAACATAAAAGTGCAA
   CIN2/3-1511 ATACAGCAGTGTCGTCTACATGGCATTGGACAGGACATAATGTAAAACATAAAAGTGCAA
   CIN2/3-1801 ATACTGCAGTGTCGTCTACATGGCATTGGACAGGACATAATGTAAAACATAAAAGTGCAA
   CIN2/3-2219 ----------------------------GACAGGACATAATGTAAAACATAAAAGTGCAA
   CIN2/3-2227 ATAC------------------------GACAGGACATAATGTAAAACATAAAAGTGCAA
   CIN2/3-2229 ----------------------------GACAGGACATAATGTAAAACATAAAAGTGCAA
   CIN2/3-2237 ----------------------------GACAGGACATAATGTAAAACATAAAAGTGCAA
   CIN2/3-3009 ATACAGCAGTGTCGTC------------GACAGGACATAATGTAAAACATAAAAGTGCAA
   CIN2/3-3035 ----------------------------GACAGGACATAATGTAAAACATAAAAGTGCAA
   CIN2/3-4242 ATACAGCAGTGTCGTC------------GACAGGACATAATGTAAAACATAAAAGTGCAA
     CA-07C381 ------------------------------------------------------------
    HSIL-75857 ----------------------------GACAGGACATAATGTAAAACATAAAAGTGCAA
   CIN2/3-0004 ----------------------------GACAGGACATAATGTAAAACATAAAAGTGCAA
  CIN2/3-4238a ----------------------------GACAGGACATAATGTAAAACATAAAAGTGCAA
     CA-07C368 ----------------------------GACAGGACATAATGTAAAACATAAAAGTGCAA
    HSIL-66019 ------------------------------------------------------------
    HSIL-61979 ------------------------------------------------------------
    LSIL-75022 ------------------------------------------------------------
```

```
                    .         .         .         .         .        .3809
      HPV16R  TTGTTACACTTACATATGATAGTGAATGGCAACGTGACCAATTTTTGTCTCAAGTTAAAA
     MRI-H186 TTGTTACACTTACATATGATAGTGAATGGCAACGTGACCAATTTTTGTCTCAAGTTAAAA
     MRI-H196 TTGTTACACTTACATATGATAGTGAATGGCAACGTGACCAATTTTTGTCTCAAGTTAAAA
        SiHa  TTGTTACACTTACATATGATAGTGAATGGCAACGTGACCAATTTTTGTCTCAAGTTAAAA
       CaSki  TTGTTACACTTACATATGATAGTGAATGGCAACGTGACCAATTTTTGTCTCAAGTTAAAA
   CIN2/3-0001 TTGTTACACTTACATATGATAGTGAATGGCAACGTGACCAATTTTTGTCTCAAGTTAAAA
   CIN2/3-0002 TTGTTACACTTACATATGATAGTGAATGGCAACGTGACCAATTTTTGTCTCAAGTTAAAA
   CIN2/3-0005 TTGTTACACTTACATATGATAGTGAATGTCAACGTGAACAATTTTTGTCTCAAGTCAAAA
   CIN2/3-1503 TTGTTACACTTACATATGATAGTGAATGGCAACGTGACCAATTTTTGTCTCAAGTTAAAA
   CIN2/3-1511 TTGTTACACTTACATATGATAGTGAATGTCAACGTGAACAATTTTTGTCTCAAGTCAAAA
   CIN2/3-1801 TTGTTACACTTACATATGATAGTGAATGGCAACGTGACCAATTTTTGTCTCAAGTTAAAA
   CIN2/3-2219 TTGTTACACTTACATATGATAGTGAATGTCAACGGGAACAATTTTTGTCTCAAGTTAAAA
   CIN2/3-2227 TTGTTACACTTACATATGATAGTGAATGGCAACGTGACCAATTTTTGTCTCAAGTTAAAA
   CIN2/3-2229 TTGTTACACTTACATATGATAGTGAATGGCAACGTGACCAATTTTTGTCTCAAGTTAAAA
   CIN2/3-2237 TTGTTACACTTACATATGATAGTGAATGTCAACGTGAACAATTTTTGTCTCAAGTGAAAA
   CIN2/3-3009 TTGTTACACTTACATATGATAGTGAATGTCAACGTGAACAATTTTTGTCTCAAGTGAAAA
   CIN2/3-3035 TTGTTACACTTACATATGATAGTGAATGTCAACGTGAACAATTTTT--CTCAAGTGAAAA
   CIN2/3-4242 TTGTTACACTTACATATGATAGTGAATGTCAACGTGAACAATTTTTGTCTCAAGTGAAAA
     CA-07C381 ------------------------------------------------------------
    HSIL-75857 TTGTTACACTTACATATGATAGTGAATGGCAACGTGACCATTTTTTGTCTCAAGTTAAAA
   CIN2/3-0004 TTGTTACACTTACATATGATAGTGAATGGCAACGTGACCAATTTTTGTCTCAAGTTAAAA
  CIN2/3-4238a TTGTTACACTTACATATGATAGTGAATGGCAACGTGACCAATTTTTGTCTCAAGTTAAAA
     CA-07C368 TTGTTACACTTACATATGATAGTGAATGGCAACGTGACCAATTTTTGTCTCAAGTTAAAA
    HSIL-66019 ------------------------------------------------------------
    HSIL-61979 ------------------------------------------------------------
    LSIL-75022 ------------------------------------------------------------

                    .         .         .         .         .    3853
      HPV16R  TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
     MRI-H186 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
     MRI-H196 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
        SiHa  TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
       CaSki  TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
   CIN2/3-0001 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
   CIN2/3-0002 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
   CIN2/3-0005 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTGTATATGA
   CIN2/3-1503 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
   CIN2/3-1511 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTGTATATGA
   CIN2/3-1801 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
   CIN2/3-2219 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATA---
   CIN2/3-2227 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
   CIN2/3-2229 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
   CIN2/3-2237 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
   CIN2/3-3009 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
   CIN2/3-3035 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
   CIN2/3-4242 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
     CA-07C381 ------------------------------------------
    HSIL-75857 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
   CIN2/3-0004 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
  CIN2/3-4238a TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
     CA-07C368 TACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGA
    HSIL-66019 ------------------------------------------
    HSIL-61979 ------------------------------------------
    LSIL-75022 ------------------------------------------
```

A6. Nucleotide sequences of HPV16 ORF E6 of 25 DNA samples in ASP16 experiments

See Results section 2.2.2.9. The complete HPV16 ORF E6 nucleotide sequences of the 25 DNA samples (see Results, Table 2.13 and Table 2.18), covering pos. 83-559 are shown as an alignment.

```
                .         .         .         .         .         . 142
      HPV16R    ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
  CIN2/3-0004   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
    CA-07C368   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
     MRI-H186   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
  CIN2/3-2229   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
 CIN2/3-4238a   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
    HSIL-66019   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
    HSIL-61979   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
  CIN2/3-0001   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
  CIN2/3-0002   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
  CIN2/3-1503   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
  CIN2/3-1801   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
  CIN2/3-2227   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
    HSIL-75857   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
     MRI-H196   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
         SiHa   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
        CaSki   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCGGAAAGTTACCA
    CA-07C381   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
    LSIL-75022   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
  CIN2/3-3009   ATGCACCAAATGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
  CIN2/3-3035   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
  CIN2/3-4242   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCAGAAAGTTACCA
  CIN2/3-0005   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCGGAAAGTTACCA
  CIN2/3-1511   ATGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCGGAAAGTTACCA
  CIN2/3-2219   CTGCACCAAAAGAGAACTGCAATGTTTCAGGACCCACAGGAGCGACCCACAAAGTTACCA
  CIN2/3-2237   ATGCACCAAAAGAGAACTGCAATGTTCCAGGACCCACAGGAGCGACCCATAAAGTTACCA

                .         .         .         .         .         . 202
      HPV16R    CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
  CIN2/3-0004   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
    CA-07C368   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
     MRI-H186   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
  CIN2/3-2229   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
 CIN2/3-4238a   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
    HSIL-66019   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
    HSIL-61979   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
  CIN2/3-0001   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
  CIN2/3-0002   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
  CIN2/3-1503   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
  CIN2/3-1801   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
  CIN2/3-2227   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
    HSIL-75857   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
     MRI-H196   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
         SiHa   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
        CaSki   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
    CA-07C381   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
    LSIL-75022   CAGTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
  CIN2/3-3009   CATTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
  CIN2/3-3035   CATTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
  CIN2/3-4242   CATTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
  CIN2/3-0005   GATTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
  CIN2/3-1511   GATTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
  CIN2/3-2219   GATTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
  CIN2/3-2237   GATTTATGCACAGAGCTGCAAACAACTATACATGATATAATATTAGAATGTGTGTACTGC
```

```
                .         .         .         .         .         . 262
      HPV16R    AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
  CIN2/3-0004   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
    CA-07C368   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
     MRI-H186   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
  CIN2/3-2229   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
 CIN2/3-4238a   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
    HSIL-66019   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
    HSIL-61979   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
  CIN2/3-0001   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
  CIN2/3-0002   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
  CIN2/3-1503   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
  CIN2/3-1801   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
  CIN2/3-2227   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
    HSIL-75857   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
     MRI-H196   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
         SiHa   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
        CaSki   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
    CA-07C381   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
    LSIL-75022   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
  CIN2/3-3009   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
  CIN2/3-3035   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
  CIN2/3-4242   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
  CIN2/3-0005   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
  CIN2/3-1511   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
  CIN2/3-2219   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA
  CIN2/3-2237   AAGCAACAGTTACTGCGACGTGAGGTATATGACTTTGCTTTTCGGGATTTATGCATAGTA

                .         .         .         .         .         . 322
      HPV16R    TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
  CIN2/3-0004   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
    CA-07C368   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
     MRI-H186   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
  CIN2/3-2229   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
 CIN2/3-4238a   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
    HSIL-66019   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
    HSIL-61979   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
  CIN2/3-0001   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
  CIN2/3-0002   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
  CIN2/3-1503   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
  CIN2/3-1801   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
  CIN2/3-2227   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
    HSIL-75857   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
     MRI-H196   TATAGAGATGGGAATCCATATGCTGTGTGTGATAAATGTTTAAAGTTTTATTCTAAAATT
         SiHa   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
        CaSki   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
    CA-07C381   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
    LSIL-75022   TATAGAGATGGGAATCCATATGCTGTATGTGATAAATGTTTAAAGTTTTATTCTAAAATT
  CIN2/3-3009   TATAGAGATGGGAATCCATATGCAGTGTGTGATAAATGTTTAAAGTTTTATTCTAAAATT
  CIN2/3-3035   TATAGAGATGGGAATCCATATGCAGTGTGTGATAAATGTTTAAAGTTTTATTCTAAAATT
  CIN2/3-4242   TATAGAGATGGGAATCCATATGCAGTGTGTGATAAATGTTTAAAGTTTTATTCTAAAATT
  CIN2/3-0005   TATAGAGATGGGAATCCATATGCAGTGTGTGAGAAATGTTTAAAGTTTTATTCTAAAATT
  CIN2/3-1511   TATAGAGATGGGAATCCATATGCAGTGTGTGAGAAATGTTTAAAGTTTTATTCTAAAATT
  CIN2/3-2219   TATAGAGATGGGAATCCATATGCAGTGTGTGATAAATGTTTAAAGTTTTATTCTAAAATT
  CIN2/3-2237   TATAGAGATGGGAATCCATATGCAGTGTGTGATAAATGTTTAAAGTTTTATTCTAAAATT
```

```
                 .         .         .         .         .         . 382
HPV16R     AGTGAGTATAGACATTATTGTTATAGTTTGTATGGAACAACATTAGAACAGCAATACAAC
CIN2/3-0004 AGTGAGTATAGACATTATTGTTATAGTTTGTATGGAACAACATTAGAACAGCAATACAAC
CA-07C368  AGTGAGTATAGACATTATTGTTATAGTTTGTATGGAACAACATTAGAACAGCAATACAAC
MRI-H186   AGTGAGTATAGACATTATTGTTATAGTTTGTATGGAACAACATTAGAACAGCAATACAAC
CIN2/3-2229 AGTGAGTATAGACATTATTGTTATAGTTTGTATGGAACAACATTAGAACAGCAATACAAC
CIN2/3-4238a AGTGAGTATAGACATTATTGTTATAGTTTGTATGGAACAACATTAGAACAGCAATACAAC
HSIL-66019 AGTGAGTATAGACATTATTGTTATAGTTTGTATGGAACAACATTAGAACAGCAATACAAC
HSIL-61979 AGTGAGTATAGACATTATTGTTATAGTTTGTATGGAACAACATTAGAACAGCAATACAAC
CIN2/3-0001 AGTGAGTATAGACATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
CIN2/3-0002 AGTGAGTATAGACATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
CIN2/3-1503 AGTGAGTATAGACATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
CIN2/3-1801 AGTGAGTATAGACATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
CIN2/3-2227 AGTGAGTATAGACATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
HSIL-75857 AGTGAGTATAGACATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
MRI-H196   AGTGAGTATAGACATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
SiHa       AGTGAGTATAGACATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
CaSki      AGTGAGTATAGACATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
CA-07C381  AGTGAGTATAGACATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
LSIL-75022 AGTGAGTATAGACATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
CIN2/3-3009 AGTGAGTATAGATATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
CIN2/3-3035 AGTGAGTATAGATATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
CIN2/3-4242 AGTGAGTATAGATATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
CIN2/3-0005 AGTGAGTATAGATATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
CIN2/3-1511 AGTGAGTATAGATATTATTGTTATAGTGTGTATGGAACAACATTAGAACAGCAATACAAC
CIN2/3-2219 AGTGAGTATAGATATTATTGTTATAGTTTGTATGGAACAACATTAGAACAGCAATACAAC
CIN2/3-2237 AGTGAGTATAGATATTATTGTTATAGTTTGTATGGAACAACATTAGAACAGCAATACAAC

                 .         .         .         .         .         . 442
HPV16R     AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CIN2/3-0004 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CA-07C368  AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
MRI-H186   AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CIN2/3-2229 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CIN2/3-4238a AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
HSIL-66019 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
HSIL-61979 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CIN2/3-0001 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CIN2/3-0002 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CIN2/3-1503 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CIN2/3-1801 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CIN2/3-2227 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
HSIL-75857 AAACTGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
MRI-H196   AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
SiHa       AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAC
CaSki      AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CA-07C381  AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
LSIL-75022 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CIN2/3-3009 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CIN2/3-3035 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CIN2/3-4242 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CIN2/3-0005 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CIN2/3-1511 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CIN2/3-2219 AAACCGTTGTGTGATTTGTTAATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
CIN2/3-2237 AAACCGTTGTGTGATTTGTTGATTAGGTGTATTAACTGTCAAAAGCCACTGTGTCCTGAA
```

```
                 .         .         .         .         .         . 502
HPV16R     GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CIN2/3-0004 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CA-07C368  GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
MRI-H186   GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CIN2/3-2229 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CIN2/3-4238a GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
HSIL-66019 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
HSIL-61979 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CIN2/3-0001 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CIN2/3-0002 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CIN2/3-1503 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CIN2/3-1801 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CIN2/3-2227 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
HSIL-75857 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
MRI-H196   GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
SiHa       GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CaSki      GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CA-07C381  GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
LSIL-75022 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CIN2/3-3009 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CIN2/3-3035 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CIN2/3-4242 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CIN2/3-0005 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CIN2/3-1511 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CIN2/3-2219 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC
CIN2/3-2237 GAAAAGCAAAGACATCTGGACAAAAAGCAAAGATTCCATAATATAAGGGGTCGGTGGACC

                 .         .         .         .         .         . 559
HPV16R     GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CIN2/3-0004 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CA-07C368  GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
MRI-H186   GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CIN2/3-2229 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CIN2/3-4238a GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
HSIL-66019 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
HSIL-61979 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CIN2/3-0001 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CIN2/3-0002 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CIN2/3-1503 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CIN2/3-1801 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CIN2/3-2227 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
HSIL-75857 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
MRI-H196   GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
SiHa       GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CaSki      GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CA-07C381  GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
LSIL-75022 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CIN2/3-3009 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CIN2/3-3035 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CIN2/3-4242 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CIN2/3-0005 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CIN2/3-1511 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CIN2/3-2219 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
CIN2/3-2237 GGTCGATGTATGTCTTGTTGCAGATCATCAAGAACACGTAGAGAAACCCAGCTGTAA
```

A7. Contents of the file required for ASP16 data analysis

See Results section 2.2.1.3. The contents of a text file, named "`hpv16R_EEprimer_151nt`", are shown below.

```
>HPV16R_E17_fm_810_to_960
AATGGGCACACTAGGAATTGTGTGCCCCATCTGTTCTCAGAAACCATAATCTACCATGGCTGATCCTGCAGGTACCAATGGGGAAGAGGGTACGG
GATGTAATGGATGGTTTTATGTAGAGGCTGTAGTGGAAAAAAAAACAGGGGATGCT
>HPV16R_E02_fm_1275_to_1425
TGAAGTGGAAACTCAGCAGATGTTACAGGTAGAAGGGCGCCATGAGACTGAAACACCATGTAGTCAGTATAGTGGTGGAAGTGGGGGTGGTTGCA
GTCAGTACAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGAAAGACACACTATATGC
>HPV16R_E03_fm_1576_to_1726
ATTGCTGCATTTGGACTTACACCCAGTATAGCTGACAGTATAAAAACACTATTACAACAATATTGTTTATATTTACACATTCAAAGTTTAGCATG
TTCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAAATGTGGAAAAAATAGAG
>HPV16R_E04_fm_1951_to_2101
TGGGCCTACGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATTGGCAGACACTAATAGTAATGCAAGTGCCTTTCTAAA
AAGTAATTCACAGGCAAAAATTGTAAAGGATTGTGCAACAATGTGTAGACATTATA
>HPV16R_E05_fm_2403_to_2553
AGCAGATGCCAAAATAGGTATGTTAGATGATGCTACAGTGCCCTGTTGGAACTACATAGATGACAATTTAAGAAATGCATTGGATGGAAATTTAG
TTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATTATTA
>HPV16R_E06_fm_2723_to_2873
TAAGTTTGCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTTAAATGTGTGTCAGGACAAAATACTAACACATTATGAAAA
TGATAGTACAGACCTACGTGACCATATAGACTATTGGAAACACATGCGCCTAGAAT
>HPV16R_E07_fm_3121_to_3271
TGGAGACATATGCAATACAATGCATTATACAAACTGGACACATATATATATTTGTGAAGAAGCATCAGTAACTGTGGTAGAGGGTCAAGTTGACT
ATTATGGTTTATATTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTAAA
>HPV16R_E08_fm_3555_to_3705
TCCTCACTGCATTTAACAGCTCACACAAAGGACGGATTAACTGTAATAGTAACACTACACCCATAGTACATTTAAAAGGTGATGCTAATACTTTA
AAATGTTTAAGATATAGATTTAAAAAGCATTGTACATTGTATACTGCAGTGTCGTC
>HPV16R_E09_fm_1064_to_1214
CACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGCAGTACAGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATT
AGTGGATGTGTAGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATAGAAAA
>HPV16R_E21_fm_1471_to_1621
GCAAAGGCAGCAATGTTAGCAAAATTTAAAGAGTTATACGGGGTGAGTTTTTCAGAATTAGTAAGACCATTTAAAAGTAATAAATCAACGTGTTG
CGATTGGTGTATTGCTGCATTTGGACTTACACCCAGTATAGCTGACAGTATAAAAA
>HPV16R_E11_fm_1785_to_1935
AGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTATATTGGTATAAAACAGGTATATCAAATATTAGTGAAGTGTATGGAGACACGCCAGAAT
GGATACAAAGACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTA
>HPV16R_E12_fm_2174_to_2324
GTGATTGGAAGCAAATTGTTATGTTTTTAAGGTATCAAGGTGTAGAGTTTATGTCATTTTTAACTGCATTAAAAAGATTTTTGCAAGGCATACCT
AAAAAAAATTGCATATTACTATATGGTGCAGCTAACACAGGTAAATCATTATTTGG
>HPV16R_E13_fm_2569_to_2719
AATGCTGGTACAGATTCTAGGTGGCCTTATTTACATAATAGATTGGTGGTGTTTACATTTCCTAATGAGTTTCCATTTGACGAAAACGGAAATCC
AGTGTATGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCA
>HPV16R_E14_fm_2933_to_3083
CCAACACTGGCTGTATCAAAGAATAAAGCATTACAAGCAATTGAACTGCAACTAACGTTAGAAACAATATATAACTCACAATATAGTAATGAAAA
GTGGACATTACAAGACGTTAGCCTTGAAGTGTATTTAACTGCACCAACAGGATGTA
>HPV16R_E15_fm_3339_to_3489
GTCCTACATCTGTGTTTAGCAGCAACGAAGTATCCTCTCCTGAAATTATTAGGCAGCACTTGGCCAACCACCCCGCCGCGACCCATACCAAAGCC
GTCGCCTTGGGCACCGAAGAAACACAGACGACTATCCAGCGACCAAGATCAGAGCC
>HPV16R_E16_fm_3778_to_3928
GCAACGTGACCAATTTTTGTCTCAAGTTAAAATACCAAAAACTATTACAGTGTCTACTGGATTTATGTCTATATGACAAATCTTGATACTGCATC
CACAACATTACTGGCGTGCTTTTTGCTTTGCTTTTGTGTGCTTTTGTGTGTCTGCC
>HPV16R_E18_fm_968_to_1118
ATGACGAGAACGAAAATGACAGTGATACAGGTGAAGATTTGGTAGATTTTATAGTAAATGATAATGATTATTTAACACAGGCAGAAACAGAGACA
GCACATGCGTTGTTTACTGCACAGGAAGCAAAACAACATAGAGATGCAGTACAGGT
>HPV16R_E19_fm_1115_to_1265
AGGTTCTAAAACGAAAGTATTTGGGTAGTCCACTTAGTGATATTAGTGGATGTGTAGACAATAATATTAGTCCTAGATTAAAAGCTATATGTATA
GAAAAACAAAGTAGAGCTGCAAAAAGGAGATTATTTGAAAGCGAAGACAGCGGGTA
>HPV16R_E20_fm_1361_to_1511
GTGGTTGCAGTCAGTACAGTAGTGGAAGTGGGGGAGAGGGTGTTAGTGAAAGACACACTATATGCCAAACACCACTTACAAATATTTTAAATGTA
CTAAAAACTAGTAATGCAAAGGCAGCAATGTTAGCAAAATTTAAAGAGTTATACGG
>HPV16R_E22_fm_1672_to_1822
TCATGGGGAATGGTTGTGTTACTATTAGTAAGATATAAATGTGGAAAAAATAGAGAAACAATTGAAAAATTGCTGTCTAAACTATTATGTGTGTC
TCCAATGTGTATGATGATAGAGCCTCCAAAATTGCGTAGTACAGCAGCAGCATTAT
>HPV16R_E23_fm_1860_to_2010
GTATGGAGACACGCCAGAATGGATACAAAGACAAACAGTATTACAACATAGTTTTAATGATTGTACATTTGAATTATCACAGATGGTACAATGGG
CCTACGATAATGACATAGTAGACGATAGTGAAATTGCATATAAATATGCACAATTG
>HPV16R_E24_fm_2069_to_2219
TAAAGGATTGTGCAACAATGTGTAGACATTATAAACGAGCAGAAAAAAAACAAATGAGTATGAGTCAATGGATAAAATATAGATGTGATAGGGTA
GATGATGGAGGTGATTGGAAGCAAATTGTTATGTTTTTAAGGTATCAAGGTGTAGA
>HPV16R_E25_fm_2288_to_2438
TATATGGTGCAGCTAACACAGGTAAATCATTATTTGGTATGAGTTTAATGAAATTTCTGCAAGGGTCTGTAATATGTTTTGTAAATTCTAAAAGC
CATTTTTGGTTACAACCATTAGCAGATGCCAAAATAGGTATGTTAGATGATGCTAC
>HPV16R_E26_fm_2476_to_2626
AATGCATTGGATGGAAATTTAGTTTCTATGGATGTAAAGCATAGACCATTGGTACAACTAAAATGCCCTCCATTATTAATTACATCTAACATTAA
TGCTGGTACAGATTCTAGGTGGCCTTATTTACATAATAGATTGGTGGTGTTTACAT
>HPV16R_E27_fm_2628_to_2778
TCCTAATGAGTTTCCATTTGACGAAAACGGAAATCCAGTGTATGAGCTTAATGATAAGAACTGGAAATCCTTTTTCTCAAGGACGTGGTCCAGAT
TAAGTTTGCACGAGGACGAGGACAAGGAAAACGATGGAGACTCTTTGCCAACGTTT
>HPV16R_E28_fm_2857_to_3007
ACACATGCGCCTAGAATGTGCTATTTATTACAAGGCCAGAGAAATGGGATTTAAACATATTAACCACCAAGTGGTGCCAACACTGGCTGTATCAA
AGAATAAAGCATTACAAGCAATTGAACTGCAACTAACGTTAGAAACAATATATAAC
>HPV16R_E29_fm_3037_to_3187
ACAAGACGTTAGCCTTGAAGTGTATTTAACTGCACCAACAGGATGTATAAAAAAACATGGATATACAGTGGAAGTGCAGTTTGATGGAGACATAT
GCAATACAATGCATTATACAAACTGGACACATATATATATTTGTGAAGAAGCATCA
>HPV16R_E30_fm_3199_to_3349
AGAGGGTCAAGTTGACTATTATGGTTTATATTATGTTCATGAAGGAATACGAACATATTTTGTGCAGTTTAAAGATGATGCAGAAAAATATAGTA
AAAATAAAGTATGGGAAGTTCATGCGGGTGGTCAGGTAATATTATGTCCTACATCT
>HPV16R_E31_fm_3455_to_3605
ACACAGACGACTATCCAGCGACCAAGATCAGAGCCAGACACCGGAAACCCCTGCCACACCACTAAGTTGTTGCACAGAGACTCAGTGGACAGTGC
TCCAATCCTCACTGCATTTAACAGCTCACACAAAGGACGGATTAACTGTAATAGTA
>HPV16R_E32_fm_3696_to_3846
CAGTGTCGTCTACATGGCATTGGACAGGACATAATGTAAAACATAAAAGTGCAATTGTTACACTTACATATGATAGTGAATGGCAACGTGACCAA
TTTTTGTCTCAAGTTAAAATACCAAAAACTATTACAGTGTCTACTGGATTTATGTC
```

A8. Source codes of ASP16 data analysis computer programs

See Results section 2.2.1.2 and 2.2.1.4. The Perl source codes of the computer programs for the ASP16 data analysis are shown. Four source codes for the four main program sets are described first, followed by the source codes of thirteen sub-programs for program sets 2-4.

Program set 1:

File name: **PROGRAM_SET_1_mac.pl**
Source code:

```
#!/usr/bin/perl

### About the script ###############################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
####################################################################################

####################################################################################
### MAIN PROGRAM BODY ##############################################################
####################################################################################

use strict;
use warnings;

####################################################################################
#####                        USERs can change paths here                       #####
##### PLEASE CHANGE THE PATHS FOR PROGRAMS LOCATIONS AND RESULTS LOCATION ##########
####################################################################################

######### 1 ########## Path where all Perl (scripts) programs are located on your computer:

my $pathprog = "/Volumes/MacHD2/DKFZ/PERL/";

################ 2 ################## HERE is the path where you want your results to be:

my $resultPath = "/Volumes/MacHD2/DKFZ/";

####################################################################################
#####                           USERs area ENDs                                #####
####################################################################################

my $prefix_path = $resultPath;
my @InputFile = @ARGV;
my @data1 = ();
my @STACK = ();
my @data1_name = ();
my @data1_seq = ();

my $pyroNum = int($InputFile[1]);
my $path = $prefix_path."Pyro".$pyroNum."_result/";

system ("mkdir", $path);

my $ba01 = 'TGAC';      # for sample 1
my $ba02 = 'AGAC';      # for sample 2
my $ba03 = 'TCAC';      # for sample 3
my $ba04 = 'ACAC';      # for sample 4
my $ba05 = 'TGTC';      # for sample 5
my $ba06 = 'AGTC';      # for sample 6
my $ba07 = 'TCTC';      # for sample 7
my $ba08 = 'ACTC';      # for sample 8
my $ba09 = 'CTGA';      # for sample 9
my $ba10 = 'CAGA';      # for sample 10
my $ba11 = 'CTCA';      # for sample 11
my $ba12 = 'CACA';      # for sample 12
my $ba13 = 'TAGC';      # for sample 13
my $ba14 = 'ATGC';      # for sample 14
my $ba15 = 'TACA';      # for sample 15
my $ba16 = 'ATCA';      # for sample 16
my $ba17 = 'TGCA';      # for sample 17
my $ba18 = 'TCGA';      # for sample 18
my $ba19 = 'AGCA';      # for sample 19
my $ba20 = 'ACGA';      # for sample 20
my $ba21 = 'CATC';      # for sample 21
my $ba22 = 'CTAC';      # for sample 22
my $ba23 = 'CTGC';      # for sample 23
my $ba24 = 'CAGC';      # for sample 24

my @barcode = ( $ba01, $ba02, $ba03, $ba04, $ba05, $ba06, $ba07, $ba08,
            $ba09, $ba10, $ba11, $ba12, $ba13, $ba14, $ba15, $ba16,
            $ba17, $ba18, $ba19, $ba20, $ba21, $ba22, $ba23, $ba24 );

my @sample01_name = (); my @sample01_seq = ();
my @sample02_name = (); my @sample02_seq = ();
my @sample03_name = (); my @sample03_seq = ();
my @sample04_name = (); my @sample04_seq = ();
my @sample05_name = (); my @sample05_seq = ();
my @sample06_name = (); my @sample06_seq = ();
my @sample07_name = (); my @sample07_seq = ();
my @sample08_name = (); my @sample08_seq = ();
my @sample09_name = (); my @sample09_seq = ();
my @sample10_name = (); my @sample10_seq = ();
my @sample11_name = (); my @sample11_seq = ();
my @sample12_name = (); my @sample12_seq = ();
my @sample13_name = (); my @sample13_seq = ();
my @sample14_name = (); my @sample14_seq = ();
my @sample15_name = (); my @sample15_seq = ();
my @sample16_name = (); my @sample16_seq = ();
my @sample17_name = (); my @sample17_seq = ();
my @sample18_name = (); my @sample18_seq = ();
my @sample19_name = (); my @sample19_seq = ();
my @sample20_name = (); my @sample20_seq = ();
my @sample21_name = (); my @sample21_seq = ();
my @sample22_name = (); my @sample22_seq = ();
my @sample23_name = (); my @sample23_seq = ();
my @sample24_name = (); my @sample24_seq = ();

@data1 = getFileData ($InputFile[0]);

my $barcodetosample = "barcode-to-sample-arrangement.txt";

print "\n\n=================================================================";
print "\n\n1111111111111111111111111111111111111111111111111111111111111111111111111111111";
print "\n\nPROGRAM 1 (SORTING) STARTS.....................................................\n\n";

open (BARSAM, ">$path$barcodetosample");

print BARSAM "\n\n\n";
print BARSAM "-----------------------------------------------------";
print BARSAM "\nDefault values for barcode-sample pair are used.\n\n";

print BARSAM $ba01." for sample 1\n"
         .$ba02." for sample 2\n"
         .$ba03." for sample 3\n"
         .$ba04." for sample 4\n"
         .$ba05." for sample 5\n"
         .$ba06." for sample 6\n"
         .$ba07." for sample 7\n"
         .$ba08." for sample 8\n"
         .$ba09." for sample 9\n"
         .$ba10." for sample 10\n"
         .$ba11." for sample 11\n"
         .$ba12." for sample 12\n"
         .$ba13." for sample 13\n"
         .$ba14." for sample 14\n"
         .$ba15." for sample 15\n"
         .$ba16." for sample 16\n"
         .$ba17." for sample 17\n"
         .$ba18." for sample 18\n"
         .$ba19." for sample 19\n"
         .$ba20." for sample 20\n"
         .$ba21." for sample 21\n"
         .$ba22." for sample 22\n"
         .$ba23." for sample 23\n"
         .$ba24." for sample 24\n\n";
```

```
print BARSAM "-------------------------------------------------";
close BARSAM;

@STACK = extractFastaStack (@data1);

extractFastaName (\@data1, \@STACK, \@data1_name);
extractFastaSeq (\@data1, \@STACK, \@data1_seq);

scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample01_name,\@sample01_seq,\@barcode,"0" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample02_name,\@sample02_seq,\@barcode,"1" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample03_name,\@sample03_seq,\@barcode,"2" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample04_name,\@sample04_seq,\@barcode,"3" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample05_name,\@sample05_seq,\@barcode,"4" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample06_name,\@sample06_seq,\@barcode,"5" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample07_name,\@sample07_seq,\@barcode,"6" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample08_name,\@sample08_seq,\@barcode,"7" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample09_name,\@sample09_seq,\@barcode,"8" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample10_name,\@sample10_seq,\@barcode,"9" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample11_name,\@sample11_seq,\@barcode,"10" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample12_name,\@sample12_seq,\@barcode,"11" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample13_name,\@sample13_seq,\@barcode,"12" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample14_name,\@sample14_seq,\@barcode,"13" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample15_name,\@sample15_seq,\@barcode,"14" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample16_name,\@sample16_seq,\@barcode,"15" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample17_name,\@sample17_seq,\@barcode,"16" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample18_name,\@sample18_seq,\@barcode,"17" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample19_name,\@sample19_seq,\@barcode,"18" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample20_name,\@sample20_seq,\@barcode,"19" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample21_name,\@sample21_seq,\@barcode,"20" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample22_name,\@sample22_seq,\@barcode,"21" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample23_name,\@sample23_seq,\@barcode,"22" );
scanSortBarcode2 (\@data1_name,\@data1_seq,\@sample24_name,\@sample24_seq,\@barcode,"23" );

saveSortSeqtoFile (\@sample01_name, \@sample01_seq, "sample01.txt", $path, "01");
saveSortSeqtoFile (\@sample02_name, \@sample02_seq, "sample02.txt", $path, "02");
saveSortSeqtoFile (\@sample03_name, \@sample03_seq, "sample03.txt", $path, "03");
saveSortSeqtoFile (\@sample04_name, \@sample04_seq, "sample04.txt", $path, "04");
saveSortSeqtoFile (\@sample05_name, \@sample05_seq, "sample05.txt", $path, "05");
saveSortSeqtoFile (\@sample06_name, \@sample06_seq, "sample06.txt", $path, "06");
saveSortSeqtoFile (\@sample07_name, \@sample07_seq, "sample07.txt", $path, "07");
saveSortSeqtoFile (\@sample08_name, \@sample08_seq, "sample08.txt", $path, "08");
saveSortSeqtoFile (\@sample09_name, \@sample09_seq, "sample09.txt", $path, "09");
saveSortSeqtoFile (\@sample10_name, \@sample10_seq, "sample10.txt", $path, "10");
saveSortSeqtoFile (\@sample11_name, \@sample11_seq, "sample11.txt", $path, "11");
saveSortSeqtoFile (\@sample12_name, \@sample12_seq, "sample12.txt", $path, "12");
saveSortSeqtoFile (\@sample13_name, \@sample13_seq, "sample13.txt", $path, "13");
saveSortSeqtoFile (\@sample14_name, \@sample14_seq, "sample14.txt", $path, "14");
saveSortSeqtoFile (\@sample15_name, \@sample15_seq, "sample15.txt", $path, "15");
saveSortSeqtoFile (\@sample16_name, \@sample16_seq, "sample16.txt", $path, "16");
saveSortSeqtoFile (\@sample17_name, \@sample17_seq, "sample17.txt", $path, "17");
saveSortSeqtoFile (\@sample18_name, \@sample18_seq, "sample18.txt", $path, "18");
saveSortSeqtoFile (\@sample19_name, \@sample19_seq, "sample19.txt", $path, "19");
saveSortSeqtoFile (\@sample20_name, \@sample20_seq, "sample20.txt", $path, "20");
saveSortSeqtoFile (\@sample21_name, \@sample21_seq, "sample21.txt", $path, "21");
saveSortSeqtoFile (\@sample22_name, \@sample22_seq, "sample22.txt", $path, "22");
saveSortSeqtoFile (\@sample23_name, \@sample23_seq, "sample23.txt", $path, "23");
saveSortSeqtoFile (\@sample24_name, \@sample24_seq, "sample24.txt", $path, "24");

print "\n\nBarcode-to-Sample arrangement can be found at";
print "\n ".$path.$barcodetosample;
print "\n\n-------------------------------------------------------------------------------";
print "\n\nThe sequences are sorted into 24 files";
print "\n   each corresponds to one barcode.";
print "\n   They are saved under directories :";
print "\n   ".$path."XX(sample-number)/";
print "\n\nPROGRAM 1 (SORTING) FINISHED.....................................................\n\n";
print "\n\n1111111111111111111111111111111111111111111111111111111111111111111111111111111111\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
##############################################################################################



##############################################################################################
###    SUBROUTINES    ########################################################################
##############################################################################################

sub printData {
        my (@nameqq) = @_;
        use strict;
        use warnings;

        print @nameqq;
        print "\n\n";
}

sub getFileData {
        my ($filename) = @_;
        use strict;
        use warnings;

        my @filedata = ();
        unless ( open(FILE_DATA, $filename) ) {
           print STDERR "\n\nThe program can't open the files \"$filename\"\n\n";
           print "Please re-check input file name and its location\n";
           print "The correct command should be :\n\n";
           print "\t\tperl PROGRAM(location+name) INPUTFastAFILE(location+name)\n\n";
           print "The command should be given under directory";
           print "  c:/Perl/bin> in case of Dos Terminal\n\n\n";
           exit;
        }
        @filedata = <FILE_DATA>;
        close FILE_DATA;
        return @filedata;
}

## getting a STACK array , membership# = line# of original file data
#         0 = name
#         1 = sequence
#         2 = others
sub extractFastaStack {
        my (@fastaFileData) = @_;
        use strict;
        use warnings;

        my $maxcounter = scalar @fastaFileData;
        my @stacklist = ();
        for (my $m = 0; $m < $maxcounter ; $m++) {
           if ($fastaFileData[$m] =~ /^\s*$/) {
                        push @stacklist, 2;
           } elsif ($fastaFileData[$m] =~ /^\s*#/) {
                        push @stacklist, 2;
           } elsif ($fastaFileData[$m] =~ />/) {
                        push @stacklist, 0;
           } else {
                        push @stacklist, 1;
           }
        }
        return @stacklist;
}

## get the sequences into an array
sub extractFastaSeq {
        my ($data1, $STACK, $data1_seq) = @_;
        use strict;
        use warnings;

        my $maxcounter = scalar @$data1;
        for (my $p = 0; $p < $maxcounter ; $p++) {
           if ($$STACK[$p] =~ "0") {
                        next;
           } elsif ( ($$STACK[$p] =~ "1") && ($$STACK[$p - 1] =~ "0")) {
                        push @$data1_seq, $$data1[$p];
           } elsif (($$STACK[$p] =~ "1") && ($$STACK[$p - 1] =~ "1")) {
                        my $lastEntry = (scalar @$data1_seq) - 1;
                        $$data1_seq[$lastEntry] .= $$data1[$p];
           } else { next; }
        }
}

## get the names into an array
sub extractFastaName {
        my ($data1, $STACK, $data1_name) = @_;
        use strict;
        use warnings;
```

```
          my $maxcounter = scalar @$data1;
          for (my $p = 0; $p < $maxcounter ; $p++) {
             if ($$STACK[$p] =~ "0") {
                        push @$data1_name, $$data1[$p];
             } else { next; }
          }
}

## sorted by barcodes, the sequences are separated into individual sample
sub scanSortBarcode2 {
          my ($data1_name, $data1_seq, $sampleXX_name, $sampleXX_seq, $barcode, $code) = @_;
          use strict;
          use warnings;

          my $maxcount = scalar @$data1_name;
          for (my $p = 0; $p < $maxcount ; $p++) {
             if ($$data1_seq[$p] =~ m/^$barcode[$code]/i) {
                        push @$sampleXX_name, $$data1_name[$p];
                        push @$sampleXX_seq, $$data1_seq[$p];
             } else { next; }
          }
}

## write the FastA formatted sequences of each sample into files in D:/perlresult/
sub saveSortSeqtoFile {
          my ($sampleXX_name, $sampleXX_seq, $filename, $path, $dirnum) = @_;
          use strict;
          use warnings;

          my @nameAndSeq = ();
          system ("mkdir", $path.$dirnum);
          my $maxcount = scalar @$sampleXX_name;
          for (my $p = 0; $p < $maxcount ; $p++) {
             push @nameAndSeq, $$sampleXX_name[$p];
             push @nameAndSeq, $$sampleXX_seq[$p];
          }
          open (NAMESEQ, ">$path$dirnum/$filename");
          print NAMESEQ @nameAndSeq;
          close NAMESEQ;

          my $namelist = "_nameList";
          open (NAME, ">$path$dirnum/sample$dirnum$namelist");
          for (my $m = 0; $m < $maxcount ; $m++) {
             print NAME $$sampleXX_name[$m];
          }
          close NAME;
}

#xx Sub-routines END HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
###############################################################################

###############################################################################################################
```

Program set 2:

File name: **PROGRAM_SET_2_mac.pl**
Source code:

```
#!/usr/bin/perl

### About the script ###################################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
#######################################################################################
### MAIN PROGRAM BODY ##################################################################
#######################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;
my $input = $sampleNo[0];
my $pyroNum = $sampleNo[1];

#######################################################################################
#####                    USERs can change paths here                  ##################
##### PLEASE CHANGE THE PATHS FOR PROGRAMS LOCATIONS AND RESULTS LOCATION ##################
#######################################################################################

######## 1 ########## Path where all Perl (scripts) programs are located on your computer:

my $pathprog = "/Volumes/MacHD2/DKFZ/PERL/";

################ 2 ################## HERE is the path where you want your results to be:

my $resultPath = "/Volumes/MacHD2/DKFZ/";

#######################################################################################
#####                        USERs area ENDs                          ##################
#######################################################################################

print "\n\n=========================================================================";
print "\n\nRUNNING PROGRAM SET 2 (P2-7) AFTER SORTING, FOR BARCODE NUMBER : ".$input."\n";
print "     For experiments using primers E01-E32\n";
print "\n\n     The program set works on the REVERSE-COMPLEMENT sequences!!!\n";

my $p2 = "set2_p2mac.pl";
my $p3ee = "set2_p3mac.pl";
my $p4 = "set2_p4mac.pl";
my $p5a = "set2_p5mac_a_noCutoff.pl";
my $p5d = "set2_p5mac_d_28bpCutoff.pl";
my $p5after = "set2_p5mac_edit.pl";
my $p6ee = "set2_p6mac_before.pl";
my $p6afteree = "set2_p6mac_fasta.pl";
my $p9ee = "set2_p7mac.pl";


system ("perl $pathprog$p2 $input $pyroNum $resultPath");
system ("perl $pathprog$p3ee $input $pyroNum $resultPath");
system ("perl $pathprog$p4 $input $pyroNum $resultPath");
system ("perl $pathprog$p5a $input $pyroNum $resultPath");
system ("perl $pathprog$p5d $input $pyroNum $resultPath");
system ("perl $pathprog$p5after $input $pyroNum $resultPath");
system ("perl $pathprog$p6ee $input $pyroNum $resultPath");
system ("perl $pathprog$p6afteree $input $pyroNum $resultPath");
system ("perl $pathprog$p9ee $input $pyroNum $resultPath");


print "\n\nRUNNING ALL PROGRAMS (2-7) AFTER SORTING, FOR BARCODE NUMBER : ".$input."\n";
print ".....................................................FINISHED\n";
print "\n\n=========================================================\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
###############################################################################################

###############################################################################################################
```

Program set 3:

File name: **PROGRAM_SET_3_mac.pl**
Source code:

```
#!/usr/bin/perl

### About the script ###################################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
#######################################################################################
### MAIN PROGRAM BODY ##################################################################
#######################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;
my $input = $sampleNo[0];
my $pyroNum = $sampleNo[1];
```

```
################################################################################
#####                    USERs can change paths here                    ##################
##### PLEASE CHANGE THE PATHS FOR PROGRAMS LOCATIONS AND RESULTS LOCATION ##################
################################################################################

######### 1 ########## Path where all Perl (scripts) programs are located on your computer:

my $pathprog = "/Volumes/MacHD2/DKFZ/PERL/";

################ 2 ################## HERE is the path where you want your results to be:

my $resultPath = "/Volumes/MacHD2/DKFZ/";

################################################################################
#####                        USERs area ENDs                          ##################
################################################################################

print "\n\n=========================================================================";
print "\n\nRUNNING PROGRAMS (8-9), FOR BARCODE NUMBER : ".$input."\n";
print "     For experiments using primers E01-E32\n";
print "\n\n     The program set works on the REVERSE-COMPLEMENT sequences!!!\n";

my $p10ee = "set3_p8mac.pl";
my $p11ee = "set3_p9mac.pl";

system ("perl $pathprog$p10ee $input $pyroNum $resultPath");
system ("perl $pathprog$p11ee $input $pyroNum $resultPath");

print "\n\nRUNNING ALL PROGRAMS (8-9), FOR BARCODE NUMBER : ".$input."\n";
print "...................................................................FINISHED\n";
print "\n\n=========================================================================\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
################################################################################

################################################################################################################
```

Program set 4:

File name: **PROGRAM_SET_4_mac.pl**
Source code:

```
#!/usr/bin/perl

### About the script ###########################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
################################################################################
### MAIN PROGRAM BODY ##########################################################
################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;
my $input = $sampleNo[0];
my $pyroNum = $sampleNo[1];

################################################################################
#####                    USERs can change paths here                    ##################
##### PLEASE CHANGE THE PATHS FOR PROGRAMS LOCATIONS AND RESULTS LOCATION ##################
################################################################################

######### 1 ########## Path where all Perl (scripts) programs are located on your computer:

my $pathprog = "/Volumes/MacHD2/DKFZ/PERL/";

################ 2 ################## HERE is the path where you want your results to be:

my $resultPath = "/Volumes/MacHD2/DKFZ/";

################################################################################
#####                        USERs area ENDs                          ##################
################################################################################

my @allbarcodes = ("01","02","03","04","05","06","07","08","09","10","11","12",
                   "13","14","15","16","17","18","19","20","21","22","23","24");

print "\n\n=========================================================================";
print "\n\nRUNNING PROGRAMS (10-11), for ALL barcodes\n";
print "     This program is required to run ONLY ONCE\n";
print "\n\n     It will perform for ALL BARCODEs, even one barcode is given as input\n";

my $pBefore22ee = "set4_p10mac.pl";
my $p22ee = "set4_p11mac.pl";

system ("perl $pathprog$pBefore22ee $input $pyroNum $resultPath");

for (my $p = 0; $p < (scalar @allbarcodes); $p++) {
        system ("perl $pathprog$p22ee $allbarcodes[$p] $pyroNum $resultPath");
}

print "\n\nRUNNING ALL PROGRAMS (10-11), FOR ALL BARCODES\n";
print "...................................................................FINISHED\n";
print "\n\n=========================================================================\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
################################################################################

################################################################################################################
```

Sub-programs for program set 2-4:

Program set 2 consists of nine sub-programs whose names start with prefix "`set2_`". Program set 3 consists of two sub-programs with prefix "`set3_`". And program set 4 consists of the last two sub-programs with prefix "`set4_`".

File name: **set2_p2mac.pl**
Source code:

```
#!/usr/bin/perl

### About the script ###########################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
################################################################################
### MAIN PROGRAM BODY ##########################################################
################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;

my $blast = "/blast-2.2.17/";                   # path of blast program/db
my $blastall = $blast."bin/blastall";           # path of blastall program
my $dbHPV = $blast."db/hpv16";                  # path of hpv16 database

my $prefix_path= $sampleNo[2];
my $pyroNum = int($sampleNo[1]);

my $path = $prefix_path."Pyro".$pyroNum."_result/";                     # path to save the result files
my $InputFile = $path.$sampleNo[0]."/sample".$sampleNo[0]."\.txt";
   # INPUT FASTAs
        # This input contains sorted sequences acc. to barcode,
        # They will be reverse-complemented before being blasted to hpv16

my $pathSave = $path.$sampleNo[0]."/blastHPV16/";           # directory to save blasted files
my $infofile = "numberOfsequences.txt";
my $workDir = $path.$sampleNo[0]."/";
my $rc = $workDir."sample".$sampleNo[0]."_RC.txt";          # file for REVERSE-COMPLEMENTed FastA seq

print "\n\nPROGRAM 2 (BLAST HPV16) STARTS.....................................\n\n";

system ("mkdir", $pathSave);

my @data2 = ();
```

```
my @STACK2 = ();
my @data2_name = ();
my @data2_seq = ();

@data2 = getFileData ($InputFile);
@STACK2 = extractFastaStack (@data2);

extractFastaName (\@data2, \@STACK2, \@data2_name);
extractFastaSeq (\@data2, \@STACK2, \@data2_seq);
bl2seqToHpv16minus (\@data2_name, \@data2_seq, $pathSave, $blastall, $dbHPV, $rc);

open (FILEINFO, ">$workDir$infofile");
print FILEINFO scalar @data2_name;
close FILEINFO;

print "FINISHED\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
################################################################################

################################################################################
###    SUBROUTINES    ##########################################################
################################################################################

sub getFileData {
          my ($filename) = @_;
          use strict;
          use warnings;

          my @filedata = ();
          unless ( open(FILE_DATA, $filename) ) {
             print STDERR "\n\nThe program can't open the files \"$filename\"\n\n";
             print "Please re-check input file name and its location\n";
             print "The correct command should be :\n\n";
             print "\t\tperl PROGRAM(location+name) INPUTFastAFILE(location+name)\n\n";
             print "The command should be given under directory";
             print " c:/Perl/bin> in case of Dos Terminal\n\n\n";
             exit;
          }
          @filedata = <FILE_DATA>;
          close FILE_DATA;
          return @filedata;
}

sub extractFastaStack {
          my (@fastaFileData) = @_;
          use strict;
          use warnings;

          my $maxcounter = scalar @fastaFileData;
          my @stacklist = ();
          for (my $m = 0; $m < $maxcounter ; $m++) {
             if ($fastaFileData[$m] =~ /^\s*$/) {
                          push @stacklist, 2;
             } elsif ($fastaFileData[$m] =~ /^\s*#/) {
                          push @stacklist, 2;
             } elsif ($fastaFileData[$m] =~ /^>/) {
                          push @stacklist, 0;
             } else {
                          push @stacklist, 1;
             }
          }
          return @stacklist;
}

sub extractFastaName {
          my ($data1, $STACK, $data1_name) = @_;
          use strict;
          use warnings;

          my $maxcounter = scalar @$data1;
          for (my $p = 0; $p < $maxcounter ; $p++) {
             if ($$STACK[$p] =~ "0") {
                          push @$data1_name, $$data1[$p];
             } else { next; }
          }
}
```

```
sub extractFastaSeq {
          my ($data1, $STACK, $data1_seq) = @_;
          use strict;
          use warnings;

          my $maxcounter = scalar @$data1;
          for (my $p = 0; $p < $maxcounter ; $p++) {
             if ($$STACK[$p] =~ "0") {
                          next;
             } elsif ( ($$STACK[$p] =~ "1") && ($$STACK[$p - 1] =~ "0")) {
                          push @$data1_seq, $$data1[$p];
             } elsif (($$STACK[$p] =~ "1") && ($$STACK[$p - 1] =~ "1")) {
                          my $lastEntry = (scalar @$data1_seq) - 1;
                          $$data1_seq[$lastEntry] .= $$data1[$p];
             } else { next; }
          }
}

# CREATE REVERSE-COMPLEMENT
sub bl2seqToHpv16minus {
          my ($name, $seq, $dir, $blastall, $db, $fastarev) = @_;
          use strict;
          use warnings;

          my $count = scalar @$name;
          my $fastAseq = "fastaseq_";
          my $fastAblast = "blasted_";

          unless (open (RC, ">$fastarev")) {print "\ncan not create $fastarev\n\n"; exit;}

          for (my $p = 0; $p < $count ; $p++) {
             my $dna = $$seq[$p];                  # complete single seq BUT with 'new line's
             $dna =~ s/\s//g;                              # get rid of 'new line's
             my @FT = split ('', $dna);                    # array of 'single' nt of each complete seq

             unless (open (FA, ">$dir$fastAseq$p")) {print "\ncan not create $dir$fastAseq$p\n\n"; exit;}
             my $nt;

             print FA $$name[$p];
             print RC $$name[$p];

             foreach $nt( reverse(@FT)) {
                          if ($nt =~ /a/i) {
                                       print FA "T";
                                       print RC "T";
                          } elsif ($nt =~ /t/i) {
                                       print FA "A";
                                       print RC "A";
                          } elsif ($nt =~ /g/i) {
                                       print FA "C";
                                       print RC "C";
                          } elsif ($nt =~ /c/i) {
                                       print FA "G";
                                       print RC "G";
                          } elsif ($nt =~ /n/i) {
                                       print FA "N";
                                       print RC "N";
                          } else {
                                       die "$0: bad nucleotide! [$nt]\n\n";
                          }
             }
             close FA;
             print RC "\n";

             system ("$blastall -p blastn -d $db -i $dir$fastAseq$p -o $dir$fastAblast$p -F F");
             system ("rm", "$dir$fastAseq$p");
          }

          close RC;
}

#xx Sub-routines END HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
################################################################################

################################################################################################################
```

File name: **set2_p3mac.pl**

Source code:

```
#!/usr/bin/perl

### About the script ###################################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
#######################################################################################
### MAIN PROGRAM BODY ##################################################################
#######################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;
my $blastPath = "/blast-2.2.17/";                                   # path of blast program/db
my $bAll = $blastPath."bin/blastall";                               # path of blastall program
my $dbEE = $blastPath."db/EEpyro3";                                 # path of EEpyro3 database
my $prefix_path= $sampleNo[2];
my $pyroNum = int($sampleNo[1]);

my $path = $prefix_path."Pyro".$pyroNum."_result/";                 # path to save the result files

my $samplePath = $path.$sampleNo[0]."/";
my $workDir = $samplePath."blastEE/";

# INPUT SEQUENCES ARE REVERSE-COMPLEMENTARY SEQUENCES FROM PROGRAM2, in the file:
my $inOrig = $samplePath."sample".$sampleNo[0]."_RC.txt";

print "\n\nPROGRAM 3 (BLAST EE primers) STARTS.....................................\n\n";
system ("mkdir", $workDir);

# Part 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
#         Blast each sequence with EE primers database

my @orig = getFileData ($inOrig);
my @stack = extractFastaStack (@orig);
my @orig_name = ();
my @orig_seq = ();

extractFastaName (\@orig, \@stack, \@orig_name);
extractFastaSeq (\@orig, \@stack, \@orig_seq);
bl2seqToEE (\@orig_name, \@orig_seq, $workDir, $bAll, $dbEE);

# Part 2222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
#         Parse the information :

my $inBlastEE = $workDir."blast_EE_";
my $report = $workDir."parsed_EE_allreport";
my $seqCount = scalar @orig_name;

parseEE ($inBlastEE, $report, $seqCount);

# Part 3333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333
my $statrep = $workDir."parsed_EE_statistics";  # stats report after selection

my $nohitrep = $workDir."parsed_EE_nohit_report";           # List of NO-HIT sequences
my $comrep = $workDir."parsed_EE_combined_report";          # List of ALL (Hit & No-HIT)

my $oE17 = $workDir."parsed_EE_E17_report";
my $oE02 = $workDir."parsed_EE_E02_report";
my $oE03 = $workDir."parsed_EE_E03_report";
my $oE04 = $workDir."parsed_EE_E04_report";
my $oE05 = $workDir."parsed_EE_E05_report";
my $oE06 = $workDir."parsed_EE_E06_report";
my $oE07 = $workDir."parsed_EE_E07_report";
my $oE08 = $workDir."parsed_EE_E08_report";
my $oE09 = $workDir."parsed_EE_E09_report";
my $oE21 = $workDir."parsed_EE_E21_report";
my $oE11 = $workDir."parsed_EE_E11_report";
my $oE12 = $workDir."parsed_EE_E12_report";
my $oE13 = $workDir."parsed_EE_E13_report";
my $oE14 = $workDir."parsed_EE_E14_report";
my $oE15 = $workDir."parsed_EE_E15_report";
my $oE16 = $workDir."parsed_EE_E16_report";
my $oE18 = $workDir."parsed_EE_E18_report";
my $oE19 = $workDir."parsed_EE_E19_report";
my $oE20 = $workDir."parsed_EE_E20_report";
my $oE22 = $workDir."parsed_EE_E22_report";
my $oE23 = $workDir."parsed_EE_E23_report";
my $oE24 = $workDir."parsed_EE_E24_report";
my $oE25 = $workDir."parsed_EE_E25_report";
my $oE26 = $workDir."parsed_EE_E26_report";
my $oE27 = $workDir."parsed_EE_E27_report";
my $oE28 = $workDir."parsed_EE_E28_report";
my $oE29 = $workDir."parsed_EE_E29_report";
my $oE30 = $workDir."parsed_EE_E30_report";
my $oE31 = $workDir."parsed_EE_E31_report";
my $oE32 = $workDir."parsed_EE_E32_report";

getStatEEpyro32 ($report, $statrep, $seqCount, $nohitrep, $comrep,
         $oE17, $oE02, $oE03, $oE04, $oE05, $oE06, $oE07, $oE08,
         $oE09, $oE21, $oE11, $oE12, $oE13, $oE14, $oE15, $oE16,
         $oE18, $oE19, $oE20, $oE22, $oE23, $oE24, $oE25, $oE26,
         $oE27, $oE28, $oE29, $oE30, $oE31, $oE32);

# Part 4444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444

getFastAprimer (\@orig_name, \@orig_seq, $oE17);
getFastAprimer (\@orig_name, \@orig_seq, $oE02);
getFastAprimer (\@orig_name, \@orig_seq, $oE03);
getFastAprimer (\@orig_name, \@orig_seq, $oE04);
getFastAprimer (\@orig_name, \@orig_seq, $oE05);
getFastAprimer (\@orig_name, \@orig_seq, $oE06);
getFastAprimer (\@orig_name, \@orig_seq, $oE07);
getFastAprimer (\@orig_name, \@orig_seq, $oE08);
getFastAprimer (\@orig_name, \@orig_seq, $oE09);
getFastAprimer (\@orig_name, \@orig_seq, $oE21);
getFastAprimer (\@orig_name, \@orig_seq, $oE11);
getFastAprimer (\@orig_name, \@orig_seq, $oE12);
getFastAprimer (\@orig_name, \@orig_seq, $oE13);
getFastAprimer (\@orig_name, \@orig_seq, $oE14);
getFastAprimer (\@orig_name, \@orig_seq, $oE15);
getFastAprimer (\@orig_name, \@orig_seq, $oE16);
getFastAprimer (\@orig_name, \@orig_seq, $oE18);
getFastAprimer (\@orig_name, \@orig_seq, $oE19);
getFastAprimer (\@orig_name, \@orig_seq, $oE20);
getFastAprimer (\@orig_name, \@orig_seq, $oE22);
getFastAprimer (\@orig_name, \@orig_seq, $oE23);
getFastAprimer (\@orig_name, \@orig_seq, $oE24);
getFastAprimer (\@orig_name, \@orig_seq, $oE25);
getFastAprimer (\@orig_name, \@orig_seq, $oE26);
getFastAprimer (\@orig_name, \@orig_seq, $oE27);
getFastAprimer (\@orig_name, \@orig_seq, $oE28);
getFastAprimer (\@orig_name, \@orig_seq, $oE29);
getFastAprimer (\@orig_name, \@orig_seq, $oE30);
getFastAprimer (\@orig_name, \@orig_seq, $oE31);
getFastAprimer (\@orig_name, \@orig_seq, $oE32);

print "FINISHED\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
#######################################################################################

#######################################################################################
###    SUBROUTINES    ###############################################################
#######################################################################################

sub getFileData {
         my ($filename) = @_;
         use strict;
         use warnings;

         my @filedata = ();
         unless ( open(FILE_DATA, $filename) ) {
            print STDERR "\n\nThe program can't open the files \"$filename\"\n\n\n\n";
            print "Please re-check input file name and its location\n";
            print "The correct command should be :\n\n";
            print "\t\tperl PROGRAM(location+name) INPUTFastAFILE(location+name)\n\n";
            print "The command should be given under directory";
            print "  c:/Perl/bin> in case of Dos Terminal\n\n\n";
```

```
            exit;
         }
         @filedata = <FILE_DATA>;
         close FILE_DATA;
         return @filedata;
}

# for Part 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
sub extractFastaStack {
         my (@fastaFileData) = @_;
         use strict;
         use warnings;

         my $maxcounter = scalar @fastaFileData;
         my @stacklist = ();
         for (my $m = 0; $m < $maxcounter ; $m++) {
            if ($fastaFileData[$m] =~ /^\s*$/) {
                     push @stacklist, 2;
            } elsif ($fastaFileData[$m] =~ /^\s*#/) {
                     push @stacklist, 2;
            } elsif ($fastaFileData[$m] =~ /^>/) {
                     push @stacklist, 0;
            } else {
                     push @stacklist, 1;
            }
         }
         return @stacklist;
}

sub extractFastaName {
         my ($data1, $STACK, $data1_name) = @_;
         use strict;
         use warnings;

         my $maxcounter = scalar @$data1;
         for (my $p = 0; $p < $maxcounter ; $p++) {
            if ($$STACK[$p] =~ "0") {
                     push @$data1_name, $$data1[$p];
            } else { next; }
         }
}

sub extractFastaSeq {
         my ($data1, $STACK, $data1_seq) = @_;
         use strict;
         use warnings;

         my $maxcounter = scalar @$data1;
         for (my $p = 0; $p < $maxcounter ; $p++) {
            if ($$STACK[$p] =~ "0") {
                     next;
            } elsif ( ($$STACK[$p] =~ "1") && ($$STACK[$p - 1] =~ "0")) {
                     push @$data1_seq, $$data1[$p];
            } elsif (($$STACK[$p] =~ "1") && ($$STACK[$p - 1] =~ "1")) {
                     my $lastEntry = (scalar @$data1_seq) - 1;
                     $$data1_seq[$lastEntry] .= $$data1[$p];
            } else { next; }
         }
}

sub bl2seqToEE {
         my ($name, $seq, $dir, $bAll, $db) = @_;
         use strict;
         use warnings;

         my $fasSeq = "fastaseq_";
         my $fasBlast = "blast_EE_";
         for (my $p = 0; $p < (scalar @$name); $p++) {
            my @fasta = ();
            push @fasta, $$name[$p];
            push @fasta, $$seq[$p];
            open (FASTA, ">$dir$fasSeq$p");
            print FASTA @fasta;
            close FASTA;
            system ("$bAll -p blastn -d $db -i $dir$fasSeq$p -o $dir$fasBlast$p -F F");
            system ("rm", "$dir$fasSeq$p");
         }
}
```

```
# for Part 2222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
sub parseEE {
         my ($inPath, $out, $count) = @_;
         use strict;
         use warnings;

         unless (open (OUT, ">$out")) {print "\n\n\n\nCan not create ".$out."\n\n\n\n"; exit;}
         for (my $p = 0; $p < $count; $p++) {
            my $inFile = $inPath.$p;
            my @in = getFileData ($inFile);
            if ($in[16] =~ m/^.......No.hits.found/) {
                        print OUT $p."#0EE";
                        $in[9] =~ /^(.*)\(([0-9]*).letters\)$/;
                        my $bpQueryNohit = $2;
                        print OUT "#".$bpQueryNohit."#noPrimer\n";
            } else {
                        print OUT $p."#1EE";
                        $in[9] =~ /^(.*)\(([0-9]*).letters\)$/;
                        my $bpQuery = $2;
                        print OUT "#".$bpQuery;

                        for (my $a = 0; $a < (scalar @in); $a++) {

                                    if ($in[$a] =~ /^>(.[0-9][0-9])(.*)$/) {
                                                # primer
                                                my $prim = $1;
                                                # primer length
                                                $in[$a+1] =~ /^.........Length(...)([0-9]*)(.*)$/;
                                                my $primLength = $2;
                                                # scoreBP & percent
                                                my @inin = split (/\s+/, $in[$a+4]);
                                                my $scorebox = $inin[3];
                                                $scorebox =~ /^([0-9]*)\/([0-9]*)$/;
                                                my $scoreBP = $2;
                                                my $percentbox = $inin[4];
                                                $percentbox =~ /^\(([0-9]*)%\)/;
                                                my $percent = $1;
                                                # strand
                                                $in[$a+5] =~ /^.Strand.=.Plus...(.*)$/;
                                                my $strand = $1;
                                                # 1stMatchQuery
                                                $in[$a+8] =~ /^Query..([0-9]*)(.*)$/;
                                                my $firstMatch = $1;
                                                # 1stMatchPrimer
                                                $in[$a+10] =~ /^Sbjct..([0-9]*)(.*)$/;
                                                my $firstMatchPrim = $1;

                                                print OUT "#".$prim."%".$primLength."%".$scoreBP."%".
                                                            $percent."%".$firstMatch."%".$strand."%".$firstMatchPrim;
                                    } else { next; }  # close if loop
                        } # close for loop
                        print OUT "\n";
            } # close else loop
         } # close for loop
         close OUT;
}

# for Part 3333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333
sub getStatEEpyro32 {
         my ($inRep, $out, $total, $nohitrep, $combi,
            $oE1, $oE2, $oE3, $oE4, $oE5, $oE6, $oE7, $oE8,
            $oL1, $oL2, $oL3, $oL4, $oL5, $oL6, $oL7, $oL8,
            $oH1, $oH2, $oH3, $oH4, $oH5, $oH6, $oH7, $oH8,
            $oK1, $oK2, $oK3, $oK4, $oK5, $oK6 ) = @_;

         use strict;
         use warnings;

         my @in = getFileData ($inRep);

         # count for each hit primer, selected according to our conditions
         my ($e1, $e2, $e3, $e4, $e5, $e6, $e7, $e8) = (0,0,0,0,0,0,0,0);
         my ($l1, $l2, $l3, $l4, $l5, $l6, $l7, $l8) = (0,0,0,0,0,0,0,0);
         my ($h1, $h2, $h3, $h4, $h5, $h6, $h7, $h8) = (0,0,0,0,0,0,0,0);
         my ($k1, $k2, $k3, $k4, $k5, $k6)           = (0,0,0,0,0,0);

         # count for each hit primer, but INcomplete sequencing rxn
         my ($ine1, $ine2, $ine3, $ine4, $ine5, $ine6, $ine7, $ine8) = (0,0,0,0,0,0,0,0);
```

```
my ($inl1, $inl2, $inl3, $inl4, $inl5, $inl6, $inl7, $inl8) = (0,0,0,0,0,0,0,0);
my ($inh1, $inh2, $inh3, $inh4, $inh5, $inh6, $inh7, $inh8) = (0,0,0,0,0,0,0,0);
my ($ink1, $ink2, $ink3, $ink4, $ink5, $ink6)               = (0,0,0,0,0,0);

# count for each hit primer, but out-selected as "not significat"
my ($ne1, $ne2, $ne3, $ne4, $ne5, $ne6, $ne7, $ne8) = (0,0,0,0,0,0,0,0);
my ($nl1, $nl2, $nl3, $nl4, $nl5, $nl6, $nl7, $nl8) = (0,0,0,0,0,0,0,0);
my ($nh1, $nh2, $nh3, $nh4, $nh5, $nh6, $nh7, $nh8) = (0,0,0,0,0,0,0,0);
my ($nk1, $nk2, $nk3, $nk4, $nk5, $nk6)             = (0,0,0,0,0,0);

my $nohitsum = 0;
my $hitOutranged = 0;

unless (open (E1, ">$oE1")) {print "\n\n\n\nCan not create ".$oE1."\n\n\n\n"; exit;}
unless (open (E2, ">$oE2")) {print "\n\n\n\nCan not create ".$oE2."\n\n\n\n"; exit;}
unless (open (E3, ">$oE3")) {print "\n\n\n\nCan not create ".$oE3."\n\n\n\n"; exit;}
unless (open (E4, ">$oE4")) {print "\n\n\n\nCan not create ".$oE4."\n\n\n\n"; exit;}
unless (open (E5, ">$oE5")) {print "\n\n\n\nCan not create ".$oE5."\n\n\n\n"; exit;}
unless (open (E6, ">$oE6")) {print "\n\n\n\nCan not create ".$oE6."\n\n\n\n"; exit;}
unless (open (E7, ">$oE7")) {print "\n\n\n\nCan not create ".$oE7."\n\n\n\n"; exit;}
unless (open (E8, ">$oE8")) {print "\n\n\n\nCan not create ".$oE8."\n\n\n\n"; exit;}
unless (open (L1, ">$oL1")) {print "\n\n\n\nCan not create ".$oL1."\n\n\n\n"; exit;}
unless (open (L2, ">$oL2")) {print "\n\n\n\nCan not create ".$oL2."\n\n\n\n"; exit;}
unless (open (L3, ">$oL3")) {print "\n\n\n\nCan not create ".$oL3."\n\n\n\n"; exit;}
unless (open (L4, ">$oL4")) {print "\n\n\n\nCan not create ".$oL4."\n\n\n\n"; exit;}
unless (open (L5, ">$oL5")) {print "\n\n\n\nCan not create ".$oL5."\n\n\n\n"; exit;}
unless (open (L6, ">$oL6")) {print "\n\n\n\nCan not create ".$oL6."\n\n\n\n"; exit;}
unless (open (L7, ">$oL7")) {print "\n\n\n\nCan not create ".$oL7."\n\n\n\n"; exit;}
unless (open (L8, ">$oL8")) {print "\n\n\n\nCan not create ".$oL8."\n\n\n\n"; exit;}
unless (open (H1, ">$oH1")) {print "\n\n\n\nCan not create ".$oH1."\n\n\n\n"; exit;}
unless (open (H2, ">$oH2")) {print "\n\n\n\nCan not create ".$oH2."\n\n\n\n"; exit;}
unless (open (H3, ">$oH3")) {print "\n\n\n\nCan not create ".$oH3."\n\n\n\n"; exit;}
unless (open (H4, ">$oH4")) {print "\n\n\n\nCan not create ".$oH4."\n\n\n\n"; exit;}
unless (open (H5, ">$oH5")) {print "\n\n\n\nCan not create ".$oH5."\n\n\n\n"; exit;}
unless (open (H6, ">$oH6")) {print "\n\n\n\nCan not create ".$oH6."\n\n\n\n"; exit;}
unless (open (H7, ">$oH7")) {print "\n\n\n\nCan not create ".$oH7."\n\n\n\n"; exit;}
unless (open (H8, ">$oH8")) {print "\n\n\n\nCan not create ".$oH8."\n\n\n\n"; exit;}
unless (open (K1, ">$oK1")) {print "\n\n\n\nCan not create ".$oK1."\n\n\n\n"; exit;}
unless (open (K2, ">$oK2")) {print "\n\n\n\nCan not create ".$oK2."\n\n\n\n"; exit;}
unless (open (K3, ">$oK3")) {print "\n\n\n\nCan not create ".$oK3."\n\n\n\n"; exit;}
unless (open (K4, ">$oK4")) {print "\n\n\n\nCan not create ".$oK4."\n\n\n\n"; exit;}
unless (open (K5, ">$oK5")) {print "\n\n\n\nCan not create ".$oK5."\n\n\n\n"; exit;}
unless (open (K6, ">$oK6")) {print "\n\n\n\nCan not create ".$oK6."\n\n\n\n"; exit;}

unless (open (NOHITREP, ">$nohitrep")) {print "\n\nCan not create ".$nohitrep."\n\n";exit;}
unless (open (COMBI, ">$combi")) {print "\n\nCan not create ".$combi."\n\n\n\n";exit;}

for (my $a = 0; $a < (scalar @in); $a++) {
   my @line = split (/#/, $in[$a]);
   my $hitrep = $line[1];
   my $zeroEL = "0EE";
   my $oneEL = "1EE";
   my ($b1_1, $b1_2, $b1_3, $b1_4, $b1_5, $b1_6, $b1_7);

   if ($hitrep =~ $zeroEL) {
               $nohitsum++;
               my $querynohit = $line[0];
               print NOHITREP $querynohit."#nohitEE#noPrimer\n";
               print COMBI $querynohit."#nohitEE#noPrimer\n";

   } elsif ($hitrep =~ $oneEL) {

               my $query = $line[0];
               my $queryLeng = $line[2];
               my @box1 = split (/%/, $line[3]);   # FIRST PRIMER HIT
               $b1_1 = $box1[0];                   # Primer name
               $b1_2 = $box1[1];                   # Primer length
               $b1_3 = $box1[2];                   # scored bp
               $b1_4 = $box1[3];                   # Percent match
               $b1_5 = $box1[4];                   # First position match of query
               $b1_6 = $box1[5];                   # Strand (must be MINUS)
               $b1_7 = $box1[6];                   # First position match of Sbjct (primer)

               # matching whole primer
               my $minus = "Minus";
               my $plus  = "Plus";
```

```
# 'perfect' & complete match  (CASE A1)
if ( ($b1_6 = $plus) && (int($b1_4) >= 90)
               && (int($b1_3) >= int($b1_2)) ) {

               if ($b1_1 =~ "E17") {
                           $e1++; print E1 $query."#E17#complete\n";
                           print COMBI $query."#E17#complete\n";
               } elsif ($b1_1 =~ "E02") {
                           $e2++; print E2 $query."#E02#complete\n";
                           print COMBI $query."#E02#complete\n";
               } elsif ($b1_1 =~ "E03") {
                           $e3++; print E3 $query."#E03#complete\n";
                           print COMBI $query."#E03#complete\n";
               } elsif ($b1_1 =~ "E04") {
                           $e4++; print E4 $query."#E04#complete\n";
                           print COMBI $query."#E04#complete\n";
               } elsif ($b1_1 =~ "E05") {
                           $e5++; print E5 $query."#E05#complete\n";
                           print COMBI $query."#E05#complete\n";
               } elsif ($b1_1 =~ "E06") {
                           $e6++; print E6 $query."#E06#complete\n";
                           print COMBI $query."#E06#complete\n";
               } elsif ($b1_1 =~ "E07") {
                           $e7++; print E7 $query."#E07#complete\n";
                           print COMBI $query."#E07#complete\n";
               } elsif ($b1_1 =~ "E08") {
                           $e8++; print E8 $query."#E08#complete\n";
                           print COMBI $query."#E08#complete\n";
               } elsif ($b1_1 =~ "E09") {
                           $l1++; print L1 $query."#E09#complete\n";
                           print COMBI $query."#E09#complete\n";
               } elsif ($b1_1 =~ "E21") {
                           $l2++; print L2 $query."#E21#complete\n";
                           print COMBI $query."#E21#complete\n";
               } elsif ($b1_1 =~ "E11") {
                           $l3++; print L3 $query."#E11#complete\n";
                           print COMBI $query."#E11#complete\n";
               } elsif ($b1_1 =~ "E12") {
                           $l4++; print L4 $query."#E12#complete\n";
                           print COMBI $query."#E12#complete\n";
               } elsif ($b1_1 =~ "E13") {
                           $l5++; print L5 $query."#E13#complete\n";
                           print COMBI $query."#E13#complete\n";
               } elsif ($b1_1 =~ "E14") {
                           $l6++; print L6 $query."#E14#complete\n";
                           print COMBI $query."#E14#complete\n";
               } elsif ($b1_1 =~ "E15") {
                           $l7++; print L7 $query."#E15#complete\n";
                           print COMBI $query."#E15#complete\n";
               } elsif ($b1_1 =~ "E16") {
                           $l8++; print L8 $query."#E16#complete\n";
                           print COMBI $query."#E16#complete\n";
               } elsif ($b1_1 =~ "E18") {
                           $h1++; print H1 $query."#E18#complete\n";
                           print COMBI $query."#E18#complete\n";
               } elsif ($b1_1 =~ "E19") {
                           $h2++; print H2 $query."#E19#complete\n";
                           print COMBI $query."#E19#complete\n";
               } elsif ($b1_1 =~ "E20") {
                           $h3++; print H3 $query."#E20#complete\n";
                           print COMBI $query."#E20#complete\n";
               } elsif ($b1_1 =~ "E22") {
                           $h4++; print H4 $query."#E22#complete\n";
                           print COMBI $query."#E22#complete\n";
               } elsif ($b1_1 =~ "E23") {
                           $h5++; print H5 $query."#E23#complete\n";
                           print COMBI $query."#E23#complete\n";
               } elsif ($b1_1 =~ "E24") {
                           $h6++; print H6 $query."#E24#complete\n";
                           print COMBI $query."#E24#complete\n";
               } elsif ($b1_1 =~ "E25") {
                           $h7++; print H7 $query."#E25#complete\n";
                           print COMBI $query."#E25#complete\n";
               } elsif ($b1_1 =~ "E26") {
                           $h8++; print H8 $query."#E26#complete\n";
                           print COMBI $query."#E26#complete\n";
               } elsif ($b1_1 =~ "E27") {
```

```
                        $k1++; print K1 $query."#E27#complete\n";
                        print COMBI $query."#E27#complete\n";
                } elsif ($b1_1 =~ "E28") {
                        $k2++; print K2 $query."#E28#complete\n";
                        print COMBI $query."#E28#complete\n";
                } elsif ($b1_1 =~ "E29") {
                        $k3++; print K3 $query."#E29#complete\n";
                        print COMBI $query."#E29#complete\n";
                } elsif ($b1_1 =~ "E30") {
                        $k4++; print K4 $query."#E30#complete\n";
                        print COMBI $query."#E30#complete\n";
                } elsif ($b1_1 =~ "E31") {
                        $k5++; print K5 $query."#E31#complete\n";
                        print COMBI $query."#E31#complete\n";
                } elsif ($b1_1 =~ "E32") {
                        $k6++; print K6 $query."#E32#complete\n";
                        print COMBI $query."#E32#complete\n";
                } else { next; }

        # 'imperfect' & complete match  (CASE A2)
        } elsif ( ($b1_6 =~ $plus) && (int($b1_4) >= 90)
                && (int($b1_3) < int($b1_2))
                && (int($b1_7) == 1)
                && ((int($b1_3)+int($b1_7)) > 20)
                && ((int($b1_2)-(int($b1_3)+int($b1_7))) < 5) ) {

                if ($b1_1 =~ "E17") {
                        $ine1++; print E1 $query."#E17#INcomplete_tilHPV\n";
                        print COMBI $query."#E17#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E02") {
                        $ine2++; print E2 $query."#E02#INcomplete_tilHPV\n";
                        print COMBI $query."#E02#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E03") {
                        $ine3++; print E3 $query."#E03#INcomplete_tilHPV\n";
                        print COMBI $query."#E03#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E04") {
                        $ine4++; print E4 $query."#E04#INcomplete_tilHPV\n";
                        print COMBI $query."#E04#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E05") {
                        $ine5++; print E5 $query."#E05#INcomplete_tilHPV\n";
                        print COMBI $query."#E05#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E06") {
                        $ine6++; print E6 $query."#E06#INcomplete_tilHPV\n";
                        print COMBI $query."#E06#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E07") {
                        $ine7++; print E7 $query."#E07#INcomplete_tilHPV\n";
                        print COMBI $query."#E07#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E08") {
                        $ine8++; print E8 $query."#E08#INcomplete_tilHPV\n";
                        print COMBI $query."#E08#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E09") {
                        $inl1++; print L1 $query."#E09#INcomplete_tilHPV\n";
                        print COMBI $query."#E09#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E21") {
                        $inl2++; print L2 $query."#E21#INcomplete_tilHPV\n";
                        print COMBI $query."#E21#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E11") {
                        $inl3++; print L3 $query."#E11#INcomplete_tilHPV\n";
                        print COMBI $query."#E11#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E12") {
                        $inl4++; print L4 $query."#E12#INcomplete_tilHPV\n";
                        print COMBI $query."#E12#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E13") {
                        $inl5++; print L5 $query."#E13#INcomplete_tilHPV\n";
                        print COMBI $query."#E13#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E14") {
                        $inl6++; print L6 $query."#E14#INcomplete_tilHPV\n";
                        print COMBI $query."#E14#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E15") {
                        $inl7++; print L7 $query."#E15#INcomplete_tilHPV\n";
                        print COMBI $query."#E15#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E16") {
                        $inl8++; print L8 $query."#E16#INcomplete_tilHPV\n";
                        print COMBI $query."#E16#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E18") {
                        $inh1++; print H1 $query."#E18#INcomplete_tilHPV\n";
                        print COMBI $query."#E18#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E19") {
                        $inh2++; print H2 $query."#E19#INcomplete_tilHPV\n";
                        print COMBI $query."#E19#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E20") {
                        $inh3++; print H3 $query."#E20#INcomplete_tilHPV\n";
                        print COMBI $query."#E20#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E22") {
                        $inh4++; print H4 $query."#E22#INcomplete_tilHPV\n";
                        print COMBI $query."#E22#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E23") {
                        $inh5++; print H5 $query."#E23#INcomplete_tilHPV\n";
                        print COMBI $query."#E23#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E24") {
                        $inh6++; print H6 $query."#E24#INcomplete_tilHPV\n";
                        print COMBI $query."#E24#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E25") {
                        $inh7++; print H7 $query."#E25#INcomplete_tilHPV\n";
                        print COMBI $query."#E25#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E26") {
                        $inh8++; print H8 $query."#E26#INcomplete_tilHPV\n";
                        print COMBI $query."#E26#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E27") {
                        $ink1++; print K1 $query."#E27#INcomplete_tilHPV\n";
                        print COMBI $query."#E27#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E28") {
                        $ink2++; print K2 $query."#E28#INcomplete_tilHPV\n";
                        print COMBI $query."#E28#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E29") {
                        $ink3++; print K3 $query."#E29#INcomplete_tilHPV\n";
                        print COMBI $query."#E29#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E30") {
                        $ink4++; print K4 $query."#E30#INcomplete_tilHPV\n";
                        print COMBI $query."#E30#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E31") {
                        $ink5++; print K5 $query."#E31#INcomplete_tilHPV\n";
                        print COMBI $query."#E31#INcomplete_tilHPV\n";
                } elsif ($b1_1 =~ "E32") {
                        $ink6++; print K6 $query."#E32#INcomplete_tilHPV\n";
                        print COMBI $query."#E32#INcomplete_tilHPV\n";
                } else { next; }

        # 'imperfect' & incomplete match  (CASES B1/2, C1/2)
        } elsif ( ($b1_6 =~ $plus) && (int($b1_4) >= 90)
                && (int($b1_3) < int($b1_2))
                && (int($b1_7) != 1)
                && ((int($b1_3)+int($b1_7)) > 20)
                && ((int($b1_2)-(int($b1_3)+int($b1_7))) < 5) ) {

                if ($b1_1 =~ "E17") {
                        $ine1++; print E1 $query."#E17#INcomplete_middle\n";
                        print COMBI $query."#E17#INcomplete_middle\n";
                } elsif ($b1_1 =~ "E02") {
                        $ine2++; print E2 $query."#E02#INcomplete_middle\n";
                        print COMBI $query."#E02#INcomplete_middle\n";
                } elsif ($b1_1 =~ "E03") {
                        $ine3++; print E3 $query."#E03#INcomplete_middle\n";
                        print COMBI $query."#E03#INcomplete_middle\n";
                } elsif ($b1_1 =~ "E04") {
                        $ine4++; print E4 $query."#E04#INcomplete_middle\n";
                        print COMBI $query."#E04#INcomplete_middle\n";
                } elsif ($b1_1 =~ "E05") {
                        $ine5++; print E5 $query."#E05#INcomplete_middle\n";
                        print COMBI $query."#E05#INcomplete_middle\n";
                } elsif ($b1_1 =~ "E06") {
                        $ine6++; print E6 $query."#E06#INcomplete_middle\n";
                        print COMBI $query."#E06#INcomplete_middle\n";
                } elsif ($b1_1 =~ "E07") {
                        $ine7++; print E7 $query."#E07#INcomplete_middle\n";
                        print COMBI $query."#E07#INcomplete_middle\n";
                } elsif ($b1_1 =~ "E08") {
                        $ine8++; print E8 $query."#E08#INcomplete_middle\n";
                        print COMBI $query."#E08#INcomplete_middle\n";
                } elsif ($b1_1 =~ "E09") {
                        $inl1++; print L1 $query."#E09#INcomplete_middle\n";
                        print COMBI $query."#E09#INcomplete_middle\n";
                } elsif ($b1_1 =~ "E21") {
                        $inl2++; print L2 $query."#E21#INcomplete_middle\n";
                        print COMBI $query."#E21#INcomplete_middle\n";
```

```
            } elsif ($b1_1 =~ "E11") {
                        $inl3++; print L3 $query."#E11#INcomplete_middle\n";
                        print COMBI $query."#E11#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E12") {
                        $inl4++; print L4 $query."#E12#INcomplete_middle\n";
                        print COMBI $query."#E12#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E13") {
                        $inl5++; print L5 $query."#E13#INcomplete_middle\n";
                        print COMBI $query."#E13#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E14") {
                        $inl6++; print L6 $query."#E14#INcomplete_middle\n";
                        print COMBI $query."#E14#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E15") {
                        $inl7++; print L7 $query."#E15#INcomplete_middle\n";
                        print COMBI $query."#E15#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E16") {
                        $inl8++; print L8 $query."#E16#INcomplete_middle\n";
                        print COMBI $query."#E16#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E18") {
                        $inh1++; print H1 $query."#E18#INcomplete_middle\n";
                        print COMBI $query."#E18#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E19") {
                        $inh2++; print H2 $query."#E19#INcomplete_middle\n";
                        print COMBI $query."#E19#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E20") {
                        $inh3++; print H3 $query."#E20#INcomplete_middle\n";
                        print COMBI $query."#E20#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E22") {
                        $inh4++; print H4 $query."#E22#INcomplete_middle\n";
                        print COMBI $query."#E22#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E23") {
                        $inh5++; print H5 $query."#E23#INcomplete_middle\n";
                        print COMBI $query."#E23#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E24") {
                        $inh6++; print H6 $query."#E24#INcomplete_middle\n";
                        print COMBI $query."#E24#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E25") {
                        $inh7++; print H7 $query."#E25#INcomplete_middle\n";
                        print COMBI $query."#E25#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E26") {
                        $inh8++; print H8 $query."#E26#INcomplete_middle\n";
                        print COMBI $query."#E26#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E27") {
                        $ink1++; print K1 $query."#E27#INcomplete_middle\n";
                        print COMBI $query."#E27#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E28") {
                        $ink2++; print K2 $query."#E28#INcomplete_middle\n";
                        print COMBI $query."#E28#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E29") {
                        $ink3++; print K3 $query."#E29#INcomplete_middle\n";
                        print COMBI $query."#E29#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E30") {
                        $ink4++; print K4 $query."#E30#INcomplete_middle\n";
                        print COMBI $query."#E30#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E31") {
                        $ink5++; print K5 $query."#E31#INcomplete_middle\n";
                        print COMBI $query."#E31#INcomplete_middle\n";
            } elsif ($b1_1 =~ "E32") {
                        $ink6++; print K6 $query."#E32#INcomplete_middle\n";
                        print COMBI $query."#E32#INcomplete_middle\n";
            } else { next; }

} elsif ( (($b1_6 =~ $plus) && (int($b1_4) < 90)) |
            ($b1_6 =~ $minus) ) {

            $hitOutranged++;

            if ($b1_1 =~ "E17") {
                        $ne1++; print E1 $query."#E17#OUT\n"; print COMBI $query."#E17#OUT\n";
            } elsif ($b1_1 =~ "E02") {
                        $ne2++; print E2 $query."#E02#OUT\n"; print COMBI $query."#E02#OUT\n";
            } elsif ($b1_1 =~ "E03") {
                        $ne3++; print E3 $query."#E03#OUT\n"; print COMBI $query."#E03#OUT\n";
            } elsif ($b1_1 =~ "E04") {
                        $ne4++; print E4 $query."#E04#OUT\n"; print COMBI $query."#E04#OUT\n";
            } elsif ($b1_1 =~ "E05") {
                        $ne5++; print E5 $query."#E05#OUT\n"; print COMBI $query."#E05#OUT\n";
            } elsif ($b1_1 =~ "E06") {
                        $ne6++; print E6 $query."#E06#OUT\n"; print COMBI $query."#E06#OUT\n";
            } elsif ($b1_1 =~ "E07") {
                        $ne7++; print E7 $query."#E07#OUT\n"; print COMBI $query."#E07#OUT\n";
            } elsif ($b1_1 =~ "E08") {
                        $ne8++; print E8 $query."#E08#OUT\n"; print COMBI $query."#E08#OUT\n";
            } elsif ($b1_1 =~ "E09") {
                        $nl1++; print L1 $query."#E09#OUT\n"; print COMBI $query."#E09#OUT\n";
            } elsif ($b1_1 =~ "E21") {
                        $nl2++; print L2 $query."#E21#OUT\n"; print COMBI $query."#E21#OUT\n";
            } elsif ($b1_1 =~ "E11") {
                        $nl3++; print L3 $query."#E11#OUT\n"; print COMBI $query."#E11#OUT\n";
            } elsif ($b1_1 =~ "E12") {
                        $nl4++; print L4 $query."#E12#OUT\n"; print COMBI $query."#E12#OUT\n";
            } elsif ($b1_1 =~ "E13") {
                        $nl5++; print L5 $query."#E13#OUT\n"; print COMBI $query."#E13#OUT\n";
            } elsif ($b1_1 =~ "E14") {
                        $nl6++; print L6 $query."#E14#OUT\n"; print COMBI $query."#E14#OUT\n";
            } elsif ($b1_1 =~ "E15") {
                        $nl7++; print L7 $query."#E15#OUT\n"; print COMBI $query."#E15#OUT\n";
            } elsif ($b1_1 =~ "E16") {
                        $nl8++; print L8 $query."#E16#OUT\n"; print COMBI $query."#E16#OUT\n";
            } elsif ($b1_1 =~ "E18") {
                        $nh1++; print H1 $query."#E18#OUT\n"; print COMBI $query."#E18#OUT\n";
            } elsif ($b1_1 =~ "E19") {
                        $nh2++; print H2 $query."#E19#OUT\n"; print COMBI $query."#E19#OUT\n";
            } elsif ($b1_1 =~ "E20") {
                        $nh3++; print H3 $query."#E20#OUT\n"; print COMBI $query."#E20#OUT\n";
            } elsif ($b1_1 =~ "E22") {
                        $nh4++; print H4 $query."#E22#OUT\n"; print COMBI $query."#E22#OUT\n";
            } elsif ($b1_1 =~ "E23") {
                        $nh5++; print H5 $query."#E23#OUT\n"; print COMBI $query."#E23#OUT\n";
            } elsif ($b1_1 =~ "E24") {
                        $nh6++; print H6 $query."#E24#OUT\n"; print COMBI $query."#E24#OUT\n";
            } elsif ($b1_1 =~ "E25") {
                        $nh7++; print H7 $query."#E25#OUT\n"; print COMBI $query."#E25#OUT\n";
            } elsif ($b1_1 =~ "E26") {
                        $nh8++; print H8 $query."#E26#OUT\n"; print COMBI $query."#E26#OUT\n";
            } elsif ($b1_1 =~ "E27") {
                        $nk1++; print K1 $query."#E27#OUT\n"; print COMBI $query."#E27#OUT\n";
            } elsif ($b1_1 =~ "E28") {
                        $nk2++; print K2 $query."#E28#OUT\n"; print COMBI $query."#E28#OUT\n";
            } elsif ($b1_1 =~ "E29") {
                        $nk3++; print K3 $query."#E29#OUT\n"; print COMBI $query."#E29#OUT\n";
            } elsif ($b1_1 =~ "E30") {
                        $nk4++; print K4 $query."#E30#OUT\n"; print COMBI $query."#E30#OUT\n";
            } elsif ($b1_1 =~ "E31") {
                        $nk5++; print K5 $query."#E31#OUT\n"; print COMBI $query."#E31#OUT\n";
            } elsif ($b1_1 =~ "E32") {
                        $nk6++; print K6 $query."#E32#OUT\n"; print COMBI $query."#E32#OUT\n";
            } else { next; }

} else {

            $hitOutranged++;

            if ($b1_1 =~ "E17") {
                        $ne1++; print E1 $query."#E17#OUT_ranged\n";
                        print COMBI $query."#E17#OUT_ranged\n";
            } elsif ($b1_1 =~ "E02") {
                        $ne2++; print E2 $query."#E02#OUT_ranged\n";
                        print COMBI $query."#E02#OUT_ranged\n";
            } elsif ($b1_1 =~ "E03") {
                        $ne3++; print E3 $query."#E03#OUT_ranged\n";
                        print COMBI $query."#E03#OUT_ranged\n";
            } elsif ($b1_1 =~ "E04") {
                        $ne4++; print E4 $query."#E04#OUT_ranged\n";
                        print COMBI $query."#E04#OUT_ranged\n";
            } elsif ($b1_1 =~ "E05") {
                        $ne5++; print E5 $query."#E05#OUT_ranged\n";
                        print COMBI $query."#E05#OUT_ranged\n";
            } elsif ($b1_1 =~ "E06") {
                        $ne6++; print E6 $query."#E06#OUT_ranged\n";
                        print COMBI $query."#E06#OUT_ranged\n";
            } elsif ($b1_1 =~ "E07") {
                        $ne7++; print E7 $query."#E07#OUT_ranged\n";
                        print COMBI $query."#E07#OUT_ranged\n";
            } elsif ($b1_1 =~ "E08") {
                        $ne8++; print E8 $query."#E08#OUT_ranged\n";
                        print COMBI $query."#E08#OUT_ranged\n";
```

```
                        } elsif ($b1_1 =~ "E09") {
                                $nl1++; print L1 $query."#E09#OUT_ranged\n";
                                print COMBI $query."#E09#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E21") {
                                $nl2++; print L2 $query."#E21#OUT_ranged\n";
                                print COMBI $query."#E21#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E11") {
                                $nl3++; print L3 $query."#E11#OUT_ranged\n";
                                print COMBI $query."#E11#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E12") {
                                $nl4++; print L4 $query."#E12#OUT_ranged\n";
                                print COMBI $query."#E12#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E13") {
                                $nl5++; print L5 $query."#E13#OUT_ranged\n";
                                print COMBI $query."#E13#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E14") {
                                $nl6++; print L6 $query."#E14#OUT_ranged\n";
                                print COMBI $query."#E14#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E15") {
                                $nl7++; print L7 $query."#E15#OUT_ranged\n";
                                print COMBI $query."#E15#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E16") {
                                $nl8++; print L8 $query."#E16#OUT_ranged\n";
                                print COMBI $query."#E16#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E18") {
                                $nh1++; print H1 $query."#E18#OUT_ranged\n";
                                print COMBI $query."#E18#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E19") {
                                $nh2++; print H2 $query."#E19#OUT_ranged\n";
                                print COMBI $query."#E19#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E20") {
                                $nh3++; print H3 $query."#E20#OUT_ranged\n";
                                print COMBI $query."#E20#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E22") {
                                $nh4++; print H4 $query."#E22#OUT_ranged\n";
                                print COMBI $query."#E22#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E23") {
                                $nh5++; print H5 $query."#E23#OUT_ranged\n";
                                print COMBI $query."#E23#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E24") {
                                $nh6++; print H6 $query."#E24#OUT_ranged\n";
                                print COMBI $query."#E24#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E25") {
                                $nh7++; print H7 $query."#E25#OUT_ranged\n";
                                print COMBI $query."#E25#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E26") {
                                $nh8++; print H8 $query."#E26#OUT_ranged\n";
                                print COMBI $query."#E26#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E27") {
                                $nk1++; print K1 $query."#E27#OUT_ranged\n";
                                print COMBI $query."#E27#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E28") {
                                $nk2++; print K2 $query."#E28#OUT_ranged\n";
                                print COMBI $query."#E28#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E29") {
                                $nk3++; print K3 $query."#E29#OUT_ranged\n";
                                print COMBI $query."#E29#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E30") {
                                $nk4++; print K4 $query."#E30#OUT_ranged\n";
                                print COMBI $query."#E30#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E31") {
                                $nk5++; print K5 $query."#E31#OUT_ranged\n";
                                print COMBI $query."#E31#OUT_ranged\n";
                        } elsif ($b1_1 =~ "E32") {
                                $nk6++; print K6 $query."#E32#OUT_ranged\n";
                                print COMBI $query."#E32#OUT_ranged\n";
                        } else { next; }
                }
   } else { next; }
}
close E1;close E2;close E3;close E4;close E5;close E6;close E7;close E8;
close L1;close L2;close L3;close L4;close L5;close L6;close L7;close L8;
close H1;close H2;close H3;close H4;close H5;close H6;close H7;close H8;
close K1;close K2;close K3;close K4;close K5;close K6;close NOHITREP; close COMBI;

unless (open (OUT, ">$out")) { print "\n\n\n\nCan not create ".$out."\n\n\n\n"; exit;}
my $txt1 = " : Complete vs INcomplete vs Outranged :\n\t";
my $txt2 = "\t ==> Significant= ";
```

```
          my $earlySum = $e1+$e2+$e3+$e4+$e5+$e6+$e7+$e8+
                         $ine1+$ine2+$ine3+$ine4+$ine5+$ine6+$ine7+$ine8;
          my $lateSum = $l1+$l2+$l3+$l4+$l5+$l6+$l7+$l8+
                         $inl1+$inl2+$inl3+$inl4+$inl5+$inl6+$inl7+$inl8;
          my $set3sum = $h1+$h3+$h5+$h7+$k1+$k3+$k5+
                         $inh1+$inh3+$inh5+$inh7+$ink1+$ink3+$ink5;
          my $set4sum = $h2+$h4+$h6+$h8+$k2+$k4+$k6+
                         $inh2+$inh4+$inh6+$inh8+$ink2+$ink4+$ink6;
          print OUT "Total number of sequences : ".$total."\n\n\n";
          print OUT "For this sample, the number of sequences with primers:\n";
          print OUT "\n\nEARLY primers SET 1 (Significant = ".$earlySum." )\n";
          print OUT "E17".$txt1.$e1."\t : ".$ine1."\t : ".$ne1.$txt2.(int($e1)+int($ine1))."\n";
          print OUT "E2 ".$txt1.$e2."\t : ".$ine2."\t : ".$ne2.$txt2.(int($e2)+int($ine2))."\n";
          print OUT "E3 ".$txt1.$e3."\t : ".$ine3."\t : ".$ne3.$txt2.(int($e3)+int($ine3))."\n";
          print OUT "E4 ".$txt1.$e4."\t : ".$ine4."\t : ".$ne4.$txt2.(int($e4)+int($ine4))."\n";
          print OUT "E5 ".$txt1.$e5."\t : ".$ine5."\t : ".$ne5.$txt2.(int($e5)+int($ine5))."\n";
          print OUT "E6 ".$txt1.$e6."\t : ".$ine6."\t : ".$ne6.$txt2.(int($e6)+int($ine6))."\n";
          print OUT "E7 ".$txt1.$e7."\t : ".$ine7."\t : ".$ne7.$txt2.(int($e7)+int($ine7))."\n";
          print OUT "E8 ".$txt1.$e8."\t : ".$ine8."\t : ".$ne8.$txt2.(int($e8)+int($ine8))."\n";
          print OUT "\n\nEARLY primers SET 2 (Significant = ".$lateSum." )\n";
          print OUT "E9 ".$txt1.$l1."\t : ".$inl1."\t : ".$nl1.$txt2.(int($l1)+int($inl1))."\n";
          print OUT "E21".$txt1.$l2."\t : ".$inl2."\t : ".$nl2.$txt2.(int($l2)+int($inl2))."\n";
          print OUT "E11".$txt1.$l3."\t : ".$inl3."\t : ".$nl3.$txt2.(int($l3)+int($inl3))."\n";
          print OUT "E12".$txt1.$l4."\t : ".$inl4."\t : ".$nl4.$txt2.(int($l4)+int($inl4))."\n";
          print OUT "E13".$txt1.$l5."\t : ".$inl5."\t : ".$nl5.$txt2.(int($l5)+int($inl5))."\n";
          print OUT "E14".$txt1.$l6."\t : ".$inl6."\t : ".$nl6.$txt2.(int($l6)+int($inl6))."\n";
          print OUT "E15".$txt1.$l7."\t : ".$inl7."\t : ".$nl7.$txt2.(int($l7)+int($inl7))."\n";
          print OUT "E16".$txt1.$l8."\t : ".$inl8."\t : ".$nl8.$txt2.(int($l8)+int($inl8))."\n";
          print OUT "\n\nEARLY primers SET 3 (Significant = ".$set3sum." )\n";
          print OUT "E18".$txt1.$h1."\t : ".$inh1."\t : ".$nh1.$txt2.(int($h1)+int($inh1))."\n";
          print OUT "E20".$txt1.$h3."\t : ".$inh3."\t : ".$nh3.$txt2.(int($h3)+int($inh3))."\n";
          print OUT "E23".$txt1.$h5."\t : ".$inh5."\t : ".$nh5.$txt2.(int($h5)+int($inh5))."\n";
          print OUT "E25".$txt1.$h7."\t : ".$inh7."\t : ".$nh7.$txt2.(int($h7)+int($inh7))."\n";
          print OUT "E27".$txt1.$k1."\t : ".$ink1."\t : ".$nk1.$txt2.(int($k1)+int($ink1))."\n";
          print OUT "E29".$txt1.$k3."\t : ".$ink3."\t : ".$nk3.$txt2.(int($k3)+int($ink3))."\n";
          print OUT "E31".$txt1.$k5."\t : ".$ink5."\t : ".$nk5.$txt2.(int($k5)+int($ink5))."\n";
          print OUT "\n\nEARLY primers SET 4 (Significant = ".$set4sum." )\n";
          print OUT "E19".$txt1.$h2."\t : ".$inh2."\t : ".$nh2.$txt2.(int($h2)+int($inh2))."\n";
          print OUT "E22".$txt1.$h4."\t : ".$inh4."\t : ".$nh4.$txt2.(int($h4)+int($inh4))."\n";
          print OUT "E24".$txt1.$h6."\t : ".$inh6."\t : ".$nh6.$txt2.(int($h6)+int($inh6))."\n";
          print OUT "E26".$txt1.$h8."\t : ".$inh8."\t : ".$nh8.$txt2.(int($h8)+int($inh8))."\n";
          print OUT "E28".$txt1.$k2."\t : ".$ink2."\t : ".$nk2.$txt2.(int($k2)+int($ink2))."\n";
          print OUT "E30".$txt1.$k4."\t : ".$ink4."\t : ".$nk4.$txt2.(int($k4)+int($ink4))."\n";
          print OUT "E32".$txt1.$k6."\t : ".$ink6."\t : ".$nk6.$txt2.(int($k6)+int($ink6))."\n";
          print OUT "\n\nThe number of sequences WITHOUT HIT with primers : ";
          print OUT $nohitsum."\n";
          print OUT "\n\nThe number of sequences WITH HIT BUT NOT SIGNIFICANT : ";
          print OUT $hitOutranged."\n";
          close OUT;
}

# for Part 4444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444
sub getFastAprimer {
          my ($name, $seq, $input) = @_;
          use strict;
          use warnings;
          my $suffix = "_onlySig_fasta";
          unless (open (FA, ">$input$suffix")) {
             print "\n\n\n\nCan not create ".$input.$suffix."\n\n\n\n\n"; exit;
          }
          my @in = getFileData ($input);
          my $count = scalar (@in);
          for (my $p = 0; $p < $count ; $p++) {
             my @line = split (/#/, $in[$p]);
             if (($line[2]=~/^complete(.*)/) | ($line[2]=~/^INcom(.*)/)) {
                          print FA $$name[$line[0]];
                          print FA $$seq[$line[0]];
             } else { next; }
          }
          close FA;
}

#xx Sub-routines END HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
###############################################################################################

###########################################################################################################################
```

File name: **set2_p4mac.pl**

Source code:

```
#!/usr/bin/perl

### About the script ###################################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
#######################################################################################
### MAIN PROGRAM BODY ##################################################################
#######################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;

my $prefix_path= $sampleNo[2];
my $pyroNum = int($sampleNo[1]);

my $path = $prefix_path."Pyro".$pyroNum."_result/";

my $blastFilePath = $path.$sampleNo[0]."/blastHPV16/";
my $workDir = $path.$sampleNo[0]."/";

my $numberOfseqFile = $workDir."numberOfsequences.txt";

my $outfile = $blastFilePath."parsed_hpv_allreport";
my $outstat = $blastFilePath."parsed_hpv_statistics";

print "\n\nPROGRAM 4 (PARSING all) STARTS.....................................";

my @totSeq = getFileData ($numberOfseqFile);
my $count  = int($totSeq[0]);

getParseDouble2 ($blastFilePath, $count, $outfile, $outstat);

print "FINISHED\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
#######################################################################################

#######################################################################################
###    SUBROUTINES    #################################################################
#######################################################################################

sub getFileData {
          my ($filename) = @_;
          use strict;
          use warnings;

          my @filedata = ();
          unless ( open(FILE_DATA, $filename) ) {
             print STDERR "\n\nThe program can't open the files \"$filename\"\n\n\n\n";
             print "Please re-check input file name and its location\n";
             print "The correct command should be :\n\n";
             print "\t\tperl PROGRAM(location+name) INPUTFastAFILE(location+name)\n\n";
             print "The command should be given under directory";
             print " c:/Perl/bin> in case of Dos Terminal\n\n\n";
             exit;
          }
          @filedata = <FILE_DATA>;
          close FILE_DATA;
          return @filedata;
}

# for Part 1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
sub getParseDouble2 {
          my ($blastpath, $count, $out, $stat) = @_;
          use strict;
          use warnings;

          unless (open (OUT, ">$out")) {
             print "\n\n\n\nCan not create ".$out."\n\n\n\n";
             exit;
          }

          my ($nohitcount, $hitcount) = (0, 0);

          for (my $p = 0; $p < $count; $p++) {

             my @blast = getFileData ($blastpath."blasted_".$p);

             $blast[9] =~ m/^.........\(([0-9]*)(.*)/;

             # Query Length
             my $QL = $1;

             if ($blast[16] =~ m/^.......No.hits.found/) {

                          $nohitcount++;
                          print OUT $p."#".$QL."#nohitHPV#noHPV\n";

             } else {
                          $hitcount++;
                          print OUT $p."#".$QL."#hitHPV";
                          for (my $m = 0; $m < (scalar @blast); $m++) {

                                       if ($blast[$m] =~ m/^ Score(.*)/) {
                                                    my @tmpIden  = split (/\s+/, $blast[$m+1]);
                                                    #scoreBP
                                                    my $scoreBox = $tmpIden[3];
                                                    $scoreBox =~ m/^([0-9]*)\/([0-9]*)$/;
                                                    my $scoreBP = $2;
                                                    #percent match
                                                    my $percentBox = $tmpIden[4];
                                                    $percentBox =~ m/^\(([0-9]*)%\)$/;
                                                    my $percent = $1;
                                                    #strand
                                                    $blast[$m+2] =~ m/^.Strand.=.Plus...(.*)$/;
                                                    my $strand = $1;
                                                    # first-Match on Query
                                                    $blast[$m+5] =~ m/^Query..([0-9]*)(.*)([0-9]*)(.*)$/;
                                                    my $firstMatchQue = $1;
                                                    # first-Match on HPV16 (can be last position in case MINUS strand)
                                                    $blast[$m+7] =~ m/^Sbjct..([0-9]*)(.*)([0-9]*)(.*)$/;
                                                    my $firstMatchHpv = $1;

                                                    print OUT "#".$scoreBP."%".$percent."%".
                                                      $firstMatchQue."%".$strand."%".$firstMatchHpv;

                                       } else { next; }
                          }

                          print OUT "\n";

             }

          }  # close for loop

          close OUT;

          unless (open (STAT, ">$stat")) {print "\n\n\n\nCan not create ".$stat."\n\n\n\n"; exit;}

          print STAT "Number of sequences WITH HIT to HPV16 (SIG & INSIGnificant) : ".$hitcount."\n";
          print STAT "Number of sequences WITH NO HIT to HPV16 (by BLASTN)        : ".$nohitcount."\n";
          print STAT "Number of total sequences having this barcode               : ".$count."\n";

          close STAT;

}

#xx Sub-routines END HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
#######################################################################################

#########################################################################################################################
```

File name: **set2_p5mac_a_noCutoff.pl**
Source code:

```
#!/usr/bin/perl

### About the script ##################################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
#######################################################################################
### MAIN PROGRAM BODY #################################################################
#######################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;

my $prefix_path= $sampleNo[2];
my $pyroNum = int($sampleNo[1]);

my $path = $prefix_path."Pyro".$pyroNum."_result/";

my $workingDir = $path.$sampleNo[0]."/";
my $blastFilePath = $workingDir."blastHPV16/";

# For different cutoff conditions, change these variables:
#        $newDir, $prefix

my $newDir = $workingDir."noCutoff/";
my $prefix = "noCutoff_";

my $selectRep = $prefix."select_report";
my $grp1 = $prefix."select_group1";
my $grp2 = $prefix."select_group2";
my $grp3 = $prefix."select_group3";
my $grp4 = $prefix."select_group4";
my $selectRepDub =  $prefix."select_report_doubleHPV";

my $reportallpos = $prefix."allreport_fullpieceposition";

my $cf2 = 94;           # CUTOFF PERCENT MATCH FOR 2nd HIT!
my $cf3 = 15;           # CUTOFF SCOREBP FOR 2nd HIT!

############## Start of PART 1 : SELECT into 4 groups   ################################
my $parsedFile = $blastFilePath."parsed_hpv_allreport";

my @allParseRep = ();

print "\n\nPROGRAM 5A (SELECT NO CUTOFF) STARTS.....................................\n\n";
print "\n\n     This program arrange each sequence in this sample group into \n\n";
print "\n\n     4 'significant&/nonsignificant HPV16 hit' groups (because no CF is used)\n\n";
print "\n\n     according to its 'non-HPV16' part\n\n";
print "\n\n     G1 => the non-HPV is >= 150 nt\n\n";
print "\n\n     G2 => the non-HPV is >= 100 AND < 150 nt\n\n";
print "\n\n     G3 => the non-HPV is >= 50  AND < 100 nt\n\n";
print "\n\n     G4 => the non-HPV is >= 0   AND < 50  nt\n\n";

# read parsed data for selection
@allParseRep = getFileData($parsedFile);

# make Dir for selected result of 'no cut-off'-program
system ("mkdir", $newDir);

unless (open (SELECT, ">$newDir$selectRep")) {print "\n\nCan not create ".$newDir.$selectRep."\n\n"; exit;}
unless (open (GRP1, ">$newDir$grp1")) {print "\n\nCan not create ".$newDir.$grp1."\n\n"; exit;}
unless (open (GRP2, ">$newDir$grp2")) {print "\n\nCan not create ".$newDir.$grp2."\n\n"; exit;}
unless (open (GRP3, ">$newDir$grp3")) {print "\n\nCan not create ".$newDir.$grp3."\n\n\n\n"; exit;}
unless (open (GRP4, ">$newDir$grp4")) {print "\n\nCan not create ".$newDir.$grp4."\n\n\n\n"; exit;}
unless (open (DUB, ">$newDir$selectRepDub")){print "\nCan not create ".$newDir.$selectRepDub."\n\n "; exit;}
unless (open (RALL,">$newDir$reportallpos")) {print "\nCan not create ".$newDir.$reportallpos."\n\n"; exit;}

for ( my $p = 0 ; $p < (scalar @allParseRep) ; $p++) {
          my @tmp_line = split (/#/, $allParseRep[$p]);
          my $Qnum = $tmp_line[0];                                  # query's seqNo
          my $QL   = $tmp_line[1];                                  # query's length
          my $temp_index = $tmp_line[2];                            # hit/nohit indice
          my $blockCount = scalar (@tmp_line);

          my ($temp_scoreBP, $temp_1stMatch, $temp_strand, $temp_1stMatchS, $use);
          my ($scoreBP2, $firstM2, $strand2, $firstMS2);
          my ($temp_lastpos);       # the position (non-hpv) next to the last hpv match position
          my ($scoreBP1, $firstM1, $strand1, $firstMS1);
          my ($series1, $series2);

          # This first IF-LOOP defies the 'virtual matching area of HPV16'
          #   The value of the 'first position' of HPV-matching area will be specified
          #   as well as other values, e.g. 'virtual scoreBP' in case of two continuous hits

          # ONE-hit only
          if (($temp_index =~ "hitHPV") && ($blockCount == 4)) {  # for ONE-HPV-HIT query
             my @tempBox = split ( /%/, $tmp_line[3]);
             $temp_scoreBP = $tempBox[0];                       # scoreBP
             $temp_1stMatch = $tempBox[2];                      # 1st-Match-position-on-QUERY-strand
             $temp_strand = $tempBox[3];                        # strand of the matched HPV
             $temp_1stMatchS = $tempBox[4];                     # 1st-Match-position-on-Sbjt-strand
             $use = "hitx1_useFIRST";

             $scoreBP1 = $tempBox[0];

             $firstM1 = $tempBox[2];
             $strand1 = $tempBox[3];
             $firstMS1 = $tempBox[4];
             $series1 = "use_Alone";

             $scoreBP2 = "none";

             $firstM2 = "none";
             $strand2 = "none";
             $firstMS2 = "none";
             $series2 = "none";

          # More than ONE hit, then two first hits will be taken into consideration
          } elsif (($temp_index =~ "hitHPV") && ($blockCount > 4)) {  # for MORE-HPV-HITs query
             my @tBox1 = split ( /%/, $tmp_line[3]);
             my $tscoreBP1 = $tBox1[0];                         # scoreBP of FIRST HIT
             my $tpercent1 = $tBox1[1];                         # percent match of FIRST HIT
             my $t1stMatch1 = $tBox1[2];         # 1st-Match-position-on-QUERY-strand of FIRST HIT
             my $tstrand1 = $tBox1[3];                          # strand of the matched HPV of FIRST HIT
             my $t1stMatchS1 = $tBox1[4];

             my @tBox2 = split ( /%/, $tmp_line[4]);
             my $tscoreBP2 = $tBox2[0];                         # scoreBP of SECOND HIT
             my $tpercent2 = $tBox2[1];                         # percent match of SECOND HIT
             my $t1stMatch2 = $tBox2[2];         # 1st-Match-position-on-QUERY-strand of SECOND HIT
             my $tstrand2 = $tBox2[3];                          # strand of the matched HPV of SECOND HIT
             my $t1stMatchS2 = $tBox2[4];

             # (1) : if ending-of-first-hit is EARLIER than ending-of-second-hit
             if ((int($t1stMatch1)+int($tscoreBP1)) < (int($t1stMatch2)+int($tscoreBP2))) {

                       # (1.1): first & second hits begin at the same pos
                       if ( int($t1stMatch1) == int($t1stMatch2) ) {
                                 $temp_scoreBP = $tscoreBP2;
                                 $temp_1stMatch = $t1stMatch2;
                                 $temp_strand = $tstrand2;
                                 $use = "hitx2_useSECOND_1stHitasSubset";
                                 $temp_1stMatchS = $t1stMatchS2;

                                 $scoreBP1 = $tscoreBP1;
                                 $firstM1 = $t1stMatch1;
                                 $strand1 = $tstrand1;
                                 $firstMS1 = int($t1stMatchS1);
                                 $series1 = "subset";

                                 $scoreBP2 = $tscoreBP2;
                                 $firstM2 = $t1stMatch2;
                                 $strand2 = $tstrand2;
                                 $firstMS2 = int($t1stMatchS2);
                                 $series2 = "use_Alone";

                       # (1.2): first hit begins first
                       } elsif ( int($t1stMatch1) < int($t1stMatch2) ) {
```

```
# If last match position of 1stHit is >= 1stMatch of 2.Hit (+2)
# (+2) is the allowance for the gap or if there is no gap then 2 nt distance
# between the two hits
if ( (int($t1stMatch1)+int($tscoreBP1)) >= (int($t1stMatch2)+2) ) {

            # Control if the 2.Hit is long enough to be significantly counted
            if ((int($tscoreBP2) >= $cf3 ) && (int($tpercent2) >= $cf2 )) {
                        $use = "hitx2_useFIRSTnSECOND_continuous";
                        # DoubleHPV-- independent of HPV16's 4 groups

                        print DUB $Qnum."#".$QL."#hitHPV#".$use."\n";
                        $temp_scoreBP = int($tscoreBP2)+(int($t1stMatch2)
                                                -int($t1stMatch1));
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;       # ignore second hit strand
                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "use_1st";

                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "use_2nd";
            } else {
                        $use = "hitx2_useFIRST_tooSmall2ndHit";
                        $temp_scoreBP = $tscoreBP1;
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "use_Alone";

                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "smallerThan_".$cf3;
            }
} else {
            if ((int($tscoreBP2) >= $cf3 ) && (int($tpercent2) >= $cf2 )) {
                        $use = "hitx2_useFIRSTnSECOND_discontinuous";
                        # DoubleHPV-- independent of HPV16's 4 groups

                        print DUB $Qnum."#".$QL."#hitHPV#".$use."\n";
                        $temp_scoreBP = $tscoreBP1; # DIFFERENT FROM "continuous"
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;
                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "use_1st";

                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "use_2nd_discont";
            } else {
                        $use = "hitx2_useFIRST_tooSmall2ndHit";
                        $temp_scoreBP = $tscoreBP1;
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                                                $strand1 = $tstrand1;
                                                $firstMS1 = int($t1stMatchS1);
                                                $series1 = "use_Alone";

                                                $scoreBP2 = $tscoreBP2;
                                                $firstM2 = $t1stMatch2;
                                                $strand2 = $tstrand2;
                                                $firstMS2 = int($t1stMatchS2);
                                                $series2 = "smallerThan_".$cf3;
                                    }
                        }

                # (1.3): second hit begins first
                } else {
                        $temp_scoreBP = $tscoreBP2;
                        $temp_1stMatch = $t1stMatch2;
                        $temp_strand = $tstrand2;
                        $use = "hitx2_useSECOND_1stHitasSubset";
                        $temp_1stMatchS = $t1stMatchS2;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "subset";

                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "use_Alone";
                }

        # (2) : if ending-of-first-hit is THE SAME as ending-of-second-hit
        } elsif ((int($t1stMatch1)+int($tscoreBP1))==(int($t1stMatch2)+int($tscoreBP2))) {

                # (2.1): first & second hits begin at the same pos
                if ( int($t1stMatch1) == int($t1stMatch2) ) {
                        $temp_scoreBP = $tscoreBP1;
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;
                        $use = "hitx2_useFIRST_2ndHitasSubset";
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "use_Alone";

                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "subset";

                # (2.2): first hit begins first
                } elsif ( int($t1stMatch1) < int($t1stMatch2) ) {
                        $temp_scoreBP = $tscoreBP1;
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;
                        $use = "hitx2_useFIRST_2ndHitasSubset";
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "use_Alone";

                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "subset";
```

```
            # (2.3): second hit begins first
            } else {
                        if ((int($tpercent1)>=int($tpercent2))&&((int($t1stMatch1)-int($t1stMatch2))<3)){
                                    $temp_scoreBP = $tscoreBP1;
                                    $temp_1stMatch = $t1stMatch1;
                                    $temp_strand = $tstrand1;
                                    $use = "hitx2_useFIRST_2ndHitasSubset";

                                    $temp_1stMatchS = $t1stMatchS1;
                                    $scoreBP1 = $tscoreBP1;
                                    $firstM1 = $t1stMatch1;
                                    $strand1 = $tstrand1;
                                    $firstMS1 = int($t1stMatchS1);
                                    $series1 = "use_Alone";

                                    $scoreBP2 = $tscoreBP2;
                                    $firstM2 = $t1stMatch2;
                                    $strand2 = $tstrand2;
                                    $firstMS2 = int($t1stMatchS2);
                                    $series2 = "subset";
                        } else {
                                    $temp_scoreBP = $tscoreBP2;
                                    $temp_1stMatch = $t1stMatch2;
                                    $temp_strand = $tstrand2;
                                    $use = "hitx2_useSECOND_1stHitasSubset";
                                    $temp_1stMatchS = $t1stMatchS2;

                                    $scoreBP1 = $tscoreBP1;
                                    $firstM1 = $t1stMatch1;
                                    $strand1 = $tstrand1;
                                    $firstMS1 = int($t1stMatchS1);
                                    $series1 = "subset";

                                    $scoreBP2 = $tscoreBP2;
                                    $firstM2 = $t1stMatch2;
                                    $strand2 = $tstrand2;
                                    $firstMS2 = int($t1stMatchS2);
                                    $series2 = "use_Alone";
                        }
            }
# (3) : if ending-of-first-hit is LATER than ending-of-second-hit
} else {

            # (3.1): first & second hits begin at the same pos
            if ( int($t1stMatch1) == int($t1stMatch2) ) {
                        $temp_scoreBP = $tscoreBP1;
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;
                        $use = "hitx2_useFIRST_2ndHitasSubset";
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "use_Alone";

                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "subset";

            # (3.2): first hit begins first
            } elsif ( int($t1stMatch1) < int($t1stMatch2) ) {
                        $temp_scoreBP = $tscoreBP1;
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;
                        $use = "hitx2_useFIRST_2ndHitasSubset";
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "use_Alone";

                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "subset";

            # (3.3): second hit begins first
            } else {

                        if ( (int($tscoreBP2) >= $cf3 ) && (int($tpercent2) >= $cf2 )) {
                                    if ( (int($t1stMatch2)+int($tscoreBP2)) >= (int($t1stMatch1)+2) ) {
                                                $use = "hitx2_useSECONDnFIRST_continuous";
                                                # DoubleHPV-- independent of HPV16's 4 groups

                                                print DUB $Qnum."#".$QL."#hitHPV#".$use."\n";
                                                $temp_scoreBP = int($tscoreBP1)+(int($t1stMatch1)
                                                                                -int($t1stMatch2));
                                                $temp_1stMatch = $t1stMatch2;
                                                $temp_strand = $tstrand2;
                                                $scoreBP2 = $tscoreBP2;
                                                $firstM2 = $t1stMatch2;
                                                $strand2 = $tstrand2;

                                                $temp_1stMatchS = $t1stMatchS2;
                                                $scoreBP1 = $tscoreBP1;
                                                $firstM1 = $t1stMatch1;
                                                $strand1 = $tstrand1;
                                                $firstMS1 = int($t1stMatchS1);
                                                $series1 = "use_2nd";

                                                $firstMS2 = int($t1stMatchS2);
                                                $series2 = "use_1st";

                                    } else {
                                                $use = "hitx2_useSECONDnFIRST_discontinuous";
                                                # DoubleHPV-- independent of HPV16's 4 groups
                                                print DUB $Qnum."#".$QL."#hitHPV#".$use."\n";
                                                $temp_scoreBP = $tscoreBP2;  # DIFFERENT FROM "continuous"
                                                $temp_1stMatch = $t1stMatch2;
                                                $temp_strand = $tstrand2;
                                                $scoreBP2 = $tscoreBP2;
                                                $firstM2 = $t1stMatch2;
                                                $strand2 = $tstrand2;

                                                $temp_1stMatchS = $t1stMatchS2;
                                                $scoreBP1 = $tscoreBP1;
                                                $firstM1 = $t1stMatch1;
                                                $strand1 = $tstrand1;
                                                $firstMS1 = int($t1stMatchS1);
                                                $series1 = "use_2nd_discont";

                                                $firstMS2 = int($t1stMatchS2);
                                                $series2 = "use_1st";
                                    }
                        } else {
                                    $temp_scoreBP = $tscoreBP1;
                                    $temp_1stMatch = $t1stMatch1;
                                    $temp_strand = $tstrand1;
                                    $use = "hitx2_useFIRST_tooSmall2ndHit";
                                    $temp_1stMatchS = $t1stMatchS1;

                                    $scoreBP1 = $tscoreBP1;
                                    $firstM1 = $t1stMatch1;
                                    $strand1 = $tstrand1;
                                    $firstMS1 = int($t1stMatchS1);
                                    $series1 = "use_Alone";

                                    $scoreBP2 = $tscoreBP2;
                                    $firstM2 = $t1stMatch2;
                                    $strand2 = $tstrand2;
                                    $firstMS2 = int($t1stMatchS2);
                                    $series2 = "smallerThan_".$cf3;
                        }
            }
      }

   } else {
                        # for no-hit
      $use = "noHPV";
   }
```

```
        # SECOND IF-LOOP to SORT each sequence (based on HPV16 matching values)
        #   into G1,G2,G3,G4

        if ($temp_index =~ "nohitHPV") {

           print SELECT $Qnum."#".$QL."#nohitHPV#noHPV#noHPV#noHPV#noHPV\n";
           print RALL $Qnum."#".$QL."#nohitHPV#noHPV#noHPV#noHPV#noHPV\n";

        } else {

           # must be CHANGED when there is CUTOFF
           # '+2' is used to make it the Last position of HPV match

           $temp_lastpos = int($temp_1stMatch)+int($temp_scoreBP);
           my $therest = int($QL) - int($temp_lastpos) + 2 ;

           my $firstMS = int($temp_1stMatchS);
           print RALL $Qnum."#".$QL."#".$temp_index."#".$temp_scoreBP."%".
                      $temp_1stMatch."%".$temp_strand."%".
                      $firstMS."%".$temp_lastpos."#".$use."#".
                      $series1."%".$scoreBP1."%".$firstM1."%".$strand1."%".int($firstMS1)."#".
                      $series2."%".$scoreBP2."%".$firstM2."%".$strand2."%".$firstMS2."\n";

           if ( int($therest) >=150 ) {
                      print SELECT $Qnum."#".$QL."#hitHPV#".$use."#G1\n"; print GRP1 $Qnum."\n";
           } elsif ( (100<=int($therest)) && (int($therest)<150) ) {
                      print SELECT $Qnum."#".$QL."#hitHPV#".$use."#G2\n"; print GRP2 $Qnum."\n";
           } elsif ( (50<=int($therest)) && (int($therest)<100)  ) {
                      print SELECT $Qnum."#".$QL."#hitHPV#".$use."#G3\n"; print GRP3 $Qnum."\n";
           } elsif ( int($therest)<50 )  {
                      print SELECT $Qnum."#".$QL."#hitHPV#".$use."#G4\n"; print GRP4 $Qnum."\n";
           } else {
                      print SELECT $Qnum."#".$QL."#hitHPV#".$use."#no_group\n";
           }
        }
}

close SELECT; close GRP1; close GRP2; close GRP3; close GRP4; close DUB; close RALL;

################ END of PART 1 : SELECT into 4 groups   ###################################

############## Start of PART 2+3 : Calling Sequence NAMES & STATISTICS ####################

my $seqNames = $workingDir."sample".$sampleNo[0]."_nameList";

my $seqNames_grp1 = $prefix."select_group1_names";
my $seqNames_grp2 = $prefix."select_group2_names";
my $seqNames_grp3 = $prefix."select_group3_names";
my $seqNames_grp4 = $prefix."select_group4_names";

my @seqNameFile = getFileData($seqNames);

getName3 (\@seqNameFile, $newDir, $grp1, $seqNames_grp1);
getName3 (\@seqNameFile, $newDir, $grp2, $seqNames_grp2);
getName3 (\@seqNameFile, $newDir, $grp3, $seqNames_grp3);
getName3 (\@seqNameFile, $newDir, $grp4, $seqNames_grp4);

my $statRep = $prefix."select_statistics";

getStat2a2 (\@seqNameFile, $newDir, $grp1, $grp2, $grp3, $grp4, $statRep);

################ End of PART 2+3 : Calling Sequence NAMES & STATISTICS ####################

print "\n\nPROGRAM 5A (SELECT NO CUTOFF).................................FINISHED\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
##########################################################################################

##########################################################################################
###     SUBROUTINES     ##################################################################
##########################################################################################

sub getFileData {
          my ($filename) = @_;
          use strict;
          use warnings;

          my @filedata = ();

          unless ( open(FILE_DATA, $filename) ) {

             print STDERR "\n\nThe program can't open the files \"$filename\"\n\n\n\n";
             print "Please re-check input file name and its location\n";
             print "The correct command should be :\n\n";
             print "\t\tperl PROGRAM(location+name) INPUTFastAFILE(location+name)\n\n";
             print "The command should be given under directory";
             print " c:/Perl/bin> in case of Dos Terminal\n\n\n";
             exit;
          }

          @filedata = <FILE_DATA>;
          close FILE_DATA;
          return @filedata;
}

# for PART 2222222222222222222222222222222222222222222222222222222222222222222222222222222222222222

sub getName3 {
          my ($seqNameFile, $dir, $in, $out) = @_;
          use strict;
          use warnings;

          unless (open (OUT, ">$dir$out")) {print "\n\nCan not create ".$dir.$out."\n\n\n\n\n"; exit;}

          my $loc_in = $dir.$in;
          my @input =  getFileData ($loc_in);

          for ( my $p = 0; $p < (scalar @input) ; $p++) {
             if ( $input[$p] =~ m/^([0-9]*)(.*)$/) {
                        my $seqNum = $1;
                        print OUT $seqNum."#".$$seqNameFile[$seqNum];
             }
          }
          close OUT;
}

sub getStat2a2 {
          my ($seqNameFile, $dir, $in1, $in2, $in3, $in4, $out) = @_;
          use strict;
          use warnings;

          unless (open (OUT, ">$dir$out")) {print "\n\nCan not create ".$dir.$out."\n\n\n\n\n"; exit;}

          my $in1_loc = $dir.$in1;
          my $in2_loc = $dir.$in2;
          my $in3_loc = $dir.$in3;
          my $in4_loc = $dir.$in4;

          my @in1_dt = getFileData ($in1_loc);   my $noG1 = (scalar @in1_dt);
          my @in2_dt = getFileData ($in2_loc);   my $noG2 = (scalar @in2_dt);
          my @in3_dt = getFileData ($in3_loc);   my $noG3 = (scalar @in3_dt);
          my @in4_dt = getFileData ($in4_loc);   my $noG4 = (scalar @in4_dt);

          my $noTotalSeq = (scalar @$seqNameFile);
          my $noUnclass  = $noTotalSeq - $noG1 - $noG2 - $noG3 - $noG4;

          print OUT "Total number of sequences having this barcode is : ".$noTotalSeq."\n";
          print OUT "Number of sequences in group 1           : ".$noG1."\n";
          print OUT "Number of sequences in group 2           : ".$noG2."\n";
          print OUT "Number of sequences in group 3           : ".$noG3."\n";
          print OUT "Number of sequences in group 4           : ".$noG4."\n";
          print OUT "Number of sequences with NO-HIT to HPV16 : ".$noUnclass."\n";

          close OUT;
}

#xx Sub-routines END HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
##########################################################################################

##########################################################################################
```

File name: **set2_p5mac_d_28bpCutoff.pl**
Source code:

```
#!/usr/bin/perl

### About the script ##################################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
#######################################################################################
### MAIN PROGRAM BODY #################################################################
#######################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;
my $prefix_path= $sampleNo[2];
my $pyroNum = int($sampleNo[1]);

my $path = $prefix_path."Pyro".$pyroNum."_result/";
my $workingDir = $path.$sampleNo[0]."/";
my $blastFilePath = $workingDir."blastHPV16/";

# For different cutoff conditions, change these variables:
#         $newDir, $prefix, $cf

my $newDir = $workingDir."28bpCutoff/";
my $prefix = "28bpCutoff_";

my $selectRep = $prefix."select_report";
my $grp1 = $prefix."select_group1";
my $grp2 = $prefix."select_group2";
my $grp3 = $prefix."select_group3";
my $grp4 = $prefix."select_group4";
my $selectRepDub =  $prefix."select_report_doubleHPV";

my $reportallpos = $prefix."allreport_fullpieceposition";

my $cf = 28;                        # CUT-OFF VALUE
my $cf2 = 94;                       # CUTOFF PERCENT MATCH FOR 2nd HIT!!!!
my $cf3 = 15;                       # CUTOFF SCOREBP FOR 2nd HIT!!!!

############## Start of PART 1 : SELECT into 4 groups   ################################

my $parsedFile = $blastFilePath."parsed_hpv_allreport";

my @allParseRep = ();

print "\n\nPROGRAM 5D (SELECT 28bp CUTOFF) STARTS.....................................\n\n";
print "\n\n     This program arrange each sequence in this sample group into \n\n";
print "\n     4 'significantHPV16 hit' groups (using CF=28 nt)\n\n";
print "\n     according to its 'non-HPV16' part\n\n";
print "     G1 => the non-HPV is >= 150 nt\n";
print "     G2 => the non-HPV is >= 100 AND < 150 nt\n";
print "     G3 => the non-HPV is >= 50  AND < 100 nt\n";
print "     G4 => the non-HPV is >= 0   AND < 50  nt\n";

# read parsed data for selection
@allParseRep = getFileData($parsedFile);

# make Dir for selected result of 'no cut-off'-program
system ("mkdir", $newDir);

unless (open (SELECT, ">$newDir$selectRep")) {print "\n\nCan not create ".$newDir.$selectRep."\n\n"; exit;}
unless (open (GRP1, ">$newDir$grp1")) {print "\n\nCan not create ".$newDir.$grp1."\n\n\n\n"; exit;}
unless (open (GRP2, ">$newDir$grp2")) {print "\n\nCan not create ".$newDir.$grp2."\n\n\n\n"; exit;}
unless (open (GRP3, ">$newDir$grp3")) {print "\n\nCan not create ".$newDir.$grp3."\n\n\n\n"; exit;}
unless (open (GRP4, ">$newDir$grp4")) {print "\n\nCan not create ".$newDir.$grp4."\n\n\n\n"; exit;}
unless (open (DUB, ">$newDir$selectRepDub")) {print "\nCan not create ".$newDir.$selectRepDub."\n\n"; exit;}
unless (open (RALL,">$newDir$reportallpos")) {print "\nCan not create ".$newDir.$reportallpos."\n\n"; exit;}

for ( my $p = 0 ; $p < (scalar @allParseRep) ; $p++) {
          my @tmp_line = split (/#/, $allParseRep[$p]);
          my $Qnum = $tmp_line[0];                                  # query's seqNo
          my $QL   = $tmp_line[1];                                  # query's length
          my $temp_index = $tmp_line[2];                            # hit/nohit indice

          my $blockCount = scalar (@tmp_line);

          my ($temp_scoreBP, $temp_1stMatch, $temp_strand, $temp_1stMatchS, $use);
          my ($scoreBP2, $firstM2, $strand2, $firstMS2);
          my ($temp_lastpos);       # the position (non-hpv) next to the last hpv match position
          my ($scoreBP1, $firstM1, $strand1, $firstMS1);
          my ($series1, $series2);

          # This first IF-LOOP defies the 'virtual matching area of HPV16'
          #   The value of the 'first position' of HPV-matching area will be specified
          #   as well as other values, e.g. 'virtual scoreBP' in case of two continuous hits

          # ONE-hit only
          if (($temp_index =~ "hitHPV") && ($blockCount == 4)) {  # for ONE-HPV-HIT query
             my @tempBox = split ( /%/, $tmp_line[3]);
             $temp_scoreBP = $tempBox[0];                   # scoreBP
             $temp_1stMatch = $tempBox[2];                  # 1st-Match-position-on-QUERY-strand
             $temp_strand = $tempBox[3];                            # strand of the matched HPV
             $temp_1stMatchS = $tempBox[4];                 # 1st-Match-position-on-Sbjt-strand
             $use = "hitx1_useFIRST";

             $scoreBP1 = $tempBox[0];
             $firstM1 = $tempBox[2];
             $strand1 = $tempBox[3];
             $firstMS1 = $tempBox[4];
             $series1 = "use_Alone";

             $scoreBP2 = "none";
             $firstM2 = "none";
             $strand2 = "none";
             $firstMS2 = "none";
             $series2 = "none";

          # More than ONE hit, then two first hits will be taken into consideration
          } elsif (($temp_index =~ "hitHPV") && ($blockCount > 4)) {  # for MORE-HPV-HITs query

             my @tBox1 = split ( /%/, $tmp_line[3]);
             my $tscoreBP1 = $tBox1[0];                     # scoreBP of FIRST HIT
             my $tpercent1 = $tBox1[1];                     # percent match of FIRST HIT
             my $t1stMatch1 = $tBox1[2];         # 1st-Match-position-on-QUERY-strand of FIRST HIT
             my $tstrand1 = $tBox1[3];                      # strand of the matched HPV of FIRST HIT
             my $t1stMatchS1 = $tBox1[4];

             my @tBox2 = split ( /%/, $tmp_line[4]);
             my $tscoreBP2 = $tBox2[0];                     # scoreBP of SECOND HIT
             my $tpercent2 = $tBox2[1];                     # percent match of SECOND HIT
             my $t1stMatch2 = $tBox2[2];         # 1st-Match-position-on-QUERY-strand of SECOND HIT
             my $tstrand2 = $tBox2[3];                      # strand of the matched HPV of SECOND HIT
             my $t1stMatchS2 = $tBox2[4];

             # (1) : if ending-of-first-hit is EARLIER than ending-of-second-hit
             if ((int($t1stMatch1)+int($tscoreBP1)) < (int($t1stMatch2)+int($tscoreBP2))) {

                       # (1.1): first & second hits begin at the same pos
                       if ( int($t1stMatch1) == int($t1stMatch2) ) {
                                 $temp_scoreBP = $tscoreBP2;
                                 $temp_1stMatch = $t1stMatch2;
                                 $temp_strand = $tstrand2;
                                 $use = "hitx2_useSECOND_1stHitasSubset";
                                 $temp_1stMatchS = $t1stMatchS2;

                                 $scoreBP1 = $tscoreBP1;
                                 $firstM1 = $t1stMatch1;
                                 $strand1 = $tstrand1;
                                 $firstMS1 = int($t1stMatchS1);
                                 $series1 = "subset";

                                 $scoreBP2 = $tscoreBP2;
                                 $firstM2 = $t1stMatch2;
                                 $strand2 = $tstrand2;
                                 $firstMS2 = int($t1stMatchS2);
                                 $series2 = "use_Alone";

                       # (1.2): first hit begins first
                       } elsif ( int($t1stMatch1) < int($t1stMatch2) ) {
```

```
# If last match position of 1stHit is >= 1stMatch of 2.Hit (+2)
# (+2) is the allowance for the gap or if there is no gap then 2 nt distance
# between the two hits
if ( (int($t1stMatch1)+int($tscoreBP1)) >= (int($t1stMatch2)+2) ) {

            # Control if the 2.Hit is long enough to be significantly counted
            if ((int($tscoreBP2) >= $cf3 ) && (int($tpercent2) >= $cf2 )) {
                        $use = "hitx2_useFIRSTnSECOND_continuous";
                        # DoubleHPV-- independent of HPV16's 4 groups

                        print DUB $Qnum."#".$QL."#hitHPV#".$use."\n";
                        $temp_scoreBP = int($tscoreBP2)+(int($t1stMatch2)
                                                -int($t1stMatch1));
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;   # ignore second hit strand
                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "use_1st";

                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "use_2nd";
            } else {
                        $use = "hitx2_useFIRST_tooSmall2ndHit";
                        $temp_scoreBP = $tscoreBP1;
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "use_Alone";

                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "smallerThan_".$cf3;
            }
} else {
            if ((int($tscoreBP2) >= $cf3 ) && (int($tpercent2) >= $cf2 )) {
                        $use = "hitx2_useFIRSTnSECOND_discontinuous";
                        # DoubleHPV-- independent of HPV16's 4 groups

                        print DUB $Qnum."#".$QL."#hitHPV#".$use."\n";
                        $temp_scoreBP = $tscoreBP1; # DIFFERENT FROM "continuous"
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;
                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "use_1st";

                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "use_2nd_discont";

            } else {
                        $use = "hitx2_useFIRST_tooSmall2ndHit";
                        $temp_scoreBP = $tscoreBP1;
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                                                $firstM1 = $t1stMatch1;
                                                $strand1 = $tstrand1;
                                                $firstMS1 = int($t1stMatchS1);
                                                $series1 = "use_Alone";

                                                $scoreBP2 = $tscoreBP2;
                                                $firstM2 = $t1stMatch2;
                                                $strand2 = $tstrand2;
                                                $firstMS2 = int($t1stMatchS2);
                                                $series2 = "smallerThan_".$cf3;
                                    }
                        }
            # (1.3): second hit begins first
            } else {
                        $temp_scoreBP = $tscoreBP2;
                        $temp_1stMatch = $t1stMatch2;
                        $temp_strand = $tstrand2;
                        $use = "hitx2_useSECOND_1stHitasSubset";
                        $temp_1stMatchS = $t1stMatchS2;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "subset";

                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "use_Alone";
            }

    # (2) : if ending-of-first-hit is THE SAME as ending-of-second-hit
    } elsif ((int($t1stMatch1)+int($tscoreBP1))==(int($t1stMatch2)+int($tscoreBP2))) {

                # (2.1): first & second hits begin at the same pos
                if ( int($t1stMatch1) == int($t1stMatch2) ) {
                        $temp_scoreBP = $tscoreBP1;
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;
                        $use = "hitx2_useFIRST_2ndHitasSubset";
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "use_Alone";

                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "subset";

                # (2.2): first hit begins first
                } elsif ( int($t1stMatch1) < int($t1stMatch2) ) {
                        $temp_scoreBP = $tscoreBP1;
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;
                        $use = "hitx2_useFIRST_2ndHitasSubset";
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "use_Alone";

                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "subset";

                # (2.3): second hit begins first
                } else {
```

```
                        if ((int($tpercent1)>=int($tpercent2))&&((int($t1stMatch1)-int($t1stMatch2))<3)) {
                                $temp_scoreBP = $tscoreBP1;
                                $temp_1stMatch = $t1stMatch1;
                                $temp_strand = $tstrand1;
                                $use = "hitx2_useFIRST_2ndHitasSubset";

                                $temp_1stMatchS = $t1stMatchS1;
                                $scoreBP1 = $tscoreBP1;
                                $firstM1 = $t1stMatch1;
                                $strand1 = $tstrand1;
                                $firstMS1 = int($t1stMatchS1);
                                $series1 = "use_Alone";

                                $scoreBP2 = $tscoreBP2;
                                $firstM2 = $t1stMatch2;
                                $strand2 = $tstrand2;
                                $firstMS2 = int($t1stMatchS2);
                                $series2 = "subset";
                        } else {
                                $temp_scoreBP = $tscoreBP2;
                                $temp_1stMatch = $t1stMatch2;
                                $temp_strand = $tstrand2;
                                $use = "hitx2_useSECOND_1stHitasSubset";
                                $temp_1stMatchS = $t1stMatchS2;

                                $scoreBP1 = $tscoreBP1;
                                $firstM1 = $t1stMatch1;
                                $strand1 = $tstrand1;
                                $firstMS1 = int($t1stMatchS1);
                                $series1 = "subset";

                                $scoreBP2 = $tscoreBP2;
                                $firstM2 = $t1stMatch2;
                                $strand2 = $tstrand2;
                                $firstMS2 = int($t1stMatchS2);
                                $series2 = "use_Alone";
                        }
                }
        # (3) : if ending-of-first-hit is LATER than ending-of-second-hit
        } else {

                # (3.1): first & second hits begin at the same pos
                if ( int($t1stMatch1) == int($t1stMatch2) ) {
                        $temp_scoreBP = $tscoreBP1;
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;
                        $use = "hitx2_useFIRST_2ndHitasSubset";
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "use_Alone";

                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "subset";

                # (3.2): first hit begins first
                } elsif ( int($t1stMatch1) < int($t1stMatch2) ) {
                        $temp_scoreBP = $tscoreBP1;
                        $temp_1stMatch = $t1stMatch1;
                        $temp_strand = $tstrand1;
                        $use = "hitx2_useFIRST_2ndHitasSubset";
                        $temp_1stMatchS = $t1stMatchS1;

                        $scoreBP1 = $tscoreBP1;
                        $firstM1 = $t1stMatch1;
                        $strand1 = $tstrand1;
                        $firstMS1 = int($t1stMatchS1);
                        $series1 = "use_Alone";

                        $scoreBP2 = $tscoreBP2;
                        $firstM2 = $t1stMatch2;
                        $strand2 = $tstrand2;
                        $firstMS2 = int($t1stMatchS2);
                        $series2 = "subset";

                # (3.3): second hit begins first
                } else {
                        if ( (int($tscoreBP2) >= $cf3 ) && (int($tpercent2) >= $cf2 )) {

                                if ( (int($t1stMatch2)+int($tscoreBP2)) >= (int($t1stMatch1)+2) ) {
                                        $use = "hitx2_useSECONDnFIRST_continuous";
                                        # DoubleHPV-- independent of HPV16's 4 groups

                                        print DUB $Qnum."#".$QL."#hitHPV#".$use."\n";
                                        $temp_scoreBP = int($tscoreBP1)+(int($t1stMatch1)
                                                                        -int($t1stMatch2));
                                        $temp_1stMatch = $t1stMatch2;
                                        $temp_strand = $tstrand2;
                                        $scoreBP2 = $tscoreBP2;
                                        $firstM2 = $t1stMatch2;
                                        $strand2 = $tstrand2;

                                        $temp_1stMatchS = $t1stMatchS2;
                                        $scoreBP1 = $tscoreBP1;
                                        $firstM1 = $t1stMatch1;
                                        $strand1 = $tstrand1;
                                        $firstMS1 = int($t1stMatchS1);
                                        $series1 = "use_2nd";

                                        $firstMS2 = int($t1stMatchS2);
                                        $series2 = "use_1st";
                                } else {
                                        $use = "hitx2_useSECONDnFIRST_discontinuous";
                                        # DoubleHPV-- independent of HPV16's 4 groups

                                        print DUB $Qnum."#".$QL."#hitHPV#".$use."\n";
                                        $temp_scoreBP = $tscoreBP2; # DIFFERENT FROM "continuous"
                                        $temp_1stMatch = $t1stMatch2;
                                        $temp_strand = $tstrand2;
                                        $scoreBP2 = $tscoreBP2;
                                        $firstM2 = $t1stMatch2;
                                        $strand2 = $tstrand2;

                                        $temp_1stMatchS = $t1stMatchS2;
                                        $scoreBP1 = $tscoreBP1;
                                        $firstM1 = $t1stMatch1;
                                        $strand1 = $tstrand1;
                                        $firstMS1 = int($t1stMatchS1);
                                        $series1 = "use_2nd_discont";

                                        $firstMS2 = int($t1stMatchS2);
                                        $series2 = "use_1st";
                                }
                        } else {
                                $temp_scoreBP = $tscoreBP1;
                                $temp_1stMatch = $t1stMatch1;
                                $temp_strand = $tstrand1;
                                $use = "hitx2_useFIRST_tooSmall2ndHit";
                                $temp_1stMatchS = $t1stMatchS1;

                                $scoreBP1 = $tscoreBP1;
                                $firstM1 = $t1stMatch1;
                                $strand1 = $tstrand1;
                                $firstMS1 = int($t1stMatchS1);
                                $series1 = "use_Alone";

                                $scoreBP2 = $tscoreBP2;
                                $firstM2 = $t1stMatch2;
                                $strand2 = $tstrand2;
                                $firstMS2 = int($t1stMatchS2);
                                $series2 = "smallerThan_".$cf3;
                        }
                }
           }

        } else { # for no-hit
           $use = "noHPV";
        }
```

```
        # SECOND IF-LOOP to SORT each sequence (based on HPV16 matching values)
        #   into G1,G2,G3,G4
        if ($temp_index =~ "nohitHPV") {

           print SELECT $Qnum."#".$QL."#nohitHPV#noHPV#noHPV#noHPV#noHPV\n";
           print RALL $Qnum."#".$QL."#nohitHPV#noHPV#noHPV#noHPV#noHPV\n";

        } else {

           # Must be CHANGED when there is CUTOFF
           # '+2' is used to make it the Last position of HPV match!
           $temp_lastpos = int($temp_1stMatch)+int($temp_scoreBP);
           my $therest = int($QL) - int($temp_lastpos) + 2 ;

           my $firstMS = int($temp_1stMatchS);
           print RALL $Qnum."#".$QL."#".$temp_index."#".$temp_scoreBP."%".$temp_1stMatch."%".
                  $temp_strand."%".$firstMS."%".$temp_lastpos."#".$use."#".
                        $series1."%".$scoreBP1."%".$firstM1."%".$strand1."%".int($firstMS1)."#".
                        $series2."%".$scoreBP2."%".$firstM2."%".$strand2."%".$firstMS2."\n";

           if ( ( int($therest)>=150) && ( int($temp_scoreBP)>=$cf) ) {
                       print SELECT $Qnum."#".$QL."#hitHPV#".$use."#G1\n";
                       print GRP1 $Qnum."\n";
           } elsif ( (int($therest)>=150) && (int($temp_scoreBP)<$cf) ) {
                       print SELECT $Qnum."#".$QL."#hitHPV#nothing_analyzed#out_ranged_G1\n";

           } elsif ( (100<=int($therest)) &&
             (int($therest)<150) && (int($temp_scoreBP)>=$cf) ) {
                       print SELECT $Qnum."#".$QL."#hitHPV#".$use."#G2\n";
                       print GRP2 $Qnum."\n";
           } elsif ( (100<=int($therest)) &&
             (int($therest)<150) && (int($temp_scoreBP)<$cf) ) {
                       print SELECT $Qnum."#".$QL."#hitHPV#nothing_analyzed#out_ranged_G2\n";

           } elsif ( (50<=int($therest)) &&
             (int($therest)<100) && (int($temp_scoreBP)>=$cf) ) {
                       print SELECT $Qnum."#".$QL."#hitHPV#".$use."#G3\n";
                       print GRP3 $Qnum."\n";
           } elsif ( (50<=int($therest)) &&
             (int($therest)<100) && (int($temp_scoreBP)<$cf) ) {
                       print SELECT $Qnum."#".$QL."#hitHPV#nothing_analyzed#out_ranged_G3\n";

           } elsif ( (int($therest)<50) && (int($temp_scoreBP)>=$cf) ) {
                       print SELECT $Qnum."#".$QL."#hitHPV#".$use."#G4\n";
                       print GRP4 $Qnum."\n";
           } elsif ( (int($therest)<50) && (int($temp_scoreBP)<$cf) ) {
                       print SELECT $Qnum."#".$QL."#hitHPV#nothing_analyzed#out_ranged_G4\n";

           } else {
                       print SELECT $Qnum."#".$QL."#hitHPV#".$use."#no_group\n";
           }
        }
}

close SELECT; close GRP1; close GRP2; close GRP3; close GRP4; close DUB; close RALL;

################ END of PART 1 : SELECT into 4 groups   ###################################

############## Start of PART 2+3 : Calling Sequence NAMES & STATISTICS ###################

my $seqNames = $workingDir."sample".$sampleNo[0]."_nameList";

my $seqNames_grp1 = $prefix."select_group1_names";
my $seqNames_grp2 = $prefix."select_group2_names";
my $seqNames_grp3 = $prefix."select_group3_names";
my $seqNames_grp4 = $prefix."select_group4_names";

my @seqNameFile = getFileData($seqNames);

getName3 (\@seqNameFile, $newDir, $grp1, $seqNames_grp1);
getName3 (\@seqNameFile, $newDir, $grp2, $seqNames_grp2);
getName3 (\@seqNameFile, $newDir, $grp3, $seqNames_grp3);
getName3 (\@seqNameFile, $newDir, $grp4, $seqNames_grp4);

my $statRep = $prefix."select_statistics";

getStat2b2 (\@seqNameFile, $newDir, $grp1, $grp2, $grp3, $grp4, $selectRep, $statRep, $cf);
```

```
################ End of PART 2+3 : Calling Sequence NAMES & STATISTICS ###################

print "\n\nPROGRAM 5D (SELECT 28bp CUTOFF).............................FINISHED\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
##############################################################################################

##############################################################################################
###    SUBROUTINES    ######################################################################
##############################################################################################
sub getFileData {
         my ($filename) = @_;
         use strict;
         use warnings;

         my @filedata = ();
         unless ( open(FILE_DATA, $filename) ) {
            print STDERR "\n\nThe program can't open the files \"$filename\"\n\n\n\n";
            print "Please re-check input file name and its location\n";
            print "The correct command should be :\n\n";
            print "\t\tperl PROGRAM(location+name) INPUTFastAFILE(location+name)\n\n";
            print "The command should be given under directory";
            print "  c:/Perl/bin> in case of Dos Terminal\n\n\n";
            exit;
         }
         @filedata = <FILE_DATA>;
         close FILE_DATA;
         return @filedata;
}

# for PART 2222222222222222222222222222222222222233333333333333333333333333333333333333333
sub getName3 {
         my ($seqNameFile, $dir, $in, $out) = @_;
         use strict;
         use warnings;

         unless (open (OUT, ">$dir$out")) {print "\n\nCan not create ".$dir.$out."\n\n\n\n\n";exit;}

         my $loc_in = $dir.$in;
         my @input =  getFileData ($loc_in);
         for ( my $p = 0; $p < (scalar @input) ; $p++) {
            if ( $input[$p] =~ m/^([0-9]*)(.*)$/) {
                        my $seqNum = $1;
                        print OUT $seqNum."#".$$seqNameFile[$seqNum];
            }
         }
         close OUT;
}

sub getStat2b2 {
         my ($seqNameFile, $dir, $in1, $in2, $in3, $in4, $inall, $out, $cf) = @_;
         use strict;
         use warnings;

         unless (open (OUT, ">$dir$out")) {print "\n\nCan not create ".$dir.$out."\n\n\n\n\n";exit;}

         my $in1_loc = $dir.$in1;
         my $in2_loc = $dir.$in2;
         my $in3_loc = $dir.$in3;
         my $in4_loc = $dir.$in4;
         my $inall_loc = $dir.$inall;

         my @in1_dt = getFileData ($in1_loc);   my $noG1a = (scalar @in1_dt);
         my @in2_dt = getFileData ($in2_loc);   my $noG2a = (scalar @in2_dt);
         my @in3_dt = getFileData ($in3_loc);   my $noG3a = (scalar @in3_dt);
         my @in4_dt = getFileData ($in4_loc);   my $noG4a = (scalar @in4_dt);

         my @inall_dt = getFileData ($inall_loc);

         my ($noG1b, $noG2b, $noG3b, $noG4b) = (0,0,0,0);
         my ($outG1, $outG2, $outG3, $outG4) = (0,0,0,0);
         my $nohpv = 0;

         for (my $p = 0; $p < (scalar @inall_dt); $p++) {
            my @line = split (/#/, $inall_dt[$p]);
            my $status = $line[4];
```

```
            if (($status =~ m/^noHPV(.*)$/)|($status =~ m/^no_group(.*)$/)) {$nohpv++;}
            elsif ($status =~ m/^G1(.*)$/) {$noG1b++;}
            elsif ($status =~ m/^G2(.*)$/) {$noG2b++;}
            elsif ($status =~ m/^G3(.*)$/) {$noG3b++;}
            elsif ($status =~ m/^G4(.*)$/) {$noG4b++;       }
            elsif ($status =~ m/^out_ranged_G1(.*)$/) {$outG1++;}
            elsif ($status =~ m/^out_ranged_G2(.*)$/) {$outG2++;}
            elsif ($status =~ m/^out_ranged_G3(.*)$/) {$outG3++;}
            elsif ($status =~ m/^out_ranged_G4(.*)$/) {$outG4++;}

         }

         my ($noG1, $noG2, $noG3, $noG4);
         my $error = 0;

         if ( $noG1a == $noG1b ) { $noG1 = $noG1a; } else { $error++; }
         if ( $noG2a == $noG2b ) { $noG2 = $noG2a; } else { $error++; }
         if ( $noG3a == $noG3b ) { $noG3 = $noG3a; } else { $error++; }
         if ( $noG4a == $noG4b ) { $noG4 = $noG4a; } else { $error++; }

         my $noTotalSeq = (scalar @$seqNameFile);

         print OUT "Total number of sequences having this barcode is  : ".$noTotalSeq."\n\n";
         print OUT "CUT-OFF value used for minimum matching length is : ".$cf." bp\n\n";

         unless ($error != 0) {
            my $noUnclass  = $noTotalSeq - $noG1 - $noG2 - $noG3 - $noG4;
            print OUT "Number of sequences in group 1               : ".$noG1."\n";
            print OUT "Number of sequences in group 2               : ".$noG2."\n";
            print OUT "Number of sequences in group 3               : ".$noG3."\n";
            print OUT "Number of sequences in group 4               : ".$noG4."\n\n";
            print OUT "Number of sequences with UNCLASSIFIED-status : ".$noUnclass."\n";
            print OUT "   -> OUT-Ranged G1 : ".$outG1."\n";
            print OUT "   -> OUT-Ranged G2 : ".$outG2."\n";
            print OUT "   -> OUT-Ranged G3 : ".$outG3."\n";
            print OUT "   -> OUT-Ranged G4 : ".$outG4."\n";
            print OUT "   -> no-hit-to-HPV : ".$nohpv."\n";
         }
         print OUT "\nNote that UNCLASSIFIED can mean either :\n";
         print OUT "     a) OUT-RANGED from the four groups, i.e. Matching is shorter than CUTOFF";
         print OUT "\nor   b) the sequence has NO-HIT-to-HPV16\n";
         close OUT;
}

#xx Sub-routines END HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
################################################################################

################################################################################################################
```

File name: **set2_p5mac_edit.pl**

Source code:

```
#!/usr/bin/perl

### About the script ###############################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
################################################################################
### MAIN PROGRAM BODY ##########################################################
################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;

my $prefix_path= $sampleNo[2];
my $pyroNum = int($sampleNo[1]);

my $path = $prefix_path."Pyro".$pyroNum."_result/";
my $workingDir = $path.$sampleNo[0]."/";
my $fastaRC = $workingDir."sample".$sampleNo[0]."_RC.txt";

my $cut00 = "noCutoff";
my $cut28 = "28bpCutoff";
my $newDir00 = $workingDir.$cut00."/";
my $newDir28 = $workingDir.$cut28."/";

print "\n\nPROGRAM GETTING REVERSE COMPLEMENT FASTA SEQ LIST OF 4 GROUP STARTS.............";

getRevComFastaG1to4 ($workingDir, $fastaRC, $cut00);
getRevComFastaG1to4 ($workingDir, $fastaRC, $cut28);

print "FINISHED\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
################################################################################

################################################################################
###    SUBROUTINES    ##########################################################
################################################################################

sub getFileData {
         my ($filename) = @_;
         use strict;
         use warnings;

         my @filedata = ();
         unless ( open(FILE_DATA, $filename) ) {
            print STDERR "\n\nThe program can't open the files \"$filename\"\n\n\n\n";
            print "Please re-check input file name and its location\n";
            print "The correct command should be :\n\n";
            print "\t\tperl PROGRAM(location+name) INPUTFastAFILE(location+name)\n\n";
            print "The command should be given under directory";
            print "  c:/Perl/bin> in case of Dos Terminal\n\n\n";
            exit;
         }
         @filedata = <FILE_DATA>;
         close FILE_DATA;
         return @filedata;
}

sub getRevComFastaG1to4 {
         my ($workdir, $fasInFile, $prefix) = @_;

         my $dir = $workdir.$prefix."/";
         my $outsuffix = "_fasta_RC";
         my @fas = getFileData ($fasInFile);

         for ( my $p = 1; $p < 5; $p++) {
            my $grouplist = $dir.$prefix."_select_group".$p."_names";
            my @list = getFileData ($grouplist);

            my $out = $dir.$prefix."_select_group".$p.$outsuffix;
            unless (open (FA, ">$out")) {print "\n\nCan not create ".$out."\n\n\n\n"; exit;}

            for (my $a = 0; $a < (scalar @list);  $a++) {
                        my @box = split (/#/, $list[$a]);
                        my $seqnum = $box[0];
                        if ( $seqnum =~ /^([0-9]*)$/) {
                                    my $num = int($seqnum);
                                    my $infasName = $fas[2*$num];
                                    $infasName =~ s/\s//g;
                                    my $infasSeq = $fas[(2*$num)+1];
                                    print FA $infasName."\n".$infasSeq;

                        } else {
                                    print "\n\nError with seq number list\n\n";
                                    exit;
                        }
            }
            close FA;
         }
}

#xx Sub-routines END HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
################################################################################

################################################################################################################
```

File name: **set2_p6mac_before.pl**
Source code:

```
#!/usr/bin/perl

### About the script ##################################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
#######################################################################################
### MAIN PROGRAM BODY ##################################################################
#######################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;

my $prefix_path= $sampleNo[2];
my $pyroNum = int($sampleNo[1]);

my $path = $prefix_path."Pyro".$pyroNum."_result/";
my $samplePath = $path.$sampleNo[0]."/";
my $workDir = $samplePath."intersection/";
my $inOrig = $samplePath."sample".$sampleNo[0]."_RC.txt";
my $barcodenumber = int($sampleNo[0]);

my $cut00 = "noCutoff";
my $cut28 = "28bpCutoff";

print "\n\nProgram6 intersection....for EE... STARTs.............";

system ("mkdir", $workDir);

my $inEE = $samplePath."blastEE/parsed_EE_combined_report";

my @inP = getFileData ($inEE);

getCombined3 ($samplePath, \@inP, $cut00, $workDir);
getCombined3 ($samplePath, \@inP, $cut28, $workDir);

my @primerListEEpyro3 = ("E17","E02","E03","E04","E05","E06","E07","E08",
                         "E09", "E21", "E11", "E12", "E13", "E14", "E15", "E16",
                         "E18", "E19", "E20", "E22", "E23", "E24", "E25", "E26",
                         "E27", "E28", "E29", "E30", "E31", "E32", "nohitEE");
getStatCorrect3 ($workDir, $cut00, \@primerListEEpyro3, $barcodenumber);
getStatCorrect3 ($workDir, $cut28, \@primerListEEpyro3, $barcodenumber);

print "FINISHED\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
#######################################################################################

#######################################################################################
###    SUBROUTINES    ###################################################################
#######################################################################################
sub getFileData {
         my ($filename) = @_;
         use strict;
         use warnings;

         my @filedata = ();
         unless ( open(FILE_DATA, $filename) ) {
            print STDERR "\n\nThe program can't open the files \"$filename\"\n\n\n\n";
            print "Please re-check input file name and its location\n";
            print "The correct command should be :\n\n";
            print "\t\tperl PROGRAM(location+name) INPUTFastAFILE(location+name)\n\n";
            print "The command should be given under directory";
            print " c:/Perl/bin> in case of Dos Terminal\n\n\n";
            exit;
         }
         @filedata = <FILE_DATA>;
         close FILE_DATA;
         return @filedata;
}

sub getCombined3 {
         my ($path, $inP, $prefix, $outDir) = @_;
         use strict;
         use warnings;

         my $cutDir = $path.$prefix."/";
         my $inCutFile = $cutDir.$prefix."_select_report";
         my $outDircutDir = $outDir.$prefix."_intersect/";

         system ("mkdir", $outDircutDir);
         my $outrep = $outDircutDir.$prefix."_intersect_combined";

         unless (open (OUT, ">$outrep")) {print "\n\n\n\nCan not create ".$outrep."\n\n\n\n"; exit;}
         my @inCut = getFileData ($inCutFile);
         for (my $p = 0; $p < (scalar @inCut); $p++) {
            my @tmpLine = split (/#/, $inCut[$p]);
            my $nameNum = $tmpLine[0];
            my $grpCat = $tmpLine[4];
            chomp ($grpCat);

            my $inPnum = int($nameNum);
            my @tmpP = split (/#/, $$inP[$inPnum]);
            my $seqNumP = $tmpP[0];
            if (int($seqNumP) == int($inPnum)) {
                     print OUT $seqNumP."#".$grpCat."#".$$inP[$inPnum];
            }
         }
         close OUT;
}

sub getStatCorrect3 {
         my ($path, $prefix, $pL, $barcodenumber) = @_;
         use strict;
         use warnings;

         my $subPath = $path.$prefix."_intersect/";
         my $inFile = $subPath.$prefix."_intersect_combined";
         my $outsigPrim = $subPath.$prefix."_intersect_statistics";
         my $nameResult = $subPath.$prefix."sum_result_after_intersection";

         # for each primer category -->
         # IF SIGNIFICANT PRIMER MATCH --> which group of HPV match category?
         my ($e1g1, $e1g2, $e1g3, $e1g4, $e1out, $e1nh) = (0,0,0,0,0,0);
         my ($e2g1, $e2g2, $e2g3, $e2g4, $e2out, $e2nh) = (0,0,0,0,0,0);
         my ($e3g1, $e3g2, $e3g3, $e3g4, $e3out, $e3nh) = (0,0,0,0,0,0);
         my ($e4g1, $e4g2, $e4g3, $e4g4, $e4out, $e4nh) = (0,0,0,0,0,0);
         my ($e5g1, $e5g2, $e5g3, $e5g4, $e5out, $e5nh) = (0,0,0,0,0,0);
         my ($e6g1, $e6g2, $e6g3, $e6g4, $e6out, $e6nh) = (0,0,0,0,0,0);
         my ($e7g1, $e7g2, $e7g3, $e7g4, $e7out, $e7nh) = (0,0,0,0,0,0);
         my ($e8g1, $e8g2, $e8g3, $e8g4, $e8out, $e8nh) = (0,0,0,0,0,0);
         my ($e9g1, $e9g2, $e9g3, $e9g4, $e9out, $e9nh) = (0,0,0,0,0,0);
         my ($e10g1, $e10g2, $e10g3, $e10g4, $e10out, $e10nh) = (0,0,0,0,0,0);
         my ($e11g1, $e11g2, $e11g3, $e11g4, $e11out, $e11nh) = (0,0,0,0,0,0);
         my ($e12g1, $e12g2, $e12g3, $e12g4, $e12out, $e12nh) = (0,0,0,0,0,0);
         my ($e13g1, $e13g2, $e13g3, $e13g4, $e13out, $e13nh) = (0,0,0,0,0,0);
         my ($e14g1, $e14g2, $e14g3, $e14g4, $e14out, $e14nh) = (0,0,0,0,0,0);
         my ($e15g1, $e15g2, $e15g3, $e15g4, $e15out, $e15nh) = (0,0,0,0,0,0);
         my ($e16g1, $e16g2, $e16g3, $e16g4, $e16out, $e16nh) = (0,0,0,0,0,0);
         my ($e18g1, $e18g2, $e18g3, $e18g4, $e18out, $e18nh) = (0,0,0,0,0,0);
         my ($e19g1, $e19g2, $e19g3, $e19g4, $e19out, $e19nh) = (0,0,0,0,0,0);
         my ($e20g1, $e20g2, $e20g3, $e20g4, $e20out, $e20nh) = (0,0,0,0,0,0);
         my ($e22g1, $e22g2, $e22g3, $e22g4, $e22out, $e22nh) = (0,0,0,0,0,0);
         my ($e23g1, $e23g2, $e23g3, $e23g4, $e23out, $e23nh) = (0,0,0,0,0,0);
         my ($e24g1, $e24g2, $e24g3, $e24g4, $e24out, $e24nh) = (0,0,0,0,0,0);
         my ($e25g1, $e25g2, $e25g3, $e25g4, $e25out, $e25nh) = (0,0,0,0,0,0);
         my ($e26g1, $e26g2, $e26g3, $e26g4, $e26out, $e26nh) = (0,0,0,0,0,0);
         my ($e27g1, $e27g2, $e27g3, $e27g4, $e27out, $e27nh) = (0,0,0,0,0,0);
         my ($e28g1, $e28g2, $e28g3, $e28g4, $e28out, $e28nh) = (0,0,0,0,0,0);
         my ($e29g1, $e29g2, $e29g3, $e29g4, $e29out, $e29nh) = (0,0,0,0,0,0);
         my ($e30g1, $e30g2, $e30g3, $e30g4, $e30out, $e30nh) = (0,0,0,0,0,0);
         my ($e31g1, $e31g2, $e31g3, $e31g4, $e31out, $e31nh) = (0,0,0,0,0,0);
         my ($e32g1, $e32g2, $e32g3, $e32g4, $e32out, $e32nh) = (0,0,0,0,0,0);

         # IF INSIG primer match
         my ($oe1g1, $oe1g2, $oe1g3, $oe1g4, $oe1out, $oe1nh) = (0,0,0,0,0,0);
         my ($oe2g1, $oe2g2, $oe2g3, $oe2g4, $oe2out, $oe2nh) = (0,0,0,0,0,0);
         my ($oe3g1, $oe3g2, $oe3g3, $oe3g4, $oe3out, $oe3nh) = (0,0,0,0,0,0);
```

```
my ($oe4g1, $oe4g2, $oe4g3, $oe4g4, $oe4out, $oe4nh) = (0,0,0,0,0,0);
my ($oe5g1, $oe5g2, $oe5g3, $oe5g4, $oe5out, $oe5nh) = (0,0,0,0,0,0);
my ($oe6g1, $oe6g2, $oe6g3, $oe6g4, $oe6out, $oe6nh) = (0,0,0,0,0,0);
my ($oe7g1, $oe7g2, $oe7g3, $oe7g4, $oe7out, $oe7nh) = (0,0,0,0,0,0);
my ($oe8g1, $oe8g2, $oe8g3, $oe8g4, $oe8out, $oe8nh) = (0,0,0,0,0,0);
my ($oe9g1, $oe9g2, $oe9g3, $oe9g4, $oe9out, $oe9nh) = (0,0,0,0,0,0);
my ($oe10g1, $oe10g2, $oe10g3, $oe10g4, $oe10out, $oe10nh) = (0,0,0,0,0,0);
my ($oe11g1, $oe11g2, $oe11g3, $oe11g4, $oe11out, $oe11nh) = (0,0,0,0,0,0);
my ($oe12g1, $oe12g2, $oe12g3, $oe12g4, $oe12out, $oe12nh) = (0,0,0,0,0,0);
my ($oe13g1, $oe13g2, $oe13g3, $oe13g4, $oe13out, $oe13nh) = (0,0,0,0,0,0);
my ($oe14g1, $oe14g2, $oe14g3, $oe14g4, $oe14out, $oe14nh) = (0,0,0,0,0,0);
my ($oe15g1, $oe15g2, $oe15g3, $oe15g4, $oe15out, $oe15nh) = (0,0,0,0,0,0);
my ($oe16g1, $oe16g2, $oe16g3, $oe16g4, $oe16out, $oe16nh) = (0,0,0,0,0,0);
my ($oe18g1, $oe18g2, $oe18g3, $oe18g4, $oe18out, $oe18nh) = (0,0,0,0,0,0);
my ($oe19g1, $oe19g2, $oe19g3, $oe19g4, $oe19out, $oe19nh) = (0,0,0,0,0,0);
my ($oe20g1, $oe20g2, $oe20g3, $oe20g4, $oe20out, $oe20nh) = (0,0,0,0,0,0);
my ($oe22g1, $oe22g2, $oe22g3, $oe22g4, $oe22out, $oe22nh) = (0,0,0,0,0,0);
my ($oe23g1, $oe23g2, $oe23g3, $oe23g4, $oe23out, $oe23nh) = (0,0,0,0,0,0);
my ($oe24g1, $oe24g2, $oe24g3, $oe24g4, $oe24out, $oe24nh) = (0,0,0,0,0,0);
my ($oe25g1, $oe25g2, $oe25g3, $oe25g4, $oe25out, $oe25nh) = (0,0,0,0,0,0);
my ($oe26g1, $oe26g2, $oe26g3, $oe26g4, $oe26out, $oe26nh) = (0,0,0,0,0,0);
my ($oe27g1, $oe27g2, $oe27g3, $oe27g4, $oe27out, $oe27nh) = (0,0,0,0,0,0);
my ($oe28g1, $oe28g2, $oe28g3, $oe28g4, $oe28out, $oe28nh) = (0,0,0,0,0,0);
my ($oe29g1, $oe29g2, $oe29g3, $oe29g4, $oe29out, $oe29nh) = (0,0,0,0,0,0);
my ($oe30g1, $oe30g2, $oe30g3, $oe30g4, $oe30out, $oe30nh) = (0,0,0,0,0,0);
my ($oe31g1, $oe31g2, $oe31g3, $oe31g4, $oe31out, $oe31nh) = (0,0,0,0,0,0);
my ($oe32g1, $oe32g2, $oe32g3, $oe32g4, $oe32out, $oe32nh) = (0,0,0,0,0,0);

# IF NO PRIMER MATCH
my ($npg1, $npg2, $npg3, $npg4, $npout, $npnh) = (0,0,0,0,0,0);

unless (open (RE, ">$nameResult")) {print "\n\nCan not create ".$nameResult."\n\n"; exit;}

print RE "Barcode_$barcodenumber\n";
print RE "\t\t\tFor_$prefix\n\n";
print RE "SeqNum\tPrimer\tPrimer_matching\tHPV_category\n";

my $lie1 = $subPath.$prefix.$$pL[0]."_n_sigHPV_list";
my $lie2 = $subPath.$prefix.$$pL[1]."_n_sigHPV_list";
my $lie3 = $subPath.$prefix.$$pL[2]."_n_sigHPV_list";
my $lie4 = $subPath.$prefix.$$pL[3]."_n_sigHPV_list";
my $lie5 = $subPath.$prefix.$$pL[4]."_n_sigHPV_list";
my $lie6 = $subPath.$prefix.$$pL[5]."_n_sigHPV_list";
my $lie7 = $subPath.$prefix.$$pL[6]."_n_sigHPV_list";
my $lie8 = $subPath.$prefix.$$pL[7]."_n_sigHPV_list";
my $lil1 = $subPath.$prefix.$$pL[8]."_n_sigHPV_list";
my $lil2 = $subPath.$prefix.$$pL[9]."_n_sigHPV_list";
my $lil3 = $subPath.$prefix.$$pL[10]."_n_sigHPV_list";
my $lil4 = $subPath.$prefix.$$pL[11]."_n_sigHPV_list";
my $lil5 = $subPath.$prefix.$$pL[12]."_n_sigHPV_list";
my $lil6 = $subPath.$prefix.$$pL[13]."_n_sigHPV_list";
my $lil7 = $subPath.$prefix.$$pL[14]."_n_sigHPV_list";
my $lil8 = $subPath.$prefix.$$pL[15]."_n_sigHPV_list";
my $lie18 = $subPath.$prefix.$$pL[16]."_n_sigHPV_list";
my $lie19 = $subPath.$prefix.$$pL[17]."_n_sigHPV_list";
my $lie20 = $subPath.$prefix.$$pL[18]."_n_sigHPV_list";
my $lie22 = $subPath.$prefix.$$pL[19]."_n_sigHPV_list";
my $lie23 = $subPath.$prefix.$$pL[20]."_n_sigHPV_list";
my $lie24 = $subPath.$prefix.$$pL[21]."_n_sigHPV_list";
my $lie25 = $subPath.$prefix.$$pL[22]."_n_sigHPV_list";
my $lie26 = $subPath.$prefix.$$pL[23]."_n_sigHPV_list";
my $lie27 = $subPath.$prefix.$$pL[24]."_n_sigHPV_list";
my $lie28 = $subPath.$prefix.$$pL[25]."_n_sigHPV_list";
my $lie29 = $subPath.$prefix.$$pL[26]."_n_sigHPV_list";
my $lie30 = $subPath.$prefix.$$pL[27]."_n_sigHPV_list";
my $lie31 = $subPath.$prefix.$$pL[28]."_n_sigHPV_list";
my $lie32 = $subPath.$prefix.$$pL[29]."_n_sigHPV_list";

unless (open (LE1, ">$lie1")) {print "\n\n\n\nCan not create ".$lie1."\n\n\n\n"; exit;}
unless (open (LE2, ">$lie2")) {print "\n\n\n\nCan not create ".$lie2."\n\n\n\n"; exit;}
unless (open (LE3, ">$lie3")) {print "\n\n\n\nCan not create ".$lie3."\n\n\n\n"; exit;}
unless (open (LE4, ">$lie4")) {print "\n\n\n\nCan not create ".$lie4."\n\n\n\n"; exit;}
unless (open (LE5, ">$lie5")) {print "\n\n\n\nCan not create ".$lie5."\n\n\n\n"; exit;}
unless (open (LE6, ">$lie6")) {print "\n\n\n\nCan not create ".$lie6."\n\n\n\n"; exit;}
unless (open (LE7, ">$lie7")) {print "\n\n\n\nCan not create ".$lie7."\n\n\n\n"; exit;}
unless (open (LE8, ">$lie8")) {print "\n\n\n\nCan not create ".$lie8."\n\n\n\n"; exit;}
unless (open (LL1, ">$lil1")) {print "\n\n\n\nCan not create ".$lil1."\n\n\n\n"; exit;}
unless (open (LL2, ">$lil2")) {print "\n\n\n\nCan not create ".$lil2."\n\n\n\n"; exit;}
unless (open (LL3, ">$lil3")) {print "\n\n\n\nCan not create ".$lil3."\n\n\n\n"; exit;}
unless (open (LL4, ">$lil4")) {print "\n\n\n\nCan not create ".$lil4."\n\n\n\n"; exit;}
unless (open (LL5, ">$lil5")) {print "\n\n\n\nCan not create ".$lil5."\n\n\n\n"; exit;}
unless (open (LL6, ">$lil6")) {print "\n\n\n\nCan not create ".$lil6."\n\n\n\n"; exit;}
unless (open (LL7, ">$lil7")) {print "\n\n\n\nCan not create ".$lil7."\n\n\n\n"; exit;}
unless (open (LL8, ">$lil8")) {print "\n\n\n\nCan not create ".$lil8."\n\n\n\n"; exit;}
unless (open (LE18, ">$lie18")) {print "\n\n\n\nCan not create ".$lie18."\n\n\n\n"; exit;}
unless (open (LE19, ">$lie19")) {print "\n\n\n\nCan not create ".$lie19."\n\n\n\n"; exit;}
unless (open (LE20, ">$lie20")) {print "\n\n\n\nCan not create ".$lie20."\n\n\n\n"; exit;}
unless (open (LE22, ">$lie22")) {print "\n\n\n\nCan not create ".$lie22."\n\n\n\n"; exit;}
unless (open (LE23, ">$lie23")) {print "\n\n\n\nCan not create ".$lie23."\n\n\n\n"; exit;}
unless (open (LE24, ">$lie24")) {print "\n\n\n\nCan not create ".$lie24."\n\n\n\n"; exit;}
unless (open (LE25, ">$lie25")) {print "\n\n\n\nCan not create ".$lie25."\n\n\n\n"; exit;}
unless (open (LE26, ">$lie26")) {print "\n\n\n\nCan not create ".$lie26."\n\n\n\n"; exit;}
unless (open (LE27, ">$lie27")) {print "\n\n\n\nCan not create ".$lie27."\n\n\n\n"; exit;}
unless (open (LE28, ">$lie28")) {print "\n\n\n\nCan not create ".$lie28."\n\n\n\n"; exit;}
unless (open (LE29, ">$lie29")) {print "\n\n\n\nCan not create ".$lie29."\n\n\n\n"; exit;}
unless (open (LE30, ">$lie30")) {print "\n\n\n\nCan not create ".$lie30."\n\n\n\n"; exit;}
unless (open (LE31, ">$lie31")) {print "\n\n\n\nCan not create ".$lie31."\n\n\n\n"; exit;}
unless (open (LE32, ">$lie32")) {print "\n\n\n\nCan not create ".$lie32."\n\n\n\n"; exit;}

my @in = getFileData ($inFile);
for (my $m = 0; $m < (scalar @in); $m++) {
   my @line = split (/#/, $in[$m]);
   my $num = $line[0];
   my $grp = $line[1];
   my $prim = $line[3];
   my $cate = $line[4];
   chomp ($cate);

   if ($prim =~ /^$$pL[0]$/) {
            if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                     if ($grp =~ /^G1$/) {
                              $e1g1++;print RE "$num\t$$pL[0]\t$cate\t$grp\n";print LE1 "$num\n";}
                     elsif ($grp =~ /^G2$/) {
                              $e1g2++; print RE "$num\t$$pL[0]\t$cate\t$grp\n"; print LE1 "$num\n";}
                     elsif ($grp =~ /^G3$/) {
                              $e1g3++; print RE "$num\t$$pL[0]\t$cate\t$grp\n"; print LE1 "$num\n";}
                     elsif ($grp =~ /^G4$/) {
                              $e1g4++; print RE "$num\t$$pL[0]\t$cate\t$grp\n"; print LE1 "$num\n";}
                     elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                              $e1out++; print RE "$num\t$$pL[0]\t$cate\t$grp\n";}
                     elsif ($grp =~ /^no_group$/) {$e1out++; print RE "$num\t$$pL[0]\t$cate\t$grp\n";}
                     elsif ($grp =~ /^noHPV$/) {$e1nh++; print RE "$num\t$$pL[0]\t$cate\t$grp\n";}
                     else {print "\n\nError\n\n"; exit;}
            } elsif (($cate =~ /^OUT(.*)$/)) {
                     if    ($grp =~ /^G1$/) {$oe1g1++; print RE "$num\tINsig_$$pL[0]\t$cate\t$grp\n";}
                     elsif ($grp =~ /^G2$/) {$oe1g2++; print RE "$num\tINsig_$$pL[0]\t$cate\t$grp\n";}
                     elsif ($grp =~ /^G3$/) {$oe1g3++; print RE "$num\tINsig_$$pL[0]\t$cate\t$grp\n";}
                     elsif ($grp =~ /^G4$/) {$oe1g4++; print RE "$num\tINsig_$$pL[0]\t$cate\t$grp\n";}
                     elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                              $oe1out++; print RE "$num\tINsig_$$pL[0]\t$cate\t$grp\n";}
                     elsif ($grp =~ /^no_group$/) {
                              $oe1out++; print RE "$num\tINsig_$$pL[0]\t$cate\t$grp\n";}
                     elsif ($grp =~ /^noHPV$/) {$oe1nh++; print RE "$num\tINsig_$$pL[0]\t$cate\t$grp\n";}
                     else {print "\n\nError\n\n"; exit;}
            } else {
                     print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
            }
   } elsif ($prim =~ /^$$pL[1]$/) {
            if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                     if ($grp =~ /^G1$/) {
                              $e2g1++; print RE "$num\t$$pL[1]\t$cate\t$grp\n"; print LE2 "$num\n";}
                     elsif ($grp =~ /^G2$/) {
                              $e2g2++; print RE "$num\t$$pL[1]\t$cate\t$grp\n"; print LE2 "$num\n";}
                     elsif ($grp =~ /^G3$/) {
                              $e2g3++; print RE "$num\t$$pL[1]\t$cate\t$grp\n"; print LE2 "$num\n";}
                     elsif ($grp =~ /^G4$/) {
                              $e2g4++; print RE "$num\t$$pL[1]\t$cate\t$grp\n"; print LE2 "$num\n";}
                     elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                              $e2out++; print RE "$num\t$$pL[1]\t$cate\t$grp\n";}
                     elsif ($grp =~ /^no_group$/) {$e2out++; print RE "$num\t$$pL[1]\t$cate\t$grp\n";}
                     elsif ($grp =~ /^noHPV$/) {$e2nh++; print RE "$num\t$$pL[1]\t$cate\t$grp\n";}
                     else {print "\n\nError\n\n"; exit;}
            } elsif (($cate =~ /^OUT(.*)$/)) {
                     if    ($grp =~ /^G1$/) {$oe2g1++; print RE "$num\tINsig_$$pL[1]\t$cate\t$grp\n";}
                     elsif ($grp =~ /^G2$/) {$oe2g2++; print RE "$num\tINsig_$$pL[1]\t$cate\t$grp\n";}
                     elsif ($grp =~ /^G3$/) {$oe2g3++; print RE "$num\tINsig_$$pL[1]\t$cate\t$grp\n";}
```

```
                        elsif ($grp =~ /^G4$/) {$oe2g4++; print RE "$num\tINsig_$$pL[1]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe2out++; print RE "$num\tINsig_$$pL[1]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe2out++; print RE "$num\tINsig_$$pL[1]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$oe2nh++; print RE "$num\tINsig_$$pL[1]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[2]$/) {
                if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e3g1++; print RE "$num\t$$pL[2]\t$cate\t$grp\n"; print LE3 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e3g2++; print RE "$num\t$$pL[2]\t$cate\t$grp\n"; print LE3 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e3g3++; print RE "$num\t$$pL[2]\t$cate\t$grp\n"; print LE3 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e3g4++; print RE "$num\t$$pL[2]\t$cate\t$grp\n"; print LE3 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e3out++; print RE "$num\t$$pL[2]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e3out++; print RE "$num\t$$pL[2]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e3nh++; print RE "$num\t$$pL[2]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe3g1++; print RE "$num\tINsig_$$pL[2]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe3g2++; print RE "$num\tINsig_$$pL[2]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe3g3++; print RE "$num\tINsig_$$pL[2]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe3g4++; print RE "$num\tINsig_$$pL[2]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe3out++; print RE "$num\tINsig_$$pL[2]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe3out++; print RE "$num\tINsig_$$pL[2]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$oe3nh++; print RE "$num\tINsig_$$pL[2]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[3]$/) {
                if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e4g1++; print RE "$num\t$$pL[3]\t$cate\t$grp\n"; print LE4 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e4g2++; print RE "$num\t$$pL[3]\t$cate\t$grp\n"; print LE4 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e4g3++; print RE "$num\t$$pL[3]\t$cate\t$grp\n"; print LE4 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e4g4++; print RE "$num\t$$pL[3]\t$cate\t$grp\n"; print LE4 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e4out++; print RE "$num\t$$pL[3]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e4out++; print RE "$num\t$$pL[3]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e4nh++; print RE "$num\t$$pL[3]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe4g1++; print RE "$num\tINsig_$$pL[3]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe4g2++; print RE "$num\tINsig_$$pL[3]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe4g3++; print RE "$num\tINsig_$$pL[3]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe4g4++; print RE "$num\tINsig_$$pL[3]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe4out++; print RE "$num\tINsig_$$pL[3]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe4out++; print RE "$num\tINsig_$$pL[3]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$oe4nh++; print RE "$num\tINsig_$$pL[3]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[4]$/) {
                if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e5g1++; print RE "$num\t$$pL[4]\t$cate\t$grp\n"; print LE5 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e5g2++; print RE "$num\t$$pL[4]\t$cate\t$grp\n"; print LE5 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e5g3++; print RE "$num\t$$pL[4]\t$cate\t$grp\n"; print LE5 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e5g4++; print RE "$num\t$$pL[4]\t$cate\t$grp\n"; print LE5 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e5out++; print RE "$num\t$$pL[4]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e5out++; print RE "$num\t$$pL[4]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e5nh++; print RE "$num\t$$pL[4]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe5g1++; print RE "$num\tINsig_$$pL[4]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe5g2++; print RE "$num\tINsig_$$pL[4]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe5g3++; print RE "$num\tINsig_$$pL[4]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe5g4++; print RE "$num\tINsig_$$pL[4]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe5out++; print RE "$num\tINsig_$$pL[4]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe5out++; print RE "$num\tINsig_$$pL[4]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$oe5nh++; print RE "$num\tINsig_$$pL[4]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[5]$/) {
                if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e6g1++; print RE "$num\t$$pL[5]\t$cate\t$grp\n"; print LE6 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e6g2++; print RE "$num\t$$pL[5]\t$cate\t$grp\n"; print LE6 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e6g3++; print RE "$num\t$$pL[5]\t$cate\t$grp\n"; print LE6 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e6g4++; print RE "$num\t$$pL[5]\t$cate\t$grp\n"; print LE6 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e6out++; print RE "$num\t$$pL[5]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e6out++; print RE "$num\t$$pL[5]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e6nh++; print RE "$num\t$$pL[5]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe6g1++; print RE "$num\tINsig_$$pL[5]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe6g2++; print RE "$num\tINsig_$$pL[5]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe6g3++; print RE "$num\tINsig_$$pL[5]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe6g4++; print RE "$num\tINsig_$$pL[5]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe6out++; print RE "$num\tINsig_$$pL[5]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe6out++; print RE "$num\tINsig_$$pL[5]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$oe6nh++; print RE "$num\tINsig_$$pL[5]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[6]$/) {
                if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e7g1++; print RE "$num\t$$pL[6]\t$cate\t$grp\n"; print LE7 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e7g2++; print RE "$num\t$$pL[6]\t$cate\t$grp\n"; print LE7 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e7g3++; print RE "$num\t$$pL[6]\t$cate\t$grp\n"; print LE7 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e7g4++; print RE "$num\t$$pL[6]\t$cate\t$grp\n"; print LE7 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e7out++; print RE "$num\t$$pL[6]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e7out++; print RE "$num\t$$pL[6]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e7nh++; print RE "$num\t$$pL[6]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe7g1++; print RE "$num\tINsig_$$pL[6]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe7g2++; print RE "$num\tINsig_$$pL[6]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe7g3++; print RE "$num\tINsig_$$pL[6]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe7g4++; print RE "$num\tINsig_$$pL[6]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe7out++; print RE "$num\tINsig_$$pL[6]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe7out++; print RE "$num\tINsig_$$pL[6]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$oe7nh++; print RE "$num\tINsig_$$pL[6]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[7]$/) {
```

```
        if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                if    ($grp =~ /^G1$/) {
                        $e8g1++; print RE "$num\t$$pL[7]\t$cate\t$grp\n"; print LE8 "$num\n";}
                elsif ($grp =~ /^G2$/) {
                        $e8g2++; print RE "$num\t$$pL[7]\t$cate\t$grp\n"; print LE8 "$num\n";}
                elsif ($grp =~ /^G3$/) {
                        $e8g3++; print RE "$num\t$$pL[7]\t$cate\t$grp\n"; print LE8 "$num\n";}
                elsif ($grp =~ /^G4$/) {
                        $e8g4++; print RE "$num\t$$pL[7]\t$cate\t$grp\n"; print LE8 "$num\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $e8out++; print RE "$num\t$$pL[7]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {$e8out++; print RE "$num\t$$pL[7]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/) {$e8nh++; print RE "$num\t$$pL[7]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } elsif (($cate =~ /^OUT(.*)$/)) {
                if    ($grp =~ /^G1$/) {$oe8g1++; print RE "$num\tINsig_$$pL[7]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G2$/) {$oe8g2++; print RE "$num\tINsig_$$pL[7]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G3$/) {$oe8g3++; print RE "$num\tINsig_$$pL[7]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G4$/) {$oe8g4++; print RE "$num\tINsig_$$pL[7]\t$cate\t$grp\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $oe8out++; print RE "$num\tINsig_$$pL[7]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {
                        $oe8out++; print RE "$num\tINsig_$$pL[7]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/) {$oe8nh++; print RE "$num\tINsig_$$pL[7]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } else {
                print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
        }
} elsif ($prim =~ /^$$pL[8]$/) {
        if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                if    ($grp =~ /^G1$/) {
                        $e9g1++; print RE "$num\t$$pL[8]\t$cate\t$grp\n"; print LL1 "$num\n";}
                elsif ($grp =~ /^G2$/) {
                        $e9g2++; print RE "$num\t$$pL[8]\t$cate\t$grp\n"; print LL1 "$num\n";}
                elsif ($grp =~ /^G3$/) {
                        $e9g3++; print RE "$num\t$$pL[8]\t$cate\t$grp\n"; print LL1 "$num\n";}
                elsif ($grp =~ /^G4$/) {
                        $e9g4++; print RE "$num\t$$pL[8]\t$cate\t$grp\n"; print LL1 "$num\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $e9out++; print RE "$num\t$$pL[8]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {$e9out++; print RE "$num\t$$pL[8]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/) {$e9nh++; print RE "$num\t$$pL[8]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } elsif (($cate =~ /^OUT(.*)$/)) {
                if    ($grp =~ /^G1$/) {$oe9g1++; print RE "$num\tINsig_$$pL[8]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G2$/) {$oe9g2++; print RE "$num\tINsig_$$pL[8]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G3$/) {$oe9g3++; print RE "$num\tINsig_$$pL[8]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G4$/) {$oe9g4++; print RE "$num\tINsig_$$pL[8]\t$cate\t$grp\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $oe9out++; print RE "$num\tINsig_$$pL[8]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {
                        $oe9out++; print RE "$num\tINsig_$$pL[8]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/) {$oe9nh++; print RE "$num\tINsig_$$pL[8]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } else {
                print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
        }
} elsif ($prim =~ /^$$pL[9]$/) { # E21
        if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                if    ($grp =~ /^G1$/) {
                        $e10g1++; print RE "$num\t$$pL[9]\t$cate\t$grp\n"; print LL2 "$num\n";}
                elsif ($grp =~ /^G2$/) {
                        $e10g2++; print RE "$num\t$$pL[9]\t$cate\t$grp\n"; print LL2 "$num\n";}
                elsif ($grp =~ /^G3$/) {
                        $e10g3++; print RE "$num\t$$pL[9]\t$cate\t$grp\n"; print LL2 "$num\n";}
                elsif ($grp =~ /^G4$/) {
                        $e10g4++; print RE "$num\t$$pL[9]\t$cate\t$grp\n"; print LL2 "$num\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $e10out++; print RE "$num\t$$pL[9]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {$e10out++; print RE "$num\t$$pL[9]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/) {$e10nh++; print RE "$num\t$$pL[9]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } elsif (($cate =~ /^OUT(.*)$/)) {
                if    ($grp =~ /^G1$/) {$oe10g1++; print RE "$num\tINsig_$$pL[9]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G2$/) {$oe10g2++; print RE "$num\tINsig_$$pL[9]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G3$/) {$oe10g3++; print RE "$num\tINsig_$$pL[9]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G4$/) {$oe10g4++; print RE "$num\tINsig_$$pL[9]\t$cate\t$grp\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $oe10out++; print RE "$num\tINsig_$$pL[9]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {
                        $oe10out++; print RE "$num\tINsig_$$pL[9]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/) {$oe10nh++; print RE "$num\tINsig_$$pL[9]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } else {
                print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
        }
} elsif ($prim =~ /^$$pL[10]$/) {
        if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                if    ($grp =~ /^G1$/) {
                        $e11g1++; print RE "$num\t$$pL[10]\t$cate\t$grp\n"; print LL3 "$num\n";}
                elsif ($grp =~ /^G2$/) {
                        $e11g2++; print RE "$num\t$$pL[10]\t$cate\t$grp\n"; print LL3 "$num\n";}
                elsif ($grp =~ /^G3$/) {
                        $e11g3++; print RE "$num\t$$pL[10]\t$cate\t$grp\n"; print LL3 "$num\n";}
                elsif ($grp =~ /^G4$/) {
                        $e11g4++; print RE "$num\t$$pL[10]\t$cate\t$grp\n"; print LL3 "$num\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $e11out++; print RE "$num\t$$pL[10]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {$e11out++; print RE "$num\t$$pL[10]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/) {$e11nh++; print RE "$num\t$$pL[10]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } elsif (($cate =~ /^OUT(.*)$/)) {
                if    ($grp =~ /^G1$/) {$oe11g1++; print RE "$num\tINsig_$$pL[10]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G2$/) {$oe11g2++; print RE "$num\tINsig_$$pL[10]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G3$/) {$oe11g3++; print RE "$num\tINsig_$$pL[10]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G4$/) {$oe11g4++; print RE "$num\tINsig_$$pL[10]\t$cate\t$grp\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $oe11out++; print RE "$num\tINsig_$$pL[10]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {
                        $oe11out++; print RE "$num\tINsig_$$pL[10]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/){$oe11nh++; print RE "$num\tINsig_$$pL[10]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } else {
                print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
        }
} elsif ($prim =~ /^$$pL[11]$/) {
        if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                if    ($grp =~ /^G1$/) {
                        $e12g1++; print RE "$num\t$$pL[11]\t$cate\t$grp\n"; print LL4 "$num\n";}
                elsif ($grp =~ /^G2$/) {
                        $e12g2++; print RE "$num\t$$pL[11]\t$cate\t$grp\n"; print LL4 "$num\n";}
                elsif ($grp =~ /^G3$/) {
                        $e12g3++; print RE "$num\t$$pL[11]\t$cate\t$grp\n"; print LL4 "$num\n";}
                elsif ($grp =~ /^G4$/) {
                        $e12g4++; print RE "$num\t$$pL[11]\t$cate\t$grp\n"; print LL4 "$num\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $e12out++; print RE "$num\t$$pL[11]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {$e12out++; print RE "$num\t$$pL[11]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/) {$e12nh++; print RE "$num\t$$pL[11]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } elsif (($cate =~ /^OUT(.*)$/)) {
                if    ($grp =~ /^G1$/) {$oe12g1++; print RE "$num\tINsig_$$pL[11]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G2$/) {$oe12g2++; print RE "$num\tINsig_$$pL[11]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G3$/) {$oe12g3++; print RE "$num\tINsig_$$pL[11]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G4$/) {$oe12g4++; print RE "$num\tINsig_$$pL[11]\t$cate\t$grp\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $oe12out++; print RE "$num\tINsig_$$pL[11]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {
                        $oe12out++; print RE "$num\tINsig_$$pL[11]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/){$oe12nh++; print RE "$num\tINsig_$$pL[11]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } else {
                print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
        }
} elsif ($prim =~ /^$$pL[12]$/) {
        if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                if    ($grp =~ /^G1$/) {
                        $e13g1++; print RE "$num\t$$pL[12]\t$cate\t$grp\n"; print LL5 "$num\n";}
                elsif ($grp =~ /^G2$/) {
                        $e13g2++; print RE "$num\t$$pL[12]\t$cate\t$grp\n"; print LL5 "$num\n";}
                elsif ($grp =~ /^G3$/) {
                        $e13g3++; print RE "$num\t$$pL[12]\t$cate\t$grp\n"; print LL5 "$num\n";}
                elsif ($grp =~ /^G4$/) {
                        $e13g4++; print RE "$num\t$$pL[12]\t$cate\t$grp\n"; print LL5 "$num\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $e13out++; print RE "$num\t$$pL[12]\t$cate\t$grp\n";}
```

```
                        elsif ($grp =~ /^no_group$/) {$e13out++; print RE "$num\t$$pL[12]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e13nh++; print RE "$num\t$$pL[12]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe13g1++; print RE "$num\tINsig_$$pL[12]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe13g2++; print RE "$num\tINsig_$$pL[12]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe13g3++; print RE "$num\tINsig_$$pL[12]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe13g4++; print RE "$num\tINsig_$$pL[12]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe13out++; print RE "$num\tINsig_$$pL[12]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe13out++; print RE "$num\tINsig_$$pL[12]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/){$oe13nh++; print RE "$num\tINsig_$$pL[12]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[13]$/) {
                if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e14g1++; print RE "$num\t$$pL[13]\t$cate\t$grp\n"; print LL6 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e14g2++; print RE "$num\t$$pL[13]\t$cate\t$grp\n"; print LL6 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e14g3++; print RE "$num\t$$pL[13]\t$cate\t$grp\n"; print LL6 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e14g4++; print RE "$num\t$$pL[13]\t$cate\t$grp\n"; print LL6 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e14out++; print RE "$num\t$$pL[13]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e14out++; print RE "$num\t$$pL[13]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e14nh++; print RE "$num\t$$pL[13]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe14g1++; print RE "$num\tINsig_$$pL[13]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe14g2++; print RE "$num\tINsig_$$pL[13]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe14g3++; print RE "$num\tINsig_$$pL[13]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe14g4++; print RE "$num\tINsig_$$pL[13]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe14out++; print RE "$num\tINsig_$$pL[13]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe14out++; print RE "$num\tINsig_$$pL[13]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/){$oe14nh++; print RE "$num\tINsig_$$pL[13]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[14]$/) {
                if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e15g1++; print RE "$num\t$$pL[14]\t$cate\t$grp\n"; print LL7 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e15g2++; print RE "$num\t$$pL[14]\t$cate\t$grp\n"; print LL7 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e15g3++; print RE "$num\t$$pL[14]\t$cate\t$grp\n"; print LL7 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e15g4++; print RE "$num\t$$pL[14]\t$cate\t$grp\n"; print LL7 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e15out++; print RE "$num\t$$pL[14]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e15out++; print RE "$num\t$$pL[14]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e15nh++; print RE "$num\t$$pL[14]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe15g1++; print RE "$num\tINsig_$$pL[14]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe15g2++; print RE "$num\tINsig_$$pL[14]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe15g3++; print RE "$num\tINsig_$$pL[14]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe15g4++; print RE "$num\tINsig_$$pL[14]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe15out++; print RE "$num\tINsig_$$pL[14]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe15out++; print RE "$num\tINsig_$$pL[14]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/){$oe15nh++; print RE "$num\tINsig_$$pL[14]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[15]$/) {
                if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e16g1++; print RE "$num\t$$pL[15]\t$cate\t$grp\n"; print LL8 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e16g2++; print RE "$num\t$$pL[15]\t$cate\t$grp\n"; print LL8 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e16g3++; print RE "$num\t$$pL[15]\t$cate\t$grp\n"; print LL8 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e16g4++; print RE "$num\t$$pL[15]\t$cate\t$grp\n"; print LL8 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e16out++; print RE "$num\t$$pL[15]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e16out++; print RE "$num\t$$pL[15]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e16nh++; print RE "$num\t$$pL[15]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe16g1++; print RE "$num\tINsig_$$pL[15]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe16g2++; print RE "$num\tINsig_$$pL[15]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe16g3++; print RE "$num\tINsig_$$pL[15]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe16g4++; print RE "$num\tINsig_$$pL[15]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe16out++; print RE "$num\tINsig_$$pL[15]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe16out++; print RE "$num\tINsig_$$pL[15]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/){$oe16nh++; print RE "$num\tINsig_$$pL[15]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[16]$/) {
                if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e18g1++; print RE "$num\t$$pL[16]\t$cate\t$grp\n"; print LE18 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e18g2++; print RE "$num\t$$pL[16]\t$cate\t$grp\n"; print LE18 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e18g3++; print RE "$num\t$$pL[16]\t$cate\t$grp\n"; print LE18 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e18g4++; print RE "$num\t$$pL[16]\t$cate\t$grp\n"; print LE18 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e18out++; print RE "$num\t$$pL[16]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e18out++; print RE "$num\t$$pL[16]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e18nh++; print RE "$num\t$$pL[16]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe18g1++; print RE "$num\tINsig_$$pL[16]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe18g2++; print RE "$num\tINsig_$$pL[16]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe18g3++; print RE "$num\tINsig_$$pL[16]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe18g4++; print RE "$num\tINsig_$$pL[16]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe18out++; print RE "$num\tINsig_$$pL[16]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe18out++; print RE "$num\tINsig_$$pL[16]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/){$oe18nh++; print RE "$num\tINsig_$$pL[16]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[17]$/) {
                if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e19g1++; print RE "$num\t$$pL[17]\t$cate\t$grp\n"; print LE19 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e19g2++; print RE "$num\t$$pL[17]\t$cate\t$grp\n"; print LE19 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e19g3++; print RE "$num\t$$pL[17]\t$cate\t$grp\n"; print LE19 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e19g4++; print RE "$num\t$$pL[17]\t$cate\t$grp\n"; print LE19 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e19out++; print RE "$num\t$$pL[17]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e19out++; print RE "$num\t$$pL[17]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e19nh++; print RE "$num\t$$pL[17]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe19g1++; print RE "$num\tINsig_$$pL[17]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe19g2++; print RE "$num\tINsig_$$pL[17]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe19g3++; print RE "$num\tINsig_$$pL[17]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe19g4++; print RE "$num\tINsig_$$pL[17]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe19out++; print RE "$num\tINsig_$$pL[17]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
```

```
                        $oe19out++; print RE "$num\tINsig_$$pL[17]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/){$oe19nh++; print RE "$num\tINsig_$$pL[17]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } else {
                print "\n\nThere is error in PrimerMatchingCategory...$num\n\n";
                exit;
        }
} elsif ($prim =~ /^$$pL[18]$/) {
        if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                if    ($grp =~ /^G1$/) {
                        $e20g1++; print RE "$num\t$$pL[18]\t$cate\t$grp\n"; print LE20 "$num\n";}
                elsif ($grp =~ /^G2$/) {
                        $e20g2++; print RE "$num\t$$pL[18]\t$cate\t$grp\n"; print LE20 "$num\n";}
                elsif ($grp =~ /^G3$/) {
                        $e20g3++; print RE "$num\t$$pL[18]\t$cate\t$grp\n"; print LE20 "$num\n";}
                elsif ($grp =~ /^G4$/) {
                        $e20g4++; print RE "$num\t$$pL[18]\t$cate\t$grp\n"; print LE20 "$num\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $e20out++; print RE "$num\t$$pL[18]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {$e20out++; print RE "$num\t$$pL[18]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/) {$e20nh++; print RE "$num\t$$pL[18]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } elsif (($cate =~ /^OUT(.*)$/)) {
                if    ($grp =~ /^G1$/) {$oe20g1++; print RE "$num\tINsig_$$pL[18]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G2$/) {$oe20g2++; print RE "$num\tINsig_$$pL[18]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G3$/) {$oe20g3++; print RE "$num\tINsig_$$pL[18]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G4$/) {$oe20g4++; print RE "$num\tINsig_$$pL[18]\t$cate\t$grp\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $oe20out++; print RE "$num\tINsig_$$pL[18]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {
                        $oe20out++; print RE "$num\tINsig_$$pL[18]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/){$oe20nh++; print RE "$num\tINsig_$$pL[18]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } else {
                print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
        }
} elsif ($prim =~ /^$$pL[19]$/) {
        if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                if    ($grp =~ /^G1$/) {
                        $e22g1++; print RE "$num\t$$pL[19]\t$cate\t$grp\n"; print LE22 "$num\n";}
                elsif ($grp =~ /^G2$/) {
                        $e22g2++; print RE "$num\t$$pL[19]\t$cate\t$grp\n"; print LE22 "$num\n";}
                elsif ($grp =~ /^G3$/) {
                        $e22g3++; print RE "$num\t$$pL[19]\t$cate\t$grp\n"; print LE22 "$num\n";}
                elsif ($grp =~ /^G4$/) {
                        $e22g4++; print RE "$num\t$$pL[19]\t$cate\t$grp\n"; print LE22 "$num\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $e22out++; print RE "$num\t$$pL[19]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {$e22out++; print RE "$num\t$$pL[19]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/) {$e22nh++; print RE "$num\t$$pL[19]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } elsif (($cate =~ /^OUT(.*)$/)) {
                if    ($grp =~ /^G1$/) {$oe22g1++; print RE "$num\tINsig_$$pL[19]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G2$/) {$oe22g2++; print RE "$num\tINsig_$$pL[19]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G3$/) {$oe22g3++; print RE "$num\tINsig_$$pL[19]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G4$/) {$oe22g4++; print RE "$num\tINsig_$$pL[19]\t$cate\t$grp\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $oe22out++; print RE "$num\tINsig_$$pL[19]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {
                        $oe22out++; print RE "$num\tINsig_$$pL[19]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/){$oe22nh++; print RE "$num\tINsig_$$pL[19]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } else {
                print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
        }
} elsif ($prim =~ /^$$pL[20]$/) {
        if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                if    ($grp =~ /^G1$/) {
                        $e23g1++; print RE "$num\t$$pL[20]\t$cate\t$grp\n"; print LE23 "$num\n";}
                elsif ($grp =~ /^G2$/) {
                        $e23g2++; print RE "$num\t$$pL[20]\t$cate\t$grp\n"; print LE23 "$num\n";}
                elsif ($grp =~ /^G3$/) {
                        $e23g3++; print RE "$num\t$$pL[20]\t$cate\t$grp\n"; print LE23 "$num\n";}
                elsif ($grp =~ /^G4$/) {
                        $e23g4++; print RE "$num\t$$pL[20]\t$cate\t$grp\n"; print LE23 "$num\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $e23out++; print RE "$num\t$$pL[20]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {
                        $e23out++; print RE "$num\t$$pL[20]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/) {$e23nh++; print RE "$num\t$$pL[20]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } elsif (($cate =~ /^OUT(.*)$/)) {
                if    ($grp =~ /^G1$/) {$oe23g1++; print RE "$num\tINsig_$$pL[20]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G2$/) {$oe23g2++; print RE "$num\tINsig_$$pL[20]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G3$/) {$oe23g3++; print RE "$num\tINsig_$$pL[20]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G4$/) {$oe23g4++; print RE "$num\tINsig_$$pL[20]\t$cate\t$grp\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $oe23out++; print RE "$num\tINsig_$$pL[20]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {
                        $oe23out++; print RE "$num\tINsig_$$pL[20]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/){$oe23nh++; print RE "$num\tINsig_$$pL[20]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } else {
                print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
        }
} elsif ($prim =~ /^$$pL[21]$/) {
        if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                if    ($grp =~ /^G1$/) {
                        $e24g1++; print RE "$num\t$$pL[21]\t$cate\t$grp\n"; print LE24 "$num\n";}
                elsif ($grp =~ /^G2$/) {
                        $e24g2++; print RE "$num\t$$pL[21]\t$cate\t$grp\n"; print LE24 "$num\n";}
                elsif ($grp =~ /^G3$/) {
                        $e24g3++; print RE "$num\t$$pL[21]\t$cate\t$grp\n"; print LE24 "$num\n";}
                elsif ($grp =~ /^G4$/) {
                        $e24g4++; print RE "$num\t$$pL[21]\t$cate\t$grp\n"; print LE24 "$num\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $e24out++; print RE "$num\t$$pL[21]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {$e24out++; print RE "$num\t$$pL[21]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/) {$e24nh++; print RE "$num\t$$pL[21]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } elsif (($cate =~ /^OUT(.*)$/)) {
                if    ($grp =~ /^G1$/) {$oe24g1++; print RE "$num\tINsig_$$pL[21]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G2$/) {$oe24g2++; print RE "$num\tINsig_$$pL[21]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G3$/) {$oe24g3++; print RE "$num\tINsig_$$pL[21]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G4$/) {$oe24g4++; print RE "$num\tINsig_$$pL[21]\t$cate\t$grp\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $oe24out++; print RE "$num\tINsig_$$pL[21]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {
                        $oe24out++; print RE "$num\tINsig_$$pL[21]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/){$oe24nh++; print RE "$num\tINsig_$$pL[21]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } else {
                print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
        }
} elsif ($prim =~ /^$$pL[22]$/) {
        if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                if    ($grp =~ /^G1$/) {
                        $e25g1++; print RE "$num\t$$pL[22]\t$cate\t$grp\n"; print LE25 "$num\n";}
                elsif ($grp =~ /^G2$/) {
                        $e25g2++; print RE "$num\t$$pL[22]\t$cate\t$grp\n"; print LE25 "$num\n";}
                elsif ($grp =~ /^G3$/) {
                        $e25g3++; print RE "$num\t$$pL[22]\t$cate\t$grp\n"; print LE25 "$num\n";}
                elsif ($grp =~ /^G4$/) {
                        $e25g4++; print RE "$num\t$$pL[22]\t$cate\t$grp\n"; print LE25 "$num\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $e25out++; print RE "$num\t$$pL[22]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {$e25out++; print RE "$num\t$$pL[22]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/) {$e25nh++; print RE "$num\t$$pL[22]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } elsif (($cate =~ /^OUT(.*)$/)) {
                if    ($grp =~ /^G1$/) {$oe25g1++; print RE "$num\tINsig_$$pL[22]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G2$/) {$oe25g2++; print RE "$num\tINsig_$$pL[22]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G3$/) {$oe25g3++; print RE "$num\tINsig_$$pL[22]\t$cate\t$grp\n";}
                elsif ($grp =~ /^G4$/) {$oe25g4++; print RE "$num\tINsig_$$pL[22]\t$cate\t$grp\n";}
                elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                        $oe25out++; print RE "$num\tINsig_$$pL[22]\t$cate\t$grp\n";}
                elsif ($grp =~ /^no_group$/) {
                        $oe25out++; print RE "$num\tINsig_$$pL[22]\t$cate\t$grp\n";}
                elsif ($grp =~ /^noHPV$/){$oe25nh++; print RE "$num\tINsig_$$pL[22]\t$cate\t$grp\n";}
                else {print "\n\nError\n\n"; exit;}
        } else {
                print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
        }
} elsif ($prim =~ /^$$pL[23]$/) {
        if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                if    ($grp =~ /^G1$/) {
```

```
                                $e26g1++; print RE "$num\t$$pL[23]\t$cate\t$grp\n"; print LE26 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e26g2++; print RE "$num\t$$pL[23]\t$cate\t$grp\n"; print LE26 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e26g3++; print RE "$num\t$$pL[23]\t$cate\t$grp\n"; print LE26 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e26g4++; print RE "$num\t$$pL[23]\t$cate\t$grp\n"; print LE26 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e26out++; print RE "$num\t$$pL[23]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e26out++; print RE "$num\t$$pL[23]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e26nh++; print RE "$num\t$$pL[23]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe26g1++; print RE "$num\tINsig_$$pL[23]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe26g2++; print RE "$num\tINsig_$$pL[23]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe26g3++; print RE "$num\tINsig_$$pL[23]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe26g4++; print RE "$num\tINsig_$$pL[23]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe26out++; print RE "$num\tINsig_$$pL[23]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe26out++; print RE "$num\tINsig_$$pL[23]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/){$oe26nh++; print RE "$num\tINsig_$$pL[23]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[24]$/) {
                if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e27g1++; print RE "$num\t$$pL[24]\t$cate\t$grp\n"; print LE27 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e27g2++; print RE "$num\t$$pL[24]\t$cate\t$grp\n"; print LE27 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e27g3++; print RE "$num\t$$pL[24]\t$cate\t$grp\n"; print LE27 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e27g4++; print RE "$num\t$$pL[24]\t$cate\t$grp\n"; print LE27 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e27out++; print RE "$num\t$$pL[24]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e27out++; print RE "$num\t$$pL[24]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e27nh++; print RE "$num\t$$pL[24]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe27g1++; print RE "$num\tINsig_$$pL[24]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe27g2++; print RE "$num\tINsig_$$pL[24]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe27g3++; print RE "$num\tINsig_$$pL[24]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe27g4++; print RE "$num\tINsig_$$pL[24]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe27out++; print RE "$num\tINsig_$$pL[24]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe27out++; print RE "$num\tINsig_$$pL[24]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/){$oe27nh++; print RE "$num\tINsig_$$pL[24]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[25]$/) {
                if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e28g1++; print RE "$num\t$$pL[25]\t$cate\t$grp\n"; print LE28 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e28g2++; print RE "$num\t$$pL[25]\t$cate\t$grp\n"; print LE28 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e28g3++; print RE "$num\t$$pL[25]\t$cate\t$grp\n"; print LE28 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e28g4++; print RE "$num\t$$pL[25]\t$cate\t$grp\n"; print LE28 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e28out++; print RE "$num\t$$pL[25]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e28out++; print RE "$num\t$$pL[25]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e28nh++; print RE "$num\t$$pL[25]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe28g1++; print RE "$num\tINsig_$$pL[25]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe28g2++; print RE "$num\tINsig_$$pL[25]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe28g3++; print RE "$num\tINsig_$$pL[25]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe28g4++; print RE "$num\tINsig_$$pL[25]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe28out++; print RE "$num\tINsig_$$pL[25]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe28out++; print RE "$num\tINsig_$$pL[25]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/){$oe28nh++; print RE "$num\tINsig_$$pL[25]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[26]$/) {
                if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e29g1++; print RE "$num\t$$pL[26]\t$cate\t$grp\n"; print LE29 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e29g2++; print RE "$num\t$$pL[26]\t$cate\t$grp\n"; print LE29 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e29g3++; print RE "$num\t$$pL[26]\t$cate\t$grp\n"; print LE29 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e29g4++; print RE "$num\t$$pL[26]\t$cate\t$grp\n"; print LE29 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e29out++; print RE "$num\t$$pL[26]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e29out++; print RE "$num\t$$pL[26]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/){$e29nh++; print RE "$num\t$$pL[26]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe29g1++; print RE "$num\tINsig_$$pL[26]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe29g2++; print RE "$num\tINsig_$$pL[26]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe29g3++; print RE "$num\tINsig_$$pL[26]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe29g4++; print RE "$num\tINsig_$$pL[26]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe29out++; print RE "$num\tINsig_$$pL[26]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe29out++; print RE "$num\tINsig_$$pL[26]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/){$oe29nh++; print RE "$num\tINsig_$$pL[26]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[27]$/) {
                if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e30g1++; print RE "$num\t$$pL[27]\t$cate\t$grp\n"; print LE30 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e30g2++; print RE "$num\t$$pL[27]\t$cate\t$grp\n"; print LE30 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e30g3++; print RE "$num\t$$pL[27]\t$cate\t$grp\n"; print LE30 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e30g4++; print RE "$num\t$$pL[27]\t$cate\t$grp\n"; print LE30 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e30out++; print RE "$num\t$$pL[27]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e30out++; print RE "$num\t$$pL[27]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e30nh++; print RE "$num\t$$pL[27]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe30g1++; print RE "$num\tINsig_$$pL[27]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe30g2++; print RE "$num\tINsig_$$pL[27]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe30g3++; print RE "$num\tINsig_$$pL[27]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe30g4++; print RE "$num\tINsig_$$pL[27]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe30out++; print RE "$num\tINsig_$$pL[27]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe30out++; print RE "$num\tINsig_$$pL[27]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/){$oe30nh++; print RE "$num\tINsig_$$pL[27]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }
        } elsif ($prim =~ /^$$pL[28]$/) {
                if (($cate =~ /^complete$/)|($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e31g1++; print RE "$num\t$$pL[28]\t$cate\t$grp\n"; print LE31 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e31g2++; print RE "$num\t$$pL[28]\t$cate\t$grp\n"; print LE31 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e31g3++; print RE "$num\t$$pL[28]\t$cate\t$grp\n"; print LE31 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e31g4++; print RE "$num\t$$pL[28]\t$cate\t$grp\n"; print LE31 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e31out++; print RE "$num\t$$pL[28]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e31out++; print RE "$num\t$$pL[28]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e31nh++; print RE "$num\t$$pL[28]\t$cate\t$grp\n";}
```

```
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe31g1++; print RE "$num\tINsig_$$pL[28]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe31g2++; print RE "$num\tINsig_$$pL[28]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe31g3++; print RE "$num\tINsig_$$pL[28]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe31g4++; print RE "$num\tINsig_$$pL[28]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe31out++; print RE "$num\tINsig_$$pL[28]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe31out++; print RE "$num\tINsig_$$pL[28]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/){$oe31nh++; print RE "$num\tINsig_$$pL[28]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }

        } elsif ($prim =~ /^$$pL[29]$/) {
                if (($cate =~ /^complete$/)||($cate =~ /^INcom(.*)$/)) {
                        if    ($grp =~ /^G1$/) {
                                $e32g1++; print RE "$num\t$$pL[29]\t$cate\t$grp\n"; print LE32 "$num\n";}
                        elsif ($grp =~ /^G2$/) {
                                $e32g2++; print RE "$num\t$$pL[29]\t$cate\t$grp\n"; print LE32 "$num\n";}
                        elsif ($grp =~ /^G3$/) {
                                $e32g3++; print RE "$num\t$$pL[29]\t$cate\t$grp\n"; print LE32 "$num\n";}
                        elsif ($grp =~ /^G4$/) {
                                $e32g4++; print RE "$num\t$$pL[29]\t$cate\t$grp\n"; print LE32 "$num\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $e32out++; print RE "$num\t$$pL[29]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$e32out++; print RE "$num\t$$pL[29]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$e32nh++; print RE "$num\t$$pL[29]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } elsif (($cate =~ /^OUT(.*)$/)) {
                        if    ($grp =~ /^G1$/) {$oe32g1++; print RE "$num\tINsig_$$pL[29]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$oe32g2++; print RE "$num\tINsig_$$pL[29]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$oe32g3++; print RE "$num\tINsig_$$pL[29]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$oe32g4++; print RE "$num\tINsig_$$pL[29]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $oe32out++; print RE "$num\tINsig_$$pL[29]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {
                                $oe32out++; print RE "$num\tINsig_$$pL[29]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/){$oe32nh++; print RE "$num\tINsig_$$pL[29]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...$num\n\n"; exit;
                }

        } elsif ($prim =~ /^$$pL[30]$/) {
                if (($cate =~ /^noPrimer$/)) {
                        if    ($grp =~ /^G1$/) {$npg1++; print RE "$num\t$$pL[30]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G2$/) {$npg2++; print RE "$num\t$$pL[30]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G3$/) {$npg3++; print RE "$num\t$$pL[30]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^G4$/) {$npg4++; print RE "$num\t$$pL[30]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^out_ranged_G[0-9]$/) {
                                $npout++; print RE "$num\t$$pL[30]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^no_group$/) {$npout++; print RE "$num\t$$pL[30]\t$cate\t$grp\n";}
                        elsif ($grp =~ /^noHPV$/) {$npnh++; print RE "$num\t$$pL[30]\t$cate\t$grp\n";}
                        else {print "\n\nError\n\n"; exit;}
                } else {
                        print "\n\nThere is error in PrimerMatchingCategory...\n\n"; exit;
                }

        } else {
                print "\n\nThere is some error in primer name...$num\n\n"; exit;
        }

}

close RE;
close LE1;close LE2;close LE3;close LE4;close LE5;close LE6;close LE7;close LE8;
close LL1;close LL2;close LL3;close LL4;close LL5;close LL6;close LL7;close LL8;
close LE18;close LE19;close LE20;close LE22;close LE23;close LE24;close LE25;close LE26;
close LE27;close LE28;close LE29;close LE30;close LE31;close LE32;

unless (open (SP, ">$outsigPrim")) {print "\n\n\n\nCan not create ".$outsigPrim."\n\n\n\n"; exit;}

print SP "Barcode_$barcodenumber\n";
print SP "For_$prefix\n\n";

        print SP "Primer\thpvG1_sigP\thpvG2_sigP\thpvG3_sigP\thpvG4_sigP\thpvOutranged_sigP\tnoHPV_sigP";
        print SP "\thpvG1_outP\thpvG2_outP\thpvG3_outP\thpvG4_outP\thpvOutranged_outP\tnoHPV_outP\n";

        print SP $$pL[0]."\t".$e1g1."\t".$e1g2."\t".$e1g3."\t".$e1g4."\t".$e1out."\t".$e1nh.
                        "\t".$oe1g1."\t".$oe1g2."\t".$oe1g3."\t".$oe1g4."\t".$oe1out."\t".$oe1nh."\n";
        print SP $$pL[1]."\t".$e2g1."\t".$e2g2."\t".$e2g3."\t".$e2g4."\t".$e2out."\t".$e2nh.
                        "\t".$oe2g1."\t".$oe2g2."\t".$oe2g3."\t".$oe2g4."\t".$oe2out."\t".$oe2nh."\n";
        print SP $$pL[2]."\t".$e3g1."\t".$e3g2."\t".$e3g3."\t".$e3g4."\t".$e3out."\t".$e3nh.
                        "\t".$oe3g1."\t".$oe3g2."\t".$oe3g3."\t".$oe3g4."\t".$oe3out."\t".$oe3nh."\n";
        print SP $$pL[3]."\t".$e4g1."\t".$e4g2."\t".$e4g3."\t".$e4g4."\t".$e4out."\t".$e4nh.
                        "\t".$oe4g1."\t".$oe4g2."\t".$oe4g3."\t".$oe4g4."\t".$oe4out."\t".$oe4nh."\n";
        print SP $$pL[4]."\t".$e5g1."\t".$e5g2."\t".$e5g3."\t".$e5g4."\t".$e5out."\t".$e5nh.
                        "\t".$oe5g1."\t".$oe5g2."\t".$oe5g3."\t".$oe5g4."\t".$oe5out."\t".$oe5nh."\n";
        print SP $$pL[5]."\t".$e6g1."\t".$e6g2."\t".$e6g3."\t".$e6g4."\t".$e6out."\t".$e6nh.
                        "\t".$oe6g1."\t".$oe6g2."\t".$oe6g3."\t".$oe6g4."\t".$oe6out."\t".$oe6nh."\n";
        print SP $$pL[6]."\t".$e7g1."\t".$e7g2."\t".$e7g3."\t".$e7g4."\t".$e7out."\t".$e7nh.
                        "\t".$oe7g1."\t".$oe7g2."\t".$oe7g3."\t".$oe7g4."\t".$oe7out."\t".$oe7nh."\n";
        print SP $$pL[7]."\t".$e8g1."\t".$e8g2."\t".$e8g3."\t".$e8g4."\t".$e8out."\t".$e8nh.
                        "\t".$oe8g1."\t".$oe8g2."\t".$oe8g3."\t".$oe8g4."\t".$oe8out."\t".$oe8nh."\n";
        print SP $$pL[8]."\t".$e9g1."\t".$e9g2."\t".$e9g3."\t".$e9g4."\t".$e9out."\t".$e9nh.
                        "\t".$oe9g1."\t".$oe9g2."\t".$oe9g3."\t".$oe9g4."\t".$oe9out."\t".$oe9nh."\n";
        print SP $$pL[9]."\t".$e10g1."\t".$e10g2."\t".$e10g3."\t".$e10g4."\t".$e10out."\t".$e10nh.
                        "\t".$oe10g1."\t".$oe10g2."\t".$oe10g3."\t".$oe10g4."\t".$oe10out."\t".$oe10nh."\n";
        print SP $$pL[10]."\t".$e11g1."\t".$e11g2."\t".$e11g3."\t".$e11g4."\t".$e11out."\t".$e11nh.
                        "\t".$oe11g1."\t".$oe11g2."\t".$oe11g3."\t".$oe11g4."\t".$oe11out."\t".$oe11nh."\n";
        print SP $$pL[11]."\t".$e12g1."\t".$e12g2."\t".$e12g3."\t".$e12g4."\t".$e12out."\t".$e12nh.
                        "\t".$oe12g1."\t".$oe12g2."\t".$oe12g3."\t".$oe12g4."\t".$oe12out."\t".$oe12nh."\n";
        print SP $$pL[12]."\t".$e13g1."\t".$e13g2."\t".$e13g3."\t".$e13g4."\t".$e13out."\t".$e13nh.
                        "\t".$oe13g1."\t".$oe13g2."\t".$oe13g3."\t".$oe13g4."\t".$oe13out."\t".$oe13nh."\n";
        print SP $$pL[13]."\t".$e14g1."\t".$e14g2."\t".$e14g3."\t".$e14g4."\t".$e14out."\t".$e14nh.
                        "\t".$oe14g1."\t".$oe14g2."\t".$oe14g3."\t".$oe14g4."\t".$oe14out."\t".$oe14nh."\n";
        print SP $$pL[14]."\t".$e15g1."\t".$e15g2."\t".$e15g3."\t".$e15g4."\t".$e15out."\t".$e15nh.
                        "\t".$oe15g1."\t".$oe15g2."\t".$oe15g3."\t".$oe15g4."\t".$oe15out."\t".$oe15nh."\n";
        print SP $$pL[15]."\t".$e16g1."\t".$e16g2."\t".$e16g3."\t".$e16g4."\t".$e16out."\t".$e16nh.
                        "\t".$oe16g1."\t".$oe16g2."\t".$oe16g3."\t".$oe16g4."\t".$oe16out."\t".$oe16nh."\n";
        print SP $$pL[16]."\t".$e18g1."\t".$e18g2."\t".$e18g3."\t".$e18g4."\t".$e18out."\t".$e18nh.
                        "\t".$oe18g1."\t".$oe18g2."\t".$oe18g3."\t".$oe18g4."\t".$oe18out."\t".$oe18nh."\n";
        print SP $$pL[17]."\t".$e19g1."\t".$e19g2."\t".$e19g3."\t".$e19g4."\t".$e19out."\t".$e19nh.
                        "\t".$oe19g1."\t".$oe19g2."\t".$oe19g3."\t".$oe19g4."\t".$oe19out."\t".$oe19nh."\n";
        print SP $$pL[18]."\t".$e20g1."\t".$e20g2."\t".$e20g3."\t".$e20g4."\t".$e20out."\t".$e20nh.
                        "\t".$oe20g1."\t".$oe20g2."\t".$oe20g3."\t".$oe20g4."\t".$oe20out."\t".$oe20nh."\n";
        print SP $$pL[19]."\t".$e22g1."\t".$e22g2."\t".$e22g3."\t".$e22g4."\t".$e22out."\t".$e22nh.
                        "\t".$oe22g1."\t".$oe22g2."\t".$oe22g3."\t".$oe22g4."\t".$oe22out."\t".$oe22nh."\n";
        print SP $$pL[20]."\t".$e23g1."\t".$e23g2."\t".$e23g3."\t".$e23g4."\t".$e23out."\t".$e23nh.
                        "\t".$oe23g1."\t".$oe23g2."\t".$oe23g3."\t".$oe23g4."\t".$oe23out."\t".$oe23nh."\n";
        print SP $$pL[21]."\t".$e24g1."\t".$e24g2."\t".$e24g3."\t".$e24g4."\t".$e24out."\t".$e24nh.
                        "\t".$oe24g1."\t".$oe24g2."\t".$oe24g3."\t".$oe24g4."\t".$oe24out."\t".$oe24nh."\n";
        print SP $$pL[22]."\t".$e25g1."\t".$e25g2."\t".$e25g3."\t".$e25g4."\t".$e25out."\t".$e25nh.
                        "\t".$oe25g1."\t".$oe25g2."\t".$oe25g3."\t".$oe25g4."\t".$oe25out."\t".$oe25nh."\n";
        print SP $$pL[23]."\t".$e26g1."\t".$e26g2."\t".$e26g3."\t".$e26g4."\t".$e26out."\t".$e26nh.
                        "\t".$oe26g1."\t".$oe26g2."\t".$oe26g3."\t".$oe26g4."\t".$oe26out."\t".$oe26nh."\n";
        print SP $$pL[24]."\t".$e27g1."\t".$e27g2."\t".$e27g3."\t".$e27g4."\t".$e27out."\t".$e27nh.
                        "\t".$oe27g1."\t".$oe27g2."\t".$oe27g3."\t".$oe27g4."\t".$oe27out."\t".$oe27nh."\n";
        print SP $$pL[25]."\t".$e28g1."\t".$e28g2."\t".$e28g3."\t".$e28g4."\t".$e28out."\t".$e28nh.
                        "\t".$oe28g1."\t".$oe28g2."\t".$oe28g3."\t".$oe28g4."\t".$oe28out."\t".$oe28nh."\n";
        print SP $$pL[26]."\t".$e29g1."\t".$e29g2."\t".$e29g3."\t".$e29g4."\t".$e29out."\t".$e29nh.
                        "\t".$oe29g1."\t".$oe29g2."\t".$oe29g3."\t".$oe29g4."\t".$oe29out."\t".$oe29nh."\n";
        print SP $$pL[27]."\t".$e30g1."\t".$e30g2."\t".$e30g3."\t".$e30g4."\t".$e30out."\t".$e30nh.
                        "\t".$oe30g1."\t".$oe30g2."\t".$oe30g3."\t".$oe30g4."\t".$oe30out."\t".$oe30nh."\n";
        print SP $$pL[28]."\t".$e31g1."\t".$e31g2."\t".$e31g3."\t".$e31g4."\t".$e31out."\t".$e31nh.
                        "\t".$oe31g1."\t".$oe31g2."\t".$oe31g3."\t".$oe31g4."\t".$oe31out."\t".$oe31nh."\n";
        print SP $$pL[29]."\t".$e32g1."\t".$e32g2."\t".$e32g3."\t".$e32g4."\t".$e32out."\t".$e32nh.
                        "\t".$oe32g1."\t".$oe32g2."\t".$oe32g3."\t".$oe32g4."\t".$oe32out."\t".$oe32nh."\n\n\n";

        print SP "noPrimerHit\thpvG1\thpvG2\thpvG3\thpvG4\thpvOutranged\tnoHPV\n";

        print SP $$pL[30]."\t".$npg1."\t".$npg2."\t".$npg3."\t".$npg4."\t".$npout."\t".$npnh."\n";

        close SP;
}

#xx Sub-routines END HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
###############################################################################################

###################################################################################################################
```

File name: **set2_p6mac_fasta.pl**

Source code:

```
#!/usr/bin/perl

### About the script ###################################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
#######################################################################################
### MAIN PROGRAM BODY ##################################################################
#######################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;

my $prefix_path= $sampleNo[2];
my $pyroNum = int($sampleNo[1]);

my $path = $prefix_path."Pyro".$pyroNum."_result/";
my $workingDir = $path.$sampleNo[0]."/";
my $fastaRC = $workingDir."sample".$sampleNo[0]."_RC.txt";

my $cut00 = "noCutoff";
my $cut28 = "28bpCutoff";

my @primerListEEpyro3 = ("E17","E02","E03","E04","E05","E06","E07","E08",
                         "E09", "E21", "E11", "E12", "E13", "E14", "E15", "E16",
                         "E18", "E19", "E20", "E22", "E23", "E24", "E25", "E26",
                         "E27", "E28", "E29", "E30", "E31", "E32");

print "\n\nAdding Fasta sequences after Program6...starts.........";

getFasta ($workingDir, $fastaRC, $cut00, \@primerListEEpyro3);
getFasta ($workingDir, $fastaRC, $cut28, \@primerListEEpyro3);

print "FINISHED\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
#######################################################################################

#######################################################################################
###    SUBROUTINES    ###############################################################
#######################################################################################
sub getFileData {
          my ($filename) = @_;
          use strict;
          use warnings;

          my @filedata = ();
          unless ( open(FILE_DATA, $filename) ) {
             print STDERR "\n\nThe program can't open the files \"$filename\"\n\n\n\n";
             print "Please re-check input file name and its location\n";
             print "The correct command should be :\n\n";
             print "\t\tperl PROGRAM(location+name) INPUTFastAFILE(location+name)\n\n";
             print "The command should be given under directory";
             print " c:/Perl/bin> in case of Dos Terminal\n\n\n";
             exit;
          }
          @filedata = <FILE_DATA>;
          close FILE_DATA;
          return @filedata;
}

sub getFasta {
          my ($workdir, $fasInFile, $prefix, $prim) = @_;
          use strict;
          use warnings;

          my @inseq = getFileData ($fasInFile);
          my $dir = $workdir."intersection/".$prefix."_intersect/";
          my $sufin  = "_n_sigHPV_list";
          my $sufout = "_n_sigHPV_fasta";

          for (my $p = 0; $p < (scalar @$prim); $p++) {
             my $listloc = $dir.$prefix.$$prim[$p].$sufin;
             my $outloc  = $dir.$prefix.$$prim[$p].$sufout;
             my @list = getFileData ($listloc);
             unless (open (FA, ">$outloc")) {print "\n\nCan not create ".$outloc."\n\n\n\n"; exit;}
             for (my $r = 0; $r < (scalar @list); $r++) {
                          my $seqnum = int($list[$r]);
                          my $nameline = 2*$seqnum;
                          my $seqline = (2*$seqnum)+1;
                          print FA $inseq[$nameline].$inseq[$seqline];
             }
             close FA;
          }
}

#xx Sub-routines END HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
#######################################################################################

#######################################################################################################################
```

File name: **set2_p7mac.pl**

Source code:

```
#!/usr/bin/perl

### About the script ###################################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
#######################################################################################
### MAIN PROGRAM BODY ##################################################################
#######################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;
my $prefix_path= $sampleNo[2];
my $pyroNum = int($sampleNo[1]);

my $path = $prefix_path."Pyro".$pyroNum."_result/";
my $samplePath = $path.$sampleNo[0]."/";
my $barcode = $sampleNo[0];

print "\n\nPROGRAM 7 (Find common 'breakpoint', if exists) STARTs.....................\n\n";

my $cut00 = "noCutoff";
my $cut28 = "28bpCutoff";
my $cfrep_suffix = "_allreport_fullpieceposition";

my @primerListEEpy3ro = ("E17","E02","E03","E04","E05","E06","E07","E08",
                         "E09", "E21", "E11", "E12", "E13", "E14", "E15", "E16",
                         "E18", "E19", "E20", "E22", "E23", "E24", "E25", "E26",
                         "E27", "E28", "E29", "E30", "E31", "E32");

getBreakPoint2 ($samplePath, $cfrep_suffix, $cut00, \@primerListEEpy3ro, $barcode);
getBreakPoint2 ($samplePath, $cfrep_suffix, $cut28, \@primerListEEpy3ro, $barcode);

my $oldDirName = $samplePath."intersection/";
my $newDirName = $samplePath."intersection-".$pyroNum."B".$sampleNo[0]."/";

system ("mv $oldDirName $newDirName");

print "FINISHED\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
#######################################################################################

#######################################################################################
###    SUBROUTINES    ###############################################################
#######################################################################################
sub getFileData {
          my ($filename) = @_;
          use strict;
          use warnings;
```

```
        my @filedata = ();
        unless ( open(FILE_DATA, $filename) ) {
           print STDERR "\n\nThe program can't open the files \"$filename\"\n\n\n\n";
           print "Please re-check input file name and its location\n";
           print "The correct command should be :\n\n";
           print "\t\tperl PROGRAM(location+name) INPUTFastAFILE(location+name)\n\n";
           print "The command should be given under directory";
           print " c:/Perl/bin> in case of Dos Terminal\n\n\n";
           exit;
        }
        @filedata = <FILE_DATA>;
        close FILE_DATA;
        return @filedata;
}

sub getBreakPoint2 {
        my ($path, $repsuffix, $prefix, $primList, $bc) = @_;
        use strict;
        use warnings;

        my $subPathCf = $path.$prefix."/";
        my $CFrepFile = $subPathCf.$prefix."_allreport_fullpieceposition";
        my @inCf = getFileData ($CFrepFile);

        my $subPathInter = $path."intersection/".$prefix."_intersect/";
        my $suffix = "_n_sigHPV_list";
        my $sufas = "_n_sigHPV_fasta";
        my $outall2 = $subPathInter."PrimerAll_breakpoint_".$prefix."_2";

        unless (open (ALL2, ">$outall2")) {print "\n\n\n\nCan not create ".$outall2."\n\n\n\n"; exit;}

        print ALL2 "In_".$prefix."\n\n";
        print ALL2 "barcode\tprimer\tSeqNo\tSeqLength\tLastPositionQuery\tLastPositionHPV";
        print ALL2 "\tMatchBP";
        print ALL2 "\tLastPositionQuery_2ndHit\tLastPositionHPV_2ndHit";
        print ALL2 "\tMatchBP_2ndHit\tOverlap\tOriginalName\tSequence\n";

        # for each primer
        for (my $p = 0; $p < (scalar @$primList); $p++) {

           my $inlistFile = $subPathInter.$prefix.$$primList[$p].$suffix;
           my $infasFile = $subPathInter.$prefix.$$primList[$p].$sufas;
           my $out = $subPathInter.$$primList[$p]."_breakpoint_".$prefix;

           unless (open (OUT, ">$out")) {print "\n\n\n\nCan not create ".$out."\n\n\n\n"; exit;}

           print OUT "For_PRIMER_GROUP_".$$primList[$p]."\n";
           print OUT "SeqNo\tSeqLength\tLastPositionQuery\tLastPositionHPV\t";
           print OUT "MatchBP\tOriginalName\tSequence\n";

           my @inlist = getFileData ($inlistFile);
           my @infas = getFileData ($infasFile);

           for (my $a = 0; $a < (scalar @inlist); $a++) {
                        my $seqnum = int($inlist[$a]);

                        my @temp_cfline = split (/#/, $inCf[$seqnum]);
                        my $tnum = $temp_cfline[0];                     # sequence number
                        my $tQL = $temp_cfline[1];                      # sequence length

                        my @temp_block = split (/%/, $temp_cfline[3]);
                        my $tBP = $temp_block[0];                       # used (&calculated) hpv-matched score-BP
                        my $tMS = $temp_block[3];                       # used first hpv-matched position on HPV strand
                        my $tposQ = $temp_block[4];         # calculated last hpv-matched position on Query
                        my ($tposS, $infasName, $infasSeq);
                        my ($tposQ2, $tposS2, $tBP2, $overlap);

                        my $usage = $temp_cfline[4];        # indicates which hit(s) is used AND how it's used

                        my @thit1 = split (/%/, $temp_cfline[5]);
                        my $series1 = $thit1[0];
                        my $score1 = $thit1[1];
                        my $firstM1 = $thit1[2];
                        my $strand1 = $thit1[3];
                        my $firstMS1 = $thit1[4];

                        my @thit2 = split (/%/, $temp_cfline[6]);
                        my $series2 = $thit2[0];
                        my $score2 = $thit2[1];
                        my $firstM2 = $thit2[2];
                        my $strand2 = $thit2[3];
                        my $firstMS2 = $thit2[4];

                        $infasName = $infas[2*$a];
                        $infasName =~ s/\s//g;
                        $infasSeq = $infas[(2*$a)+1];

                        if ($seqnum =~ $tnum) {

                                    my $tposQcorrect = $tposQ - 1 ;

                                    print OUT $tnum."\t".$tQL."\t".$tposQcorrect."\t";
                                    print ALL2 $bc."\t".$$primList[$p];
                                    print ALL2 "\t".$tnum."\t".$tQL."\t".$tposQcorrect."\t";

                                    if ($series1 =~ /^use_Alone$/) {
                                                $tposS2 = "none";
                                                $tposQ2 = "none";
                                                $tBP2 = "none";
                                                $overlap = "none";

                                                if ($strand1 =~ /^Plus/) {
                                                            $tposS = int($firstMS1) + int($score1) - 1 ;
                                                } elsif ($strand1 =~ /^Minus/) {
                                                            $tposS = int($firstMS1) - int($score1) + 1 ;
                                                } else {
                                                            print "\n\nError occurs at strand name\n\n"; exit;
                                                }
                                    } elsif ($series1 =~ /^subset$/) {
                                                $tposS2 = "none";
                                                $tposQ2 = "none";
                                                $tBP2 = "none";
                                                $overlap = "none";

                                                if ($strand2 =~ /^Plus/) {
                                                            $tposS = int($firstMS2) + int($score2) - 1 ;
                                                } elsif ($strand2 =~ /^Minus/) {
                                                            $tposS = int($firstMS2) - int($score2) + 1 ;
                                                } else {
                                                            print "\n\nError occurs at strand name\n\n"; exit;
                                                }

                                    } elsif (($series1 =~ /^use_1st$/) && ($series2 =~ /^use_2nd$/)){
                                                $tposQ2 = "combined";
                                                $tBP2 = $score2;
                                                my $difference = int($firstM1) + int($score1) - int($firstM2);
                                                $overlap = "overlap=".$difference;

                                                if ($strand1 =~ /^Plus/) {
                                                            $tposS = int($firstMS1) + int($score1) - 1 ;
                                                } elsif ($strand1 =~ /^Minus/) {
                                                            $tposS = int($firstMS1) - int($score1) + 1 ;
                                                } else {
                                                            print "\n\nError occurs at strand name\n\n"; exit;
                                                }

                                                if ($strand2 =~ /^Plus/) {
                                                            $tposS2 = int($firstMS2) + int($score2) - 1 ;
                                                } elsif ($strand2 =~ /^Minus/) {
                                                            $tposS2 = int($firstMS2) - int($score2) + 1 ;
                                                } else {
                                                            print "\n\nError occurs at strand name\n\n"; exit;
                                                }

                                    } elsif (($series1 =~ /^use_1st$/) && ($series2 =~ /^use_2nd_discont$/)){
                                                $tposQ2 = int($firstM2) + int($score2) - 1;
                                                $tBP2 = $score2;
                                                $overlap = "none";

                                                if ($strand1 =~ /^Plus/) {
                                                            $tposS = int($firstMS1) + int($score1) - 1 ;
                                                } elsif ($strand1 =~ /^Minus/) {
                                                            $tposS = int($firstMS1) - int($score1) + 1 ;
                                                } else {
                                                            print "\n\nError occurs at strand name\n\n"; exit;
                                                }
```

```
                        if ($strand2 =~ /^Plus/) {
                                $tposS2 = int($firstMS2) + int($score2) - 1 ;
                        } elsif ($strand2 =~ /^Minus/) {
                                $tposS2 = int($firstMS2) - int($score2) + 1 ;
                        } else {
                                print "\n\nError occurs at strand name\n\n"; exit;
                        }

                } elsif ($series1 =~ /^use_2nd$/) {
                        $tposQ2 = "combined";
                        $tBP2 = $score1;
                        my $difference = int($firstM2) + int($score2) - int($firstM1);
                        $overlap = "overlap=".$difference;

                        if ($strand1 =~ /^Plus/) {
                                $tposS2 = int($firstMS1) + int($score1) - 1 ;
                        } elsif ($strand1 =~ /^Minus/) {
                                $tposS2 = int($firstMS1) - int($score1) + 1 ;
                        } else {
                                print "\n\nError occurs at strand name\n\n"; exit;
                        }

                        if ($strand2 =~ /^Plus/) {
                                $tposS = int($firstMS2) + int($score2) - 1 ;
                        } elsif ($strand2 =~ /^Minus/) {
                                $tposS = int($firstMS2) - int($score2) + 1 ;
                        } else {
                                print "\n\nError occurs at strand name\n\n"; exit;
                        }

                } elsif ($series1 =~ /^use_2nd_discont$/) {
                        $tposQ2 = int($firstM1) + int($score1) - 1;
                        $tBP2 = $score1;
                        $overlap = "none";

                        if ($strand1 =~ /^Plus/) {
                                $tposS2 = int($firstMS1) + int($score1) - 1 ;
                        } elsif ($strand1 =~ /^Minus/) {
                                $tposS2 = int($firstMS1) - int($score1) + 1 ;
                        } else {
                                print "\n\nError occurs at strand name\n\n"; exit;
                        }

                        if ($strand2 =~ /^Plus/) {
                                $tposS = int($firstMS2) + int($score2) - 1 ;
                        } elsif ($strand2 =~ /^Minus/) {
                                $tposS = int($firstMS2) - int($score2) + 1 ;
                        } else {
                                print "\n\nError occurs at strand name\n\n"; exit;
                        }

                } else {
                        next;
                }

                print OUT $tposS."\t".$tBP."\t".$infasName."\t".$infasSeq;
                print ALL2 $tposS."\t".$tBP."\t".$tposQ2."\t".$tposS2."\t".$tBP2."\t";
                print ALL2 $overlap."\t".$infasName."\t".$infasSeq;

        } else {
                next;
        }
    }

    print OUT "\n";
    print ALL2 "\n";
    close OUT;
  }
  close ALL2;
}

#xx Sub-routines END HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
###############################################################################

#################################################################################################################
```

File name: **set3_p8mac.pl**

Source code:

```
#!/usr/bin/perl

### About the script ###############################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
###################################################################################
### MAIN PROGRAM BODY ##############################################################
###################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;
my $prefix_path= $sampleNo[2];
my $pyRunNo = int($sampleNo[1]);

my $path = $prefix_path."Pyro".$pyRunNo."_result/";
my $clus_loc = "/clustalw-2.0.10-macosx/clustalw2";         # path of clustalw2
my $samplePath = $path.$sampleNo[0]."/";
my $barcode = $sampleNo[0];
my $barcodeB = $pyRunNo."B".$barcode;
my $fname1 = ">".$pyRunNo."B".$barcode."_";
my $hpvEE_file = $prefix_path."PERL/ForPyro3/hpv16R_EEprimer_151nt"; # path of EEprimer_151nt
my $cut00 = "noCutoff";
my $cut28 = "28bpCutoff";

print "\n\nPROGRAM 8 (multiple alignment, automatic) STARTs.....................";

my @hpvEE = getFileData ($hpvEE_file);
my @primerListEEpy3ro = ("E17","E02","E03","E04","E05","E06","E07","E08",
                         "E09", "E21", "E11", "E12", "E13", "E14", "E15", "E16",
                         "E18", "E19", "E20", "E22", "E23", "E24", "E25", "E26",
                         "E27", "E28", "E29", "E30", "E31", "E32");
my @hpvPrimerEEposPy3ro = ("810","1275","1576","1951","2403","2723","3121","3555",
                         "1064", "1471", "1785", "2174", "2569", "2933", "3339", "3778",
                         "968", "1115", "1361", "1672", "1860", "2069", "2288", "2476",
                         "2628", "2857", "3037", "3199", "3455", "3696");

# (1) Prepare FASTA-formatted SELECTED sequences of EACH BARCODE, EACH PRIMER
getPrepSeq ($samplePath, $cut00, \@primerListEEpy3ro, $barcodeB, $fname1, \@hpvEE, \@hpvPrimerEEposPy3ro);
getPrepSeq ($samplePath, $cut28, \@primerListEEpy3ro, $barcodeB, $fname1, \@hpvEE, \@hpvPrimerEEposPy3ro);

# (2) ALIGN using CLUSTAL W, and get OUTPUTs (in FASTA format) into a new directory
getMultiAlignClustalW ($samplePath, $cut00, \@primerListEEpy3ro, $barcodeB, $clus_loc);
getMultiAlignClustalW ($samplePath, $cut28, \@primerListEEpy3ro, $barcodeB, $clus_loc);

# (3) MOVE the .dnd OUTPUTs to the same directory as the ALIGNED sequence outputs
getDndMoved ($samplePath, $cut00, \@primerListEEpy3ro, $barcodeB);
getDndMoved ($samplePath, $cut28, \@primerListEEpy3ro, $barcodeB);

print "FINISHED\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
###############################################################################

###################################################################################
###    SUBROUTINES    ###########################################################
###################################################################################
sub getFileData {
        my ($filename) = @_;
        use strict;
        use warnings;

        my @filedata = ();
        unless ( open(FILE_DATA, $filename) ) {
          print STDERR "\n\nThe program can't open the files \"$filename\"\n\n\n\n";
          print "Please re-check input file name and its location\n";
          print "The correct command should be :\n\n";
          print "\t\tperl PROGRAM(location+name) INPUTFastAFILE(location+name)\n\n";
          print "The command should be given under directory";
          print "  c:/Perl/bin> in case of Dos Terminal\n\n\n";
```

```
            exit;
        }
        @filedata = <FILE_DATA>;
        close FILE_DATA;
        return @filedata;
}

sub getPrepSeq {
        my ($path, $prefix, $primList, $bc, $fname1, $hpvEE, $hpvPosArray) = @_;
        use strict;
        use warnings;

        my $subPathInter = $path."intersection-".$bc."/".$prefix."_intersect/";

        # for each primer
        for (my $p = 0; $p < (scalar @$primList); $p++) {
           my $fname2 = $fname1.$$primList[$p]."_";
           my $outprefix = $bc."_".$$primList[$p];

           # for input data
           my $inrepExx = $subPathInter.$$primList[$p]."_breakpoint_".$prefix;
           my @inRep = getFileData ($inrepExx);

           # for OUTPUT , i.e. '/BXX_EXX_editFastA_CF'
           my $out = $subPathInter.$outprefix."_editFastA_unselected_".$prefix;
           unless (open (OUT, ">$out")) {print "\n\n\n\nCan not create ".$out."\n\n\n\n"; exit;}

           # for OUTPUT, with HPV16-EXX sequence for alignment analysis
           my $out3 = $subPathInter.$outprefix."_editFastA_HpvEE_SELECTED_".$prefix;
           unless (open (OUT3, ">$out3")) {print "\n\n\n\nCan not create ".$out3."\n\n\n\n"; exit;}

           # get the fasta of the HPV16_EEprimer portion
           my $hpvName = $$hpvEE[2*$p];
           my $hpvSeq = $$hpvEE[(2*$p)+1];
           my $hpvPos = $$hpvPosArray[$p];

           # go for EACH input FASTA-SEQ from inputRep
           if ((scalar @inRep) >= 3) {
                        my ($seqNum, $lastPosHpv, $matchBP, $origName, $seq);
                        my ($out_name0, $out_name, $out_seq);

                        print OUT3 $hpvName;
                        print OUT3 $hpvSeq;
                        for (my $a = 2; $a < (scalar @inRep); $a++) {
                                if ($inRep[$a] =~ /^[0-9](.*)$/) {
                                        my @tmp = split (/\t/, $inRep[$a]);
                                        $seqNum = $tmp[0];
                                        $lastPosHpv = $tmp[3];
                                        $matchBP = $tmp[4];
                                        $origName = $tmp[5];
                                        $seq = $tmp[6];

                                        #cut off 19 nt (RocheA)
                                        $out_seq = substr($seq, 19, $matchBP);
                                                # output edited seq, starting from 20th position, for the
                                                # length of $matchBP (matched HPV16)
                                        # use substr to take a portion of the original name
                                        if ($origName =~ /^(>)(.*)$/) {
                                                $out_name0 = substr ($origName, 1);
                                                $out_name = $fname2.$seqNum."_".$out_name0;
                                        } else {
                                                $out_name0 = substr ($origName, 0);
                                                $out_name = $fname2.$seqNum."_".$out_name0;
                                        }
                                        print OUT $out_name."\n";
                                        print OUT $out_seq."\n";

                                        # sort-out the 'mis-priming' sequences --> ROUGHLY
                                        if ( abs($lastPosHpv - ($hpvPos + $matchBP)) <= 10 ) {
                                                print OUT3 $out_name."\n";
                                                print OUT3 $out_seq."\n";
                                        }
                                } else { next; }
                        }
           } else { next; }
           close OUT; close OUT3;
        }
}

sub getMultiAlignClustalW {
        my ($path, $prefix, $primList, $bc, $clus_loc) = @_;
        use strict;
        use warnings;

        my $subPathInter = $path."intersection-".$bc."/".$prefix."_intersect/";
        my $outdir = $path."ALIGNMENTS-".$bc."/";
        system ("mkdir", $outdir);

        # for each primer
        for (my $p = 0; $p < (scalar @$primList); $p++) {
           my $inprefix = $bc."_".$$primList[$p];
           my $input = $subPathInter.$inprefix."_editFastA_HpvEE_SELECTED_".$prefix;
           my $output = $outdir.$inprefix."_editFA_HpvEE_selected_".$prefix."_ALIGNED";
           my $command = $clus_loc." -INFILE=".$input." -ALIGN -OUTFILE=".$output." -OUTPUT=FASTA";
           system ("$command");
        }
}

sub getDndMoved {

        my ($path, $prefix, $primList, $bc) = @_;
        use strict;
        use warnings;

        my $subPathInter = $path."intersection-".$bc."/".$prefix."_intersect/";
        my $outdir = $path."ALIGNMENTS-".$bc."/";

        # for each primer
        for (my $p = 0; $p < (scalar @$primList); $p++) {
           my $inprefix = $bc."_".$$primList[$p];
           my $input = $subPathInter.$inprefix."_editFastA_HpvEE_SELECTED_".$prefix;
           my $command = "mv ".$input.".dnd ".$outdir;
           system ("$command");
        }
}

#xx Sub-routines END HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
###############################################################################################

#########################################################################################################################
```

File name: **set3_p9mac.pl**

Source code:

```
#!/usr/bin/perl

### About the script ###################################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
########################################################################################
### MAIN PROGRAM BODY ##################################################################
########################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;
my $prefix_path= $sampleNo[2];
my $pyRunNo = int($sampleNo[1]);

my $path = $prefix_path."Pyro".$pyRunNo."_result/";
my $samplePath = $path.$sampleNo[0]."/";
my $barcode = $sampleNo[0];

my $barcodeB = $pyRunNo."B".$barcode;
my $cut00 = "noCutoff";
my $cut28 = "28bpCutoff";

print "\n\nPROGRAM 9 (sort ALIGNED-fastA sequences acc to size) STARTs.....................";

my @primerListEEpy3ro = ("E17","E02","E03","E04","E05","E06","E07","E08",
                         "E09", "E21", "E11", "E12", "E13", "E14", "E15", "E16",
                         "E18", "E19", "E20", "E22", "E23", "E24", "E25", "E26",
                         "E27", "E28", "E29", "E30", "E31", "E32");
```

```
getAlignedSeqSorted ($samplePath, $cut28, \@primerListEEpy3ro, $barcodeB);
getAlignedSeqSorted ($samplePath, $cut00, \@primerListEEpy3ro, $barcodeB);

print "FINISHED\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
##############################################################################################

##############################################################################################
###    SUBROUTINES    ########################################################################
##############################################################################################
sub getFileData {
         my ($filename) = @_;
         use strict;
         use warnings;

         my @filedata = ();
         unless ( open(FILE_DATA, $filename) ) {
            print STDERR "\n\nThe program can't open the files \"$filename\"\n\n\n\n";
            print "Please re-check input file name and its location\n";
            print "The correct command should be :\n\n";
            print "\t\tperl PROGRAM(location+name) INPUTFastAFILE(location+name)\n\n";
            print "The command should be given under directory";
            print " c:/Perl/bin> in case of Dos Terminal\n\n\n";
            exit;
         }
         @filedata = <FILE_DATA>;
         close FILE_DATA;
         return @filedata;
}

sub getAlignedSeqSorted {
         my ($path, $prefix, $primList, $bc) = @_;
         use strict;
         use warnings;

         my $indir = $path."ALIGNMENTS-".$bc."/";

         # for each primer
         for (my $p = 0; $p < (scalar @$primList); $p++) {

            my $inprefix = $bc."_".$$primList[$p];

            my $input = $indir.$inprefix."_editFA_HpvEE_selected_".$prefix."_ALIGNED";
            my $output = $indir.$inprefix."_editFA_HpvEE_selected_".$prefix."_ALIGNED_unsorted";
            my $output2 = $indir.$inprefix."_editFA_HpvEE_selected_".$prefix."_ALIGNED_SORTED";
            my $output3 = $indir.$inprefix."_editFA_HpvEE_selected_".$prefix."_ALIGNED_SORTED_FastA";

            my @infas = getFileData ($input);
            my @STACK2 = extractFastaStack (@infas);
            my @fname = ();
            my @fseq = ();
            extractFastaName2 (\@infas, \@STACK2, \@fname);
            extractFastaSeq2 (\@infas, \@STACK2, \@fseq);

            unless (open (OUT, ">$output")) {print "\n\n\n\nCan not create ".$output."\n\n\n\n"; exit;}
            unless (open (OUT2, ">$output2")) {print "\n\n\n\nCan not create ".$output2."\n\n\n\n"; exit;}
            unless (open (OUT3, ">$output3")) {print "\n\n\n\nCan not create ".$output3."\n\n\n\n"; exit;}

            print OUT "name\tsequence\tgap-counted\tindex\n";
            print OUT2 "name\tsequence\tunsorted_index\n";

            my %N_index;            # define hash: key=SeqName, value=index(as in order after Clustal alignment)
            my %N_seq;              # define hash: key=SeqName, value=Seq
            my %N_gapcount;         # define hash: key=SeqName, value=gapcount=Number of the '-'character, from 3'end
            my @sortedN_keys;       # define array of the SeqName after sorting by 'gapcount'

            # for each fastA seq
            for (my $m = 0; $m < (scalar @fname); $m++) {

                        my ($gapstring, $gapcount);

                        my $modiName = $fname[$m];                    # fasta name, still with a new line at the end
                        my $modiSeq = $fseq[$m];                      # fasta seq, still with new-lines

                        $modiName =~ s/\s//g;              # get rid of 'new line' character(s)
                        $modiSeq =~ s/\s//g;               # get rid of 'new line' character(s)

                        $N_index{$modiName} = "$m";  # input the 'value' (index) for 'key' (SeqName) for hash %N_index
                        $N_seq{$modiName} = "$modiSeq";  # input the 'value' (Seq) for 'key' (SeqName) for hash %N_seq

                        # control the 'ending' part of the Seq -- should be '-' character
                        # IF it's the longest seq and has NO '-' character at the end,
                        #  the 'gapcount' for this case was given as '0' by the program
                        if ( $modiSeq =~ /[A-Z]((-)*)$/i ) {

                                    $gapstring = $1;
                                    $gapcount = length ($gapstring);    # count the number of '-' character
                                    print OUT $modiName."\t".$modiSeq."\t".$gapcount."\t".$m."\n";

                                    # input the 'value'(gapcount) for 'key'(SeqName) for hash %N_gapcount
                                    $N_gapcount{$modiName} = "$gapcount";
                        }
            }

            my @keysN = keys (%N_gapcount); # unsorted keys of the hash %N_gapcount, put into an array @keysN

            # sorted keys of the hash %N_gapcount (sorted by its VALUES, descending) are put in array @sortedN_keys
            #   i.e. sort function  { using condition: hash-values (decending) } of the keys of the hash
            #        and then return the keys of the hash
            #   --> $a and $b represent in this case the keys
            @sortedN_keys = sort { $N_gapcount{$a} <=> $N_gapcount{$b} } keys(%N_gapcount) ;

            # for each value of the array @sortedN_keys which contains the sorted SeqNames
            #   the keys (SeqName) are called to get the corresponding values (Seq) from hash %N_seq
            #   and print them in FastA-formatted output ( OUT3 )
            for (my $n = 0; $n < (scalar @sortedN_keys); $n++) {
                        print OUT2 $sortedN_keys[$n]."\t".$N_seq{$sortedN_keys[$n]}."\t".
                                                          $N_index{$sortedN_keys[$n]}."\n";
                        print OUT3 $sortedN_keys[$n]."\n".$N_seq{$sortedN_keys[$n]}."\n";
            }

            close OUT; close OUT2; close OUT3;

         }
}

sub extractFastaStack {
         my (@fastaFileData) = @_;
         use strict;
         use warnings;

         my $maxcounter = scalar @fastaFileData;
         my @stacklist = ();
         for (my $m = 0; $m < $maxcounter ; $m++) {
            if ($fastaFileData[$m] =~ /^\s*$/) {
                        push @stacklist, 2;
            } elsif ($fastaFileData[$m] =~ /^\s*#/) {
                        push @stacklist, 2;
            } elsif ($fastaFileData[$m] =~ /^>/) {
                        push @stacklist, 0;
            } else {
                        push @stacklist, 1;
            }
         }
         return @stacklist;
}

sub extractFastaName2 {
         my ($data1, $STACK, $data1_name) = @_;
         use strict;
         use warnings;

         my $maxcounter = scalar @$data1;
         for (my $p = 0; $p < $maxcounter ; $p++) {
            if ($$STACK[$p] =~ "0") {
                        push @$data1_name, $$data1[$p];
            } else { next; }
         }
}

sub extractFastaSeq2 {
         my ($data1, $STACK, $data1_seq) = @_;
         use strict;
         use warnings;
```

```
         my $maxcounter = scalar @$data1;
         for (my $p = 0; $p < $maxcounter ; $p++) {
            if ($$STACK[$p] =~ "0") {
                        next;
            } elsif ( ($$STACK[$p] =~ "1") && ($$STACK[$p - 1] =~ "0")) {
                        push @$data1_seq, $$data1[$p];
            } elsif (($$STACK[$p] =~ "1") && ($$STACK[$p - 1] =~ "1")) {
                        my $lastEntry = (scalar @$data1_seq) - 1;
                        $$data1_seq[$lastEntry] .= $$data1[$p];
            } else { next; }
         }
}

#xx Sub-routines END HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
################################################################################

################################################################################################################
```

File name: **set4_p10mac.pl**
Source code:

```
#!/usr/bin/perl

### About the script ###########################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
################################################################################
### MAIN PROGRAM BODY ##########################################################
################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;

my $input = $sampleNo[0];
my $prefix_path = $sampleNo[2];
my $pyRunNo = int($sampleNo[1]);

print "\n\n=============================================================================";
print "\n\nRUNNING PROGRAM 10, FOR ALL BARCODES, ASP14 No: ".$pyRunNo."\n";

my $outpath =  $prefix_path."Pyro".$pyRunNo."_result_ALIGN/";

my @primerListEEpy3ro = ("E17","E02","E03","E04","E05","E06","E07","E08",
                         "E09", "E21", "E11", "E12", "E13", "E14", "E15", "E16",
                         "E18", "E19", "E20", "E22", "E23", "E24", "E25", "E26",
                         "E27", "E28", "E29", "E30", "E31", "E32");
system ("mkdir", $outpath);

getFolders (\@primerListEEpy3ro, $outpath);

print "FINISHED\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
################################################################################

### Sub-routine ################################################################################
sub getFolders {
         my ($primList, $outpathprefix) = @_;
         use strict;
         use warnings;

         # for each primer
         for (my $p = 0; $p < (scalar @$primList); $p++) {
            my $newfolder = $outpathprefix."/".$$primList[$p]."/";      # prefix for input 'BXX_EXX'
            system ("mkdir", $newfolder);
         }
}

#xx Sub-routines END HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
################################################################################

################################################################################################################
```

File name: **set4_p11mac.pl**
Source code:

```
#!/usr/bin/perl

### About the script ###########################################################
#
# Created by Sasithorn Chotewutmontri, Jan 2009.
#
################################################################################
### MAIN PROGRAM BODY ##########################################################
################################################################################

use strict;
use warnings;

my @sampleNo = @ARGV;

my $prefix_path = $sampleNo[2];
my $pyRunNo = int($sampleNo[1]);

my $path = $prefix_path."Pyro".$pyRunNo."_result/";
my $outpath =  $prefix_path."Pyro".$pyRunNo."_result_ALIGN/";

my $samplePath = $path.$sampleNo[0]."/";
my $barcode = $sampleNo[0];
my $barcodeB = $pyRunNo."B".$barcode;

# INPUT SEQUENCES ARE REVERSE-COMPLEMENTARY SEQUENCES FROM PROGRAM2, in the file:
my $inOrigRevComp = $samplePath."sample".$sampleNo[0]."_RC.txt";

my $cut00 = "noCutoff";
my $cut28 = "28bpCutoff";

print "\n\nPROGRAM 11 (generate FastA of FULL seq after MultiClustered-aligned-sorted";

print "sequences) STARTs.....................\n\n";

my @primerListEEpy3ro = ("E17","E02","E03","E04","E05","E06","E07","E08",
                         "E09", "E21", "E11", "E12", "E13", "E14", "E15", "E16",
                         "E18", "E19", "E20", "E22", "E23", "E24", "E25", "E26",
                         "E27", "E28", "E29", "E30", "E31", "E32");

getFullSortedSeq ($samplePath, $cut28, \@primerListEEpy3ro, $barcodeB, $inOrigRevComp, $outpath);
getFullSortedSeq ($samplePath, $cut00, \@primerListEEpy3ro, $barcodeB, $inOrigRevComp, $outpath);

print "...for barcode...".$barcode."...FINISHED\n\n";
exit;

#xx Main Program ENDs HERE xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
################################################################################

################################################################################
###    SUBROUTINES    ##########################################################
################################################################################

sub getFileData {
         my ($filename) = @_;
         use strict;
         use warnings;

         my @filedata = ();
         unless ( open(FILE_DATA, $filename) ) {
            print STDERR "\n\nThe program can't open the files \"$filename\"\n\n\n\n";
            print "Please re-check input file name and its location\n";
            print "The correct command should be :\n\n";
            print "\t\tperl PROGRAM(location+name) INPUTFastAFILE(location+name)\n\n";
            print "The command should be given under directory";
            print " c:/Perl/bin> in case of Dos Terminal\n\n\n";
            exit;
         }
         @filedata = <FILE_DATA>;
         close FILE_DATA;
         return @filedata;
}
```

```
sub getFullSortedSeq {
         my ($path, $prefix, $primList, $bc, $inRCFile, $outpathprefix) = @_;
         use strict;
         use warnings;

         my $indir = $path."ALIGNMENTS-".$bc."/";
         my @rc = getFileData ($inRCFile);

         # for each primer
         for (my $p = 0; $p < (scalar @$primList); $p++) {

            my $inprefix = $bc."_".$$primList[$p];

            my $inSortFile =
$indir.$inprefix."_editFA_HpvEE_selected_".$prefix."_ALIGNED_SORTED_FastA";

            my @insort = getFileData ($inSortFile);

            my $outFileOne =
$indir.$inprefix."_editFA_HpvEE_selected_".$prefix."_ALIGNED_SORTED_FastA_FUL
LSEQ";

            unless (open (OUT, ">$outFileOne")) {print "\n\n\n\nCan not
create ".$outFileOne."\n\n\n\n"; exit;}

            # for each line of the Aligned-Sorted-FastA file (of each primer)
            for (my $s = 0; $s < (scalar @insort); $s++) {

                        # go for each line, look at the 'name' line
                        if ($insort[$s] =~ /^>[0-9]B[0-9][0-9]_(.*)$/) {

                                    my @insortline = split (/_/,
$insort[$s]);
                                    my $barcodeName = $insortline[0];
                                    my $primerName = $insortline[1];
                                    my $seqNum = $insortline[2];

                                    my @origNameTemp = split (/ /,
$rc[($seqNum*2)]);
                                    my $extraName = substr ($origNameTemp[0],
1);
                                    print OUT
$barcodeName."_".$primerName."_".$seqNum."_".$extraName."\n";
                                    print OUT $rc[($seqNum*2)+1];

                        } elsif ($insort[$s] =~ /^>HPV16R_(.*)$/) {

                                    print OUT $insort[$s];
                                    print OUT $insort[$s+1];

                        } else { next; }

            }

            close OUT;

            my $outpath = $outpathprefix.$$primList[$p]."/";
            my $outFileNameTwo =
$outpath.$inprefix."_".$prefix."_ALIGNED_SORTED_FastA_FULLSEQ";
            my $command1 = "cp ".$outFileOne." ".$outFileNameTwo;

            system ("$command1");

         }
}

#xx Sub-routines END HERE
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
################################################################################
###############

################################################################################
########################################
```