Chromosomal clustering of tissue restricted antigens

Doktorarbeit

Technical Appendix

Part A: Programming Code

August 9, 2017

Contents

1 Technical Appendix - Part A: Programming Code

1.1 R Scripts, Perl Scripts, Shell Scripts

For the analysis of this work we used the open source statistical programming language R, Perl and ordinary Shell scripts.

1.1.1 Reading in and Pre Processing of Microarray data

This script includes data input, pre processing, quality control, normalization, calculation of mean.vsnrma over the double measurements of each tissue in Microarrays.

```
#analysis_of_mouse_gngnf1_chips.R
#analysis_of_mouse_4302_chips.R
#analysis_human_hgu133a_chips.R
#analysis_human_roth.R

#0. Microarray data used
#------------------------
#Affymetrix microarrays, five datasets, CEL files were retrieved from the GEO microarray
#database.
#-----------------------
#mouse novartis data, gngnf1 chip, Su et al. 2002, 2004, GEO: GSE1133
#human novartis data, hgu133a chips, Su et al. 2002, 2004, GEO: GSE1133
#rat novartis data, Su et al. 2002, 2004
#mouse 4301 chips, GSE10246
#human roth data, hgu133 plus 2.0 chips, GEO: GSE3526, 65 tissues

#1. Data import
#--------------
#In order to analyse microarray Affymetrix chips, we used the ReadAffy() function for
#the import of CEL files on the basis of Ensembl Transcripts, rather than AffyIDs, this
#can be done with packages provided by brainarray, which are regulary updated.

#The R-packages for each chip for the annotation of CDF environment and probe set data
#have to be dowonloaded from brainarray, CustomCDF Files, installed and read in while
#using the ReadAffy() function from the affy() Package. For this the cdfName Variable
#has to be changed.

#Go to http://brainarray.mbni.med.umich.edu: -> brainarray -> Download CDF Files ->
#-> Latest Version (Version 18: Jan 23, 2013) -> ENST (ensemble transcripts) ->
#-> your chip name, and download the source code both for cdf environment and probe set
#data.

#or use the following links with wget directly on the terminal in the window you want to
#install the packages in.

#mouse gngnf1
#------------
#cdf environment
http://brainarray.mbni.med.umich.edu/Brainarray/Database/CustomCDF/18.0.0/enst.download/
gngnf1musammenstcdf_18.0.0.tar.gz

#probe set data
```

```
http://brainarray.mbni.med.umich.edu/Brainarray/Database/CustomCDF/18.0.0/enst.download/
gngnf1musammenstprobe_18.0.0.tar.gz

#mouse 4302
#----------
#cdf environment
http://brainarray.mbni.med.umich.edu/Brainarray/Database/CustomCDF/18.0.0/enst.download/
mouse4302mmenstcdf_18.0.0.tar.gz

#probe set data
http://brainarray.mbni.med.umich.edu/Brainarray/Database/CustomCDF/18.0.0/enst.download/
mouse4302mmenstprobe_18.0.0.tar.gz

#human novartis
#--------------
#cdf environment
http://brainarray.mbni.med.umich.edu/Brainarray/Database/CustomCDF/18.0.0/enst.download/
hgu133ahsenstcdf_18.0.0.tar.gz

#probe set data
http://brainarray.mbni.med.umich.edu/Brainarray/Database/CustomCDF/18.0.0/enst.download/
hgu133ahsenstprobe_18.0.0.tar.gz

#human roth
#----------
#cdf environment
http://brainarray.mbni.med.umich.edu/Brainarray/Database/CustomCDF/18.0.0/enst.download/
hgu133plus2hsenstcdf_18.0.0.tar.gz

#probe set data
http://brainarray.mbni.med.umich.edu/Brainarray/Database/CustomCDF/18.0.0/enst.download/
hgu133plus2hsenstprobe_18.0.0.tar.gz

#Extract and Install Packages.R
#------------------------------
#gunzip
gunzip gngnf1musammenstprobe_18.0.0.tar.gz
gunzip gngnf1musammenstcdf_18.0.0.tar.gz

#tar
tar xvf gngnf1musammenstprobe_18.0.0.tar
tar xvf gngnf1musammenstprobe_18.0.0.tar

#install with full path in a different folder than you extracted the files
#use the same R version you use later on on the cluster node for compatability reason

#R CMD INSTALL -l ~/R-libs/ ./hgu133ahsenstprobe
#R CMD INSTALL -l ~/R-libs/ ./hgu133ahsenstcdf

#These packages depend upon AnnotationDbi_1 from bioconductor, so you might also have to
#update and install this version upon your current system, AnnotationDbi_1.24.0

#1.1 librarys
#------------
library(affy)
library(vsn)

#ueberlagern sich gegenseitig, nur beim Einlesen nutzen?
#gngnf1 chip
library(gngnf1musammenstcdf, lib.loc="/home/dinkelac/R-libs/")
library(gngnf1musammenstprobe, lib.loc="/home/dinkelac/R-libs/")
```

```
#mouse 4302 chips
library(mouse4302mmenstcdf, lib.loc="/home/dinkelac/R-libs/")
library(mouse4302mmenstprobe, lib.loc="/home/dinkelac/R-libs/")

#human novartis
library(hgu133ahsenstprobe, lib.loc="/home/dinkelac/R-libs/")
library(hgu133ahsenstcdf, lib.loc="/home/dinkelac/R-libs/")

#human roth
library(hgu133plus2hsenstprobe, lib.loc="/home/dinkelac/R-libs/")
library(hgu133plus2hsenstcdf, lib.loc="/home/dinkelac/R-libs/")

#sessionInfo()
AnnotationDbi_1.24.0s

#1.2 read in CEL files
#----------------------
#mouse gngnf1
setwd("/home/dinkelac/data/mouse/rawdata/gngnf1")

data.mouse=ReadAffy()
data.mouse@cdfName<-"GN_GNF_1MusA_MM_ENST"

setwd("/home/dinkelac/data/mouse/sessions/rda")
save.image(file="rawdata.gngnf1.rda")

#mouse 4301
setwd("/home/dinkelac/data/mouse/rawdata/mouse4302")

data.mouse4301=ReadAffy()
data.mouse4301@cdfName<-"MOUSE_4302_MM_ENST"

pdata4301=read.table(file="pdata_mouse4302.txt")
tissue.names4301=as.character(pdata4301[,2])
names(tissue.names4301)=pdata4301[,1]

setwd("/home/dinkelac/data/mouse/sessions/rda")
save.image(file="rawdata.4302.rda")

#human novartis
setwd("/home/dinkelac/data/mouse/rawdata/HGU133A/")

pheno.data.human.novartis=read.table(file="phenodata.txt")
file.names.human.novartis=as.character(pheno.data.human.novartis[,1])
tissues.human.novartis=as.character(pheno.data.human.novartis[,2])
tissue.names.human.novartis=substr(tissues.human.novartis,1,
nchar(tissues.human.novartis)-9)

setwd("/home/dinkelac/data/human/rawdata/HGU133A/celfiles/")

data.human=ReadAffy(filenames=file.names)
data.human@cdfName<-"HGU133A_Hs_ENST"

setwd("/home/dinkelac/data/mouse/sessions/rda/")
save.image(file="human.rawdata.rda")

#human roth
setwd("/home/dinkelac/data/mouse/rawdata/human_roth/")

pheno.data.human.roth=read.table(file="phenodata.txt")
file.names.human.roth=paste(as.character(pheno.data.human.roth[,1]),".CEL",sep="")
tissues.human.roth=as.character(pheno.data.human.roth[,2])
```

```
setwd("/home/dinkelac/data/mouse/rawdata/human_roth/celfiles/")

data.human.roth=ReadAffy(filenames=file.names.human.roth)
data.human.roth@cdfName<-"HGU133Plus2_Hs_ENST"

setwd("/home/dinkelac/data/mouse/sessions/rda/")
save.image(file="human.roth.rawdata.rda")

#2. Quality control
#------------------
#2.1 Single chip control
#-----------------------

#mouse gngnf1
#------------
#show chip images of the raw intensities
image(data.mouse, col=rainbow(100, start=0, end=0.75)[100:1])

#weisser Strich ist ein artefakt der Plot Groesse
#Atrachea.CEL fingerprint?

image(data.mouse[5], col=rainbow(100, start=0, end=0.75)[100:1])
#5- "MGJZ030207041Auterus.CEL"

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="quality_control_mouse_gngnf1_Auterus.eps")

image(data.mouse[109], col=rainbow(100, start=0, end=0.75)[100:1])
#109 - MGMH030311039Atrachea.CEL

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="quality_control_mouse_gngnf1_Atrachea.eps")

#mouse 4302
#----------
#show chip images of the raw intensities
image(data.mouse4301, col=rainbow(100, start=0, end=0.75)[100:1])

setwd("/home/dinkelac/data/mouse/plots")

#schlechte chips
image(data.mouse4301[6], col=rainbow(100, start=0, end=0.75)[100:1])
dev.copy2eps(file="quality_control_4302_GSM258614.eps")
#adipose_white_B

image(data.mouse4301[9], col=rainbow(100, start=0, end=0.75)[100:1])
dev.copy2eps(file="quality_control_4302_GSM258617.eps")
#amygdala_A

image(data.mouse4301[72], col=rainbow(100, start=0, end=0.75)[100:1])
dev.copy2eps(file="quality_control_4302_GSM258680.eps")
#iris_B

image(data.mouse4301[158], col=rainbow(100, start=0, end=0.75)[100:1])
dev.copy2eps(file="quality_control_4302_GSM258766.eps")
#spinal_cord_B

#human roth
#----------
#show chip images of the raw intensities
image(data.human.roth, col=rainbow(100, start=0, end=0.75)[100:1])
```

```
#schlechte chips
image(data.human.roth[22], col=rainbow(100, start=0, end=0.75)[100:1])
# dev.copy2eps(file="quality_control_human_roth_GSM80582.eps")
#scratch, bronchus_4

image(data.human.roth[84], col=rainbow(100, start=0, end=0.75)[100:1])
# dev.copy2eps(file="quality_control_human_roth_GSM80644.eps")
#dark fingerprint at the side, corpus callosum 7

image(data.human.roth[127], col=rainbow(100, start=0, end=0.75)[100:1])
# dev.copy2eps(file="quality_control_human_roth_GSM80687.eps")
#scratch, kidney_cortex_2

image(data.human.roth[143], col=rainbow(100, start=0, end=0.75)[100:1])
# dev.copy2eps(file="quality_control_human_roth_GSM80703.eps")
#dunkel, man sieht nichts, Bananenform beim scatterplot, midbrain_5

image(data.human.roth[145], col=rainbow(100, start=0, end=0.75)[100:1])
# dev.copy2eps(file="quality_control_human_roth_GSM80705.eps")
#dark slopes, midbrain_9

image(data.human.roth[146], col=rainbow(100, start=0, end=0.75)[100:1])
# dev.copy2eps(file="quality_control_human_roth_GSM80706.eps")
#dark edges, midbrain_10

image(data.human.roth[197], col=rainbow(100, start=0, end=0.75)[100:1])
# dev.copy2eps(file="quality_control_human_roth_GSM80757.eps")
#-> sehr hell, ovary_8

image(data.human.roth[216], col=rainbow(100, start=0, end=0.75)[100:1])
# dev.copy2eps(file="quality_control_human_roth_GSM80776.eps")
#-> schliere

image(data.human.roth[217], col=rainbow(100, start=0, end=0.75)[100:1])
# dev.copy2eps(file="quality_control_human_roth_GSM80777.eps")
#-> fingerprint

image(data.human.roth[313], col=rainbow(100, start=0, end=0.75)[100:1])
# dev.copy2eps(file="quality_control_human_roth_GSM80873.eps")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="quality_control_human_roth_all_chips.eps")

#-> fingerprint

#human novartis
#--------------
setwd("/home/dinkelac/data/mouse/plots")
image(data.human, col=rainbow(100, start=0, end=0.75)[100:1])

#schlechte chips
image(data.human[26], col=rainbow(100, start=0, end=0.75)[100:1])
dev.copy2eps(file="quality_control_human_GSM19006.eps")
#->kleiner tip

image(data.human[74], col=rainbow(100, start=0, end=0.75)[100:1])
dev.copy2eps(file="quality_control_human_GSM18996.eps")
#->schliere

image(data.human[83], col=rainbow(100, start=0, end=0.75)[100:1])
dev.copy2eps(file="quality_control_human_GSM19015.eps")
```

```
#->helle Ecken

image(data.human[84], col=rainbow(100, start=0, end=0.75)[100:1])
dev.copy2eps(file="quality_control_human_GSM19016.eps")
#-> sehr helles chip, and fingerprint

image(data.human[92], col=rainbow(100, start=0, end=0.75)[100:1])
dev.copy2eps(file="quality_control_human_GSM18986.eps")
#->helle Ecke

image(data.human[95], col=rainbow(100, start=0, end=0.75)[100:1])
dev.copy2eps(file="quality_control_human_GSM18983.eps")

image(data.human[126], col=rainbow(100, start=0, end=0.75)[100:1])
dev.copy2eps(file="quality_control_human_GSM18970.eps")

#2.2 pheno data
#--------------
#adjust filenames and read in pheno data

#mouse gngnf1
#------------
file.names.mouse=rownames(pData(phenoData(data.mouse)))
colnames(exprs(data.mouse))=file.names.mouse

#get tissue names
#extract tissue names
tissue.names.mouse=substr(file.names.mouse,3,nchar(file.names.mouse))
letters=substr(tissue.names.mouse,1,1)
tissues=substr(tissue.names.mouse,2,nchar(tissue.names.mouse))

x=paste(letters,tissues,sep="_",colapse=NULL)
new.file.names=substr(x,1,nchar(x)-1)

#change tissue names
rownames(pData(phenoData(data.mouse)))=new.file.names
rownames(pData(protocolData(data.mouse)))=new.file.names
colnames(exprs(data.mouse))=new.file.names

#mouse 4302
#----------
setwd("/home/dinkelac/data/mouse/rawdata/mouse4302")
pheno.data.mouse4302=read.table(file="pdata_mouse4302.txt")
tissues.mouse4302=as.character(pheno.data.mouse4302[,2])
filenames.mouse4302=as.character(pheno.data.mouse4302[,1])
filenames1.mouse4302=paste(filenames.mouse4302,".CEL",sep="")
names(tissues.mouse4302)=filenames1.mouse4302

x=colnames(exprs(data.mouse4301))
new.tissues=as.character(tissues.mouse4302[x])

change filenames
colnames(exprs(data.mouse4301))=new.tissues
colnames(exprs(data.mouse4301)

tissue.names.mouse4302=as.character(pheno.data.mouse4302[,2])
tissues.mouse4302=substr(tissue.names.mouse4302,1,nchar(tissue.names.mouse4302)-2)

#human roth
#----------
colnames(exprs(data.human.roth))=tissues.human.roth
```

```
a=tissues.human.roth
b=sub("10","0",a)
tissue.names.human.roth=substr(b,1,nchar(b)-2)
d=unique(c)

roth.tissues.all=(c)
roth.tissues=(d)

setwd("/home/dinkelac/data/mouse/tables/")
write.table(roth.tissues,"tissue_types.roth.txt",row.names=F)
write.table(roth.tissues.all,"tissues_all.roth.txt",row.names=F)

#human novartis
#--------------
colnames(exprs(data.human))=tissue.names.human

how many different tissues?

b=substr(tissue.names,1,nchar(tissue.names)-1)
a=unique(b)

setwd("/home/dinkelac/data/mouse/sessions/rda/")
save.image(file="human.mouse.rawdata.rda")

#3 Normalization
#---------------
#For Normalization vsnrma normalization was taken, Huber et al. 2002
#citation("vsn")

library(vsn)

mouse.vsnrma<-vsnrma(data.mouse)
mouse4301.vsnrma<-vsnrma(data.mouse4301)
human.vsnrma<-vsnrma(data.human)
human.roth.vsnrma<-vsnrma(data.human.roth)

setwd("/home/dinkelac/data/mouse/sessions/rda")
save.image(file="normalized.data.rda")

#4 Quality Plots
#---------------
#4.1 meandSdPlot
#---------------
#mouse gngnf1
#------------
meanSdPlot(mouse.vsnrma)
title(main="Gngnf1- mouse meanSdPlot\n after vsnrma normalization")
title(sub="row standard deviations versus row means")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="meanSdPlot.mouse.vsnrma.normalized.eps")

#mouse 4301
#----------
meanSdPlot(mouse4301.vsnrma)
title(main="Mouse 4302- meanSdPlot\n after vsnrma normalization")
title(sub="row standard deviations versus row means")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="meanSdPlot.mouse4302.vsnrma.normalized.eps")

#human novartis
```

```
#--------------
meanSdPlot(human.vsnrma)
title(main="Human hgu133A - meanSdPlot\n after vsnrma normalization")
title(sub="row standard deviations versus row means")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="meanSdPlot.human.vsnrma.normalized.eps")

#human roth
#----------

meanSdPlot(human.roth.vsnrma)
title(main="Human hgu133A Plus2 - meanSdPlot\n after vsnrma normalization")
title(sub="row standard deviations versus row means")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="meanSdPlot.human.roth.vsnrma.normalized.eps")

#4.2 Boxplot
#-----------
#rawdata
#-------

#mouse gngnf1

x11(w=10,h=7)
#90 Grad beschriften
par(las=2)
#Rand zum Beschriften unten, li, oben, re
mmi=c(1.7,0.7,1.0477939,0.5366749)
par(mai=mmi)
boxplot(data.mouse, col=rainbow(150),cex.axis=0.5,
main="Gngnf1- Gene expression in 61 different tissues of the mouse\n
before normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.mouse.rawdata.eps")

#mouse 4301
x11(w=12,h=7)
#90 Grad beschriften
par(las=2)
#Rand zum Beschriften unten, li, oben, re
mmi=c(1.9,0.7,1.0477939,0.5366749)
par(mai=mmi)

boxplot(data.mouse4301, col=rainbow(150),cex.axis=0.5,
main="Mouse 4302- Gene expression in 91 different tissues of the mouse\n
before normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.mouse4302.rawdata.eps")

#human novartis
x11(w=12,h=7)
#90 Grad beschriften
par(las=2)
#Rand zum Beschriften unten, li, oben, re
mmi=c(1.9,0.7,1.0477939,0.5366749)
par(mai=mmi)

boxplot(data.human, col=rainbow(150),cex.axis=0.5,
```

```
main="Human hgu133A- Gene expression in 71 different human tissues\n
before normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.human.rawdata.eps")

#human roth
x11(w=12,h=7)
#90 Grad beschriften
par(las=2)
#Rand zum Beschriften unten, li, oben, re
mmi=c(1.9,0.7,1.0477939,0.5366749)
par(mai=mmi)

boxplot(data.human.roth, col=rainbow(150),cex.axis=0.5,
main="Human hgu133_Plus2- Gene expression in 65 different human tissues\n
before normalization (Roth)")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.human.roth.rawdata.eps")

#normalized data
#---------------
#mouse gngnf1

#vsnrma
#-------
x11(w=10,h=7)
par(las=2)
mmi=c(1.7,0.7,1.0477939,0.5366749)
par(mai=mmi)

boxplot(exprs(mouse.vsnrma), col=rainbow(150),cex.axis=0.5,
main="Gngnf1- Gene expression in 61 different tissues of the mouse\n
after vsnrma normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.gngnf1.vsnrma.normalized.eps")

#mouse 4302
x11(w=12,h=7)
par(las=2)
mmi=c(1.9,0.7,1.0477939,0.5366749)
par(mai=mmi)

boxplot(exprs(mouse4301.vsnrma), col=rainbow(150),cex.axis=0.5,
main="Mouse 4302- Gene expression in 91 different tissues of the mouse\n
after vsnrma normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.4302.vsnrma.normalized.eps")

#human novartis
x11(w=12,h=7)
par(las=2)
mmi=c(1.9,0.7,1.0477939,0.5366749)
par(mai=mmi)

boxplot(exprs(human.vsnrma), col=rainbow(150),cex.axis=0.5,
main="Human hgu133A - Gene expression in 71 different human tissues\n
after vsnrma normalization")
```

```
setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.human.vsnrma.normalized.eps")

#human roth
x11(w=12,h=7)
par(las=2)
mmi=c(1.9,0.7,1.0477939,0.5366749)
par(mai=mmi)

boxplot(exprs(human.roth.vsnrma), col=rainbow(150),cex.axis=0.5,
main="Human hgu133_Plus2 - Gene expression in 65 different human tissues\n
after vsnrma normalization (Roth)")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.human.roth.vsnrma.normalized.eps")

#4.3 Histogram
#-------------
#rawdata
#-------
#mouse gngnf1
x11(w=10,h=7)
hist(data.mouse, col=rainbow(150),
main="Gene expression in 61 different tissues of the mouse (gngnf1)\n
before normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.mouse.rawdata.eps")

#mouse 4302
x11(w=10,h=7)

hist(data.mouse4301, col=rainbow(150),
main="Gene expression in 91 different tissues of the mouse (4302)\n
before normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.mouse4302.rawdata.eps")

#human novartis
x11(w=10,h=7)

hist(data.human, col=rainbow(150),
main="Gene expression in 71 different human tissues\n before normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.human.rawdata.eps")

#human roth
x11(w=10,h=7)

hist(data.human.roth[,1:353],col=rainbow(353),
main="Gene expression in 65 different human tissues (Roth data)\n
before normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.human.roth.rawdata.eps")

#normalized
#----------
#vsnrma
```

```
#mouse gngnf1

eset = exprs(mouse.vsnrma)

plot(density(eset[,1]), type="n", ylim=c(0,1),
main="Gene expression in 61 different tissues of the mouse (gngnf1)\n
after vsnrma normalization")
for (i in 1:ncol(eset)){
  lines(density(eset[,i]), col=rainbow(150)[i])
}

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.gngnf1.vsnrma.normalized.eps")

#mouse 4302
eset = exprs(mouse4301.vsnrma)

plot(density(eset[,1]), type="n", ylim=c(0,1),
main="Gene expression in 91 different tissues of the mouse (4302)\n
after vsnrma normalization")
for (i in 1:ncol(eset)){
  lines(density(eset[,i]), col=rainbow(180)[i])
}

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.4302.vsnrma.normalized.eps")
#human novartis
eset = exprs(human.vsnrma)

plot(density(eset[,1]), type="n", ylim=c(0,1),
main="Gene expression in 79 different human tissues\n after vsnrma normalization")
for (i in 1:ncol(eset)){
  lines(density(eset[,i]), col=rainbow(180)[i])
}

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.human.vsnrma.normalized.eps")

#human roth
eset = exprs(human.roth.vsnrma)

plot(density(eset[,1]), type="n", ylim=c(0,2),
main="Gene expression in 65 different human tissues (Roth data)\n
after vsnrma normalization")
for (i in 1:ncol(eset)){
  lines(density(eset[,i]), col=rainbow(353)[i])
}

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.human.roth.vsnrma.normalized.eps")

#4.4 RNA degeneration plot
#-------------------------
#mouse gngnf1
rnadeg.raw = AffyRNAdeg(data.mouse)

plotAffyRNAdeg(rnadeg.raw, col=rainbow(150))
title(sub="mouse gngnf1 rawdata")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="rnadeg.gngnf1.rawdata.eps")
```

```
plotAffyRNAdeg(rnadeg.raw, col=rainbow(150), transform="shift.only")
title(sub="mouse gngnf1 rawdata")

dev.copy2eps(file="rnadeg1.gngnf1.rawdata.eps")

#mouse 4302
rnadeg.raw = AffyRNAdeg(data.mouse4301)

plotAffyRNAdeg(rnadeg.raw, col=rainbow(180))
title(sub="mouse 4302 rawdata")

dev.copy2eps(file="rnadeg.4302.rawdata.eps")

plotAffyRNAdeg(rnadeg.raw, col=rainbow(180), transform="shift.only")
title(sub="mouse 4302 rawdata")

dev.copy2eps(file="rnadeg1.4302.rawdata.eps")

#human novartis
rnadeg.raw = AffyRNAdeg(data.human)

plotAffyRNAdeg(rnadeg.raw, col=rainbow(180))
title(sub="human rawdata")

dev.copy2eps(file="rnadeg.human.rawdata.eps")

plotAffyRNAdeg(rnadeg.raw, col=rainbow(180), transform="shift.only")
title(sub="human rawdata")

dev.copy2eps(file="rnadeg1.human.rawdata.eps")

#human roth
rnadeg.raw = AffyRNAdeg(data.human.roth)

plotAffyRNAdeg(rnadeg.raw, col=rainbow(353))

title(sub="human rawdata (Roth data)")

dev.copy2eps(file="rnadeg.human.roth.rawdata.eps")

plotAffyRNAdeg(rnadeg.raw, col=rainbow(353), transform="shift.only")
title(sub="human rawdata (Roth data)")

dev.copy2eps(file="rnadeg1.human.roth.rawdata.eps")

#5. Calculate the mean over double messurements
#-----------------------------------------------
#mouse gngnf1
#------------
tissue.names=colnames(exprs(mouse.vsnrma))

#all tissues exist twice, only cortex and cerebralcortex once,
#is this the same?

which(tissue.names=="B_cerebralcortex")
which(tissue.names=="A_cortex")

#22,111

plot(exprs(mouse.vsnrma)[,c(22,111)], pch=".")
title(main="Gene expression values of the mouse gngnf1 dataset\n
cortex versus cerbralcortex")
```

```
setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="gngnf1.cortex.cerebralcortex.eps")

colnames(exprs(mouse.vsnrma))[22]="B_cortex"

#mean.vsnrma
tissues=substr(tissue.names,3,nchar(tissue.names))
tissues.unique=unique(tissues)

mean.mouse.vsnrma = matrix(NA, nr=nrow(exprs(mouse.vsnrma)), nc=61)
rownames(mean.mouse.vsnrma) = featureNames(mouse.vsnrma)
colnames(mean.mouse.vsnrma) = tissues.unique

for (t in tissues.unique) {
  index.tissue = which(tissues == t) # alternativ: which(tissues.all %in% t)
  if (length(index.tissue) > 1) {
    mean.mouse.vsnrma[, t] = apply(exprs(mouse.vsnrma)[,index.tissue], 1,
    mean, na.rm=T)
  }
}

#boxplot of mean.vsnrma
x11(w=10,h=7)
par(las=2)
mmi=c(1.7,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
boxplot(data.frame(mean.mouse.vsnrma), col=rainbow(100), cex.axis=0.5)
title(main="Gngnf1 - distribution of averaged expressions\n
in 61 mouse tissues after vsnrma normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.gngnf1.mean.vsnrma.eps")

#mouse 4301
colnames(exprs(mouse4301.vsnrma))=as.character(pdata4301[,2])

tissues=colnames(exprs(mouse4301.vsnrma))
tissue.names=substr(tissues,1,nchar(tissues)-2)

tissues.unique=unique(tissue.names)

mean4301.vsnrma = matrix(NA, nr=nrow(exprs(mouse4301.vsnrma)), nc=91)
rownames(mean4301.vsnrma) = featureNames(mouse4301.vsnrma)
colnames(mean4301.vsnrma) = tissues.unique

for (t in tissues.unique) {
  index.tissue = which(tissue.names == t) # alternativ: which(tissues.all %in% t)
  if (length(index.tissue) > 1) {
    mean4301.vsnrma[, t] = apply(exprs(mouse4301.vsnrma)[,index.tissue], 1,
    mean, na.rm=T)
  }
}

#boxplot mouse 4301 mean vsnrma

x11(w=10,h=7)
par(las=2)
mmi=c(2.3,0.7,1.0477939,0.5366749)
par(mai=mmi)
boxplot(data.frame(mean4301.vsnrma), col=rainbow(100), cex.axis=0.6)
title(main="Mouse 4302 - distribution of averaged expressions\n
```

```
in 91 mouse tissues")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.4302.mean.vsnrma.eps")

#human roth

tissue.names=colnames(exprs(data.human.roth))
tissues=substr(1,length)

colnames(exprs(human.roth.vsnrma))=tissues.human.roth

#scatter plots
#-------------
#mouse gngnf1

# x11(w=9,h=5)
# par(mfrow=c(1,2))

tiss.index=which(tissue.names.mouse=="trachea")

plot(exprs(mouse.vsnrma)[,c(tiss.index[1],tiss.index[2])],pch=".")
abline(0,1,col="red")

title(main="Scatterplot")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="scatterplot.gngnf1.Atrachea.eps")

#mouse 4302
#adipose_white_B, amygdala_A, iris_B, spinal_cord_B
#6,9,72,158

tiss.index=which(tissues.mouse4302=="spinal_cord")

plot(exprs(mouse4301.vsnrma)[,c(tiss.index[1],tiss.index[2])],pch=".")
abline(0,1,col="red")

title(main="Scatterplot")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="scatterplot.mouse4302.spinal_cord.eps")

#human roth
#bronchus_4,corpus_callosum,kidney_cortex_2,midbrain_5,9,10
#ovary_8,oral_mucosa_1,oral_mucosa_2,ventral_tegmental_area_4
#22,84,127,143,145,146,197,313

tiss.index=which(tissue.names.human.roth=="ventral_tegmental_area")

x11(w=9,h=10)
par(mfrow=c(3,3))

plot(exprs(human.roth.vsnrma)[,c(tiss.index[6],tiss.index[1])],pch=".")
abline(0,1,col="red")

title(main="Scatterplot")

plot(exprs(human.roth.vsnrma)[,c(tiss.index[6],tiss.index[2])],pch=".")
abline(0,1,col="red")

title(main="human Roth data")
```

```
plot(exprs(human.roth.vsnrma)[,c(tiss.index[6],tiss.index[3])],pch=".")
abline(0,1,col="red")

plot(exprs(human.roth.vsnrma)[,c(tiss.index[6],tiss.index[4])],pch=".")
abline(0,1,col="red")

plot(exprs(human.roth.vsnrma)[,c(tiss.index[6],tiss.index[5])],pch=".")
abline(0,1,col="red")

plot(exprs(human.roth.vsnrma)[,c(tiss.index[6],tiss.index[7])],pch=".")
abline(0,1,col="red")

plot(exprs(human.roth.vsnrma)[,c(tiss.index[6],tiss.index[8])],pch=".")
abline(0,1,col="red")

plot(exprs(human.roth.vsnrma)[,c(tiss.index[1],tiss.index[2])],pch=".")
abline(0,1,col="red")

plot(exprs(human.roth.vsnrma)[,c(tiss.index[1],tiss.index[3])],pch=".")
abline(0,1,col="red")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="scatterplot.human.roth.ventral_tegmental_area_9.eps")

#midbrain 5,9,10, ovary 8 raus
#143,145,146,197,313,334 raus

human.roth.index.out=c(143,145,146,197,313,334)

human.roth.vsnrma.all=human.roth.vsnrma
human.roth.vsnrma=human.roth.vsnrma.all[,-human.roth.index.out]

#347 chips, left over, aber 353 filenames

human.roth.tissues

#nach chip rausnehmen, tissues noch einmal neu festlegen
#-------------------------------------------------------
human.roth.tissues=colnames(exprs(human.roth.vsnrma))

a=human.roth.tissues
b=sub("10","0",a)
c=substr(b,1,nchar(b)-2)
d=unique(c)

roth.tissues.all=(c)
roth.tissues=(d)

tissue.names=sub("10","0",human.roth.tissues)
tissue.names1=substr(tissue.names,1,nchar(tissue.names)-2)
tissues.unique=unique(tissue.names1)

#5.1 mean.vsnrma
#---------------

mean.human.roth.vsnrma = matrix(NA, nr=nrow(exprs(human.roth.vsnrma)), nc=65)
rownames(mean.human.roth.vsnrma) = featureNames(human.roth.vsnrma)
colnames(mean.human.roth.vsnrma) = tissues.unique

for (t in tissues.unique) {
  index.tissue = which(tissue.names1 == t) # alternativ: which(tissues.all %in% t)
```

```
  if (length(index.tissue) > 1) {
    mean.human.roth.vsnrma[, t] = apply(exprs(human.roth.vsnrma)[,index.tissue], 1,
    mean, na.rm=T)
  }
}

setwd("/home/dinkelac/data/mouse/sessions/rda")
save.image(file="mean.vsnrma.human.roth.rda")

#boxplot of mean.vsnrma
#----------------------
x11(w=10,h=7)
par(las=2)
mmi=c(2.3,0.7,1.0477939,0.5366749)
par(mai=mmi)
boxplot(data.frame(mean.human.roth.vsnrma), col=rainbow(100), cex.axis=0.6)
title(main="Human - distribution of averaged expressions\n
in 65 human tissues after vsnrma normalization (Roth data)")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.human.roth.mean.vsnrma.eps")

#human novartis
#--------------
#26,74,83,84,92,95,126
#trigeminal ganglion, adrenal cortex, uterus corpus (84),
#testis germ cell, testis leydig cell, cardiac myocytes (126)

pheno.data.human.novartis[26,]
tissue.names.human.novartis[26]
tissues.human.novartis=substr(tissue.names.human.novartis,0,
nchar(tissue.names.human.novartis)-2)

#umbenennen
tiss.index=which(tissues.human.novartis=="TestisLeydigCell")

plot(exprs(human.vsnrma)[,c(tiss.index[1],tiss.index[2])],pch=".")
abline(0,1,col="red")

title(main="Scatterplot")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="scatterplot.human.novartis.CardiacMyocytes.eps")

human.novartis.index.out=c(84,126)

colnames(human.vsnrma)=tissue.names.human.novartis

human.novartis.vsnrma=human.vsnrma[,-human.novartis.index.out]

tissues.human.novartis.unique=unique(tissues.human.novartis)

tissues1=substr(colnames(human.novartis.vsnrma),1,
nchar(colnames(human.novartis.vsnrma))-2)

mean.human.vsnrma = matrix(NA, nr=nrow(exprs(human.novartis.vsnrma)), nc=79)
rownames(mean.human.vsnrma) = featureNames(human.novartis.vsnrma)
colnames(mean.human.vsnrma) = tissues.human.novartis.unique

for (t in tissues.human.novartis.unique) {
  index.tissue = which(tissues1 == t) # alternativ: which(tissues.all %in% t)
  if (length(index.tissue) > 1) {
```

```
    mean.human.vsnrma[, t] = apply(exprs(human.novartis.vsnrma)[,index.tissue], 1,
    mean, na.rm=T)
  }
  else {
  print(t)
    mean.human.vsnrma[, t] = (exprs(human.novartis.vsnrma)[,index.tissue])
  }
}

#boxplot of mean.human.vsnrma

x11(w=10,h=7)
par(las=2)
mmi=c(2.3,0.7,1.0477939,0.5366749)
par(mai=mmi)
boxplot(data.frame(mean.human.vsnrma), col=rainbow(100), cex.axis=0.6)
title(main="Human - distribution of averaged expressions\n
in 79 human tissues after vsnrma normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.human.mean.vsnrma.eps")

setwd("/home/dinkelac/data/mouse/sessions/rda")
save.image(file="mean.data.rda")

#Piecharts tissue types
#----------------------

x11(w=10,h=8)

slices <- rep(2,length(tissues.unique))
pie(slices, labels = tissues.unique, main="53 Tissue Types GTEX data",
col=farben.gtex.data, cex=0.8)

setwd("/home/dinkelac/data/mouse/plots")

dev.copy2eps(file="piechart.tissue.types.gtex.data.eps")
```

1.1.2 Reading in and Pre Processing of the Human GTEX data

This script is reading in the RPKM values of the human GTEX data and analysing it for tissue restricted antigens.

```
#analyse_human_gtex_data.R
#--------------------------

#get GTEX data from GTEX, download RPKM values per transcript

setwd("/home/dinkelac/data/GTEX/")

gtex_data=read.table("GTEx_Analysis_V4_RNA-seq_Flux1.6_transcript_rpkm.txt",header=T)

#Spalten sind Transcript ID, Gene ID, Chr. Coord, 2920 Gewebe

#read in Sample Annotation

gtex_annotation=read.csv(file="GTEx_Data_V4_Annotations_SampleAttributesDS.txt",sep="\t")
```

```
gtex.names.annotation=as.character(gtex_annotation[,1])

# [1] "GTEX-N7MS-0007-SM-26GME" "GTEX-N7MS-0007-SM-26GMV"
# [3] "GTEX-N7MS-0007-SM-2D43E" "GTEX-N7MS-0007-SM-2D7W1"
# [5] "GTEX-N7MS-0008-SM-4E3JI" "GTEX-N7MS-0009-SM-2BWY4"

gtex.names.annotation.new=gsub("\\.","-",gtex.names.annotation)
#4501

gtex.names.data=names(gtex_data)[-c(1:4)]
gtex.names.data.new=gsub("\\.","-",gtex.names.data)
#2916

intersect(gtex.names.annotation.new,gtex.names.data.new)
#2916

setwd("/icgc/dkfzlsdf/analysis/G200/dinkelac/sessions/rda/")
save.image(file="gtex.data.new.rda")

gtex.tissue=as.character(gtex_annotation[,7])
tissue.type=sort(unique(gtex.tissue))
tissue.type.one=tissue.type[2:55]
#54 tissue types

hist.tissue=vector(len=length(tissue.type.one))
names(hist.tissue)=(tissue.type.one)

for(i in 1:length(tissue.type.one)){
tiss=tissue.type.one[i]
hist.tissue[i]=sum(gtex.tissue==tiss)
}

#letzte loeschen
#hist.tissue=hist.tissue[2:55]

x11(w=10,h=7)
mmi=c(3,0.7,2,0.5366749)
par(mai=mmi)

barplot(hist.tissue,col=c(rep("blue",7),rep("green",13),rep("blue",12),rep("red",2),
rep("blue",13)),main="Frequency of tissue types in the GTEX data set\n\n\n",las=2,
cex.names=0.75)
mtext("(max sample size = 648 whole blood \nmin sample size = 4 endocervix\n
average sample size = 83 all tissues)\n\n")

abline(h=c(100,200,300,400,500,600),col="grey")
abline(h=c(648),col="red")
abline(h=c(83),col="green")
abline(h=c(4),col="blue")

mtext("max",side=4,adj=1.1,col="red")
mtext("average",side=4,adj=0.1,col="green")
mtext("min",side=4,adj=-0.1,col="blue")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="sample.size.per.tissue.gtex.data.eps")

setwd("/home/dinkelac/data/mouse/tables")
write.table(hist.tissue,file="frequency.gtex.tissue.type.csv")

#tissue grouping
#---------------
```

```
> tissue.type

[2] "Adipose - Subcutaneous"                #----adipose tissue---#
[3] "Adipose - Visceral (Omentum)"          #----adipose tissue---#

[4] "Adrenal Gland"

[5] "Artery - Aorta"                         #------heart-----#
[6] "Artery - Coronary"                      #------heart-----#
[7] "Artery - Tibial"                        #------heart-----#

[8] "Bladder"

[9] "Brain - Amygdala"                                  #-----cns----#
[10] "Brain - Anterior cingulate cortex (BA24)" #-----cns----#
[11] "Brain - Caudate (basal ganglia)"                  #-----cns----#
[12] "Brain - Cerebellar Hemisphere"                    #-----cns----#
[13] "Brain - Cerebellum"                               #-----cns----#
[14] "Brain - Cortex"                                   #-----cns----#
[15] "Brain - Frontal Cortex (BA9)"                     #-----cns----#
[16] "Brain - Hippocampus"                              #-----cns----#
[17] "Brain - Hypothalamus"                             #-----cns----#
[18] "Brain - Nucleus accumbens (basal ganglia)"#-----cns----#
[19] "Brain - Putamen (basal ganglia)"                  #-----cns----#
[20] "Brain - Spinal cord (cervical c-1)"               #-----cns----#
[21] "Brain - Substantia nigra"                         #-----cns----#

[22] "Breast - Mammary Tissue"                          #-----mammary gland----#

[23] "Cells - EBV-transformed lymphocytes"              #---- cell lines, out----#
[24] "Cells - Leukemia cell line (CML)"                 #---- cell lines, out----#
[25] "Cells - Transformed fibroblasts"                  #---- cell lines, out----

[26] "Cervix - Ectocervix"                              #----uterus----#
[27] "Cervix - Endocervix"                              #----uterus----#

[28] "Colon - Sigmoid"                                  #-----colon------#
[29] "Colon - Transverse"                               #-----colon------#

[30] "Esophagus - Gastroesophageal Junction"    #------esophagus--------#
[31] "Esophagus - Mucosa"                               #------esophagus--------#
[32] "Esophagus - Muscularis"                           #------esophagus--------#

[33] "Fallopian Tube"                                   #--------uterus------#

[34] "Heart - Atrial Appendage"                         #----heart----#
[35] "Heart - Left Ventricle"                           #----heart----#

[36] "Kidney - Cortex"
[37] "Liver"
[38] "Lung"

[39] "Minor Salivary Gland"                             #-----salivary gland----#

[40] "Muscle - Skeletal"

[41] "Nerve - Tibial"                                   #-----peripheral nervous system? (pns)--------#

[42] "Ovary"
[43] "Pancreas"
[44] "Pituitary"
```

```
[45] "Prostate"

[46] "Skin - Not Sun Exposed (Suprapubic)"       #-----epidermis-----#
[47] "Skin - Sun Exposed (Lower leg)"            #-----epidermis-----#

[48] "Small Intestine - Terminal Ileum"          #-----intestine-------#
[49] "Spleen"
[50] "Stomach"
[51] "Testis"
[52] "Thyroid"

[53] "Uterus"                                  #----uterus-------#
[54] "Vagina"                                  #----uterus-------#

[55] "Whole Blood"

#Zuordnung der Samples zu Tissues
#--------------------------------
names(gtex.tissue)=gtex.names.annotation.new
gtex.data.renames=gtex.tissue[gtex.names.data.new]

names(gtex_data)[5:2920]=gtex.data.renames
pdata=gtex_data[,c(1:4)]
data.matrix=as.matrix(gtex_data[,-c(1:4)])

x=apply(data.matrix, 2, as.numeric)
names(x)=as.character(gtex.data.renames)
gtex.data=cbind(pdata,x)

Namen=names(gtex.data)
neue_Namen=strsplit(Namen,"[.]")
new_Names=vector(len=length(neue_Namen))

for(i in 1:length(neue_Namen)){
new_Names[i]=neue_Namen[[i]][1]
}

setwd("/icgc/dkfzlsdf/analysis/G200/dinkelac/sessions/rda/")
save.image(file="gtex.data.16.01.2017.rda")

#doppelte Transcripte
tissues=unique(gtex.tissue)
tissues.unique=sort(tissues[1:54])

setwd("/home/dinkelac/data/mouse/tables/tissues/mouse/")
farben.gngnf1=read.csv("tissue.colors.gngnf1.csv",sep="\t",header=F)

farben.mouse=as.character(farben.gngnf1[,4])
names(farben.mouse)=as.character(farben.gngnf1[,2])

setwd("/home/dinkelac/data/mouse/tables/tissues/human/human_roth/")
x=read.csv("tissue_colors_human_roth.csv",sep="\t",header=FALSE)

farben.human.roth=as.character(x[,6])
names(farben.human.roth)=as.character(x[,4])

tiss=c("Whole Blood","Cells - Transformed fibroblasts","Brain - Frontal Cortex (BA9)",
"Brain - Cerebellar Hemisphere","Brain - Hippocampus","Brain - Substantia nigra",
"Brain - Anterior cingulate cortex (BA24)","Brain - Amygdala",
"Brain - Caudate (basal ganglia)","Brain - Nucleus accumbens (basal ganglia)",
"Brain - Putamen (basal ganglia)","Brain - Hypothalamus","Testis",
"Skin - Sun Exposed (Lower leg)","Adipose - Subcutaneous","Muscle - Skeletal",
```

```
"Nerve - Tibial","Artery - Tibial","Heart - Left Ventricle","Lung",
"Esophagus - Muscularis","Esophagus - Mucosa","Kidney - Cortex","Thyroid",
"Brain - Cortex","Brain - Cerebellum","Pituitary","Uterus","Artery - Aorta",
"Pancreas","Vagina","Stomach","Adrenal Gland","Colon - Transverse","Prostate",
"Brain - Spinal cord (cervical c-1)","Artery - Coronary","Spleen","Liver",
"Fallopian Tube","Ovary","Breast - Mammary Tissue","Cells - EBV-transformed lymphocytes",
"Bladder","Cervix - Ectocervix","Cervix - Endocervix","Skin - Not Sun Exposed (Suprapubic)",
"Heart - Atrial Appendage","Esophagus - Gastroesophageal Junction",
"Adipose - Visceral (Omentum)","Small Intestine - Terminal Ileum","Colon - Sigmoid",
"Minor Salivary Gland","Cells - Leukemia cell line (CML)")

#Farben raus finden mit:
#farben.human.roth[which(names(farben.human.roth)=="thyroid")]

farben.gtex.data=c("#00B4FFFF","#00B4FFFF","violet",rep("red",3),"rosybrown",
rep("green",13),"violet","white","white","#00B4FFFF","#00B4FFFF","#3900FFFF",
"#3900FFFF","yellow","yellow","yellow","rosybrown","red","red","brown","cornflowerblue",
"slateblue","violet","lightblue","#FF8100FF","cornflowerblue","navy","#9000FFFF","gold",
"navajowhite","navajowhite","blue","seashell2","#00B4FFFF","thistle","#005DFFFF",
"navajowhite","navajowhite","red")
names(farben.gtex.data)=tiss

setwd("/home/dinkelac/data/mouse/tables/tissues/")
write.table(farben.gtex.data,file="farben.gtex.data.csv",sep=",")

Ensembl.Transcript.IDs=as.character(gtex.data$TargetID)
doppelte.transcripte=which(duplicated(Ensembl.Transcript.IDs))
#62

Ensembl.Transcript.IDs.unique=Ensembl.Transcript.IDs[-c(doppelte.transcripte)]
#194783

#tissues.unique=tissues.unique[-c(23)]
gtex.data.sum=matrix(nrow=length(Ensembl.Transcript.IDs),ncol=53,
dimnames=list(Ensembl.Transcript.IDs,tissues.unique))

tissues.unique[2]="Adipose - Visceral"
tissues.unique[9]="Brain - Anterior cingulate cortex"
tissues.unique[10]="Brain - Caudate"
tissues.unique[14]="Brain - Frontal Cortex"
tissues.unique[17]="Brain - Nucleus accumbens"
tissues.unique[18]="Brain - Putamen"
tissues.unique[19]="Brain - Spinal cord"
tissues.unique[44]="Skin - Not Sun Exposed"
tissues.unique[45]="Skin - Sun Exposed"

#plot GTEX data
#--------------

for(i in 1:length(tissues.unique)){
gewebe=tissues.unique[i]
ind=grep(gewebe,names(gtex.data))
gtex.data.sum[,i]=apply(gtex.data[,ind],1,median)
}

setwd("/icgc/dkfzlsdf/analysis/G200/dinkelac/sessions/rda/")
save.image(file="gtex.data.11.01.2017.rda")

# order=c(1:3,7:20,42,39,21:30,4:6,33:38,40,41,43:52,31,53)

x11(w=10,h=7)
#90 Grad beschriften
```

```
par(las=2)
#Rand zum Beschriften unten, li, oben, re
mmi=c(3,0.7,1.0477939,0.5366749)
par(mai=mmi)

transcript.name=dimnames(gtex.data.sum)[[1]][i]
gene.symbol=human.ensembl.87.symbols[substr(transcript.name,0,nchar(transcript.name)-2)]
title=paste(transcript.name,"\n",gene.symbol)

i=2

barplot(gtex.data.sum[i,],col=farben.gtex.data,cex.names=0.75,main=title)

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="gtex.sample.zwei.eps")

#calculate TRAs
#----------------

#fuer jedes Transcript
adipose.gtex.names=c("Adipose - Subcutaneous","Adipose - Visceral")
heart.gtex.names=c("Artery - Aorta","Artery - Coronary","Artery - Tibial",
"Heart - Atrial Appendage","Heart - Left Ventricle")
cns.gtex.names=c("Brain - Amygdala","Brain - Anterior cingulate cortex",
"Brain - Caudate","Brain - Cerebellar Hemisphere","Brain - Cerebellum",
"Brain - Cortex","Brain - Frontal Cortex","Brain - Hippocampus","Brain - Hypothalamus",
"Brain - Nucleus accumbens","Brain - Putamen","Brain - Spinal cord",
"Brain - Substantia nigra","Pituitary")
cell.lines.gtex.names=c("Cells - EBV-transformed lymphocytes",
"Cells - Transformed fibroblasts")
#leukemia cells ausgeschlossen

uterus.gtex.names=c("Cervix - Ectocervix","Cervix - Endocervix","Uterus","Vagina")
colon.gtex.names=c("Colon - Sigmoid","Colon - Transverse")
esophagus.gtex.names=c("Esophagus - Gastroesophageal Junction","Esophagus - Mucosa",
"Esophagus - Muscularis")
ovary.gtex.names=c("Ovary","Fallopian Tube")
epidermis.gtex.names=c("Skin - Not Sun Exposed","Skin - Sun Exposed")

# > length(adipose.gtex.names)
# [1] 2
# > length(heart.gtex.names)
# [1] 5
# > length(cns.gtex.names)
# [1] 14
# > length(cell.lines.gtex.names)
# [1] 2
# > length(uterus.gtex.names)
# [1] 4
# > length(colon.gtex.names)
# [1] 2
# > length(esophagus.gtex.names)
# [1] 3
# > length(ovary.gtex.names)
# [1] 2
# > length(epidermis.gtex.names)
# [1] 2

adipose.gtex=which(colnames(gtex.data.sum)%in%adipose.gtex.names==T)
heart.gtex=which(colnames(gtex.data.sum)%in%heart.gtex.names==T)
cns.gtex=which(colnames(gtex.data.sum)%in%cns.gtex.names==T)
cell.lines.gtex=which(colnames(gtex.data.sum)%in%cell.lines.gtex.names==T)
```

```
uterus.gtex=which(colnames(gtex.data.sum)%in%uterus.gtex.names==T)
colon.gtex=which(colnames(gtex.data.sum)%in%colon.gtex.names==T)
esophagus.gtex=which(colnames(gtex.data.sum)%in%esophagus.gtex.names==T)
ovary.gtex=which(colnames(gtex.data.sum)%in%ovary.gtex.names==T)
epidermis.gtex=which(colnames(gtex.data.sum)%in%epidermis.gtex.names==T)

count.over.median.gtex.data=function(evec,crit){
x=(evec > (median(evec)*crit))

if(length(unique(x)==1&is.na(unique(x))))sumx=0 else{
if(sum(x[cns.gtex]>0))sumx=1 else sumx=0
if(sum(x[uterus.gtex]>0))sumx=sumx+1
if(sum(x[epidermis.gtex]>0))sumx=sumx+1
if(sum(x[heart.gtex]>0))sumx=sumx+1
if(sum(x[esophagus.gtex]>0))sumx=sumx+1
if(sum(x[colon.gtex]>0))sumx=sumx+1
if(sum(x[adipose.gtex]>0))sumx=sumx+1
if(sum(x[ovary.gtex]>0))sumx=sumx+1
}
sumx=sumx+sum(x[-c(cns.gtex,uterus.gtex,epidermis.gtex,heart.gtex,esophagus.gtex,
colon.gtex,adipose.gtex,ovary.gtex,cell.lines.gtex)])
}

calculate.tra.gtex=function(crit){
x=apply(gtex.data.sum,1,count.over.median.gtex.data,crit=crit)
tra.index=which(x>0&x<6)
return(tra.index)
}

calculate.tra.gtex.cutoff.eight=function(crit){
x=apply(gtex.data.sum,1,count.over.median.gtex.data,crit=crit)
tra.index=which(x>0&x<8)
return(tra.index)
}

tra.3x=calculate.tra.gtex(3)
#71297(390)
tra.5x=calculate.tra.gtex(5)
#60131(333)
tra.10x=calculate.tra.gtex(10)
#51352(302)
tra.20x=calculate.tra.gtex(20)
#46776(267)

tra.3x.eight=calculate.tra.gtex.cutoff.eight(3)
#77407
tra.5x.eight=calculate.tra.gtex.cutoff.eight(5)
#64861
tra.10x.eight=calculate.tra.gtex.cutoff.eight(10)
#55606
tra.20x.eight=calculate.tra.gtex.cutoff.eight(20)
#50834

tra.3x.median.gtex=apply(gtex.data.sum,1,count.over.median.gtex.data,crit=3)
tra.5x.median.gtex=apply(gtex.data.sum,1,count.over.median.gtex.data,crit=5)
tra.10x.median.gtex=apply(gtex.data.sum,1,count.over.median.gtex.data,crit=10)
tra.20x.median.gtex=apply(gtex.data.sum,1,count.over.median.gtex.data,crit=20)

tra.3x.median.gtex.new=tra.3x.median.gtex[-c(which(is.na(tra.3x.median.gtex)==T))]
tra.5x.median.gtex.new=tra.5x.median.gtex[-c(which(is.na(tra.5x.median.gtex)==T))]
tra.10x.median.gtex.new=tra.10x.median.gtex[-c(which(is.na(tra.10x.median.gtex)==T))]
tra.20x.median.gtex.new=tra.20x.median.gtex[-c(which(is.na(tra.20x.median.gtex)==T))]
```

```
maxtiss.over.3xmedian = vector(len=53)
maxtiss.over.5xmedian = vector(len=53)
maxtiss.over.10xmedian = vector(len=53)
maxtiss.over.20xmedian = vector(len=53)

for (i in 1:53){
maxtiss.over.3xmedian[i] = sum(tra.3x.median.gtex.new> 0 & tra.3x.median.gtex.new < i)
maxtiss.over.5xmedian[i] = sum(tra.5x.median.gtex.new> 0 & tra.5x.median.gtex.new < i)
maxtiss.over.10xmedian[i] = sum(tra.10x.median.gtex.new> 0 & tra.10x.median.gtex.new < i)
maxtiss.over.20xmedian[i] = sum(tra.20x.median.gtex.new> 0 & tra.20x.median.gtex.new < i)
}

# graphik plotten
#-----------------
x11(h=8,w=8)
plot(maxtiss.over.3xmedian, type="p",xlab="number of tissues",ylab="number of transcripts")
points(maxtiss.over.5xmedian, col="red")
points(maxtiss.over.10xmedian, col="green")
points(maxtiss.over.20xmedian, col="blue")
abline(v=5)
legend("topright",pch="o",c("","3x over median","5x over median","10x over median",
"20x over median"),col=c("white","black","red","green","blue")
)

title(main="Number of transcripts over 3x, 5x, 10x, 20x the median \n
in 53 human tissues (GTEX data)")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="saturationplot.human.gtex.5.eps")

#annotate tables
setwd("/home/dinkelac/data/mouse/tables")

human.ensembl.87.table=read.csv(file="ensembl_human_87.txt",sep=",")
#242903
human.ensembl.87.transcripte=as.character(human.ensembl.87.table[,2])
#215929
doppelte.transcripte=which(duplicated(human.ensembl.87.transcripte))

human.ensembl.87.gene=as.character(human.ensembl.87.table[,1])
names(human.ensembl.87.gene)=as.character(human.ensembl.87.table[,2])
#63305 gene IDs

human.ensembl.87.genes.new=human.ensembl.87.gene[-doppelte.transcripte]
human.ensembl.87.symbols=as.character(human.ensembl.87.table[,3])
names(human.ensembl.87.symbols)=as.character(human.ensembl.87.table[,2])
human.ensembl.87.symbols.new=human.ensembl.87.symbols[-doppelte.transcripte]
#35755 symbols unique

human.ensembl.87.unigene=as.character(human.ensembl.87.table[,4])
names(human.ensembl.87.unigene)=as.character(human.ensembl.87.table[,2])
human.ensembl.87.unigene.new=human.ensembl.87.unigene[-doppelte.transcripte]
#23003 unigene IDs unique

human.ensembl.87.chrom=as.character(human.ensembl.87.table[,5])
names(human.ensembl.87.chrom)=as.character(human.ensembl.87.table[,2])
human.ensembl.87.chrom.new=human.ensembl.87.chrom[-doppelte.transcripte]
#350 chomosomal Ids (CTG Islands)

human.ensembl.87.start=as.character(human.ensembl.87.table[,6])
names(human.ensembl.87.start)=as.character(human.ensembl.87.table[,2])
```

```
human.ensembl.87.start.new=human.ensembl.87.start[-doppelte.transcripte]
#62309 start sites

human.ensembl.87.entrez=as.character(human.ensembl.87.table[,7])
names(human.ensembl.87.entrez)=as.character(human.ensembl.87.table[,2])
human.ensembl.87.entrez.new=human.ensembl.87.entrez[-doppelte.transcripte]
#25052 entrez Ids

#tissues over the cutoff for all TRA lists
#----------------------------------------
#5 tissues over cutoff
#---------------------
tissues.tra.3x=tra.3x

for(i in 1:length(tra.3x)){
tissues.tra.3x[i]=paste(names(which(gtex.data.sum[tra.3x[i],]>
3*median(gtex.data.sum[tra.3x[i],]))),collapse="/")
}

tissues.tra.5x=tra.5x

for(i in 1:length(tra.5x)){
tissues.tra.5x[i]=paste(names(which(gtex.data.sum[tra.5x[i],]>
5*median(gtex.data.sum[tra.5x[i],]))),collapse="/")
}

tissues.tra.10x=tra.10x

for(i in 1:length(tra.10x)){
tissues.tra.10x[i]=paste(names(which(gtex.data.sum[tra.10x[i],]>
10*median(gtex.data.sum[tra.10x[i],]))),collapse="/")
}

tissues.tra.20x=tra.20x

for(i in 1:length(tra.20x)){
tissues.tra.20x[i]=paste(names(which(gtex.data.sum[tra.20x[i],]>
20*median(gtex.data.sum[tra.20x[i],]))),collapse="/")
}

#8 tissues over cutoff
#---------------------
tissues.tra.3x.eight=tra.3x.eight

for(i in 1:length(tra.3x.eight)){
tissues.tra.3x.eight[i]=paste(names(which(gtex.data.sum[tra.3x.eight[i],]>
3*median(gtex.data.sum[tra.3x.eight[i],]))),collapse="/")
}

tissues.tra.5x.eight=tra.5x.eight

for(i in 1:length(tra.5x.eight)){
tissues.tra.5x.eight[i]=paste(names(which(gtex.data.sum[tra.5x.eight[i],]>
5*median(gtex.data.sum[tra.5x.eight[i],]))),collapse="/")
}

tissues.tra.10x.eight=tra.10x.eight

for(i in 1:length(tra.10x.eight)){
tissues.tra.10x.eight[i]=paste(names(which(gtex.data.sum[tra.10x.eight[i],]>
10*median(gtex.data.sum[tra.10x.eight[i],]))),collapse="/")
}
```

```
tissues.tra.20x.eight=tra.20x.eight

for(i in 1:length(tra.20x.eight)){
tissues.tra.20x.eight[i]=paste(names(which(gtex.data.sum[tra.20x.eight[i],]>
20*median(gtex.data.sum[tra.20x.eight[i],]))),collapse="/")
}

#max tissues over the cutoff for all TRA lists
#-------------------------------------------
max.tissue.tra.3x=tra.3x

for(i in 1:length(tra.3x)){
max.tissue.tra.3x[i]=names(which(gtex.data.sum[tra.3x[i],]==
max(gtex.data.sum[tra.3x[i],])))
}

max.tissue.tra.5x=tra.5x

for(i in 1:length(tra.5x)){
max.tissue.tra.5x[i]=names(which(gtex.data.sum[tra.5x[i],]==
max(gtex.data.sum[tra.5x[i],])))
}

max.tissue.tra.10x=tra.10x

for(i in 1:length(tra.10x)){
max.tissue.tra.10x[i]=names(which(gtex.data.sum[tra.10x[i],]==
max(gtex.data.sum[tra.10x[i],])))
}

max.tissue.tra.20x=tra.20x

for(i in 1:length(tra.20x)){
max.tissue.tra.20x[i]=names(which(gtex.data.sum[tra.20x[i],]==
max(gtex.data.sum[tra.20x[i],])))
}

#8 tissues over cutoff
#---------------------
max.tissue.tra.3x.eight=tra.3x.eight

for(i in 1:length(tra.3x.eight)){
max.tissue.tra.3x.eight[i]=names(which(gtex.data.sum[tra.3x.eight[i],]==
max(gtex.data.sum[tra.3x.eight[i],])))
}

max.tissue.tra.5x.eight=tra.5x.eight

for(i in 1:length(tra.5x.eight)){
max.tissue.tra.5x.eight[i]=names(which(gtex.data.sum[tra.5x.eight[i],]==
max(gtex.data.sum[tra.5x.eight[i],])))
}

max.tissue.tra.10x.eight=tra.10x.eight

for(i in 1:length(tra.10x.eight)){
max.tissue.tra.10x.eight[i]=names(which(gtex.data.sum[tra.10x.eight[i],]==
max(gtex.data.sum[tra.10x.eight[i],])))
}

max.tissue.tra.20x.eight=tra.20x.eight
```

```
for(i in 1:length(tra.20x.eight)){
max.tissue.tra.20x.eight[i]=names(which(gtex.data.sum[tra.20x.eight[i],]==
max(gtex.data.sum[tra.20x.eight[i],])))
}

#tra listen annotieren
#----------------------
ensembl.transcript.IDs=names(tra.20x.eight)
ensembl.transcript=substr(names(tra.20x.eight),0,15)
ensembl.gene=human.ensembl.87.genes.new[ensembl.transcript]
ensembl.symbol=human.ensembl.87.symbols.new[ensembl.transcript]
ensembl.unigene=human.ensembl.87.unigene.new[ensembl.transcript]
ensembl.entrez=human.ensembl.87.entrez.new[ensembl.transcript]
ensembl.chrom=human.ensembl.87.chrom.new[ensembl.transcript]
ensembl.start=human.ensembl.87.start.new[ensembl.transcript]

tiss.number=tra.20x.median.gtex[ensembl.transcript.IDs]
tissues=tissues.tra.20x.eight
max.tissue=max.tissue.tra.20x.eight

tra.human.gtex.20x.table.eight=cbind(ensembl.transcript,ensembl.gene,ensembl.symbol,
ensembl.unigene,ensembl.entrez,ensembl.chrom,ensembl.start,tiss.number,tissues,max.tissue)

#write tables
setwd("/home/dinkelac/data/mouse/tables")

#5 tissues
#---------
write.table(tra.human.gtex.3x.table,"tra.2017.human.gtex.3x.table.csv",row.names=F,sep="\t")
#29625 genes
write.table(tra.human.gtex.5x.table,"tra.2017.human.gtex.5x.table.csv",row.names=F,sep="\t")
#27339 genes
write.table(tra.human.gtex.10x.table,"tra.2017.human.gtex.10x.table.csv",row.names=F,sep="\t")
#25145 genes
write.table(tra.human.gtex.20x.table,"tra.2017.human.gtex.20x.table.csv",row.names=F,sep="\t")
#23808 genes

#8 tissues
#---------
write.table(tra.human.gtex.3x.table.eight,"tra.2017.human.gtex.3x.table.eight.csv",
row.names=F,sep="\t")
#30899 genes
write.table(tra.human.gtex.5x.table.eight,"tra.2017.human.gtex.5x.table.eight.csv",
row.names=F,sep="\t")
#28474 genes
write.table(tra.human.gtex.10x.table.eight,"tra.2017.human.gtex.10x.table.eight.csv",
row.names=F,sep="\t")
#26369 genes
write.table(tra.human.gtex.20x.table.eight,"tra.2017.human.gtex.20x.table.eight.csv",
row.names=F,sep="\t")
#25076 genes

#Verteilung der RPKM values pro tissue type in den ersten 1000 Transkripten

plot(gtex.data.sum[1,],col="red",type="p",ylim=c(0,1000),ylab="RPKM values",xlab="Tissue Types")
color=rainbow(1000)

for(i in 2:1000){
points(gtex.data.sum[i,],col=color[i])
}
```

```
title(main="The distribution of RPKM values in the GTEX Dataset\n
of the first 1000 Transcripts over 54 tissues")
setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="RPKM.value.per.tissue.type.first.1000.transcripts.eps")

max=apply(gtex.data.sum,2,max,na.rm=T)
barplot(max)

x11(w=10,h=7)
#90 Grad beschriften
par(las=2)
#Rand zum Beschriften unten, li, oben, re
mmi=c(3,1,1.0477939,0.5366749)
par(mai=mmi)

boxplot(log(gtex.data.sum),col=farben.gtex.data,cex.names=0.50,main="RPKM values per tissue type"
ylab="RPKM value")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="max.rpkm.per.tissue.type.eps")

tissues.unique[2]="Adipose - Visceral"
tissues.unique[9]="Brain - Anterior cingulate cortex"
tissues.unique[10]="Brain - Caudate"
tissues.unique[14]="Brain - Frontal Cortex"
tissues.unique[17]="Brain - Nucleus accumbens"
tissues.unique[18]="Brain - Putamen"
tissues.unique[19]="Brain - Spinal cord"
tissues.unique[44]="Skin - Not Sun Exposed"
tissues.unique[45]="Skin - Sun Exposed"

#gtex.data.sum=gtex.data.median
gtex.data.median=matrix(nrow=length(Ensembl.Transcript.IDs),ncol=53,
dimnames=list(Ensembl.Transcript.IDs,tissues.unique))

for(i in 1:length(tissues.unique)){
gewebe=tissues.unique[i]
ind=grep(gewebe,names(gtex.data))
gtex.data.median[,i]=apply(gtex.data[,ind],1,median)
}

gtex.data.mean=matrix(nrow=length(Ensembl.Transcript.IDs),ncol=53,
dimnames=list(Ensembl.Transcript.IDs,tissues.unique))

#plot GTEX data
#--------------
for(i in 1:length(tissues.unique)){
gewebe=tissues.unique[i]
ind=grep(gewebe,names(gtex.data))
gtex.data.mean[,i]=apply(gtex.data[,ind],1,mean)
}

gtex.data.var=matrix(nrow=length(Ensembl.Transcript.IDs),ncol=53,
dimnames=list(Ensembl.Transcript.IDs,tissues.unique))

#plot GTEX data
#--------------
for(i in 1:length(tissues.unique)){
gewebe=tissues.unique[i]
ind=grep(gewebe,names(gtex.data))
gtex.data.var[,i]=apply(gtex.data[,ind],1,var)
}
```

```
setwd("/icgc/dkfzlsdf/analysis/G200/dinkelac/sessions/rda/")
save.image(file="gtex.25.01.2017.rda")

#calculate mean over all tissues per transcript
# mean.rpkm=matrix(ncol=53,nrow=194844)
# colnames(mean.rpkm)=dimnames(gtex.data.sum)[[2]]
# rownames(mean.rpkm)=dimnames(gtex.data.sum)[[1]]

# for (i in 1:length(dimnames(gtex.data.sum)[[2]])){
# tiss.ind=which(dimnames(gtex.data.sum)[[2]]==dimnames(gtex.data.sum)[[2]][i])
# print(dimnames(gtex.data.sum)[[2]][i])
# mean.rpkm[,i]=apply(gtex.data.sum[,tiss.ind],1,mean)
# }

#median over all tissues per transcript
# median.rpkm=matrix(ncol=53,nrow=194844)
# colnames(median.rpkm)=dimnames(gtex.data.sum)[[2]]
# rownames(median.rpkm)=dimnames(gtex.data.sum)[[1]]
#
# for (i in 1:length(dimnames(gtex.data.sum)[[2]])){
# tiss.ind=which(names(gtex.data.sum)==dimnames(gtex.data.sum)[[2]][i])
# print(dimnames(gtex.data.sum)[[2]][i])
# median.rpkm[,i]=apply(gtex.data.sum[,tiss.ind],1,median)
# }

# mean.var.rpkm=matrix(ncol=53,nrow=194844)
# colnames(mean.var.rpkm)=dimnames(gtex.data.sum)[[2]]
# rownames(mean.var.rpkm)=dimnames(gtex.data.sum)[[1]]
#
# for (i in 1:length(dimnames(gtex.data.sum)[[2]])){
# tiss.ind=which(names(gtex.data.sum)==dimnames(gtex.data.sum)[[2]][i])
# print(dimnames(gtex.data.sum)[[2]][i])
# mean.var.rpkm[,i]=apply(gtex.data.sum[,tiss.ind],1,var)
# }

#berechne boxplot pro Transcript pro Tissue type

> head(gtex.data.mean)
                 Adipose - Subcutaneous Adipose - Visceral (Omentum)
ENST00000390859.1           0.000000e+00                  0.00000000
ENST00000290551.4           1.135218e+02                288.01169284
ENST00000475157.1           5.239265e-01                  1.26596419
ENST00000429660.1           3.663203e-04                  0.00000000
ENST00000473120.1           3.898236e-02                  0.04927048
ENST00000207457.3           1.069361e-02                  0.06866887

> head(gtex.data.var)
                 Adipose - Subcutaneous Adipose - Visceral (Omentum)
ENST00000390859.1           0.000000e+00                0.000000e+00
ENST00000290551.4           3.083492e+03                5.046076e+04
ENST00000475157.1           4.901084e-01                3.289320e+00
ENST00000429660.1           1.717639e-05                0.000000e+00
ENST00000473120.1           2.285956e-02                3.818877e-02
ENST00000207457.3           4.955633e-03                3.783810e-02

> head(gtex.data.median)
                 Adipose - Subcutaneous Adipose - Visceral (Omentum)
ENST00000390859.1            0.000000000                   0.0000000
ENST00000290551.4           55.529197773                 224.6347179
ENST00000475157.1            0.700077411                   1.8136482
ENST00000429660.1            0.004144441                   0.0000000
```

```
ENST00000473120.1            0.151193789                    0.1954195
ENST00000207457.3            0.070396257                    0.1945202

gtex.data.sd=sqrt(gtex.data.var)
gtex.data.mean.sd.up=gtex.data.mean+gtex.data.sd
gtex.data.mean.sd.down=gtex.data.mean-gtex.data.sd

#****
transcript.names=rownames(gtex.data.mean)
symbol.names=human.ensembl.87.symbols[substr(transcript.names,0,
nchar(transcript.names)-2)]

source("~/R-functions/errorbars.R")

x11(w=10,h=7)
par(las=2)
#Rand zum Beschriften unten, li, oben, re
mmi=c(2.5,0.9,1.0477939,0.5366749)
par(mai=mmi)

x=3
plot.transcript=function(x){

# title=paste(ensembl.symbols.new[substr(transcript.names[x],0,
nchar(transcript.names[x])-2)],"\n",transcript.names[x])

if(max(gtex.data.mean[x,])<30){
max.scale=30
}else{
if(max(gtex.data.mean[x,])<50){
max.scale=50
}else{
max.scale=round(max(gtex.data.mean[x,]),digits=-2)
}
}

if(median(gtex.data.mean[x,])*5>100&median(gtex.data.mean[x,])*5>max.scale){
max.scale=round(median(gtex.data.mean[x,])*5,digits=-2)
}

mean=gtex.data.mean[x,]
max=gtex.data.mean.sd.up[x,]
min=gtex.data.mean.sd.down[x,]

if(max(max)>max.scale){
max.scale=round(max(max),digits=-2)
}

barplot(gtex.data.mean[x,],col=farben.gtex.data,cex.names=0.7,ylab="RPKM value",
main=title,ylim=c(0,max.scale))
abline(h=median(gtex.data.mean[x,]),lty=3)
abline(h=median(gtex.data.mean[x,])*5,col="red",lty=3)

# points(mean.sd.up.rpkm[2,],col="red")
# points(mean.sd.down.rpkm[2,],col="green")
# errorbars

z=barplot(gtex.data.mean[x,],beside=T,plot=FALSE)

for(i in 1:length(gtex.data.mean[x,])){
ebars(z[i],mean[i],min[i],max[i])
}
```

```
}

#plot.transcript function

plot.transcript(2)

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="GTEX.Transcript.2.eps")

#boxplots

data=gtex.data.sum[1:10,c(61,141)]

x11(w=10,h=7)
#90 Grad beschriften
par(las=2)
#Rand zum Beschriften unten, li, oben, re
mmi=c(2.5,1,1.0477939,0.5366749)
par(mai=mmi)

ind=grep(gewebe,names(gtex.data))

boxplot.transcript=function(nr,crit){
title=paste(human.ensembl.87.symbols[substr(rownames(gtex.data.mean)[nr],0,
nchar(rownames(gtex.data.mean)[nr])-2)],"\n",substr(rownames(gtex.data.mean)[nr],0,
nchar(rownames(gtex.data.mean)[nr])-2))

median.expr=vector(len=53)
iqr.expr=vector(len=53)
box.matrix=matrix(ncol=53,nrow=190)

colnames(box.matrix)=tissues.unique

for(i in 1:length(tissues.unique)){
#tiss.ind=which(dimnames(gtex.data.mean)[[2]]==tissues.unique[i])
tiss.ind=grep(tissues.unique[i],names(gtex.data))
expr.nr=unlist(c(gtex.data[nr,tiss.ind]))
median.expr[i]=median(expr.nr)
iqr.expr[i]=quantile(expr.nr)[4]
box.matrix[c(1:length(expr.nr)),i]=expr.nr
}

loc.max=max(iqr.expr)

if(loc.max<10){
scale.max=10
}else{
if(loc.max<50){
scale.max=50
}else{
if(loc.max<100){
scale.max=100
}else{
if(loc.max<500){
scale.max=500
}else{
if(loc.max<1000){
scale.max=1000
}else{
scale.max=round(loc.max,digits=-2)
}
}
```

```
}
}
}

# print(scale.max)

boxplot(box.matrix,col=farben.gtex.data,cex.axis=0.7,ylab="RPKM value",main=title,ylim=
c(0,scale.max),outline=T)

#stripchart(box.matrix[,order],method="jitter",add=TRUE,pch=16,col="blue")

if(crit==3){
farbe="green"
}else{
if(crit==5){
farbe="red"
}else{
if(crit==10){
farbe="blue"
}else{
if(crit==20){
farbe="brown"
}
}
}
}

abline(h=median(median.expr),lty=3)
abline(h=median(median.expr)*crit,lty=3,col=farbe)
}

boxplot.transcript(2)

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="ENST00000472152_plot_with_errorbars.eps")

#plotten aller transcripte png
plotten.gtex.png=function(ind,crit){
gene.name=substr(transcript.names,0,15)[ind]
file.name=paste("human.gtex.",gene.name,".png",sep="")
#x11(w=10,h=7)
png(filename=file.name,width=1000,height=700,res=72)

#90 Grad beschriften
par(las=2)
#Rand zum Beschriften unten, li, oben, re
mmi=c(2.5,1,1.0477939,0.5366749)
par(mai=mmi)

boxplot.transcript(ind,crit)

if(crit==3){
legend("topright",pch="-",c("median","3 x median"),col=c("black","green"),cex=0.8)
}else{
if(crit==5){
legend("topright",pch="-",c("median","5 x median"),col=c("black","red"),cex=0.8)
}else{
if(crit==10){
legend("topright",pch="-",c("median","10 x median"),col=c("black","blue"),cex=0.8)
}else{
if(crit==20){
legend("topright",pch="-",c("median","20 x median"),col=c("black","brown"),cex=0.8)
```

```
}
}
}
}

dev.off()

}

setwd("/icgc/dkfzlsdf/analysis/G200/dinkelac/plots/")
#plotten
setwd("/icgc/dkfzlsdf/analysis/G200/dinkelac/plots/tras.2017.human.gtex.3x")

for(i in 1:length(tra.3x)){
print(i)
plotten.gtex.png(tra.3x[i],3)
}

#****

setwd("/icgc/dkfzlsdf/analysis/G200/dinkelac/plots/tras.2017.human.gtex.5x")

for(i in 1:length(tra.5x)){
print(i)
plotten.gtex.png(tra.5x[i],5)
}

setwd("/icgc/dkfzlsdf/analysis/G200/dinkelac/plots/tras.2017.human.gtex.10x")

for(i in 1:length(tra.10x)){
print(i)
plotten.gtex.png(tra.10x[i],10)
}

setwd("/icgc/dkfzlsdf/analysis/G200/dinkelac/plots/tras.2017.human.gtex.20x")

for(i in 1:length(tra.20x)){
print(i)
plotten.gtex.png(tra.20x[i],20)
}

setwd("/icgc/dkfzlsdf/analysis/G200/dinkelac/plots/tras.2017.human.gtex.3x.eight")

for(i in 1:length(tra.3x.eight)){
print(i)
plotten.gtex.png(tra.3x.eight[i],3)
}

setwd("/icgc/dkfzlsdf/analysis/G200/dinkelac/plots/tras.2017.human.gtex.5x.eight")

for(i in 1:length(tra.5x.eight)){
print(i)
plotten.gtex.png(tra.5x.eight[i],5)
}

setwd("/icgc/dkfzlsdf/analysis/G200/dinkelac/plots/tras.2017.human.gtex.10x.eight")

for(i in 1:length(tra.10x.eight)){
print(i)
plotten.gtex.png(tra.10x.eight[i],10)
}
```

```
setwd("/icgc/dkfzlsdf/analysis/G200/dinkelac/plots/tras.2017.human.gtex.20x.eight")

for(i in 1:length(tra.20x.eight)){
print(i)
plotten.gtex.png(tra.20x.eight[i],20)
}
#*******
#calculate TRA clustering
#violinplot
#jitterplot
boxplot(NUMS ~ GRP, data = ddf, lwd = 2, ylab = 'NUMS')
stripchart(NUMS ~ GRP, vertical = TRUE, data = ddf,
             method = "jitter", add = TRUE, pch = 16, col = 'blue')

#annotate with ensembl 81
#-------------------------
setwd("/home/dinkelac/data/mouse/tables/")
ensembl.human.81=read.csv(file="ensembl.human.81.txt",sep="\t")
ensembl.transcripts=as.character(ensembl.human.81[,2])
#233883, 216826 (unique)

doppelte.transcripte=which(duplicated(ensembl.transcripts)==T)
#17057

ensembl.symbols.all=as.character(ensembl.human.81[,7])
names(ensembl.symbols.all)=ensembl.transcripts
ensembl.symbols=ensembl.symbols.all[-doppelte.transcripte]
doppelte.transcripte.transcript.IDs=unique(ensembl.transcripts[doppelte.transcripte])
#14485

#sucht alle potentiellen symbole raus, die fuer die gleiche transcript
#ID doppelt belegt sind

ind.new=c("")
for (i in 1:length(doppelte.transcripte.transcript.IDs)){
transcript.id=doppelte.transcripte.transcript.IDs[i]
ind=which(names(ensembl.symbols.all)==transcript.id)
symbols=unique(ensembl.symbols.all[ind])
if(length(symbols)>1){
ind.new=c(ind.new,transcript.id)
}
}

#paste doppelte symbole pro transcript
ensembl.symbols.new=ensembl.symbols

for (i in 1:length(ind.new)){
ind=which(names(ensembl.symbols.all)==transcript.id)
symbols=unique(ensembl.symbols.all[ind])
new.symbol=paste(symbols,collapse="/")
print(new.symbol)
ensembl.symbols.new[transcript.id]=new.symbol
}

#fuehrt doppelte symbole in ensembl.symbols.new zurueck
#****
test_data=gtex_data[1:10,1:10]
gtex.transcripts=as.character(test_data[,1])
#100 transcripts
gtex.genes=as.character(test_data[,2])
#37 genes
gtex.chr=as.character(test_data[,3])
```

```
gtex.coord=as.character(test_data[,4])
gtex.data=test_data[,-c(1:4)]
gtex.tissue.names=names(gtex.data)
gtex.tissues=gtex.tissue.names
gtex.tissues=sub("\\.","-",gtex.tissues)
#4x

tissue.names=as.character(gtex.tissue[gtex.tissues])
names(test_data)[-c(1:4)]=tissue.names
pdata=test_data[,c(1:4)]
data.matrix=as.matrix(test_data[,-c(1:4)])
x=apply(data.matrix, 2, as.numeric)
final.matrix=cbind(pdata,x)
rownames(final.matrix)=final.matrix[,1]

#1.1 librarys
#------------
library(affy)
library(vsn)

#sessionInfo()
# AnnotationDbi_1.24.0s

#1.2 read in RNA SEQ data
#------------------------
setwd("/home/dinkelac/data/GTEX/")
#dkfzlsdf/G200/
#ersten 100 Zeilen
#gtex_data=read.table("GTEx_Analysis_V4_RNA-seq_Flux1.6_transcript_rpkm.txt",
#nrows=100,header=T)

gtex_data_orig=read.table("GTEx_Analysis_V4_RNA-seq_Flux1.6_transcript_rpkm.txt",
header=T)
setwd("/icgc/dkfzlsdf/analysis/G200/dinkelac/sessions/rda/")
save.image(file="gtex.rawdata.rda")
setwd("/home/dinkelac/data/GTEX/")

#Daten im Verzeichnis auf der Konsole anschauen
#head -n 10 GTEx_Analysis_V4_RNA-seq_Flux1.6_transcript_rpkm.txt | cut -f 1-10
gtex_data=gtex_data_orig[,-c(1:4)]
#reading of pdata

setwd("/home/dinkelac/data/GTEX/")
sample.pdata=read.csv(file="GTEx_Data_V4_Annotations_SampleAttributesDS.txt",sep="\t")
patient.pdata=read.csv(file="GTEx_Data_V4_Annotations_SubjectPhenotypes_DS.txt",sep="\t")
patient.id=as.character(patient.pdata[,1])
#GTEX-U3ZN, 214 Patienten
gender=as.character(patient.pdata[,2])
names(gender)=patient.id
#138 (1), 76 (2)

age=as.character(patient.pdata[,3])
names(age)=patient.id
a=strsplit(age,split="-")
alter=vector(len=214)

for(i in 1:214){
alter[i]=a[[i]][1]
}

patient.age=as.numeric(alter)
names(patient.age)=patient.id
```

```
hist(patient.age,col="blue",main="Age of 214 Patients of GTEX Data")
abline(h=c(20,40,60,80),lty=3)

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.GTEX.patient.age.eps")

hist(as.numeric(gender),col=c("blue","green","blue"),
main="Gender of 214 Patients of GTEX Data",4,xlab="gender",ylim=c(0,150))
abline(h=c(50,100,150),lty=3)
legend("topright",c("1-male","2-female"))

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.GTEX.patient.gender.eps")

ind.men=which(gender=="1")
ind.women=which(gender=="2")

hist(patient.age,ylim=c(0,80))$counts
hist(patient.age[ind.men],ylim=c(0,80))$counts
hist(patient.age[ind.women],ylim=c(0,80))$counts

x=c(14,9,0,12,4,0,24,18,0,55,22,0,32,21,0,1,1,0)
names(x)=c(20,NA,NA,30,NA,NA,40,NA,NA,50,NA,NA,60,NA,NA,70,NA,NA)

barplot(x,col=c("blue","red","white"),ylim=c(0,80),
main="Age of 214 Patients in GTEX Data per Sex",ylab="Frequency",xlab="Age")
abline(h=c(10,20,30,40,50,60,70,80),lty=3)
legend("topright",c("male","female"),col=c("blue","red"),pch=1)

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.GTEX.patient.age.per.gender.eps")

todesursache=as.character(patient.pdata[,4])
names(todesursache)=patient.id
hist(as.numeric(todesursache),col=c("blue","green","white","yellow","white","orange",
"white","red"),main="Cause of Death of 214 Patients of GTEX Data",10,xlab="gender",ylim=c(0,150))
abline(h=c(50,100,150),lty=3)
legend("topright",c("0- ventilator","1- fast violent (<10 min)","2- fast unexpected (<1 hr)",
"3- ill unexpected (<24hrs)","3- ill expected (>24 hrs)"),pch=1)

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.GTEX.patient.death.eps")

#tissue.samples

x=substr(names(tissue.samples[which(tissue.samples=="Adipose - Subcutaneous")]),0,9)
#128 adipose tissue ist von 128 versch. Patienten
#190 whole blood ist von 190 versch. Patienten

sample.patient=substr(names(tissue.samples),0,9)
pat=unique(sample.patient)
sample.sum=vector(len=length(pat))
names(sample.sum)=pat

for(i in 1:length(pat)){
sample.sum[i]=sum(sample.patient==pat[i])
}

hist(sample.sum,main="Number of Tissues taken per Patient in GTEX Data",ylab="Number of Tissue",
xlab="Number of Patients",100,ylim=c(0,20),col="blue")
abline(h=c(5,10,15,20),lty=3)
```

```
setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.GTEX.number.of.tissues.per.patient.eps")

#tisse number per gender
male_female=gender[names(sample.sum)]

x=which(male_female=="1")
y=which(male_female=="2")

werte.m=m$counts
names(werte.m)=c(2:29)

werte.f=f$counts
names(werte.f)=c(2:27)

x=c(3,1,0,2,2,0,5,1,0,2,2,0,4,1,0,5,5,0,6,0,0,9,3,0,8,5,0,7,2,0,7,3,0,8,2,0,13,6,0,6,7,0,7,5,0,
10,3,0,6,7,0,4,6,0,7,2,0,6,2,0,3,1,0,1,1,0,2,1,0,1,2,0,0,1,0,0,3,0,1,0,0,2,0,0)
names(x)=sort(rep(c(2:29),3))
barplot(x,col=c("blue","red","white"),ylim=c(0,20),
main="Number of tissues taken per patient in GTEX Data\n per gender",
ylab="Number of Tissues",xlab="Number of Patients")
abline(h=c(5,10,15,20),lty=3)
legend("topright",c("male","female"),col=c("blue","red"),pch=1)

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.GTEX.number.of.tissues.per.patient.per.gender.eps")

#Durchschnittsalter der Maenner, Frauen

#2. Quality control
#-------------------
#2.1 Single chip control
#------------------------

GTEX_data=gtex_data[,-c(1:4)]

# #2.2 pheno data
# #--------------
# #adjust filenames and read in pheno data
#
# #mouse gngnf1
# #------------
# file.names.mouse=rownames(pData(phenoData(data.mouse)))
# colnames(exprs(data.mouse))=file.names.mouse
#
# #get tissue names
# #extract tissue names
# tissue.names.mouse=substr(file.names.mouse,3,nchar(file.names.mouse))
# letters=substr(tissue.names.mouse,1,1)
# tissues=substr(tissue.names.mouse,2,nchar(tissue.names.mouse))
#
# x=paste(letters,tissues,sep="_",colapse=NULL)
# new.file.names=substr(x,1,nchar(x)-1)
#
# #change tissue names
# rownames(pData(phenoData(data.mouse)))=new.file.names
# rownames(pData(protocolData(data.mouse)))=new.file.names
# colnames(exprs(data.mouse))=new.file.names

#3 Normalization
#---------------
#For Normalization vsnrma normalization was taken, Huber et al. 2002
```

```
#citation("vsn")
library(vsn)

gtex.vsnrma<-vsnrma(as.matrix(GTEX_data))

setwd("/home/dinkelac/data/mouse/sessions/rda")
save.image(file="gtex.normalized.rda")

#4 Quality Plots
#---------------
#4.1 meandSdPlot
#---------------
#mouse gngnf1
#------------
meanSdPlot(mouse.vsnrma)
title(main="Gngnf1- mouse meanSdPlot\n after vsnrma normalization")
title(sub="row standard deviations versus row means")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="meanSdPlot.mouse.vsnrma.normalized.eps")

#mouse 4301
#----------
meanSdPlot(mouse4301.vsnrma)
title(main="Mouse 4302- meanSdPlot\n after vsnrma normalization")
title(sub="row standard deviations versus row means")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="meanSdPlot.mouse4302.vsnrma.normalized.eps")

#human novartis
#--------------
meanSdPlot(human.vsnrma)
title(main="Human hgu133A - meanSdPlot\n after vsnrma normalization")
title(sub="row standard deviations versus row means")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="meanSdPlot.human.vsnrma.normalized.eps")

#human roth
#----------
meanSdPlot(human.roth.vsnrma)
title(main="Human hgu133A Plus2 - meanSdPlot\n after vsnrma normalization")
title(sub="row standard deviations versus row means")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="meanSdPlot.human.roth.vsnrma.normalized.eps")

#4.2 Boxplot
#-----------
#rawdata
#-------
#mouse gngnf1
x11(w=10,h=7)
#90 Grad beschriften
par(las=2)
#Rand zum Beschriften unten, li, oben, re
mmi=c(1.7,0.7,1.0477939,0.5366749)
par(mai=mmi)
boxplot(final.matrix[,-c(1:4)], col=rainbow(150),cex.axis=0.5,
main="Gngnf1- Gene expression in 61 different tissues of the mouse\n
before normalization")
```

```
setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.mouse.rawdata.eps")

#mouse 4301
x11(w=12,h=7)
#90 Grad beschriften
par(las=2)
#Rand zum Beschriften unten, li, oben, re
mmi=c(1.9,0.7,1.0477939,0.5366749)
par(mai=mmi)

boxplot(data.mouse4301, col=rainbow(150),cex.axis=0.5,
main="Mouse 4302- Gene expression in 91 different tissues of the mouse\n
before normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.mouse4302.rawdata.eps")

#human novartis
x11(w=12,h=7)
#90 Grad beschriften
par(las=2)
#Rand zum Beschriften unten, li, oben, re
mmi=c(1.9,0.7,1.0477939,0.5366749)
par(mai=mmi)

boxplot(data.human, col=rainbow(150),cex.axis=0.5,
main="Human hgu133A- Gene expression in 71 different human tissues\n
before normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.human.rawdata.eps")

#human roth
x11(w=12,h=7)
#90 Grad beschriften
par(las=2)
#Rand zum Beschriften unten, li, oben, re
mmi=c(1.9,0.7,1.0477939,0.5366749)
par(mai=mmi)

boxplot(data.human.roth, col=rainbow(150),cex.axis=0.5,
main="Human hgu133_Plus2- Gene expression in 65 different human tissues\n
before normalization (Roth)")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.human.roth.rawdata.eps")

#normalized data
#---------------
#mouse gngnf1

#vsnrma
#-------
x11(w=10,h=7)
par(las=2)
mmi=c(1.7,0.7,1.0477939,0.5366749)
par(mai=mmi)

boxplot(exprs(mouse.vsnrma), col=rainbow(150),cex.axis=0.5,
main="Gngnf1- Gene expression in 61 different tissues of the mouse\n
```

```
after vsnrma normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.gngnf1.vsnrma.normalized.eps")

#mouse 4302
x11(w=12,h=7)
par(las=2)
mmi=c(1.9,0.7,1.0477939,0.5366749)
par(mai=mmi)

boxplot(exprs(mouse4301.vsnrma), col=rainbow(150),cex.axis=0.5,
main="Mouse 4302- Gene expression in 91 different tissues of the mouse\n
after vsnrma normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.4302.vsnrma.normalized.eps")

#human novartis
x11(w=12,h=7)
par(las=2)
mmi=c(1.9,0.7,1.0477939,0.5366749)
par(mai=mmi)

boxplot(exprs(human.vsnrma), col=rainbow(150),cex.axis=0.5,
main="Human hgu133A - Gene expression in 71 different human tissues\n
after vsnrma normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.human.vsnrma.normalized.eps")

#human roth
x11(w=12,h=7)
par(las=2)
mmi=c(1.9,0.7,1.0477939,0.5366749)
par(mai=mmi)

boxplot(exprs(human.roth.vsnrma), col=rainbow(150),cex.axis=0.5,
main="Human hgu133_Plus2 - Gene expression in 65 different human tissues\n
after vsnrma normalization (Roth)")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.human.roth.vsnrma.normalized.eps")

#4.3 Histogram
#-------------
#rawdata
#-------
#mouse gngnf1
x11(w=10,h=7)
hist(data.mouse, col=rainbow(150),
main="Gene expression in 61 different tissues of the mouse (gngnf1)\n
before normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.mouse.rawdata.eps")

#mouse 4302
x11(w=10,h=7)

hist(data.mouse4301, col=rainbow(150),
main="Gene expression in 91 different tissues of the mouse (4302)\n
```

```
before normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.mouse4302.rawdata.eps")

#human novartis
x11(w=10,h=7)

hist(data.human, col=rainbow(150),
main="Gene expression in 71 different human tissues\n
before normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.human.rawdata.eps")

#human roth
x11(w=10,h=7)

hist(data.human.roth[,1:353],col=rainbow(353),
main="Gene expression in 65 different human tissues (Roth data)\n
before normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.human.roth.rawdata.eps")

#normalized
#----------
#vsnrma
#mouse gngnf1

eset = exprs(mouse.vsnrma)

plot(density(eset[,1]), type="n", ylim=c(0,1),
main="Gene expression in 61 different tissues of the mouse (gngnf1)\n
after vsnrma normalization")
for (i in 1:ncol(eset)){
  lines(density(eset[,i]), col=rainbow(150)[i])
}

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.gngnf1.vsnrma.normalized.eps")

#mouse 4302
eset = exprs(mouse4301.vsnrma)

plot(density(eset[,1]), type="n", ylim=c(0,1),
main="Gene expression in 91 different tissues of the mouse (4302)\n
after vsnrma normalization")
for (i in 1:ncol(eset)){
  lines(density(eset[,i]), col=rainbow(180)[i])
}

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.4302.vsnrma.normalized.eps")
#human novartis
eset = exprs(human.vsnrma)

plot(density(eset[,1]), type="n", ylim=c(0,1),
main="Gene expression in 79 different human tissues\n
after vsnrma normalization")
for (i in 1:ncol(eset)){
  lines(density(eset[,i]), col=rainbow(180)[i])
```

```
}

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.human.vsnrma.normalized.eps")

#human roth
eset = exprs(human.roth.vsnrma)

plot(density(eset[,1]), type="n", ylim=c(0,2),
main="Gene expression in 65 different human tissues (Roth data)\n
after vsnrma normalization")
for (i in 1:ncol(eset)){
  lines(density(eset[,i]), col=rainbow(353)[i])
}

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="hist.human.roth.vsnrma.normalized.eps")

#4.4 RNA degeneration plot
#-------------------------
#mouse gngnf1
rnadeg.raw = AffyRNAdeg(data.mouse)

plotAffyRNAdeg(rnadeg.raw, col=rainbow(150))
title(sub="mouse gngnf1 rawdata")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="rnadeg.gngnf1.rawdata.eps")

plotAffyRNAdeg(rnadeg.raw, col=rainbow(150), transform="shift.only")
title(sub="mouse gngnf1 rawdata")

dev.copy2eps(file="rnadeg1.gngnf1.rawdata.eps")

#mouse 4302
rnadeg.raw = AffyRNAdeg(data.mouse4301)

plotAffyRNAdeg(rnadeg.raw, col=rainbow(180))
title(sub="mouse 4302 rawdata")

dev.copy2eps(file="rnadeg.4302.rawdata.eps")

plotAffyRNAdeg(rnadeg.raw, col=rainbow(180), transform="shift.only")
title(sub="mouse 4302 rawdata")

dev.copy2eps(file="rnadeg1.4302.rawdata.eps")

#human novartis
rnadeg.raw = AffyRNAdeg(data.human)

plotAffyRNAdeg(rnadeg.raw, col=rainbow(180))
title(sub="human rawdata")

dev.copy2eps(file="rnadeg.human.rawdata.eps")

plotAffyRNAdeg(rnadeg.raw, col=rainbow(180), transform="shift.only")
title(sub="human rawdata")

dev.copy2eps(file="rnadeg1.human.rawdata.eps")

#human roth
rnadeg.raw = AffyRNAdeg(data.human.roth)
```

```
plotAffyRNAdeg(rnadeg.raw, col=rainbow(353))

title(sub="human rawdata (Roth data)")

dev.copy2eps(file="rnadeg.human.roth.rawdata.eps")

plotAffyRNAdeg(rnadeg.raw, col=rainbow(353), transform="shift.only")
title(sub="human rawdata (Roth data)")

dev.copy2eps(file="rnadeg1.human.roth.rawdata.eps")

#5. Calculate the mean over double messurements
#-----------------------------------------------
#mouse gngnf1
#------------
tissue.names=colnames(exprs(mouse.vsnrma))

#all tissues exist twice, only cortex and cerebralcortex once,
#is this the same?

which(tissue.names=="B_cerebralcortex")
which(tissue.names=="A_cortex")

#22,111

plot(exprs(mouse.vsnrma)[,c(22,111)], pch=".")
title(main="Gene expression values of the mouse gngnf1 dataset\n
cortex versus cerbralcortex")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="gngnf1.cortex.cerebralcortex.eps")

colnames(exprs(mouse.vsnrma))[22]="B_cortex"

#mean.vsnrma
tissues=substr(tissue.names,3,nchar(tissue.names))
tissues.unique=unique(tissues)

mean.mouse.vsnrma = matrix(NA, nr=nrow(exprs(mouse.vsnrma)), nc=61)
rownames(mean.mouse.vsnrma) = featureNames(mouse.vsnrma)
colnames(mean.mouse.vsnrma) = tissues.unique

for (t in tissues.unique) {
  index.tissue = which(tissues == t) # alternativ: which(tissues.all %in% t)
  if (length(index.tissue) > 1) {
    mean.mouse.vsnrma[, t] = apply(exprs(mouse.vsnrma)[,index.tissue], 1, mean, na.rm=T)
  }
}

#boxplot of mean.vsnrma
x11(w=10,h=7)
par(las=2)
mmi=c(1.7,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
boxplot(data.frame(mean.mouse.vsnrma), col=rainbow(100), cex.axis=0.5)
title(main="Gngnf1 - distribution of averaged expressions\n
in 61 mouse tissues after vsnrma normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.gngnf1.mean.vsnrma.eps")
```

```
#mouse 4301
colnames(exprs(mouse4301.vsnrma))=as.character(pdata4301[,2])

tissues=colnames(exprs(mouse4301.vsnrma))
tissue.names=substr(tissues,1,nchar(tissues)-2)

tissues.unique=unique(tissue.names)

mean4301.vsnrma = matrix(NA, nr=nrow(exprs(mouse4301.vsnrma)), nc=91)
rownames(mean4301.vsnrma) = featureNames(mouse4301.vsnrma)
colnames(mean4301.vsnrma) = tissues.unique

for (t in tissues.unique) {
  index.tissue = which(tissue.names == t) # alternativ: which(tissues.all %in% t)
  if (length(index.tissue) > 1) {
    mean4301.vsnrma[, t] = apply(exprs(mouse4301.vsnrma)[,index.tissue], 1,
    mean, na.rm=T)
  }
}

#boxplot mouse 4301 mean vsnrma
x11(w=10,h=7)
par(las=2)
mmi=c(2.3,0.7,1.0477939,0.5366749)
par(mai=mmi)
boxplot(data.frame(mean4301.vsnrma), col=rainbow(100), cex.axis=0.6)
title(main="Mouse 4302 - distribution of averaged expressions\n
in 91 mouse tissues")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.4302.mean.vsnrma.eps")

#human roth
tissue.names=colnames(exprs(data.human.roth))
tissues=substr(1,length)

colnames(exprs(human.roth.vsnrma))=tissues.human.roth

#scatter plots
#--------------
#mouse gngnf1

# x11(w=9,h=5)
# par(mfrow=c(1,2))
tiss.index=which(tissue.names.mouse=="trachea")

plot(exprs(mouse.vsnrma)[,c(tiss.index[1],tiss.index[2])],pch=".")
abline(0,1,col="red")

title(main="Scatterplot")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="scatterplot.gngnf1.Atrachea.eps")

#mouse 4302
#adipose_white_B, amygdala_A, iris_B, spinal_cord_B
#6,9,72,158

tiss.index=which(tissues.mouse4302=="spinal_cord")

plot(exprs(mouse4301.vsnrma)[,c(tiss.index[1],tiss.index[2])],pch=".")
abline(0,1,col="red")
```

```
title(main="Scatterplot")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="scatterplot.mouse4302.spinal_cord.eps")

#human roth
#bronchus_4,corpus_callosum,kidney_cortex_2,midbrain_5,9,10
#ovary_8,oral_mucosa_1,oral_mucosa_2,ventral_tegmental_area_4
#22,84,127,143,145,146,197,313

tiss.index=which(tissue.names.human.roth=="ventral_tegmental_area")

x11(w=9,h=10)
par(mfrow=c(3,3))

plot(exprs(human.roth.vsnrma)[,c(tiss.index[6],tiss.index[1])],pch=".")
abline(0,1,col="red")

title(main="Scatterplot")

plot(exprs(human.roth.vsnrma)[,c(tiss.index[6],tiss.index[2])],pch=".")
abline(0,1,col="red")

title(main="human Roth data")

plot(exprs(human.roth.vsnrma)[,c(tiss.index[6],tiss.index[3])],pch=".")
abline(0,1,col="red")

plot(exprs(human.roth.vsnrma)[,c(tiss.index[6],tiss.index[4])],pch=".")
abline(0,1,col="red")

plot(exprs(human.roth.vsnrma)[,c(tiss.index[6],tiss.index[5])],pch=".")
abline(0,1,col="red")

plot(exprs(human.roth.vsnrma)[,c(tiss.index[6],tiss.index[7])],pch=".")
abline(0,1,col="red")

plot(exprs(human.roth.vsnrma)[,c(tiss.index[6],tiss.index[8])],pch=".")
abline(0,1,col="red")

plot(exprs(human.roth.vsnrma)[,c(tiss.index[1],tiss.index[2])],pch=".")
abline(0,1,col="red")

plot(exprs(human.roth.vsnrma)[,c(tiss.index[1],tiss.index[3])],pch=".")
abline(0,1,col="red")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="scatterplot.human.roth.ventral_tegmental_area_9.eps")

#midbrain 5,9,10, ovary 8 raus
#143,145,146,197,313,334 raus

human.roth.index.out=c(143,145,146,197,313,334)
human.roth.vsnrma.all=human.roth.vsnrma
human.roth.vsnrma=human.roth.vsnrma.all[,-human.roth.index.out]

#347 chips, left over, aber 353 filenames
human.roth.tissues

#nach chip rausnehmen, tissues noch einmal neu festlegen
#-----------------------------------------------------
```

```
human.roth.tissues=colnames(exprs(human.roth.vsnrma))

a=human.roth.tissues
b=sub("10","0",a)
c=substr(b,1,nchar(b)-2)
d=unique(c)

roth.tissues.all=(c)
roth.tissues=(d)

tissue.names=sub("10","0",human.roth.tissues)
tissue.names1=substr(tissue.names,1,nchar(tissue.names)-2)
tissues.unique=unique(tissue.names1)

#5.1 mean.vsnrma
#---------------
mean.human.roth.vsnrma = matrix(NA, nr=nrow(exprs(human.roth.vsnrma)), nc=65)
rownames(mean.human.roth.vsnrma) = featureNames(human.roth.vsnrma)
colnames(mean.human.roth.vsnrma) = tissues.unique

for (t in tissues.unique) {
  index.tissue = which(tissue.names1 == t) # alternativ: which(tissues.all %in% t)
  if (length(index.tissue) > 1) {
    mean.human.roth.vsnrma[, t] = apply(exprs(human.roth.vsnrma)[,index.tissue], 1,
    mean, na.rm=T)
  }
}

setwd("/home/dinkelac/data/mouse/sessions/rda")
save.image(file="mean.vsnrma.human.roth.rda")

#boxplot of mean.vsnrma
#----------------------
x11(w=10,h=7)
par(las=2)
mmi=c(2.3,0.7,1.0477939,0.5366749)
par(mai=mmi)
boxplot(data.frame(mean.human.roth.vsnrma), col=rainbow(100), cex.axis=0.6)
title(main="Human - distribution of averaged expressions\n
in 65 human tissues after vsnrma normalization (Roth data)")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.human.roth.mean.vsnrma.eps")

#human novartis
#--------------
#26,74,83,84,92,95,126
#trigeminal ganglion, adrenal cortex, uterus corpus (84), testis germ cell,
#testis leydig cell, cardiac myocytes (126)

pheno.data.human.novartis[26,]
tissue.names.human.novartis[26]
tissues.human.novartis=substr(tissue.names.human.novartis,0,
nchar(tissue.names.human.novartis)-2)

#umbenennen
tiss.index=which(tissues.human.novartis=="TestisLeydigCell")

plot(exprs(human.vsnrma)[,c(tiss.index[1],tiss.index[2])],pch=".")
abline(0,1,col="red")

title(main="Scatterplot")
```

```
setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="scatterplot.human.novartis.CardiacMyocytes.eps")

human.novartis.index.out=c(84,126)
colnames(human.vsnrma)=tissue.names.human.novartis
human.novartis.vsnrma=human.vsnrma[,-human.novartis.index.out]
tissues.human.novartis.unique=unique(tissues.human.novartis)
tissues1=substr(colnames(human.novartis.vsnrma),1,
nchar(colnames(human.novartis.vsnrma))-2)

mean.human.vsnrma = matrix(NA, nr=nrow(exprs(human.novartis.vsnrma)), nc=79)
rownames(mean.human.vsnrma) = featureNames(human.novartis.vsnrma)
colnames(mean.human.vsnrma) = tissues.human.novartis.unique

for (t in tissues.human.novartis.unique) {
  index.tissue = which(tissues1 == t) # alternativ: which(tissues.all %in% t)
  if (length(index.tissue) > 1) {
    mean.human.vsnrma[, t] = apply(exprs(human.novartis.vsnrma)[,index.tissue], 1,
    mean, na.rm=T)
  }
  else {
  print(t)
    mean.human.vsnrma[, t] = (exprs(human.novartis.vsnrma)[,index.tissue])
  }
}

#boxplot of mean.human.vsnrma
x11(w=10,h=7)
par(las=2)
mmi=c(2.3,0.7,1.0477939,0.5366749)
par(mai=mmi)
boxplot(data.frame(mean.human.vsnrma), col=rainbow(100), cex.axis=0.6)
title(main="Human - distribution of averaged expressions\n
in 79 human tissues after vsnrma normalization")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="boxplot.human.mean.vsnrma.eps")

setwd("/home/dinkelac/data/mouse/sessions/rda")
save.image(file="mean.data.rda")

#clustering of GTEX data
#------------------------
mkdir observed_ntuples_human_gtex_5_tiss_3x

#5 tissues, 8 tissues, jeweils 3x, 5x, 10x, 20x

#tra listen erstellen
#ensgeneID, Chrom, Startsite
#hash aller GTEX gene
#skript.pl

setwd("/home/dinkelac/data/mouse/tables/")

gtex.5.3x=read.csv("tra.2017.human.gtex.3x.table.csv",sep="\t")
gtex.5.5x=read.csv("tra.2017.human.gtex.5x.table.csv",sep="\t")
gtex.5.10x=read.csv("tra.2017.human.gtex.10x.table.csv",sep="\t")
gtex.5.20x=read.csv("tra.2017.human.gtex.20x.table.csv",sep="\t")

gtex.8.3x=read.csv("tra.2017.human.gtex.3x.table.eight.csv",sep="\t")
gtex.8.5x=read.csv("tra.2017.human.gtex.5x.table.eight.csv",sep="\t")
```

```
gtex.8.10x=read.csv("tra.2017.human.gtex.10x.table.eight.csv",sep="\t")
gtex.8.20x=read.csv("tra.2017.human.gtex.20x.table.eight.csv",sep="\t")

gtex.all=read.csv("tra.2017.human.gtex.3x.table.csv",sep="\t")

#doppelte raus !!!!
#auf genebene !!!!!!!!!

doppelte.5.3x=which(duplicated(as.character(gtex.5.3x[,2])))
doppelte.5.5x=which(duplicated(as.character(gtex.5.5x[,2])))
doppelte.5.10x=which(duplicated(as.character(gtex.5.10x[,2])))
doppelte.5.20x=which(duplicated(as.character(gtex.5.20x[,2])))

doppelte.8.3x=which(duplicated(as.character(gtex.8.3x[,2])))
doppelte.8.5x=which(duplicated(as.character(gtex.8.5x[,2])))
doppelte.8.10x=which(duplicated(as.character(gtex.8.10x[,2])))
doppelte.8.20x=which(duplicated(as.character(gtex.8.20x[,2])))

tras.gtex.5.3x=gtex.5.3x[-doppelte.5.3x,c(2,6,7)]
tras.gtex.5.5x=gtex.5.5x[-doppelte.5.5x,c(2,6,7)]
tras.gtex.5.10x=gtex.5.10x[-doppelte.5.10x,c(2,6,7)]
tras.gtex.5.20x=gtex.5.20x[-doppelte.5.20x,c(2,6,7)]

tras.gtex.8.3x=gtex.8.3x[-doppelte.8.3x,c(2,6,7)]
tras.gtex.8.5x=gtex.8.5x[-doppelte.8.5x,c(2,6,7)]
tras.gtex.8.10x=gtex.8.10x[-doppelte.8.10x,c(2,6,7)]
tras.gtex.8.20x=gtex.8.20x[-doppelte.8.20x,c(2,6,7)]

setwd("/home/dinkelac/data/mouse/tables/")

write.table(tras.gtex.5.3x,file="tra.2017.human.gtex.5.3x.genes.txt",
row.names=F,sep="\t",col.names=F)
write.table(tras.gtex.5.5x,file="tra.2017.human.gtex.5.5x.genes.txt",
row.names=F,sep="\t",col.names=F)
write.table(tras.gtex.5.10x,file="tra.2017.human.gtex.5.10x.genes.txt",
row.names=F,sep="\t",col.names=F)
write.table(tras.gtex.5.20x,file="tra.2017.human.gtex.5.20x.genes.txt",
row.names=F,sep="\t",col.names=F)

write.table(tras.gtex.8.3x,file="tra.2017.human.gtex.8.3x.genes.txt",
row.names=F,sep="\t",col.names=F)
write.table(tras.gtex.8.5x,file="tra.2017.human.gtex.8.5x.genes.txt",
row.names=F,sep="\t",col.names=F)
write.table(tras.gtex.8.10x,file="tra.2017.human.gtex.8.10x.genes.txt",
row.names=F,sep="\t",col.names=F)
write.table(tras.gtex.8.20x,file="tra.2017.human.gtex.8.20x.genes.txt",
row.names=F,sep="\t",col.names=F)

#load gtex.data.rda (anderer Name)
#---------------------------------
setwd("/icgc/dkfzlsdf/analysis/G200/dinkelac/sessions/rda/")
load("gtex.25.01.2017.rda")

gtex.transcripts=row.names(gtex.data.sum)
gtex.transcript.names=unique(substr(gtex.transcripts,0,nchar(gtex.transcripts)-2))
gtex.gene.names=vector(len=length(gtex.transcript.names))
gtex.chrom=vector(len=length(gtex.transcript.names))
gtex.startsite=vector(len=length(gtex.transcript.names))

for(i in 1:length(gtex.transcript.names)){
gene=ensembl.gene[gtex.transcript.names[i]]
chr=ensembl.chrom[gtex.transcript.names[i]]
```

```
start=ensembl.start[gtex.transcript.names[i]]

if(is.na(gene)){
gtex.gene.names[i]=NA
}else{
gtex.gene.names[i]=gene
}

if(is.na(chr)){
gtex.chrom[i]=NA
}else{
gtex.chrom[i]=chr
}

if(is.na(start)){
gtex.startsite[i]=NA
}else{
gtex.startsite[i]=start
}
print(i)
}

gtex.all.genes.annotated=cbind(gtex.gene.names,gtex.chrom,gtex.startsite)
doppelte=which(duplicated(gtex.gene.names)==T)
gtex.chrhash=gtex.all.genes.annotated[-doppelte,]

setwd("/home/dinkelac/data/mouse/tables/")
write.table(gtex.chrhash,file="gtex.chrhash.2017.txt",row.names=F,sep="\t",col.names=F)

#in the folders
perl observed_ntuples_mouse.pl tra.2014.mouse.3x.genes.txt mouse.chrhash.genes.txt >
results.txt
```

1.1.3 Calculate Tissue Restricted Antigens (TRAs)

This script is calculating tissue restricted antigens (TRAs) in Microarray data.

```
#calculate_tras_mouse_gngnf1.R
#------------------------------
#This script uses the R session mean.data.rda and calculates tissue restricted antigens
#(TRAs) in the four different datasets, mouse sue data, human novartis data, human roth
#data and mouse 4301 data since the rat data was not very good to use, since only cns
#tissues were present in the microarray data for the rat TRAs homologues from the mouse
#are calculated via a homology mapping from the NCBI.

#version 2014
#------------
#tra.2014.mouse.3x, 5x, 10x, 20x (su dataset mouse)
#tra.2014.human.3x, 5x, 10x, 20x (su dataset human)
#tra.2014.mouse4301.3x, 5x, 10x, 20x (lattin dataset)
#tra.2014.human.roth.3x, 5x, 10x, 20x (roth dataset)

#readin RNASeq data
#------------------
#GTEX data 2014

setwd("/home/dinkelac/data/RNAseq/")
GTEx.rpkm.data=read.csv(file="GTEx_Analysis_2014-01-17_RNA-seq_Flux1.6_transcript_rpkm.txt",
```

```
sep="\t")

GTEx.pdata=read.csv(file="RNA_Seq_pdata.txt",sep="\t")
rownames(GTEx.pdata)=GTEx.pdata[,1]

transcript.IDs=as.character(GTEx.rpkm.data[,1])
#194844
unique.transcript.IDs=unique(transcript.IDs)
#194821

doppelte.transcripte=which(duplicated(transcript.IDs))
#GTEx.rpkm.data.unique=GTEx.rpkm.data[-doppelte.transcripte,]
#194821
#GTEx.rpkm.data=GTEx.rpkm.data.unique
rownames(GTEx.rpkm.data)=GTEx.rpkm.data[,1]

#library
#-------
library(affy)
library(gngnf1musammenstcdf, lib.loc="/home/dinkelac/R-libs")
library(hgu133ahsenstcdf, lib.loc="/home/dinkelac/R-libs")
library(mouse4302mmenstcdf, lib.loc="/home/dinkelac/R-libs")
library(hgu133plus2hsenstcdf, lib.loc="/home/dinkelac/R-libs/")

#functions
#---------
#source("/home/dinkelac/R-functions/count.over.median.mouse.gngnf1.R")

#load session
#------------
setwd("/home/dinkelac/data/mouse/sessions/rda")
load("mean.data.rda")

#mean.mouse.vsnrma, mean.human.vsnrma, mean.human.roth.vsnrma, mean4301.vsnrma
#tissues available in each dataset

tissues.mouse=colnames(mean.mouse.vsnrma)
#61
tissues.human=colnames(mean.human.vsnrma)
#79
tissues.mouse4301=colnames(mean4301.vsnrma)
#91
tissues.human.roth=colnames(mean.human.roth.vsnrma)
#65

#tissue grouping (colnames(mean.vsnrma))
#---------------------------------------
#tissue grouping mouse novartis
#------------------------------
cns.mouse.names=c("amygdala","frontalcortex","preoptic","cerebellum","cortex",
"dorsalstriatum","hippocampus","hypothalamus","olfactorybulb","spinalcordlower",
"spinalcordupper","substantianigra","pituitary")
epidermis.mouse.names=c("digits","epidermis","snoutepidermis","tongueepidermis")
immune.cells.mouse.names=c("cd4+Tcell","cd8+Tcell","b220+bcell")
intestine.mouse.names=c("largeintestine","smallintestine")
ovary.mouse.names=c("ovary","oocyte")
pns.mouse.names=c("trigeminal","dorsalrootganglion","medialolfactoryepithelium(MOE)",
"vomeralnasalorgan(VMO)")

#excluded mouse novartis
#-----------------------
embryos.mouse.names=c("embryoday6.5","embryoday7.5","embryoday8.5","embryoday9.5",
```

```
"embryoday10.5")

#tissue grouping human novartis
#------------------------------
adrenalgland.human.names=c("AdrenalCortex","adrenalgland")
cns.human.names=c("WholeBrain","CerebellumPeduncles","CingulateCortex","globuspallidus",
"MedullaOblongata","OccipitalLobe","OlfactoryBulb","ParietalLobe","Pituitary","Pons",
"PrefrontalCortex","subthalamicnucleus","TemporalLobe","Amygdala","caudatenucleus",
"cerebellum","Hypothalamus","spinalcord","Thalamus")
epidermis.human.names=c("TONGUE","skin")
lymphoid.structure.human.names=c("Appendix","Tonsil","lymphnode")
muscle.human.names=c("SkeletalMuscle","SmoothMuscle")
pancreas.human.names=c("Pancreas","PancreaticIslets")
pns.human.names=c("ciliaryganglion","SuperiorCervicalGanglion","TrigeminalGanglion","DRG")
testis.human.names=c("testis","TestisGermCell","TestisInterstitial","TestisLeydigCell",
"TestisSeminiferousTubule")
uterus.human.names=c("Uterus","UterusCorpus")

#excluded human novartis
#-----------------------
cancer.cells.human.names=c("lymphomaburkittsRaji","lymphomaburkittsDaudi",
"leukemiapromyelocytic(hl60)","leukemialymphoblastic(molt4)",
"leukemiachronicmyelogenous(k562)","ColorectalAdenocarcinoma")
cell.lines.human.names=c("bronchialepithelialcells","CardiacMyocytes")
embryos.human.names=c("fetalbrain","fetalliver","fetallung","fetalThyroid")
immune.cells.human.names=c("PB-CD56+NKCells","PB-CD8+Tcells","PB-CD4+Tcells",
"PB-CD14+Monocytes","PB-CD19+Bcells","PB-BDCA4+Dentritic_Cells","721_B_lymphoblasts",
"BM-CD33+Myeloid","BM-CD34+","BM-CD71+EarlyErythroid","BM-CD105+Endothelial")

#tissue grouping mouse 4301
#--------------------------
adipose.tissue.mouse4301.names=c("adipose_white","adipose_brown")
cns.mouse4301.names=c("amygdala","cerebellum","cerebral_cortex","dorsal_striatum",
"hippocampus","hypothalamus","pituitary","spinal_cord","olfactory_bulb",
"cerebral_cortex_prefrontal","nucleus_accumbens","microglia")
eyes.mouse4301.names=c("retina","iris","lacrimal_gland","lens","eyecup","ciliary_bodies",
"retinal_pigment_epithelium","cornea")
intestine.mouse4301.names=c("intestine_small","intestine_large")
mammary.gland.mouse4301.names=c("mammary_gland_non-lactating","mammary_gland__lact")

#excluded mouse 4301
#-------------------
embryos.mouse4301.names=c("embryonic_stem_line_Bruce4_p13","embryonic_stem_line_V26_2_p16")
cell.lines.mouse4301.names=c("3T3-L1","C2C12","Baf3","C3H_10T1_2","min6","nih_3T3",
"mIMCD-3",
"neuro2a","RAW_264_7","osteoblast_day14","osteoblast_day21","osteoblast_day5","osteoclasts")
immune.cells.mouse4301.names=c("follicular_B-cells","B-cells_marginal_zone",
"dendritic_cells_lymphoid_CD8a+","dendritic_plasmacytoid_B220+",
"dendritic_cells_myeloid_CD8a-","T-cells_CD4+","T-cells_CD8+","T-cells_foxP3+",
"macrophage_bone_marrow_0hr","macrophage_peri_LPS_thio_0hrs","mast_cells",
"mast_cells_IgE","thymocyte_DP_CD4+CD8+","thymocyte_SP_CD8+","thymocyte_SP_CD4+",
"mega_erythrocyte_progenitor","granulo_mono_progenitor","granulocytes_mac1+gr1+",
"stem_cells__HSC","common_myeloid_progenitor","NK_cells","macrophage_bone_marrow_24h_LPS",
"macrophage_bone_marrow_6hr_LPS","macrophage_peri_LPS_thio_1hrs",
"mast_cells_IgE+antigen_6hr","mast_cells_IgE+antigen_1hr","macrophage_peri_LPS_thio_7hrs",
"macrophage_bone_marrow_2hr_LPS")

#tissue grouping human roth
#--------------------------
adipose.tissues.human.roth.names=c("adipose_tissue_omental","adipose_tissue",
"adipose_tissue_subcutaneous")
cns.human.roth.names=c("cerebellum","cerebral_cortex","amygdala","temporal_lobe",
```

```
"subthalamic_nucleus","pituitary_gland","parietal_lobe","medulla","hypothalamus",
"spinal_cord","hippocampus","substantia_nigra","thalamus","accumbens","corpus_callosum",
"frontal_lobe","midbrain","occipital_lobe","putamen","ventral_tegmental_area",
"vestibular_nuclei_superior","nodose_nucleus")
epidermis.human.roth.names=c("tongue_main_corpus","tongue_superior_part_w/_papillae")
heart.human.roth.names=c("heart_atrium","heart_ventricle","coronary_artery")
kidney.human.roth.names=c("kidney_cortex","kidney_medulla")
lymphoid.structure.human.roth.names=c("lymph_nodes","tonsil")
mammary.gland.human.roth.names=c("mammary_gland","nipple_cross-section")
pns.human.roth.names=c("trigeminal_ganglia","dorsal_root_ganglia")
stomach.human.roth.names=c("stomach_cardiac","stomach_pyloric","stomach_fundus")
uterus.human.roth.names=c("cervix","endometrium","myometrium","vagina","urethra","vulva")

#tissue indexing
#---------------
#mouse novartis
#--------------
cns.mouse=which(colnames(mean.mouse.vsnrma)%in%cns.mouse.names==T)
epidermis.mouse=which(colnames(mean.mouse.vsnrma)%in%epidermis.mouse.names==T)
intestine.mouse=which(colnames(mean.mouse.vsnrma)%in%intestine.mouse.names==T)
ovary.mouse=which(colnames(mean.mouse.vsnrma)%in%ovary.mouse.names==T)
pns.mouse=which(colnames(mean.mouse.vsnrma)%in%pns.mouse.names==T)
embryos.mouse=which(colnames(mean.mouse.vsnrma)%in%embryos.mouse.names==T)
immune.cells.mouse=which(colnames(mean.mouse.vsnrma)%in%immune.cells.mouse.names==T)

#human novartis
#--------------
adrenal.gland.human=which(colnames(mean.human.vsnrma)%in%adrenalgland.human.names==T)
cancer.cells.human=which(colnames(mean.human.vsnrma)%in%cancer.cells.human.names==T)
cell.lines.human=which(colnames(mean.human.vsnrma)%in%cell.lines.human.names==T)
cns.human=which(colnames(mean.human.vsnrma)%in%cns.human.names==T)
embryos.human=which(colnames(mean.human.vsnrma)%in%embryos.human.names==T)
epidermis.human=which(colnames(mean.human.vsnrma)%in%epidermis.human.names==T)
immune.cells.human=which(colnames(mean.human.vsnrma)%in%immune.cells.human.names==T)
lymphnode.human=which(colnames(mean.human.vsnrma)%in%lymphoid.structure.human.names==T)
muscle.human=which(colnames(mean.human.vsnrma)%in%muscle.human.names==T)
pns.human=which(colnames(mean.human.vsnrma)%in%pns.human.names==T)
pancreas.human=which(colnames(mean.human.vsnrma)%in%pancreas.human.names==T)
testis.human=which(colnames(mean.human.vsnrma)%in%testis.human.names==T)
uterus.human=which(colnames(mean.human.vsnrma)%in%uterus.human.names==T)

#mouse 4301
#----------
adipose.tissue.mouse4301=which(colnames(mean4301.vsnrma)%in%
adipose.tissue.mouse4301.names==T)
cell.lines.mouse4301=which(colnames(mean4301.vsnrma)%in%cell.lines.mouse4301.names==T)
cns.mouse4301=which(colnames(mean4301.vsnrma)%in%cns.mouse4301.names==T)
embryos.mouse4301=which(colnames(mean4301.vsnrma)%in%embryos.mouse4301.names==T)
eyes.mouse4301=which(colnames(mean4301.vsnrma)%in%eyes.mouse4301.names==T)
immune.cells.mouse4301=which(colnames(mean4301.vsnrma)%in%immune.cells.mouse4301.names==T)
intestine.mouse4301=which(colnames(mean4301.vsnrma)%in%intestine.mouse4301.names==T)
mammary.gland.mouse4301=which(colnames(mean4301.vsnrma)%in%mammary.gland.mouse4301.names==T)

#human roth
#----------
adipose.tissue.human.roth=which(colnames(mean.human.roth.vsnrma)%in%
adipose.tissues.human.roth.names==T)
cns.human.roth=which(colnames(mean.human.roth.vsnrma)%in%cns.human.roth.names==T)
pns.human.roth=which(colnames(mean.human.roth.vsnrma)%in%pns.human.roth.names==T)
epidermis.human.roth=which(colnames(mean.human.roth.vsnrma)%in%
epidermis.human.roth.names==T)
kidney.human.roth=which(colnames(mean.human.roth.vsnrma)%in%kidney.human.roth.names==T)
```

```
uterus.human.roth=which(colnames(mean.human.roth.vsnrma)%in%uterus.human.roth.names==T)
heart.human.roth=which(colnames(mean.human.roth.vsnrma)%in%heart.human.roth.names==T)
lymphnode.human.roth=which(colnames(mean.human.roth.vsnrma)%in%
lymphoid.structure.human.roth.names==T)
stomach.human.roth=which(colnames(mean.human.roth.vsnrma)%in%stomach.human.roth.names==T)
mammary.gland.human.roth=which(colnames(mean.human.roth.vsnrma)%in%
mammary.gland.human.roth.names==T)

#2. count.over.median
#--------------------
#counts how many tissues are over the cutoff crit

#mouse novartis
#--------------
count.over.median.mouse=function(evec,crit){
x=(evec > (median(evec)+log(crit)))
if(sum(x[cns.mouse]>0))sumx=1 else sumx=0
if(sum(x[epidermis.mouse]>0))sumx=sumx+1
if(sum(x[intestine.mouse]>0))sumx=sumx+1
if(sum(x[ovary.mouse]>0))sumx=sumx+1
if(sum(x[pns.mouse]>0))sumx=sumx+1
sumx=sumx+sum(x[-c(cns.mouse,epidermis.mouse,embryos.mouse,intestine.mouse,ovary.mouse,
pns.mouse,immune.cells.mouse)])
}

calculate.tra.mouse=function(crit){
x=apply(mean.mouse.vsnrma,1,count.over.median.mouse,crit=crit)
tra.index=which(x>0&x<6)
return(tra.index)
}

calculate.housekeeping.mouse=function(crit){
x=apply(mean.mouse.vsnrma,1,count.over.median.mouse,crit=crit)
housekeeping.index=which(x<1)
return(housekeeping.index)
}

#human novartis
#--------------
count.over.median.human=function(evec,crit){
x=(evec > (median(evec)+log(crit)))
if(sum(x[adrenal.gland.human]>0))sumx=1 else sumx=0
if(sum(x[cns.human]>0))sumx=sumx+1
if(sum(x[pns.human]>0))sumx=sumx+1
if(sum(x[pancreas.human]>0))sumx=sumx+1
if(sum(x[uterus.human]>0))sumx=sumx+1
if(sum(x[testis.human]>0))sumx=sumx+1
if(sum(x[lymphnode.human]>0))sumx=sumx+1
if(sum(x[epidermis.human]>0))sumx=sumx+1
if(sum(x[muscle.human]>0))sumx=sumx+1
sumx=sumx+sum(x[-c(adrenal.gland.human,muscle.human,cns.human,pns.human,pancreas.human,
uterus.human,testis.human,
lymphnode.human,immune.cells.human,cancer.cells.human,embryos.human,cell.lines.human)])
}

calculate.tra.human=function(crit){
x=apply(mean.human.vsnrma,1,count.over.median.human,crit=crit)
tra.index=which(x>0&x<6)
return(tra.index)
}

calculate.housekeeping.human=function(crit){
```

```
x=apply(mean.human.vsnrma,1,count.over.median.human,crit=crit)
housekeeping.index=which(x<1)
return(housekeeping.index)
}

#mouse 4301
#----------
count.over.median.mouse4301=function(evec,crit){
x=(evec > (median(evec)+log(crit)))
if(sum(x[cns.mouse4301]>0))sumx=1 else sumx=0
if(sum(x[adipose.tissue.mouse4301]>0))sumx=sumx+1
if(sum(x[intestine.mouse4301]>0))sumx=sumx+1
if(sum(x[eyes.mouse4301]>0))sumx=sumx+1
if(sum(x[mammary.gland.mouse4301]>0))sumx=sumx+1
sumx=sumx+sum(x[-c(cns.mouse4301,adipose.tissue.mouse4301,intestine.mouse4301,
eyes.mouse4301,mammary.gland.mouse4301,embryos.mouse4301,cell.lines.mouse4301,
immune.cells.mouse4301)])
}

calculate.tra.mouse4301=function(crit){
x=apply(mean4301.vsnrma,1,count.over.median.mouse4301,crit=crit)
tra.index=which(x>0&x<6)
return(tra.index)
}

calculate.housekeeping.mouse4301=function(crit){
x=apply(mean4301.vsnrma,1,count.over.median.mouse4301,crit=crit)
housekeeping.index=which(x<1)
return(housekeeping.index)
}

#human roth
#----------
count.over.median.human.roth=function(evec,crit){
x=(evec > (median(evec)+log(crit)))
if(sum(x[adipose.tissue.human.roth]>0))sumx=1 else sumx=0
if(sum(x[cns.human.roth]>0))sumx=sumx+1
if(sum(x[pns.human.roth]>0))sumx=sumx+1
if(sum(x[kidney.human.roth]>0))sumx=sumx+1
if(sum(x[uterus.human.roth]>0))sumx=sumx+1
if(sum(x[heart.human.roth]>0))sumx=sumx+1
if(sum(x[lymphnode.human.roth]>0))sumx=sumx+1
if(sum(x[epidermis.human.roth]>0))sumx=sumx+1
if(sum(x[stomach.human.roth]>0))sumx=sumx+1
if(sum(x[mammary.gland.human.roth]>0))sumx=sumx+1
sumx=sumx+sum(x[-c(adipose.tissue.human.roth,cns.human.roth,epidermis.human.roth,
pns.human.roth,kidney.human.roth,uterus.human.roth,heart.human.roth,lymphnode.human.roth,
stomach.human.roth,mammary.gland.human.roth)])
}

calculate.tra.human.roth=function(crit){
x=apply(mean.human.roth.vsnrma,1,count.over.median.human.roth,crit=crit)
tra.index=which(x>0&x<6)
return(tra.index)
}

calculate.housekeeping.human.roth=function(crit){
x=apply(mean.human.roth.vsnrma,1,count.over.median.human.roth,crit=crit)
housekeeping.index=which(x<1)
return(housekeeping.index)
}
```

```
#3. calculate tras
#-----------------
#mouse novartis
tra.3x.median.mouse=apply(mean.mouse.vsnrma,1,count.over.median.mouse,crit=3)
tra.5x.median.mouse=apply(mean.mouse.vsnrma,1,count.over.median.mouse,crit=5)
tra.10x.median.mouse=apply(mean.mouse.vsnrma,1,count.over.median.mouse,crit=10)
tra.20x.median.mouse=apply(mean.mouse.vsnrma,1,count.over.median.mouse,crit=20)

#human novartis
tra.3x.median.human=apply(mean.human.vsnrma,1,count.over.median.human,crit=3)
tra.5x.median.human=apply(mean.human.vsnrma,1,count.over.median.human,crit=5)
tra.10x.median.human=apply(mean.human.vsnrma,1,count.over.median.human,crit=10)
tra.20x.median.human=apply(mean.human.vsnrma,1,count.over.median.human,crit=20)

#mouse 4301
tra.3x.median.mouse4301=apply(mean4301.vsnrma,1,count.over.median.mouse4301,crit=3)
tra.5x.median.mouse4301=apply(mean4301.vsnrma,1,count.over.median.mouse4301,crit=5)
tra.10x.median.mouse4301=apply(mean4301.vsnrma,1,count.over.median.mouse4301,crit=10)
tra.20x.median.mouse4301=apply(mean4301.vsnrma,1,count.over.median.mouse4301,crit=20)

#human roth
tra.3x.median.human.roth=apply(mean.human.roth.vsnrma,1,count.over.median.human.roth,
crit=3)
tra.5x.median.human.roth=apply(mean.human.roth.vsnrma,1,count.over.median.human.roth,
crit=5)
tra.10x.median.human.roth=apply(mean.human.roth.vsnrma,1,count.over.median.human.roth,
crit=10)
tra.20x.median.human.roth=apply(mean.human.roth.vsnrma,1,count.over.median.human.roth,
crit=20)

#4. saturationplot.R
#-------------------
#mouse novartis
maxtiss.over.3xmedian = vector(len=61)
maxtiss.over.5xmedian = vector(len=61)
maxtiss.over.10xmedian = vector(len=61)
maxtiss.over.20xmedian = vector(len=61)

for (i in 1:61){
maxtiss.over.3xmedian[i] = sum(tra.3x.median.mouse> 0 & tra.3x.median.mouse < i)
maxtiss.over.5xmedian[i] = sum(tra.5x.median.mouse> 0 & tra.5x.median.mouse < i)
maxtiss.over.10xmedian[i] = sum(tra.10x.median.mouse> 0 & tra.10x.median.mouse < i)
maxtiss.over.20xmedian[i] = sum(tra.20x.median.mouse> 0 & tra.20x.median.mouse < i)
}

# graphik plotten
#----------------
x11(h=8,w=8)
plot(maxtiss.over.3xmedian, type="p",xlab="number of tissues",ylab="number of transcripts")
points(maxtiss.over.5xmedian, col="red")
points(maxtiss.over.10xmedian, col="green")
points(maxtiss.over.20xmedian, col="blue")
abline(v=5)

legend("topright",pch="o",c("","3x over median","5x over median","10x over median",
"20x over median"),col=c("white","black","red","green","blue")
)

title(main="Number of transcripts over 3x, 5x, 10x, 20x the median \n
in 61 mouse tissues (gngnf1)")

setwd("/home/dinkelac/data/mouse/plots")
```

```
dev.copy2eps(file="saturationplot.mouse.eps")

#human novartis
maxtiss.over.3xmedian = vector(len=79)
maxtiss.over.5xmedian = vector(len=79)
maxtiss.over.10xmedian = vector(len=79)
maxtiss.over.20xmedian = vector(len=79)

for (i in 1:79){
maxtiss.over.3xmedian[i] = sum(tra.3x.median.human> 0 & tra.3x.median.human < i)
maxtiss.over.5xmedian[i] = sum(tra.5x.median.human> 0 & tra.5x.median.human < i)
maxtiss.over.10xmedian[i] = sum(tra.10x.median.human> 0 & tra.10x.median.human < i)
maxtiss.over.20xmedian[i] = sum(tra.20x.median.human> 0 & tra.20x.median.human < i)
}

# graphik plotten
#----------------
x11(h=8,w=8)
plot(maxtiss.over.3xmedian, type="p",xlab="number of tissues",ylab="number of transcripts")
points(maxtiss.over.5xmedian, col="red")
points(maxtiss.over.10xmedian, col="green")
points(maxtiss.over.20xmedian, col="blue")
abline(v=5)

legend("topright",pch="o",c("","3x over median","5x over median","10x over median",
"20x over median"),col=c("white","black","red","green","blue")
)

title(main="Number of transcripts over 3x, 5x, 10x, 20x the median \n
in 79 human tissues")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="saturationplot.human.eps")

#mouse4301
maxtiss.over.3x.median = vector(len=91)
maxtiss.over.5x.median = vector(len=91)
maxtiss.over.10x.median = vector(len=91)
maxtiss.over.20x.median = vector(len=91)

for (i in 1:91){
maxtiss.over.3xmedian[i] = sum(tra.3x.median.mouse4301>0& tra.3x.median.mouse4301<i)
maxtiss.over.5xmedian[i] = sum(tra.5x.median.mouse4301>0& tra.5x.median.mouse4301<i)
maxtiss.over.10xmedian[i] = sum(tra.10x.median.mouse4301>0& tra.10x.median.mouse4301<i)
maxtiss.over.20xmedian[i] = sum(tra.20x.median.mouse4301>0& tra.20x.median.mouse4301<i)
}

# graphik plotten
#----------------
x11(h=8,w=8)
plot(maxtiss.over.3xmedian, type="p",xlab="number of tissues",ylab="number of transcripts")
points(maxtiss.over.5xmedian, col="red")
points(maxtiss.over.10xmedian, col="green")
points(maxtiss.over.20xmedian, col="blue")
abline(v=5,col="blue")
abline(v=9,col="red")
abline(v=6,col="green")
abline(v=12)

legend("topright",pch="o",c("","3x over median, 12 tissues","5x over median, 9 tissues",
"10x over median, 6 tissues","20x over median, 5 tissues"),col=c("white","black","red",
"green","blue")
```

```
)

title(main="Number of transcripts over 3x, 5x, 10x, 20x the median \n
in 91 mouse tissues (4302)")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="saturationplot.mouse4302.eps")

#human roth
maxtiss.over.3xmedian = vector(len=65)
maxtiss.over.5xmedian = vector(len=65)
maxtiss.over.10xmedian = vector(len=65)
maxtiss.over.20xmedian = vector(len=65)

for (i in 1:65){
maxtiss.over.3xmedian[i] = sum(tra.3x.median.human.roth> 0 & tra.3x.median.human.roth < i)
maxtiss.over.5xmedian[i] = sum(tra.5x.median.human.roth> 0 & tra.5x.median.human.roth < i)
maxtiss.over.10xmedian[i] = sum(tra.10x.median.human.roth> 0 & tra.10x.median.human.roth < i)
maxtiss.over.20xmedian[i] = sum(tra.20x.median.human.roth> 0 & tra.20x.median.human.roth < i)
}

# graphik plotten
#----------------
x11(h=8,w=8)
plot(maxtiss.over.3xmedian, type="p",xlab="number of tissues",ylab="number of transcripts")
points(maxtiss.over.5xmedian, col="red")
points(maxtiss.over.10xmedian, col="green")
points(maxtiss.over.20xmedian, col="blue")
abline(v=5)

legend("topright",pch="o",c("","3x over median","5x over median","10x over median",
"20x over median"),col=c("white","black","red","green","blue")
)

title(main="Number of transcripts over 3x, 5x, 10x, 20x the median \n
in 65 human tissues (Roth data)")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="saturationplot.human.roth.eps")

setwd("/home/dinkelac/data/mouse/sessions/rda")
save.image(file="tras.rda")

#5. calculate the tra.index
#--------------------------
tra.index.mouse.3x=calculate.tra.mouse(3)
tra.index.mouse.5x=calculate.tra.mouse(5)
tra.index.mouse.10x=calculate.tra.mouse(10)
tra.index.mouse.20x=calculate.tra.mouse(20)

tra.index.human.3x=calculate.tra.human(3)
tra.index.human.5x=calculate.tra.human(5)
tra.index.human.10x=calculate.tra.human(10)
tra.index.human.20x=calculate.tra.human(20)

tra.index.mouse4301.3x=calculate.tra.mouse4301(3)
tra.index.mouse4301.5x=calculate.tra.mouse4301(5)
tra.index.mouse4301.10x=calculate.tra.mouse4301(10)
tra.index.mouse4301.20x=calculate.tra.mouse4301(20)

tra.index.human.roth.3x=calculate.tra.human.roth(3)
tra.index.human.roth.5x=calculate.tra.human.roth(5)
```

```
tra.index.human.roth.10x=calculate.tra.human.roth(10)
tra.index.human.roth.20x=calculate.tra.human.roth(20)

#[-AFFX Sonden]
tra.mouse.3x=tra.index.mouse.3x[29:length(tra.index.mouse.3x)]
tra.mouse.5x=tra.index.mouse.5x[26:length(tra.index.mouse.5x)]
tra.mouse.10x=tra.index.mouse.10x[5:length(tra.index.mouse.10x)]
tra.mouse.20x=tra.index.mouse.20x[4:length(tra.index.mouse.20x)]

tra.human.3x=tra.index.human.3x[21:length(tra.index.human.3x)]
tra.human.5x=tra.index.human.5x[15:length(tra.index.human.5x)]
tra.human.10x=tra.index.human.10x[7:length(tra.index.human.10x)]
tra.human.20x=tra.index.human.20x[1:length(tra.index.human.20x)]

tra.mouse4301.3x=tra.index.mouse4301.3x[28:length(tra.index.mouse4301.3x)]
tra.mouse4301.5x=tra.index.mouse4301.5x[22:length(tra.index.mouse4301.5x)]
tra.mouse4301.10x=tra.index.mouse4301.10x[30:length(tra.index.mouse4301.10x)]
tra.mouse4301.20x=tra.index.mouse4301.20x[12:length(tra.index.mouse4301.20x)]

tra.human.roth.3x=tra.index.human.roth.3x[8:length(tra.index.human.roth.3x)]
tra.human.roth.5x=tra.index.human.roth.5x[5:length(tra.index.human.roth.5x)]
tra.human.roth.10x=tra.index.human.roth.10x[2:length(tra.index.human.roth.10x)]
tra.human.roth.20x=tra.index.human.roth.20x[2:length(tra.index.human.roth.20x)]

#calculate housekeeping genes
#----------------------------
housekeeping.mouse=calculate.housekeeping.mouse(2)
housekeeping.mouse.2x=housekeeping.mouse[29:length(housekeeping.mouse)]

housekeeping.human=calculate.housekeeping.human(2)
housekeeping.human.2x=housekeeping.human[33:length(housekeeping.human)]

housekeeping.mouse4301=calculate.housekeeping.mouse4301(2)
housekeeping.mouse4301.2x=housekeeping.mouse4301[6:length(housekeeping.mouse4301)]

housekeeping.human.roth=calculate.housekeeping.human.roth(2)
housekeeping.human.roth.2x=housekeeping.human.roth[44:length(housekeeping.human.roth)]

setwd("/home/dinkelac/data/mouse/tables/")

write.table(names(tra.mouse.3x),"tra.mouse.3x.transcripts.txt")
write.table(names(tra.mouse.5x),"tra.mouse.5x.transcripts.txt")
write.table(names(tra.mouse.10x),"tra.mouse.10x.transcripts.txt")
write.table(names(tra.mouse.20x),"tra.mouse.20x.transcripts.txt")

write.table(names(tra.human.3x),"tra.human.3x.transcripts.txt")
write.table(names(tra.human.5x),"tra.human.5x.transcripts.txt")
write.table(names(tra.human.10x),"tra.human.10x.transcripts.txt")
write.table(names(tra.human.20x),"tra.human.20x.transcripts.txt")

write.table(names(tra.mouse4301.3x),"tra.mouse4301.3x.transcripts.txt")
write.table(names(tra.mouse4301.5x),"tra.mouse4301.5x.transcripts.txt")
write.table(names(tra.mouse4301.10x),"tra.mouse4301.10x.transcripts.txt")
write.table(names(tra.mouse4301.20x),"tra.mouse4301.20x.transcripts.txt")

write.table(names(tra.human.roth.3x),"tra.human.roth.3x.transcripts.txt")
write.table(names(tra.human.roth.5x),"tra.human.roth.5x.transcripts.txt")
write.table(names(tra.human.roth.10x),"tra.human.roth.10x.transcripts.txt")
write.table(names(tra.human.roth.20x),"tra.human.roth.20x.transcripts.txt")

write.table(names(housekeeping.mouse.2x),"housekeeping.mouse.2x.transcripts.txt")
write.table(names(housekeeping.human.2x),"housekeeping.human.2x.transcripts.txt")
```

```
write.table(names(housekeeping.mouse4301.2x),"housekeeping.mouse4301.2x.transcripts.txt")
write.table(names(housekeeping.human.roth.2x),"human.roth.2x.transcripts.txt")

write.table(rownames(mean.mouse.vsnrma),"transcripts.all.mouse.txt")
write.table(rownames(mean.human.vsnrma),"transcripts.all.human.txt")
write.table(rownames(mean4301.vsnrma),"transcripts.all.mouse4301.txt")
write.table(rownames(mean.human.roth.vsnrma),"transcripts.all.human.roth.txt")

#in oocalc AFFX Sonden loeschen und erste Zeile loeschen

x=apply(mean.mouse.vsnrma,1,count.over.median.mouse,5)
tra.mouse.index1=which(x==1)
tra.mouse.index2=which(x==2)
tra.mouse.index3=which(x==3)
tra.mouse.index4=which(x==4)
tra.mouse.index5=which(x==5)
tra.mouse.index6=which(x>6)
tra.mouse.index0=which(x<1)

x=apply(mean.human.vsnrma,1,count.over.median.human,5)
tra.human.index1=which(x==1)
tra.human.index2=which(x==2)
tra.human.index3=which(x==3)
tra.human.index4=which(x==4)
tra.human.index5=which(x==5)
tra.human.index6=which(x>6)
tra.human.index0=which(x<1)

x=apply(mean4301.vsnrma,1,count.over.median.mouse4301,5)
tra.mouse4301.index1=which(x==1)
tra.mouse4301.index2=which(x==2)
tra.mouse4301.index3=which(x==3)
tra.mouse4301.index4=which(x==4)
tra.mouse4301.index5=which(x==5)
tra.mouse4301.index6=which(x>6)
tra.mouse4301.index0=which(x<1)

x=apply(mean.human.roth.vsnrma,1,count.over.median.human.roth,5)
tra.human.roth.index1=which(x==1)
tra.human.roth.index2=which(x==2)
tra.human.roth.index3=which(x==3)
tra.human.roth.index4=which(x==4)
tra.human.roth.index5=which(x==5)
tra.human.roth.index6=which(x>6)
tra.human.roth.index0=which(x<1)

setwd("/home/dinkelac/data/mouse/sessions/rda/")
save.image(file="tras.rda")

#venn Diagram

#mouse
transcripts=union(names(tra.mouse.5x),names(tra.mouse4301.5x))
matches=matrix(ncol=2,nrow=length(transcripts))
matches[,1]=as.numeric(is.element(transcripts,names(tra.mouse.5x)))
matches[,2]=as.numeric(is.element(transcripts,names(tra.mouse4301.5x)))
colnames(matches)=c("tra.mouse","tra.mouse4301")
vennDiagram(matches,circle.col=c("red","orange"),lwd=3)

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="venn.tra.mouse.transcripts.eps")
```

```
#human
transcripts=union(names(tra.human.5x),names(tra.human.roth.5x))
matches=matrix(ncol=2,nrow=length(transcripts))
matches[,1]=as.numeric(is.element(transcripts,names(tra.human.5x)))
matches[,2]=as.numeric(is.element(transcripts,names(tra.human.roth.5x)))
colnames(matches)=c("tra.human","tra.human.roth")
vennDiagram(matches,circle.col=c("blue","lightblue"),lwd=3)

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="venn.tra.human.transcripts.eps")


#6.1 Download annotation
#-----------------------
#http://www.ensembl.org/biomart/martview/177a6c7a103e40daf78d7b149b7a1920
#ensembl 75 (80)
#homo sapiens GRCh37.p13, mus musculus GRCm38.p2 (GRCH38.p2)
#mus musculus, 80, GRCm39.p3
#filters
#(->ensemble human gene IDs only)
#attributes
#->features
#Gene
#Ensembl Gene ID,Ensembl Transcript ID,Description,Chromosome Name, Gene Start,
#Strand, Band
#External
#MGI symbol

#als csv abspeichern unter "ensembl.75.mouse.txt","ensembl.75.human.txt"
setwd("/home/dinkelac/data/mouse/tables/")

ensembl.75.mouse=read.csv(file="ensembl.75.mouse.txt")
ensembl.75.human=read.csv(file="ensembl.75.human.txt")

#doppelte transcripte raus
doppelte.mouse=which(duplicated(ensembl.75.mouse[,2]))
doppelte.human=which(duplicated(ensembl.75.human[,2]))

ensembl.mouse=ensembl.75.mouse[-doppelte.mouse,]
rownames(ensembl.mouse)=ensembl.mouse[,2]

ensembl.human=ensembl.75.human[-doppelte.human,]
rownames(ensembl.human)=ensembl.human[,2]

setwd("/home/dinkelac/data/mouse/tables/")

write.table(ensembl.mouse,file="ensembl.75.mouse.transcripts.csv",sep=",",row.names=F)
write.table(ensembl.human,file="ensembl.75.human.transcripts.csv",sep=",",row.names=F)

#Gensymbole
symbol.mouse=as.character(ensembl.mouse[,8])
names(symbol.mouse)=ensembl.mouse[,2]

symbol.human=as.character(ensembl.human[,8])
names(symbol.human)=ensembl.human[,2]

#auf genebene
#------------
ensembl.genes.mouse=unique(as.character(ensembl.75.mouse[,1]))
ensembl.transcripts.mouse=vector(len=length(ensembl.genes.mouse))
ensembl.description.mouse=vector(len=length(ensembl.genes.mouse))
ensembl.chromosome.mouse=vector(len=length(ensembl.genes.mouse))
```

```
ensembl.startsite.mouse=vector(len=length(ensembl.genes.mouse))
ensembl.strand.mouse=vector(len=length(ensembl.genes.mouse))
ensembl.band.mouse=vector(len=length(ensembl.genes.mouse))
ensembl.symbol.mouse=vector(len=length(ensembl.genes.mouse))

for(i in 1:length(ensembl.genes.mouse)){
gene.name=ensembl.genes.mouse[i]
index=which(ensembl.75.mouse[,1]==gene.name)
transcript.names=unique(as.character(ensembl.75.mouse[index,2]))
descript.names=unique(as.character(ensembl.75.mouse[index,3]))
chromosome=unique(as.character(ensembl.75.mouse[index,4]))
startsite=unique(as.character(ensembl.75.mouse[index,5]))
strand=unique(as.character(ensembl.75.mouse[index,6]))
band=unique(as.character(ensembl.75.mouse[index,7]))
symbol=unique(as.character(ensembl.75.mouse[index,8]))

if(length(transcript.names)>1){
ensembl.transcripts.mouse[i]=paste(transcript.names,collapse="/")
}else{
ensembl.transcripts.mouse[i]=transcript.names
}

if(length(descript.names)>1){
ensembl.description.mouse[i]=paste(descript.names,collapse="/")
}else{
ensembl.description.mouse[i]=descript.names
}

if(length(chromosome)>1){
ensembl.chromosome.mouse[i]=paste(chromosome,collapes="/")
}else{
ensembl.chromosome.mouse[i]=chromosome
}

if(length(startsite)>1){
ensembl.startsite.mouse[i]=paste(startsite,collapes="/")
}else{
ensembl.startsite.mouse[i]=startsite
}

if(length(strand)>1){
ensembl.strand.mouse[i]=paste(strand,collapes="/")
}else{
ensembl.strand.mouse[i]=strand
}

if(length(band)>1){
ensembl.band.mouse[i]=paste(band,collapes="/")
}else{
ensembl.band.mouse[i]=band
}

if(length(symbol)>1){
ensembl.symbol.mouse[i]=paste(symbol,collapes="/")
}else{
ensembl.symbol.mouse[i]=symbol
}
}

ensembl.75.mouse.table=cbind(ensembl.genes.mouse,ensembl.transcripts.mouse,
ensembl.description.mouse,
ensembl.chromosome.mouse,ensembl.startsite.mouse,ensembl.strand.mouse,ensembl.band.mouse,
```

```
ensembl.symbol.mouse)

rownames(ensembl.75.mouse.table)=ensembl.75.mouse.table[,1]

setwd("/home/dinkelac/data/mouse/tables/")

write.table(ensembl.75.human.table,file="ensembl.75.human.genes.csv",sep=",",row.names=F)
write.table(ensembl.75.mouse.table,file="ensembl.75.mouse.genes.csv",sep=",",row.names=F)

#6.2 annotate chips
#------------------
mouse.transcripts.all=rownames(mouse.vsnrma)
mouse.transcripts=mouse.transcripts.all[65:length(mouse.transcripts.all)]

mouse.4301.transcripts.all=rownames(mean4301.vsnrma)
mouse.4301.transcripts=mouse.4301.transcripts.all[65:length(mouse.4301.transcripts.all)]

human.transcripts.all=rownames(mean.human.vsnrma)
human.transcripts=human.transcripts.all[69:length(human.transcripts.all)]

human.roth.transcripts.all=rownames(mean.human.roth.vsnrma)
human.roth.transcripts=human.roth.transcripts.all[63:length(human.roth.transcripts.all)]

#6.2.1 annotate chips on transcript level
#----------------------------------------
transcript.names=substr(mouse.transcripts,1,nchar(mouse.transcripts)-3)
chromosome=as.character(ensembl.human[transcript.names,4])
startside=ensembl.human[transcript.names,5]
strand=ensembl.human[transcript.names,6]
band=as.character(ensembl.human[transcript.names,7])
symbol=as.character(ensembl.human[transcript.names,8])

#names(symbol)=gngnf1.transcript.names

human.roth.transcript.annotated=cbind(transcript.names,chromosome,startside,strand,band,
symbol)

setwd("/home/dinkelac/data/mouse/tables/")
write.table(human.roth.transcript.annotated,file="human.roth.chip.transcript.annotated.csv",
sep=",",row.names=F)

#6.2.2 annotated chips on gene level
#-----------------------------------
mouse.genes=unique(as.character(ensembl.mouse[transcript.names,1]))
mouse.4301.genes=unique(as.character(ensembl.mouse[transcript.names,1]))
human.genes=unique(as.character(ensembl.human[transcript.names,1]))
human.roth.genes=unique(as.character(ensembl.human[transcript.names,1]))
#----------------
gene.names=human.roth.genes
chromosome=as.character(ensembl.75.human.table[gene.names,4])
startside=ensembl.75.human.table[gene.names,5]
strand=ensembl.75.human.table[gene.names,6]
band=as.character(ensembl.75.human.table[gene.names,7])
symbol=as.character(ensembl.75.human.table[gene.names,8])

#names(symbol)=gngnf1.transcript.names

human.roth.genes.annotated=cbind(gene.names,chromosome,startside,strand,band,symbol)

setwd("/home/dinkelac/data/mouse/tables/")
write.table(human.roth.genes.annotated,file="human.roth.chip.genes.annotated.csv",
sep=",",row.names=F)
```

```
#7. plot TRAs
#------------
farben.mouse=c("darkred","red3","oldlace","tomato2","#FF5600FF","red","#FF8100FF",
"orange","oldlace","oldlace","#FFED00FF","#FFED00FF","green","green","green","green",
"green","green","green","green","green","green","green","green","green","seagreen3",
"seagreen3","seagreen3","seagreen3","white","white","white","white","white","lightblue",
"lightblue","lightblue","lightblue","#00F4FFFF","#00F4FFFF","#00B4FFFF","#005DFFFF",
"blue2","blue3","navy","purple4","slateblue","#9000FFFF","purple","#BB00FFFF","violet",
"thistle","mistyrose","pink","pink2","steelblue","steelblue2","steelblue3","oldlace",
"thistle2","blue4")

tissue.order.mouse=c(6,7,31,8,51,21,60,43,29,30,22,37,34,1,12,13,11,14,39,19,61,24,40,
41,20,27,26,36,25,45,46,47,48,52,53,17,54,56,50,58,49,4,9,10,32,2,44,42,28,15,33,16,59,
23,38,5,55,35,57,18,3)

farben.human=c("darkred","red3","tomato2","#FF5600FF","wheat","wheat","wheat","green",
"green","green","green","green","green","green","green","green","green","green","green",
"green","green","green","green","green","green","green","seagreen3","seagreen3","seagreen3",
"seagreen3","white","white","white","white","#FF8100FF","orange","tan","yellow","oldlace",
"orange","oldlace","oldlace","oldlace","#00B4FFFF","#00B4FFFF","oldlace","oldlace","oldlace",
"oldlace","oldlace","oldlace","white","white","white","white","white","white","lightblue",
"steelblue","#005DFFFF","navy","navy","#0007FFFF","#3900FFFF","slateblue","slateblue",
"rosybrown","rosybrown","purple","purple","purple","purple","purple","thistle","#BB00FFFF",
"violet","oldlace","darkblue","darkblue")

tissue.order.human=c(38,37,36,50,51,58,71,1,15,16,2,3,17,4,5,6,7,8,67,9,10,19,18,11,12,14,20,
21,13,22,23,25,26,24,28,63,49,79,52,31,64,32,33,34,35,27,53,54,55,56,57,73:78,60,59,39,29,
30,40,43,66,68,61,62,44,46,47,48,45,69,70,65,72,41,42)

farben.mouse4301=c("red3","red3","peru","white","white","tan","#FF8100FF","orange","green",
"green","green","green","green","green","green","green","green","green","seagreen3","red",
"red","red","red","red","red","#00F4FFFF","#00F4FFFF","white","white","white","white",
"white","bisque1","white","white","white","lightblue","#00E0FFFF","#00C2FFFF","#00A3FFFF",
"#0085FFFF","#0066FFFF","white","white","#000AFFFF","#1400FFFF","white","white","#8F00FFFF",
"white","#CC00FFFF","#EB00FFFF","#FF00F5FF","#FF00F5FF","#FF00D6FF","#FF00B8FF","#FF0099FF",
"lightblue","lightblue","lightblue","#00D1FFFF","white","white","white","#00BDFFFF","white",
"white","white","#0080FFFF","#007AFFFF")

tissue.order.mouse4301=c(2:4,7,28,8:10,5,13:15,23,32:33,71,79,64,22,27,16,36,38,39,75,34,35,
17,81,29,30,56,18:21,26,31,37,40:43,47,50:53,57,62:63,68:70,72:73,76:78,80,82:91)

farben.human.roth=c("yellow","green","green","green","deeppink1","coral","yellow","yellow",
"gold","cornflowerblue","green4","navajowhite","green","green","#005DFFFF","green",
"cornflowerblue","cornflowerblue","green","green","green","navajowhite","lightblue",
"orangered4","green","slateblue","green","purple","navajowhite","slateblue","lightblue",
"lightcyan","purple","green","wheat","green","green","lavender","tomato2","steelblue",
"green","indianred","blue","green","red3","navajowhite","steelblue","steelblue",
"#FF5600FF","green","green","green","green","blue3","#00B4FFFF","#FF8100FF","green",
"navajowhite","green4","green","blue3","lightcyan","tomato2","navajowhite","navajowhite")

tissue.order.human.roth=c(7,1,8,9,13,16,2,52,50,44,34,27,21,41,20,51,4,14,19,25,37,3,57,
60,36,53,59,11,43,54,61,23,31,12,22,29,58,65,64,38,55,17,10,18,32,62,40,47,48,15,5,30,26,
28,33,45,49,46,56,63,6,24,39,35,42)

#rename human tissues

#  human.tissues=colnames(mean.human.vsnrma)
#  human.tissues[22]="dorsal root ganglion"
#  human.tissues[35]="whole blood"
#  human.tissues[38]="adipocyte"
#  human.tissues[40]="placenta"
```

```
#  human.tissues[65]="tongue"

#  colnames(mean.human.vsnrma)=human.tissues

#TRA plotten
#-----------
#mouse novartis
plotten.mouse=function(gene.name){
gene.name1=substr(gene.name,1,nchar(gene.name)-3)
s=symbol.mouse[gene.name1]

file.name=paste("gngnf1.10.5x_",gene.name1,".eps",sep="")
titel=paste(s, gene.name1,sep="\n")

x11(w=10,h=7)
par(las=2)
mmi=c(2.3,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
 barplot(exp(mean.mouse.vsnrma[gene.name,])[tissue.order.mouse],col=farben.mouse,
 cex.names=0.8,cex.axis=0.8)
 med.act=exp(median(mean.mouse.vsnrma[gene.name,]))
 abline(h=med.act,lty=2,col="black")
 abline(h=5*med.act,lty=2,col="red")
 title(main=titel)
legend("topright",pch="-",c("median","5xmedian"),col=c("black","red"),cex=0.8)
}

plotten.human=function(gene.name){
gene.name1=substr(gene.name,1,nchar(gene.name)-3)
s=symbol.human[gene.name1]

file.name=paste("human_tra_2014_",gene.name1,".eps",sep="")
titel=paste(s, gene.name1,sep="\n")

x11(w=12,h=7)
par(las=2)
mmi=c(2.9,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
 barplot(exp(mean.human.vsnrma[gene.name,])[tissue.order.human],col=farben.human,
 cex.names=0.8,cex.axis=0.8)
 med.act=exp(median(mean.human.vsnrma[gene.name,]))
 abline(h=med.act,lty=2,col="black")
 abline(h=2*med.act,lty=2,col="green")
 abline(h=5*med.act,lty=2,col="red")
 title(main=titel)
legend("topright",pch="-",c("median","5xmedian"),col=c("black","red"),cex=0.8)
}

plotten.mouse4301=function(gene.name){
gene.name1=substr(gene.name,1,nchar(gene.name)-3)
file.name=paste("4302_tra_2014_",gene.name1,".eps",sep="")
s=symbol.mouse[gene.name1]
titel=paste(s,gene.name1,sep="\n")

x11(w=12,h=7)
#png(file=file.name,width=1000,height=700,res=72)
par(las=2)
#par(cex=1.3)
mmi=c(2.9,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
 barplot(exp(mean4301.vsnrma[gene.name,])[tissue.order.mouse4301],col=farben.mouse4301,
 cex.names=0.8,cex.axis=0.8)
```

```
 med.act=exp(median(mean4301.vsnrma[gene.name,]))
 abline(h=med.act,lty=2,col="black")
 abline(h=5*med.act,lty=2,col="red")
 title(main=titel)
legend("topright",pch="-",c("median","5xmedian"),col=c("black","red"),cex=0.8)
}

plotten.human.roth=function(gene.name){
gene.name1=substr(gene.name,1,nchar(gene.name)-3)
s=symbol.human[gene.name1]

file.name=paste("human.roth_",gene.name1,".eps",sep="")
titel=paste(s, gene.name1,sep="\n")

x11(w=12,h=7)
par(las=2)
mmi=c(2.9,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
 barplot(exp(mean.human.roth.vsnrma[gene.name,])[tissue.order.human.roth],
 col=farben.human.roth[tissue.order.human.roth],cex.names=0.8,cex.axis=0.8)
 med.act=exp(median(mean.human.roth.vsnrma[gene.name,]))
 abline(h=med.act,lty=2,col="black")
 abline(h=2*med.act,lty=2,col="green")
 abline(h=5*med.act,lty=2,col="red")
 title(main=titel)
legend("topright",pch="-",c("median","5xmedian"),col=c("black","red"),cex=0.8)
}

#plotten.png
#-----------
#mouse novartis
plotten.mouse.png=function(gene.name,median){
gene.name1=substr(gene.name,1,nchar(gene.name)-3)
file.name=paste("mouse.",gene.name1,".png",sep="")
s=symbol.mouse[gene.name1]
titel=paste(s, gene.name1,sep="\n")
#x11(w=10,h=7)
png(filename=file.name,width=1000,height=700,res=72)

par(las=2)
par(cex=1.3)
mmi=c(2.9,1.0477939,1.0477939,0.5366749)
par(mai=mmi)

 barplot(exp(mean.mouse.vsnrma[gene.name,])[tissue.order.mouse],col=farben.mouse,
 cex.names=0.8,cex.axis=0.8)

 med.act=exp(median(mean.mouse.vsnrma[gene.name,]))
 abline(h=med.act,lty=2,col="black")

 if(median==3){
 abline(h=3*med.act,lty=2,col="green")
  title(main=titel)
legend("topright",pch="-",c("median","3 x median"),col=c("black","green"),cex=0.8)
 }
 if(median==5){
 abline(h=5*med.act,lty=2,col="red")
  title(main=titel)
legend("topright",pch="-",c("median","5 x median"),col=c("black","red"),cex=0.8)
 }
 if(median==10){
 abline(h=10*med.act,lty=2,col="blue")
```

```
  title(main=titel)
legend("topright",pch="-",c("median","10 x median"),col=c("black","blue"),cex=0.8)
 }
 if(median==20){
 abline(h=20*med.act,lty=2,col="brown")
  title(main=titel)
legend("topright",pch="-",c("median","20 x median"),col=c("black","brown"),cex=0.8)
 }

dev.off()
}

mean.human.vsnrma.plotten=mean.human.vsnrma
human.tissue.names=dimnames(mean.human.vsnrma.plotten)[2]

human.tissue.names[[1]][15]="CaudateNucleus"
human.tissue.names[[1]][69]="Thymus"
human.tissue.names[[1]][79]="Kidney"
human.tissue.names[[1]][68]="Skin"
human.tissue.names[[1]][72]="Trachea"
human.tissue.names[[1]][75]="Leukemialymphoblastic(molt4)"
human.tissue.names[[1]][76]="leukemiapromyelocytic(hl60)"
human.tissue.names[[1]][77]="LymphomaBurkittsDaudi"
human.tissue.names[[1]][76]="LeukemiaPromyelocytic(hl60)"
human.tissue.names[[1]][66]="Salivarygland"
human.tissue.names[[1]][17]="GlobusPallidus"
human.tissue.names[[1]][23]="FetalBrain"
human.tissue.names[[1]][58]="Lymphnode"
human.tissue.names[[1]][16]="Cerebellum"
human.tissue.names[[1]][78]="LymphomaBurkittsRaji"
human.tissue.names[[1]][44]="Testis"
human.tissue.names[[1]][50]="AtrioventricularNode"
human.tissue.names[[1]][65]="Tongue"
human.tissue.names[[1]][40]="Placenta"
human.tissue.names[[1]][38]="Adipocyte"
human.tissue.names[[1]][35]="WholeBlood"
human.tissue.names[[1]][36]="AdrenalGland"
human.tissue.names[[1]][25]="FetalLiver"
human.tissue.names[[1]][19]="SpinalCord"
human.tissue.names[[1]][18]="SubthalamicNucleus"
human.tissue.names[[1]][26]="FetalLung"
human.tissue.names[[1]][24]="FetalThyroid"
human.tissue.names[[1]][74]="LeukemiaChronicMyelogenous(k562)"
human.tissue.names[[1]][28]="BronchialEpithelialCells"
human.tissue.names[[1]][34]="Bonemarrow"
human.tissue.names[[1]][20]="CiliaryGanglion"
human.tissue.names[[1]][22]="DorsalRootGanglion"
human.tissue.names[[1]][53]="Bcells(PB-CD19+)"
human.tissue.names[[1]][54]="Tcells(PB-CD4+)"
human.tissue.names[[1]][52]="BLymphoblasts(721)"
human.tissue.names[[1]][55]="NKcells(PB-CD56)"
human.tissue.names[[1]][56]="Tcells(PB-CD8)"
human.tissue.names[[1]][27]="DentriticCells(DC PB-BDCA4+)"

dimnames(mean.human.vsnrma.plotten)[2]=human.tissue.names

#human
plotten.human.png=function(gene.name,median){
gene.name1=substr(gene.name,1,nchar(gene.name)-3)
file.name=paste("human.",gene.name1,".png",sep="")
s=symbol.human[gene.name1]
titel=paste(s, gene.name1,sep="\n")
```

```
#w=12 ?, width=1200
#x11(w=10,h=7)
png(file=file.name,width=1000,height=700,res=72)
par(las=2)
#1.5 ?
par(cex=1.3)
mmi=c(2.9,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
#cex.names=0.6
 barplot(exp(mean.human.vsnrma.plotten[gene.name,])[tissue.order.human],col=farben.human,
 cex.names=0.7,cex.axis=0.8)
 med.act=exp(median(mean.human.vsnrma.plotten[gene.name,]))
 abline(h=med.act,lty=2,col="black")

if(median==3){
 abline(h=3*med.act,lty=2,col="green")
 title(main=titel)
legend("topright",pch="-",c("median","3xmedian"),col=c("black","green"),cex=0.8)
}
if(median==5){
 abline(h=5*med.act,lty=2,col="red")
 title(main=titel)
legend("topright",pch="-",c("median","5xmedian"),col=c("black","red"),cex=0.8)
}
if(median==10){
 abline(h=10*med.act,lty=2,col="blue")
 title(main=titel)
legend("topright",pch="-",c("median","10xmedian"),col=c("black","blue"),cex=0.8)
}
if(median==20){
 abline(h=20*med.act,lty=2,col="brown")
 title(main=titel)
legend("topright",pch="-",c("median","20xmedian"),col=c("black","brown"),cex=0.8)
}
dev.off()
}

#mouse4301
plotten.mouse4301.png=function(gene.name,median){
gene.name1=substr(gene.name,1,nchar(gene.name)-3)
file.name=paste("mouse4301.",gene.name1,".png",sep="")
s=symbol.mouse[gene.name1]
titel=paste(s,gene.name1,sep="\n")

#x11(w=12,h=7)
png(file=file.name,width=1200,height=700,res=72)
par(las=2)
#1.3
par(cex=1.5)
mmi=c(2.9,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
 barplot(exp(mean4301.vsnrma[gene.name,])[tissue.order.mouse4301],col=farben.mouse4301,
 cex.names=0.6,cex.axis=0.8)
 med.act=exp(median(mean4301.vsnrma[gene.name,]))
 abline(h=med.act,lty=2,col="black")

if(median==3){
 abline(h=3*med.act,lty=2,col="green")
 title(main=titel)
legend("topright",pch="-",c("median","3xmedian"),col=c("black","green"),cex=0.8)
}
```

```
if(median==5){
 abline(h=5*med.act,lty=2,col="red")
 title(main=titel)
legend("topright",pch="-",c("median","5xmedian"),col=c("black","red"),cex=0.8)
}
if(median==10){
 abline(h=10*med.act,lty=2,col="blue")
 title(main=titel)
legend("topright",pch="-",c("median","10xmedian"),col=c("black","blue"),cex=0.8)
}
if(median==20){
 abline(h=20*med.act,lty=2,col="brown")
 title(main=titel)
legend("topright",pch="-",c("median","20xmedian"),col=c("black","brown"),cex=0.8)
}
dev.off()
}

plotten.human.roth.png=function(gene.name,median){
gene.name1=substr(gene.name,1,nchar(gene.name)-3)
file.name=paste("human.roth.",gene.name1,".png",sep="")
s=symbol.human[gene.name1]
titel=paste(s, gene.name1,sep="\n")

#w=12 ?, width=1200
#x11(w=10,h=7)
png(file=file.name,width=1000,height=700,res=72)
par(las=2)
#1.5 ?
par(cex=1.3)
mmi=c(2.9,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
#cex.names=0.6
 barplot(exp(mean.human.roth.vsnrma[gene.name,])[tissue.order.human.roth],
 col=farben.human.roth[tissue.order.human.roth],cex.names=0.7,cex.axis=0.8)
 med.act=exp(median(mean.human.roth.vsnrma[gene.name,]))
 abline(h=med.act,lty=2,col="black")

if(median==3){
 abline(h=3*med.act,lty=2,col="green")
 title(main=titel)
legend("topright",pch="-",c("median","3xmedian"),col=c("black","green"),cex=0.8)
}
if(median==5){
 abline(h=5*med.act,lty=2,col="red")
 title(main=titel)
legend("topright",pch="-",c("median","5xmedian"),col=c("black","red"),cex=0.8)
}
if(median==10){
 abline(h=10*med.act,lty=2,col="blue")
 title(main=titel)
legend("topright",pch="-",c("median","10xmedian"),col=c("black","blue"),cex=0.8)
}
if(median==20){
 abline(h=20*med.act,lty=2,col="brown")
 title(main=titel)
legend("topright",pch="-",c("median","20xmedian"),col=c("black","brown"),cex=0.8)
}
dev.off()
}

#***
```

```
#plotten.eps
#-----------
#mouse novartis
plotten.mouse.eps=function(gene.name,tissue.no){
gene.name1=substr(gene.name,1,nchar(gene.name)-3)
s=symbol.mouse[gene.name1]

file.name=paste("mouse_housekeeping_2014_",tissue.no,"_",s,".eps",sep="")
titel=paste(s, gene.name1,sep="\n")

x11(w=10,h=7)
par(las=2)
mmi=c(2.3,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
 barplot(exp(mean.mouse.vsnrma[gene.name,])[tissue.order.mouse],col=farben.mouse,
 cex.names=0.8,cex.axis=0.8)
 med.act=exp(median(mean.mouse.vsnrma[gene.name,]))
 abline(h=med.act,lty=2,col="black")
 abline(h=5*med.act,lty=5,col="red")
 title(main=titel)
legend("topright",pch="-",c("median","5xmedian"),col=c("black","red"),cex=0.8)
dev.copy2eps(file=file.name)
dev.off()
}

plotten.human.eps=function(gene.name,tissue.no){
gene.name1=substr(gene.name,1,nchar(gene.name)-3)
s=symbol.human[gene.name1]

file.name=paste("human_housekeeping_2014_",tissue.no,"_",s,".eps",sep="")
titel=paste(s,gene.name1,sep="\n")

#w=12 ?
x11(w=12,h=7)
par(las=2)
mmi=c(2.9,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
 barplot(exp(mean.human.vsnrma[gene.name,])[tissue.order.human],col=farben.human,
 cex.names=0.8,cex.axis=0.8)
 med.act=exp(median(mean.human.vsnrma[gene.name,]))
 abline(h=med.act,lty=2,col="black")
 abline(h=2*med.act,lty=2,col="green")
 title(main=titel)
legend("topright",pch="-",c("median","2xmedian"),col=c("black","green"),cex=0.8)
dev.copy2eps(file=file.name)
dev.off()
}

plotten.mouse4301.eps=function(gene.name,tissue.no){
gene.name1=substr(gene.name,1,nchar(gene.name)-3)
s=symbol.mouse[gene.name1]
file.name=paste("4302_housekeeping_2014_",tissue.no,"_",s,".eps",sep="")
titel=paste(s, gene.name1,sep="\n")

x11(w=12,h=7)
#png(file=file.name,width=1000,height=700,res=72)
par(las=2)
#par(cex=1.3)
mmi=c(2.9,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
 barplot(exp(mean4301.vsnrma[gene.name,])[tissue.order.mouse4301],col=farben.mouse4301,
```

```
 cex.names=0.8,cex.axis=0.8)
 med.act=exp(median(mean4301.vsnrma[gene.name,]))
 abline(h=med.act,lty=2,col="black")
 abline(h=2*med.act,lty=2,col="green")
 title(main=titel)
legend("topright",pch="-",c("median","2xmedian"),col=c("black","green"),cex=0.8)
dev.copy2eps(file=file.name)
dev.off()
}

plotten.human.roth.eps=function(gene.name,tissue.no){
gene.name1=substr(gene.name,1,nchar(gene.name)-3)
s=symbol.human[gene.name1]

file.name=paste("human.roth_housekeeping_",tissue.no,"_",s,".eps",sep="")
titel=paste(s, gene.name1,sep="\n")

#w=12 ?
x11(w=12,h=7)
par(las=2)
mmi=c(2.9,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
 barplot(exp(mean.human.roth.vsnrma[gene.name,])[tissue.order.human.roth],
 col=farben.human.roth[tissue.order.human.roth],cex.names=0.8,cex.axis=0.8)
 med.act=exp(median(mean.human.roth.vsnrma[gene.name,]))
 abline(h=med.act,lty=2,col="black")
 abline(h=2*med.act,lty=2,col="green")
 title(main=titel)
legend("topright",pch="-",c("median","2xmedian"),col=c("black","green"),cex=0.8)
dev.copy2eps(file=file.name)
dev.off()
}

#7.2.1 plot TRAs
#---------------
#mouse novartis
gene.name=names(tra.mouse.5x[45])
plotten.mouse(gene.name)

gene.name=names(tra.human.5x[100])
plotten.human(gene.name)

gene.name=names(tra.mouse4301.5x[100])
plotten.mouse4301(gene.name)

gene.name=names(tra.human.roth.5x[100])
plotten.human.roth(gene.name)

#7.2.2 save TRAs as png
#----------------------
#mouse novartis (3x, 5x, 10x, 20x
setwd("/home/dinkelac/data/mouse/plots/tras/tras.2014.mouse.10x")

for(i in 1:length(tra.mouse.10x)){
print(i)
plotten.mouse.png(names(tra.mouse.10x[i]),10)
}

setwd("/home/dinkelac/data/mouse/sessions/rda/")
save.image(file="tras_annotated.rda")

#human 3x, 5x, 10x, 20x
```

```
setwd("/home/dinkelac/data/mouse/plots/tras/tras.2014.human.5x/")
for(i in 1:length(tra.human.5x)){
print(i)
plotten.human.png(names(tra.human.5x[i]),5)
}

#mouse 4301 3x, 5x, 10x, 20x
setwd("/home/dinkelac/data/mouse/plots/tras/tras.2014.mouse4301.5x/")

for(i in 1:length(tra.mouse4301.5x)){
print(i)
plotten.mouse4301.png(names(tra.mouse4301.5x[i]),5)
}

#human roth
setwd("/home/dinkelac/data/mouse/plots/tras/tras.2014.human.roth.3x/")

for(i in 1:length(tra.human.roth.3x)){
print(i)
plotten.human.roth.png(names(tra.human.roth.3x[i]),3)
}

setwd("/home/dinkelac/data/mouse/plots/tras/tras.2014.human.roth.5x/")

for(i in 1:length(tra.human.roth.5x)){
print(i)
plotten.human.roth.png(names(tra.human.roth.5x[i]),5)
}

setwd("/home/dinkelac/data/mouse/plots/tras/tras.2014.human.roth.10x/")

for(i in 1:length(tra.human.roth.10x)){
print(i)
plotten.human.roth.png(names(tra.human.roth.10x[i]),10)
}

setwd("/home/dinkelac/data/mouse/plots/tras/tras.2014.human.roth.20x/")

for(i in 1:length(tra.human.roth.20x)){
print(i)
plotten.human.roth.png(names(tra.human.roth.20x[i]),20)
}

#7.2.3 save some example plots
#-----------------------------
setwd("/home/dinkelac/data/mouse/plots/tras/mouse/mouse_tras_2014_examples/")

a=sample(tra.mouse.index0,5)
for(i in 1:5){
plotten.mouse.eps(names(a[i]),"0_tissues")
}

a=sample(tra.mouse.index1,5)
for(i in 1:5){
plotten.mouse.eps(names(a[i]),"1_tissue")
}

a=sample(tra.mouse.index2,5)
for(i in 1:5){
plotten.mouse.eps(names(a[i]),"2_tissues")
}
```

```
a=sample(tra.mouse.index3,5)
for(i in 1:5){
plotten.mouse.eps(names(a[i]),"3_tissues")
}

a=sample(tra.mouse.index4,5)
for(i in 1:5){
plotten.mouse.eps(names(a[i]),"4_tissues")
}

a=sample(tra.mouse.index5,5)
for(i in 1:5){
plotten.mouse.eps(names(a[i]),"5_tissues")
}

a=sample(tra.mouse.index6,5)
for(i in 1:5){
plotten.mouse.eps(names(a[i]),"more_tissues")
}

setwd("/home/dinkelac/data/mouse/plots/tras/human/human_tras_2014_examples")

a=sample(tra.human.index0,5)
for(i in 1:5){
plotten.human.eps(names(a[i]),"0_tissues")
}

a=sample(tra.human.index1,5)
for(i in 1:5){
plotten.human.eps(names(a[i]),"1_tissue")
}

a=sample(tra.human.index2,5)
for(i in 1:5){
plotten.human.eps(names(a[i]),"2_tissues")
}

a=sample(tra.human.index3,5)
for(i in 1:5){
plotten.human.eps(names(a[i]),"3_tissues")
}

a=sample(tra.human.index4,5)
for(i in 1:5){
plotten.human.eps(names(a[i]),"4_tissues")
}

a=sample(tra.human.index5,5)
for(i in 1:5){
plotten.human.eps(names(a[i]),"5_tissues")
}

a=sample(tra.human.index6,5)
for(i in 1:5){
plotten.human.eps(names(a[i]),"more_tissues")
}

#mouse4301
setwd("/home/dinkelac/data/mouse/plots/tras/mouse_4302/mouse_4302_tras_2014_examples/")

a=sample(tra.mouse4301.index0,5)
for(i in 1:5){
```

```
plotten.mouse4301.eps(names(a[i]),".0_tissue")
}

a=sample(tra.mouse4301.index1,5)
for(i in 1:5){
plotten.mouse4301.eps(names(a[i]),".1_tissue")
}

a=sample(tra.mouse4301.index2,5)
for(i in 1:5){
plotten.mouse4301.eps(names(a[i]),".2_tissues")
}

a=sample(tra.mouse4301.index3,5)
for(i in 1:5){
plotten.mouse4301.eps(names(a[i]),".3_tissues")
}

a=sample(tra.mouse4301.index4,5)
for(i in 1:5){
plotten.mouse4301.eps(names(a[i]),".4_tissues")
}

a=sample(tra.mouse4301.index5,5)
for(i in 1:5){
plotten.mouse4301.eps(names(a[i]),".5_tissues")
}

a=sample(tra.mouse4301.index6,5)
for(i in 1:5){
plotten.mouse4301.eps(names(a[i]),".more_tissues")
}

#human roth
setwd("/home/dinkelac/data/mouse/plots/tras/human_roth/human_roth_tras_2014_examples/")

a=sample(tra.human.roth.index0,5)
for(i in 1:5){
plotten.human.roth.eps(names(a[i]),"0_tissues")
}

a=sample(tra.human.roth.index1,5)
for(i in 1:5){
plotten.human.roth.eps(names(a[i]),"1_tissue")
}

a=sample(tra.human.roth.index2,5)
for(i in 1:5){
plotten.human.roth.eps(names(a[i]),"2_tissues")
}

a=sample(tra.human.roth.index3,5)
for(i in 1:5){




      rdesktop -kde tbi-wts1 &


plotten.human.roth.eps(names(a[i]),"3_tissues")
```

```
}

a=sample(tra.human.roth.index4,5)
for(i in 1:5){
plotten.human.roth.eps(names(a[i]),"4_tissues")
}

a=sample(tra.human.roth.index5,5)
for(i in 1:5){
plotten.human.roth.eps(names(a[i]),"5_tissues")
}

a=sample(tra.human.roth.index6,5)
for(i in 1:5){
plotten.human.roth.eps(names(a[i]),"more_tissues")
}

#housekeeping genes
#------------------
#2* median in green, legend
setwd("/home/dinkelac/data/mouse/plots/tras/mouse/mouse_housekeeping_2014_examples/")

a=sample(housekeeping.mouse.2x,25)
for(i in 1:25){
plotten.mouse.eps(names(a[i]),"")
}

setwd("/home/dinkelac/data/mouse/plots/tras/human/human_housekeeping_2014_examples/")

a=sample(housekeeping.human.2x,25)
for(i in 1:25){
plotten.human.eps(names(a[i]),"")
}

setwd("/home/dinkelac/data/mouse/plots/tras/mouse_4302/
mouse_4302_housekeeping_2014_examples/")

a=sample(housekeeping.mouse4301.2x,25)
for(i in 1:25){
plotten.mouse4301.eps(names(a[i]),"")
}

setwd("/home/dinkelac/data/mouse/plots/tras/human_roth/
human_roth_housekeeping_2014_examples/")

a=sample(housekeeping.human.roth.2x,25)
for(i in 1:25){
plotten.human.roth.eps(names(a[i]),"housekeeping")
}

setwd("/home/dinkelac/data/mouse/sessions/rda/")
save.image(file="tras.rda")

#8. TRA tables
#-------------
#mouse, human, mouse4301, human.roth

#transcripts
transcript.ids=names(housekeeping.human.roth.2x)
transcript.id=substr(transcript.ids,1,nchar(transcript.ids)-3)

chromosome=as.character(ensembl.human[transcript.id,4])
```

```
startside=ensembl.human[transcript.id,5]

table.housekeeping.human.roth.2x=cbind(transcript.id,chromosome,startside)

setwd("/home/dinkelac/data/mouse/tables/")

write.table(table.housekeeping.human.roth.2x,
"housekeeping.2014.human.roth.2x.transcripts.txt",
col.names=F,row.names=F,quote=F,sep="\t")

#genes
gene.ids=ensembl.human[transcript.id,1]
gene.id=as.character(unique(gene.ids))

chromosomes=as.character(ensembl.75.human.chrom[gene.id])
startside=ensembl.75.human.start[gene.id]

table.housekeeping.human.roth.genes.2x=cbind(gene.id,chromosomes,startside)

setwd("/home/dinkelac/data/mouse/tables/")

write.table(table.housekeeping.human.roth.genes.2x,
"housekeeping.2014.human.roth.2x.genes.txt",
col.names=F,row.names=F,quote=F,sep="\t")


#8.2 chrhash of chips transcript level
#-------------------------------------
gene.names=rownames(mean.human.roth.vsnrma)
gene.ids=gene.names[63:length(gene.names)]
gene.id=substr(gene.ids,1,nchar(gene.ids)-3)

chrom=as.character(ensembl.human[gene.id,4])
start=ensembl.human[gene.id,5]

table.human.roth.chrhash.transcripts=cbind(gene.ids,chrom,start)

write.table(table.human.roth.chrhash.transcripts,"human.roth.chrhash.transcripts.txt",
col.names=F,row.names=F,quote=F,sep="\t")

#8.3 chrhash of chips on gene level
#----------------------------------

table.mouse.chrhash.genes=mouse.genes.annotated[,c(1:3)]
table.mouse.4301.chrhash.genes=mouse.4301.genes.annotated[,c(1:3)]
table.human.chrhash.genes=human.genes.annotated[,c(1:3)]
table.human.roth.chrhash.genes=human.roth.genes.annotated[,c(1:3)]

setwd("/home/dinkelac/data/mouse/tables/")
write.table(table.human.roth.chrhash.genes,"human.roth.chrhash.genes.txt",col.names=F,
row.names=F,quote=F,sep="\t")

#==> cluster analyse gehen

setwd("/home/dinkelac/data/mouse/sessions/rda/")
save.image(file="tras_annotated_lists.rda")

#annotate TRA lists for the database
#-----------------------------------
#tra table auf transcript basis
#tra table auf gene basis
#tra table fuer die datenbank
```

```
#8.4 TRA table fuer die Datenbank
#-------------------------------

transcript.ids=names(tra.mouse.5x)
transcript.id=substr(transcript.ids,1,nchar(transcript.ids)-3)
gene.id=ensembl.mouse[transcript.id,1]
chromosome=as.character(ensembl.mouse[transcript.id,4])
startside=ensembl.mouse[transcript.id,5]
strand=ensembl.mouse[transcript.id,6]
band=ensembl.mouse[transcript.id,7]
symbol=ensembl.mouse[transcript.id,8]

#8.4 TRAs annotated, transcripts
#-------------------------------
gene.ids=names(tra.index10.10x)
gene.id=substr(gene.ids,1,nchar(gene.ids)-3)

chrom=as.character(ensembl.59.mouse.transcripts[gene.id,3])
start=ensembl.59.mouse.transcripts[gene.id,4]
strand=ensembl.59.mouse.transcripts[gene.id,6]
band=as.character(ensembl.59.mouse.transcripts[gene.id,7])
symbol=as.character(ensembl.59.mouse.transcripts[gene.id,8])

table.tra.index10.10x.transcripts.annotated=cbind(gene.ids,chrom,start,strand,
band,symbol)

write.table(table.tra.index10.10x.transcripts.annotated,
"tra.gngnf1.index10.10x.transcripts.annotated.csv",sep="\t",row.names=F)

#9. Annotation on gene level
#---------------------------
#map transcripts to genes
#------------------------
deaffy = function(x) sub("_at$", "", x)

#9.1 chrhash, housekeeping genes
#-------------------------------
index.mouse=which(ensembl.75.mouse[,2]%in%deaffy(names(housekeeping.mouse.2x)))
index.human=which(ensembl.75.human[,2]%in%deaffy(names(housekeeping.human.2x)))
index.mouse4301=which(ensembl.75.mouse[,2]%in%deaffy(names(housekeeping.mouse4301.2x)))
index.human.roth=which(ensembl.75.human[,2]%in%deaffy(names(housekeeping.human.roth.2x)))

housekeeping.genes.mouse.2x=as.character(unique(ensembl.75.mouse[index.mouse,1]))
housekeeping.genes.human.2x=as.character(unique(ensembl.75.human[index.human,1]))
housekeeping.genes.mouse4301.2x=as.character(unique(ensembl.75.mouse[index.mouse4301,1]))
housekeeping.genes.human.roth.2x=as.character(unique(ensembl.75.human[index.human.roth,1]))

chrom.mouse=housekeeping.genes.mouse.2x
chrom.human=housekeeping.genes.human.2x
chrom.mouse4301=housekeeping.genes.mouse4301.2x
chrom.human.roth=housekeeping.genes.human.roth.2x

start.mouse=housekeeping.genes.mouse.2x
start.human=housekeeping.genes.human.2x
start.mouse4301=housekeeping.genes.mouse4301.2x
start.human.roth=housekeeping.genes.human.roth.2x

transcript.mouse=housekeeping.genes.mouse.2x
transcript.human=housekeeping.genes.human.2x
transcript.mouse4301=housekeeping.genes.mouse4301.2x
transcript.human.roth=housekeeping.genes.human.roth.2x
```

```
symbol.mouse=housekeeping.genes.mouse.2x
symbol.human=housekeeping.genes.human.2x
symbol.mouse4301=housekeeping.genes.mouse4301.2x
symbol.human.roth=housekeeping.genes.human.roth.2x

#mouse
for(i in 1:length(housekeeping.genes.mouse.2x)){
ind=which(ensembl.75.mouse[,1]==housekeeping.genes.mouse.2x[i])

#chrom.mouse[i]=unique(as.character(ensembl.75.mouse[ind,4]))
#start.mouse[i]=unique(ensembl.75.mouse[ind,5])
#if(length(chrom.mouse[i])>1){
#print("stop: ")
#}
transcript.mouse[i]=paste(unique(as.character(ensembl.75.mouse[ind,2])),collapse="/")
symbol.mouse[i]=paste(unique(as.character(ensembl.75.mouse[ind,8])),collapse="/")
print(i)
}

#human
for(i in 1:length(housekeeping.genes.human.2x)){
ind=which(ensembl.75.human[,1]==housekeeping.genes.human.2x[i])

#chrom.human[i]=unique(as.character(ensembl.75.human[ind,4]))
#start.human[i]=unique(ensembl.75.human[ind,5])
#if(length(chrom.human[i])>1){
#print("stop: ")
#}
transcript.human[i]=paste(unique(as.character(ensembl.75.human[ind,2])),collapse="/")
symbol.human[i]=paste(unique(as.character(ensembl.75.human[ind,8])))
print(i)
}

#mouse4301
for(i in 1:length(housekeeping.genes.mouse4301.2x)){
ind=which(ensembl.75.mouse[,1]==housekeeping.genes.mouse4301.2x[i])
#chrom.mouse4301[i]=unique(as.character(ensembl.75.mouse[ind,4]))
#start.mouse4301[i]=unique(ensembl.75.mouse[ind,5])
#if(length(chrom.mouse4301[i])>1){
#print("stop: ")
#}
transcript.mouse4301[i]=paste(unique(as.character(ensembl.75.mouse[ind,2])),collapse="/")
symbol.mouse4301[i]=paste(unique(as.character(ensembl.75.mouse[ind,8])))

print(i)
}

#human.roth
for(i in 1:length(housekeeping.genes.human.roth.2x)){
ind=which(ensembl.75.human[,1]==housekeeping.genes.human.roth.2x[i])
#chrom.human.roth[i]=unique(as.character(ensembl.75.human[ind,4]))
#start.human.roth[i]=unique(ensembl.75.human[ind,5])
#if(length(chrom.human.roth[i])>1){
#print("stop: ")
#}
transcript.human.roth[i]=paste(unique(as.character(ensembl.75.human[ind,2])),collapse="/")
symbol.human.roth[i]=paste(unique(as.character(ensembl.75.human[ind,8])))

print(i)
}
```

```
table.housekeeping.genes.2x.mouse=cbind(housekeeping.genes.mouse.2x,chrom.mouse,start.mouse)
table.housekeeping.genes.2x.human=cbind(housekeeping.genes.human.2x,chrom.human,start.human)
table.housekeeping.genes.2x.mouse4301=cbind(housekeeping.genes.mouse4301.2x,chrom.mouse4301,
start.mouse4301)
table.housekeeping.genes.2x.human.roth=cbind(housekeeping.genes.human.roth.2x,chrom.human.roth,
start.human.roth)

setwd("/home/dinkelac/data/mouse/tables/")
write.table(table.housekeeping.genes.2x.mouse,"housekeeping.genes.mouse.2x.txt")
write.table(table.housekeeping.genes.2x.human,"housekeeping.genes.human.2x.txt")
write.table(table.housekeeping.genes.2x.mouse4301,"housekeeping.genes.mouse4301.2x.txt")
write.table(table.housekeeping.genes.2x.human.roth,"housekeeping.genes.human.roth.2x.txt")

table.housekeeping.genes.2x.mouse.all=cbind(housekeeping.genes.mouse.2x,chrom.mouse,
start.mouse,transcript.mouse,symbol.mouse)
table.housekeeping.genes.2x.human.all=cbind(housekeeping.genes.human.2x,chrom.human,
start.human,transcript.human,symbol.human)
table.housekeeping.genes.2x.mouse4301.all=cbind(housekeeping.genes.mouse4301.2x,
chrom.mouse4301,start.mouse4301,transcript.mouse4301,symbol.mouse4301)
table.housekeeping.genes.2x.human.roth.all=cbind(housekeeping.genes.human.roth.2x,
chrom.human.roth,start.human.roth,transcript.human.roth,symbol.human.roth)

setwd("/home/dinkelac/data/mouse/tables/")
write.table(table.housekeeping.genes.2x.mouse.all,"housekeeping.genes.mouse.2x.annotated.txt")
write.table(table.housekeeping.genes.2x.human.all,"housekeeping.genes.human.2x.annotated.txt")
write.table(table.housekeeping.genes.2x.mouse4301.all,
"housekeeping.genes.mouse4301.2x.annotated.txt")
write.table(table.housekeeping.genes.2x.human.roth.all,
"housekeeping.genes.human.roth.2x.annotated.txt")

#9.1.1 housekeeping genes, annotated
#-------------------------------------
strand=ensembl.59.mouse.genes[housekeeping.genes.2x,6]
band=as.character(ensembl.59.mouse.genes[housekeeping.genes.2x,7])
symbol=as.character(ensembl.59.mouse.genes[housekeeping.genes.2x,8])

table.housekeeping.genes.2x.annotated=cbind(gene.ids,chrom,start,strand,band,symbol)

setwd("/home/dinkelac/data/mouse/tables/")
write.table(table.housekeeping.genes.2x,"housekeeping.genes.mouse.2x.annotated.txt")

#9.2 chrhash, tra.index10.5x, 10x genes
#---------------------------------------
genes.tra10.20x=
as.character(unique(ensembl.59.mouse.transcripts[deaffy(names(tra.index10.20x)),1]))
#3x: 6180 genes, 12033 transcripts
#5x: 4118 genes, 7735 transcripts
#10x: 2306 genes, 4202 transcripts
#20x: 1277 genes, 2207 transcripts

gene.ids=genes.tra10.10x
chrom=as.character(ensembl.59.mouse.genes[gene.ids,3])
start=ensembl.59.mouse.genes[gene.ids,4]

table.genes.tra.index10.10x=cbind(gene.ids,chrom,start)

setwd("/home/dinkelac/data/mouse/tables/")

write.table(table.genes.tra.index10.5x,"tra.index10.5x.genes.txt",col.names=F,
row.names=F,quote=F,sep="\t")
write.table(table.genes.tra.index10.10x,"tra.index10.10x.genes.txt",col.names=F,
row.names=F,quote=F,sep="\t")
```

```
#9.2.1 tra.index, genes, annotated
#---------------------------------
strand=ensembl.59.mouse.genes[gene.ids,6]
band=as.character(ensembl.59.mouse.genes[gene.ids,7])
symbol=as.character(ensembl.59.mouse.genes[gene.ids,8])

table.genes.tra.index10.10x.annotated=cbind(gene.ids,chrom,start,strand,band,symbol)

setwd("/home/dinkelac/data/mouse/tables/")

write.table(table.genes.tra.index10.10x.annotated,"tra.index10.10x.genes.annotated.csv",
col.names=F,row.names=F,quote=F,sep="\t")

#9.3 chrhash, gngnf1, genes
#--------------------------
gene.names=rownames(mean.vsnrma)
gene.ids=gene.names[65:length(gene.names)]

genes.gngnf1=as.character(unique(ensembl.59.mouse.transcripts[deaffy(gene.ids),1]))
#17121 genes, 34589 transcripts

gene.ids=genes.gngnf1
chrom=as.character(ensembl.59.mouse.genes[gene.ids,3])
start=ensembl.59.mouse.genes[gene.ids,4]

table.genes.gngnf1=cbind(gene.ids,chrom,start)

write.table(table.genes.gngnf1,"gngnf1.ensembl.59.chrhash.genes.txt",col.names=F,
row.names=F,quote=F,sep="\t")

#9.3.1 gngnf1, genes,annotated
#-----------------------------
strand=ensembl.59.mouse.genes[gene.ids,6]
band=as.character(ensembl.59.mouse.genes[gene.ids,7])
symbol=as.character(ensembl.59.mouse.genes[gene.ids,8])

table.genes.gngnf1.annotated=cbind(gene.ids,chrom,start,strand,band,symbol)

write.table(table.genes.gngnf1.annotated,"gngnf1.ensembl.59.genes.annotated.csv",
col.names=F,row.names=F,quote=F,sep="\t")

#9.4 ensembl, genes
#------------------
gene.ids=ensembl.59.genes
chrom=as.character(ensembl.59.mouse.genes[gene.ids,3])
start=ensembl.59.mouse.genes[gene.ids,4]

table.chrhash.genes.ensembl.59=cbind(gene.ids,chrom,start)

setwd("/home/dinkelac/data/mouse/tables/")
write.table(table.chrhash.genes.ensembl.59,"ensembl.59.chrhash.genes.txt",col.names=F,
row.names=F,quote=F,sep="\t")

#readin ensemble entrez and unigene IDs
#--------------------------------------

setwd("/home/dinkelac/data/mouse/tables/")

ensembl.75.mouse.entrezID=read.csv(file="ensembl.75.mouse.entrez.unigene.txt")
ensembl.75.human.entrezID=read.csv(file="ensembl.75.human.entrez.unigene.txt")
```

```
mouse.genes=unique(ensembl.75.mouse.entrezID[,1])
mouse.transcripts=unique(ensembl.75.mouse.entrezID[,2])

human.genes=unique(ensembl.75.human.entrezID[,1])
human.transcripts=unique(ensembl.75.human.entrezID[,2])

mouse.entrez=vector(len=length(mouse.genes))
mouse.unigene=vector(len=length(mouse.genes))

names(mouse.entrez)=mouse.genes
names(mouse.unigene)=mouse.genes

for(i in 1:length(mouse.genes)){
ind=which(ensembl.75.mouse.entrezID[,1]==mouse.genes[i])
mouse.entrez[i]=paste(unique(ensembl.75.mouse.entrezID[ind,3]),collapse="/")
mouse.unigene[i]=paste(unique(ensembl.75.mouse.entrezID[ind,4]),collapse="/")
print(i)
}

human.entrez=vector(len=length(human.genes))
human.unigene=vector(len=length(human.genes))

names(human.entrez)=human.genes
names(human.unigene)=human.genes

for(i in 1:length(human.genes)){
ind=which(ensembl.75.human.entrezID[,1]==human.genes[i])
human.entrez[i]=paste(unique(ensembl.75.human.entrezID[ind,3]),collapse="/")
human.unigene[i]=paste(unique(ensembl.75.human.entrezID[ind,4]),collapse="/")
print(i)
}

table.mouse.genes.entrezID.unigene=cbind(mouse.entrez,mouse.unigene)
table.human.genes.entrezID.unigene=cbind(human.entrez,human.unigene)

setwd("/home/dinkelac/data/mouse/tables/")
write.table(table.mouse.genes.entrezID.unigene,"mouse.entrez.unigene.ensembl.75.txt",
row.names=F,sep="\t")
write.table(table.human.genes.entrezID.unigene,"human.entrez.unigene.ensembl.75.txt",
row.names=F,sep="\t")

#on a transcript level
#---------------------
mouse.entrez.transcripts=vector(len=length(mouse.transcripts))
names(mouse.entrez.transcripts)=mouse.transcripts

mouse.unigene.transcripts=vector(len=length(mouse.transcripts))
names(mouse.unigene.transcripts)=mouse.transcripts

human.entrez.transcripts=vector(len=length(human.transcripts))
names(human.entrez.transcripts)=human.transcripts

human.unigene.transcripts=vector(len=length(human.transcripts))
names(human.unigene.transcripts)=human.transcripts

for(i in 1:length(mouse.transcripts)){
ind=which(ensembl.75.mouse.entrezID[,2]==mouse.transcripts[i])
mouse.entrez.transcripts[i]=paste(unique(ensembl.75.mouse.entrezID[ind,3]),collapse="/")
mouse.unigene.transcripts[i]=paste(unique(ensembl.75.mouse.entrezID[ind,4]),collapse="/")
print(i)
}
```

```
for(i in 1:length(human.transcripts)){
ind=which(ensembl.75.human.entrezID[,2]==human.transcripts[i])
human.entrez.transcripts[i]=paste(unique(ensembl.75.human.entrezID[ind,3]),collapse="/")
human.unigene.transcripts[i]=paste(unique(ensembl.75.human.entrezID[ind,4]),collapse="/")
print(i)
}

#10.2 max tissue (transcript level)
#-----------------------------------
transcript.ids=names(tra.mouse4301.3x)
transcript.id=substr(transcript.ids,1,nchar(transcript.ids)-3)

max.tissue.mouse4301.3x=list()

for(i in 1:length(transcript.ids)){
print(i)
max.value=max(mean4301.vsnrma[transcript.ids[i],])
max.tissue.mouse4301.3x[i]=names(which(mean4301.vsnrma[transcript.ids[i],]==max.value))
}

max.tissue.mouse4301.3x.transcripts=unlist(max.tissue.mouse4301.3x)
names(max.tissue.mouse4301.3x.transcripts)=transcript.id

setwd("/home/dinkelac/data/mouse/sessions/rda/")
save.image(file="tras.rda")

#mouse 3x,5x,10x,20x
#human 3x,5x,10x,20x
#mouse 4301 3x,5x,10x,20x
#human roth 3x,5x,10x,20x

#tra listen gene wise
#---------------------
tra.mouse.genes.5x=rownames(table.tra.mouse.genes.5x)
tra.human.genes.5x=rownames(table.tra.human.genes.5x)
tra.mouse4301.genes.5x=rownames(table.tra.mouse.4301.genes.5x)
tra.human.roth.genes.5x=rownames(table.tra.human.roth.genes.5x)

#10.2 max tissue (gene level)
#----------------------------

max.tissue.human.roth.genes=vector(len=length(tra.human.roth.genes.5x))
names(max.tissue.human.roth.genes)=tra.human.roth.genes.5x

for(i in 1:length(tra.human.roth.genes.5x)){
index=which(ensembl.human==tra.human.roth.genes.5x[i])
transcript.names=unique(as.character(ensembl.human[index,2]))
max.tissues=max.tissue.human.roth.transcripts[transcript.names]
max.tissue=unique(max.tissues[!is.na(max.tissues)])
max.tissue.human.roth.genes[i]=paste(max.tissue,collapse="/")
print(i)
}

#10.3 number of tissues over cutoff
#-----------------------------------
#per transcript
#--------------
#mouse
x=apply(mean.mouse.vsnrma,1,count.over.median.mouse,crit=5)
tiss.no.over.cutoff.mouse.5x.transcripts=x[tra.mouse.5x]
names(tiss.no.over.cutoff.mouse.5x.transcripts)=
deaffy(names(tiss.no.over.cutoff.mouse.5x.transcripts))
```

```
x=apply(mean.mouse.vsnrma,1,count.over.median.mouse,crit=3)
tiss.no.over.cutoff.mouse.3x.transcripts=x[tra.mouse.3x]
names(tiss.no.over.cutoff.mouse.3x.transcripts)=
deaffy(names(tiss.no.over.cutoff.mouse.3x.transcripts))

x=apply(mean.mouse.vsnrma,1,count.over.median.mouse,crit=10)
tiss.no.over.cutoff.mouse.10x.transcripts=x[tra.mouse.10x]
names(tiss.no.over.cutoff.mouse.10x.transcripts)=
deaffy(names(tiss.no.over.cutoff.mouse.10x.transcripts))

x=apply(mean.mouse.vsnrma,1,count.over.median.mouse,crit=20)
tiss.no.over.cutoff.mouse.20x.transcripts=x[tra.mouse.20x]
names(tiss.no.over.cutoff.mouse.20x.transcripts)=
deaffy(names(tiss.no.over.cutoff.mouse.20x.transcripts))

#human
x=apply(mean.human.vsnrma,1,count.over.median.human,crit=5)
tiss.no.over.cutoff.human.5x.transcripts=x[tra.human.5x]
names(tiss.no.over.cutoff.human.5x.transcripts)=
deaffy(names(tiss.no.over.cutoff.human.5x.transcripts))

x=apply(mean.human.vsnrma,1,count.over.median.human,crit=3)
tiss.no.over.cutoff.human.3x.transcripts=x[tra.human.3x]
names(tiss.no.over.cutoff.human.3x.transcripts)=
deaffy(names(tiss.no.over.cutoff.human.3x.transcripts))

x=apply(mean.human.vsnrma,1,count.over.median.human,crit=10)
tiss.no.over.cutoff.human.10x.transcripts=x[tra.human.10x]
names(tiss.no.over.cutoff.human.10x.transcripts)=
deaffy(names(tiss.no.over.cutoff.human.10x.transcripts))

x=apply(mean.human.vsnrma,1,count.over.median.human,crit=20)
tiss.no.over.cutoff.human.20x.transcripts=x[tra.human.20x]
names(tiss.no.over.cutoff.human.20x.transcripts)=
deaffy(names(tiss.no.over.cutoff.human.20x.transcripts))

#mouse 4301
x=apply(mean4301.vsnrma,1,count.over.median.mouse4301,crit=5)
tiss.no.over.cutoff.mouse4301.5x.transcripts=x[tra.mouse4301.5x]
names(tiss.no.over.cutoff.mouse4301.5x.transcripts)=
deaffy(names(tiss.no.over.cutoff.mouse4301.5x.transcripts))

x=apply(mean4301.vsnrma,1,count.over.median.mouse4301,crit=3)
tiss.no.over.cutoff.mouse4301.3x.transcripts=x[tra.mouse4301.3x]
names(tiss.no.over.cutoff.mouse4301.3x.transcripts)=
deaffy(names(tiss.no.over.cutoff.mouse4301.3x.transcripts))

x=apply(mean4301.vsnrma,1,count.over.median.mouse4301,crit=10)
tiss.no.over.cutoff.mouse4301.10x.transcripts=x[tra.mouse4301.10x]
names(tiss.no.over.cutoff.mouse4301.10x.transcripts)=
deaffy(names(tiss.no.over.cutoff.mouse4301.10x.transcripts))

x=apply(mean4301.vsnrma,1,count.over.median.mouse4301,crit=20)
tiss.no.over.cutoff.mouse4301.20x.transcripts=x[tra.mouse4301.20x]
names(tiss.no.over.cutoff.mouse4301.20x.transcripts)=
deaffy(names(tiss.no.over.cutoff.mouse4301.20x.transcripts))

#human roth
x=apply(mean.human.roth.vsnrma,1,count.over.median.human.roth,crit=5)
tiss.no.over.cutoff.human.roth.5x.transcripts=x[tra.human.roth.5x]
names(tiss.no.over.cutoff.human.roth.5x.transcripts)=
```

```
deaffy(names(tiss.no.over.cutoff.human.roth.5x.transcripts))

x=apply(mean.human.roth.vsnrma,1,count.over.median.human.roth,crit=3)
tiss.no.over.cutoff.human.roth.3x.transcripts=x[tra.human.roth.3x]
names(tiss.no.over.cutoff.human.roth.3x.transcripts)=
deaffy(names(tiss.no.over.cutoff.human.roth.3x.transcripts))

x=apply(mean.human.roth.vsnrma,1,count.over.median.human.roth,crit=10)
tiss.no.over.cutoff.human.roth.10x.transcripts=x[tra.human.roth.10x]
names(tiss.no.over.cutoff.human.roth.10x.transcripts)=
deaffy(names(tiss.no.over.cutoff.human.roth.10x.transcripts))

x=apply(mean.human.roth.vsnrma,1,count.over.median.human.roth,crit=20)
tiss.no.over.cutoff.human.roth.20x.transcripts=x[tra.human.roth.20x]
names(tiss.no.over.cutoff.human.roth.20x.transcripts)=
deaffy(names(tiss.no.over.cutoff.human.roth.20x.transcripts))

#per gene
#--------
tiss.no.over.cutoff.human.roth.genes=vector(len=length(tra.human.roth.genes.5x))
names(tiss.no.over.cutoff.human.roth.genes)=tra.human.roth.genes.5x

for(i in 1:length(tra.human.roth.genes.5x)){
index=which(ensembl.human[,1]==tra.human.roth.genes.5x[i])
transcript.names=unique(as.character(ensembl.human[index,2]))
tiss.numbers=tiss.no.over.cutoff.human.transcripts[transcript.names]
tiss.number=unique(tiss.numbers[!is.na(tiss.numbers)])
tiss.no.over.cutoff.human.roth.genes[i]=paste(tiss.number,collapse="/")
print(i)
}

#10.4 tissues over cutoff
#------------------------
#human roth
#3x

#mouse
x3.mouse=apply(mean.mouse.vsnrma,1,count.over.median.mouse,crit=3)
x5.mouse=apply(mean.mouse.vsnrma,1,count.over.median.mouse,crit=5)
x10.mouse=apply(mean.mouse.vsnrma,1,count.over.median.mouse,crit=10)
x20.mouse=apply(mean.mouse.vsnrma,1,count.over.median.mouse,crit=20)

#human
x3.human=apply(mean.human.vsnrma,1,count.over.median.human,crit=3)
x5.human=apply(mean.human.vsnrma,1,count.over.median.human,crit=5)
x10.human=apply(mean.human.vsnrma,1,count.over.median.human,crit=10)
x20.human=apply(mean.human.vsnrma,1,count.over.median.human,crit=20)

#mouse 4301
x3.mouse.4301=apply(mean4301.vsnrma,1,count.over.median.mouse4301,crit=3)
x5.mouse.4301=apply(mean4301.vsnrma,1,count.over.median.mouse4301,crit=5)
x10.mouse.4301=apply(mean4301.vsnrma,1,count.over.median.mouse4301,crit=10)
x20.mouse.4301=apply(mean4301.vsnrma,1,count.over.median.mouse4301,crit=20)

#human roth
x3.human.roth=apply(mean.human.roth.vsnrma,1,count.over.median.human.roth,crit=3)
x5.human.roth=apply(mean.human.roth.vsnrma,1,count.over.median.human.roth,crit=5)
x10.human.roth=apply(mean.human.roth.vsnrma,1,count.over.median.human.roth,crit=10)
x20.human.roth=apply(mean.human.roth.vsnrma,1,count.over.median.human.roth,crit=20)

#mouse
tiss.act=list()
```

```
transcript.ids=names(x20.mouse.4301[tra.mouse4301.20x])

for(i in 1:length(transcript.ids)){
med.act=exp(median(mean4301.vsnrma[transcript.ids[i],]))
tiss.act[i]=list(names(which(exp(mean4301.vsnrma[transcript.ids[i],])>20*med.act)))
print(i)
}

tissues.over.cutoff.mouse.4301.20x.transcripts=pasteList(tiss.act)
names(tissues.over.cutoff.mouse.4301.20x.transcripts)=deaffy(transcript.ids)

#mouse:3x,5x,10x,20x
#human: 3x,5x,10x,20x
#mouse 4301: 3x,5x,10x,20x
#human roth: 3x,5x,10x,20x

#per gene
#--------
tissues.over.cutoff.human.roth.genes=vector(len=length(tra.human.roth.genes.5x))
names(tissues.over.cutoff.human.roth.genes)=tra.human.roth.genes.5x

for(i in 1:length(tra.human.roth.genes.5x)){
index=which(ensembl.75.human==tra.human.roth.genes.5x[i])
transcript.names=unique(as.character(ensembl.75.human[index,2]))
tissues=tissues.over.cutoff.human.roth.transcripts[transcript.names]
tissue=unique(tissues[!is.na(tissues)])
tissues.over.cutoff.human.roth.genes[i]=paste(tissue,collapse="//")
print(i)
}

setwd("/home/dinkelac/data/mouse/sessions/rda/")
save.image(file="tras_annotated.rda")

#10.0 ensembl Tabellen zum annotieren vorbereiten
#------------------------------------------------

ensembl.genes=as.character(ensembl.75.mouse[,1])
ensembl.transcripts=as.character(ensembl.75.mouse[,2])
description=as.character(ensembl.75.mouse[,3])
chrom=as.character(ensembl.75.mouse[,4])
startsite=as.character(ensembl.75.mouse[,5])
strand=as.character(ensembl.75.mouse[,6])
band=as.character(ensembl.75.mouse[,7])
MGI.symbol=as.character(ensembl.75.mouse[,8])

ensembl.75.mouse.table=cbind(ensembl.genes,ensembl.transcripts,description,chrom,
startsite,strand,band,MGI.symbol)

doppelte.transcripte=
unique(ensembl.75.mouse.table[which(duplicated(ensembl.transcripts)==T),2])
raus=c()

for(i in 1:length(doppelte.transcripte)){
ind=which(ensembl.75.mouse.table[,2]==doppelte.transcripte[i])
first.entry=ind[1]

for(j in 1:length(colnames(ensembl.75.mouse.table))){
ensembl.75.mouse.table[ind,j]
a=unique(ensembl.75.mouse.table[ind,j])
b=paste(a[!is.na(a)],collapse="/")
ensembl.75.mouse.table[first.entry,j]=b
}
```

```
raus=c(raus,ind[-1])
}

ensembl.75.mouse.table.new=ensembl.75.mouse.table[-raus,]
rownames(ensembl.75.mouse.table.new)=as.character(ensembl.75.mouse.table.new[,2])

#human
ensembl.genes=as.character(ensembl.75.human[,1])
ensembl.transcripts=as.character(ensembl.75.human[,2])
description=as.character(ensembl.75.human[,3])
chrom=as.character(ensembl.75.human[,4])
startsite=as.character(ensembl.75.human[,5])
strand=as.character(ensembl.75.human[,6])
band=as.character(ensembl.75.human[,7])
MGI.symbol=as.character(ensembl.75.human[,8])

ensembl.75.human.table=cbind(ensembl.genes,ensembl.transcripts,description,chrom,
startsite,strand,band,MGI.symbol)

doppelte.transcripte=
unique(ensembl.75.human.table[which(duplicated(ensembl.transcripts)==T),2])
raus=c()

for(i in 1:length(doppelte.transcripte)){
ind=which(ensembl.75.human.table[,2]==doppelte.transcripte[i])
first.entry=ind[1]

for(j in 1:length(colnames(ensembl.75.human.table))){
ensembl.75.human.table[ind,j]
a=unique(ensembl.75.human.table[ind,j])
b=paste(a[!is.na(a)],collapse="/")
ensembl.75.human.table[first.entry,j]=b
}
raus=c(raus,ind[-1])
}

ensembl.75.human.table.new=ensembl.75.human.table[-raus,]
rownames(ensembl.75.human.table.new)=as.character(ensembl.75.human.table.new[,2])

#10.5 TRA tables annotated (transcripts)
#----------------------------------------
#Ensembl Trascript ID, Ensembl Gene ID, gene symbol, entrezID, refseq ID, unigene ID,
#chrom, startside, no. tissues over cutoff, tissues, max tissue

#mouse
ensembl.transcript=table.tra.human.roth.20x[,1]
ensembl.gene=as.character(ensembl.75.human.table.new[ensembl.transcript,1])
gene.symbol=as.character(ensembl.75.human.table.new[ensembl.transcript,8])
entrezID=human.entrez.transcripts[ensembl.transcript]
refseqID=human.unigene.transcripts[ensembl.transcript]
unigeneID=human.unigene.transcripts[ensembl.transcript]
chrom=table.tra.human.roth.20x[,2]
startsite=table.tra.human.roth.20x[,3]
tiss.number=tiss.no.over.cutoff.human.roth.20x.transcripts
tissues=tissues.over.cutoff.human.roth.20x.transcripts
max.tissue=max.tissue.human.roth.20x.transcripts

tra.human.roth.20x.table=cbind(ensembl.transcript,ensembl.gene,gene.symbol,entrezID,
refseqID,unigeneID,chrom,startsite,tiss.number,tissues,max.tissue)

#human roth
setwd("/home/dinkelac/data/mouse/tables")
```

```
write.table(tra.human.3x.table,"tra.2014.human.3x.table.csv",row.names=F,sep="\t")
write.table(tra.human.5x.table,"tra.2014.human.5x.table.csv",row.names=F,sep="\t")
write.table(tra.human.10x.table,"tra.2014.human.10x.table.csv",row.names=F,sep="\t")
write.table(tra.human.20x.table,"tra.2014.human.20x.table.csv",row.names=F,sep="\t")

write.table(tra.mouse.3x.table,"tra.2014.mouse.3x.table.csv",row.names=F,sep="\t")
write.table(tra.mouse.5x.table,"tra.2014.mouse.5x.table.csv",row.names=F,sep="\t")
write.table(tra.mouse.10x.table,"tra.2014.mouse.10x.table.csv",row.names=F,sep="\t")
write.table(tra.mouse.20x.table,"tra.2014.mouse.20x.table.csv",row.names=F,sep="\t")

write.table(tra.mouse4301.3x.table,"tra.2014.mouse.4301.3x.table.csv",row.names=F,
sep="\t")
write.table(tra.mouse4301.5x.table,"tra.2014.mouse.4301.5x.table.csv",row.names=F,
sep="\t")
write.table(tra.mouse4301.10x.table,"tra.2014.mouse.4301.10x.table.csv",row.names=F,
sep="\t")
write.table(tra.mouse4301.20x.table,"tra.2014.mouse.4301.20x.table.csv",row.names=F,
sep="\t")

write.table(tra.human.roth.3x.table,"tra.2014.human.roth.3x.table.csv",row.names=F,
sep="\t")
write.table(tra.human.roth.5x.table,"tra.2014.human.roth.5x.table.csv",row.names=F,
sep="\t")
write.table(tra.human.roth.10x.table,"tra.2014.human.roth.10x.table.csv",row.names=F,
sep="\t")
write.table(tra.human.roth.20x.table,"tra.2014.human.roth.20x.table.csv",row.names=F,
sep="\t")

#10.5 table TRA annotated + entrezID + tissues over cutoff (transcripts)
#------------------------------------------------------------------------
entrezID=gngnf1.transcripts.entrez[deaffy(names(tra.index10.5x))]
tiss.no.over.cutoff=tiss.no.over.cutoff.transcripts
max.tissue=max.tissue.transcripts

table.transcripts.tra.index10.5x.annotated=cbind(table.tra.index10.5x.transcripts.annotated,
entrezID,tiss.no.over.cutoff,max.tissue,tissues.over.cutoff.transcripts)

setwd("/home/dinkelac/data/mouse/tables")
write.table(table.transcripts.tra.index10.5x.annotated,
"tra.gngnf1.index2.10x.transcripts.annotated.tissues.csv",row.names=F,sep="\t")

#10.6 table TRA annotated + entrezID + tissues over cutoff (genes)
#-----------------------------------------------------------------
entrezID=gngnf1.genes.entrez[genes.tra10.5x]
tiss.no.over.cutoff=tiss.no.over.cutoff.genes
max.tissue=max.tissue.genes
tissues.over.cutoff=tissues.over.cutoff.genes

table.genes.tra.index10.5x.entrez.annotated=cbind(table.genes.tra.index10.5x.annotated,
entrezID,tiss.no.over.cutoff,max.tissue,tissues.over.cutoff)

setwd("/home/dinkelac/data/mouse/tables/")
write.table(table.genes.tra.index10.5x.entrez.annotated,
"tra.human.index10.5x.genes.annotated.tissues.csv",row.names=F,sep="\t")

setwd("/home/dinkelac/data/mouse/sessions/rda")
save.image(file="tras.gngnf1.rda")

#Function: pasteList
#-------------------
pasteList=function(list){
```

```
result=list()
        for(i in 1:length(list)){
        tiss=unlist(list[i])
        res=tiss[1]
        if(length(tiss)>1){
                for(j in 2:length(tiss)){
                res=paste(res,tiss[j], sep="/")
                }
        }
        result[i]=res
        result=unlist(result)
        }
return(result)
}

#next skript: calc_clustering_gngnf1.R

#comparison to other databases
#------------------------------
setwd("/home/dinkelac/data/tras_tiger/")
tiger_table_refseq=read.csv(file="tissue_specific_genes_refseq.txt",sep="\t",header=F)
tiger_table_unigene=read.csv(file="Tissue_specific_genes_unigene.txt",sep="\t",header=F)

tiger_tras_refseq_all=sort(unique(as.character(tiger_table[,1])))
tiger_tras_unigene_all=sort(unique(as.character(tiger_table_unigene[,1])))
tiger_tras_refseq=tiger_tras_refseq_all[-c(1:8,6590:6601)]
tiger_tras_unigene=tiger_tras_unigene_all[-c(1:2,6701:6718)]

setwd("/home/dinkelac/data/mouse/tables/")
human.refseq.table=read.csv(file="ensembl.75.human.refseq_unigene.txt")
mouse.refseq.table=read.csv(file="ensembl.75.mouse.refseq_unigene.txt")

human.unigene=as.character(human.refseq.table[,3])
names(human.unigene)=human.refseq.table[,2]

human.refseq_NM=as.character(human.refseq.table[,4])
names(human.refseq_NM)=human.refseq.table[,2]

human.refseq_XM=as.character(human.refseq.table[,5])
names(human.refseq_XM)=human.refseq.table[,2]

mouse.unigene=as.character(mouse.refseq.table[,3])
names(mouse.unigene)=mouse.refseq.table[,2]

human.tras=deaffy(names(tra.human.5x))

for(i in 1:length(human.tras)){
ind=which(names(human.unigene)==human.tras[i])
index=unique(c(index,ind))
}

human.tras.unigene.new=unique(human.unigene[index])
human.tras.unigene=unique(as.character(human.unigene[human.tras]))

for(i in 1:length(human.tras)){
ind=which(names(human.refseq_NM)==human.tras[i])
index=unique(c(index,ind))
}

human.tras.refseq.new=unique(human.refseq_NM[index])
human.tras.refseq_NM=unique(as.character(human.refseq_NM[human.tras]))
```

```
#1. pie charts
#-------------
#gtex data
#---------

setwd("/home/dinkelac/data/mouse/sessions/rda/")
load("mean.data.rda")

tissues.mouse.novartis=dimnames(exp(mean.mouse.vsnrma))[[2]]
# colors.mouse.novartis=as.character(farben.gngnf1[,4])
tissues.mouse.lattin=dimnames(exp(mean4301.vsnrma))[[2]]
tissues.human.roth=unique(tissue.names.human.roth)
tissues.human.novartis=tissue.names.human
tissues.human.gtex=unique(gtex.tissue)

x11(w=10,h=8)
slices <- rep(2,70)

pie(slices, labels = tissues.lattin[tissue.order.mouse4301],
main="70 Tissue Types mouse Lattin data",col=farben.mouse4301,cex=0.7)

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="piechart.tissue.types.mouse.lattin.data.eps")

#tissue overlap in different datasets
#-------------------------------------
tissues.mouse.novartis=sort(tissues.mouse.novartis)
#61
tissues.mouse.lattin=sort(tissues.mouse.lattin)
#91
tissues.human.roth=sort(tissues.human.roth)
#81
tissues.human.novartis=sort(tissues.human.novartis)
#157/2 (_1, _2)
tissues.human.gtex=sort(tissues.human.gtex)
#55
tissues.mouse.novartis[1]="adipose_tissue"
tissues.mouse.novartis[2]="adrenal_gland"
tissues.mouse.novartis[4]="B_cells"
tissues.mouse.novartis[8]="bone_marrow"
tissues.mouse.novartis[17:21]="embryos"
tissues.mouse.novartis[15]="dorsal_root_ganglion"
tissues.mouse.novartis[16]="dorsal_striatum"
tissues.mouse.novartis[24]="frontal_cortex"
tissues.mouse.novartis[29]="intestine_large"
tissues.mouse.novartis[32]="lymph_nodes"
tissues.mouse.novartis[33]="mammary_gland"
tissues.mouse.novartis[34]="medialolfactoryepithelium"
tissues.mouse.novartis[35]="olfactory_bulb"
tissues.mouse.novartis[44]="salivary_gland"
tissues.mouse.novartis[45]="skeletal_muscle"
tissues.mouse.novartis[48:49]="spinal_cord"
tissues.mouse.novartis[52]="substantia_nigra"
tissues.mouse.novartis[56]="tongue"
tissues.mouse.novartis[59]="umbilical_cord"
tissues.mouse.novartis[61]="vomeralnasalorgan"
#neu
tissues.human.novartis=unique(tissues.human.novartis)

tissues.human.novartis[1:2]="B_lymphoblasts"
tissues.human.novartis[3:4]="adipocyte"
tissues.human.novartis[5:6]="adrenal_cortex"
```

```
tissues.human.novartis[7:8]="adrenal_gland"
tissues.human.novartis[9:10]="amygdala"
tissues.human.novartis[11:12]="appendix"
tissues.human.novartis[13:14]="AtrioventricularNode"
tissues.human.novartis[15:22]="BM_cells"
tissues.human.novartis[23:24]="bone_marrow"
tissues.human.novartis[25:26]="bronchial_epithelial_cells"
tissues.human.novartis[27:28]="Cardiac_myocytes"
tissues.human.novartis[29:30]="caudate_nucleus"
tissues.human.novartis[31:32]="cerebellum"
tissues.human.novartis[33:34]="cerebellum_peduncles"
tissues.human.novartis[35:36]="ciliary_ganglion"
tissues.human.novartis[37:38]="cingulate_cortex"
tissues.human.novartis[39:40]="colorectal_adenocarcinoma"
tissues.human.novartis[41:42]="dorsal_root_ganglion"
tissues.human.novartis[43:50]="embryos"
tissues.human.novartis[51:52]="globus_pallidus"
tissues.human.novartis[53:54]="heart"
tissues.human.novartis[55:56]="hypothalamus"
tissues.human.novartis[57:58]="kidney"
tissues.human.novartis[59:64]="leukemia_cells"
tissues.human.novartis[65:66]="liver"
tissues.human.novartis[67:68]="lung"
tissues.human.novartis[69:70]="lymphnode"
tissues.human.novartis[71:74]="lymphoma"
tissues.human.novartis[75:76]="medulla_oblongata"
tissues.human.novartis[77:78]="occipital_lobe"
tissues.human.novartis[79:80]="olfactory_bulb"
tissues.human.novartis[81:82]="ovary"
tissues.human.novartis[83:84]="pancreas"
tissues.human.novartis[85:86]="pancreatic_islets"
tissues.human.novartis[87:88]="parietal_lobe"
tissues.human.novartis[89:90]="dendritic_cells"
tissues.human.novartis[91:92]="monocytes"
tissues.human.novartis[93:94]="B_cells"
tissues.human.novartis[95:96]="CD4+T_cells"
tissues.human.novartis[97:98]="NK_cells"
tissues.human.novartis[99:100]="CD8+T_cells"
tissues.human.novartis[101:102]="pituitary"
tissues.human.novartis[103:104]="placenta"
tissues.human.novartis[105:106]="pons"
tissues.human.novartis[107:108]="prefrontal_cortex"
tissues.human.novartis[109:110]="prostate"
tissues.human.novartis[111:112]="salivary_gland"
tissues.human.novartis[113:114]="skeletal_muscle"
tissues.human.novartis[115:116]="skin"
tissues.human.novartis[117:118]="smooth_muscle"
tissues.human.novartis[119:120]="spinal_cord"
tissues.human.novartis[121:122]="subthalamic_nucleus"
tissues.human.novartis[123:124]="superior_cervical_ganglion"
tissues.human.novartis[125:126]="temporal_lobe"
tissues.human.novartis[127:136]="testis"
tissues.human.novartis[137:138]="thalamus"
tissues.human.novartis[139:140]="thymus"
tissues.human.novartis[141:142]="thyroid"
tissues.human.novartis[143:144]="tongue"
tissues.human.novartis[145:146]="tonsil"
tissues.human.novartis[147:148]="trachea"
tissues.human.novartis[149:150]="trigeminal_ganglion"
tissues.human.novartis[151:154]="uterus"
tissues.human.novartis[155:156]="WholeBlood"
tissues.human.novartis[157:158]="WholeBrain"
```

```
tissues=sort(unique(c(tissues.mouse.novartis,tissues.mouse.lattin,tissues.human.roth,
tissues.human.gtex)))

tissues.mouse.lattin[2:3]="adipose_tissue"
tissues.mouse.lattin[83]="cd4+Tcell"
tissues.mouse.lattin[84]="cd8+Tcell"
tissues.mouse.lattin[87:89]="thymus"
tissues.mouse.lattin[65:67]="osteoblasts"
tissues.mouse.lattin[43:49]="macrophages"
tissues.mouse.lattin[50:51]="mammary_gland"
tissues.mouse.lattin[76]="retina"
tissues.mouse.lattin[53:55]="mast_cells"
tissues.mouse.lattin[7]="B_cells"
tissues.mouse.lattin[15]="cerebral_cortex"
tissues.mouse.lattin[19:21]="dendritic_cells"
tissues.mouse.lattin[22]="dorsal_root_ganglion"
tissues.mouse.lattin[24:25]="embryos"
tissues.mouse.lattin[29:30]="granulocytes"

#neu
tissues.human.roth[1]="adipose_tissue"
tissues.human.roth[9]="adrenal_gland"
tissues.human.roth[11]="dorsal_root_ganglion"
tissues.human.roth[16]="cortex"
tissues.human.roth[17:18]="heart"
tissues.human.roth[23]="kidney"
tissues.human.roth[45]="prostate"
tissues.human.roth[44]="pituitary"
tissues.human.roth[45]="intestine_small"
tissues.human.roth[47:48]="stomach"
tissues.human.roth[56]="thyroid"
tissues.human.roth[61]="tongue"
tissues.human.roth[59]="trigeminal"
tissues.human.roth[8]="adipose_tissue"
tissues.human.roth[15]="colon"
tissues.human.roth[31]="kidney"
tissues.human.roth[55]="testis"
tissues.human.roth[40]="stomach"
tissues.human.roth[54]="tongue"
tissues.human.gtex[55]="vagina"
tissues.human.gtex[52]="colon"
tissues.human.gtex[51]="small_intestine"
tissues.human.gtex[50]="adipose_tissue"
tissues.human.gtex[49]="esophagus"
tissues.human.gtex[48]="heart"
tissues.human.gtex[47]="skin"
tissues.human.gtex[45:46]="cervix"
tissues.human.gtex[44]="bladder"
tissues.human.gtex[42]="mammary_gland"
tissues.human.gtex[41]="ovary"
tissues.human.gtex[39]="liver"
tissues.human.gtex[38]="spleen"
tissues.human.gtex[37]="artery"
tissues.human.gtex[36]="spinal_cord"
tissues.human.gtex[35]="prostate"
tissues.human.gtex[34]="colon"
tissues.human.gtex[33]="adrenal_gland"
tissues.human.gtex[32]="stomach"
tissues.human.gtex[31]="vagina"
tissues.human.gtex[30]="pancreas"
tissues.human.gtex[29]="artery"
```

```
tissues.human.gtex[28]="uterus"
tissues.human.gtex[27]="pituitary"
tissues.human.gtex[26]="cerebellum"
tissues.human.gtex[25]="cortex"
tissues.human.gtex[24]="thyroid"
tissues.human.gtex[23]="kidney"
tissues.human.gtex[21:22]="esophagus"
tissues.human.gtex[20]="lung"
tissues.human.gtex[19]="heart"
tissues.human.gtex[17:18]="tibial"
tissues.human.gtex[16]="skeletal_muscle"
tissues.human.gtex[15]="adipose_tissue"
tissues.human.gtex[14]="skin"
tissues.human.gtex[13]="testis"
tissues.human.gtex[12]="hypothalamus"
tissues.human.gtex[11]="putamen"
tissues.human.gtex[10]="nucleus_accumbens"
tissues.human.gtex[8]="amygdala"
tissues.human.gtex[6]="substantia_nigra"
tissues.human.gtex[5]="hippocampus"
tissues.human.gtex[3]="frontal_cortex"
tissues.human.gtex[4]="brain"
tissues.human.gtex[7]="brain"
tissues.human.gtex[9]="brain"
tissues.human.gtex[43]="EBV_lymphocytes"
```

```
      rdesktop -kde tbi-wts1 &
```

```
tissues.human.gtex[54]="cml"
tissues.human.gtex[2]="fibroblasts"

n1=tissues.mouse.novartis
n2=tissues.human.novartis
n3=tissues.mouse.lattin
n4=tissues.human.roth
n5=tissues.human.gtex

library(VennDiagram)
x11(h=7,w=10)

draw.quintuple.venn(
              area1 = length(n1),
              area2 = length(n2),
              area3 = length(n3),
              area4 = length(n4),
              area5 = length(n5),

              n12 = length(intersect(n1,n2)),
              n13 = length(intersect(n1,n3)),
              n14 = length(intersect(n1,n4)),
              n15 = length(intersect(n1,n5)),
              n23 = length(intersect(n2,n3)),
              n24 = length(intersect(n2,n4)),
              n25 = length(intersect(n2,n5)),
              n34 = length(intersect(n3,n4)),
              n35 = length(intersect(n3,n5)),
              n45 = length(intersect(n4,n5)),
```

```
                n123 = length(intersect(intersect(n1,n2),n3)),
                n124 = length(intersect(intersect(n1,n2),n4)),
                n125 = length(intersect(intersect(n1,n2),n5)),
                n134 = length(intersect(intersect(n1,n3),n4)),
                n135 = length(intersect(intersect(n1,n3),n5)),
                n145 = length(intersect(intersect(n1,n4),n5)),
                n234 = length(intersect(intersect(n2,n3),n4)),
                n235 = length(intersect(intersect(n2,n3),n5)),
                n245 = length(intersect(intersect(n2,n4),n5)),
                n345 = length(intersect(intersect(n3,n4),n5)),

                n1234 = length(intersect(intersect(intersect(n1,n2),n3),n4)),
                n1235 = length(intersect(intersect(intersect(n1,n2),n3),n5)),
                n1245 = length(intersect(intersect(intersect(n1,n2),n4),n5)),
                n1345 = length(intersect(intersect(intersect(n1,n3),n4),n5)),
                n2345 = length(intersect(intersect(intersect(n2,n3),n4),n5)),
                n12345 = length(intersect(intersect(intersect(intersect(n1,n2),
                n3),n4),n5)),

                category = c("mouse Novartis", "human Novartis", "mouse Lattin",
                "human Roth","human GTEX"),
                fill = c("dodgerblue", "goldenrod1", "darkorange1", "seagreen3", "orchid3"),
                cat.col = c("dodgerblue", "goldenrod1", "darkorange1", "seagreen3", "orchid3"),
                cat.cex = 1,
                margin = 0.05,

                cex = c(1.5, 1.5, 1.5, 1.5, 1.5, 1, 0.8, 1, 0.8, 1, 0.8, 1, 0.8, 1, 0.8,
                1, 0.55, 1, 0.55, 1, 0.55, 1, 0.55, 1, 0.55, 1, 1, 1, 1, 1, 1.5),
                ind = TRUE, main="Tissue types in all datasets"
                )

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="tissue.overlap.all.datasets.eps")

install.packages("Cairo")
library(Cairo)
cairo_ps("test.eps", family = "Times")
plot(rnorm(100))
dev.off()

#TRAs
#----
setwd("/home/dinkelac/data/mouse/tables/")

tras.human.3x.novartis=read.table("tra.2014.human.3x.table.csv")
tras.human.5x.novartis=read.table("tra.2014.human.5x.table.csv")
tras.human.10x.novartis=read.table("tra.2014.human.10x.table.csv")
tras.human.20x.novartis=read.table("tra.2014.human.20x.table.csv")

tras.mouse.3x.novartis=read.table("tra.2014.mouse.3x.table.csv")
tras.mouse.5x.novartis=read.table("tra.2014.mouse.5x.table.csv")
tras.mouse.10x.novartis=read.table("tra.2014.mouse.10x.table.csv")
tras.mouse.20x.novartis=read.table("tra.2014.mouse.20x.table.csv")

tras.human.roth.3x=read.table("tra.2014.human.roth.3x.table.csv")
tras.human.roth.5x=read.table("tra.2014.human.roth.5x.table.csv")
tras.human.roth.10x=read.table("tra.2014.human.roth.10x.table.csv")
tras.human.roth.20x=read.table("tra.2014.human.roth.20x.table.csv")

tras.mouse.lattin.3x=read.table("tra.2014.mouse.4301.3x.table.csv")
tras.mouse.lattin.5x=read.table("tra.2014.mouse.4301.5x.table.csv")
```

```
tras.mouse.lattin.10x=read.table("tra.2014.mouse.4301.10x.table.csv")
tras.mouse.lattin.20x=read.table("tra.2014.mouse.4301.20x.table.csv")

tras.human.gtex.3x=read.table("tra.2017.human.gtex.3x.table.csv")
tras.human.gtex.5x=read.table("tra.2017.human.gtex.3x.table.csv")
tras.human.gtex.10x=read.table("tra.2017.human.gtex.3x.table.csv")
tras.human.gtex.20x=read.table("tra.2017.human.gtex.3x.table.csv")

tras.human.gtex.3x.eight=read.table("tra.2017.human.gtex.3x.table.eight.csv")
tras.human.gtex.5x.eight=read.table("tra.2017.human.gtex.3x.table.eight.csv")
tras.human.gtex.10x.eight=read.table("tra.2017.human.gtex.3x.table.eight.csv")
tras.human.gtex.20x.eight=read.table("tra.2017.human.gtex.3x.table.eight.csv")

#tras (genes)
human.novartis.3x=unique(as.character(tras.human.3x.novartis[,2]))
human.novartis.5x=unique(as.character(tras.human.5x.novartis[,2]))
human.novartis.10x=unique(as.character(tras.human.10x.novartis[,2]))
human.novartis.20x=unique(as.character(tras.human.20x.novartis[,2]))

mouse.novartis.3x=unique(as.character(tras.mouse.3x.novartis[,2]))
mouse.novartis.5x=unique(as.character(tras.mouse.5x.novartis[,2]))
mouse.novartis.10x=unique(as.character(tras.mouse.10x.novartis[,2]))
mouse.novartis.20x=unique(as.character(tras.mouse.20x.novartis[,2]))

human.roth.3x=unique(as.character(tras.human.roth.3x[,2]))
human.roth.5x=unique(as.character(tras.human.roth.5x[,2]))
human.roth.10x=unique(as.character(tras.human.roth.10x[,2]))
human.roth.20x=unique(as.character(tras.human.roth.20x[,2]))

mouse.lattin.3x=unique(as.character(tras.mouse.lattin.3x[,2]))
mouse.lattin.5x=unique(as.character(tras.mouse.lattin.5x[,2]))
mouse.lattin.10x=unique(as.character(tras.mouse.lattin.10x[,2]))
mouse.lattin.20x=unique(as.character(tras.mouse.lattin.20x[,2]))

human.gtex.3x=unique(as.character(tras.human.gtex.3x[,2]))
human.gtex.5x=unique(as.character(tras.human.gtex.5x[,2]))
human.gtex.10x=unique(as.character(tras.human.gtex.10x[,2]))
human.gtex.20x=unique(as.character(tras.human.gtex.20x[,2]))

human.gtex.3x.eight=unique(as.character(tras.human.gtex.3x.eight[,2]))
human.gtex.5x.eight=unique(as.character(tras.human.gtex.5x.eight[,2]))
human.gtex.10x.eight=unique(as.character(tras.human.gtex.10x.eight[,2]))
human.gtex.20x.eight=unique(as.character(tras.human.gtex.20x.eight[,2]))

n1=human.novartis.5x
n2=human.gtex.5x
n3=intersect(n1,n2)

venn.plot <- draw.pairwise.venn(length(n1),length(n2),length(n3), c("human Novartis",
"human GTEX"),col=c("red","blue"));
grid.newpage();

a1=human.novartis.5x
a2=human.roth.5x
a3=human.gtex.5x

a4=intersect(a1,a2)
a5=intersect(a2,a3)
a6=intersect(a1,a3)
a7=intersect(a4,a3)

venn.plot <- draw.triple.venn(length(a1),length(a2),length(a3),length(a4),length(a5),
```

```
length(a6),length(a7),c("human Novartis","human Roth","human GTEX"),col=c("red","blue",
"green"))
grid.newpage();

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="venn.diagram.tras.human.novartis.3x.human.roth.3x.eps")

library(Cairo)
cairo_ps("venn.tras.5x.mouse.eps", family = "Times")

venn.plot <- draw.pairwise.venn(length(n1),length(n2),length(n3), c("mouse Novartis",
"mouse Lattin"),col=c("red","blue"));
#venn.plot <- draw.triple.venn(length(a1),length(a2),length(a3),length(a4),length(a5),
#length(a6),length(a7),c("human Novartis","human Roth","human GTEX"),col=c("red","blue",
"green"))

dev.off()

#Tissues in TRA data
#--------------------
farben.human=c("darkred","red3","tomato2","#FF5600FF","wheat","wheat","wheat",
"green","green","green","green","green","green","green","green","green","green","green",
"green","green","green","green","green","green","green","green","seagreen3","seagreen3",
"seagreen3","seagreen3","white","white","white","white","#FF8100FF","orange","tan","yellow",
"oldlace","orange","oldlace","oldlace","oldlace","#00B4FFFF","#00B4FFFF","oldlace","oldlace",
"oldlace","oldlace","oldlace","oldlace","white","white","white","white","white","white",
"lightblue","steelblue","#005DFFFF","navy","navy","#0007FFFF","#3900FFFF","slateblue",
"slateblue","rosybrown","rosybrown","purple","purple","purple","purple","purple","thistle",
"#BB00FFFF","violet","oldlace","darkblue","darkblue")

tra.symbols.5x.human.novartis=as.character(sort(unique(tras.human.5x.novartis[,3])))
tra.symbols.5x.human.GTEX=as.character(sort(unique(tras.human.gtex.5x[,3])))
tra.symbols.5x.mouse.novartis=as.character(sort(unique(tras.mouse.5x.novartis[,3])))

tras.all.human.symbols=
sort(unique(as.character(tras.human.5x.novartis[which(tras.human.5x.novartis[,2]%in%a7),3])))
max.tiss.5x.human.novartis=as.character((tras.human.5x.novartis[,11]))
human.novartis.tissues=unique(sort(max.tiss.5x.human.novartis))
hist.tiss.novartis=vector(len=length(human.novartis.tissues))
names(hist.tiss.novartis)=human.novartis.tissues

for(i in 1:length(hist.tiss.novartis)){
hist.tiss.novartis[i]=length(which(max.tiss.5x.human.novartis==
human.novartis.tissues[i]))
}

max.tiss.5x.mouse.novartis=as.character((tras.mouse.5x.novartis[,11]))
mouse.novartis.tissues=unique(sort(max.tiss.5x.mouse.novartis))
hist.tiss.novartis.mouse=vector(len=length(mouse.novartis.tissues))
names(hist.tiss.novartis.mouse)=mouse.novartis.tissues

for(i in 1:length(hist.tiss.novartis.mouse)){
hist.tiss.novartis.mouse[i]=length(which(max.tiss.5x.mouse.novartis==
mouse.novartis.tissues[i]))
}

max.tiss.5x.human.roth=as.character((tras.human.roth.5x[,11]))
human.roth.tissues=unique(sort(max.tiss.5x.human.roth))
hist.tiss.roth=vector(len=length(human.roth.tissues))
names(hist.tiss.roth)=human.roth.tissues

for(i in 1:length(hist.tiss.roth)){
```

```
hist.tiss.roth[i]=length(which(max.tiss.5x.human.roth==human.roth.tissues[i]))
}

max.tiss.5x.mouse.lattin=as.character((tras.mouse.lattin.5x[,11]))
mouse.lattin.tissues=unique(sort(max.tiss.5x.mouse.lattin))
hist.tiss.lattin.mouse=vector(len=length(mouse.lattin.tissues))
names(hist.tiss.lattin.mouse)=mouse.lattin.tissues

for(i in 1:length(hist.tiss.lattin.mouse)){
hist.tiss.lattin.mouse[i]=length(which(max.tiss.5x.mouse.lattin==
mouse.lattin.tissues[i]))
}

max.tiss.5x.gtex=as.character((tras.human.gtex.5x[,10]))
gtex.tissues=unique(sort(max.tiss.5x.gtex))
hist.tiss.gtex=vector(len=length(gtex.tissues))
names(hist.tiss.gtex)=gtex.tissues

for(i in 1:length(hist.tiss.gtex)){
hist.tiss.gtex[i]=length(which(max.tiss.5x.gtex==gtex.tissues[i]))
}

immune.cells=rep("white",61)
immune.cells[c(1,6:10,28,54,57,60)]="red"
immune.cells[c(1,6:9,34:39)]="wheat"
immune.cells[c(3:5,13:15,29,30,42,61)]="green"
immune.cells[c(49:52)]="blue"

col.mouse=rep("white",61)
col.mouse[c(4,10,11)]="wheat"
col.mouse[c(8,6,31,53)]="red"
col.mouse[c(1:3,12,13,15,16,23,25,26,32,43,48,51,57)]="green"

col.gtex=rep("white",53)
col.gtex[c(8:20,39)]="green"
col.gtex[53]="red"
col.gtex[c()]="wheat"

x11(h=7,w=10)
par(las=2)
par(mai=c(2.3,1,1,1))

setwd("/home/dinkelac/data/mouse/plots")

barplot(hist.tiss.novartis.mouse,cex.names=0.7,col="orange",
main="TRAs per tissue type in the mouse Novartis data")
abline(h=c(0,200,400,600,800,1000,1200))

dev.copy2eps(file="tras.per.tissue.type.mouse.novartis.data.eps")

barplot(hist.tiss.novartis,cex.names=0.7,col="navy",
main="TRAs per tissue type in the human Novartis data")

dev.copy2eps(file="tras.per.tissue.type.human.novartis.data.eps")

#abline(h=c(0,200,400,600,800,1000,1200))
barplot(hist.tiss.roth,cex.names=0.7,col="red3",
main="TRAs per tissue type in the human Roth data")
abline(h=c(0,1000,2000,3000,4000))

dev.copy2eps(file="tras.per.tissue.type.human.roth.data.red.eps")
```

```
barplot(hist.tiss.lattin.mouse,cex.names=0.7,col="yellow2",
main="TRAs per tissue type in the mouse Lattin data")
abline(h=c(0,200,400,600,800,1000,1200))

dev.copy2eps(file="tras.per.tissue.type.mouse.lattin.data.orange.eps")

barplot(hist.tiss.gtex,cex.names=0.7,col="thistle",
main="TRAs per tissue type in the GTEX data")
abline(h=c(0,5000,10000,15000,20000))

dev.copy2eps(file="tras.per.tissue.type.human.gtex.data.thistle.eps")

tra.col=rep("black",length(tra.symbols.5x.human.novartis))

#HLA (MHC molekuele, auf Immunzellen hauptsaechlich)
tra.col[687:708]="red"
#Serpin (protease inhibitoren)
tra.col[1382:1396]="green"
#cxcl (chemokine)
tra.col[401:410]="blue"
#slc (Carrier, Transport)
tra.col[1409:1439]="red"
#CD (Antigene)
tra.col[228:255]="red"
#S100
tra.col[1339:1347]="red"

#GTEX daten

#1800 A
#1700 B
#2000 C
#5000 D

x11(h=12,w=18)
x11(h=10,w=14)
par(mar=c(1,1,0.3,1))
plot(0,col="white",axes=F,xlab="",ylab="")
#ncol=18
legend("bottomleft",tras.all.human.symbols,ncol=15,cex=0.55,text.col=tra.col)
title(main="\n\n\nTissue restricted antigens (TRAs) in all human datasets")
grid.newpage();

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="tras.per.tissue.type.gtex.data.eps")

col.mouse=rep("white",62)
col.mouse[c(8,16,17,26)]="red"
col.mouse[c(23,41,42)]="wheat"
col.mouse[c(2,7,11,12,15,19,22,29,36,38,39,50,51,54,56,59,61)]="green"

col.human.novartis=rep("white",60)
col.human.novartis[c(60,58,53,50,49,39,38,36,34,32,29)]="green"
col.human.novartis[c()]="wheat"
col.human.novartis[c()]="red"

x11(w=10,h=8)
slices <- sort(hist.tiss.novartis[-c(33,62)])

pie(slices,col=col.human.novartis,labels = names(sort(hist.tiss.novartis[-c(33,62)])),
main="Tissue types of TRAs in the human Novartis data",cex=0.7,init.angle=90)
```

```
dev.copy2eps(file="piechart.tras.per.tissue.type.human.novartis.data.eps")

symbols.human.novartis.5x=tras.human.5x.novartis[,3]
genes.human.novartis.5x=tras.human.5x.novartis[,2]
chrom.human.novartis.5x=tras.human.5x.novartis[,7]
names(chrom.human.novartis.5x)=symbols.human.novartis.5x
names(symbols.human.novartis.5x)=chrom.human.novartis.5x

chromosomen=c(1:24)
chromosomen[23]="X"
chromosomen[24]="Y"
chromosomen.human=as.character(sort(unique(chrom.human.novartis.5x)))[-23]
chromosomen.human.sort=chromosomen.human[c(1,12,16:22,2:11,13:15,80,81,79,72:78,
56:71,23:55)]

hist.chrom=vector(len=81)
names(hist.chrom)=chromosomen.human.sort

hist.chrom.all=vector(len=81)
names(hist.chrom.all)=chromosomen.human.sort

ensembl.75.human.genes.table=read.csv(file="ensembl.75.human.genes.csv")
ensembl.75.human.genes=ensembl.75.human.genes.table[,1]
ensembl.75.human.chrom=ensembl.75.human.genes.table[,4]

names(ensembl.75.human.chrom)=ensembl.75.human.genes

for(i in 1:81){
hist.chrom.all[i]=length(which(ensembl.75.human.chrom==chromosomen[i]))
}

for(i in 1:81){
hist.chrom[i]=length(which(chrom.human.novartis.5x==chromosomen[i]))
}

hist.all=c(hist.chrom.all,hist.chrom)[c(1,25,2,26,3,27,4,28,5,29,6,30,7,31,8,32,9,
33,10,34,11,35,12,36,13,37,14,38,15,39,16,40,17,41,18,42,19,43,20,44,21,45,22,46,23,
47,24,48)]

x11(h=8,w=10)
barplot(hist.chrom.all,col="grey")
barplot(hist.chrom[c(1:24)],col="red")
abline(h=c(100,200,300,400,500,600),col="lightgrey")
title(main="TRAs per Chromosome in the human Novartis Data")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="tras.per.chrom.novartis.human.eps")

barplot(hist.all,col=c("grey","red"),legend.text=c("background genes","TRAs"))
abline(h=c(1000,2000,3000,4000,5000),col="lightgrey")
title(main="TRAs per Chromosome in the human Novartis Data\n
in comparison to background genes")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="tras.per.chrom.novartis.human.with.background.eps")

table.human.novartis=tras.human.5x.novartis[,c(3,7,8,11)]
table.human.novartis.ohne.doppelte=table.human.novartis[!duplicated(table.human.novartis),]
table.human.new=
table.human.novartis.ohne.doppelte[order(table.human.novartis.ohne.doppelte$V3),,]

x=paste(table.human.novartis[,1],as.character(table.human.novartis[,2]),
```

```
as.character(table.human.novartis[,3]),as.character(table.human.novartis[,4]),sep=" - ")
y=unique(x)
z=sort(y)

x11(h=10,w=14)
par(mar=c(1,1,0.3,1))
plot(0,col="white",axes=F,xlab="",ylab="")

#ncol=18
legend("bottomleft",z[1:800],ncol=15,cex=0.55,text.col="black")
title(main="\n\n\nTissue restricted antigens (TRAs) in all human datasets")
setwd("/home/dinkelac/data/mouse/tables/")
write.csv(z,"tras.human.novartis.annot.txt",row.names=F)

tissues=as.character(tras.human.5x.novartis[,11])
tissues.all=unlist(strsplit(as.character(tras.human.5x.novartis[,10]),split="/"))
single.tissues=sort(unique(tissues.all))

tiss=vector(len=length(single.tissues))
names(tiss)=single.tissues

for(i in 1:length(single.tissues)){
tiss[i]=length(which(tissues==single.tissues[i]))
}

tiss.1=sort(tiss)

x11(h=7,w=10)
par(las=2)
mmi=c(2.3,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
barplot(tiss[-1],col="green",cex.names=0.7)
title(main="Tissue expression of TRAs in all tissue types (human Novartis data)")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="all.tissues.tras.human.novartis.eps")

#HLA Gene
all.tissues=
as.character(tras.human.5x.novartis[grep("TUBA",tras.human.5x.novartis$V3),11])
tissues=
as.character(unique(tras.human.5x.novartis[grep("TUBA",tras.human.5x.novartis$V3),11]))
tiss=vector(len=length(tissues))
names(tiss)=tissues

for(i in 1: length(tissues)){
tiss[i]=length(which(all.tissues==tissues[i]))
}

x11(h=5,w=3)
par(las=2)
mmi=c(2,0.5,0.5,0.5)
par(mai=mmi)
barplot(tiss,col="red",cex.names=0.7,cex.axis=0.7)
title(main="Tissues for TUBA TRAs")

setwd("/home/dinkelac/data/mouse/plots")
dev.copy2eps(file="TUBA.tras.novartis.human.eps")
```

1.1.4 Calculate Tissue Restricted Antigens (TRAs) in GTEX data

This script is calculating tissue restricted antigens (TRAs) in GTEX data.

```
#calc_tras_human_GTEX_data.R
#----------------------------
setwd("/home/dinkelac/data/mouse/tables/")

#6.2 Apply annotation to the hgu133A chip
#----------------------------------------
ensembl.59.human.table.input=read.csv(file="ensembl.59.human.txt",sep="\t")

#6.2.1 table. with transcript IDs
#--------------------------------
ensembl.59.transcripts=as.character(ensembl.59.human.table.input[,2])
double.true=duplicated(ensembl.59.transcripts)#F,T
double.index=which(double.true==TRUE)#no
#2086 double transcripts

ensembl.59.transcripts=ensembl.59.transcripts[-double.index]
#151250

ensembl.59.human.transcripts=ensembl.59.human.table.input[-double.index,]
rownames(ensembl.59.human.transcripts)=ensembl.59.human.transcripts[,2]

#6.2.2 table with gene IDs
#-------------------------
ensembl.59.genes=as.character(ensembl.59.human.table.input[,1])
double.true=duplicated(ensembl.59.genes)#T,F
double.index=which(double.true==TRUE)#no
#101599 double genes

ensembl.59.genes=ensembl.59.genes[-double.index]
#51737

ensembl.59.human.genes=ensembl.59.human.table.input[-double.index,]
rownames(ensembl.59.human.genes)=ensembl.59.human.genes[,1]

#all Transcripts on hgu133A chip
#-------------------------------
hgu133A.transcript.names=rownames(mean.vsnrma)
hgu133A.transcript.names=hgu133A.transcript.names[69:length(hgu133A.transcript.names)]
hgu133A.transcript.names=substr(hgu133A.transcript.names,1,nchar(hgu133A.transcript.names)-3)
#49028

gene.id=hgu133A.transcript.names
chromosome=as.character(ensembl.59.human.transcripts[hgu133A.transcript.names,3])
startside=ensembl.59.human.transcripts[hgu133A.transcript.names,4]
strand=ensembl.59.human.transcripts[hgu133A.transcript.names,6]
band=as.character(ensembl.59.human.transcripts[hgu133A.transcript.names,7])
symbol=as.character(ensembl.59.human.transcripts[hgu133A.transcript.names,8])
names(symbol)=hgu133A.transcript.names

human.hgu133A.transcript.ensembl.59.annotated=cbind(gene.id,chromosome,startside,strand,
band,symbol)

setwd("/home/dinkelac/data/mouse/tables/")
write.table(human.hgu133A.transcript.ensembl.59.annotated,
file="hgu133A.transcripts.annotated.csv",sep=",")

#8. TRA tables
#-------------
```

```
#chrhash

#8.1.1 tra.index2.5x, tra.index2.10x transcripts
#-----------------------------------------------
gene.ids=names(tra.index2.10x)
gene.id=substr(gene.ids,1,nchar(gene.ids)-3)

chrom=as.character(ensembl.59.human.transcripts[gene.id,3])
start=ensembl.59.human.transcripts[gene.id,4]

table.tra.index2.10x=cbind(gene.ids,chrom,start)

setwd("/home/dinkelac/data/mouse/tables/")

write.table(table.tra.index2.10x,"tra.human.index2.10x.transcripts.txt",
col.names=F,row.names=F,quote=F,sep="\t")

#8.2 chrhash.all erstellen, transcripts
#--------------------------------------
gene.names=rownames(mean.vsnrma)
gene.ids=gene.names[69:length(gene.names)]
gene.id=substr(gene.ids,1,nchar(gene.ids)-3)

chrom=as.character(ensembl.59.human.transcripts[gene.id,3])
start=ensembl.59.human.transcripts[gene.id,4]

table.ensembl.59.transcripts=cbind(gene.ids,chrom,start)

write.table(table.ensembl.59.transcripts,"ensembl.59.chrhash.all.transcripts.txt",
col.names=F,row.names=F,quote=F,sep="\t")

#annotated

#8.3 hgu133A.genes.on.chip, transcripts
#--------------------------------------
gene.names=rownames(mean.vsnrma)
gene.ids=gene.names[69:length(gene.names)]
gene.id=substr(gene.ids,1,nchar(gene.ids)-3)

chrom=as.character(ensembl.59.human.transcripts[gene.id,3])

start=ensembl.59.human.transcripts[gene.id,4]
strand=ensembl.59.human.transcripts[gene.id,6]
band=as.character(ensembl.59.human.transcripts[gene.id,7])
symbol=as.character(ensembl.59.human.transcripts[gene.id,8])

table.hgu133A.ensembl.59.transcripts=cbind(gene.ids,chrom,start,strand,band,symbol)

write.table(table.hgu133A.ensembl.59.transcripts,
"hgu133A.transcripts.ensembl.59.annotated.csv", sep="\t",row.names=F)

#8.4 TRAs annotated, transcripts
#-------------------------------
gene.ids=names(tra.index2.10x)
gene.id=substr(gene.ids,1,nchar(gene.ids)-3)

chrom=as.character(ensembl.59.human.transcripts[gene.id,3])

start=ensembl.59.human.transcripts[gene.id,4]
strand=ensembl.59.human.transcripts[gene.id,6]
band=as.character(ensembl.59.human.transcripts[gene.id,7])
symbol=as.character(ensembl.59.human.transcripts[gene.id,8])
```

```
table.tra.index2.10x.annotated=cbind(gene.ids,chrom,start,strand,band,symbol)

write.table(table.tra.index2.10x.annotated,"tra.human.index2.10x.transcripts.annotated.csv",
sep="\t",row.names=F)

setwd("/home/dinkelac/data/mouse/session/rda")
save.image(file="tras.human.rda")

#9. Annotation on gene level
#---------------------------
#map transcripts to genes
#------------------------
deaffy = function(x) sub("_at$", "", x)

#9.1 housekeeping genes
#----------------------
housekeeping.genes.2x=
as.character(unique(ensembl.59.human.transcripts[deaffy(names(housekeeping.index)),1]))

gene.ids=housekeeping.genes.2x
chrom=as.character(ensembl.59.human.genes[housekeeping.genes.2x,3])
start=ensembl.59.human.genes[housekeeping.genes.2x,4]

table.housekeeping.genes.2x=cbind(gene.ids,chrom,start)

setwd("/home/dinkelac/data/mouse/tables/")
write.table(table.housekeeping.genes.2x,"housekeeping.genes.human.2x.txt",
col.names=F,row.names=F,quote=F,sep="\t")

#9.2. tra.index2.5x, genes
#-------------------------
genes.tra2.5x=
as.character(unique(ensembl.59.human.transcripts[deaffy(names(tra.index2.5x)),1]))
genes.tra2.10x=
as.character(unique(ensembl.59.human.transcripts[deaffy(names(tra.index2.10x)),1]))

#index2.5x: 2760 genes, 9499 transcripts
#index2.10x:1406 genes, 4881 transcripts

gene.ids=genes.tra2.10x
chrom=as.character(ensembl.59.human.genes[genes.tra2.10x,3])
start=ensembl.59.human.genes[genes.tra2.10x,4]

strand=ensembl.59.human.genes[genes.tra2.10x,6]
band=as.character(ensembl.59.human.genes[genes.tra2.10x,7])
symbol=as.character(ensembl.59.human.genes[genes.tra2.10x,8])

table.genes.tra.index2.10x=cbind(gene.ids,chrom,start)
table.genes.tra.index2.10x.annotated=cbind(gene.ids,chrom,start,strand,band,symbol)

setwd("/home/dinkelac/data/mouse/tables/")

write.table(table.genes.tra.index2.10x,"tra.human.index2.10x.genes.txt",col.names=F,
row.names=F,quote=F,sep="\t")
write.table(table.genes.tra.index2.10x.annotated,"tra.human.index2.10x.genes.annotated.csv",
row.names=F,sep="\t")

#9.3 hgu133A, genes
#------------------
gene.names=rownames(mean.vsnrma)
gene.ids=gene.names[69:length(gene.names)]
```

```
genes.hgu133A=as.character(unique(ensembl.59.human.transcripts[deaffy(gene.ids),1]))
#13663 genes, 49028 transcripts

gene.ids=genes.hgu133A
chrom=as.character(ensembl.59.human.genes[genes.hgu133A,3])
start=ensembl.59.human.genes[genes.hgu133A,4]

strand=ensembl.59.human.genes[genes.hgu133A,6]
band=as.character(ensembl.59.human.genes[genes.hgu133A,7])
symbol=as.character(ensembl.59.human.genes[genes.hgu133A,8])

table.genes.hgu133A=cbind(gene.ids,chrom,start)
table.genes.hgu133A.annotated=cbind(gene.ids,chrom,start,strand,band,symbol)

write.table(table.genes.hgu133A,"hgu133A.chrhash.all.genes.txt",col.names=F,row.names=F,
quote=F,sep="\t")
write.table(table.genes.hgu133A.annotated,"hgu133A.genes.ensembl.59.annotated.csv",
row.names=F,sep="\t")

#9.4 ensembl, genes
#------------------
gene.ids=ensembl.59.genes
#51737
chrom=as.character(ensembl.59.human.genes[ensembl.59.genes,3])
start=ensembl.59.human.genes[ensembl.59.genes,4]

table.chrhash.all.ensembl.59=cbind(gene.ids,chrom,start)

setwd("/home/dinkelac/data/mouse/tables/")
write.table(table.chrhash.all.ensembl.59,"ensembl.59.chrhash.all.genes.txt",col.names=F,
row.names=F,quote=F,sep="\t")

#10 annotated TRA lists
#----------------------
#10.1 entrezID
#-------------
ensembl.59.entrezID=read.csv(file="ensembl.59.human.entrezID.txt")
#for 21916 genes, 19470 RefseqIDs

#10.1.1. hgu133A genes + entrezID
#--------------------------------
hgu133A.genes.entrez=vector(len=length(genes.hgu133A))
names(hgu133A.genes.entrez)=genes.hgu133A

for(i in 1:length(hgu133A.genes.entrez)){
gene.ID=names(hgu133A.genes.entrez[i])
index=which(ensembl.59.entrezID[,1]==gene.ID)
#no entrez ID
if(length(index)==0){
hgu133A.genes.entrez[i]=NA
}
else{
entrezIDs=ensembl.59.entrezID[index,3]
entrezID=unique(entrezIDs[!is.na(entrezIDs)])
if(length(entrezID)>=1){
entrezID=paste(entrezID,collapse="/")
}
if(length(entrezID)==0){
entrezID=NA
}
hgu133A.genes.entrez[i]=entrezID
```

```
}#end else
}#end for

geneID=genes.hgu133A
entrezID=hgu133A.genes.entrez

table.hgu133A.genes.entrezID=cbind(geneID,entrezID)
#12055 entrezIDs for 13663 genes on the chip

setwd("/home/dinkelac/data/mouse/tables/")
write.table(table.hgu133A.genes.entrezID,"hgu133A.genes.entrezID.ensembl.59.txt",
row.names=F,sep="\t")

#10.1.2. hgu133A transcripts + entrezID
#--------------------------------------
hgu133A.transcripts.entrez=vector(len=length(hgu133A.transcript.names))
names(hgu133A.transcripts.entrez)=hgu133A.transcript.names

for(i in 1:length(hgu133A.transcripts.entrez)){
transcript.ID=names(hgu133A.transcripts.entrez[i])
index=which(ensembl.59.entrezID[,2]==transcript.ID)
#no entrez ID
if(length(index)==0){
hgu133A.transcripts.entrez[i]=NA
}
else{
entrezIDs=ensembl.59.entrezID[index,3]
entrezID=unique(entrezIDs[!is.na(entrezIDs)])
if(length(entrezID)>=1){
entrezID=paste(entrezID,collapse="/")
}
if(length(entrezID)==0){
entrezID=NA
}
hgu133A.transcripts.entrez[i]=entrezID
}#end else
}#end for

#10.2. max tissue
#----------------
#dauert ! (of all TRAs)
transcript.ids=names(tra.index2.5x)

max.tiss=list()

for (i in 1:length(transcript.ids)){
#print(i)
max.value=max(mean.vsnrma[transcript.ids[i],])
max.tiss[i]=names(which(mean.vsnrma[transcript.ids[i],]==max.value))
}

max.tissue.transcripts=unlist(max.tiss)
names(max.tissue.transcripts)=deaffy(transcript.ids)

#per gene
#--------
max.tissue.genes=vector(len=length(genes.tra2.5x))
names(max.tissue.genes)=genes.tra2.5x

for(i in 1:length(genes.tra2.5x)){
index=which(ensembl.59.human.table.input==genes.tra2.5x[i])
transcript.names=unique(as.character(ensembl.59.human.table.input[index,2]))
```

```
max.tissues=max.tissue.transcripts[transcript.names]
max.tissue=unique(max.tissues[!is.na(max.tissues)])
max.tissue.genes[i]=paste(max.tissue,collapse="/")
}

#10.3. number of tissues over cutoff
#-----------------------------------
x=apply(mean.vsnrma[1,],1,count.over.median,crit=5)
tiss.no.over.cutoff.transcripts=x[transcript.ids]
names(tiss.no.over.cutoff.transcripts)=deaffy(names(tiss.no.over.cutoff.transcripts))

#per gene
#--------
tiss.no.over.cutoff.genes=vector(len=length(genes.tra2.5x))
names(tiss.no.over.cutoff.genes)=genes.tra2.5x

for(i in 1:length(genes.tra2.5x)){
index=which(ensembl.59.human.table.input==genes.tra2.5x[i])
transcript.names=unique(as.character(ensembl.59.human.table.input[index,2]))
tiss.numbers=tiss.no.over.cutoff.transcripts[transcript.names]
tiss.number=unique(tiss.numbers[!is.na(tiss.numbers)])
tiss.no.over.cutoff.genes[i]=paste(tiss.number,collapse="/")
}

#10.4. tissues over cutoff
#-------------------------
tiss.act=list()

for (i in 1:length(transcript.ids)){
med.act=exp(median(mean.vsnrma[transcript.ids[i],]))
tiss.act[i]=list(names(which(exp(mean.vsnrma[transcript.ids[i],])>5*med.act)))
}

tissues.over.cutoff.transcripts=pasteList(tiss.act)
names(tissues.over.cutoff.transcripts)=deaffy(transcript.ids)

#per gene
#--------
tissues.over.cutoff.genes=vector(len=length(genes.tra2.5x))
names(tissues.over.cutoff.genes)=genes.tra2.5x

for(i in 1:length(genes.tra2.5x)){
index=which(ensembl.59.human.table.input==genes.tra2.5x[i])
transcript.names=unique(as.character(ensembl.59.human.table.input[index,2]))
tissues=tissues.over.cutoff.transcripts[transcript.names]
tissue=unique(tissues[!is.na(tissues)])
tissues.over.cutoff.genes[i]=paste(tissue,collapse="//")
}

#10.5. table TRA annotated + entrezID + tissues over cutoff (transcripts)
#-------------------------------------------------------------------------
entrezID=hgu133A.transcripts.entrez[deaffy(names(tra.index2.5x))]
tiss.no.over.cutoff=tiss.no.over.cutoff.transcripts
max.tissue=max.tissue.transcripts

table.transcripts.tra.index2.5x.annotated=cbind(table.tra.index2.5x.annotated,
entrezID, tiss.no.over.cutoff,max.tissue,tissues.over.cutoff)

setwd("/home/dinkelac/data/mouse/tables/")
write.table(table.transcripts.tra.index2.5x.annotated,
"tra.human.index2.5x.transcripts.annotated.tissues.csv",row.names=F,sep="\t")
```

```
#10.6. table TRA annotated + entrezID + tissues over cutoff (genes)
#-----------------------------------------------------------------
entrezID=hgu133A.genes.entrez[genes.tra2.5x]
tiss.no.over.cutoff=tiss.no.over.cutoff.genes
max.tissue=max.tissue.genes
tissues.over.cutoff=tissues.over.cutoff.genes

table.genes.tra.index2.5x.entrez.annotated=cbind(table.genes.tra.index2.5x.annotated,
entrezID, tiss.no.over.cutoff,max.tissue,tissues.over.cutoff)

setwd("/home/dinkelac/data/mouse/tables/")
write.table(table.genes.tra.index2.5x.entrez.annotated,
"tra.human.index2.5x.genes.annotated.tissues.csv",row.names=F,sep="\t")

setwd("/home/dinkelac/data/mouse/sessions/rda")
save.image(file="tras.human.rda")

#Function: pasteList
#-------------------
pasteList=function(list){
result=list()
        for(i in 1:length(list)){
        tiss=unlist(list[i])
        res=tiss[1]
        if(length(tiss)>1){
                for(j in 2:length(tiss)){
                res=paste(res,tiss[j], sep="/")
                }
        }
        result[i]=res
        result=unlist(result)
        }
return(result)
}
#next skript: calc_clustering_human.R
```

1.1.5 Calculate clustering of TRAs in Microarray data

This script is for the calculation of chromosomal clustering in the Microarray data.

```
#calc_clustering_gngnf1.R
#-------------------------

#1. clustering
#-------------
#/home/dinkelac/data/mouse/clustering/
#annotated, ensembl 75

#1.1 Observed_ntuples
#--------------------
#make a directory for every run
#mkdir observed_ntuples_mouse_3x
#mkdir observed_ntuples_mouse_5x
#mkdir observed_ntuples_mouse_10x
#mkdir observed_ntuples_mouse_20x

mouse.chrhash.genes.txt
```

```
tra.2014.mouse.3x.genes.txt
tra.2014.mouse.5x.genes.txt
tra.2014.mouse.10x.genes.txt
tra.2014.mouse.20x.genes.txt

housekeeping.genes.mouse.2x.txt
#has to be changed for the chromosomes of species and different windowsizes

#get the perl script observed_ntuples_mouse.pl
observed_ntuples_mouse.pl

#run perl script
perl observed_ntuples_mouse.pl tra.2014.mouse.3x.genes.txt mouse.chrhash.genes.txt >results.txt

#1.2 validate_ntuples
#--------------------
#mkdir validate_ntupes_mouse_3x
#mkdir validate_ntupes_mouse_5x
#mkdir validate_ntupes_mouse_10x
#mkdir validate_ntupes_mouse_20x

mouse.chrhash.genes.txt  #umbenennen in chrhash.txt

#get script start_perl_batch.sh
start_perl_batch.sh #change path

#get script validate_ntuples.pl
validate_ntuples.pl # 4x fuer andere Fenstergroessen anpassen

#NAs zaehlen (Konsole)
#---------------------
wc -l traindex.txt
grep -c "NA" traindex.txt

#rechte setzen
chmod 755 start_perl_batch.sh

#shell script starten
#--------------------
qsub -l walltime=300:00:00 -M m.dinkelacker@dkfz.de -mea start_perl_batch.sh

#2 Neighbors
#-----------
#mydata.rda erstellen

#library
library(affy)

#session
#-------
setwd("/home/dinkelac/data/mouse/sessions/rda")
load("tras_annotated.rda")

#function
#--------
source(file="/home/dinkelac/R-functions/dist.genloc.R")
deaffy=function(x){sub("_at$","",x)}

table.mouse.chrhash.genes
table.human.chrhash.genes
table.human.roth.chrhash.genes
table.mouse.4301.chrhash.genes
```

```
mouse.genes=table.mouse.chrhash.genes[,1]
human.genes=table.human.chrhash.genes[,1]
human.roth.genes=table.human.roth.chrhash.genes[,1]
mouse.4301.genes=table.mouse.4301.chrhash.genes[,1]

#chromosomes (chip)
mouse.chrs=table.mouse.chrhash.genes[,2]
names(mouse.chrs)=mouse.genes

human.chrs=table.human.chrhash.genes[,2]
names(human.chrs)=human.genes

human.roth.chrs=table.human.roth.chrhash.genes[,2]
names(human.roth.chrs)=human.roth.genes

mouse.4301.chrs=table.mouse.4301.chrhash.genes[,2]
names(mouse.4301.chrs)=mouse.4301.genes

#startsides (chip)
mouse.gstr=table.mouse.chrhash.genes[,3]
names(mouse.gstr)=mouse.genes

human.gstr=table.human.chrhash.genes[,3]
names(human.gstr)=human.genes

human.roth.gstr=table.human.roth.chrhash.genes[,3]
names(human.roth.gstr)=human.roth.genes

mouse.4301.gstr=table.mouse.4301.chrhash.genes[,3]
names(mouse.4301.gstr)=mouse.4301.genes

#tras
housekeeping.genes.mouse.2x=as.character(table.housekeeping.mouse.genes.2x[,1])

tra.genes.index.mouse.3x=which(mouse.genes%in%tra.mouse.genes.3x)
tra.genes.index.mouse.5x=which(mouse.genes%in%tra.mouse.genes.5x)
tra.genes.index.mouse.10x=which(mouse.genes%in%tra.mouse.genes.10x)
tra.genes.index.mouse.20x=which(mouse.genes%in%tra.mouse.genes.20x)

housekeeping.genes.index.mouse.2x=which(mouse.genes%in%housekeeping.genes.mouse.2x)

tra.human.genes.3x=as.character(table.tra.human.genes.3x[,1])
tra.human.genes.5x=as.character(table.tra.human.genes.5x[,1])
tra.human.genes.10x=as.character(table.tra.human.genes.10x[,1])
tra.human.genes.20x=as.character(table.tra.human.genes.20x[,1])

housekeeping.genes.human.2x=as.character(table.housekeeping.human.genes.2x[,1])

tra.genes.index.human.3x=which(human.genes%in%tra.human.genes.3x)
tra.genes.index.human.5x=which(human.genes%in%tra.human.genes.5x)
tra.genes.index.human.10x=which(human.genes%in%tra.human.genes.10x)
tra.genes.index.human.20x=which(human.genes%in%tra.human.genes.20x)

housekeeping.genes.index.human.2x=which(human.genes%in%housekeeping.genes.human.2x)

tra.mouse4301.genes.3x=as.character(table.tra.mouse.4301.genes.3x[,1])
tra.mouse4301.genes.5x=as.character(table.tra.mouse.4301.genes.5x[,1])
tra.mouse4301.genes.10x=as.character(table.tra.mouse.4301.genes.10x[,1])
tra.mouse4301.genes.20x=as.character(table.tra.mouse.4301.genes.20x[,1])

housekeeping.mouse4301.2x=as.character(table.housekeeping.mouse4301.genes.2x[,1])
```

```
tra.genes.index.mouse4301.3x=which(mouse.4301.genes%in%tra.mouse4301.genes.3x)
tra.genes.index.mouse4301.5x=which(mouse.4301.genes%in%tra.mouse4301.genes.5x)
tra.genes.index.mouse4301.10x=which(mouse.4301.genes%in%tra.mouse4301.genes.10x)
tra.genes.index.mouse4301.20x=which(mouse.4301.genes%in%tra.mouse4301.genes.20x)

housekeeping.genes.index.mouse4301.2x=which(mouse.4301.genes%in%housekeeping.mouse4301.2x)

tra.human.roth.genes.3x=as.character(table.tra.human.roth.genes.3x[,1])
tra.human.roth.genes.5x=as.character(table.tra.human.roth.genes.5x[,1])
tra.human.roth.genes.10x=as.character(table.tra.human.roth.genes.10x[,1])
tra.human.roth.genes.20x=as.character(table.tra.human.roth.genes.20x[,1])

housekeeping.human.roth.genes.2x=as.character(table.housekeeping.human.roth.genes.2x[,1])

tra.genes.index.human.roth.3x=which(human.roth.genes%in%tra.human.roth.genes.3x)
tra.genes.index.human.roth.5x=which(human.roth.genes%in%tra.human.roth.genes.5x)
tra.genes.index.human.roth.10x=which(human.roth.genes%in%tra.human.roth.genes.10x)
tra.genes.index.human.roth.20x=which(human.roth.genes%in%tra.human.roth.genes.20x)

housekeeping.genes.index.human.roth.2x=
which(human.roth.genes%in%housekeeping.human.roth.genes.2x)

#housekeeping gehen nicht bisher

setwd("/home/dinkelac/data/mouse/sessions/rda")

chrs=human.roth.chrs
gstr=human.roth.gstr
housekeeping.genes.index=housekeeping.genes.index.human.roth.2x
chip.genes=human.roth.genes

save(chrs,gstr,housekeeping.genes.index,chip.genes,dist.genloc,
file="mydata.human.roth.housekeeping.2x.rda")

setwd("/home/dinkelac/data/mouse/sessions/rda")
save.image("tras_clustering.rda")

#nachbessern:

load("mydata.human.roth.housekeeping.2x.rda")

names=names(gstr)
start=as.numeric(gstr)
names(start)=names

gstr=start

save(chrs,gstr,housekeeping.genes.index,chip.genes,dist.genloc,
file="mydata.human.roth.housekeeping.2x.rda")

#2.2. Permutation ausfuehren
#---------------------------
#/home/dinkelac/data/mouse/neighbors/

#mkdir /batch_gngnf1_10_5x_1000

*mydata9.5x.rda
*permute_chrloc.R#anpassen!
*start_R_batch.sh#anpassen!

#qsub -l walltime=800:00:00,mem=24000m start_R_batch.sh
```

```
#2.3. Plotten (missing)
#------------
library(affy)

# setwd("/home/dinkelac/data/mouse/neighbors/mouse.2014.3x/")
setwd("/home/dinkelac/data/mouse/sessions/rda/")

load("mydata.human.roth.20x.rda")
names=names(gstr)
start=as.numeric(gstr)
names(start)=names

gstr=start

#transcripte????
tra.index=tra.genes.index

#distance matrix
dist.mat.alltra=dist.genloc(tra.index,tra.index)

#mouse4301 - 3x

#dist.genloc in jeder session neu ueberladen, war bis Jan 2015 falsch

dist.genloc=function (X, Y)
{
    dX <- length(X)
    dY <- length(Y)
    chr.mat <- matrix(nr = dX, nc = dY)
    for (x in 1:dX) {
        for (y in 1:dY) {
            if (!is.na(chrs[X[x]]) && !is.na(chrs[Y[y]])) {
                match <- unlist(chrs[X[x]]) == unlist(chrs[Y[y]])
            }
            else {
                match <- FALSE
            }
            if (match) {
                chr.mat[x, y] <- 0
            }
            else {
                chr.mat[x, y] <- NA
            }
        }
    }

#value changed to as.numeric
    diff.mat <- outer(as.numeric(gstr[X]),as.numeric(gstr[Y]),"-")
    result.mat <- diff.mat + chr.mat
    return(abs(result.mat))
}

#save rda
setwd("/home/dinkelac/data/mouse/neighbors/human.roth.2014.20x/")

save(dist.mat.alltra,file="distancematrix.tra.human.roth.20x.rda")
#20x bis hier gespeichert
#mouse.2014.3x (rechnet gerade die Permutation

load("results_permute_chrloc_1000.rda")
#datei fehlt bei mouse.4301.3x
```

```
#fehlt bei human.roth.20x

windows=c(50000,100000,200000,500000,800000,2e6,5e6)
windowsize=c("50k","100k","200k","500k","800k","2m","5m")
windowsize1=c("50kb","100kb","200kb","500kb","800kb","2mb","5mb")
distances.tra.index.human.roth.20x=matrix(nr=1,nc=7)

rownames(distances.tra.index.human.roth.20x)=list(c("number of genes"))
colnames(distances.tra.index.human.roth.20x)=windowsize

#1x7 distance matrix
for(i in 1:7){
distances.tra.index.human.roth.20x[i]=
sum(dist.mat.alltra[upper.tri(dist.mat.alltra)] < windows[i], na.rm=T)
}

#write.csv(distances.tra.index10.10x,file="distances.tra.index10.10x.csv")

#histogramm
n=1000
value=distances.tra.index.human.roth.20x[1:7]

#colnames(value)=windowsize
min=c(000,00,1000,2000,6000,12000,20000)
max=c(1500,2500,4000,8000,10000,20000,45000)

x11(h=9,w=8)

#function
#--------
plot.hist.all=function(windowsize,value,min,max,n){
par(mfrow=c(4,2),oma=c(0,2,2,0)+0.1,mar=c(4,4,2,2)+0.1)

for(i in 1:7){
x=get(paste("pairs.in.",windowsize[i],sep=""))

hist(x,col="lightgrey",main=paste("windowsize =",windowsize1[i],sep=""),
ylab="frequency",xlab=paste("number of neighbors within the windowsize of ",
windowsize1[i],sep=""),xlim=c(min[i],max[i]) )
abline(v=value[i],col="red",lwd="3")
pwert=1/n
legend(bg="white","topright",paste("p<",pwert,sep=""),text.col="red")
}

title(main=paste("Gene neighbors: TRAs human roth 20x over median, ",n," repeats",sep=""),
outer=TRUE)
}

plot.hist.all(windowsize,value,min,max,n)

setwd("/home/dinkelac/data/mouse/plots/")
dev.copy2eps(file="neighbors_tra.human.roth.2014.20x_1000x.eps")

#3. plot 10-gene window
#----------------------
#function
source("~/R-functions/errorbars.R")

# in result erste zeile durch 2 tab 3 tab ... ersetzen
# letzte zeilen rausstreichen
setwd("/home/dinkelac/data/mouse/clustering/validate_ntuples_human_roth_housekeeping")
```

```
a=read.table("results.txt",fill=TRUE,header=TRUE)

#for (i in 1:10){print(sum(!is.na(a[,i])))}
names(a)=c(2:10)
means=apply(a,2,mean,na.rm=T)
sd=apply(a,2,sd,na.rm=T)
z=barplot(means,plot=FALSE)
#max. occurence des hochsten clusters
max1=max(a[,1],na.rm=T)

#observed n-tuples
#-----------------
setwd("/home/dinkelac/data/mouse/clustering/observed_ntuples_human_roth_housekeeping")

#use the last lines of results.txt to put them into results.clusters.txt
#b=read.table("results.clusters.txt",fill=TRUE)
b=read.table("results.txt",fill=TRUE)
ende=dim(b)[1]
b1=b[(ende-1):ende,]
b2=b1[,1:30]
b3a=as.character(unlist(b2[1,]))
b3b=as.character(unlist(b2[2,]))
# b3c=as.character(unlist(b2[3,]))
b3=c(b3a,b3b)
b4a=as.character(unlist(b2[3,]))
b4b=as.character(unlist(b2[4,]))
# b4c=as.character(unlist(b2[6,]))
b4=c(b4a,b4b)

gene.cluster=as.numeric(as.character(unlist(b2[2,])))
names(gene.cluster)=as.character(unlist(b2[1,]))

# gene.cluster=as.numeric(b4)
# names(gene.cluster)=b3
list.names=union(names(means),names(gene.cluster))
hist.list=matrix(nrow=2,ncol=length(list.names))
colnames(hist.list)=list.names
rownames(hist.list)=c("gene.cluster","means")

for(i in 1:length(list.names)){
if(list.names[i]%in%names(gene.cluster)==TRUE){
hist.list[1,i]=as.numeric(gene.cluster[list.names[i]])
}
if(list.names[i]%in%names(means)==TRUE){
hist.list[2,i]=as.numeric(means[list.names[i]])
}
}

maxim=max(max1,hist.list[1,1])

#barplot both lists
#------------------
x11(h=7,w=10)
barplot(hist.list,ylim=c(0,maxim),col=c("red","white"),beside=T,

legend.text=c("human.roth.housekeeping.genes","1000 random lists"),cex.names=0.7,
xlim=c(0,200),
xlab="clustersize k compared to 20774 human genes",ylab="number of clusters of size k")

title(main="10 gene window \n number of clusters of size k")
#title(sub="number of clusters of size k")
z=barplot(hist.list,beside=T,plot=FALSE)
```

```
for(i in 1:length(means)) ebars(z[2*i],means[i],means[i]+sd[i],means[i]-sd[i])

setwd("/home/dinkelac/data/mouse/plots/")

dev.copy2eps(file="cluster.human.roth.housekeeping.genes.eps")

#TO DO LIST - ACTUAL
#--------------------
#run the clustering script
#repeat housekeeping genes and mouse.3x/mouse.4301.3x
#find error in 10 gene window perl script (random list stops at max. 10) why?
#GO Analysis
#Plot Clusters
#Plot Syntheny Maps
#Count Numbers

#4. Clustering of housekeeping genes
#------------------------------------
x11(h=7,w=10)
barplot(as.matrix(hist.list),ylim=c(0,20),col=c("red","white"),beside=T,

legend.text=c("17 tca genes","1000 random lists"),cex.names=0.7,xlim=c(0,50),
xlab="clustersize k against 17121 gngnf1 genes",ylab="number of clusters of size k")

title(main="10 gene window method \n number of clusters of size k")

z=barplot(as.matrix(hist.list),beside=T,plot=FALSE)
for(i in 1:12) ebars(z[2*i],means[i],means[i]+sd[i],means[i]-sd[i])

setwd("/home/dinkelac/data/mouse/plots/")

dev.copy2eps(file="cluster.gngnf1.ensembl.59.tca.eps")

#4.1 Download annotation
#-----------------------
#ensembl 78 Go Terms human, mouse
setwd("/home/dinkelac/data/mouse/tables/")

#4.2 get housekeeping genes
#--------------------------
a=read.csv(file="ensembl.59.mouse.go.txt",sep="\t")

ensembl.59.genes=as.character(a[,1])
ensembl.59.transcripts=as.character(a[,2])
ensembl.59.go_bp=as.character(a[,3])
ensembl.59.go_cc=as.character(a[,4])
ensembl.59.go_mf=as.character(a[,5])
ensembl.59.go.table=cbind(ensembl.59.genes,ensembl.59.transcripts,ensembl.59.go_bp,
ensembl.59.go_cc,ensembl.59.go_mf)

rownames(ensembl.59.go.table)=ensembl.59.genes
get.genes=function(go.term){

b1=which(ensembl.59.go.table[,3]==go.term)
b2=which(ensembl.59.go.table[,4]==go.term)
b3=which(ensembl.59.go.table[,5]==go.term)

c=unique(c(b1,b2,b3))
d=ensembl.59.go.table[c,]
if(length(c)==1){
genes=unique(d[1])
}else{
```

```
genes=unique(d[,1])
}
return(genes)
}

#GO Analysis, clustering of Housekeeping genes, certain Pathways
#------------
#librarys
#--------
library(GO.db)

#sessionInfo()
#-------------
#GO.db_2.10.1 (3.0.0)
#AnnotationDbi_1.24.0 (1.28.)

setwd("/home/dinkelac/data/mouse/sessions/rda")
load("tras_clustering.rda")

#4.2.0 genes on the chips
#------------------------
mouse.genes=table.mouse.chrhash.genes[,1]
human.genes=table.human.chrhash.genes[,1]
mouse.4301.genes=table.mouse.4301.chrhash.genes[,1]
human.roth.genes=table.human.roth.chrhash.genes[,1]

#4.2.0 get genes from gngnf1 chip
#--------------------------------
setwd("/home/dinkelac/data/mouse/tables")
gngnf1.gene.table=read.csv(file="gngnf1.ensembl.59.chrhash.genes.txt",sep="\t",header=F)
gngnf1.genes=as.character(gngnf1.gene.table[,1])

#4.2.1 cell cycle
#----------------
cell.cycle.id="GO:0007049"
cell.cycle.offsprings=mget(cell.cycle.id,GOBPOFFSPRING)

go.terms=cell.cycle.offsprings$'GO:0007049'
#700

cell.cycle.genes=get.genes(cell.cycle.id)

#very slow
for(i in 1:length(go.terms)){
new.genes=get.genes(go.terms[i])
cell.cycle.genes=unique(c(cell.cycle.genes,new.genes))
print(i)
print(length(cell.cycle.genes))
}
#830 cell cycle genes, 467 go terms
#--------
setwd("/home/dinkelac/data/mouse/tables")
ensembl.59.chrhash=read.csv(file="ensembl.59.mouse.chrhash.genes.txt",sep="\t",header=F)
ensembl.59.chrhash=ensembl.59.chrhash[-5,]
rownames(ensembl.59.chrhash)=ensembl.59.chrhash[,1]
colnames(ensembl.59.chrhash)=c("geneID","chrom","startside")

cell.cycle.chrom=as.character(ensembl.59.chrhash[cell.cycle.genes,2])
cell.cycle.startside=as.character(ensembl.59.chrhash[cell.cycle.genes,3])
cell.cycle.genes.table=cbind(cell.cycle.genes,cell.cycle.chrom,cell.cycle.startside)

setwd("/home/dinkelac/data/mouse/tables")
```

```
write.table(cell.cycle.genes.table,"ensembl.59.gngnf1.cell.cycle.genes.txt", col.names=F,
row.names=F, quote=F, sep="\t")

#genes on chip
#-------------
genes.on.chip=cell.cycle.genes%in%gngnf1.genes
cell.cycle.genes.gngnf1.table=cell.cycle.genes.table[genes.on.chip,]

write.table(cell.cycle.genes.gngnf1.table,"gngnf1.cell.cycle.genes.txt", col.names=F,
row.names=F, quote=F, sep="\t")

#4.2.2 tca
#---------
tca.id="GO:0006099"
tca.offsprings=mget(tca.id,GOBPOFFSPRING)
go.terms=tca.offsprings$'GO:0006099'
#1

tca.genes=get.genes(tca.id)

#very slow
for(i in 1:length(go.terms)){
new.genes=get.genes(go.terms[i])
tca.genes=unique(c(tca.genes,new.genes))
print(i)
print(length(tca.genes))
}

tca.chrom=as.character(ensembl.59.chrhash[tca.genes,2])
tca.startside=as.character(ensembl.59.chrhash[tca.genes,3])
tca.genes.table=cbind(tca.genes,tca.chrom,tca.startside)

setwd("/home/dinkelac/data/mouse/tables")
write.table(tca.genes.table,"ensembl.59.gngnf1.tca.genes.txt", col.names=F, row.names=F,
quote=F, sep="\t")

#genes on chip
#-------------
genes.on.chip=tca.genes%in%gngnf1.genes
tca.genes.gngnf1.table=tca.genes.table[genes.on.chip,]

write.table(tca.genes.gngnf1.table,"gngnf1.tca.genes.txt", col.names=F, row.names=F,
quote=F, sep="\t")

#14.2.3 cytoskeleton
#-------------------
cytoskeleton.id="GO:0005856"
cytoskeleton.offsprings=mget(cytoskeleton.id,GOCCOFFSPRING)
go.terms=cytoskeleton.offsprings$'GO:0005856'
#159
cytoskeleton.genes=get.genes(cytoskeleton.id)

#very slow
for(i in 1:length(go.terms)){
new.genes=get.genes(go.terms[i])
cytoskeleton.genes=unique(c(cytoskeleton.genes,new.genes))
print(i)
print(length(cytoskeleton.genes))
}

#1302 cytoskeleton genes, 159 go terms
```

```
cytoskeleton.chrom=as.character(ensembl.59.chrhash[cytoskeleton.genes,2])
cytoskeleton.startside=as.character(ensembl.59.chrhash[cytoskeleton.genes,3])
cytoskeleton.genes.table=cbind(cytoskeleton.genes,cytoskeleton.chrom,cytoskeleton.startside)

setwd("/home/dinkelac/data/mouse/tables")
write.table(cytoskeleton.genes.table,"ensembl.59.gngnf1.cytoskeleton.genes.txt", col.names=F,
row.names=F, quote=F, sep="\t")

#genes on chip
#-------------
genes.on.chip=cytoskeleton.genes%in%gngnf1.genes
cytoskeleton.genes.gngnf1.table=cytoskeleton.genes.table[genes.on.chip,]

write.table(cytoskeleton.genes.gngnf1.table,"gngnf1.cytoskeleton.genes.txt", col.names=F,
row.names=F, quote=F, sep="\t")

#4.2.4 actin cytoskeleton
#------------------------
actin.cytoskeleton.id="GO:0015629"
actin.cytoskeleton.offsprings=mget(actin.cytoskeleton.id,GOCCOFFSPRING)
#51
actin.cytoskeleton.genes=get.genes(actin.cytoskeleton.id)
#2 genes
go.terms=actin.cytoskeleton.offsprings$'GO:0015629'

#very slow
for(i in 1:length(go.terms)){
new.genes=get.genes(go.terms[i])
actin.cytoskeleton.genes=unique(c(actin.cytoskeleton.genes,new.genes))
print(i)
print(length(actin.cytoskeleton.genes))
}

#245 actin cytoskeleton genes, 52 go terms

actin.cytoskeleton.chrom=as.character(ensembl.59.chrhash[actin.cytoskeleton.genes,2])
actin.cytoskeleton.startside=as.character(ensembl.59.chrhash[actin.cytoskeleton.genes,3])
actin.cytoskeleton.genes.table=cbind(actin.cytoskeleton.genes,actin.cytoskeleton.chrom,
actin.cytoskeleton.startside)

setwd("/home/dinkelac/data/mouse/tables")
write.table(actin.cytoskeleton.genes.table,"ensembl.59.gngnf1.actin.cytoskeleton.genes.txt",
col.names=F, row.names=F, quote=F, sep="\t")

#genes on chip
#-------------
genes.on.chip=actin.cytoskeleton.genes%in%gngnf1.genes
actin.cytoskeleton.genes.gngnf1.table=actin.cytoskeleton.genes.table[genes.on.chip,]

write.table(actin.cytoskeleton.genes.gngnf1.table,"gngnf1.actin.cytoskeleton.genes.txt",
col.names=F, row.names=F, quote=F, sep="\t")

#4.2.5 glycolysis
#----------------
glycolysis.id="GO:0006096"
glycolysis.offsprings=mget(glycolysis.id,GOBPOFFSPRING)
#5
glycolysis.genes=get.genes(glycolysis.id)
#1 gene
go.terms=glycolysis.offsprings$'GO:0006096'

#very slow
```

```
for(i in 1:length(go.terms)){
new.genes=get.genes(go.terms[i])
glycolysis.genes=unique(c(glycolysis.genes,new.genes))
print(i)
print(length(glycolysis.genes))
}

#55 glycolysis genes, 3 go terms

glycolysis.chrom=as.character(ensembl.59.chrhash[glycolysis.genes,2])
glycolysis.startside=as.character(ensembl.59.chrhash[glycolysis.genes,3])
glycolysis.genes.table=cbind(glycolysis.genes,glycolysis.chrom,glycolysis.startside)

setwd("/home/dinkelac/data/mouse/tables")
write.table(glycolysis.genes.table,"ensembl.59.gngnf1.glycolysis.genes.txt", col.names=F,
row.names=F, quote=F, sep="\t")

#genes on chip
#-------------
genes.on.chip=glycolysis.genes%in%gngnf1.genes
glycolysis.genes.gngnf1.table=glycolysis.genes.table[genes.on.chip,]

write.table(glycolysis.genes.gngnf1.table,"gngnf1.glycolysis.genes.txt", col.names=F,
row.names=F, quote=F, sep="\t")

#4.2.6 muscle
#------------
muscle.id="GO:0003012"
muscle.offsprings=mget(muscle.id,GOBPOFFSPRING)
#143

muscle.genes=get.genes(muscle.id)
go.terms=muscle.offsprings$'GO:0003012'

#very slow
for(i in 1:length(go.terms)){
new.genes=get.genes(go.terms[i])
muscle.genes=unique(c(muscle.genes,new.genes))
print(i)
print(length(muscle.genes))
}

#152 muscle genes, 143 go terms

muscle.chrom=as.character(ensembl.59.chrhash[muscle.genes,2])
muscle.startside=as.character(ensembl.59.chrhash[muscle.genes,3])
muscle.genes.table=cbind(muscle.genes,muscle.chrom,muscle.startside)

setwd("/home/dinkelac/data/mouse/tables")
write.table(muscle.genes.table,"ensembl.59.gngnf1.muscle.genes.txt", col.names=F, row.names=F,
quote=F, sep="\t")

#genes on chip
#-------------
genes.on.chip=muscle.genes%in%gngnf1.genes
muscle.genes.gngnf1.table=muscle.genes.table[genes.on.chip,]

write.table(muscle.genes.gngnf1.table,"gngnf1.muscle.genes.txt", col.names=F, row.names=F,
quote=F, sep="\t")

#4.2.7 apoptosis
#---------------
```

```
apoptosis.id="GO:0006915"
apoptosis.offsprings=mget(apoptosis.id,GOBPOFFSPRING)
#164
apoptosis.genes=get.genes(apoptosis.id)
go.terms=apoptosis.offsprings$'GO:0006915'

#very slow
for(i in 1:length(go.terms)){
new.genes=get.genes(go.terms[i])
apoptosis.genes=unique(c(apoptosis.genes,new.genes))
print(i)
print(length(apoptosis.genes))
}

#973 apoptosis genes, 164 go terms

apoptosis.chrom=as.character(ensembl.59.chrhash[apoptosis.genes,2])
apoptosis.startside=as.character(ensembl.59.chrhash[apoptosis.genes,3])
apoptosis.genes.table=cbind(apoptosis.genes,apoptosis.chrom,apoptosis.startside)

setwd("/home/dinkelac/data/mouse/tables")
write.table(apoptosis.genes.table,"ensembl.59.gngnf1.apoptosis.genes.txt", col.names=F,
row.names=F, quote=F, sep="\t")

#genes on chip
#-------------
genes.on.chip=apoptosis.genes%in%gngnf1.genes
apoptosis.genes.gngnf1.table=apoptosis.genes.table[genes.on.chip,]

write.table(apoptosis.genes.gngnf1.table,"gngnf1.apoptosis.genes.txt", col.names=F,
row.names=F, quote=F, sep="\t")

#4.2.8 caspase activity
#----------------------
caspase.id="GO:0006919"
caspase.offsprings=mget(caspase.id,GOBPOFFSPRING)
caspase.genes=get.genes(caspase.id)
go.terms=caspase.offsprings$'GO:0006919'

#very slow
for(i in 1:length(go.terms)){
new.genes=get.genes(go.terms[i])
caspase.genes=unique(c(caspase.genes,new.genes))
print(i)
print(length(caspase.genes))
}

caspase.chrom=as.character(ensembl.59.chrhash[caspase.genes,2])
caspase.startside=as.character(ensembl.59.chrhash[caspase.genes,3])
caspase.genes.table=cbind(caspase.genes,caspase.chrom,caspase.startside)

setwd("/home/dinkelac/data/mouse/tables")
write.table(caspase.genes.table,"ensembl.59.gngnf1.caspase.genes.txt", col.names=F,
row.names=F, quote=F, sep="\t")

#genes on chip
#-------------
genes.on.chip=caspase.genes%in%gngnf1.genes
caspase.genes.gngnf1.table=caspase.genes.table[genes.on.chip,]

write.table(caspase.genes.gngnf1.table,"gngnf1.caspase.genes.txt", col.names=F,
row.names=F, quote=F, sep="\t")
```

```
#--------------------
setwd("/home/dinkelac/data/mouse/sessions/rda")
save.image(file="housekeeping.gngnf1.rda")

#5. clustering of tissue types
#-----------------------------
setwd("/home/dinkelac/data/mouse/tables/tras/gngnf1/tra_index10/")
a=read.csv(file="tra.index10.5x.genes.annotated.tissues.csv",sep="\t")

tissues=as.character(a[,10])
x=unique(unlist(strsplit(tissues,split="/")))
tissue.types=(sort(x[-24]))

cns.names=c("amygdala","frontalcortex","preoptic","cerebellum","cortex","dorsalstriatum",
"hippocampus","hypothalamus","olfactorybulb","spinalcordlower","spinalcordupper",
"substantianigra","pituitary")
epidermis.names=c("digits","epidermis","snoutepidermis","tongueepidermis")
intestine.names=c("largeintestine","smallintestine")
ovary.names=c("ovary","oocyte")
pns.names=c("trigeminal","dorsalrootganglion","medialolfactoryepithelium(MOE)",
"vomeralnasalorgan(VMO)")

#not used tissues
#----------------
embryos.names=c("embryoday6.5","embryoday7.5","embryoday8.5","embryoday9.5",
"embryoday10.5")
immune.names=c("cd4+Tcell","cd8+Tcell","b220+bcell")

tissue.groups=tissue.types
tissue.groups[which(tissue.groups%in%cns.names)]="cns"
tissue.groups[which(tissue.groups%in%epidermis.names)]="epidermis"
tissue.groups[which(tissue.groups%in%intestine.names)]="intestine"
tissue.groups[which(tissue.groups%in%ovary.names)]="ovary"
tissue.groups[which(tissue.groups%in%pns.names)]="pns"
tissue.groups[which(tissue.groups%in%embryos.names)]="embryos"
tissue.groups[which(tissue.groups%in%immune.names)]="immune.cells"
tissue.groups=unique(tissue.groups)

groups=c("cns","epidermis","intestine","ovary","pns","embryos","immune.cells")

setwd("/home/dinkelac/data/mouse/tables/")

gene.table=matrix(nrow=828,ncol=length(tissue.groups))
colnames(gene.table)=tissue.groups

for(i in 1:length(tissue.groups)){
tissue=tissue.groups[i]
if(tissue%in%groups){
if(tissue=="cns"){
tissues=cns.names
}
if(tissue=="epidermis"){
tissues=epidermis.names
}
if(tissue=="intestine"){
tissues=intestine.names
}
if(tissue=="ovary"){
tissues=ovary.names
}
if(tissue=="pns"){
```

```
tissues=pns.names
}
if(tissue=="embryos"){
tissues=embryos.names
}
if(tissue=="immune.cells"){
tissues=immune.names
}
index=c()
for(j in 1:length(tissues)){
index0=grep(tissues[j],a[,10])
index=sort(unique(c(index,index0)))
}
}else{
index=grep(tissue,a[,10])
}
table=a[index,c(1,2,3,6)]

genes=sort(unique(as.character(table[,4])))
riken.index=grep("Rik",genes)
if(length(riken.index)>0){
genes=genes[-riken.index]
}
leer.index=which(genes=="")
if(length(leer.index)>0){
genes=genes[-leer.index]
}
for(k in 1:length(genes)){
gene.table[k,i]=genes[k]
}
colnames(gene.table)[i]=tissue

print(tissue)
write.table(table,file=paste("gngnf1.tra.index.10.5x.",tissue,".txt",sep=""),sep="\t",
row.names=F)
write.table(table[,c(1:3)],file=paste("chrhash.gngnf1.tra.index.10.5x.",tissue,".txt",sep=""),
sep="\t",row.names=F,col.names=F)
}
write.table(gene.table,file="gngnf1.genes.per.tissue.type.csv",sep="\t",row.names=F)

gngnf1.tra.index.10.5x.adiposetissue.txt
gngnf1.tra.index.10.5x.adrenalgland.txt
gngnf1.tra.index.10.5x.bladder.txt
gngnf1.tra.index.10.5x.blastocysts.txt
gngnf1.tra.index.10.5x.bonemarrow.txt
gngnf1.tra.index.10.5x.bone.txt
gngnf1.tra.index.10.5x.brownfat.txt
gngnf1.tra.index.10.5x.cns.txt
gngnf1.tra.index.10.5x.embryos.txt
gngnf1.tra.index.10.5x.epidermis.txt
gngnf1.tra.index.10.5x.fertilizedegg.txt
gngnf1.tra.index.10.5x.heart.txt
gngnf1.tra.index.10.5x.immune.cells.txt
gngnf1.tra.index.10.5x.intestine.txt
gngnf1.tra.index.10.5x.kidney.txt
gngnf1.tra.index.10.5x.liver.txt
gngnf1.tra.index.10.5x.lung.txt
gngnf1.tra.index.10.5x.lymphnode.txt
gngnf1.tra.index.10.5x.mammarygland(lact).txt
gngnf1.tra.index.10.5x.ovary.txt
gngnf1.tra.index.10.5x.pancreas.txt
gngnf1.tra.index.10.5x.placenta.txt
```

```
gngnf1.tra.index.10.5x.pns.txt
gngnf1.tra.index.10.5x.prostate.txt
gngnf1.tra.index.10.5x.retina.txt
gngnf1.tra.index.10.5x.salivarygland.txt
gngnf1.tra.index.10.5x.skeletalmuscle.txt
gngnf1.tra.index.10.5x.spleen.txt
gngnf1.tra.index.10.5x.stomach.txt
gngnf1.tra.index.10.5x.testis.txt
gngnf1.tra.index.10.5x.thymus.txt
gngnf1.tra.index.10.5x.thyroid.txt
gngnf1.tra.index.10.5x.trachea.txt
gngnf1.tra.index.10.5x.umbilicalcord.txt
gngnf1.tra.index.10.5x.uterus.txt
```

1.1.6 Perl Programs for the analysis of chromosomal clustering, 10 Gene Window Method

This Perl Program **observed_ntuples_mouse.pl** calculates the number of duplets, triplets, etc. in a sliding 10 Gene window method. The input is a gene list with Gene Identifier, chrosomome and startside.

```
#!/usr/bin/perl

#observed_ntuples.pl
#--------------------

# This programm calculates the number of doublets, triplets etc.
# in a sliding 10-gene window.
# input is a gene list tra.txt mit GeneID, Chromosome, Startsite
# z.B.: NM_008020        19 6969975

# reading in genelist from experiment
# ex. TRAs tissue restricted antigens

# usage: perl observed_ntuples.pl traindex3.txt[teilliste] #chrhash.txt[gesamtliste] >results.txt

$in = $ARGV[0];
$hash = $ARGV[1];
# $in = shift;
# $hash = shift;

sub unite {
  my $item = "";
  my %hash = ();
  while ($item = shift) {
    $hash{$item} = 1;
  }
  return (sort keys %hash);
}

sub max {
  my $max = shift (@_);
  foreach $foo (@_) {
    $max = $foo if ($max < $foo);
  }
  return $max;
}
```

```
##### test code; ignore #############
# @a1 = (1,2,3,4,5,6,7);
# @a2 = (4,5,6,7,8,9,10,11);
# @r = unite(@a1,@a2);
# $m = max(unite(@a1,@a2));
# print "$m\n";

# foreach $e (@r) {
#   print "$e\n";
# }

# die;
#####################################

open THYMDIFF, "$in";

$header = 0;

while (<THYMDIFF>) {
  chomp;
#Zeilen mit NAs rauswerfen
  next if (/^\S+\s+.*NA/);
  $thymdiff{ $1 } = 1 if (/^(\S+)/);
}

close THYMDIFF;

# reading in table chip_ids chromosome TSS

open CHRHASH, "$hash" or
  die "error while opening $hash: $!";

while (<CHRHASH>) {
  @l = split /\t/;
  #Zeilen mit NAs rauswerfen
  next if ($l[1]=~/NA/ || $l[2]=~/NA/);
  $chrhash{$l[0]} = $l[1]; # chip_id -> chromosome
  $sitehash{$l[0]} = $l[2]; # chip_id -> tss
}

close CHRHASH;

@chipids = sort (keys %chrhash);
#chromosomenliste MT, NT_123456 ?
@chrs = ("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12",
           "13", "14", "15", "16", "17", "18", "19","X", "Y");

foreach $chr (@chrs) {          # get genes per chromosome
  @genesperchr = ();
  foreach $id (@chipids) {
    if ($chrhash{ $id } eq $chr && $sitehash{ $id }!~/NA/) {
      push @genesperchr, $id;
    }
  }
  # order by TSS
  @orderedgenes = sort {$sitehash{$a} <=> $sitehash{$b}} @genesperchr;
  @clusters = ();
  %clusterids = ();
  $maxsizeold = 0;
  $maxsize = 0;
#   for ($a=0;$a<=($#orderedgenes-9);$a++) {  # start sliding window of size 10
#     $csz = 0;
```

```
#     for ($b=$a;$b<($a+10);$b++) { # evaluate for each site in window
#       # wheter contained in random list
#       if ($thymdiff{ $orderedgenes[$b] } == 1) {
#       $csz++;                 # cluster size
#       }
#     }
#     if ($csz > $maxsize) {    # increasing cluster size
#       $maxsizeold = $maxsize;
#       $maxsize = $csz;
#     } elsif ($csz != 0 && $maxsize > 1 && $csz < $maxsize) { # decreasing again
#       push @clusters, $maxsize;
#       @clids = ();
#       for ($c=($a-1);$c<($a+9);$c++) { # remember gene list of last high $csz
#       push @clids, $orderedgenes[$c];
#       }
#       $clusterids{$a} = [@clids];
#       $maxsize = 0;
#     }
#   }
  for ($a=0;$a<=($#orderedgenes-9);$a++) {  # start sliding window of size 10
    $csz = 0;
    for ($b=$a;$b<($a+10);$b++) { # evaluate for each site in window
      # whether contained in random list
      if ($thymdiff{ $orderedgenes[$b] } == 1) {
        $csz++;                 # cluster size
      }
    }
    if ($csz > $maxsize) {      # increasing cluster size
      $maxsizeold = $maxsize;
      $maxsize = $csz;
    } elsif ($csz != 0 && $maxsize > 1 && $csz <= $maxsize) { # decreasing again
      @clids = ();
      for ($c=($a-1);$c<($a+9);$c++) { # remember gene list of last high $csz
        push @clids, $orderedgenes[$c] if (defined $thymdiff{$orderedgenes[$c]});
      }
      ### check whether cluster exists less than 10 genes before
      if (defined (keys %clusterids) && $a < (max(keys %clusterids)+11)) {
        $lastindex = max (keys %clusterids);
        @newlist = ();
        @newlist = unite( @{$clusterids{$lastindex}}, @clids );
        delete $clusterids{$lastindex};
        $clusterids{$a-1} = [@newlist];
        pop  @clusters; # removes last element
        push @clusters, scalar(@newlist);
        $maxsize = 0;
      } else {
        $clusterids{$a-1} = [@clids];
        push @clusters, $maxsize;
        $maxsize = 0;
      }
    }
  }

  $resultsperchr{$chr} = {      # save ererything as hash of hashes (hash of arrays)
                          clusters => [ @clusters ],
                          clusterids => { %clusterids }
                         };
}

# $doub = 0;                    # count doublets, triplets etc.: initialize
# $trip = 0;
# $quad = 0;
```

```
# $quint = 0;
# $hex = 0;
# $hept = 0;
# $oct = 0;

foreach $chr (@chrs) {
  print "Results for chromosome $chr:\n";
  print "cluster sizes:\t@{ $resultsperchr{$chr}{clusters} }\n";
  foreach $k (sort {$a<=>$b} keys ( %{$resultsperchr{$chr}{clusterids}} )) {
    print "Cluster:";
    foreach $id_c (@{ $resultsperchr{$chr}{clusterids}{$k} }) {
      print "\t$id_c" if (defined $thymdiff{ $id_c });
    }
    print "\n";
  }
  print "\n\n";
}

foreach $chr (@chrs) {
  foreach $si ( @{ $resultsperchr{$chr}{clusters} } ) {
#     if ($si == 2) {
#       $doub++;
#     } elsif ($si == 3) {
#       $trip++;
#     } elsif ($si == 4) {
#       $quad++;
#     } elsif ($si == 5) {
#       $quint++
#     } elsif ($si == 6) {
#       $hex++;
#     } elsif ($si == 7) {
#       $hept++;
#     } elsif ($si == 8) {
#       $oct++;
#     }
#   }
    if (defined $counthash{$si}) {
      $counthash{$si}++;
    } else {
      $counthash{$si} = 1;
    }
  }
}

print "Distribution of cluster sizes in thymus differential genes list\n\n";

@sizes = sort {$a <=> $b} keys %counthash;
print "$sizes[0]";
for $i (1..$#sizes) {
  print "\t$sizes[$i]";
}
print "\n";
print "$counthash{$sizes[0]}";
for $i (1..$#sizes) {
  print "\t$counthash{$sizes[$i]}";
}
print "\n";

# end;
```

1.1.7 Chromosomal Clustering, 10 Gene Window Method of 1000 Random Gene Lists

This program **start_perl_batch_mouse.sh** calclulates the chromosomal clustering of 1000 random gene lists of the same length as the target gene list from the program above.

```
#!/bin/sh
cd /home/dinkelac/data/mouse/clustering/validate_ntuples_mouse_3x
# PERL5LIB=/home/bbrors/amd64_software/lib/myperl/lib:/home/bbrors/amd64_software/lib/myperl:$PER
# export PERL5LIB
#length of genlist -l
perl validate_ntuples_mouse.pl -l 4118 > results.txt
```

1.1.8 Chromosomal Clustering, 10 Gene Window Method of 1000 Random Gene Lists

This program **validate_ntuples_mouse.pl** calculates the chromsomal clustering of random gene lists of the same length as the target gene list. It is called 1000 times by the function above.

```
#!/usr/bin/perl

#Dieses Programm berechnet die Chromosomale Dichte einer Liste von #Genen.
#
#Input: chrhash.txt, eine Tabelle mit GeneID, Chromosome, Startsite
#z.B.: NM_008020         19 6969975
#
#take gene list; calculate number of doublets, triplets etc. in
#sliding 10-gene window;

#calculate 1000-fold permutation statistic for these numbers
#Output: results_validate_ntuples.txt a list with douplets, triplets, #etc.

print "job started\n";

use Getopt::Std;
getopt('l');
$lgenlist = $opt_l; # length of genelist; needs to be fixed later

# generate random sequence of length $k from sequence of length $n
# WITHOUT replacement

use Math::Random::MT qw(rand, srand);

srand(time ^ $$ ^ unpack "%32L*", `ps axww | gzip`);

sub randSample {
  my ($n, $k) = @_;
  my $i, $j;
  my @x, @y;
  for ($i=0;$i<$n;$i++) {
    $x[$i] = $i;
  }
```

```
  for ($i=0;$i<$k;$i++) {
    $j = int($n * rand());
    $y[$i] = $x[$j] + 1;
    $x[$j] = $x[--$n];
  }
  return @y;
}

sub unite {
  my $item = "";
  my %hash = ();
  while ($item = shift) {
    $hash{$item} = 1;
  }
  return (sort keys %hash);
}

sub max {
  my $max = shift (@_);
  foreach $foo (@_) {
    $max = $foo if ($max < $foo);
  }
  return $max;
}

# reading in table affy_id chromosome TSS
print "hashtable einlesen\n";
open CHRHASH, "chrhash.txt" or
  die "error while opening chrhash.txt: $!";

while (<CHRHASH>) {
  @l = split /\t/;
  #Zeilen mit NAs rauswerfen
  next if ($l[1]=~/NA/ || $l[2]=~/NA/);
  $chrhash{$l[0]} = $l[1]; # affy_id -> chromosome
  $sitehash{$l[0]} = $l[2]; # affy_id -> tss
}
print "hashtable eingelesen\n";
close CHRHASH;

@affyids = sort (keys %chrhash);
print "affyids sortieren\n";
# @doublets = ();
# @triplets = ();
# @quadruplets = ();
# @quintuplets = ();
# @hexatuplets = ();
# @heptatuplets = ();
# @octatuplets = ();

@count_ktuples = ();
print "count ktuples\n";
print "get genes per chromosome";
print "results speichern";
for $m (1 .. 1000) {

  # produce random gene lists (as hashes);
  %genlist = ();
  @rndindex = &randSample(scalar(@affyids), $lgenlist) ;
  foreach $idx (@rndindex) {
    $genlist{ $affyids[$idx] } = 1;
  }
```

```
#Chromosomen
  @chrs = ("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12",
           "13", "14", "15", "16", "17", "18", "19",
           "X", "Y");

  foreach $chr (@chrs) { # get genes per chromosome

    @genesperchr = ();
    foreach $id (@affyids) {
      if ($chrhash{ $id } eq $chr && $sitehash{ $id }!~/NA/) {
        push @genesperchr, $id;
      }
    }
    # order by TSS
    @orderedgenes = sort {$sitehash{$a} <=> $sitehash{$b}} @genesperchr;
    @clusters = ();
    %clusterids = ();
    $maxsizeold = 0;
    $maxsize = 0;
    for ($a=0;$a<=($#orderedgenes-9);$a++) { # start sliding window of size 10
      $csz = 0;
      for ($b=$a;$b<($a+10);$b++) { # evaluate for each site in window
        # whether contained in random list
        if ($genlist{ $orderedgenes[$b] } == 1) {
          $csz++; # cluster size
        }
      }
      if ($csz > $maxsize) { # increasing cluster size
        $maxsizeold = $maxsize;
        $maxsize = $csz;
      }
      elsif ($csz != 0 && $maxsize > 1 && $csz < $maxsize) { # decreasing again
        @clids = ();
        for ($c=($a-1);$c<($a+9);$c++) { # remember gene list of last high $csz
          push @clids, $orderedgenes[$c] if (defined $genlist{$orderedgenes[$c]});#$thymdiff war
        }
        ### check whether cluster exists less than 10 genes before
        if (defined (keys %clusterids) && $a < (max(keys %clusterids)+11)) {
          $lastindex = max (keys %clusterids);
          @newlist = ();
          @newlist = unite( @{$clusterids{$lastindex}}, @clids );
          delete $clusterids{$lastindex};
          $clusterids{$a-1} = [@newlist];
          pop  @clusters; # removes last element
          push @clusters, scalar(@newlist);
          $maxsize = 0;
        } else {
          $clusterids{$a-1} = [@clids];
          push @clusters, $maxsize;
          $maxsize = 0;
        }
      }
    }

    $resultsperchr{$chr} = { # save erverything as hash of hashes (hash of arrays)

                             clusters => [ @clusters ],
                             clusterids => { %clusterids }
                            };
  }
```

```
#   $doub = 0; # count doublets, triplets etc.: initialize
#   $trip = 0;
#   $quad = 0;
#   $quint = 0;
#   $hex = 0;
#   $hept = 0;
#   $oct = 0;


  foreach $chrom (@chrs) {
    foreach $clus (@{ $resultsperchr{$chrom}{'clusters'} }) {
      if (defined ${$count_ktuples[$m]}{$clus} ){
        ${$count_ktuples[$m]}{$clus}++;
      } else {
        ${$count_ktuples[$m]}{$clus} = 1;
      }
    }
  }
  for $z (2 .. max(keys %{$count_ktuples[$m]}) ) {
    ${$count_ktuples[$m]}{$z} = 0 if (! defined ${$count_ktuples[$m]}{$z});
  }
}

for $w (1 .. 1000) {
  print "${$count_ktuples[$w]}{2}";
  $mm = max(keys %{$count_ktuples[$w]});
  if ($mm > 2) {
    for $ii (3 .. $mm) {
      print "\t${$count_ktuples[$w]}{$ii}";
    }
    print "\n";
  } else {
    print "\n";
  }
}
print "job beendet";
# end;
```

1.1.9 Chromosomal Clustering, Sliding Gene Window of Fixed Size

This program **permute_chrloc.R** calculates the number of neighbors in a sliding gene window of fixed size.

```
setwd("/home/dinkelac/data/mouse/neighbors/mouse.2014.3x")
load("mydata.mouse.3x.rda")
print("data loaded")

pairs.in.50k = vector(length=1000)
pairs.in.100k = vector(length=1000)
pairs.in.200k = vector(length=1000)
pairs.in.500k = vector(length=1000)
pairs.in.800k = vector(length=1000)
pairs.in.2m = vector(length=1000)
pairs.in.5m = vector(length=1000)

l.orig = length(tra.genes.index)
l.total = length(chip.genes)
```

```
for (i in 1:1000) {
  index.rnd.act = sample(1:l.total, l.orig, replace=F)
  dist.mat.rnd = dist.genloc(index.rnd.act,index.rnd.act)
  pairs.in.50k[i] = sum(dist.mat.rnd[upper.tri(dist.mat.rnd)] < 50000,na.rm=T)
  pairs.in.100k[i] = sum(dist.mat.rnd[upper.tri(dist.mat.rnd)] < 100000,na.rm=T)
  pairs.in.200k[i] = sum(dist.mat.rnd[upper.tri(dist.mat.rnd)] < 200000,na.rm=T)
  pairs.in.500k[i] = sum(dist.mat.rnd[upper.tri(dist.mat.rnd)] < 500000,na.rm=T)
  pairs.in.800k[i] = sum(dist.mat.rnd[upper.tri(dist.mat.rnd)] < 800000,na.rm=T)
  pairs.in.2m[i] = sum(dist.mat.rnd[upper.tri(dist.mat.rnd)] < 2e6,na.rm=T)
  pairs.in.5m[i] = sum(dist.mat.rnd[upper.tri(dist.mat.rnd)] < 5e6,na.rm=T)
  if (i %% 10 == 0)
    cat("#")
  if (i %% 100 == 0)
     cat(".")
}

save(pairs.in.50k,pairs.in.100k,pairs.in.200k,pairs.in.500k,pairs.in.800k,
pairs.in.2m,pairs.in.5m,file="results_permute_chrloc_1000.rda")
```

1.1.10 Chromosomal Clustering, Sliding Gene Window of Fixed Size for 1000 Random Gene lists

This program **start_R_batch.sh** calculates the sliding gene window of fixed size for 1000 random gene lists.

1.1.11 Different plots of TRA clusters

This script plots the different TRA clusters.

```
#cluster_plots_gngnf1.R
#--------------------
#function
#--------
source("~/R-functions/print.cluster1.R")

#1. table of clustered TRAs
#--------------------------
setwd("/home/dinkelac/data/mouse/tables/")
mouse.tras=read.csv(file="table.gngnf1.tra.index10.5x.cluster.csv",sep="\t")

#2. table of all genes
#---------------------
setwd("/home/dinkelac/data/mouse/tables/")
#chrhash=read.table("gngnf1.chrhash.all.genes.txt",sep="\t")

#3. some numbers
#---------------
#clusters on chromosomes
chromosomes=as.character(unique(mouse.tras[,3]))
nr.chromosomes=length(chromosomes)

#4. for each chromosome
#----------------------
no.of.clusters=vector(len=length(chromosomes))
max.clustersize=vector(len=length(chromosomes))
cluster.startsides=vector(len=length(chromosomes))
```

```
cluster.tissues=vector(len=length(chromosomes))
max.bp=vector(len=length(chromosomes))
cluster.sizes=vector(len=length(chromosomes))

for(i in 1:length(chromosomes)){
chromosome=chromosomes[i]
index=which(mouse.tras[,3]==chromosome)
clustered.tras.on.chromosome=mouse.tras[index,]
cluster.no=unique(clustered.tras.on.chromosome[,8])
cluster.size=clustered.tras.on.chromosome[,7]

#fuer jedes cluster pro chromosome
cluster.startside=vector(len=length(cluster.no))
cluster.tissue=vector(len=length(cluster.no))

for(k in 1:length(cluster.no)){
index=which(clustered.tras.on.chromosome[,8]==k)
cluster.startside[k]=paste(clustered.tras.on.chromosome[index,4],collapse=",")
cluster.tissue[k]=paste(as.character(clustered.tras.on.chromosome[index,10]),collapse=",")
}

cluster=cluster.size[1]
for(j in 2:length(cluster.size)){
if(cluster.size[(j-1)]!=cluster.size[j]){
cluster=c(cluster,cluster.size[j])
}
}

cluster.sizes[i]=paste(cluster,collapse=",")
cluster.startsides[i]=paste(cluster.startside,collapse="//")
cluster.tissues[i]=paste(cluster.tissue,collapse="//")
no.of.clusters[i]=length(cluster.no)
max.clustersize[i]=max(cluster.size)
#index1=which(clustered.tras.on.chromosome[,7]==min.clustersize[i])
index2=which(clustered.tras.on.chromosome[,7]==max.clustersize[i])
max.bp[i]=max(clustered.tras.on.chromosome[index2,4])-min(clustered.tras.on.chromosome[index2,4])
}

chrom.table1=cbind(chromosomes,no.of.clusters,cluster.sizes,max.clustersize,max.bp)

#cluster table
setwd("/home/dinkelac/data/mouse/tables/")
write.csv(chrom.table1,file="gngnf1.chrom.cluster.tra.index10.5x.csv")

setwd("/home/dinkelac/data/mouse/sessions/rda/")
save.image(file="tra.clusters.gngnf1.10.5x.rda")

#5. Plot: Number of TRA clusters per chromosome
#--------
chrom.table2=no.of.clusters
names(chrom.table2)=chromosomes
dimnames(chrom.table2)=c("chromosomes","no of clusters")

barplot(chrom.table2,col=c("red"),cex.names=0.7,ylim=c(0,25),xlab="chromosome",
ylab="no of TRA clusters")
abline(h=c(5,10,15,20),lty=3)
title(main="Number of TRA clusters per Chromosome")
title(sub="(gngnf1 - tra.index10.5x)")

setwd("/home/dinkelac/data/mouse/plots/")
dev.copy2eps(file="gngnf1.clusters.per.chromosome.10.5x.eps")
```

```
#6. plot clusters on all chromosomes
#-------------------------------------
maximum=0
for(i in 1:length(chromosomes)){
cluster=as.numeric(unlist(strsplit(cluster.sizes[i],split=",")))
summe=sum(cluster)+(no.of.clusters[i]*3)
maximum=max(maximum,summe)
}

x11(h=7,w=10)

y1=c(length(chromosomes):1)
y=y1*2
plot(y,axes=F,col="white",xlim=c(0,maximum),xlab="",ylab="")
abline(h=y)
title(main="TRA clusters on chromosomes")
title(sub="gngnf1-tra.index10.5x")
mtext("chromosome",side=2,at=42,line=-1,las=2)

for(i in 1:length(chromosomes)){
cluster=as.numeric(unlist(strsplit(cluster.sizes[i],split=",")))
a=1
for(j in 1:length(cluster)){
b=a+cluster[j]
rect(a,y[i]-0.3,b,y[i]+0.3, col="red", border="black")
text(mean(c(a,b)),y[i]+0.9,cluster[j],cex=0.7)
a=b+3
}
mtext(chromosomes[i],side=2,at=y[i],line=2,las=2)
}

setwd("/home/dinkelac/data/mouse/plots/")
dev.copy2eps(file="gngnf1.clusters.on.chromosomes.tra.index10.5x.eps")

#7. number of significant cluster
clusters=as.numeric(unlist(strsplit(unlist(cluster.sizes),split=",")))
clusters=sort(clusters)
sig.clusters=clusters[61:length(clusters)]
length(sig.clusters)

#8. plot clusters on each chromosome
setwd("/home/dinkelac/data/mouse/plots/")
x11(h=4,w=10)

for (i in 1:length(chromosomes)){

#fuer jedes chromosom i
chromosome=chromosomes[i]
index=which(tras[,3]==chromosome)
clustered.tras.on.chromosome=tras[index,]

cluster.no=unique(clustered.tras.on.chromosome[,8])
clustersize=as.numeric(unlist(strsplit(cluster.sizes[i],split=",")))
#startsides fuer alle cluster auf chromosom i mit //getrennt
cluster.startsides[i]
#startsides pro cluster [[1]][1]
startsides.per.cluster=strsplit(cluster.startsides[i],split="//")

#alle startsides auf dem chromosom
all.startsides=c()
for(j in 1:length(cluster.no)){
all.startsides=c(all.startsides,unlist(strsplit(startsides.per.cluster[[1]][j],split=",")))
```

```
}

startsides=as.numeric(all.startsides)

a=min(startsides)
b=max(startsides)

y=0
plot(y,axes=F,col="white",xlim=c(a,b),xlab="",ylab="")
abline(h=y)
title(main=paste("TRA clusters on chromosome",chromosome))
title(sub="gngnf1-tra.index9.5x")

e=c()
for(j in 1:length(cluster.no)){
starts=as.numeric(unlist(strsplit(startsides.per.cluster[[1]][j],split=",")))
c=min(starts)
d=max(starts)
e=c(e,c,d)
rect(c,-0.3,d,+0.3,col="red",border="black")
text(mean(c(c,d)),+.5,clustersize[j],cex=0.7)
}
#value angeben
f=relax(e,value)
text(f,-.7,e,cex=0.7,srt=90)

#------------------------
relax=function(f,value){
f=sort(f)
for(i in 2:length(f)){
if((f[i]-f[i-1])<value){
f[i]=f[i-1]+value
}
}
return(f)
}
#-------------------------

dev.copy2eps(file=paste("gngnf1.cluster.on.chromosome.",chromosome,".tra.index9.5x.eps",
sep=""))
}

#8. Alle Cluster auf einem Chromosom mit farben fuer tisses plotten

setwd("/home/dinkelac/data/mouse/sessions/rda/")
load("mean.vsnrma.gngnf1_1.rda")

setwd("/home/dinkelac/data/mouse/plots/")
x11(h=4,w=10)

#8.1 color tissue zuordnung
all.tissues=names(mean.vsnrma[1,])

tissue.order=c(6,7,31,8,51,21,60,43,29,30,22,37,34,1,12,13,11,14,39,19,61,24,40,41,20,27,
26,36,25,45,46,47,48,52,53,17,54,56,50,58,49,4,9,10,32,2,44,42,28,15,33,16,59,23,38,5,55,
35,57,18,3)

all.tissues=all.tissues[tissue.order]

farben=c("darkred","red3","peru","tomato2","#FF5600FF","tan","#FF8100FF","orange","wheat",
"yellow","#FFED00FF","#FFED00FF","green","green","green","green","green","green","green",
"green","green","green","green","green","seagreen3","seagreen3","seagreen3",
```

```
"seagreen3","white","white","white","white","white","bisque1","bisque1","bisque1","bisque1",
"#00F4FFFF","#00F4FFFF","lightblue","#00B4FFFF","lightblue","#005DFFFF","steelblue",
"#0032FFFF","navy","#0007FFFF","#0E00FFFF","seashell2","#3900FFFF","rosybrown","purple",
"thistle","#9000FFFF","slateblue","#BB00FFFF","mistyrose","violet","oldlace","darkblue")

tissue.colors=farben
names(tissue.colors)=all.tissues

setwd("/home/dinkelac/data/mouse/tables/")
write.table(tissue.colors,file="tissue.colors.gngnf1.csv",sep=",")

setwd("/home/dinkelac/data/mouse/sessions/rda/")
save.image(file="tra.clusters.gngnf1.rda")

#8.2 plot all clusters on each chromosome with tissue colors
#-------------------------------------------------------------
setwd("/home/dinkelac/data/mouse/plots/")
x11(h=7,w=10)

for (i in 1:length(chromosomes)){

#fuer jedes chromosom i
chromosome=chromosomes[i]
index=which(tras[,3]==chromosome)
clustered.tras.on.chromosome=tras[index,]

cluster.no=unique(clustered.tras.on.chromosome[,8])
clustersize=as.numeric(unlist(strsplit(cluster.sizes[i],split=",")))
#startsides fuer alle cluster auf chromosom i mit //getrennt
#startsides pro cluster [[1]][1]
tissue=as.character(clustered.tras.on.chromosome[,10])

startsides.per.cluster=strsplit(cluster.startsides[i],split="//")
tissues.per.cluster=strsplit(cluster.tissues[i],split="//")

#alle startsides auf dem chromosom
all.startsides=c()
for(j in 1:length(cluster.no)){
all.startsides=c(all.startsides,unlist(strsplit(startsides.per.cluster[[1]][j],split=",")))
}

#?
tissues.used=c()
for(j in 1:length(cluster.no)){
tissues.used=c(tissues.used,unlist(strsplit(tissues.per.cluster[[1]][j],split=",")))
}

unique.tissues=sort(unique(tissues.used))
colors.act=tissue.colors[unique.tissues]

startsides=as.numeric(all.startsides)

a=min(startsides)
b=max(startsides)

y=0
plot(y,axes=F,col="white",ylim=c(-3,1),xlim=c(a,b),xlab="",ylab="")
abline(h=y)
title(main=paste("TRA clusters on chromosome",chromosome))
title(sub="gngnf1-tra.index9.5x")

e=c()
```

```
for(j in 1:length(cluster.no)){
#for(j in 2:2){ #test

starts=as.numeric(unlist(strsplit(startsides.per.cluster[[1]][j],split=",")))
tissues=unlist(strsplit(tissues.per.cluster[[1]][j],split=","))
colors=tissue.colors[tissues]
c=min(starts)
d=max(starts)
width=(d-c)/length(tissues)
e=c(e,c,d)
#colored tissues
x=c
for(k in 1:length(tissues)){
rect(x,-0.3,(x+width),0.3,col=colors[k],border=colors[k])
x=x+width
}

rect(c,y-0.3,d,y+0.3,border="black")

text(mean(c(c,d)),y+.5,clustersize[j],cex=0.7)
}

legend("bottom",pch=22,names(colors.act),pt.bg=colors.act,col=colors.act,ncol=4,cex=0.8)

#value=2500000, chromosome c(1,2,3,6,13,16,X)
#value=1000000, chromosome c(15,17,18,19)
#value=chromosome c(4,5,7,8,9,10,11,12,14)
value=2000000
#value angeben
f=relax(e,value)
text(f,y-.7,e,cex=0.6,srt=90)

relax=function(f,value){
f=sort(f)
for(i in 2:length(f)){
if((f[i]-f[i-1])<value){
f[i]=f[i-1]+value
}
}
return(f)
}

dev.copy2eps(file=paste("gngnf1.color.cluster.on.chromosome.",chromosome,".tra.index9.5x.eps",
sep=""))
}

#8.3 plot each cluster by itself
#--------------------------------
setwd("/home/dinkelac/data/mouse/plots/")
x11(h=7,w=10)

#fuer jedes chromosom
for (i in 1:length(chromosomes)){
#for (i in 14){

chromosome=chromosomes[i]

index1=which(mouse.tras[,3]==chromosome)
clustered.tras.on.chromosome=mouse.tras[index1,]
cluster.no=unique(clustered.tras.on.chromosome[,8])

#j schleife
```

```
for(j in 1:length(cluster.no)){
#for(j in 2){

cluster=cluster.no[j]

index2=which(mouse.tras[,3]==chromosome&mouse.tras[,8]==cluster)
clustered.tras.on.cluster=mouse.tras[index2,]

#clustersize
clustersize=length(rownames(clustered.tras.on.cluster))

if(clustersize>29){

#filename=paste("gngnf1.cluster.",chromosome,".",cluster,".",clustersize,".png",sep="")

#pdf(file=filename,width=1000,height=500)

#tissues
max.tissues=as.character(clustered.tras.on.cluster[,10])
unique.max.tissues.on.cluster=sort(unique(max.tissues))

max.colors=tissue.colors[max.tissues]
legend.colors=tissue.colors[unique.max.tissues.on.cluster]

startsides=clustered.tras.on.cluster[,4]
symbols=clustered.tras.on.cluster[,2]
startsides.sorted=sort(startsides)
a=min(startsides)
b=max(startsides)

#basisplot
#---------
y=0
plot(y,axes=F,col="white",ylim=c(-3,2),xlim=c(a,b),xlab="",ylab="")
abline(h=y)
title(main=paste("TRA clusters on chromosome",chromosome,", cluster ",cluster))
title(sub="gngnf1-tra.index10.5x")

#color width
width=(b-a)/clustersize
x=a
z=a

#plot tissue colors
for(k in 1:length(max.tissues)){
rect(x,-0.3,(x+width),0.3,col=max.colors[k],border=max.colors[k])
x=x+width
z=c(z,x)
}
z=z[1:length(z)-1]
#plot box
rect(a,y-0.3,b,y+0.3,border="black")
#text(mean(c(a,b)),y+.5,clustersize,cex=0.7)
text(mean(c(a,b)),y-1.3,paste(clustersize," TRAs"),cex=0.9)

#legend
legend("bottom",pch=22,names(legend.colors),pt.bg=legend.colors,col=legend.colors,
ncol=4,cex=0.8)
#legend("bottom",pch=22,names(legend.colors),pt.bg=legend.colors,col=legend.colors,
#ncol=4,cex=1)

#plot startsides
```

```
writing=60
if(clustersize>59){
writing=90
}

#adjusted to gene symbols
text(z,y-0.7,symbols,cex=0.6,srt=90)
#text(z,y-0.7,startsides,cex=0.6,srt=90)
text(z,y+0.5,max.tissues,cex=0.7,srt=writing,adj=0)
#text(z,y-0.7,startsides,cex=0.8,srt=90)

#dev.off()

#save.plot
dev.copy2eps(file=paste("gngnf1.tra.index10.5x.cluster.",chromosome,".",cluster,".",
clustersize,".eps",sep=""))
}#end of if clustersize>29
}#end of cluster
print(i)
}#end of chromosome

#9. Plot TRA clusters per chromosome
#-----------------------------------
#lies tabelle fuer alle Gene mit startsides ein
#a. alle auf dem chip

#15859
gngnf1_chrhash=read.table("/home/dinkelac/data/mouse/tables/gngnf1.chrhash.all.genes.txt",
sep="\t")

#31804
#b. alle auf ensembl
ensembl.mouse_chrhash=
read.table("/home/dinkelac/data/mouse/tables/ensemble.chrhash.all.genes.txt",sep="\t")

setwd("/home/dinkelac/data/mouse/plots/")
x11(h=7,w=10)

min=min(gngnf1_chrhash[,3])
max=max(gngnf1_chrhash[,3])

#basisplot
y=0
y1=c(20:1)
y2=y1*3

plot(y,xlim=c(min,max),xlab="",ylab="",ylim=c(0,60),axes=F,col="white")
abline(h=y2)
title(main="clustered TRAs on chromosomes")
title(sub="gngnf1-tra.index9.5x")

mtext("chromosome",side=2,at=63,line=-1,las=2)

for(i in 1:length(chromosomes)){
chromosome=chromosomes[i]
print(chromosome)

index1=which(tras[,3]==chromosome)
clustered.tras.on.chromosome=tras[index1,]

index1a=which(gngnf1_chrhash[,2]==chromosome)
gngnf1.genes.on.chromosome=gngnf1_chrhash[index1a,]
```

```
index1b=which(ensembl.mouse_chrhash[,2]==chromosome)
ensembl.genes.on.chromosome=ensembl.mouse_chrhash[index1b,]

tra.startsides=clustered.tras.on.chromosome[,4]
gngnf1.startsides=gngnf1.genes.on.chromosome[,3]
ensembl.startsides=ensembl.genes.on.chromosome[,3]

min.ensembl=min(ensembl.startsides)
max.ensembl=max(ensembl.startsides)

min.gngnf1=min(gngnf1.startsides)
max.gngnf1=max(gngnf1.startsides)

#min=min.ensembl
#max=max.ensembl

#cluster enden
cluster.no=unique(clustered.tras.on.chromosome[,8])
cluster.size=unique(clustered.tras.on.chromosome[,7])

cluster.min=vector(len=length(cluster.no))
cluster.max=vector(len=length(cluster.no))
cluster.size=vector(len=length(cluster.no))

for(j in 1:length(cluster.no)){
index2a=which(clustered.tras.on.chromosome[,8]==j)
cluster.startside=clustered.tras.on.chromosome[index2a,4]
cluster.min[j]=min(cluster.startside)
cluster.max[j]=max(cluster.startside)
}

mtext(chromosome,side=2,at=y2[i],line=2,las=2)

#plot all genes
x=sort(c(gngnf1.startsides,gngnf1.startsides))
k=1
while(k<=length(x)){
lines(c(x[k],x[k+1]),c(y2[i]-1,y2[i]+1),type="l",col="blue")
k=k+2
}

#plot TRAs
y=sort(c(tra.startsides,tra.startsides))
k=1
while(k<=length(y)){
lines(c(y[k],y[k+1]),c(y2[i]-1,y2[i]+1),type="l",col="red")
k=k+2
}

}#chromosom ende

setwd("/home/dinkelac/data/mouse/plots/")
dev.copy2eps(file="gngnf1.tras.on.chromosomes.eps")

#10. plot each chromosome alone
#---------------------------
x11(h=4,w=10)

for(i in 1:length(chromosomes)){
chromosome=chromosomes[i]
print(chromosome)
```

```
index1=which(tras[,3]==chromosome)
clustered.tras.on.chromosome=tras[index1,]

index1a=which(gngnf1_chrhash[,2]==chromosome)
gngnf1.genes.on.chromosome=gngnf1_chrhash[index1a,]

index1b=which(ensembl.mouse_chrhash[,2]==chromosome)
ensembl.genes.on.chromosome=ensembl.mouse_chrhash[index1b,]

tra.startsides=clustered.tras.on.chromosome[,4]
gngnf1.startsides=gngnf1.genes.on.chromosome[,3]
ensembl.startsides=ensembl.genes.on.chromosome[,3]

min.ensembl=min(ensembl.startsides)
max.ensembl=max(ensembl.startsides)

min.gngnf1=min(gngnf1.startsides)
max.gngnf1=max(gngnf1.startsides)

min=min.gngnf1
max=max.gngnf1

#basisplot
y=0

plot(y,xlim=c(min,max),xlab="",ylab="",ylim=c(-5,5),axes=F,col="white")
abline(h=y)
title(main=paste("clustered TRAs on chromosome ",chromosome))
title(sub="gngnf1-tra.index9.5x")

#cluster enden
cluster.no=unique(clustered.tras.on.chromosome[,8])
cluster.size=unique(clustered.tras.on.chromosome[,7])

cluster.min=vector(len=length(cluster.no))
cluster.max=vector(len=length(cluster.no))
#cluster.size=vector(len=length(cluster.no))

for(j in 1:length(cluster.no)){
index2a=which(clustered.tras.on.chromosome[,8]==j)
cluster.startside=clustered.tras.on.chromosome[index2a,4]
cluster.min[j]=min(cluster.startside)
cluster.max[j]=max(cluster.startside)
}

#plot all genes
x=sort(c(gngnf1.startsides,gngnf1.startsides))
k=1
while(k<=length(x)){
lines(c(x[k],x[k+1]),c(y-1,y+1),type="l",col="blue")
k=k+2
}

#plot TRAs
x=sort(c(tra.startsides,tra.startsides))
k=1
while(k<=length(x)){
lines(c(x[k],x[k+1]),c(y-1,y+1),type="l",col="red")
k=k+2
}
```

```
value=1000000

#cluster als horizontale linie einzeichnen
for(j in 1:length(cluster.no)){
lines(c(cluster.min[j],cluster.max[j]),c(y+2,y+2),type="l",col="red")
lines(c(cluster.min[j],cluster.min[j]),c(y+1.7,y+2.3),type="l",col="red")
lines(c(cluster.max[j],cluster.max[j]),c(y+1.7,y+2.3),type="l",col="red")
text(mean(c(cluster.min[j],cluster.max[j])),y+2.5,cluster.size[j],cex=0.9)
text(cluster.min[j]+value,y-3.8,cluster.min[j],cex=0.6,srt=90)
text(cluster.max[j]-value,y-3.8,cluster.max[j],cex=0.6,srt=90)
}

setwd("/home/dinkelac/data/mouse/plots/")
dev.copy2eps(file=paste("gngnf1.clustered.tras.chrom.",chromosome,".eps",))

}#chromosom ende

#11. Plot each cluster with non TRAs
#-------------------------------------
#information from above, for each cluster, print cluster
#with startside ,symbol of tras, tissue
#save cluster

x11(h=5,w=10)

for(i in 1:length(chromosomes)){
chromosome=chromosomes[i]
print(chromosome)

index1=which(tras[,3]==chromosome)
clustered.tras.on.chromosome=tras[index1,]

index1a=which(gngnf1_chrhash[,2]==chromosome)
gngnf1.genes.on.chromosome=gngnf1_chrhash[index1a,]

index1b=which(ensembl.mouse_chrhash[,2]==chromosome)
ensembl.genes.on.chromosome=ensembl.mouse_chrhash[index1b,]

tra.startsides=clustered.tras.on.chromosome[,4]
gngnf1.startsides=gngnf1.genes.on.chromosome[,3]
ensembl.startsides=ensembl.genes.on.chromosome[,3]

min.ensembl=min(ensembl.startsides)
max.ensembl=max(ensembl.startsides)

min.gngnf1=min(gngnf1.startsides)
max.gngnf1=max(gngnf1.startsides)

#cluster enden
cluster.no=unique(clustered.tras.on.chromosome[,8])
cluster.size=unique(clustered.tras.on.chromosome[,7])

cluster.min=vector(len=length(cluster.no))
cluster.max=vector(len=length(cluster.no))
#cluster.size=vector(len=length(cluster.no))

for(j in 1:length(cluster.no)){#anfang cluster
print(cluster.no[j])
print(cluster.size[j])
index2a=which(clustered.tras.on.chromosome[,8]==j)
cluster.startside=clustered.tras.on.chromosome[index2a,4]
#cluster.min[j]=min(cluster.startside)
```

```
#cluster.max[j]=max(cluster.startside)

cluster.tras=clustered.tras.on.chromosome[index2a,]
tra.startsides=cluster.tras[,4]

min=min(cluster.startside)
max=max(cluster.startside)

#basisplot
y=0
plot(y,xlim=c(min,max),xlab="",ylab="",ylim=c(-12,7),axes=F,col="white")
abline(h=y)

title(main=paste("TRA cluster",cluster.no[j],"chromosome",chromosome,"\n",cluster.size[j],
"TRAs"))
title(sub="gngnf1-tra.index9.5x")

#plot all genes
index2=which(gngnf1.startsides>=min&gngnf1.startsides<=max)
cluster.startsides=sort(gngnf1.startsides[index2])

x=sort(c(cluster.startsides,cluster.startsides))

k=1
while(k<=length(x)){
lines(c(x[k],x[k+1]),c(y-1,y+1),type="l",col="black")
k=k+2
}

#plot TRAs
x1=sort(c(tra.startsides,tra.startsides))
k=1
while(k<=length(x)){
lines(c(x1[k],x1[k+1]),c(y-1,y+1),type="l",col="red")
k=k+2
}

average.distance=(max-min)/cluster.size[j]

if(cluster.size[j]<20){
distance=100000
}else if(cluster.size[j]<40){
distance=200000
}else if(cluster.size[j]<60){
distance=300000
}else{
distance=400000
}

if(distance>average.distance){
distance=average.distance
}
print(distance)

printed.startsides=relax(cluster.tras[,4],distance)

for(m in 1:length(cluster.tras[,1])){
#startside
text(printed.startsides[m],y-3.8,cluster.tras[m,4],cex=0.6,srt=90,col="red")
#symbol
text(printed.startsides[m],y+2,as.character(cluster.tras[m,2]),cex=0.6,srt=90,col="red",
adj=0)
```

```
#tissue
text(printed.startsides[m],y-6,as.character(cluster.tras[m,10]),cex=0.6,srt=90,col="black",
adj=1)
}

print("save image")
print(chromosome)
print(cluster.no[j])

setwd("/home/dinkelac/data/mouse/plots/")
dev.copy2eps(file=paste("gngnf1.chr.",chromosome,".cluster.",cluster.no[j],".eps",sep=""))

print("cluster ende")
}#ende cluster
print("chromosom ende")
}#ende chromosom

#funktioniert nur bis chrom 2, cluster 15 Error in if (cluster.size[j] < 20) { :
#missing value where TRUE/FALSE needed

#12. Correlation between TRA cluster size and kb
#-----------------------------------------------
line=vector(len=4)
table=line
for (i in 1:length(chromosomes)){
chromosome=chromosomes[i]

index1=which(mouse.tras[,3]==chromosome)
clustered.tras.on.chromosome=mouse.tras[index1,]

cluster.no=unique(clustered.tras.on.chromosome[,8])
#cluster.size=clustered.tras.on.chromosome[,7]

#cluster.min=vector(len=length(cluster.no))
#cluster.max=vector(len=length(cluster.no))
#cluster.size=vector(len=length(cluster.no))

for(j in 1:length(cluster.no)){#anfang cluster

cluster=cluster.no[j]
index2=which(clustered.tras.on.chromosome[,8]==j)
tras.in.cluster=clustered.tras.on.chromosome[index2,]
startsides=tras.in.cluster[,4]
TRAs=unique(tras.in.cluster[,7])

size=max(startsides)-min(startsides)

line=cbind(chromosome,cluster,TRAs,size)
table=rbind(table,line)
}#for cluster.no
}#for chromosomes

table1=table[-1,]

setwd("/home/dinkelac/data/mouse/tables/")
write.table(table1,file="gngnf1.tras.vs.kb_in.clusters.csv",row.names=F)

#plot the correlation between TRA no. and clustersize in kb
#----------------------------------------------------------
TRA.no=table1[,3]
size.kb=table1[,4]
```

```
corelation=cor.test(as.numeric(size.kb),as.numeric(TRA.no),method="spearman")

#coefficient=0.82
#p-value=2.2e-16

#plot(size.kb,TRA.no,
#main="Correlation of no of TRAs in clusters versus the size in kb")
scatter.smooth(size.kb,TRA.no,
main="Correlation of no of TRAs in clusters versus the size in kb")

text(5e+07,10,"spearman corelation coefficient=0.82\np-value=2.2e-16")

setwd("/home/dinkelac/data/mouse/plots/")
dev.copy2eps(file="gngnf1.index.10.5x_TRAs_vs_size.eps")
```

1.1.12 Cluster Table

This script writes out all TRAs, which are clustered.

```
#cluster_table_gngnf1.R
#-----------------------
#input: tra.index10.5x, tra.index10.10x, cluster results.txt
#annotation:Ensembl gene, genesymbol, chr, startside, strand, band, clustersize, clusternr, tissu
#output: table.tra.index10.5x.cluster.csv")

#1. TRAs einlesen
#----------------
setwd("/home/dinkelac/data/mouse/sessions/rda")
load("tras.gngnf1.rda")

#table.genes.tra.index10.5x.annotated=table.genes.tra.index10.5x.annotated[-2,]

#genesymbol
symbol=table.genes.tra.index10.5x.annotated[,6]

#chromosome
chrom=table.genes.tra.index10.5x.annotated[,2]

#startside
startside=table.genes.tra.index10.5x.annotated[,3]

#strand
strand=table.genes.tra.index10.5x.annotated[,4]

#band
band=table.genes.tra.index10.5x.annotated[,5]

#gene id
gene.ids=table.genes.tra.index10.5x.annotated[,1]

#2. TRA cluster einlesen
#-----------------------
setwd("/home/dinkelac/data/mouse/clustering/gngnf1/index_10/observed_ntuples_gngnf1_index10.5x")

x=scan(file="results.txt",what='character',sep="\n")

#clustersize leeres objekt
clustersize=gene.ids
clustersize[1:length(clustersize)]=NA
```

```
#clusterindex leeres Objekt
clusternr=gene.ids
clusternr[1:length(clusternr)]=NA
cluster.table=cbind(clustersize,clusternr)
rownames(cluster.table)=gene.ids

clusterindex=0
for (i in 1:length(x)){#zeilenweise einlesen
 zeile=x[i]
 zeile=unlist(strsplit(zeile,"\t"))
 if(zeile[1]=="Cluster:"){
  clusterindex=clusterindex+1
  for(j in 2:length(zeile)){
   cluster.table[zeile[j],1]=(length(zeile)-1)#subscription out of bounds
   cluster.table[zeile[j],2]=clusterindex
  }
 }
 else if((unlist(strsplit(zeile[1]," ")))[1]=="Results"){
 clusterindex=0
 }
}

table=as.matrix(cbind(gene.ids,symbol,chrom,startside,strand,band,cluster.table))

#numbers
#-------
no.tras.clustered=dim(table[!is.na(table[,7]),])[1]
persent.tras.clustered=no.tras.clustered*100/dim(table)[1]

#table in dem die clusternr nicht NA ist, sprich nur die geclusterten TRAs

table.clustered.tras.10.5x=table[!is.na(table[,7]),]

#tissue

#liste fuer alle transkripte, welche gewebe
#->gene.ids runterrechnen

#functions
source("~/R-functions/pasteList.R")

#sucht tissues (dauert!)
#---------------------------
tra.transcripts=names(tra.index10.5x)

tissues=list()
tiss=list()
tiss.act=list()
med.act=vector()

for (i in 1:length(tra.transcripts)){
med.act[i]=exp(median(mean.vsnrma[tra.transcripts[i],]))
tiss.act[i]=list(names(which(exp(mean.vsnrma[tra.transcripts[i],])>5*med.act[i])))
}

#maximum tissue (dauert!)
#--------------
max.value=vector()
max.tiss=list()

for (i in 1:length(tra.transcripts)){
max.value=max(mean.vsnrma[tra.transcripts[i],])
```

```
max.tiss[i]=names(which(mean.vsnrma[tra.transcripts[i],]==max.value))
}

#Function: pasteList
#--------------------
pasteList=function(list){
result=list()
        for(i in 1:length(list)){
        tiss=unlist(list[i])
        res=tiss[1]
        if(length(tiss)>1){
                for(j in 2:length(tiss)){
                res=paste(res,tiss[j], sep="/")
                }
        }
        result[i]=res
        result=unlist(result)
        }
return(result)
}

maximum.tissue=unlist(max.tiss)
names(maximum.tissue)=tra.transcripts

tissues=pasteList(tiss.act)
names(tissues)=tra.transcripts

gene.names=as.character(ensembl.59.mouse.transcripts[deaffy(tra.transcripts),1])

#evtl. doppelte?
names(tissues)=gene.names
names(maximum.tissue)=gene.names

gene.ids=rownames(table.clustered.tras.10.5x)
tissue=tissues[gene.ids]
max.tissue=maximum.tissue[gene.ids]

tabelle=cbind(table.clustered.tras.10.5x,tissue,max.tissue)

setwd("/home/dinkelac/data/mouse/tables/")
write.csv(tabelle,"table.tra.index10.5x.cluster.csv",row.names=F)

#tabelle sortieren nach a) chromosomen, b) clusternr, c) startside
```

1.1.13 R script for Errorbars

This R script can be imported to draw errorbars in barplots.

```
#errorbars.R
#-----------
#description
#-----------
# produce vertical error bars on a plot-- aliased vbars in anticipation
# of the day when I need hbars to plot horizontal error bars....
#
# error bars originate at (x, y0) and radiate to y0 + y1 and y0 - y2
#
# example (adds +- 1.0 SE to a barplot of means for data vector x):
#
#usage
```

```
#-----
# y.mean <- mean(x) # mean
# y.se <- sqrt(var(x,na.rm=TRUE)/length(x)) # std err
# z <- barplot(y.mean,plot=FALSE) # location of bars
# barplot(y.mean,...) # the real plot
# ebars(z,y.mean,y.mean+y.se,y.mean-y.se) # add error bars
#
# --Mike C.
#---------------

#Funktionen
#----------
ebars <- vbars <- function (x, y0, y1, y2)
{
  for(i in 1:length(x))
  {
    if(y0[i]!=0 & !is.na(y0[i]) & y1[i]!=0 & !is.na(y1[i]) &
      (y0[i]!=y2[i]))
    {
      arrows (x[i], y0[i], x[i], y1[i], angle=90, length=0.05)
    }
    if(y0[i]!=0 & !is.na(y0[i]) & y2[i]!=0 & !is.na(y2[i]) &
      (y0[i]!=y2[i]))
    {
      arrows (x[i], y0[i], x[i], y2[i], angle=90, length=0.05)
    }
  }
}

plot.w.ebars <- function(X, ...) {
  y.mean = apply(X, 2, mean, na.rm=T)
  y.se = sqrt(apply(X, 2, var, na.rm=T)/nrow(X))
  z <- barplot(y.mean, plot=F)
  barplot(y.mean, ...)
  for (i in 1:length(y.mean)) {
    ebars(z[i], y.mean[i], y.mean[i]+y.se[i], y.mean[i]-y.se[i])
  }
}
```

1.1.14 Aire genes in TRAs

This script is calculating the overlap of Aire genes in TRAs, the results of this calculation are in the PHD thesis of Dr. Sheena Pinto and her papers with Prof. Dr. Bruno Kyewski.

```
#map.illumina.genes.in tras.R
#-----------------------------
#-> find and show genes in TRA clusters

library(limma)

#1. Input data
#-------------
mouse.wt.upregulated=read.csv(file="/home/dinkelac/data/sheena/tables/mouse_wt/
mouse_wt.top.2532.pvalue.0.01.fc2_x_up.csv",header=TRUE,sep="\t")

mouse.wt.downregulated=read.csv(file="/home/dinkelac/data/sheena/tables/mouse_wt/
mouse_wt.top.2532.pvalue.0.01.fc2_x_down.csv",header=TRUE,sep="\t")
```

```
mouse.ko.upregulated=read.csv(file="/home/dinkelac/data/sheena/tables/mouse_ko/
mouse_ko.top.2867.pvalue.0.01.fc2_x_up.csv",header=TRUE,sep="\t")

mouse.ko.downregulated=read.csv(file="/home/dinkelac/data/sheena/tables/mouse_ko/
mouse_ko.top.2867.pvalue.0.01.fc2_x_down.csv",header=TRUE,sep="\t")

illuminaIDs.ko.up=as.character(mouse.ko.upregulated[,1])
illuminaIDs.wt.up=as.character(mouse.wt.upregulated[,1])
illuminaIDs.ko.down=as.character(mouse.ko.downregulated[,1])
illuminaIDs.wt.down=as.character(mouse.wt.downregulated[,1])

affyIDs.ko.up=get.affy.ID(illuminaIDs.ko.up,"ko")
affyIDs.ko.down=get.affy.ID(illuminaIDs.ko.down,"ko")
affyIDs.wt.up=get.affy.ID(illuminaIDs.wt.up,"wt")
affyIDs.wt.down=get.affy.ID(illuminaIDs.wt.down,"wt")

affyIDs.ko.up=unique(affyIDs.ko.up)
#1430
affyIDs.ko.down=unique(affyIDs.ko.down)
#1518
affyIDs.wt.up=unique(affyIDs.wt.up)
#1466
affyIDs.wt.down=unique(affyIDs.wt.down)
#1178

mouse.tras=read.table(file="/home/dinkelac/data/mouse/tables/tra.index7.5x.txt")
mouse.tras.affys=as.character(mouse.tras[,1])
#5852

mouse.clusters=read.csv(file="/home/dinkelac/data/mouse/tables/
mouse_clusters_tra.index.5x.csv")
mouse.clusters.affys=as.character(mouse.clusters[,1])
#4770

setwd("/home/dinkelac/data/sheena/sessions/rda")
#save.image(file="illumina.genes.in.tra.clusters.rda")

#2. Venn Diagrams
#-----------------
#2.1 tras
#--------
par(mfrow=c(2,2))

x=affyIDs.wt.up
y=mouse.tras.affys
z=union(x,y)
wt.up=1*(is.element(z,x))
TRAs=1*(is.element(z,y))
m=cbind(wt.up,TRAs)
vennDiagram(m,circle.col=c("red","blue"),lwd=3,cex=1)
title(main="Venn Diagram\n mouse wt")
title(sub="mouse wt, upregulated genes in CD80hi mTECs\n\n\n")

mtext("upregulated",side=2,line=2,cex=0.9)

x=affyIDs.ko.up
y=mouse.tras.affys
z=union(x,y)
ko.up=1*(is.element(z,x))
TRAs=1*(is.element(z,y))
m=cbind(ko.up,TRAs)
```

```
vennDiagram(m,circle.col=c("red","blue"),lwd=3,cex=1)
title(main="Venn Diagram\n mouse ko")
title(sub="mouse ko, upregulated genes in CD80hi mTECs\n\n\n")


x=affyIDs.wt.down
y=mouse.tras.affys
z=union(x,y)
wt.down=1*(is.element(z,x))
TRAs=1*(is.element(z,y))
m=cbind(wt.down,TRAs)
vennDiagram(m,circle.col=c("red","blue"),lwd=3,cex=1)
title(sub="mouse wt, downregulated genes in CD80hi mTECs\n\n\n")

mtext("downregulated",side=2,line=2,cex=0.9)

x=affyIDs.ko.down
y=mouse.tras.affys
z=union(x,y)
ko.down=1*(is.element(z,x))
TRAs=1*(is.element(z,y))
m=cbind(ko.down,TRAs)
vennDiagram(m,circle.col=c("red","blue"),lwd=3,cex=1)
title(sub="mouse ko, downregulated genes in CD80hi mTECs\n\n\n")


setwd("/home/dinkelac/data/sheena/plots/")
#dev.copy2eps(file="venn.diagram.mouse_tras.eps")

#2.2 clustered tras
#------------------
x=affyIDs.wt.up
y=mouse.clusters.affys
z=union(x,y)
wt.up=1*(is.element(z,x))
clustered.TRAs=1*(is.element(z,y))
m=cbind(wt.up,clustered.TRAs)
vennDiagram(m,circle.col=c("red","blue"),lwd=3,cex=1)
title(main="Venn Diagram\n mouse wt")
title(sub="mouse wt, upregulated genes in CD80hi mTECs\n\n\n")

mtext("upregulated",side=2,line=2,cex=0.9)

x=affyIDs.ko.up
y=mouse.clusters.affys
z=union(x,y)
ko.up=1*(is.element(z,x))
clustered.TRAs=1*(is.element(z,y))
m=cbind(ko.up,clustered.TRAs)
vennDiagram(m,circle.col=c("red","blue"),lwd=3,cex=1)
title(main="Venn Diagram\n mouse ko")
title(sub="mouse ko, upregulated genes in CD80hi mTECs\n\n\n")

x=affyIDs.wt.down
y=mouse.clusters.affys
z=union(x,y)
wt.down=1*(is.element(z,x))
clustered.TRAs=1*(is.element(z,y))
m=cbind(wt.down,clustered.TRAs)
vennDiagram(m,circle.col=c("red","blue"),lwd=3,cex=1)
title(sub="mouse wt, downregulated genes in CD80hi mTECs\n\n\n")
```

```
mtext("downregulated",side=2,line=2,cex=0.9)

x=affyIDs.ko.down
y=mouse.clusters.affys
z=union(x,y)
ko.down=1*(is.element(z,x))
clustered.TRAs=1*(is.element(z,y))
m=cbind(ko.down,clustered.TRAs)
vennDiagram(m,circle.col=c("red","blue"),lwd=3,cex=1)
title(sub="mouse ko, downregulated genes in CD80hi mTECs\n\n\n")


setwd("/home/dinkelac/data/sheena/plots/")
#dev.copy2eps(file="venn.diagram.mouse_clustered.tras.eps")

#3. in welchen clustern sind illumina gene
#-----------------------------------------
chromosomen=as.character(unique(mouse.clusters[,3]))
#262
cluster.ids=unique(paste(mouse.clusters[,3],mouse.clusters[,6],sep=","))

#3.5 save in table
#-----------------
table=matrix(ncol=7,nrow=262)
colnames(table)=c("mouse_chrom","cluster_nr","clustersize","wt.up","wt.down",
"ko.up","ko.down")

for(i in 1:length(cluster.ids)){
a=cluster.ids[i]
b=strsplit(a,split=",")
chromosom=b[[1]][1]
table[i,1]=chromosom
clusternr=b[[1]][2]
table[i,2]=clusternr
c=which(mouse.clusters[,3]==chromosom&mouse.clusters[,6]==clusternr)
clustersize=mouse.clusters[c[1],5]
table[i,3]=clustersize
affyIDs=as.character(mouse.clusters[c,1])
wt.up=intersect(affyIDs.wt.up,affyIDs)
wt.down=intersect(affyIDs.wt.down,affyIDs)
ko.up=intersect(affyIDs.ko.up,affyIDs)
ko.down=intersect(affyIDs.ko.down,affyIDs)
table[i,4]=length(wt.up)
table[i,5]=length(wt.down)
table[i,6]=length(ko.up)
table[i,7]=length(ko.down)
}

setwd("/home/dinkelac/data/sheena/tables/")

#4. print clusters mit illumina genes
#------------------------------------
#functions
#---------
#1. get.affy.ID
#--------------
#with an illuminaID
get.affy.ID=function(illuminaIDs,wtko){
if(wtko=="wt"){
table=read.csv(file="/home/dinkelac/data/sheena/tables/affy_illumina_mouse_wt.csv")
}
if(wtko=="ko"){
```

```
table=read.csv(file="/home/dinkelac/data/sheena/tables/affy_illumina_mouse_ko.csv")
}
affyIDs=""
for(i in 1:length(illuminaIDs)){
rows=grep(illuminaIDs[i],as.character(table[,5]))
if(length(rows)>0){
affyID=as.character(table[rows,2])
affyIDs=c(affyIDs,affyID)
}
}#end for [i]
return(affyIDs)
}#end function
```

1.1.15 Gene numbers of different versions of Annotations

This script is a documentation of gene and transcript numbers of the different Versions of Annotations.

```
#annotations.R
#-------------
#ensembl, biomart

#1. mouse
#--------
#ensembl 58
#----------
#84869 transcripts
#35958 genes

#brainarray version 13
#----------------------ss
#mouse 4302
#----------
#29590 transcripts
#16864 genes

#ensembl 59
#----------
#88186 transcripte
#36536 genes

#brainarray version 13
#---------------------
#gngnf1
#------
#34589 transcripts
#17121 genes

#2. rat
#------
#ensembl 57
#----------
#38144 transcripts
#28111 genes

#brainarray
#----------
#rgu34A
#------
```

```
#5793 transcripts
#4017 genes

#ensembl 60
#----------
#36536 mouse genes -> 20850 rat genes


#3. human
#--------
#ensembl 59
#----------
#151250 transcripts
#51737 genes

#brainarray version 13
#---------------------
#hgu133A
#-------
#49023 transcripts
#13663 genes

#ensembl 61
#----------
#28249 Illumina IDs
#24353 Ensemble Transcript IDs
#18678 Ensemble Gene IDs
#Chrom
#Startsite
#Entrez
#Refseq DNA
#HGNC symbol
```

1.1.16 Chromosomal map of the distribution of TRAs on the chromosomes

Map of TRAs on the chromosomes.

```
#chromosomenmap.R
#----------------

#library
library(geneplotter)
library(gnf1musa,lib.loc="/home/dinkelac/R-libs/")

setwd("/home/dinkelac/data/mouse/redundanzen/")
load("redundanzen.rda")

setwd("/home/dinkelac/data/mouse/nachbarn/batch6_10x_1000/")
load("mydata.rda")

chrObj=buildChromLocation("gnf1musa")
cPlot(chrObj,main="Mus musculus \n Gencluster(tra.index5.10x)")
index=tra.index5.5x
probes=names(genesymbol.ids[index])
cColor(probes,"red",chrObj)

setwd("/home/dinkelac/data/mouse/tra_5x_median/")
dev.copy2eps(file="chrom.map.tra.index5.10x.eps")
```

```
#genecluster drucken
setwd("/home/dinkelac/data/mouse/perl_cluster_check/observed_ntuples_tra.index5.5x/")
x=read.csv("table.tra.index5.5x.cluster.csv")
...
probes=as.character(x[,1])
...
dev.copy2eps(file="chrom.map.clustered.tra.index5.5x.eps")
```

1.1.17 Homology in clusters

This script calculates the homology in TRA clusters.

```
#homology_in_clusters.R
#-----------------------

#0. get skripts from Benedikt
#-----------------------------
/bbrors/data/sheena/homology_in_clusters

compare_seqs.pl

#1. compose directories for homology
#-----------------------------------
#format_cluster_output.pl

perl format_cluster_output.pl results.txt >clusters.tra_ensembl_mouse.txt

#1.1 download fasta files from cdna of all genes from ensembl
#-------------------------------------------------------------
#www.ensembl.org -> downloads-> FTP -> fasta cdna -> all

#mouse
#ftp://ftp.ensembl.org/pub/current/fasta/mus_musculus/cdna/
#Mus_musculus.NCBIM37.60.cdna.all.fa.gz

#human
#ftp://ftp.ensembl.org/pub/current/fasta/homo_sapiens/cdna/
#Homo_sapiens.GRCh37.60.cdna.all.fa.gz

#rat
#ftp://ftp.ensembl.org/pub/current/fasta/rattus_norvegicus/cdna/
#Rattus_norvegicus.RGSC3.4.60.cdna.all.fa.gz

compose_dirs_for_homology.pl
#needs bioperl
#change directories, and Identifier

#2. start_batch.pl
#-----------------
start_batch.pl
#start on tbi-pbs
#uses needle (needlemann wunsch) with a path from Benedikt
#gets the %identity and composes for each directory a file

#-> compare_seqs.out

#3. calculate_homology_in_clusters_gngnf1.R (1. part)
#----------------------------------------------------
#obere dreiecksmatrix, flatten matrix
```

```
#-> homology_gngnf1.eps

#boxplot pro spezies

#4. random clusters
#------------------
#rand 5,10,15,...,200 x 10
grep -c ">"Mus_musculus.NCBIM37.60.cdna.all.fa
#82508 mouse
#147141 human
#34721 rat

#in R
numbers=as.integer(runif(5350,min=1,max=34721))

setwd("/home/dinkelac/data/mouse/homology_in_clusters/rat/")
write.table(numbers,file="random_indices_seqs.txt",row.names=F,col.names=F)

#---------

perl create_random_clusters.pl

#5. start_batch_random.pl
#------------------------
#auf tbi-pbs starten

start_batch_random.pl

#walltime von 4 auf 10 std gesetzt
#calculate_homology_in_clusters_gngnf1.R (2. part)
```

1.1.18 Homology

This scripts calculates the homology of TRAs.

```
#homology.R
#----------
#The Table homologene.csv, extract from homologene.data matches NP_, XP_ of human, mouse,
#rat
#Now we can do any comparisons

#1. readin homologene.csv
#------------------------
homologene=read.csv(file="/home/dinkelac/data/sheena/homology/homologene.csv",sep="\t",
header=TRUE)

#a=as.character(homologene[,9])#mouse
a=as.character(homologene[,5])#human
b=strsplit(a,"\\.")
c=a
for(i in 1:length(b)){
c[i]=b[[i]][1]
}
#mouse.ref=c
human.ref=c
hom.ref=cbind(mouse.ref,human.ref)
#NP
#2. Compare TRA mouse, TRA human
#-------------------------------
#2.1 readin TRA mouse (tra.index7.5x)
```

```
#-------------------------------------
#5852 TRA.index7.5x
#4072 TRA.index7.5x.nr

tra.mouse=read.csv(file="/home/dinkelac/data/mouse/tables/tra.index7.5x.nr.annot.csv",
dec="/", sep="\t",header=TRUE)
tra.mouse=tra.mouse[,1:5]
mouse.refseq=as.character(tra.mouse[,3])
names(mouse.refseq)=tra.mouse[,1]

#2.2 readin TRA human (tra.index1.5x)
#-------------------------------------
#4180 tra.index1.5x
#3294 tra.index1.5x.nr

setwd("/home/wasserma/human/tables/")
table=read.table(file="human.tra.index1.5x.nr.txt")
tra.human=as.character(table[,1])

library(hgu133a, lib.loc="/home/dinkelac/R-libs")
source("/home/dinkelac/R-functions/pasteList.R")

refseq=mget(ls(env=hgu133aREFSEQ), env=hgu133aREFSEQ)
RefseqID=pasteList(refseq)
names(RefseqID)=names(refseq)
human.refseq=refseq[tra.human]
Symbol=mget(ls(env=hgu133aSYMBOL), env=hgu133aSYMBOL)

table=matrix(nrow=length(mouse.refseq),ncol=6)
colnames(table)=c("mouseID","mouseSymbol","mouseRefseq","humanRefseq","humanSymbol",
"humanID")
#mouseID (tra.index mouse)
table[,1]=names(mouse.refseq[])
table[,2]=as.character(tra.mouse[,2])

a=mouse.refseq[table[,1]]
b=strsplit(a,"/")
c=a
for(i in 1:length(b)){
c[i]=b[[i]][2]
}
#mouse refseq NP (tra list)
table[,3]=c[table[,1]]
#humanRefseq NP (homology list)
table[,4]=hom.ref[match(table[,3],hom.ref[,1]),2]

#human ID (tra list human)
get.human.id=function(a){
if(!is.na(a)){
b=names(which(lapply(lapply(human.refseq,match, a),sum,na.rm=T)>0))
if (length(b)==0){
return(NA)
}else{
return(b)
}
}else{
return(NA)
}
}

table[,6]=unlist(lapply(table[,4],get.human.id))
```

```
#humanSymbol

get.human.Symbol=function(a){
if(!is.na(a)){
#gib Symbol zurueck
Symbol[a]
}else{
return(NA)
}
}

table[,5]=unlist(lapply(table[,6],get.human.Symbol))

write.table(table,file="tra.mouse.human.homologs.csv")
```

1.1.19 Merge Tables

This script merges two tables for the display of TRA clusters.

```
#!/usr/bin/perl
#merge.table.pl
#---------------
#um TRA cluster darzustellen brauchen wir eine tabelle, fuer alle gnf gene
#dazu nehmen wir aus
#chrhash (old): AffyID, Chr, Startside
#gnf1m.annotiert: AffyID, Symbol,
#zusammenfuegen zu: gnf1m.annotiert.2006: AffyID, Symbol, Chr, Startside

%symbol;
%chromosome;
%startside;
open(FILEHANDLE,"./old.tables/chrhash_2006.txt");
while($line=<FILEHANDLE>){
chop $line;
@array=split("\t",$line);
$affyID=shift @array;
$chr=shift @array;
$start=shift @array;
#print $affyID."\n";
$chromosome{$affyID}=$chr;
$startside{$affyID}=$start;
}
close(FILEHANDLE);

open(FILEHANDLE,"gnf1m.annotiert.2007.csv");
while($line=<FILEHANDLE>){
chop $line;
@array=split("\t",$line);
$affyID=shift @array;#mit ueberschrift
$sym=shift @array;
#print $affyID." ".$sym."\n";
$symbol{$affyID}=$sym;
}
close(FILEHANDLE);

open(FILEHANDLE,">gnf1m.annotiert.2006.csv");
foreach $key(sort keys %chromosome){
print FILEHANDLE "$key"."\t"."$symbol{$key}"."\t"."$chromosome{$key}"."\t".
"$startside{$key}"."\n";
#print "$key"."\t"."$symbol{$key}"."\n";
```

```
}

close(FILEHANDLE);
__END__
```

1.1.20 Syntheny Maps

```
#syntheny_maps.R
#---------------
#m.dinkelacker@dkfz.de
#date: 11.2.2011

#task: compare tra clusters from mouse and human and rat, are they conserved?

#1. read in
#----------
#1.1 cluster tables human
#------------------------
setwd("/home/dinkelac/data/mouse/tables/")

human.cluster.table=read.csv(file="table.human.tra.index2.5x.cluster.csv",sep="\t")
mouse.cluster.table=read.csv(file="table.gngnf1.tra.index10.5x.cluster.csv",sep="\t")
rat.cluster.table=read.csv(file="table.rat.homologs.tra.index10.5x.cluster.csv",sep="\t")

#1.3 homology table
#------------------
#homology between ensembl gene human, mouse, rat

homology.table=read.csv(file="/home/dinkelac/data/mouse/tables/ensembl.60.homology.mouse.rat.huma

#2 calculate matching clusters
#-----------------------------
#function drum herum
a=get.cluster(1,2,"mouse")

ensembl.ids.mouse=as.character(a[,1])
ensembl.ids.human=unique(get.homologs(ensembl.ids.mouse,"human"))
ensembl.ids.rat=unique(get.homologs(ensembl.ids.mouse,"rat"))

mouse.hits=length(ensembl.ids.mouse)
human.hits=sum(!is.na(ensembl.ids.human)&(ensembl.ids.human!=""))
rat.hits=sum(!is.na(ensembl.ids.rat)&(ensembl.ids.rat!=""))

matching.cluster.human=get.cluster.id(ensembl.ids.human,"human")
matching.cluster.rat=get.cluster.id(ensembl.ids.rat,"rat")

#cluster matching table
#----------------------
chrom=as.character(mouse.cluster.table[,3])
cluster.nr=mouse.cluster.table[,8]

#mouse.cluster
mouse.cluster.ids=unique(paste("mouse ",chrom,":",cluster.nr,sep=""))

table=list()

for(i in 1:length(mouse.cluster.ids)){
#for(i in 1:2){
print(i)
```

```
mouse.cluster.id=mouse.cluster.ids[i]
species=strsplit(mouse.cluster.id,split=" ")[[1]][1]
cluster=strsplit(mouse.cluster.id,split=" ")[[1]][2]
chrom=strsplit(cluster,split=":")[[1]][1]
cluster.no=strsplit(cluster,split=":")[[1]][2]

a=get.cluster(chrom,cluster.no,species)
ensembl.ids.mouse=as.character(a[,1])
mouse.genes=(length(ensembl.ids.mouse))

ensembl.ids.human=get.homologs(ensembl.ids.mouse,"human")
human.homologs=sum(!is.na(ensembl.ids.human)&(ensembl.ids.human!=""))

ensembl.ids.rat=get.homologs(ensembl.ids.mouse,"rat")
rat.homologs=sum(!is.na(ensembl.ids.rat)&(ensembl.ids.rat!=""))

matching.cluster.human=get.cluster.id(ensembl.ids.human,"human")
matching.cluster.rat=get.cluster.id(ensembl.ids.rat,"rat")

human.cluster=unique(matching.cluster.human)
rat.cluster=unique(matching.cluster.rat)

for(k in 1:max(length(human.cluster),length(rat.cluster))){
h.hits=NA
h.size=NA

if(!is.na(human.cluster[k])){
h.hits=sum(matching.cluster.human==human.cluster[k])
human.cluster.id=human.cluster[k]
species=strsplit(human.cluster.id,split=" ")[[1]][1]
cluster=strsplit(human.cluster.id,split=" ")[[1]][2]
chrom=strsplit(cluster,split=":")[[1]][1]
cluster.no=strsplit(cluster,split=":")[[1]][2]
b=get.cluster(chrom,cluster.no,species)
h.size=b[1,7]
}

r.hits=NA
r.size=NA

if(!is.na(rat.cluster[k])){
r.hits=sum(matching.cluster.rat==rat.cluster[k])
rat.cluster.id=rat.cluster[k]
species=strsplit(rat.cluster.id,split=" ")[[1]][1]
cluster=strsplit(rat.cluster.id,split=" ")[[1]][2]
chrom=strsplit(cluster,split=":")[[1]][1]
cluster.no=strsplit(cluster,split=":")[[1]][2]
b=get.cluster(chrom,cluster.no,species)
r.size=b[1,7]
}

zeile=cbind(mouse.cluster.id,mouse.genes,human.homologs,rat.homologs,human.cluster[k],h.hits,h.si
table=rbind(table,zeile)

}#end for k
}#end for i

colnames(table)=c("mouse.cluster.id","mouse.genes","human.homologs","rat.homologs","human.cluster

setwd("/home/dinkelac/data/mouse/tables")
write.table(table,file="syntheny.table_new.csv",sep="\t",row.names=F)
```

```
#functions
#---------
#get mouse cluster
#-----------------
get.cluster=function(chromosome,clusternr,species){
chromosome=as.character(chromosome)
if(species=="mouse"){
x=mouse.cluster.table[which(mouse.cluster.table[,3]==chromosome),]
cluster=x[which(x[,8]==as.character(clusternr)),]
}
if(species=="rat"){
x=rat.cluster.table[which(rat.cluster.table[,3]==chromosome),]
cluster=x[which(x[,8]==as.character(clusternr)),]
}
if(species=="human"){
x=human.cluster.table[which(human.cluster.table[,3]==chromosome),]
cluster=x[which(x[,8]==as.character(clusternr)),]
}
return(cluster)
}

#get human and rat homologs
#--------------------------
ensembl.ids.human=get.homologs(ensembl.ids.mouse,"human")


get.homologs=function(ensembl.ids.mouse,species){
if(species=="human"){
human.homologs=vector()
for(i in 1:length(ensembl.ids.mouse)){
human.homolog=unique(as.character(homology.table[which(homology.table[,1]==ensembl.ids.mouse[i]),
human.homologs=c(human.homologs,human.homolog)
}
return(human.homologs)
}
if(species=="rat"){
rat.homologs=vector()
for(i in 1:length(ensembl.ids.mouse)){
rat.homolog=unique(as.character(homology.table[which(homology.table[,1]==ensembl.ids.mouse[i]),3]
rat.homologs=c(rat.homologs,rat.homolog)
}
return(rat.homologs)
}
}

#get cluster id
#--------------
#usage:
#matching.cluster.human=get.cluster.id(ensembl.ids.human,"human")
#33
#------
get.cluster.id=function(ensembl.ids,species){
#human
if(species=="human"){
line=vector()
for(i in 1:length(ensembl.ids)){
line=c(line,which(human.cluster.table[,1]==ensembl.ids[i]))
}
if(length(line)==0){
return(NA)
}else{
chromosome=as.character(human.cluster.table[line,3])
```

```
clusternr=human.cluster.table[line,8]
return(paste(species," ",chromosome,":",clusternr,sep=""))
}
}
#rat
if(species=="rat"){
line=vector()
for(i in 1:length(ensembl.ids)){
line=c(line,which(rat.cluster.table[,1]==ensembl.ids[i]))
}
if(length(line)==0){
return(NA)
}else{
chromosome=as.character(rat.cluster.table[line,3])
clusternr=rat.cluster.table[line,8]
return(paste(species," ",chromosome,":",clusternr,sep=""))
}
}
}
```

1.1.21 Print Syntheny Maps

This script plots out syntheny maps of TRA clusters.

```
#print.syntheny.maps.R
#----------------------
#tables
#------
setwd("/home/dinkelac/data/mouse/tables/")

human.cluster.table=read.csv(file="table.human.tra.index2.5x.cluster.csv",sep="\t")
mouse.cluster.table=read.csv(file="table.gngnf1.tra.index10.5x.cluster.csv",sep="\t")
rat.cluster.table=read.csv(file="table.rat.homologs.tra.index10.5x.cluster.csv",sep="\t")

human.chrhash.table=read.table("./ensembl_genes/human/ensembl.59.human.txt",sep="\t",
header=T)
chrom=as.character(human.chrhash.table[,3])
startside=human.chrhash.table[,4]
human.chrhash=cbind(chrom,startside)
rownames(human.chrhash)=as.character(human.chrhash.table[,1])
human.chrhash=human.chrhash[!duplicated(rownames(human.chrhash)),]

mouse.chrhash.table=read.table("./ensembl_genes/mouse/ensembl.59.mouse.chrhash.genes.txt",
sep="\t")
chrom=as.character(mouse.chrhash.table[,2])
startside=mouse.chrhash.table[,3]
mouse.chrhash=cbind(chrom,startside)
rownames(mouse.chrhash)=as.character(mouse.chrhash.table[,1])
mouse.chrhash=mouse.chrhash[!duplicated(rownames(mouse.chrhash)),]

rat.chrhash.table=read.table("./ensembl.60.rat.txt",sep="\t",header=T)
chrom=as.character(rat.chrhash.table[,3])
startside=rat.chrhash.table[,4]
rat.chrhash=cbind(chrom,startside)
rownames(rat.chrhash)=as.character(rat.chrhash.table[,1])
rat.chrhash=rat.chrhash[!duplicated(rownames(rat.chrhash)),]

homology.table=read.csv(file="/home/dinkelac/data/mouse/tables/
ensembl.60.homology.mouse.rat.human.txt",sep="\t")

human.ensembl.id=as.character(homology.table[,5])
names(human.ensembl.id)=homology.table[,1]
```

```
rat.ensembl.id=as.character(homology.table[,3])
names(rat.ensembl.id)=homology.table[,1]
#----------------------

setwd("/home/dinkelac/data/mouse/sessions/rda/")
#save.image("syntheny_maps.rda")

load("syntheny_maps.rda")

#cluster.matching.table

#plot.cluster("mouse 1:3","human 2:16","rat 9:3")
mouse="mouse 1:2"
human="human 10:1"
rat="rat"
plot.cluster("mouse 1:2","human 10:1","rat 9:2")

#function
#--------
#plot.cluster
plot.cluster=function(mouse,human,rat){

species1=strsplit(mouse,split=" ")[[1]][1]
cluster.id.mouse=strsplit(mouse,split=" ")[[1]][2]
chrom.mouse=strsplit(cluster.id.mouse,split=":")[[1]][1]
no.mouse=strsplit(cluster.id.mouse,split=":")[[1]][2]

species2=strsplit(human,split=" ")[[1]][1]
cluster.id.human=strsplit(human,split=" ")[[1]][2]
chrom.human=strsplit(cluster.id.human,split=":")[[1]][1]
no.human=strsplit(cluster.id.human,split=":")[[1]][2]

species3=strsplit(rat,split=" ")[[1]][1]
cluster.id.rat=strsplit(rat,split=" ")[[1]][2]
chrom.rat=strsplit(cluster.id.rat,split=":")[[1]][1]
no.rat=strsplit(cluster.id.rat,split=":")[[1]][2]

mouse.cluster=get.cluster(chrom.mouse,no.mouse,species1)
human.cluster=get.cluster(chrom.human,no.human,species2)
rat.cluster=get.cluster(chrom.rat,no.rat,species3)

min.mouse=min(mouse.cluster[,4])
min.human=min(human.cluster[,4])
min.rat=min(rat.cluster[,4])
min.global=min(min.mouse,min.human,min.rat)

max.mouse=max(mouse.cluster[,4])
max.human=max(human.cluster[,4])
max.rat=max(rat.cluster[,4])
max.global=max(max.mouse,max.human,max.rat)

mouse.startsides=as.numeric(get.startsides(chrom.mouse,min.mouse,max.mouse,
species1)[,2])
names(mouse.startsides)=names(get.startsides(chrom.mouse,min.mouse,max.mouse,
species1)[,2])

human.startsides=as.numeric(get.startsides(chrom.human,min.human,max.human,
species2)[,2])
names(human.startsides)=names(get.startsides(chrom.human,min.human,max.human,
species2)[,2])
```

```
rat.startsides=as.numeric(get.startsides(chrom.rat,min.rat,max.rat,species3)[,2])
names(rat.startsides)=names(get.startsides(chrom.rat,min.rat,max.rat,species3)[,2])

basisplot(species2,min.human,max.human,10,species1,min.mouse,max.mouse,0,species3,
min.rat,max.rat,-10)

mtext(paste("chromosome:",chrom.human,sep=""),2,par(las=1),at=8,cex=0.8)
mtext(paste("clusternr.:",no.human,sep=""),2,par(las=1),at=6,cex=0.8)
mtext(paste("chromosome:",chrom.mouse,sep=""),2,par(las=1),at=-2,cex=0.8)
mtext(paste("clusternr.:",no.mouse,sep=""),2,par(las=1),at=-4,cex=0.8)
mtext(paste("chromosome:",chrom.rat,sep=""),2,par(las=1),at=-12,cex=0.8)
mtext(paste("clusternr.:",no.rat,sep=""),2,par(las=1),at=-14,cex=0.8)
title(main="Clustering of tissue restricted antigens (TRAs)\n in mouse human and rat")

human.parameters=get.parameters(human.startsides,min.global,max.global)
mouse.parameters=get.parameters(mouse.startsides,min.global,max.global)
rat.parameters=get.parameters(rat.startsides,min.global,max.global)

rescaled.human.startsides=rescale(human.startsides,human.parameters)
rescaled.mouse.startsides=rescale(mouse.startsides,mouse.parameters)
rescaled.rat.startsides=rescale(rat.startsides,rat.parameters)

rescaled.human.tras=rescale(human.cluster[,4],human.parameters)
names(rescaled.human.tras)=human.cluster[,1]

rescaled.mouse.tras=rescale(mouse.cluster[,4],mouse.parameters)
names(rescaled.mouse.tras)=mouse.cluster[,1]

rescaled.rat.tras=rescale(rat.cluster[,4],rat.parameters)
names(rescaled.rat.tras)=rat.cluster[,1]

plot.genes(rescaled.human.startsides,"black",11,9)
plot.genes(rescaled.mouse.startsides,"black",1,-1)
plot.genes(rescaled.rat.startsides,"black",-11,-9)

plot.genes(rescaled.human.tras,"red",11,9)
plot.genes(rescaled.mouse.tras,"red",1,-1)
plot.genes(rescaled.rat.tras,"red",-11,-9)

human.symbols=as.character(human.cluster[,2])
names(human.symbols)=human.cluster[,1]

mouse.symbols=as.character(mouse.cluster[,2])
names(mouse.symbols)=mouse.cluster[,1]

rat.symbols=as.character(rat.cluster[,2])
names(rat.symbols)=rat.cluster[,1]

#plot.symbols(rescaled.human.tras,human.symbols,"red",12,0,800000)
#plot.symbols(rescaled.mouse.tras,mouse.symbols,"red",-2,1,1500000)
#plot.symbols(rescaled.rat.tras,rat.symbols,"red",-12,1,0)

no.homologs.mouse.human=connect.homologs(rescaled.mouse.startsides,
rescaled.human.startsides,"human",9,1,"black")
no.homologs.mouse.rat=connect.homologs(rescaled.mouse.startsides,
rescaled.rat.startsides,"rat",-9,-1,"black")

no.homologs.mouse.human.tras=connect.homologs(rescaled.mouse.tras,
rescaled.human.tras,"human",9,1,"red")
no.homologs.mouse.rat.tras=connect.homologs(rescaled.mouse.tras,
rescaled.rat.tras,"rat",-9,-1,"red")
```

```
#some numbers
no.genes.human=length(human.startsides)
no.genes.mouse=length(mouse.startsides)
no.genes.rat=length(rat.startsides)

no.tras.human=length(rescaled.human.tras)
no.tras.mouse=length(rescaled.mouse.tras)
no.tras.rat=length(rescaled.rat.tras)

tra.density.human=round(no.tras.human/no.genes.human*100)
tra.density.mouse=round(no.tras.mouse/no.genes.mouse*100)
tra.density.rat=round(no.tras.rat/no.genes.rat*100)

human.percent=paste(tra.density.human,"%",sep="")
mouse.percent=paste(tra.density.mouse,"%",sep="")
rat.percent=paste(tra.density.rat,"%",sep="")

human.range=round((max.human-min.human)/1000)
mouse.range=round((max.mouse-min.mouse)/1000)
rat.range=round((max.rat-min.rat)/1000)

human.tras=paste(no.tras.human,"TRAs ")
human.tras1=paste("(=",human.percent,")")

if(no.tras.human!=0){
mtext(paste(no.tras.human,"TRAs"),4,par(las=1),at=10,cex=0.8,col="red")
mtext(paste(no.genes.human,"genes"),4,par(las=1),at=8.5,cex=0.8)
mtext(paste("TRA density:",human.percent),4,par(las=1),at=7,cex=0.8)
mtext(paste("size:",human.range,"kbp"),4,par(las=1),at=5.5,cex=0.8)
mtext(paste(no.homologs.mouse.human,"mouse homologs"),4,par(las=1),at=4,cex=0.8)
}

mtext(paste(no.tras.mouse,"TRAs"),4,par(las=1),at=0,cex=0.8,col="red")
mtext(paste(no.genes.mouse,"genes"),4,par(las=1),at=-1.5,cex=0.8)
mtext(paste("TRA density:",mouse.percent),4,par(las=1),at=-3,cex=0.8)
mtext(paste("size:",mouse.range,"kbp"),4,par(las=1),at=-4.5,cex=0.8)

if(no.tras.rat!=0){
mtext(paste(no.tras.rat,"TRAs"),4,par(las=1),at=-10,cex=0.8,col="red")
mtext(paste(no.genes.rat,"genes"),4,par(las=1),at=-11.5,cex=0.8)
mtext(paste("TRA density:",rat.percent),4,par(las=1),at=-13,cex=0.8)
mtext(paste("size:",rat.range,"kbp"),4,par(las=1),at=-14.5,cex=0.8)
mtext(paste(no.homologs.mouse.rat,"mouse homologs"),4,par(las=1),at=-16,cex=0.8)
}
}#end of function plot cluster

#------------------------
#verbindet die homologen TRAs in mouse und human cluster
connect.homologs=function(mouse.genes,species.genes,species,pos1,pos2,color){
if(species=="human"){
human.ensembl_ids=names(species.genes)
mouse.ensembl_ids=names(mouse.genes)
no=0
#berechne anzahl an gefundenen
for(i in 1:length(mouse.ensembl_ids)){
mouse.ensembl_ID=mouse.ensembl_ids[i]
human.ensembl_ID=get.homolog(mouse.ensembl_ID,"human")
for(j in 1:length(human.ensembl_ID)){
if(!is.na(human.ensembl_ID[j])&is.element(human.ensembl_ID[j],human.ensembl_ids)){
#draw line
human.pos=species.genes[human.ensembl_ID[j]]
mouse.pos=mouse.genes[mouse.ensembl_ID]
```

```
lines(c(human.pos,mouse.pos),c(pos1,pos2),type="l",col=color)
no=no+1
}
}
}
}
if(species=="rat"){
rat.ensembl_ids=names(species.genes)
mouse.ensembl_ids=names(mouse.genes)
no=0
#berechne anzahl an gefundenen
for(i in 1:length(mouse.ensembl_ids)){
mouse.ensembl_ID=mouse.ensembl_ids[i]
rat.ensembl_ID=get.homolog(mouse.ensembl_ID,"rat")
for(j in 1:length(rat.ensembl_ID)){
if(!is.na(rat.ensembl_ID[j])&is.element(rat.ensembl_ID[j],rat.ensembl_ids)){
#draw line
rat.pos=species.genes[rat.ensembl_ID[j]]
mouse.pos=mouse.genes[mouse.ensembl_ID]
lines(c(rat.pos,mouse.pos),c(pos1,pos2),type="l",col=color)
no=no+1
}
}
}
}
return(no)
}

#get.cluster
#-----------
#mouse.cluster=get.cluster(chrom.mouse,no.mouse,species1)

get.cluster=function(chromosome,clusternr,species){
chromosome=as.character(chromosome)
if(species=="mouse"){
x=mouse.cluster.table[which(mouse.cluster.table[,3]==chromosome),]
cluster=x[which(x[,8]==as.character(clusternr)),]
}
if(species=="rat"){
x=rat.cluster.table[which(rat.cluster.table[,3]==chromosome),]
cluster=x[which(x[,8]==as.character(clusternr)),]
}
if(species=="human"){
x=human.cluster.table[which(human.cluster.table[,3]==chromosome),]
cluster=x[which(x[,8]==as.character(clusternr)),]
}
return(cluster)
}

basisplot=function(name1,min1,max1,pos1,name2,min2,max2,pos2,name3,min3,max3,pos3){
min=min(min1,min2,min3)
max=max(max1,max2,max3)
pos=c(pos1,pos2,pos3)
y=c(10,11)
mmi=c(1.5,1.5,1,1.5)
par(mai=mmi)
plot(y,xlim=c(min,max),ylim=c(-15,15),axes=F,xlab="start site (bp)",ylab="")
abline(h=pos)
mtext(name1,2,par(las=1),font=2,at=pos1,cex=0.8)
mtext(name2,2,par(las=1),font=2,at=pos2,cex=0.8)
mtext(name3,2,par(las=1),font=2,at=pos3,cex=0.8)
}
```

```
get.startsides=function(chrom,min,max,species){
if(species=="mouse"){
chrhash=mouse.chrhash
}
if(species=="human"){
chrhash=human.chrhash
}
if(species=="rat"){
chrhash=rat.chrhash
}
startsides=chrhash[chrhash[,1]==chrom&as.numeric(chrhash[,2])>=
min&as.numeric(chrhash[,2])<=max,]
return(startsides)
}

#plots vertical lines in positions pos
plot.genes=function(x,farbe,min,max){
#description=symbols
pos=c(min,max)
y=x
x=c(x,x)
x=sort(x)
i=1
while(i <=length(x)){
lines(c(x[i],x[i+1]),pos,type="l",col=farbe)
#startside
#text(x,-2,x,srt=90,col=farbe,cex=0.8,adj=1)
#text(y,2,description,srt=90,col=farbe,cex=0.8,adj=0)
i=i+2
}
}

mouse.parameters=get.parameters(mouse.startsides,min.global,max.global)

#estimates parameters for rescaling tras to the whole range of the plot
get.parameters=function(vector,global.min,global.max){
parameters=c()
#schleife
for(i in 1:10){
local.min=min(vector)
diff=abs(global.min-local.min)
vector=vector-diff
local.max=max(vector)
factor=global.max/local.max#alles nach re
vector=vector*factor
parameters=c(parameters,diff,factor)
}
local.min=min(vector)
diff=abs(global.min-local.min)
parameters=c(parameters,diff)
return (parameters)
}

rescale=function(vector,parameters){
i=1
while(i<=20){
vector=vector-parameters[i]
vector=vector*parameters[i+1]
i=i+2
}
vector=vector-parameters[21]
```

```
return(vector)
}

#pos = y-achse,dir 0 nach oben, 1 nach unten
plot.symbols=function(vector,symbol.vector,color,pos,dir,skip){
ensembl_ids=names(vector)
if(skip==0){
symbols=symbol.vector[ensembl_ids]
text(vector,pos,symbols,srt=90,col=color,cex=0.65,adj=dir)
}
else{
symbols=symbol.vector[ensembl_ids]
skip_index=vector()
for(i in 1:(length(vector)-1)){
if(abs(vector[i+1]-vector[i])<skip){
skip_index=c(skip_index,i+1)
}
}
symbols[skip_index]=""
text(vector,pos,symbols,srt=90,col=color,cex=0.65,adj=dir)
}
}

get.homolog=function(ensembl_mouse_id,species){
if(species=="human"){
homolog=as.character(unique(homology.table[which(homology.table[,1]==
ensembl_mouse_id),5]))
}
if(species=="rat"){
homolog=as.character(unique(homology.table[which(homology.table[,1]==
ensembl_mouse_id),3]))
}
return(homolog)
}

#old script
setwd("/home/dinkelac/data/mouse/sessions/rda/")
#save.image("syntheny_maps.rda")

load("syntheny_maps.rda")

#readin cluster matching table
#------------------------------
table=read.csv(file="/home/dinkelac/data/mouse/tables/syntheny.table_new.csv",
header=TRUE,sep="\t")

#for each line
mouse.clusters=unique(as.character(table[,1]))

#plot syntheny clusters best hits only
#--------------------------------------
setwd("/home/dinkelac/data/mouse/plots/syntheny_plots1/")

for(i in 1:length(mouse.clusters)){
#for(i in 1:1){
lines=table[as.character(table[,1])==mouse.clusters[i],]
if(is.na(lines[,6])){
human.hits=NA
}else{
human.hits=as.character(lines[,5])[lines[,6]==max(lines[,6],na.rm=T)]
human.hits=human.hits[which(human.hits!="NA")]
}
```

```
if(is.na(human.hits)){
human.hits="human"
}
if(is.na(lines[,9])){
rat.hits=NA
}else{
rat.hits=as.character(lines[,8])[lines[,9]==max(lines[,9],na.rm=T)]
rat.hits=rat.hits[which(rat.hits!="NA")]
}
if(is.na(rat.hits)){
rat.hits="rat"
}
print(i)
plot.cluster(mouse.clusters[i],human.hits,rat.hits)
name=paste(mouse.clusters[i],"_",human.hits,"_",rat.hits,".eps",sep="")
dev.copy2eps(file=name)
}

#fuer jedes cluster
nr=as.numeric(rownames(table))

for(i in 1:max(nr)){
if(!is.na(table[i,7])){
mouse.chrom=as.character(table[i,1])
mouse.clusternr=as.character(table[i,2])
best.hit=unlist(strsplit(as.character(table[i,6]),split=","))

for(j in 1:length(best.hit)){
hit=unlist(strsplit(best.hit[j],split=":"))
human.chrom=hit[1]
human.clusternr=hit[2]
compare.clusters(mouse.chrom,mouse.clusternr,human.chrom,human.clusternr,200000,200000)
name=paste("human",human.chrom,".",human.clusternr,"_mouse",mouse.chrom,".",
mouse.clusternr,".eps",sep="")
dev.copy2eps(file=name)
}#end for(j)
}#end if
}#end for(i)

#functions
#---------
#1. get.first
#-----------
#gets the first entry in a list of lists

get.first=function(x){
y=vector("list",length(x))
for(i in 1:length(x)){
y[[i]]=x[[i]][1]
y=unlist(y)
}
return(y)
}
```

1.1.22 Print Syntheny Maps Tissues

This script plots out syntheny maps of TRA clusters with tissue specificity.

```
#print.syntheny.maps_tissues.R
```

```
#---------------------
#tables
#------
setwd("/home/dinkelac/data/mouse/tables/")

human.cluster.table=read.csv(file="table.human.tra.index2.5x.cluster.csv",sep="\t")
mouse.cluster.table=read.csv(file="table.gngnf1.tra.index10.5x.cluster.csv",sep="\t")

human.chrhash.table=read.table("./ensembl_genes/human/ensembl.59.human.txt",sep="\t",
header=T)
chrom=as.character(human.chrhash.table[,3])
startside=human.chrhash.table[,4]
human.chrhash=cbind(chrom,startside)
rownames(human.chrhash)=as.character(human.chrhash.table[,1])
human.chrhash=human.chrhash[!duplicated(rownames(human.chrhash)),]

mouse.chrhash.table=read.table("./ensembl_genes/mouse/ensembl.59.mouse.chrhash.genes.txt",
sep="\t")
chrom=as.character(mouse.chrhash.table[,2])
startside=mouse.chrhash.table[,3]
mouse.chrhash=cbind(chrom,startside)
rownames(mouse.chrhash)=as.character(mouse.chrhash.table[,1])
mouse.chrhash=mouse.chrhash[!duplicated(rownames(mouse.chrhash)),]

homology.table=read.csv(file="/home/dinkelac/data/mouse/tables/
ensembl.60.homology.mouse.rat.human.txt",sep="\t")

human.ensembl.id=as.character(homology.table[,5])
names(human.ensembl.id)=homology.table[,1]

setwd("/home/dinkelac/data/mouse/tables/")
a=read.table("tissues/tissue.colors.human.csv",sep="\t")
tissue.colors.human=as.character(a[,2])
names(tissue.colors.human)=as.character(a[,1])

b=read.table("tissues/tissue.colors.gngnf1.csv",sep="\t")
tissue.colors.mouse=as.character(b[,4])
names(tissue.colors.mouse)=as.character(b[,2])
#---------------------

setwd("/home/dinkelac/data/mouse/sessions/rda/")
#save.image("syntheny_maps_tissues.rda")

load("syntheny_maps_tissues.rda")

#cluster.matching.table

mouse="mouse 1:2"
human="human 10:1"

setwd("/home/dinkelac/data/mouse/plots/")
x11(h=7,w=10)

plot.cluster("mouse 1:2","human 10:1")

#function
#--------
#plot.cluster
plot.cluster=function(mouse,human){

species1=strsplit(mouse,split=" ")[[1]][1]
cluster.id.mouse=strsplit(mouse,split=" ")[[1]][2]
```

```
chrom.mouse=strsplit(cluster.id.mouse,split=":")[[1]][1]
no.mouse=strsplit(cluster.id.mouse,split=":")[[1]][2]

species2=strsplit(human,split=" ")[[1]][1]
cluster.id.human=strsplit(human,split=" ")[[1]][2]
chrom.human=strsplit(cluster.id.human,split=":")[[1]][1]
no.human=strsplit(cluster.id.human,split=":")[[1]][2]

mouse.cluster=get.cluster(chrom.mouse,no.mouse,species1)
human.cluster=get.cluster(chrom.human,no.human,species2)

min.mouse=min(mouse.cluster[,4])
min.human=min(human.cluster[,4])
min.global=min(min.mouse,min.human)

max.mouse=max(mouse.cluster[,4])
max.human=max(human.cluster[,4])
max.global=max(max.mouse,max.human)

mouse.startsides=as.numeric(get.startsides(chrom.mouse,min.mouse,max.mouse,species1)[,2])
names(mouse.startsides)=names(get.startsides(chrom.mouse,min.mouse,max.mouse,species1)[,2])

human.startsides=as.numeric(get.startsides(chrom.human,min.human,max.human,species2)[,2])
names(human.startsides)=names(get.startsides(chrom.human,min.human,max.human,species2)[,2])

basisplot(species2,min.human,max.human,10,species1,min.mouse,max.mouse,0)

mtext(paste("chromosome:",chrom.human,sep=""),2,par(las=1),at=8,cex=0.8)
mtext(paste("clusternr.:",no.human,sep=""),2,par(las=1),at=6,cex=0.8)
mtext(paste("chromosome:",chrom.mouse,sep=""),2,par(las=1),at=-2,cex=0.8)
mtext(paste("clusternr.:",no.mouse,sep=""),2,par(las=1),at=-4,cex=0.8)
title(main="Clustering of tissue restricted antigens (TRAs)\n in mouse human")

#tissues
max.tissues.human=as.character(human.cluster[,10])
max.tissues.mouse=as.character(mouse.cluster[,10])
unique.max.tissues.on.cluster=sort(unique(tolower(c(max.tissues.human,max.tissues.mouse))))

max.colors.human=tissue.colors.human[max.tissues.human]
names(max.colors.human)=tolower(names(max.colors.human))
max.colors.mouse=tissue.colors.mouse[max.tissues.mouse]
names(max.colors.mouse)=tolower(names(max.colors.mouse))

legend.colors=vector(len=length(unique.max.tissues.on.cluster))
names(legend.colors)=unique.max.tissues.on.cluster
for(i in 1:length(unique.max.tissues.on.cluster)){
legend.colors[i]=max.colors.human[names(legend.colors[i])]
if(is.na(legend.colors[i])){
legend.colors[i]=max.colors.mouse[names(legend.colors[i])]
}
}

human.width=(max.global-min.global)/length(max.tissues.human)
mouse.width=(max.global-min.global)/length(max.tissues.mouse)

x=min.global
z.human=min.global

#plot tissue colors
for(k in 1:length(max.tissues.human)){
rect(x,9,(x+human.width),11,col=max.colors.human[k],border=max.colors.human[k])
x=x+human.width
```

```
z.human=c(z.human,x)
}
rect(min.global,9,max.global,11,border="black")
z.human=z.human[1:length(z.human)-1]
zi.human=z.human+0.5*human.width

x=min.global
z.mouse=min.global

for(k in 1:length(max.tissues.mouse)){
rect(x,-1,(x+mouse.width),1,col=max.colors.mouse[k],border=max.colors.mouse[k])
x=x+mouse.width
z.mouse=c(z.mouse,x)
}
rect(min.global,-1,max.global,1,border="black")
z.mouse=z.mouse[1:length(z.mouse)-1]
zi.mouse=z.mouse+0.5*mouse.width

#plot startsides
clustersize=max(length(max.tissues.human),length(max.tissues.mouse))

writing.human=60
if(length(max.tissues.human)>59){
writing.human=90
}

writing.mouse=60
if(length(max.tissues.mouse)>59){
writing.mouse=90
}

text(zi.human,12,max.tissues.human,cex=0.7,srt=writing.human,adj=0)
text(zi.mouse,-2,max.tissues.mouse,cex=0.7,srt=writing.mouse,adj=1)

#linien
mouse.ids=as.character(mouse.cluster[,1])
human.ids=as.character(human.cluster[,1])

f=connect.homologs(mouse.ids,human.ids,min.global,human.width,mouse.width,
max.tissues.mouse,max.tissues.human,max.colors.mouse,9,1,"black")
no.homologs=f[1]
no.same.tissues=f[2]

#legend
legend("bottom",pch=22,names(legend.colors),pt.bg=legend.colors,col=legend.colors,
ncol=4,cex=0.7)

no.tras.human=length(human.ids)
no.tras.mouse=length(mouse.ids)
human.tras=paste(no.tras.human,"TRAs ")
mouse.tiss=length(unique(max.tissues.mouse))
human.tiss=length(unique(max.tissues.human))

mtext(paste(no.tras.human,"TRAs"),4,par(las=1),at=10,cex=0.8,col="red")
mtext(paste(human.tiss,"human tissues"),4,par(las=1),at=8.5,cex=0.8,col="black")

mtext(paste(no.tras.mouse,"TRAs"),4,par(las=1),at=0,cex=0.8,col="red")
mtext(paste(mouse.tiss,"mouse tissues"),4,par(las=1),at=-1.5,cex=0.8,col="black")
mtext(paste(no.homologs,"homolog TRAs"),4,par(las=1),at=-5,cex=0.8,col="red")
mtext(paste(no.same.tissues,"homolog tissues"),4,par(las=1),at=-6.5,cex=0.8,col="black")
}#end of function plot cluster
```

```
#------------------------
#verbindet die homologen TRAs in mouse und human cluster
connect.homologs=function(mouse.ensembl_ids,human.ensembl_ids,min.global,human.width,
mouse.width,max.tissues.mouse,max.tissues.human,max.colors.mouse,pos1,pos2,color){
color1=color
human.positions=vector()
human.colors=vector()
mouse.positions=vector()
mouse.colors=vector()
no.tissues=0
#berechne anzahl an gefundenen
for(i in 1:length(mouse.ensembl_ids)){
mouse.ensembl_ID=mouse.ensembl_ids[i]
human.ensembl_ID=get.homolog(mouse.ensembl_ID)

for(i in 1:length(human.ensembl_ID)){
if(!is.na(human.ensembl_ID[i])&is.element(human.ensembl_ID[i],human.ensembl_ids)){
#prepare line
human.pos=which(human.ensembl_ids==human.ensembl_ID[i])
pos.hum=min.global+0.5*human.width+((human.pos-1)*human.width)
human.positions=c(human.positions,pos.hum)

mouse.pos=which(mouse.ensembl_ids==mouse.ensembl_ID)
pos.mouse=min.global+0.5*mouse.width+((mouse.pos-1)*mouse.width)
mouse.positions=c(mouse.positions,pos.mouse)
#line color
color=color1
if(tolower(max.tissues.mouse[mouse.pos])==tolower(max.tissues.human[human.pos])){
color=max.colors.mouse[mouse.pos]
human.colors=c(human.colors,human.pos)
mouse.colors=c(mouse.colors,mouse.pos)
}
if(max.tissues.human[human.pos]=="TestisInterstitial"&&max.tissues.mouse[mouse.pos]==
"testis"){
color=max.colors.mouse[mouse.pos]
human.colors=c(human.colors,human.pos)
mouse.colors=c(mouse.colors,mouse.pos)
}
if(max.tissues.human[human.pos]=="WholeBrain"&&max.tissues.mouse[mouse.pos]=="cortex"){
color=max.colors.mouse[mouse.pos]
human.colors=c(human.colors,human.pos)
mouse.colors=c(mouse.colors,mouse.pos)
}
if(max.tissues.human[human.pos]=="WholeBrain"&&max.tissues.mouse[mouse.pos]==
"hippocampus"){
color=max.colors.mouse[mouse.pos]
human.colors=c(human.colors,human.pos)
mouse.colors=c(mouse.colors,mouse.pos)
}
if(max.tissues.human[human.pos]=="PancreaticIslets"&&max.tissues.mouse[mouse.pos]==
"pancreas"){
color=max.colors.mouse[mouse.pos]
human.colors=c(human.colors,human.pos)
mouse.colors=c(mouse.colors,mouse.pos)
}
if(max.tissues.human[human.pos]=="fetalliver"&&max.tissues.mouse[mouse.pos]=="liver"){
color=max.colors.mouse[mouse.pos]
human.colors=c(human.colors,human.pos)
mouse.colors=c(mouse.colors,mouse.pos)
}
if(max.tissues.human[human.pos]=="spinalcord"&&max.tissues.mouse[mouse.pos]==
"spinalcordupper"){
```

```
color=max.colors.mouse[mouse.pos]
human.colors=c(human.colors,human.pos)
mouse.colors=c(mouse.colors,mouse.pos)
}
if(max.tissues.human[human.pos]=="Adipocyte"&&max.tissues.mouse[mouse.pos]==
"adiposetissue"){
color=max.colors.mouse[mouse.pos]
human.colors=c(human.colors,human.pos)
mouse.colors=c(mouse.colors,mouse.pos)
}
#draw line
lines(c(pos.hum,pos.mouse),c(pos1,pos2),type="l",col=color)
}
}
}
homologs=max(length(unique(human.positions)),length(unique(mouse.positions)))
tissues=max(length(unique(human.colors)),length(unique(mouse.colors)))
return(c(homologs,tissues))
}

#get.cluster
#-----------
get.cluster=function(chromosome,clusternr,species){
chromosome=as.character(chromosome)
if(species=="mouse"){
x=mouse.cluster.table[which(mouse.cluster.table[,3]==chromosome),]
cluster=x[which(x[,8]==as.character(clusternr)),]
}
if(species=="human"){
x=human.cluster.table[which(human.cluster.table[,3]==chromosome),]
cluster=x[which(x[,8]==as.character(clusternr)),]
}
return(cluster)
}

basisplot=function(name1,min1,max1,pos1,name2,min2,max2,pos2){
min=min(min1,min2)
max=max(max1,max2)
pos=c(pos1,pos2)
y=c(10,11)
mmi=c(0.1,1.5,1,1.5)
par(mai=mmi)
plot(y,xlim=c(min,max),ylim=c(-25,20),axes=F,xlab="",ylab="")
abline(h=pos)
mtext(name1,2,par(las=1),font=2,at=pos1,cex=0.8)
mtext(name2,2,par(las=1),font=2,at=pos2,cex=0.8)
}

get.startsides=function(chrom,min,max,species){
if(species=="mouse"){
chrhash=mouse.chrhash
}
if(species=="human"){
chrhash=human.chrhash
}
startsides=chrhash[chrhash[,1]==chrom&as.numeric(chrhash[,2])>=
min&as.numeric(chrhash[,2])<=max,]
return(startsides)
}

get.homolog=function(ensembl_mouse_id){
homolog=as.character(unique(homology.table[which(homology.table[,1]==
```

```
ensembl_mouse_id),5]))
return(homolog)
}
#end functions

#readin cluster matching table
#------------------------------
table=read.csv(file="/home/dinkelac/data/mouse/tables/syntheny.table.csv",
header=TRUE,sep="\t")

#for each line
mouse.clusters=unique(as.character(table[,1]))

#plot syntheny clusters best hits only
#-------------------------------------
setwd("/home/dinkelac/data/mouse/plots/syntheny_plots_tissues/")

for(i in 105:length(mouse.clusters)){
lines=table[as.character(table[,1])==mouse.clusters[i],]
human.hits=as.character(lines[,5])[lines[,6]==max(lines[,6])]
if(!is.na(human.hits)){
print(i)
plot.cluster(mouse.clusters[i],human.hits)
name=paste(mouse.clusters[i],"_",human.hits,"_tissues.eps",sep="")
dev.copy2eps(file=name)
}
}
```

1.1.23 Further R functions

```
#barplots.R
#----------
#zum Auswerten von results_validate_ntuples.txt

a = barplot(rbind(obs.b7hi.lo, mean.b7hi.lo), beside=T, col=c("red","white"), ylim=c(0,170))

for (i in 1:7)
 ebars(a[2*i], mean.b7hi.lo[i], mean.b7hi.lo[i]+z.95*sd.b7hi.lo[i], mean.b7hi.lo[i]-z.95*
 sd.b7hi.lo[i])

#text in graphik
text((a[5]+a[6])/2, 30, "**")
text((a[7]+a[8])/2, 20, "**")
text((a[9]+a[10])/2, 10, "***")
text((a[13]+a[14])/2, 10, "***")

title(main="number of clusters of size k in B7 hi vs. lo")
dev.copy2eps(file="b7hi_lo_clusters.eps")

#functions
#---------
barplot.special = function(obs, perm, ...) {
  z.95 = qnorm(0.975)
  mean.spec = apply(perm, 2, mean, na.rm=T)
  sd.spec   = apply(perm, 2, sd,   na.rm=T)
  aa = barplot(rbind(obs, mean.spec), beside=T, col=c("red", "white"),
    ylim = c(0, 1.5*max(c(obs,mean.spec))), ... )
  for (i in 1:length(obs))
    ebars(aa[2*i], mean.spec[i], mean.spec[i]+z.95*sd.spec[i], mean.spec[i]-z.95*sd.spec[i])
```

```
}

calc.p.empir = function(obs,perm) {
  N = nrow(perm)
  P = vector(len=length(obs))
  names = c("doublets","triplets","quadruplets","quintuplets","6-tuples","7-tuples","8-tuples")
  for (i in 1:length(obs))
    P[i] = ifelse(obs[i]==0, NA, sum(perm[,i]>obs[i], na.rm=T)/N )
  for (i in 1:length(obs))
    print(paste("P value for ", names[i], ":   ", P[i], sep=""))
}

#funktion dist genloc
#--------------------
#this program calculates the distance matrix of two indexvectors X,Y
#of a genelist with chromosomelokation, here: X=Y

#Objekte
#-------
#chrs.list
#sapply nimmt immer den ersten eintrag
#achtung falsches objekt!!!

#library
#-------
#library(gnf1musb,lib.loc="/home/dinkelac/R-libs/")

#chrom.location = mget(ls(env=gnf1musbCHRLOC), env=gnf1musbCHRLOC)
#chromosomes = sapply(chrom.location, extract.chromosome)

#chrs.list = mget(ls(env=gnf1musbCHR), env=gnf1musbCHR)
#gnf1musb.ids = ls(env=gnf1musbCHR)
#chrs = sapply(chrs.list, function(x){x[1]})#chromosom
#gstr.list = mget(gnf1musb.ids, env=gnf1musbCHRLOC)
#gstr = sapply(gstr.list, function(x){abs(x[1])})#startside

#gibt listenlaenge an
#table(sapply(chrs.list, length))

#funktion
#--------
#chr.mat: distancematrix
#chrs:genname,chromosome
#dif.mat: distancematrix with values
dist.genloc = function(X, Y) {
   dX <- length(X)
   dY <- length(Y)
   chr.mat <- matrix(nr=dX, nc=dY)
   for (x in 1:dX) {
     for (y in 1:dY) {
#if both contain values
       if (!is.na(chrs[X[x]]) && !is.na(chrs[Y[y]]) ) {
#if values eq match=TRUE, else match=FALSE
         match <- unlist(chrs[X[x]]) == unlist(chrs[Y[y]])
       } else {
         match <- FALSE
       }
#if match=TRUE (gleiches chromosom) put the diagonale =0, else NA
       if (match) {
         chr.mat[x,y] <- 0
        } else {
         chr.mat[x,y] <- NA
```

```
      }
    }
  }
#outer calculates the distance of each element in X with each in Y
  diff.mat <- outer(gstr[X], gstr[Y], "-")
  result.mat <- diff.mat + chr.mat
  return(abs(result.mat))
}

#speichern
#---------
#text="dist.mat.test = dist.genloc(tra.index, tra.index)"
#save(chrs,gstr,tra.index,gstr.list,gnf1musa.ids,dist.genloc,file="batch/mydata.rda")

#system.time
#system.time(eval(parse(text=text)))
#[1] 1012.997    3.869 1017.400    0.000    0.000

#usage
#-----
#siehe session-2006-11-17.R

#------------
#no.of.neighb.star <- vector(length=1000)

#for (i in 1:1000) {
#  neighb.star <- outer(boot.index[i,], boot.index[i,], dist.genloc)
#  upper.star <- upper.tri(neighb.star)
#  noofneighb <- sum(neighb.star[upper.star], na.rm=T)
#  no.of.neighb.star[i] <- noofneighb
#}

#ebars.R
#-------
#description see errorbars.R

ebars <- vbars <- function (x, y0, y1, y2)
{
  for(i in 1:length(x))
  {
    if(y0[i]!=0 & !is.na(y0[i]) & y1[i]!=0 & !is.na(y1[i]) &
      (y0[i]!=y2[i]))
    {
      arrows (x[i], y0[i], x[i], y1[i], angle=90, length=0.05)
    }
    if(y0[i]!=0 & !is.na(y0[i]) & y2[i]!=0 & !is.na(y2[i]) &
      (y0[i]!=y2[i]))
    {
      arrows (x[i], y0[i], x[i], y2[i], angle=90, length=0.05)
    }
  }
}

#errorbars.R
#-----------
#description
#-----------
# produce vertical error bars on a plot-- aliased vbars in anticipation
# of the day when I need hbars to plot horizontal error bars....
#
# error bars originate at (x, y0) and radiate to y0 + y1 and y0 - y2
#
```

```
# example (adds +- 1.0 SE to a barplot of means for data vector x):
#
#usage
#-----
# y.mean <- mean(x) # mean
# y.se <- sqrt(var(x,na.rm=TRUE)/length(x)) # std err
# z <- barplot(y.mean,plot=FALSE) # location of bars
# barplot(y.mean,...) # the real plot
# ebars(z,y.mean,y.mean+y.se,y.mean-y.se) # add error bars
#
# --Mike C.
#---------------

#Funktionen
#----------
ebars <- vbars <- function (x, y0, y1, y2)
{
  for(i in 1:length(x))
  {
    if(y0[i]!=0 & !is.na(y0[i]) & y1[i]!=0 & !is.na(y1[i]) &
      (y0[i]!=y2[i]))
    {
      arrows (x[i], y0[i], x[i], y1[i], angle=90, length=0.05)
    }
    if(y0[i]!=0 & !is.na(y0[i]) & y2[i]!=0 & !is.na(y2[i]) &
      (y0[i]!=y2[i]))
    {
      arrows (x[i], y0[i], x[i], y2[i], angle=90, length=0.05)
    }
  }
}

plot.w.ebars <- function(X, ...) {
  y.mean = apply(X, 2, mean, na.rm=T)
  y.se = sqrt(apply(X, 2, var, na.rm=T)/nrow(X))
  z <- barplot(y.mean, plot=F)
  barplot(y.mean, ...)
  for (i in 1:length(y.mean)) {
    ebars(z[i], y.mean[i], y.mean[i]+y.se[i], y.mean[i]-y.se[i])
  }
}

extract.chromosome = function(x) {
  if (is.na(x)) {
    return( NA ) }
  else {
    n = names(x)
    return( as.character(n[1]) )
  }
}

#findRedundant.R
#---------------

findRedundant=function(liste){
#sucht doppelte gensymbole
symbols=as.character(unlist(liste))
doppelte=liste[is.d=duplicated(symbols)]
doppelte=as.character(doppelte[!is.na(doppelte)])
#indices zu symbolen
paare=list()
 for(i in 1:length(doppelte)){
```

```
 paare[[i]]=as.numeric(which(liste==doppelte[i]))
 }
return(paare)
}
#---------------

testStartside=function(liste){
paare=list()
 for(i in 1:length(liste)){
 #zeile
 y=liste[[i]]
 #print (i)
 is.d=duplicated(startside.ids[y])
 #gibt 2.doppelte
 doppelte=y[is.d]
 #gibt startside werte,die doppelt auftreten
 #werte=as.numeric(startside.ids[doppelte])#!!!
 werte=(startside.ids[doppelte])
 #schmeiss nas raus
 raus=!is.na(werte)
 werte=werte[raus]
 #wenn werte doppelt auftreten indices abspeichern
 z=vector()
  for(j in 1:length(y)){
  z[j]=is.element(startside.ids[y[j]],werte)
  }
 paare[[i]]=y[z]
 }
return(paare)
}
#--------------
testChromosom=function(liste){
paare=list()
 for(i in 1:length(liste)){
 #zeile
 y=liste[[i]]
 is.d=duplicated(chromosome.ids[y])
 #gibt 2.doppelte
 doppelte=y[is.d]
 #gibt startside werte,die doppelt auftreten
 werte=chromosome.ids[doppelte]
 #wenn werte doppelt auftreten indices abspeichern
 z=vector()
  for(j in 1:length(y)){
  z[j]=is.element(chromosome.ids[y[j]],werte)
  }
 paare[[i]]=y[z]
 }
return(paare)
}
#----------------------------
testZeile=function(liste,object,spalte){
paare=list()
 for(i in 1:length(liste)){
 #zeile
 y=liste[[i]]
 #doppelt?
 is.d=duplicated(object[y,spalte])
 #gibt 2.doppelte
 doppelte=y[is.d]
 #gibt werte,die doppelt auftreten
 werte=object[doppelte,spalte]
```

```
 #wenn werte doppelt auftreten indices abspeichern
 z=vector()
  for(j in 1:length(y)){
  z[j]=is.element(object[y[j],spalte],werte)
  }
 paare[[i]]=y[z]
 }
return(paare)
}

#-----------------------
takefirst=function(list){
for (i in 1:length(list)){
list[i]=list[[i]][1]
}
return(list)
}
#---------------------------
selectFirst=function(list){
for (i in 1:length(list)){
list[[i]]=list[[i]][1]
}
list=unlist(list)
list=list[!is.na(list)]
list=sort(list)
return(list)
}

#human.print.cluster.R
#------------------------
#This function displays gene clusters
#input:dataframe x with
#x[,3]=Genesymbol,x[,4]=chromosome,folgende spalten sind startside, clustersize,
#clusternr, tissue

print.cluster=function(x,col){
#chromosomen und startside von allen genen
setwd("/home/dinkelac/data/human/")
chrhash=read.table("chrhash.txt",sep="\t")
#-----------------------

#genesymbols
symbols=as.character(x[,3])
startsides=as.character(x[,5])
chromosome=x[1,4]
clusternr=x[1,7]
clustersize=x[1,6]

#all genes on this chromosome
#index
all.index=which(chrhash[,2]==as.character(chromosome))
#affyid,chr,startside
d=chrhash[all.index,]

#start inner functions
#----------------------
#1 get.first

get.first=function(x){
y=vector("list",length(x))
for(i in 1:length(x)){
y[[i]]=x[[i]][1]
```

```
y=unlist(y)
}
return(y)
}
#2 basisplot
basisplot=function(min,max){
y=c(10,11)
plot(y,xlim=c(min,max),ylim=c(-10,10),axes=F,xlab="startside (bp)",ylab="")
#lines(c(min,min),c(-1,1),type="l")
abline(h=c(-8,8))
#box()
}
#3 gene
gene=function(x,farbe){
description=symbols
y=x
x=c(x,x)
x=sort(x)
i=1
while(i <=length(x)){
lines(c(x[i],x[i+1]),c(-1,1),type="l",col=farbe)
#startside

text(x,-2,x,srt=90,col=farbe,cex=0.8,adj=1)
text(y,2,description,srt=90,col=farbe,cex=0.7,adj=0)
i=i+2
title(main=paste("Chromosome",chromosome,", Cluster",clusternr,"(",clustersize,"Genes)"))
}
}
#4 other genes
othergenes=function(x,farbe){
#description=symbols
y=x
x=c(x,x)
x=sort(x)
i=1
while(i <=length(x)){
lines(c(x[i],x[i+1]),c(-1,1),type="l",col=farbe)
#startside
#text(x,-2,x,srt=90,col=farbe,cex=0.8,adj=1)
#text(y,2,description,srt=90,col=farbe,cex=0.8,adj=0)
i=i+2
}
}
#end inner functions
#--------------------
startside=abs(as.numeric(get.first(strsplit(startsides,"/"))))

min=min(abs(startside))
max=max(abs(startside))

#alle mit startside zwischen min,max
e=d[,3]>min&d[,3]<max#which genes are in range T,F
f=d[e,]#genes in range
g=f[,3]#startsides in range

basisplot(min,max)

othergenes(g,"black")
gene(startside,col)
}
```

```
#usage: print.cluster=function(x,col){
#  print.cluster(x,"black")

#basisplot(min,max)
#gene(startside,"black")

my.paired.ttest = function(vec) {
  x = vec[c(1,3,5)]
  y = vec[c(2,4,6)]
  t = t.test(x,y,paired=T)
  return(c(t$statistic, t$p.value))
}

#pasteList.R
#m.dinkelacker@dkfz.de
#date: 4.7.2008

#This function pastes more than one entry of a list of lists together

pasteList = function(list)
{
result=list()
        for(i in 1:length(list)){
        tiss=unlist(l[i])
        res=tiss[1]
        if(length(tiss)>1){
                for(j in 2:length(tiss)){
                res=paste(res,tiss[j], sep="/")
                }
        }
        result[i]=res
        result=unlist(result)
        }
return(result)
}

#plot.cluster.R
#------------------------
#With this function you can compare the homologs of two clusters one on mouse and one on
#human chromosomes, ex.: mouse: 1:10, human: 1:12
#usage: compare.clusters(1,10,1,12,200000,200000)

#test (spaeter input)
#--------------------
#mouse_chromosome=3
#mouse_clusternr=6
#human_chromosome=1
#human_clusternr=10
#dist1=280000
#dist2=280000

#source("/home/dinkelac/R-functions/compare.clusters.R")

#-----
compare.clusters=function(mouse_chromosome,mouse_clusternr,human_chromosome,human_clusternr,
dist1,dist2){

#print.two.clusters=function(x,col){

#0. get homologietabelle
#-----------------------
```

```
#1. get chromosomes, startside of all mouse and all human genes
#---------------------------------------------------------------
#human_chrhash.txt
human_chrhash=read.table("/home/dinkelac/data/mouse/tables/human_chrhash.txt",sep="\t")

#mouse_chrhash.txt
mouse_chrhash=read.table("/home/dinkelac/data/mouse/tables/mouse_chrhash.txt",sep=" ")

#*chrhash

#2. readin cluster tables human,mouse
#-------------------------------------
mouse_clusters=read.csv(file="/home/dinkelac/data/mouse/tables/
mouse_clusters_tra.index.5x.csv")

human_clusters=read.csv(file="/home/dinkelac/data/human/tables/
human_clusters_tra.index1.5x.csv")

#3. get genes in cluster
#-----------------------
#human genes on chromosome
human.genes=human_chrhash[which(human_chrhash[,2]==as.character(human_chromosome)),]
#*d

#mouse genes on chromosome
mouse.genes=mouse_chrhash[which(mouse_chrhash[,2]==as.character(mouse_chromosome)),]

#human genes in cluster
human.cluster=human_clusters[which(human_clusters[,6]==as.character(human_clusternr)&
human_clusters[,3]==as.character(human_chromosome)),]

#mouse genes in cluster
mouse.cluster=mouse_clusters[which(mouse_clusters[,6]==as.character(mouse_clusternr)&
mouse_clusters[,3]==as.character(mouse_chromosome)),]

#4. get startsides
#-----------------
#human
human.startsides=abs(as.numeric(get.first(strsplit(as.character(human.cluster[,4]),"/"))))
names(human.startsides)=human.cluster[,1]
#symbole der human.startsides
human.symbols=as.character(human.cluster[,2])
names(human.symbols)=human.cluster[,1]

human.min=min(abs(human.startsides))
human.max=max(abs(human.startsides))

#mouse
mouse.startsides=abs(as.numeric(get.first(strsplit(as.character(mouse.cluster[,4]),"/"))))
names(mouse.startsides)=mouse.cluster[,1]
#symbole der mouse.startsides
mouse.symbols=as.character(mouse.cluster[,2])
names(mouse.symbols)=mouse.cluster[,1]

mouse.min=min(abs(mouse.startsides))
mouse.max=max(abs(mouse.startsides))

global.min=min(human.min,mouse.min)
global.max=max(human.max,mouse.max)
#*startsides
```

```
#5. get all startsides
#---------------------
#human
human.all.startsides=abs(human.genes[human.genes[,3]>human.min&human.genes[,3]<human.max,3])
names(human.all.startsides)=human.genes[human.genes[,3]>human.min&human.genes[,3]<
human.max,1]
#mouse
mouse.all.startsides=abs(mouse.genes[mouse.genes[,3]>mouse.min&mouse.genes[,3]<mouse.max,3])
names(mouse.all.startsides)=mouse.genes[mouse.genes[,3]>mouse.min&mouse.genes[,3]<
mouse.max,1]

#*g

#6. plot the base lines
#----------------------
basisplot("human",human.min,human.max,7,"mouse",mouse.min,mouse.max,-7)
mtext(paste("chromosome:",human_chromosome,sep=""),2,par(las=1),at=5.5,cex=0.8)
mtext(paste("clusternr.:",human_clusternr,sep=""),2,par(las=1),at=4,cex=0.8)
mtext(paste("chromosome:",mouse_chromosome,sep=""),2,par(las=1),at=-8.5,cex=0.8)
mtext(paste("clusternr.:",mouse_clusternr,sep=""),2,par(las=1),at=-10,cex=0.8)
title(main="Clustering of tissue restricted antigens (TRAs)\n in mouse and human")

#7. calculate some numbers
#-------------------------
#7.1. no of genes in the cluster
#-------------------------------
#human
no.human.genes=sum(!is.na(human.all.startsides))

#mouse
no.mouse.genes=sum(!is.na(mouse.all.startsides))

#7.2. no of TRAs in the cluster
#------------------------------
#human
no.human.tras=human.cluster[1,5]

#mouse
no.mouse.tras=mouse.cluster[1,5]

#7.3. calculate tra density
#--------------------------
#human
human.density=round(no.human.tras/no.human.genes*100)
human.percent=paste(human.density,"%",sep="")

#mouse
mouse.density=round(no.mouse.tras/no.mouse.genes*100)
mouse.percent=paste(mouse.density,"%",sep="")

#7.4. calculate the range
#------------------------
#human
human.range=(human.max-human.min)/1000

#mouse
mouse.range=(mouse.max-mouse.min)/1000

mtext(paste(no.human.genes,"genes"),4,par(las=1),at=7,cex=0.8)
mtext(paste(no.human.tras,"TRAs"),4,par(las=1),at=5.5,cex=0.8)
mtext(paste(human.percent,"density"),4,par(las=1),at=4,cex=0.8)
mtext(paste(human.range,"kbp"),4,par(las=1),at=2.5,cex=0.8)
```

```
mtext(paste(no.mouse.genes,"genes"),4,par(las=1),at=-7,cex=0.8)
mtext(paste(no.mouse.tras,"TRAs"),4,par(las=1),at=-8.5,cex=0.8)
mtext(paste(mouse.percent,"density"),4,par(las=1),at=-10,cex=0.8)
mtext(paste(mouse.range,"kbp"),4,par(las=1),at=-11.5,cex=0.8)

#range

#8. plot genes
#-------------
#rescaling parameters
human.parameters=get.parameters(human.startsides,global.min,global.max)
mouse.parameters=get.parameters(mouse.startsides,global.min,global.max)

#human
rescaled.human.all.startsides=rescale(human.all.startsides,human.parameters)
plot.all.genes(rescaled.human.all.startsides,"black",8,6)

#mouse
rescaled.mouse.all.startsides=rescale(mouse.all.startsides,mouse.parameters)
plot.all.genes(rescaled.mouse.all.startsides,"black",-8,-6)

#human
rescaled.human.startsides=rescale(human.startsides,human.parameters)

plot.genes(rescaled.human.startsides,"red",c(8,6))
plot.symbols(relax(rescaled.human.startsides,dist1),human.symbols,"red",10,0)

#mouse
rescaled.mouse.startsides=rescale(mouse.startsides,mouse.parameters)

plot.genes(rescaled.mouse.startsides,"red",c(-8,-6))
plot.symbols(relax(rescaled.mouse.startsides,dist2),mouse.symbols,"red",-10,1)

#9. find homologs
#----------------
#number.homolog.genes=connect.homologs(rescaled.human.all.startsides,
#rescaled.mouse.all.startsides,6,-6,"black")

number.homolog.tras=connect.homologs(rescaled.human.startsides,rescaled.mouse.startsides,6,-6,
"red")
mtext(paste(number.homolog.tras,"homologs"),2,par(las=1),at=0,cex=0.8)

}#end of function

#functions
#---------
#1. get.first
#-----------
#gets the first entry in a list of lists

get.first=function(x){
y=vector("list",length(x))
for(i in 1:length(x)){
y[[i]]=x[[i]][1]
y=unlist(y)
}
return(y)
}

#2. basisplot
#-----------
```

```
#plots the straight lines at pos in the range of min max
basisplot=function(name1,min1,max1,pos1,name2,min2,max2,pos2){
min=min(min1,min2)
max=max(max1,max2)
pos=c(pos1,pos2)
y=c(10,11)
mmi=c(1.5,1.5,1,1.5)
par(mai=mmi)
plot(y,xlim=c(min,max),ylim=c(-15,15),axes=F,xlab="startside (bp)",ylab="")
abline(h=pos)
mtext(name1,2,par(las=1),font=2,at=pos1,cex=0.8)
mtext(name2,2,par(las=1),font=2,at=pos2,cex=0.8)
}

#3. plot.all.genes
#-----------------
#plots vertical lines in positions pos
plot.all.genes=function(x,farbe,min,max){
#description=symbols
pos=c(min,max)
y=x
x=c(x,x)
x=sort(x)
i=1
while(i <=length(x)){
lines(c(x[i],x[i+1]),pos,type="l",col=farbe)
#startside
#text(x,-2,x,srt=90,col=farbe,cex=0.8,adj=1)
#text(y,2,description,srt=90,col=farbe,cex=0.8,adj=0)
i=i+2
}
}

#4. plot.genes
#-------------
##plot.genes=function(x,farbe,pos,t1,textpos){
plot.genes=function(x,farbe,pos){
##descript=symbols[names(x)]
y=x
x=c(x,x)
x=sort(x)
i=1
while(i <=length(x)){
lines(c(x[i],x[i+1]),pos,type="l",col=farbe)
#startside

#text(x,-2,x,srt=90,col=farbe,cex=0.8,adj=1)
##text(y,pos[1]+(t1),descript,srt=90,col=farbe,cex=0.7,adj=textpos)
i=i+2

#text(x[1],pos[2],paste("Chromosome",chromosome,"Cluster",clusternr,
#"(",clustersize,"Genes)"))
}
}

#5. check.cluster
#----------------
#draws the cluster in an extra plot to check
check.cluster=function(a,b){
x11()
max=max(length(a),length(b))
if(length(a)>length(b)){
```

```
c=b
b=a
a=c
}
plot(b,rep(c(10),length(b)))
points(a,rep(10,length(a)),col="red")
points(b,rep(8,length(b)))
}

#6. get.parameters
#-----------------
#estimates parameters for rescaling tras to the whole range of the plot
get.parameters=function(vector,global.min,global.max){
parameters=c()
#schleife
for(i in 1:10){
local.min=min(vector)
diff=abs(global.min-local.min)
vector=vector-diff
local.max=max(vector)
factor=global.max/local.max#alles nach re
vector=vector*factor
parameters=c(parameters,diff,factor)
}
local.min=min(vector)
diff=abs(global.min-local.min)
parameters=c(parameters,diff)

return (parameters)
}

#7. rescale
#----------
rescale=function(vector,parameters){
i=1
while(i<=20){
vector=vector-parameters[i]
vector=vector*parameters[i+1]
i=i+2
}
vector=vector-parameters[21]
return(vector)
}

#8. plot.symbols
#---------------
#pos = y-achse,dir 0 nach oben, 1 nach unten
plot.symbols=function(vector,symbol.vector,color,pos,dir){
affyids=names(vector)
symbols=symbol.vector[affyids]
text(vector,pos,symbols,srt=90,col=color,cex=0.65,adj=dir)
}

#9. relax
#--------
#stretches the genesymbol entrys to the side
relax=function(vector,threshold){
abs.dist=(max(vector)-min(vector))/length(vector)
max=max(vector)
dist=min(abs.dist,threshold)
vector=sort(vector)
```

```
#durchlaufe den vector
if(length(vector)>2){
for(i in 1:(length(vector)-2)){
#berechnet distanz
act.dist=vector[i+1]-vector[i]
if(act.dist<dist){
a=vector[i]+dist
#kleiner als das letzte element
if(a<max){
vector[i+1]=vector[i]+dist
}else{
vector[i+1]=max
}
}
}
}

return(vector)
}

#10. get.homolog
#---------------
#put in human,get mouse affyid
get.homolog=function(affyid){
homologs=read.csv(file="/home/dinkelac/data/mouse/tables/tra.mouse.human.homologs.csv")
line=which(homologs[,7]==affyid)
homolog=as.character(homologs[line,2])
if(length(homolog)==0){
return (NA)
}else{
return (homolog)
}
}

#11. connect.homologs
#--------------------
#verbindet die homologen TRAs in mouse und human cluster
connect.homologs=function(human.genes,mouse.genes,pos1,pos2,color){
#affyIDs1
human.affyIDs=names(human.genes)
mouse.affyIDs=names(mouse.genes)
no=0
#berechne anzahl an gefundenen
for(i in 1:length(human.affyIDs)){
human.affyID=human.affyIDs[i]
mouse.affyID=get.homolog(human.affyID)[1]
if(!is.na(mouse.affyID)&is.element(mouse.affyID,mouse.affyIDs)){
#draw line
human.pos=human.genes[human.affyID]
mouse.pos=mouse.genes[mouse.affyID]
lines(c(human.pos,mouse.pos),c(pos1,pos2),type="l",col=color)
no=no+1
}
}
return(no)
}

#plotten.eps.R
#-------------

#library
library(gnf1musa, lib.loc="~/R-libs")
```

```
library(affy)
library(gngnf1musacdf, lib.loc="~/R-libs")

setwd("/home/dinkelac/data/mouse/tra_5x_median/")
load("database5x.rda")

#Objekte
tissue.colors=read.table(file="~/data/mouse/tables/tissue_colors.csv")
farben=as.vector(tissue.colors[2:62,3])

farben.zuordnung=c(9,1,2,7,8,3,24,41,58,12,13,15,16,26,27,33,35,48,49,52,6,17:21,23,34,37,
40,42,53,59,60,36,25,29,46,4,10,11,30,31,32,45,61,44,56,38,39,14,22,47,50,51,54,55,57,5,28,43)

setwd("/home/dinkelac/data/mouse/tra_5x_median/pictures")

#length(genname)
for (i in 1:2){
plotten(genname[i])

plotten= function(gene.name){
#filename=paste(gene.name,".png",sep="")
filename=paste(gene.name,".eps",sep="")
#postscript(file=filename)
s=genesymbol[gene.name]
titel=paste(s[[1]], gene.name,sep="\n")
x11(w=10,h=7)
par(las=2)
mmi=c(2.2,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
 barplot(exp(mean.vsn[gene.name,])[farben.zuordnung],col=farben[farben.zuordnung],
 cex.names=0.8,cex.axis=0.8)
 med.act=exp(median(mean.vsn[gene.name,]))
 abline(h=med.act,lty=2,col="black")
 abline(h=5*med.act,lty=2,col="red")
 title(main=titel)
legend("topright",pch="-",c("median","5xmedian"),col=c("black","red"),cex=0.8)
dev.copy2eps(file=filename)
#png(filename=filename,width=350,heigth=500)
#dev.off()
graphics.off()
}
}

#funktion plotten
#-----------------
#plottet die Genexpression analog zu symatlas

#Objekte
tissue.colors=read.table(file="~/data/mouse/tables/tissue_colors.csv")
farben=as.vector(tissue.colors[2:62,3])

farben.zuordnung=c(9,1,2,7,8,3,24,41,58,12,13,15,16,26,27,33,35,48,49,52,6,17:21,
23,34,37,40,42,53,59,60,36,25,29,46,4,10,11,30,31,32,45,61,44,56,38,39,14,22,47,50,51,
54,55,57,5,28,43)

plotten= function(gene.name){
filename=paste(gene.name,".png",sep="")
s=genesymbol[gene.name]
titel=paste(s[[1]], gene.name,sep="\n")
par(las=2)
mmi=c(2.2,1.0477939,1.0477939,0.5366749)
par(mai=mmi)
```

```
 barplot(exp(mean.vsn[gene.name,])[farben.zuordnung],col=farben[farben.zuordnung],
 cex.names=0.8,cex.axis=0.8)
 med.act=exp(median(mean.vsn[gene.name,]))
 abline(h=med.act,lty=2,col="black")
 #abline(h=5*med.act,lty=2,col="red")
 abline(h=10*med.act,lty=2,col="blue")
 title(main=titel)
legend("topright",pch="-",c("median","10xmedian"),col=c("black","blue"),cex=0.8)
#png(filename=filename,width=350,heigth=500)
#dev.off()
}

#usage:
gene.name="gnf1m00001_at"
plotten(gene.name)

#automatische pngs erstellen
#length(genname)
for (i in 1:length(y)){
filename=paste(y[i],".png",sep="")
png(filename,width=900,heigth=800)
plotten(y[i])
dev.off()
}

#gene 10x,nicht 5x
genname.5x=gnf1musa.ids[tra.index3]
genname.10x=gnf1musa.ids[tra.index4]
x=is.element(genname.10x[],genname.5x)

#funktion plotten
#----------------
#plottet die Genexpression analog zu symatlas

#Objekte
library(gnf1musa, lib.loc="~/R-libs")
gnf1musa.ids=ls(env=gnf1musaCHR)
genesymbol.ids=mget(gnf1musa.ids,env=gnf1musaSYMBOL)
#tissue.colors=read.table(file="~/data/mouse/tables/tissue_colors.csv")
tissue.colors=read.table(file="~/data/mouse/tables/tissue_colors_neu.csv")
#farben=as.vector(tissue.colors[2:62,3])
farben=as.vector(tissue.colors[1:61,3])

#farben.zuordnung=c(9,1,2,7,8,3,24,41,58,12,13,15,16,26,27,33,35,48,49,52,6,17:21,23,
#34,37,40,42,53,59,60,36,25,29,46,4,10,11,30,31,32,45,61,44,56,38,39,14,22,47,50,51,54,
#55,57,5,28,43)
farben.zuordnung=c(7,45,1,9,14,22,47,56,4,8,32,54,10,11,34,60,37,36,40,23,6,59,25,17:21,
5,28,2,30,3,12,13,16,24,26,27,35,41,43,48,49,52,15,33,58,61,39,44,51,38,29,46,53,42,50,
57,31,55)


plotten= function(gene.name){
filename=paste(gene.name,".eps",sep="")
s=genesymbol.ids[gene.name]
titel=paste(s[[1]], gene.name,sep="\n")
x11(w=10,h=7)
par(las=2)
mmi=c(2.2,1.0477939,1.0477939,0.5366749)
par(mai=mmi)

barplot(exp(mean.vsn[gene.name,])[farben.zuordnung],col=farben[farben.zuordnung],
cex.names=0.8,cex.axis=0.8)
```

```
#barplot(exp(mean.vsn[gene.name,]),col=farben,cex.names=0.8,cex.axis=0.8)
 med.act=exp(median(mean.vsn[gene.name,]))
 abline(h=med.act,lty=2,col="black")
 abline(h=5*med.act,lty=2,col="red")
 title(main=titel)
legend("topright",pch="-",c("median","5xmedian"),col=c("black","red"),cex=0.8)
}

#usage:
gene.name="gnf1m00001_at"
plotten(gene.name)

#auf viele gene anwenden
for (i in 1:length(genname)){
plotten(genname[i])
}
#speichern
dev.copy2eps(file=gene.name)

#plot_ybmat.R
#------------
#colour settings for heatmaps

#library
#-------
library(lattice)

#functions:
#----------
#myPalette()- passt farben vom heatmap an
#myPalette2() - passt farben vom heatmap an mit bias
#plot.mymat() - levelplot
#plot.mymat.wlines() - levelplot with lines
#plot.mymat.wlines2() - levelplot with lines

#calculate z-score (Abstand eines Wertes vom Mittelwert)
#z=x-mue/sigma
z.score = function(X) ( X - apply(X,1,mean,na.rm=T) ) / apply(X,1,sd,na.rm=T)

#myPalette
#---------
#passt farben vom heatmap an
myPalette <- function(data, low="blue", high="yellow", mid="black", k=50) {
  low <- col2rgb(low)/255
  high <- col2rgb(high)/255
  if (is.null(mid)){
    r <- seq(low[1], high[1], len=k)
    g <- seq(low[2], high[2], len=k)
    b <- seq(low[3], high[3], len=k)
  }
  if (!is.null(mid)) {
    ratio <- abs(min(data, na.rm=T))/diff(range(data, na.rm=T))
    if (ratio >= 1) {
      data <- as.vector(data) - mean(as.vector(data), na.rm=T)
      ratio <- max(data, na.rm=T)/diff(range(data, na.rm=T))
    }
    k1 <- round(ratio*k)
    k2 <- k - k1
    mid <- col2rgb(mid)/255
    r <- c(seq(low[1], mid[1], len=k1), seq(mid[1], high[1], len=k2))
    g <- c(seq(low[2], mid[2], len=k1), seq(mid[2], high[2], len=k2))
    b <- c(seq(low[3], mid[3], len=k1), seq(mid[3], high[3], len=k2))
```

```
  }
  rgb(r,g,b)
}

#myPalette2
#----------
#passt farben vom heatmap an, mit bias
myPalette2 = function(data, low="blue", high="yellow", mid="black", k=50, bias=1) {
  low <- col2rgb(low)/255
  high <- col2rgb(high)/255
  if (is.null(mid)){
    r <- seq(low[1], high[1], len=k)^bias
    g <- seq(low[2], high[2], len=k)^bias
    b <- seq(low[3], high[3], len=k)^bias
  }
  if (!is.null(mid)) {
    ratio <- abs(min(data, na.rm=T))/diff(range(data, na.rm=T))
    if (ratio >= 1) {
      data <- as.vector(data) - mean(as.vector(data), na.rm=T)
      ratio <- max(data, na.rm=T)/diff(range(data, na.rm=T))
    }
    k1 <- round(ratio*k)
    k2 <- k - k1
    mid <- col2rgb(mid)/255
    r <- c(seq(low[1], mid[1], len=k1)^bias, seq(mid[1], high[1], len=k2))^bias
    g <- c(seq(low[2], mid[2], len=k1)^bias, seq(mid[2], high[2], len=k2))^bias
    b <- c(seq(low[3], mid[3], len=k1)^bias, seq(mid[3], high[3], len=k2))^bias
  }
  rgb(r,g,b)
}

# lset(col.whitebg())

#plot.mymat - levelplot
#----------
plot.mymat <- function(data, col=myPalette(data, k=51), ...) {
  nr <- nrow(data)
  nc <- ncol(data)
  levelplot(t(as.matrix(data)),
            scales=list( x=list(at=seq(1,nc), labels=colnames(data), rot=90),
              y=list(at=seq(1,nr), labels=rownames(data))),
            cuts=35, col.regions=col, ...)
}

#plot.mymat.wlines - levelplot with lines
#-----------------
plot.mymat.wlines <- function(data, col=myPalette(data, k=51), v=NULL, ...) {
  nr <- nrow(data)
  nc <- ncol(data)
  levelplot(t(as.matrix(data)), panel=function(x, ...) {
    panel.levelplot(x, ...)
    for (i in 1:length(v)){
      panel.abline(v=v[i]+0.5, col="white", lwd=2)
    }
  }, scales=list( x=list(at=seq(1,nc), labels=colnames(data), rot=90),
       y=list(at=seq(1,nr), labels=rownames(data)) ),
    cuts=35, col.regions=col, ...)
}

#plot.mymat.wlines2 - levelplot with lines
#------------------
plot.mymat.wlines2 <- function(data, col=myPalette(data, k=51), v=NULL, vcols=NULL, ...) {
```

```
  nr <- nrow(data)
  nc <- ncol(data)
  levelplot(t(as.matrix(data)), panel=function(x, ...) {
    panel.levelplot(x, ...)
    if (is.null(vcols)) {
      for (i in 1:length(v)){
        panel.abline(v=v[i]+0.5, col="white", lwd=2)
      }
    } else {
      for (i in 1:length(v)){
        panel.abline(v=v[i]+0.5, col=vcols[i], lwd=2)
      }
    }
  }, scales=list( x=list(at=seq(1,nc), labels=colnames(data), rot=90),
       y=list(at=seq(1,nr), labels=rownames(data)) , cex=c(0.4,1)),
    cuts=50, col.regions=col, ...)
}

#print.cluster1.R
#----------------

#Description
#-----------
#This function displays single TRA clusters
#input x is a table with the following columns
#2-4 genesymbol, chromosome, startside
#7-9 clustersize, clusternr, tissue
#-----------

print.cluster=function(x,col){

#genesymbols
symbols=as.character(x[,2])
startsides=as.character(x[,4])
chromosome=as.character(x[1,3])
clusternr=x[1,8]
clustersize=x[1,7]

#all genes on this chromosome

all.index=which(chrhash[,2]==as.character(chromosome))
#affyid,chr,startside
d=chrhash[all.index,]

#start inner functions
#----------------------
#1 get.first

get.first=function(x){
y=vector("list",length(x))
for(i in 1:length(x)){
y[[i]]=x[[i]][1]
y=unlist(y)
}
return(y)
}
#2 basisplot
basisplot=function(min,max){
y=c(10,11)
plot(y,xlim=c(min,max),ylim=c(-10,10),axes=F,xlab="startside (bp)",ylab="")
#lines(c(min,min),c(-1,1),type="l")
abline(h=0)
```

```
#box()
}
#3 gene
gene=function(x,farbe){
description=symbols
y=x
x=c(x,x)
x=sort(x)
i=1
while(i <=length(x)){
lines(c(x[i],x[i+1]),c(-1,1),type="l",col=farbe)
#startside
text(x,-2,x,srt=90,col=farbe,cex=0.8,adj=1)
text(y,2,description,srt=90,col=farbe,cex=0.8,adj=0)
i=i+2
title(main=paste("Chromosome",chromosome,", Cluster",clusternr,"(",clustersize,"Genes)"))
}
}
#4 other genes
othergenes=function(x,farbe){
#description=symbols
y=x
x=c(x,x)
x=sort(x)
i=1
while(i <=length(x)){
lines(c(x[i],x[i+1]),c(-1,1),type="l",col=farbe)
#startside
#text(x,-2,x,srt=90,col=farbe,cex=0.8,adj=1)
#text(y,2,description,srt=90,col=farbe,cex=0.8,adj=0)
i=i+2
}
}
#end inner functions
#--------------------
startside=abs(as.numeric(get.first(strsplit(startsides,"/"))))

min=min(abs(startside))
max=max(abs(startside))

#alle mit startside zwischen min,max
e=d[,3]>min&d[,3]<max#which genes are in range T,F
f=d[e,]#genes in range
g=f[,3]#startsides in range

basisplot(min,max)

othergenes(g,"black")
gene(startside,col)

}

#usage: print.cluster=function(x,col){
#  print.cluster(x,"black")

#basisplot(min,max)
#gene(startside,"black")

#print.cluster.R
#-----------------------
#with this function, you can print clusters, ex.: mouse: 1:10, human: 1:12
```

```
#x11(h=7,w=10)
chrom=1
clusternr=8
dist=100000
#dist1=0
#print.cluster("mouse",chrom,clusternr,-5,dist,"")

#tra.symbols=print.cluster("mouse",chrom,clusternr,-5,dist,"symbols",plot="F")
#tra.startsides=print.cluster("mouse",chrom,clusternr,-5,dist,"startsides",plot="F")
#mouse.all.startsides=print.cluster("mouse",chrom,clusternr,-5,dist,"all.startsides",
#plot="F")
#***
#***FEHLER*** denn tra.startsides stimmen nicht mit mouse.all.startsides ueberein
#***
#mouse.all.symbols=print.cluster("mouse",chrom,clusternr,-5,dist,"all.symbols",plot="F")

#affys, which are also tras
#affys.tra=as.character(names(mouse.startsides))

#color.genes(tra.symbols,"mouse",affys.tra,"tras","lightgoldenrod",-5,dist)

#affys.all=as.character(names(mouse.all.startsides[1:10]))

#color.genes(mouse.all.symbols,"mouse",affys.all,"all","blue",-5,dist)

#test (spaeter input)
#---------------------
#mouse_chromosome=3
#mouse_clusternr=6
#human_chromosome=1
#human_clusternr=10
#dist=280000
#dist2=280000

#source("/home/dinkelac/R-functions/print.cluster.R")

#-----

#species="mouse"
#chromosome=chrom
#pos=-5
#textdist=dist
#sonstiges="startsides"
#plot="F"

print.cluster=function(species,chromosome,clusternr,pos,textdist,sonstiges,plot="T"){
dist=textdist
#human
#-----
if(species=="human"){
human_chromosome=chromosome
human_clusternr=clusternr

#1.1. human_chrhash.txt
#-----------------------
human_chrhash=read.table("/home/dinkelac/data/mouse/tables/human_chrhash.txt",sep="\t")

#1.2. human_clusters
#--------------------
human_clusters=read.csv(file="/home/dinkelac/data/human/tables/
human_clusters_tra.index1.5x.csv")
```

```
#1.3. human genes on chromosome
#---------------------
human.genes=human_chrhash[which(human_chrhash[,2]==as.character(human_chromosome)),]

#1.4. human genes in cluster
#---------------------------
human.cluster=human_clusters[which(human_clusters[,6]==as.character(human_clusternr)&
human_clusters[,3]==as.character(human_chromosome)),]

#1.5. get startsides
#-------------------
human.startsides=abs(as.numeric(get.first(strsplit(as.character(human.cluster[,4]),"/"))))
names(human.startsides)=human.cluster[,1]

#1.6. symbole der human.startsides
#---------------------------------
human.symbols=as.character(human.cluster[,2])
names(human.symbols)=human.cluster[,1]

human.min=min(abs(human.startsides))
human.max=max(abs(human.startsides))

#1.7. get all startsides
#-----------------------
human.all.startsides=abs(human.genes[human.genes[,3]>human.min&human.genes[,3]<
human.max,3])
names(human.all.startsides)=human.genes[human.genes[,3]>human.min&human.genes[,3]<
human.max,1]

#1.8. calculate some numbers
#---------------------------
#genes in cluster
no.human.genes=sum(!is.na(human.all.startsides))
#tras in cluster
no.human.tras=human.cluster[1,5]
#density of TRAs
human.density=round(no.human.tras/no.human.genes*100)
#density in %
human.percent=paste(human.density,"%",sep="")
#range
human.range=(human.max-human.min)/1000

if(plot!="F"){

basisplot("human",human.min,human.max,pos)
mtext(paste("chromosome:",human_chromosome,sep=""),2,par(las=1),at=pos,cex=0.8)
mtext(paste("clusternr.:",human_clusternr,sep=""),2,par(las=1),at=(pos-1.5),cex=0.8)
title(main="Clustering of tissue restricted antigens (TRAs)\n in human")

mtext(paste(no.human.genes,"genes"),4,par(las=1),at=pos,cex=0.8)
mtext(paste(no.human.tras,"TRAs"),4,par(las=1),at=(pos-1.5),cex=0.8)
mtext(paste(human.percent,"density"),4,par(las=1),at=(pos-3),cex=0.8)
mtext(paste(human.range,"kbp"),4,par(las=1),at=(pos-4.5),cex=0.8)

#rescaling parameters
#human.parameters=get.parameters(human.startsides,global.min,global.max)

#rescaled.human.all.startsides=rescale(human.all.startsides,human.parameters)
#plot.all.genes(rescaled.human.all.startsides,"black",8,6)
plot.all.genes(human.all.startsides,"black",(pos-1),(pos+1))

#rescaled.human.startsides=rescale(human.startsides,human.parameters)
```

```
#plot.genes(rescaled.human.startsides,"red",c(8,6))
plot.genes(human.startsides,"red",c((pos-1),(pos+1)))
#plot.symbols(relax(rescaled.human.startsides,dist1),human.symbols,"red",10,0)
plot.symbols(relax(human.startsides,dist),human.symbols,"red",(pos+2),0)
}#end plot

if(sonstiges =="symbols"){
return(human.symbols)
}
if(sonstiges =="startsides"){
return(human.startsides)
}
if(sonstiges =="all.startsides"){
return(human.all.startsides)
}
}#end human

insert.genes=function(genelist){
#genelist should have the information of affyIDs or genesymbols -> startsides
#match_ilMouse6v11ko_affyIDs.csv
}

#mouse
#-----
if(species=="mouse"){
mouse_chromosome=chromosome
mouse_clusternr=clusternr

#2.1. mouse_chrhash.txt
#----------------------
mouse_chrhash=read.table("/home/dinkelac/data/mouse/tables/mouse_chrhash.txt",sep=" ")

#2.2. mouse_clusters
#-------------------
#annotation 2006
mouse_clusters=read.csv(file="/home/dinkelac/data/mouse/tables/
mouse_clusters_tra.index.5x.csv")

#2.3. mouse genes on chromosome
#----------------------
mouse.genes=mouse_chrhash[which(mouse_chrhash[,2]==as.character(mouse_chromosome)),]

#2.4. mouse genes in cluster
#----------------------
mouse.cluster=mouse_clusters[which(mouse_clusters[,6]==as.character(mouse_clusternr)&
mouse_clusters[,3]==as.character(mouse_chromosome)),]

#2.5. get startsides
#-----------------
#falsch? != mouse.all.startsides
#original startsides
mouse.startsides=abs(as.numeric(get.first(strsplit(as.character(mouse.cluster[,4]),"/"))))
names(mouse.startsides)=mouse.cluster[,1]

#2.6. symbole der mouse.startsides
#----------------------
mouse.symbols=as.character(mouse.cluster[,2])
names(mouse.symbols)=mouse.cluster[,1]

mouse.min=min(abs(mouse.startsides))
mouse.max=max(abs(mouse.startsides))
```

```
#2.7. get all startsides
#----------------------
#falsch? != mouse.startsides
mouse.all.startsides=abs(mouse.genes[mouse.genes[,3]>mouse.min&mouse.genes[,3]<
mouse.max,3])

names(mouse.all.startsides)=mouse.genes[mouse.genes[,3]>mouse.min&mouse.genes[,3]<
mouse.max,1]

#symbole der mouse.all.startsides
#--------------------------------
#mouse.all.symbols=

#2.8. calculate some numbers
#--------------------------
#genes in cluster
no.mouse.genes=sum(!is.na(mouse.all.startsides))
#tras in cluster
no.mouse.tras=mouse.cluster[1,5]
#density of TRAs
mouse.density=round(no.mouse.tras/no.mouse.genes*100)
#density in %
mouse.percent=paste(mouse.density,"%",sep="")
#range
mouse.range=(mouse.max-mouse.min)/1000

if(plot!="F"){

basisplot("mouse",mouse.min,mouse.max,pos)
mtext(paste("chromosome:",mouse_chromosome,sep=""),2,par(las=1),at=pos,cex=0.8)
mtext(paste("clusternr.:",mouse_clusternr,sep=""),2,par(las=1),at=(pos-1.5),cex=0.8)
title(main="Clustering of tissue restricted antigens (TRAs)\n in mouse")

mtext(paste(no.mouse.genes,"genes"),4,par(las=1),at=pos,cex=0.8)
mtext(paste(no.mouse.tras,"TRAs"),4,par(las=1),at=(pos-1.5),cex=0.8)
mtext(paste(mouse.percent,"density"),4,par(las=1),at=(pos-3),cex=0.8)
mtext(paste(mouse.range,"kbp"),4,par(las=1),at=(pos-4.5),cex=0.8)

#mouse.parameters=get.parameters(mouse.startsides,global.min,global.max)
#rescaled.mouse.all.startsides=rescale(mouse.all.startsides,mouse.parameters)
#rescaled.mouse.startsides=rescale(mouse.startsides,mouse.parameters)

plot.all.genes(mouse.all.startsides,"black",(pos-1),(pos+1))

plot.genes(mouse.startsides,"red",c((pos-1),(pos+1)))
plot.symbols(relax(mouse.startsides,dist),mouse.symbols,"red",(pos+2),0)
}#end plot

if(sonstiges =="symbols"){
return(mouse.symbols)
}
if(sonstiges =="all.symbols"){
mouse.all.symbols=get.symbols("mouse",mouse.all.startsides)
return(mouse.all.symbols)
}

if(sonstiges =="startsides"){
return(mouse.startsides)
}
if(sonstiges =="all.startsides"){
return(mouse.all.startsides)
```

```
}
}#end mouse

}#end of function

#color.genes(mouse.all.symbols,"mouse",affys.all,"all","blue",-5,dist1)
#symbols=mouse.all.symbols
#species="mouse"
#affys=affys.all
#reference="all"
#col="blue"
#pos=-5
#dist=dist1
color.genes=function(symbols,species,affys,reference,col,pos,dist){
#list all,tras
#reference list

#mouse
if(species =="mouse"){
mouse.symbols=symbols
if(reference =="tras"){
reference.list=mouse.startsides
genes=reference.list[affys]
plot.genes(genes,col,c((pos-1),(pos+1)))
tras.symbols=relax(reference.list,dist)
genes.symbols=tras.symbols[affys]
plot.symbols(genes.symbols,mouse.symbols,col,(pos+2),0)
}
if(reference =="all"){
reference.list=mouse.all.startsides
#affy+startside
genes=reference.list[affys]
plot.genes(genes,col,c((pos-1),(pos+1)))
#location der symbols
all.symbols=relax(reference.list,dist)
#affy+startsides relaxed
genes.symbols=all.symbols[affys]
plot.symbols(genes.symbols,mouse.symbols,col,(pos-2),1)
}
}#end mouse

#human
if(species =="human"){
human.symbols=symbols
if(reference =="tras"){
reference.list=human.startsides
genes=reference.list[affys]
plot.genes(genes,col,c((pos-1),(pos+1)))
tras.symbols=relax(reference.list,dist)
genes.symbols=tras.symbols[affys]
plot.symbols(genes.symbols,human.symbols,col,(pos+2),0)
}
if(reference =="all"){
reference.list=human.all.startsides
genes=reference.list[affys]
plot.genes(genes,col,c((pos-1),(pos+1)))
all.symbols=relax(reference.list,dist)
genes.symbols=all.symbols[affys]
plot.symbols(genes.symbols,human.symbols,col,(pos-2),1)
}
}#end human
}#end of function
```

```
#functions
#---------
#1. get.first
#-----------
#gets the first entry in a list of lists

get.first=function(x){
y=vector("list",length(x))
for(i in 1:length(x)){
y[[i]]=x[[i]][1]
y=unlist(y)
}
return(y)
}

#2. basisplot
#-----------
#plots the straight lines at pos in the range of min max
basisplot=function(name,min,max,pos){
y=c(10,11)
mmi=c(1.5,1.5,1,1.5)
par(mai=mmi)
plot(y,xlim=c(min,max),ylim=c(-15,15),axes=F,xlab="startside (bp)",ylab="")
abline(h=pos)
mtext(name,2,par(las=1),font=2,at=(pos+1.5),cex=0.8)
}

#3. plot.all.genes
#-----------------
#plots vertical lines in positions pos
plot.all.genes=function(x,farbe,min,max){
#description=symbols
pos=c(min,max)
y=x
x=c(x,x)
x=sort(x)
i=1
while(i <=length(x)){
lines(c(x[i],x[i+1]),pos,type="l",col=farbe)
#startside
#text(x,-2,x,srt=90,col=farbe,cex=0.8,adj=1)
#text(y,2,description,srt=90,col=farbe,cex=0.8,adj=0)
i=i+2
}
}

#4. plot.genes
#-------------
##plot.genes=function(x,farbe,pos,t1,textpos){
plot.genes=function(x,farbe,pos){
##descript=symbols[names(x)]
y=x
x=c(x,x)
x=sort(x)
i=1
while(i <=length(x)){
lines(c(x[i],x[i+1]),pos,type="l",col=farbe)
#startside

#text(x,-2,x,srt=90,col=farbe,cex=0.8,adj=1)
##text(y,pos[1]+(t1),descript,srt=90,col=farbe,cex=0.7,adj=textpos)
```

```
i=i+2

#text(x[1],pos[2],paste("Chromosome",chromosome,"Cluster",clusternr,
#"(",clustersize,"Genes)"))
}
}

#5. check.cluster
#-----------------
#draws the cluster in an extra plot to check
check.cluster=function(a,b){
x11()
max=max(length(a),length(b))
if(length(a)>length(b)){
c=b
b=a
a=c
}
plot(b,rep(c(10),length(b)))
points(a,rep(10,length(a)),col="red")
points(b,rep(8,length(b)))
}

#6. get.parameters
#------------------
#estimates parameters for rescaling tras to the whole range of the plot
get.parameters=function(vector,global.min,global.max){
parameters=c()
#schleife
for(i in 1:10){
local.min=min(vector)
diff=abs(global.min-local.min)
vector=vector-diff
local.max=max(vector)
factor=global.max/local.max#alles nach re
vector=vector*factor
parameters=c(parameters,diff,factor)
}
local.min=min(vector)
diff=abs(global.min-local.min)
parameters=c(parameters,diff)

return (parameters)
}

#7. rescale
#----------
rescale=function(vector,parameters){
i=1
while(i<=20){
vector=vector-parameters[i]
vector=vector*parameters[i+1]
i=i+2
}
vector=vector-parameters[21]
return(vector)
}

#8. plot.symbols
#---------------
#pos = y-achse,dir 0 nach oben, 1 nach unten
plot.symbols=function(vector,symbol.vector,color,pos,dir){
```

```
affyids=names(vector)
symbols=symbol.vector[affyids]
text(vector,pos,symbols,srt=90,col=color,cex=0.65,adj=dir)
}

#9. relax
#--------
#stretches the genesymbol entrys to the side
relax=function(vector,threshold){
abs.dist=(max(vector,na.rm=T)-min(vector,na.rm=T))/length(vector)
max=max(vector,na.rm=T)
dist=min(abs.dist,threshold)
vector=sort(vector)
#durchlaufe den vector
if(length(vector)>2){
for(i in 1:(length(vector)-2)){
#berechnet distanz
act.dist=vector[i+1]-vector[i]
if(act.dist<dist){
a=vector[i]+dist
#kleiner als das letzte element
if(a<max){
vector[i+1]=vector[i]+dist
}else{
vector[i+1]=max
}
}
}
}
return(vector)
}

#10. get.homolog
#---------------
#put in human,get mouse affyid
get.homolog=function(affyid){
homologs=read.csv(file="/home/dinkelac/data/mouse/tables/tra.mouse.human.homologs.csv")
line=which(homologs[,7]==affyid)
homolog=as.character(homologs[line,2])
if(length(homolog)==0){
return (NA)
}else{
return (homolog)
}
}

#11. connect.homologs
#--------------------
#verbindet die homologen TRAs in mouse und human cluster
connect.homologs=function(human.genes,mouse.genes,pos1,pos2,color){
#affyIDs1
human.affyIDs=names(human.genes)
mouse.affyIDs=names(mouse.genes)
no=0
#berechne anzahl an gefundenen
for(i in 1:length(human.affyIDs)){
human.affyID=human.affyIDs[i]
mouse.affyID=get.homolog(human.affyID)[1]
if(!is.na(mouse.affyID)&is.element(mouse.affyID,mouse.affyIDs)){
#draw line
human.pos=human.genes[human.affyID]
mouse.pos=mouse.genes[mouse.affyID]
```

```
lines(c(human.pos,mouse.pos),c(pos1,pos2),type="l",col=color)
no=no+1
}
}
return(no)
}

#12. get.symbols
#---------------
#holt zu affyids, die Gensymbole
get.symbols=function(species,all.startsides){

if(species=="mouse"){
mouse.table=read.csv(file="/home/dinkelac/data/mouse/tables/gnf1m.annotiert.2007.csv",
sep="\t")

mouse.affys=names(all.startsides)
rows=vector(len=length(mouse.affys))
for(i in 1:length(mouse.affys)){
rows[i]=grep(mouse.affys[i],mouse.table[,1])
}
mouse.symbols=as.character(mouse.table[rows,2])
names(mouse.symbols)=mouse.affys

#names
return(mouse.symbols)
}#end mouse
}

#print.two.clusters.R
#-----------------------
#This function displays gene clusters
#input:dataframe x with
#x[,3]=Genesymbol,x[,4]=chromosome,folgende spalten sind startside, clustersize,
#clusternr, tissue

print.two.clusters=function(x,col){
#chromosomen und startside von allen genen
setwd("/home/dinkelac/data/human/")
chrhash=read.table("chrhash.txt",sep="\t")
#----------------------

#genesymbols
symbols=as.character(x[,3])
names(symbols)=as.character(x[,1])
startsides=as.character(x[,5])
names(startsides)=as.character(x[,1])
chromosome=x[1,4]
clusternr=x[1,7]
clustersize=x[1,6]

#all genes on this chromosome
#index
all.index=which(chrhash[,2]==as.character(chromosome))
#affyid,chr,startside
d=chrhash[all.index,]

#start inner functions
#---------------------
#1 get.first

get.first=function(x){
```

```
y=vector("list",length(x))
for(i in 1:length(x)){
y[[i]]=x[[i]][1]
y=unlist(y)
}
return(y)
}
#2 basisplot
basisplot=function(min,max,pos){
y=c(10,11)
mmi=c(1.5,1,1,1)
par(mai=mmi)
plot(y,xlim=c(min,max),ylim=c(-15,15),axes=F,xlab="startside (bp)",ylab="")
#lines(c(min,min),c(-1,1),type="l")
abline(h=pos)
#box()
}
#3 gene
gene=function(x,farbe,pos,t1,textpos){
descript=symbols[names(x)]
y=x
x=c(x,x)
x=sort(x)
i=1
while(i <=length(x)){
lines(c(x[i],x[i+1]),pos,type="l",col=farbe)
#startside

#text(x,-2,x,srt=90,col=farbe,cex=0.8,adj=1)
text(y,pos[1]+(t1),descript,srt=90,col=farbe,cex=0.7,adj=textpos)
i=i+2

#text(x[1],pos[2],paste("Chromosome",chromosome,"Cluster",clusternr,
#"(",clustersize,"Genes)"))
}
}
#4 other genes
othergenes=function(x,farbe,pos){
#description=symbols
pos
y=x
x=c(x,x)
x=sort(x)
i=1
while(i <=length(x)){
lines(c(x[i],x[i+1]),pos,type="l",col=farbe)
#startside
#text(x,-2,x,srt=90,col=farbe,cex=0.8,adj=1)
#text(y,2,description,srt=90,col=farbe,cex=0.8,adj=0)
i=i+2
}
}
#draw connecting lines
#---------------------
connect=function(genes1,genes2,pos){
#nimm gensymbole von genes1
a=symbols[names(genes1)]
b=symbols[names(genes2)]
#vergleiche die gleichen mit genes2
common.genes=intersect(a,b)
c=a%in%common.genes
d=b%in%common.genes
```

```
#get positions
pos1=abs(as.numeric(startsides[names(a[c])]))
pos2=abs(as.numeric(startsides[names(b[d])]))
#draw lines
for(i in 1:length(common.genes)){
lines(c(pos1[i],pos2[i]),c(pos[1],pos[2]),type="l")
}
}

#end inner functions
#--------------------
startside=abs(as.numeric(get.first(strsplit(startsides,"/"))))
names(startside)=names(startsides)

min=min(abs(startside))
max=max(abs(startside))

#alle mit startside zwischen min,max
e=d[,3]>min&d[,3]<max#which genes are in range T,F
f=d[e,]#genes in range
g=f[,3]#startsides in range

basisplot(min,max,c(-7,7))

othergenes(g,"black",c(-8,-6))
othergenes(g,"black",c(8,6))
#raus!
farbe="red"
col="red"

startside1=startside[c(1,3,4,10,11,23,32,35,36)]

gene(startside,col,c(-8,-6),-1,1)
gene(startside1,col,c(8,6),+1,0)
connect(startside,startside1,c(-6,6))
}

#usage:print.cluster(x,"black")

#basisplot(min,max)
#gene(startside,"black")

#stats.R
#-------

#z-score
#--------
#calculate z-score (Abstand eines Wertes vom Mittelwert)
#z=x-mue/sigma
z.score = function(X) ( X - apply(X,1,mean,na.rm=T) ) / apply(X,1,sd,na.rm=T)

#venn6dim
#--------
#This function plots a Venn Diagram in six dimensions
#
#usage:

#load function:
#--------------
#source("/path/venn6dim.R")

#open window (keep size and color :)
```

```
#-----------
#x11(h=7, w=7)
#
#call function with:
#-------------------
#venn6dim(elements1,elements2,elements3,elements4,elements5,elements6,
#c("name1","name2","name3","name4","name5","name6"),0)
#put 1 for line and 0 for no line
#
#setwd("./plots")
#transparency is not supported by dev.copy2eps
#
#save plot with:
#---------------
#dev.copy2pdf(file="name.pdf")
#

venn6dim=function(a,b,c,d,e,f,names,linetype){

#plot circles
#------------
#invisible circle
symbols(0,0,circles=4,bg=rgb(1,1,1,alpha=0),lty=0,inches=F,ylim=c(-5,5),xlim=c(-5,5),
xlab="",ylab="",axes=F)

#ersten drei
#f
symbols(-0.85,0.5,circles=3,bg=rgb(0.53,0.8,0.98,alpha=0.1),lty=linetype,add=T,inches=F)
#b
symbols(0.85,0.5,circles=3,bg=rgb(0.42,0.65,0.8,alpha=0.1),lty=linetype,add=T,inches=F)
#d
symbols(0,-1.1,circles=3,bg=rgb(0.12,0.56,0.8,alpha=0.1),lty=linetype,add=T,inches=F)

#sechs
#a
symbols(0,1.1,circles=3,bg=rgb(0,0.5,1,alpha=0.1),lty=linetype,add=T,inches=F)
#c
symbols(0.8,-0.6,circles=3,bg=rgb(0.09,0.45,0.8,alpha=0.1),lty=linetype,add=T,inches=F)
#e
symbols(-0.8,-0.6,circles=3,bg=rgb(0.28,0.46,1,alpha=0.1),lty=linetype,add=T,inches=F)

#beschriften
text(0,4.6,names[1])
text(3.7,2.5,names[2])
text(3.7,-2.5,names[3])
text(0,-4.6,names[4])
text(-3.7,-2.5,names[5])
text(-3.7,2.5,names[6])

#calculate numbers

ab=intersect(a,b)
af=intersect(a,f)
ac=intersect(a,c)
bc=intersect(b,c)
cd=intersect(c,d)
ef=intersect(e,f)
de=intersect(d,e)
df=intersect(d,f)

abc=intersect(ab,c)
abf=intersect(ab,f)
```

```
aef=intersect(a,ef)
bcd=intersect(bc,d)
cde=intersect(cd,e)
def=intersect(d,ef)


abcd=intersect(abc,d)
abcf=intersect(abc,f)
abef=intersect(ab,ef)
adef=intersect(a,def)
bcde=intersect(bc,de)
cdef=intersect(c,def)

abcde=intersect(abcd,e)
abcdf=intersect(abc,df)
abcef=intersect(abc,ef)
abdef=intersect(ab,def)
acdef=intersect(ac,def)
bcdef=intersect(bc,def)

abcdef=intersect(abcde,f)

x0=abcdef

q1=setdiff(a,bcdef)
q2=setdiff(b,acdef)
q3=setdiff(c,abdef)
q4=setdiff(d,abcef)
q5=setdiff(e,abcdf)
q6=setdiff(f,abcde)

p1=setdiff(ab,cdef)
p2=setdiff(bc,adef)
p3=setdiff(cd,abef)
p4=setdiff(de,abcf)
p5=setdiff(ef,abcd)
p6=setdiff(af,bcde)

z1=setdiff(abf,cde)
z2=setdiff(abc,def)
z3=setdiff(bcd,aef)
z4=setdiff(cde,abf)
z5=setdiff(def,abc)
z6=setdiff(aef,bcd)

y1=setdiff(abcf,de)
y2=setdiff(abcd,ef)
y3=setdiff(bcde,af)
y4=setdiff(cdef,ab)
y5=setdiff(adef,bc)
y6=setdiff(abef,cd)

x1=setdiff(abcef,d)
x2=setdiff(abcdf,e)
x3=setdiff(abcde,f)
x4=setdiff(bcdef,a)
x5=setdiff(acdef,b)
x6=setdiff(abdef,c)

#1. inner circle
text(0,0,length(x0))
```

```
text(0,2.4,length(x1))
text(1.9,1.2,length(x2))
text(1.9,-1.2,length(x3))
text(0,-2.4,length(x4))
text(-1.9,-1.2,length(x5))
text(-1.9,1.2,length(x6))

#2.inner circle

text(1.2,2.3,length(y1))
text(2.5,0,length(y2))
text(1.2,-2.3,length(y3))
text(-1.2,-2.3,length(y4))
text(-2.5,0,length(y5))
text(-1.2,2.3,length(y6))

#3. circle

text(0,3.1,length(z1))
text(2.5,1.5,length(z2))
text(2.5,-1.5,length(z3))
text(0,-3.1,length(z4))
text(-2.5,-1.5,length(z5))
text(-2.5,1.5,length(z6))

#4. circle
text(1.7,3,length(p1))
text(3.3,0,length(p2))
text(1.6,-3.1,length(p3))
text(-1.7,-3.2,length(p4))
text(-3.3,0,length(p5))
text(-1.7,3.1,length(p6))

#5.circle
text(0,4.1,length(q1))
text(3.2,1.9,length(q2))
text(3.2,-1.9,length(q3))
text(0,-4.1,length(q4))
text(-3.2,-1.9,length(q5))
text(-3.2,1.9,length(q6))

}#end of venn6dim
```