Dissertation
submitted to the
Combined Faculties of Natural Sciences and Mathematics
of the Ruperto Carola University Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

Presented by

Master of Science Lucas Christopher Schütz
born in: Heilbronn-Neckargartach
Oral examination: 14.09.2018

Analyzing the role of transient folds during axis elongation of the Drosophila embryo

Referees:

Prof. Dr. Joachim Wittbrodt

Prof. Dr. Steffen Lemke

I herewith declare, that I wrote the PhD thesis "Analyzing the role of transient folds during axis elongation of the Drosophila embryo" independently under supervision, and that I used no other sources and aids than those indicated throughout the thesis.

Date	Signature

Abstract

The body plan of every higher animal is formed in a series of tissue folding and migration events called gastrulation. At the end of gastrulation, the three tissue layers ectoderm, mesoderm and endoderm are defined. In addition to the formation of these tissue layers, the embryo of the fruit fly *Drosophila melanogaster* extends during gastrulation along its head-to-tail axis. During this process of germband extension, the embryo doubles the germband in size and folds it onto itself. Germband extension coincides at the future head and tail by invagination of the gut and is accompanied by the formation of transient folds in the head (cephalic furrow) and back (dorsal folds) of the embryo. Some of these events, like the formation of the posterior gut, provide support for germband extension. Other events such as invagination and rotation of the cephalic furrow, or anterior and posterior invagination of the dorsal folds coincide with germband extension dynamics in a spatial and temporal manner, but their function or relationship with respect to germband extension remains unknown. In my thesis I established tools for the qualitative and quantitative analysis of folds and the epithelium of the Drosophila embryo. These tools allowed me to address the putative function of these transient folds by quantitative analyses of whole mount time lapse recordings in the developing Drosophila embryo, giving me a new global picture of motion dynamics in the Drosophila embryo.

Zusammenfassung

Der Bauplan des Körpers eines jeden höheren Tieres wird durch multiple Faltungs- und Bewegungsprozesse des Gewebes definiert. Die Gesamtheit dieser Prozesse wird als Gastrulation bezeichnet und resultiert in der Anlage der drei Keimblätter Ektoderm, Mesoderm und Endoderm. Zusätzlich zur Anlage dieser drei Keimblätter, dehnt sich die Kopf-Schwanzachse des *Drosophila melanogaster* Embryos während der Gastrulation aus. Während dieses Prozesses, genannt Keimstreifverlängerung, verdoppelt der Embryo die Länge des Keimstreifs und faltet ihn auf sich selbst. Die Keimstreifverlängerung fällt mit der Invagination des Darms am Kopf und Schwanz vom Embryo zusammen, und wird weiterhin von der Invagination der transienten Kopffurche (Cephale Furche) und der beiden transienten Dorsalen Furchen begleitet. Während ein Teil dieser Prozesse, wie die Invagination des Posterioren Darms, die Keimstreifverlängerung unterstützen, fallen andere Prozesse, wie die Invagination und Rotation der Cephalen Furche, oder die Invagination der anterioren und posterioren Dorsalen Furchen zwar zeitlich und räumlich mit der Keimstreifverlängerung zusammen, allerdings ist ihre Funktion in Bezug auf die Keimstreifverlängerung bisher unbekannt. In meiner Dissertation habe ich Methoden zur qualitativen und quantitativen Analyse von Furchen und des Epithels im Drosophila Embryo entwickelt. Diese Methoden ermöglichten, mir potentielle Funktionen der transienten Furchen durch eine quantitative Analyse von *in toto* Langzeitaufnahmen, des sich entwickelnden Drosophila Embryos zu untersuchen, was mir erlaubte ein komplett neues Bild der Bewegungssdynamik im Drosophila Embryo zu generieren.

Acknowledgements

Ich möchte mich zuallererst bei meinem Chef Steffen Lemke bedanken, in dessen Labor ich über 5 tolle und lehrreiche Jahre verbringen durfte. Wir haben zusammen immer die verrücktesten Ideen zu den Funktionen der transienten Furchen gesponnen, und uns danach noch abenteuerlichere Analysemethoden ausgedacht. Zu diesen abenteuerlichen Ideen mag in Einzelfällen unter Umständen auch die gute Atmosphäre bei einer der Feiern im Labor beigetragen haben. Bei unserer Ideenentwicklung haben wir uns immer an Hermann Hesse gehalten: "Mann muss das Unmögliche versuchen, um das Mögliche zu erreichen". Der Spaß den ich dabei hatte und ich glaube auch du Steffen hattest, war unbezahlbar. Auch neben der Wissenschaft hattest du immer ein offenes Ohr für alle Belange, egal ob Fotographie, Urlaub oder sonstiges. Du standest nicht nur als Chef, sondern immer auch als Mentor an meiner Seite. Tausend Dank hierfür!
Lieber Jochen, auch dir vielen Dank für alles, insbesondere dafür, dass du mich an Steffen vermittelt hast und trotzdem immer an meiner Seite geblieben bist! Deine Begeisterung für die Wissenschaft, die du nebenbei auch noch großzügig geteilt hast, war immer ein super Ansporn. Du hast mir immer das Gefühl gegeben, dass du hinter mir und meinem Projekt stehst. Eine bessere Arbeitsatmosphäre als im 5. Stock im COS kann es kaum geben. Vielen Dank!
Ich möchte mich vor allem auch bei meiner Familie bedanken für die seelische und moralische Unterstützung während meiner Promotion, aber auch für die finanzielle Unterstützung während meines Studiums. Ihr habt mich immer unterstützt, egal ob ich von meinen Ergebnissen begeistert oder enttäuscht war. Danke auch für die

vielen tollen Urlaube. Weiterhin möchte ich euch dafür danken, dass ihr immer ein offenes Ohr gehabt und nette Worte gefunden habt, wenn mich mein Projekt oder andere Umstände mal gehörig genervt haben.
Danke auch an meine Großeltern, die während meiner Promotion immer mein leibliches und oft auch finanzielles Wohl im Blick hatten.
Ganz lieben Dank auch an Freddy Frischknecht für deine Unterstützung, auch nachdem ich dein Labor verlassen habe. Vielen Dank für dein Engagement in meinem TAC, deine Meinung war mir immer sehr wertvoll. PS: Den Kasten Bier für die Abwerbung hast du hoffentlich mittlerweile von Steffen erhalten.
Liebe Heike, vielen Dank, dass du mein Projekt über die Jahre begleitet hast. Der Input von dir war immer sehr gut und hilfreich. Ich bedauere es wirklich, dass ich es nicht geschafft habe dich als Prüferin zu zulassen.
Guido, vielen Dank, dass du dich entschieden hast als Prüfer für meine Disputation zur Verfügung zu stehen.
Ganz herzlich möchte ich mich auch bei meinen aktuellen und ehemaligen Kollegen Silvi, Maike, Francesca, Naima, Viola, Ever, Paula, Atalay, bedanken. Dank euch ist unser Labor immer noch eines der Labore mit der besten Arbeitsatmosphäre am gesamten COS. Silvi und Francesca, ihr habt mich bei meiner Laborarbeit, insbesondere am Anfang sehr unterstützt, und alle meine (zum Teil auch blöden) Fragen beantwortet. Die Labmeetings mit euch waren nicht immer leicht, aber ihr habt einen maßgeblichen Anteil daran, dass aus diesem Projekt am Ende eine Dissertation geworden ist. Vielen Dank hierfür! Maike, vielen Dank für deine Arbeit, mit der du das Labor am laufen hältst und mir meine Arbeit deutlich einfacher gemacht hast. Naima und Viola, euch möchte ich vor allem für die vielen wissenschaftlichen Diskussionen in den, aber vor allem auch unabhängig von den Labmeetings danken. Die lunch und coffee breaks mit euch waren immer witzig und wir konnten über alles reden. Tausend Dank!
Ever und Paula, vielen Lieben Dank für die tolle Zeit im Büro der Dark side (come to the dark side, we have ice cream). Ihr habt mich vor allem bei der Datenvisualisierung unterstützt, oder mich gebremst wenn ich das Rad mal wieder neu erfinden wollte. @Ever, danke für dein Mitgefühl bei Layer 8 Problemen. ;-)

Ich möchte mich auch bei den Wittbrodts bedanken, die ein einzigartiges Arbeitsklima im 5. Stock vom COS geschaffen haben, es war eine Freude mit euch zusammen zuarbeiten.
Mein Dank gilt auch Lars Hufnagel, der mir die Möglichkeit gegeben hat mit Dimitri Kromm zu kollaborieren, und mir zusätzlich Daten für meine Analysen zur Verfügung stellt hat.
Dear Yu-Chiun, thank you very much for the good collaboration, the fly lines you send us, and especially for your expertise and support for my project. I appreciate it.
Ich möchte mich auch bei ein paar meiner Freunde und Freundinnen bedanken. Vor allem bei Madeline, Emilia, Ann-Kathrin und Michelle, mit denen ich über alles reden konnte, egal ob es verrückte Urlaubsabenteuer, oder Phasen der Ernüchterung während der Promotion waren; und bei denen ich mich auch mal richtig auskotzen durfte.
Dimitri, Physiker meines Vertrauens, wir hatten nicht nur eine tolle Kollaboration, sondern wir haben immer noch eine sehr schöne Freundschaft, vielen Dank dafür.
Robin, Christian, Ann-Kathrin und Nathanael, wir haben eine wirklich schöne Freundschaft und haben im Studium und in der Promotion oder in den Alpen tolle Erlebnisse gehabt. Danke für diese Treue.
También durante el PhD hay que disfrutar la vida a bocanadas, como quien dice: "work hard, party harder". Zaida y Andrea, muchas gracias por sus fiestas, pero más aun por todas las noches en que disfrutamos juntos de comida y buen vino.
Thanks Victor for reading my thesis, your comments were very valuable. However, your cooking skills are even better, thanks for the great food!

Lieber Herr Brosowski, Sie haben mich inspiriert, vor allem durch Ihre Begeisterung für die Wissenschaft. Ich fand es immer faszinierend wie Sie diese Begeisterung zu teilen vermochten. Was mich aber richtig fasziniert, hat, waren Ihre Vorhersagen die sich, obwohl Sie den Fliegenembryo eigentlich gar nicht kannten, meistens als zutreffend herausgestellt haben.
Liebe Frau Ginkel, auch bei Ihnen möchte ich mich bedanken. Ich hatte das Glück bei Ihnen viel lernen zu dürfen. Aber was noch viel wichtiger ist, Sie haben mir ger-

aten Physik beizubehalten, damit haben sie einen maßgeblichen Anteil am Erfolg dieser Arbeit. Vielen Dank!

Weiterhin möchte ich mich bei Karin Schumacher, Ingrid Lohmann und Alexis Maizel, und ihren jeweiligen Leuten bedanken, mit denen wir uns bei unserer Erkundungstour durch die Stockwerke unseres Gebäudes die Räumlichkeiten und Ressourcen geteilt haben.

Finally I also want to thank my super heroes Storm and Wolverine my personal favorites, but also The Punisher, Hulk, The Thing, Flash, Mystique, Eric and Charles. You worked day and night for me. Of course some of you required sometimes intensive care and regular SPA sessions, but we always had a great relationship ;-)

Last but not least, thanks to all my critical ice cream and liquor sommeliers (many thanks to the very critical one from Tuscany), who permanently worked on improving my ice cream and liquor making skills.

Contents

Acronyms

A area. 36

AMG anterior midgut. 3, 7, 8, 15, 16, 22, 34, 36

AP anterior-posterior. 3, 5, 17, 20, 34–37, 40, 43, 62, 73, 84, 85, 87

AS amnioserosa. 8

btd *buttonhead*. 5, 63

C. riparius *Chironomus riparius*. 10, 14, 55, 65

CCD circumscribed circle diameter. 36, 37, 43

CF cephalic furrow. 2–6, 8, 14–16, 20–22, 26, 34–37, 39, 40, 42, 43, 46, 47, 49, 55, 61–65, 69, 73, 82, 83, 87, 113

cta *concertina*. 7

D. melanogaster *Drosophila melanogaster*. 1, 2, 8, 10, 15, 16, 39, 40, 47, 49, 51–55, 62, 63, 65, 69, 91

D. pseudoobscura *Drosophila pseudoobscura*. 17, 51, 52, 54

DF dorsal fold. 3, 6, 8, 14, 17, 22, 46, 47, 49, 52, 55, 62–65, 69, 73, 82, 83, 113

DV dorsal-ventral. 17, 20, 35–37, 42, 43, 84, 85

EL egg length (0% = anterior pole). 2, 6, 20, 34, 46

eve *even-skipped*. 5, 63

fog *folded gastrulation*. 7

GBE germband extension. 3–8, 10, 14–16, 20–23, 34–37, 39, 40, 42, 46, 49, 52, 60, 63–65, 84, 87, 117

LUT lookup table. 85

NaN not a number. 69

OF Optical Flow. 17, 85

PI Propidium Iodide. 21, 78, 80

PIV Particle Image Velocimetry. 17, 21, 22, 27, 57, 58, 120, 125

PMG posterior midgut. 3, 5–8, 15, 16, 22, 35, 36, 60, 64, 83, 85

RPM rounds per minute. 78

RT room temperature. 78

SPIM Selective plane illumination microscopy. 16, 40, 63, 65, 69, 71, 91

VF ventral furrow. 2–4, 7, 10, 16, 20, 22, 23, 26, 27, 35, 36, 61

VP ventral plate. 4

zen *zerknüllt*. 8

1 Introduction

Developmental biology is one of the most fascinating disciplines in biology. Whole complex organisms consisting of thousands or millions of cells originate from a single zygote [Gilbert and Barresi, 2016]. In order to develop such a complex organism, cells have to proliferate, but even more important they have to be organized. The process during which cells organize themselves in the 3 distinct tissue layers ectoderm, mesoderm and endoderm is called gastrulation [Gilbert and Barresi, 2016]. During gastrulation, the principles of development in biology can be observed on very short time scales as the underlying genotype is rapidly transformed into a phenotype. This rapid transformation involves large scale cell movements, maternal and somatic gene expression and a high level of conservation among all animal species [Gilbert and Barresi, 2016]. In the last century, *Drosophila melanogaster* (*D. melanogaster*) has become one of the main model organisms to study gastrulation and other processes during development. The advantages of *D. melanogaster* are, that it is easy and cheap to keep, it has short generation times with a high number of offspring and imaging of the embryos is fairly simple [Jennings, 2011]. In addition, because of its limited size,*D. melanogaster* allows for *in toto* imaging of gastrulation, opening up new possibilities of understanding the complexity of cell and tissue migration during gastrulation [Tomer et al., 2012].

1.1 Drosophila development

1.1.1 Pre-gastrulation development

The development of *D. melanogaster* starts with the fertilized egg cell which undergoes 13 cycles of nuclear division. In contrast to other animals, these division cycles do not include cell division, leading to a syncytium where all nuclei share one cell [Rabinowitz, 1941, Zalokar and Erk, 1976, Foe and Alberts, 1983]. These division cycles are synchronized among all nuclei, and follow a strict timing. Cycles 1 - 9 take around 8 minutes per cycle to complete. For the next nuclear division cycles, the duration increases, as cycle, 10, 11, 12 and 13 take on average 9, 10, 12 and 21 minutes to complete [Foe and Alberts, 1983]. The 14th cycle is asynchronous and takes between 75 and 175 minutes depending on the location of the cells [Foe, 1989]. After cycle 8, the majority of nuclei start to migrate towards the periphery [Zalokar and Erk, 1976, Foe and Alberts, 1983]. Some nuclei which will later become the yolk nuclei remain at their position in the yolk. Around stage 10 all of the migrating nuclei arrive in the periphery of the embryo. In addition, the pole cells are formed at stage 10 [Zalokar and Erk, 1976, Foe and Alberts, 1983]. During stage 14, when the 14th division cycle takes place, the nuclei elongate and cellularization occurs as plasma membrane extends from the outer surface in between the nuclei, producing individual compartments [Foe and Alberts, 1983, Warn and Robert-Nicoud, 1990, Schweisguth et al., 1991].

1.1.2 Gastrulation

A few minutes after cellularization is completed, gastrulation starts. At the beginning of gastrulation two folds invaginate almost at the same time, namely the ventral furrow (VF) and the cephalic furrow (CF) [Poulson, 1950, Sonnenblick, 1950, Turner and Mahowald, 1977]. The CF is a circumferential fold located at 33% egg length (0% = anterior pole) (EL), while the VF is a linear fold forming between 14% and 94% EL [Turner and Mahowald, 1977, Campos-Ortega and Hartenstein, 1997, Ingham, 1994]. While the cells invaginated by the VF later on

in development contribute to the mesoderm [Leptin and Grunewald, 1990, Leptin et al., 1992, Thisse et al., 1988], the cells incorporated in the CF will return back to the surface of the embryo in later development, and remain still part of the ectoderm [Turner and Mahowald, 1977, Foe, 1989, Campos-Ortega and Hartenstein, 1997]. In contrast to the CF, which continues invaginating even during germband extension (GBE), the VF invagination is quite fast [Underwood et al., 1980, Campos-Ortega and Hartenstein, 1997]. Soon after the VF closes, and the now mesodermal cells in the lumen of the embryo form a tube, the development of the embryo continues with GBE [Ingham, 1994]. During GBE ventral and lateral cells move in a continuous flow towards the ventral midline, where they intercalate and form an elongation in the anterior-posterior (AP)-direction [Irvine and Wieschaus, 1994, Butler et al., 2009, Collinet et al., 2015, Kong et al., 2017]. This elongation extends around the posterior pole and continues on the dorsal side all the way to the cephalic furrow [Ingham, 1994, Campos-Ortega and Hartenstein, 1997]. In addition to the cell driven elongation, the posterior midgut (PMG) invagination, which takes place at the same time, exerts an external pulling force on the extending germband [Collinet et al., 2015, Kong et al., 2017]. In the head region of the embryo another process takes place around the same time. The anterior midgut (AMG) invaginates on the ventral side of the embryo, just anterior to the CF. The cells in both of these midgut invaginations will later contribute to the endoderm [Poulson, 1950, Turner and Mahowald, 1977, Foe, 1989, Desprat et al., 2008]. If this simultaneously-running processes were not enough, the extraembryonic tissue known as the amnioserosa extends from its dorsally located domain towards the lateral sides, and stretches between the CF and the posterior dorsal fold (DF) [Turner and Mahowald, 1977, Lacy and Hutson, 2016]. After GBE is completed, around 3h after the onset of gastrulation, there is a break, afterwards the germband retracts and the embryo undergoes dorsal closure, followed by segmentation and theformation of the inner organs [Goldman-Levi et al., 1996, Campos-Ortega and Hartenstein, 1997, Jacinto et al., 2002, Schöck and Perrimon, 2002, Heisenberg, 2009]. In the following section I will go more into detail about the processes which take place during early gastrulation and might be im-

portant for understanding the context of the transient folds during GBE.

Ventral furrow invagination

Ventral furrow invagination starts with the apical flattening of a domain of 18 cells in width and around 60 cells in length. These cells along the so called ventral plate (VP) will eventually form the VF [Sweeton et al., 1991]. As their apical end becomes flat, the cells of the VP shorten in length [Leptin and Grunewald, 1990, Leptin et al., 1992]. Concurrently with this shortening, the nuclei of these cells shift slightly towards the basal end [Ingham, 1994]. Together with the apical basal shortening, the apices of the cells constrict, and the cells undergo a shape transition from columnar, through trapezoidal, to a final triangular shape [Sweeton et al., 1991, Kam et al., 1991, Parks and Wieschaus, 1991, Martin et al., 2008]. The triangular shape of the cells with the tip of the triangle at their apices, and the base of the triangle at their basal side, induces a curvature which is in opposition to the ellipsoidal curvature of the egg. The result is the formation of the ventral furrow [Turner and Mahowald, 1977, Sweeton et al., 1991, Ingham, 1994, He et al., 2014]. The whole process of VF formation is very fast, and is completed in 8-11 minutes [Sweeton et al., 1991].

The formation of the cephalic furrow

Same time as the onset of VF invagination, the CF starts to invaginate [Turner and Mahowald, 1977, Ingham, 1994, Vincent et al., 1997]. The mechanism of CF invagination is different to VF invagination; it starts with the shortening of a single row of cells in the lateral regions of the embryo [Vincent et al., 1997, Spencer et al., 2015]. As these cells remain attached to the yolk sac membrane during this process, their apices are pulled inwards, creatting a small indentation on the surface of the embryo [Turner and Mahowald, 1977]. As the shortening continues, the initially invaginating cells drag more cells along allowing the CF to grow in size and extension. The CF continues to grow further during GBE [Turner and Mahowald, 1977, Ingham, 1994, Campos-Ortega and Hartenstein, 1997]. After GBE the CF unfolds, bringing all cells back to the surface of the embryo, leaving no trace

behind [Turner and Mahowald, 1977]. The position of the CF is specified by an overlap of the transcription factors *buttonhead* (*btd*) and *even-skipped* (*eve*); a lack of either of those two proteins results in a lack of the CF [Vincent et al., 1997]. Both genes, *btd* and *eve* act relatively early in this process, and so far there have been no findings of genes involved further downstream. Interestingly, although the CF was described decades ago, its function is so far completely unknown. Besides speculation that the CF is only a simple separator of the head from the rest of the embryo, there have also been attempts to solve the function via computational modeling [Allena et al., 2010, Allena and Aubry, 2012, Dicko et al., 2017]. The last of these suggests that the function of the CF could be to break the symmetry, to allow for asymmetric GBE as it takes place in Drosophila.

Germband extension

GBE starts between 9 and 10 minutes after the onset of gastrulation [Sweeton et al., 1991]. During GBE cells located on the ventral and lateral sides of the embryo intercalate and extend in a coordinated manner in the AP direction around the posterior pole all the way towards the CF on the dorsal side of the embryo [Turner and Mahowald, 1977, Sweeton et al., 1991, Campos-Ortega and Hartenstein, 1997]. This extension is driven by two main processes, which can be considered the motors of GBE [Kong et al., 2017]. The first driving process is the cell intercalation which produces a cell autonomus force in the AP direction [Bertet et al., 2004, Zallen and Blankenship, 2008]. This intercalation takes place in 2 ways: cells can either exchange their neigbours via a T1-T3 junctional transition, or alternatively, the cells form rosettes where each group of 6-8 cells forms a joined center. This rosette is in a second step transformed into an elongation in the AP direction [Blankenship et al., 2006]. The second major driving force of GBE is the PMG invagination, which produces an external force, pulling cells in a dorsal-anterior direction [Collinet et al., 2015, Kong et al., 2017]. GBE is further subdivided, into a fast and a slow phase. The fast phase takes around 35 minutes and accounts for 84% of the total extension. After the fast phase, the germband extends in the slow phase for another 70 minutes before it comes to a complete stop at 27%

EL [Ingham, 1994, Campos-Ortega and Hartenstein, 1997].

Dorsal fold invagination

The anterior and posterior DF are formed at the onset of GBE on the dorsal surface of the embryo. The anterior DF is located at 50% egg length, and the posterior DF is located at roughly 65% egg length [Turner and Mahowald, 1977, Campos-Ortega and Hartenstein, 1997]. They quickly invaginate cells, grow in depth and symmetrically extend (left-right) in the lateral direction [Turner and Mahowald, 1977, Campos-Ortega and Hartenstein, 1997, Wang et al., 2012]. As the germband with the leading PMG invagination encounters the DFs, both DFs, first the posterior and then the anterior are dragged along towards the cephalic furrow [Turner and Mahowald, 1977]. During germband extension, the posterior DF grows significantly in depth and extension, while the anterior DF remains rather small [Turner and Mahowald, 1977, Wang et al., 2013]. Initiation of the DFs and therefore also their position is genetically regulated by a shift of Par-1 expression levels. This shift in expression level leads to a basal shift of junctions, as it changes the ratio between Bazooka, aPKC and Par-1, which together control the junction positions [Wang et al., 2012]. The shift introduces a junctional asymmetry and leads to a curved lateral membrane of the cells adjacent to the initiator cells [Wang et al., 2012]. In addition to this invagnation mechanism, the size of the DFs is controlled by the expression of the GTPase activating protein Rapgap1, which controls the activity of the small GTPase Rap1 [Chen et al., 1997, Wang et al., 2013]. Rapgap1 expression is mainly limited to the posterior DF allowing for a deeper invagination compared to the anterior DF [Wang et al., 2013]. The function of the DFs has so far remained unclear; some have even suggested that they are an artifact of germband extension and do not have a function at all [Sonnenblick, 1950]. However the fact that they are tightly genetically regulated argues that they may in fact have a function. Similar to the CF, the DFs unfold later in development and have no role in tissue specification.

Posterior Midgut Invagination

The posterior midgut starts to invaginate at the dorsal posterior end of the embryo at the onset of GBE [Poulson, 1950, Turner and Mahowald, 1977]. The tissue incorporated in this invagination will become endodermal tissue contributing to the PMG and possibly also the hindgut [Turner and Mahowald, 1977, Campos-Ortega and Hartenstein, 1997, Ingham, 1994]. In addition, germ cells of Drosophila, the so called pole cells, are incorporated in this invagination [Turner and Mahowald, 1977]. The PMG invagination has a similar underlying invagination mechanism as the VF invagination. This is first illustrated by the apical constriction and flattening of the posterior pole before the formation of a cup-shaped invagination, in which the pole cells are invaginated [Turner and Mahowald, 1977, Sweeton et al., 1991]. Second, the similarity is illustrated from a genetic point of view, as the VF and PMG invagination both require *folded gastrulation* (*fog*) and *concertina* (*cta*) [Zusman and Wieschaus, 1985, Parks and Wieschaus, 1991]. From the mechanical point of view, the PMG invagination belongs to one of the strongest embryonic processes, as its force is high enough to rip the ventral-posterior epithelium [Collinet et al., 2015]. In accordance with these results, the PMG invagination is able to handle the loss of convergent extension, one of the 2 main driving forces of GBE and complete GBE almost on its own, although at a slower pace [Wieschaus et al., 1984, Butler et al., 2009].

Anterior Midgut Invagination

The AMG invagination forms at the anterior tip of the VF [Turner and Mahowald, 1977, Campos-Ortega and Hartenstein, 1997]. The AMG invagination forms a y-shaped tip, still connected to the VF [Turner and Mahowald, 1977, Ingham, 1994]. The invaginated cells contribute to the endoderm of the embryo [Poulson, 1950]. In contrast to the VF invagination, the AMG invagination remains open and connected to the surface for longer [Turner and Mahowald, 1977]. A specified mitotic domain in which cells have a division axis perpendicular to the surface, also contributes to the AMG invagination, as one daughter cell invaginates, while the other on remains on the surface [Foe, 1989]. In addition it was shown that cells

which do not invaginate in the original AMG invagination, but remain longer on the surface, contribute to the AMG [Technau and Campos-Ortega, 1985]. Recently it was shown, that AMG invagination can be induced via mechanical manipulation induced gene expression [Desprat et al., 2008]. In contrast to the pulling function of the PMG invagination [Collinet et al., 2015], there is currently no evidence, that the AMG invagination executes a similar function.

Amnioserosa formation

The extraembryonic tissue in Drosophila, the so called amnioserosa (AS), extends during GBE from a dorsally located domain of around 200 cells in size [Schweisguth et al., 1991, Campos-Ortega and Hartenstein, 1997]. Characteristic for the AS are the polyploid, flat squamous cell spreading to seal the dorsal and lateral sides of the *D. melanogaster* embryo during GBE [Campos-Ortega and Hartenstein, 1997, Grimaldi and Engel, 2005]. The anterior and posterior limits of the AS during this process are provided by the CF and the posterior DF [Turner and Mahowald, 1977, Campos-Ortega and Hartenstein, 1997]. Analogously to germband extension, the spreading of the amnioserosa is divided into a fast phase of around 30 minutes, and a slow phase of 90 minutes [Campos-Ortega and Hartenstein, 1997] (reviewed in [Lacy and Hutson, 2016]). The spread of the AS is mediated via rotary cell elongation, which produces a cell autonomus force [Pope and Harris, 2008]. The extent of the AS is specified by the HOX-like transcription factor *zerknüllt* (*zen*) [Hughes and Kaufman, 2002, Schmidt-Ott et al., 2010].

Mitosis

While the first 13 division cycles in *D. melanogaster* are, except for the yolk nuclei and pole cells, globally synchronized, this synchrony is lost during the 14th division cycle [Zalokar and Erk, 1976, Turner and Mahowald, 1977, Foe and Alberts, 1983, Foe, 1989]. After the 14th cycle, when the embryo enters gastrulation, mitosis continues in certain areas of the embryo, called mitotic domains [Foe, 1989]. There are a total number of 25 mitotic domains, each of them having their own synchrony [Foe, 1989]. The role of these mitotic domains during gastrulation has

not been completely resolved yet. However, as new cells need space, in the already tightly packed embryo, a certain mechanical influence on morphogenetic movements during gastrulation is very likely [Foe, 1989, Momen-Roknabadi et al., 2016].

1.2 Tissue and cell migration

1.2.1 2D epithelial sheet migration

Cell migration whether in the form of single cells or as epithelial sheets, plays a major role during the development of any higher metazoan [Gilbert and Barresi, 2016]. In epithelial sheets, cells move in a coordinated fashion and are linked via junctions [Thiery et al., 1988, Takeichi, 1991, Gumbiner, 1996]. Multiple mechanical analyses and models have shown that the cells of an epithelial sheet can behave like an active fluid with low Reynolds numbers for longer timescales. In the opposite case, for short time scales, epithelia rather exhibit elastic properties [George et al., 2017, Delile et al., 2017, Blanch-Mercader et al., 2017]. The relevant time scales for the switch between elastic and viscous behavior depend on the intra- and intercellular kinetics [Blanch-Mercader et al., 2017]. The junctions required to ensure the integrity of an epithelium behave in a very similar way and show an elastic behavior for time scales < 1 minute, whereas they show a dissipative viscous behavior for time scales > 1 minute [Clément et al., 2017]. This junctional flexibility is required to allow an epithelium to act like a fluid [Curran et al., 2017].

The cells of an epithlium are usually connected to both a substrate and to their surrounding cells, allowing for the transmission of forces. Forces in a crawling epithelium are usually built up in the outer regions and are then transmitted to the inner epithelial cells where regions of force heterogenieties spanning a few cell widths are built up [George et al., 2017]. In cases where the heterogeniety is too high, the epithelium can rupture, interestingly, this can be prevented by cell proliferation [George et al., 2017].

1.2.2 Epithelial sheet migration in Drosophila

The above findings were made in epithelial sheets in cell culture. However, epithelial migration in the Drosophila embryo underlies the same physical principles and is therefore likely to behave in a very similar way. During VF invagination and GBE large epithelial sheets move along the almost ellipsoidal embryo [Ingham, 1994, Lye and Sanson, 2011, Lye et al., 2015]. As different regions of the embryo have a different 3D shape, including curvature and area (or simplified the diameter), a fluid like epithelium will change its properties, for example velocity, as modeled in 2D by [Blanch-Mercader et al., 2017]. This change of properties likely leads to tensions, and could possibly require neighbor exchanges via junctional remodulation or other ways of tension release. Furthermore, the tissue in the Drosophila embryo also makes use of junctional remodeling for active elongation during GBE or amnioserosa spreading [Collinet et al., 2015, Pope and Harris, 2008]. This raises the question of how flexible or stringent the epithelium in the Drosophila embryo actually is, and whether epithelial maintenance is area dependent.

1.3 The embryo as mechanical system

In contrast to physics which only requires mathematical principles and is otherwise independent, biology requires both physics and chemistry to be fully understandable. This consideration sheds a different light on findings like the velocity difference in the GBE of *Chironomus riparius* (*C. riparius*) and the GBE of *D. melanogaster* (Urbansky, personal communication), as there is not only the question for the evolutionary benefit, but also how this velocity difference is made possible from the physical point of view. Under the simple assumption that the velocity of GBE is roughly constant, the equation $E = \frac{1}{2}mv^2$ defines the kinetic energy E of the moving tissue, Where m is the mass and v is the velocity. Assuming the mass remains constant, a velocity increase by a factor of 3-4 leads to an increase of the kinetic energy by a factor of 9-16. How can this be handled by the tissue?

This point of view is important, as there are still structures like the transient folds whose function is unclear [Lye and Sanson, 2011]. However it only became possible to investigate these folds in a global context with the availability of microscopes which provide high spatial and temporal resolution [Krzic et al., 2012]. Combining this technical point of view, with classical biological and possibly also evolutionary perspectives might deliver new insights into hidden functions in the Drosophila embryo.

2 | Aim

The Drosophila embryo is a highly dynamic system, which develops by numerous serially as well as parallely coordinated developmental processes, comprising over 6000 cells. The complexity and coordination observed in the Drosophila embryo is presumably shaped by a long evolutionary history of gradual change. A notable outcome of constant tinkering of cell and tissue mechanical processes is a speed increase in tissue elongation by a factor of 3-4 in the last 250 million years. My thesis explores putative requirements and consequences that the Drosophila embryo faced to handle challenges associated with the innovation of faster and presumably more efficient tissue extension.

A velocity change in tissue extension by a factor of 3-4 is likely accompanied by changes at the cellular level like enhanced coordination, leading to faster and more efficient intercalation. However, it was recently shown that tissue elongation in Drosphila is not only dependent on cellular behavior, but involves global extrinsic forces [Butler et al., 2009]. This means that the changes on the cellular level were likely accompanied by more global changes in tissue morphogenesis, i.e. newly acquired abilities to bend and fold the tissue. To obtain a comprehensive picture of this time span in development I needed an *in toto* data set of the embryo combining high temporal with high spatial resolution. With the advent of light sheet microscopy, an imaging technology that allows for such recordings has been recently become available. Here, I make use of this microscopy methods to address tissue level innovations in flies that have a putative impact on the speed and efficiency of early embryonic morphogenesis. My analyses identified two structures

with high potential for putative impact on early morphogenesis. The cephalic furrow (CF) and the posterior dorsal fold (DF). These transient folds are only present during germband extension (GBE) and vanish without a trace in later development. Currently there is no function described for these folds, a reason could be that they seem to be highly dynamic and well integrated into GBE, rendering a detailed analysis with classic microscopes almost impossible. However, their location, the CF represents the dorsal and ventral extremes of GBE, while the posterior DF is located perpendicular like an obstacle in the track of the germband, and their evolutionary history *Chironomus riparius* (*C. riparius*) does not have these transient folds, it also has a slower GBE, suggest that they could be key structures for fast GBE.

3 | Results

During gastrulation almost all of the over 6000 cells of the fly embryo take part in various movements, including several larger invaginations and movements like germband extension (GBE) as well was anterior midgut (AMG), posterior midgut (PMG) and cephalic furrow (CF) invagination [Campos-Ortega and Hartenstein, 1997]. As space in the fly embryo is very limited it can be assumed that there has to be a tight coordination of these events, facilitating interaction, but limiting interference, to a minimum. Although most of the processes and movements have been studied in detail, most of the work was usually limited to single processes, as the *Drosophila melanogaster* (*D. melanogaster*) embryo is simply too large for imaging a large field of view at high temporal and spatial resolution. However, despite these limitations, there has been work showing interaction between different processes, as exemplified by the driving of GBE via the force generated by the PMG in the trunk of the embryo [Collinet et al., 2015]. My intention in this work is to go even a step further and show a global picture of movements taking place on the embryo during gastrulation and create a detailed *in toto* map of motion and timing to understand putative supporting and antagonizing events, as well as possible counter measures of the embryo against interference. I will do this using the process of fast GBE and associated processes as representative system.

3.1 The *in toto* approach

In order to analyze *in toto* motion of the Drosophila embryo an imaging method providing a full 3 dimensional view with a high spatial and temporal resolution was required. Confocal microscopy can provide a very high spatial resolution, but is generally slow, and light diffraction prevents an *in toto* view on the embryo, as the Drosophila embryo is too large. Spinning disc microscopy provides both a high temporal and spatial resolution, but it is still limited because of diffraction. The only current technology which can overcome this issue is lightsheet microscopy, in which the embryo can be rotated and imaged from multiple sides [Huisken et al., 2004, Krzic et al., 2012]. Using computer algorithms, these multiple views, usually 4, can be merged into one isotropic image which depicts the embryo from all sides in an equal quality (figure 3.1). In addition, the Selective plane illumination microscopy (SPIM) comes with a very low phototoxicity which allows for imaging of extended periods of time, such as the entire course of embryonic development of *D. melanogaster* [Krzic et al., 2012, Tomer et al., 2012]. This full 3 dimensional view also enables the investigation of processes and structures present on more than one side of the embryo, like the CF or GBE. This means that using this SPIM data I was not only able to have a closer view of cell movement, structures like invaginations, and processes like GBE, but also at the interaction between them. This enabled me to ask several questions:

- What happens to a cell once it joins an invagination, and vanishes from the surface? Do epithelial properties still apply?
- How do structures behave in contrast to the surface?
- Can the structures be grouped according to the cell behavior within them?

The last question was specifically interesting, as the invaginations in the fly embryo can be divided into 2 groups: 1. Invaginations specifying tissue as endoderm or mesoderm, like the ventral furrow (VF) invagination, as well as the AMG and PMG invagination. 2. Invaginations that do not change tissue fate, like the CF and

the anterior and posterior dorsal fold (DF)s. To answer these questions I needed to find a suitable method to analyze and visualize movement of cells (velocity $\vec{v}$) in the embryo. The ideal way would have been automated tracking. However, this method comes with a very serious problem caused by error propagation. The blastoderm fly embryo contains around 6000 cells which are tightly packed, and the automated segmentation of these cell proved to be difficult. this is principally because the imaging quality was not always at the same level, as cells located closer to the acquiring objective were imaged with higher intensities and sharper contours, then cells located further away on the surface or inside the embryo. This problem lead to a certain likelyhood of wrong segmentations. A segmentation error then leads to a tracking error in the track of a tracked cell. Another problem was raised by the temporal length of the data. I analyzed cells in a time frame of more than 60 minutes using a temporal resolution of 30s, which gave me more than 100 time points per track. Therefore using an automated tracking algorithm with a common accuracy of 97%-99% would have been insufficient, as statistically every track would have contained 1-3 errors. This is why I decided to use Optical Flow (OF), also known as Particle Image Velocimetry (PIV).

3.2 *In toto* motion

To measure acceleration and velocity of the cells in toto I used PIV and combined it with a heatmap based color visualization to display velocity and a color spectrum based visualization to visualize direction (see section 5.2.3). As the fly embryo is geometrically best represented by an ellipsoid, which can be simplified to a cylinder, I used a cylindrical coordinate system for assigning direction labels. Distances and therefore velocity were, however, still measured in a cartesian coordinate system, as in the case of this Drosophila embryo the time resolution (30 s) was high enough to allow for the vectors in the dorsal-ventral (DV) direction to become tangential. The vectors in the anterior-posterior (AP) direction correspond anyway to a cartesian axis. The analysis shown here was based on the work done in my master thesis for *Drosophila pseudoobscura* (*D. pseudoobscura*) [Schütz, 2014].

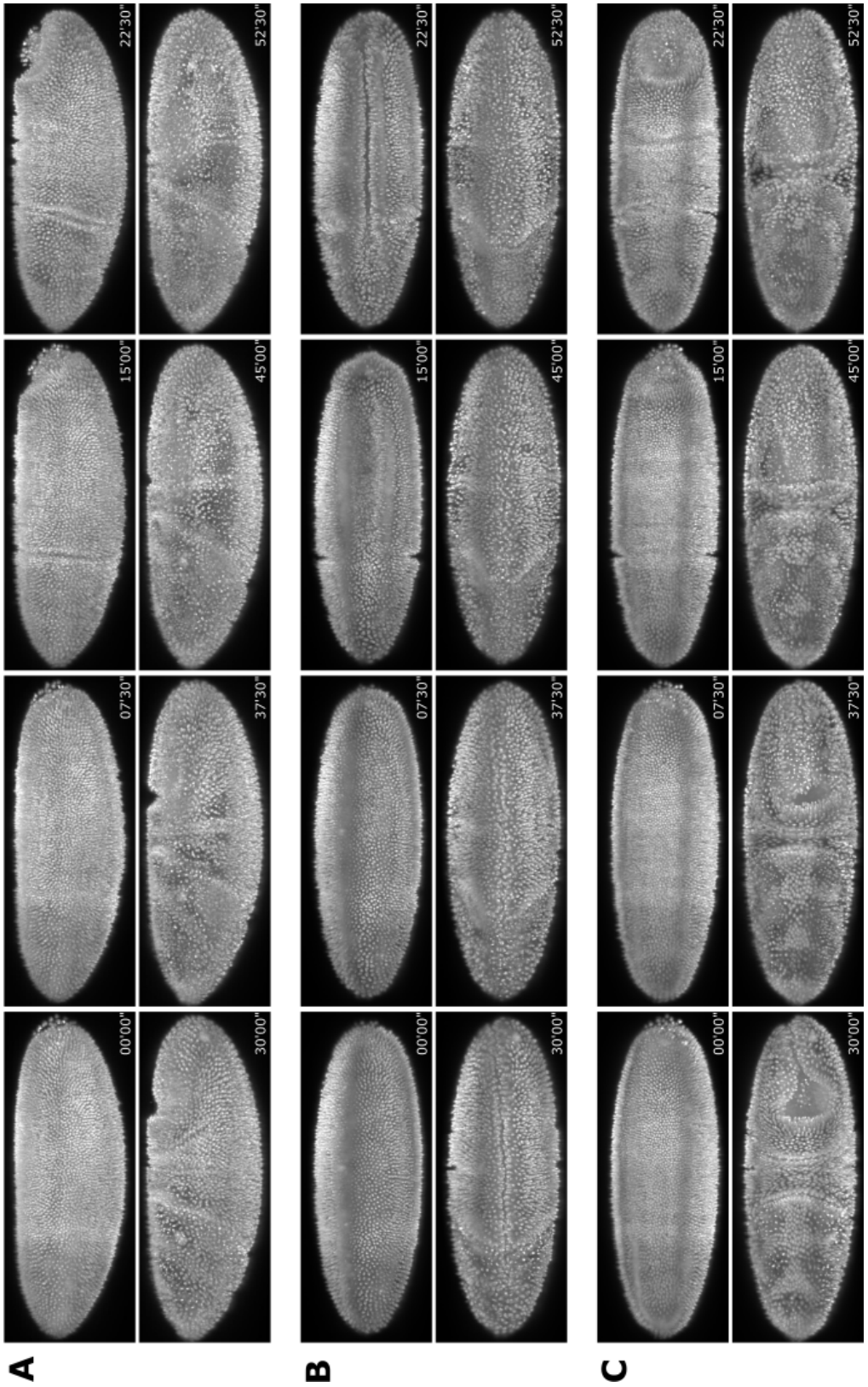

A
00'00"
07'30"
15'00"
22'30"
30'00"
37'30"
45'00"
52'30"
B
00'00"
07'30"
15'00"
22'30"
30'00"
37'30"
45'00"
52'30"
C
00'00"
07'30"
15'00"
22'30"
30'00"
37'30"
45'00"
52'30"

Figure 3.1: **Maximum intensity projections of different stages of development** Gastrulation in Drosophila is shown from the **A** lateral, **B** ventral and **C** dorsal side, as well as in an unrolled view **D**. At 0 minutes, the embryo was just before gastrulation. At 7.5 minutes, the posterior pole started to flatten and the first movements of ventral furrow invagination could be observed. At 15 minutes, the cephalic furrow was clearly visible and germband extension started with the first movements of posterior midgut invagination. At 22.5 minutes, the pole cells moved already to the dorsal side, and the anterior and posterior dorsal folds appeared. The ventral furrow was now closed and the cephalic furrow started to tilt. After 30 minutes, the germband had completed around 50% of its total track. The dorsal side was already cramped from the germband and the tilt of the cephalic furrow was clearly visible. After 37.5 minutes the dorsal folds fused on the dorsal side. At 45 minutes, the tilt of the cephalic furrow was at full extent and the germband was only a few cell sizes away from it. After 52.5 minutes, the germband reached the cephalic furrow and only the posterior dorsal fold could still be seen, while the anterior one was gone.

Prior to GBE, almost no movement was present in the embryo as illustrated by the black and purple colors of the speed visualization (figure 3.2 A), and the rather chaotic colors in the direction visualization (figure 3.2 E). The first speed changes, although they were only subtle, could be seen during the invagination of the CF and VF. However, as soon as GBE started, the cells accelerated significantly and high speeds of more than 10 $\mu m/min$ could be observed (figure 3.2 A). The highest velocities could be observed from cells located at the ventral posterior part of the embryo and on the lateral sides between 40% and 60% egg length (0% = anterior pole) (EL). To tell the direction in the speed visualization for all directions (figure 3.2 A), I did 3 further visualizations, one showing only movement in the AP direction (figure 3.2 B), one depicting movement in the posterior direction (figure 3.2 C), and the last one depicting movement in the anterior direction (figure 3.2 D). Compared to the visualization in all directions, the highlighted domains, representing fast moving cells, shrank, especially on the lateral sides in the anterior half of the embryo; i.e. the AP cell movement in the anterior half of the embryo is mainly limited to the dorsal and ventral midline (figure 3.2 A+B). In the posterior half of the embryo movement in the AP-direction was also present on the lateral sides. However, also in the posterior half there was a good share of movement in the DV-direction, which can be seen from the change to darker colors (figure 3.2 A+B). Further differences between the posterior and anterior half could be observed in the single direction AP-visualizations. There seemed to be a clear line separating the ventral development of the embryo trunk into 2 halves, even before the onset of GBE. This line was located at 60% to 65% EL, which corresponded to 50% of the line between the CF and the posterior pole (figure 3.2 C+D). During ventral furrow invagination, cells located posterior to this line moved in the anterior direction, and vice versa, cells located anterior moved in the posterior direction. At the onset of GBE, there was a switch of directions in the movement of cells on the ventral midline. Cells located posterior of the described line moved now in the posterior direction, and cells located anterior, in the anterior direction (figure 3.2 C+D). The posterior movement was faster, involved more cells and lasted longer than the anterior movement. Although the posterior movement was dominant,

the anterior movement of cells was clearly visible, which questioned the concept of GBE as an exclusively posterior directed process (figure 3.2 D). Although the main movements during gastrulation could be seen with both visualizations, direction and speed, certain questions and problems remained. The cell movement in the embryo seemed to be clustered into larger groups (figure 3.2 E), however it did not become clear how homogeneous or heterogeneous these clusters were and how their edges behaved. The resolution of the spectrum visualization might not have been good enough to answer this question, although it was already higher than the resolution of the 8 color based visualization of my master thesis. A higher resolution, using a 16- or 24-color based visualization with sharp edges could have been a solution to this problem. However, as the nuclei did not move on a linear track, but were also subject to some random motion similar to Brownian motion, their velocity vectors were averaged, which counteracted a higher resolution. Furthermore, the Propidium Iodide (PI) analysis did not show a clear differentiation between structures and surface cells. This might have been on the one hand due to the movement of this structures in context with the general cell flow, but on the other hand also due to an automated segmentation which was required by the PIV pipeline. As explained above, automated segmentations never have an ideal solution. In this particular case, the automated segmentation was configured in a way that the surface cells were recognized, but also separated, which lead to drastic losses of cells in all folds and invaginations, as their low signal was no longer considered for segmentation. The reasons listed above could explain why the flow in the head region was changing very often, and in general inconclusive. Cell movements in the head region during GBE are less described than the movements in the trunk of the embryo. A model of the gastrulating embryo even suggests almost no movements in the presence of the CF, but rather strong circular movements, similar to the ones in the tail region in absence of a CF [Dicko et al., 2017].

The above presented data gave a first impression of the motion in the embryo, but the chosen analysis was not able to fulfill the requirements for a comprehensive *in toto* picture of the embryo, mainly because of the lack of resolution and the lack in surface and structure differentiation. Especially the transient folds could only

be seen by black gaps in the visualization, due to the lack in segmented cells. This made an analysis of them in context with the cell motion impossible, leaving the questions I posed above unanswered. As a consequence a better solution for motion analysis than PIV was required, which was manual tracking.

3.3 Cell and structure tracking

To get a comprehensive picture of the embryo's whole motion dynamics, I used cell tracks, as well as relative movement of developmental landmarks. General landmarks of Drosophila embryonic development are the PMG inavagination, the VF invagination, the AMG invagination as well as the CF invagination and the anterior and posterior DF formation. While the VF, PMG and AMG invaginations are morphological events that occur in all species, the CF and the DFs are recently evolved novelties that are only present in flies with high GBE velocities [Ritter, 1889,Turner and Mahowald, 1977,Urbansky, 2016]. Furthermore, in contrast to the other invaginations, the CF and the DFs are transient folds that do not specify any tissue and disappear without any trace after GBE [Turner and Mahowald, 1977, Ingham, 1994, Campos-Ortega and Hartenstein, 1997]. Out of these three folds, the CF is the most interesting in context of GBE for several reasons: it represents the ventral and dorsal extremes of GBE, has a rather large surface area, with one edge facing the trunk, and the other edge facing the head and it rotates during the initial fast phase of GBE [Turner and Mahowald, 1977, Vincent et al., 1997]. To understand the behavior of the CF in context with GBE I tracked the following structure interest points: ventral CF, dorsal CF and GBE at the anterior edge of the PMG invagination in a lateral and dorso-ventral cross section. I combined these structure tracks with cell tracks from the ventral, the dorsal and one lateral side. I made sure that the germband extended straight, without twisting to ensure lateral symmetry in the tracked embryo.

As manual tracking software, I chose MtrackJ [Meijering et al., 2012] as I had worked with it before, where it proved to be suitable for large 3D+time data sets. There are essentially only 2 criteria for choosing manual tracking software:

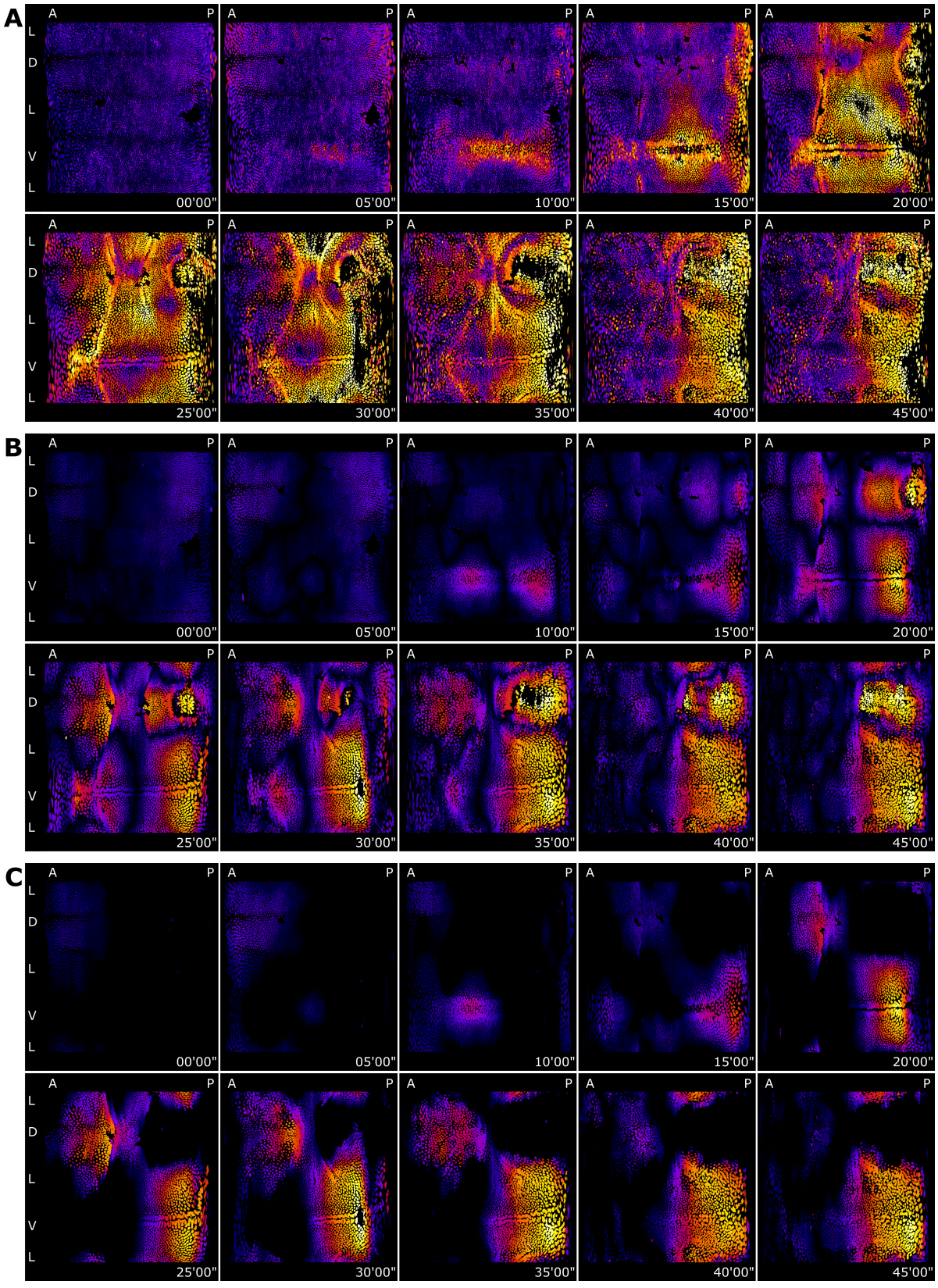
the requirement that it should run stably without wasting resources, and personal preference. Other features, such as calculated parameters, are built upon the basic tracking data obtained. I chose not to use the built-in tracking parameters, and instead I calculated them myself. This is because the included parameters are produced from a black box, you do not know and cannot check whether they are correct. I tracked around 700 cells (360 cells in embryo 1 and 340 cells in embryo 2) including cells in areas like the cephalic furrow which are hard to track as they are often just represented by subtle shade changes in a uniform grey surrounding. The second embryo served as confirmation of the results form the first one, however as the second embryo was quite motile during imaging, I was only able to confirm my results by judging the tracks, as the plotted parameters were very noisy (data not shown). To ensure a high tracking accuracy, I considered especially in areas of high cell density and fast movement along the z-axis several criteria:

- nucleus shape
- nucleus size
- nucleus intensity
- velocity vector of the tracked nucleus vs general tissue velocity vector
- previous velocity vector
- neighboring cells

I started tracking just at the time point where the first gastrulating movements of cells invaginating into the VF could be observed (figure 3.2 5 minutes). I continued tracking for 65 minutes, at which point GBE was already significantly slower than at the beginning. The tracks from the different sides of the embryo are illustrated in figure 3.3.

3.3.1 Cell clusters

Manual tracking in a 3 dimensional environment of non-confocal data is not trivial, as for example motion in the z-dimension can only be seen via the diameter of the


A
B
C
A
P
L
D
L
V
L
00'00"
05'00"
10'00"
15'00"
20'00"
25'00"
30'00"
35'00"
40'00"
45'00"

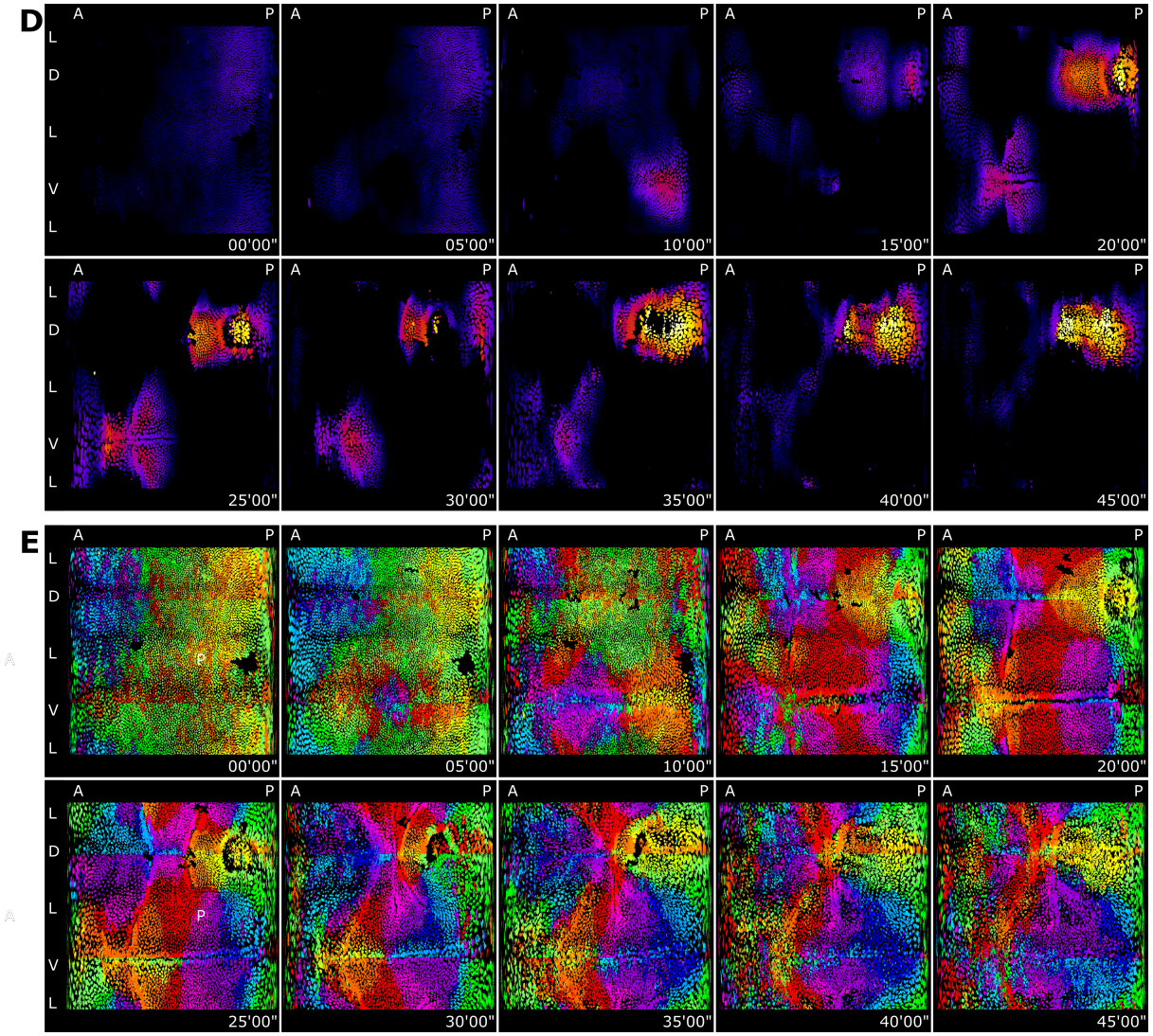


Figure 3.2: **Cell movement during germband extension** This figure shows the cell movement during germband extension using particle image velocimetry, as well as one speed and one direction based visualization. Panels **A-D** show the speed of the segmented nuclei using a heatmap, in **A** all directions, **B** anterior-posterior direction, **C** posterior and **D** anterior direction. Panel **E** shows the direction using a circle sector based spectrum visualization. At 0 minutes, there was neither directed nor fast movement as shown by the dark colors in**A-D** and random colors **E**. At 5 minutes the first movements in dorso-ventral direction could be observed at the ventral midline as shown by a significant label change in panels **A** and **E**. After 10 minutes, the movements of the ventral furrow invagination had significantly increased and involved now also anterior-posterior motion **A-E**. The anterior half of the ventral furrow moved in posterior direction **C**, while the posterior half moved in anterior direction **D**.

Figure 3.2: At 15 minutes, the ventral furrow became clearly visible as a deep invagination. At the same time the first germband extension movements could be seen as the anterior-posterior movements on the ventral side were switched **C-D** and the posterior pole started to move **A** and **E**. From 20-35 minutes germband extension accelerated and progressed at speeds partially above 10 microns/minute (white color in **A-D**). During this time germband extension was bidirectional on the ventral side **C** and **D**. At 40 and 45 minutes when the dorsal side was cramped **E**, the overall speed decreased **A** and ventral anterior germband extension faded **D**.

tracked nucleus. As the epithelium was tightly packed, confusion of neighboring nuclei was possible, which made a quality control necessary. Neighboring cells in the epithelium on the surface of the embryo are linked via elastic junctions [Clément et al., 2017], which meant that they should have remained in close proximity during the whole time, which on the other hand suggested, that their trajectories had to be parallel. These requirements were confirmed during tracking as it very soon became obvious that cells from similar origins had similar tracks regarding trajectory and velocity, and often also divided in similar areas. However the tracks were not often only similar, but rather parallel, which confirmed the expected high level of epithelial integrity, as neighboring cells remained in close proximity over the entire track length. This quality control allowed me also to cluster the cells into groups using a hierarchical classification with the following criteria:

1. Tracking area (Ventral, Dorsal, Lateral)
2. General anterior-posterior position (Head, Trunk)
3. Specific position and general heading (heading optional) (Dorsal, Lateral, Ventral, CF, VF, Pole cells)
4. Specific heading (optional) (Remaining, Dorsal, Lateral, Ventral, Anterior, Posterior)

I obtained a total number of 25 clusters, 9 from the ventral, 7 from the dorsal and 9 from the lateral tracking area (figure 3.4). All cells that were present at 5 minutes after onset of gastrulation (start of tracking) with a trajectory similar

enough to other trajectories were assigned to one of the 25 groups. Daughter cells originating from mitosis, and cells which could not be assigned to a single group with enough confidence were omitted. In figure 3.5 all cells assigned to one of the 25 clusters are shown. Their track colors correspond to the colors in the classification scheme in figure 3.4.

I used these clusters to calculate average motion parameters for cells from the same region. In combination with the same parameters for the structures I was able to obtain a detailed *in toto* motion map of the Drosophila embryo (figure 3.6). This *in toto* motion map not only allowed the description of motion in the embryo, but it also allowed for the analysis of the timing of the individual motions with unprecedented precision.

3.3.2 Gastrulation is made up of highly integrated single movements

One of the common problems in analyzing Drosophila gastrulation is the tight coupling of events: everything seems to move at the same time, except for VF formation which has been described as the first process [Campos-Ortega and Hartenstein, 1997, Lye and Sanson, 2011]. So far, it was very hard to differentiate which cells or regions move first, which means that the question for the motors of gastrulation had no clear answer and was still open for speculation. Using the *in toto* motion map from the tracking I was able to highlight the regions of the embryo where the primary movement occurs, and even more to put all relevant tracked movements into a chronological order (figure 3.6). In order to do this I had to separate directed motion from background Brownian-like motion. I did this by defining a threshold of 1.2 $\mu m/min$ (dotted line in figure 3.6 A). I defined this threshold using the information from the PIV analysis, which showed that in the first minutes of gastrulation, only the VF invagination took place. The other cells did not move in a certain direction at this time, but rather wobbled around, reaching velocities up to 1.1 $\mu m/min$.

The first movement observed was VF invagination, as already predicted by the PIV analysis (figure 3.6 A). Around minute 12-13, VF invagination was followed

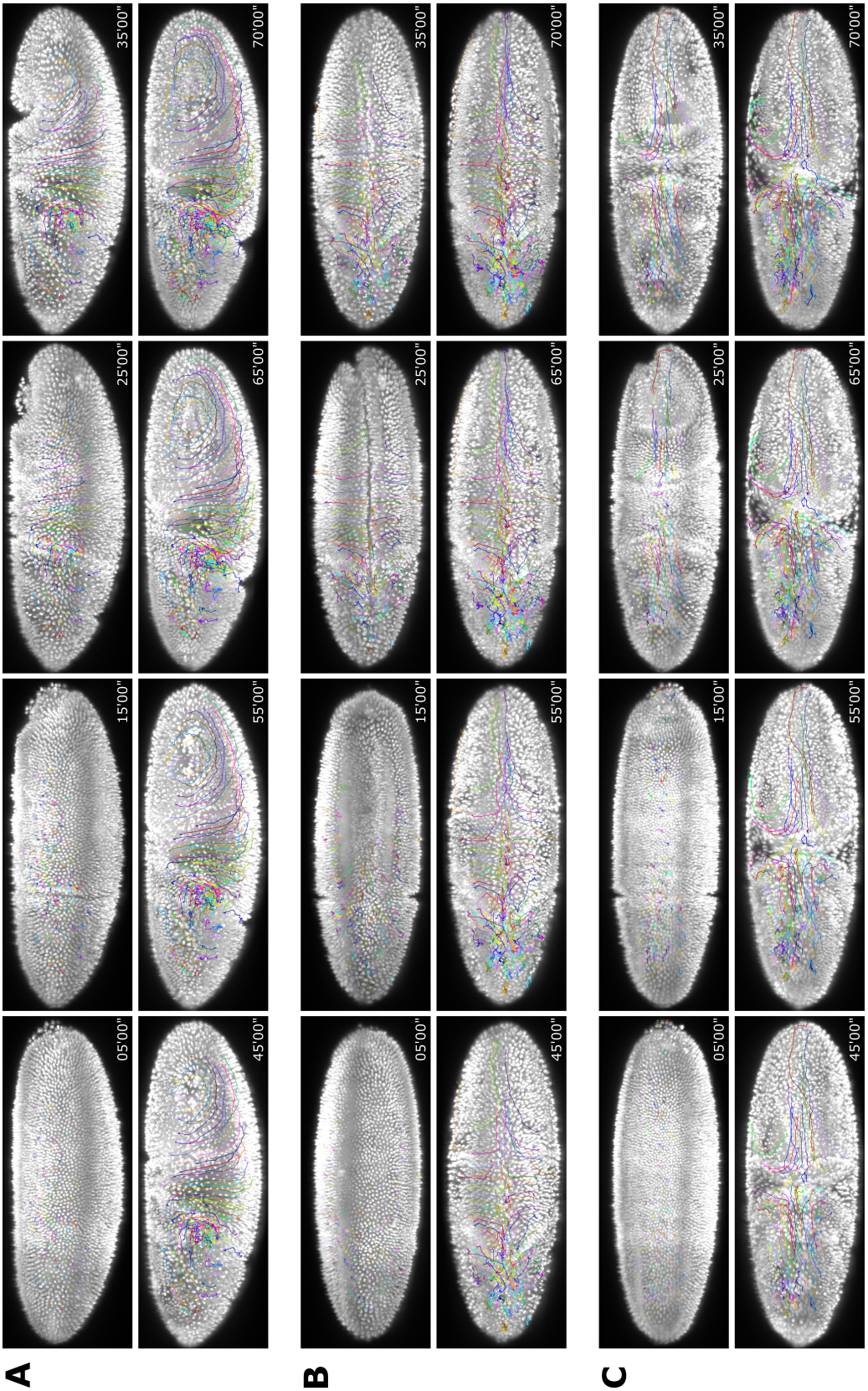

A
B
C
05'00"
15'00"
25'00"
35'00"
45'00"
55'00"
65'00"
70'00"

Figure 3.3: **Cell and structure tracks** Panels **A-C** depict the cell tracks tracked on the lateral, ventral and dorsal side. A total number (including mitosis) of 360 cells were tracked for a total time of 65 minutes starting at 5 minutes past the previously defined onset of gastrulation. Cell tracks were often parallel, especially on the lateral side **A**, in addition they were bilateral symmetric on the ventral and dorsal sides **B** and **C**. In panels **D** and **E** the following structure tracks are depicted: Germband extension progress (green), dorsal cephalic furrow (blue), ventral cephalic furrow (pink), lateral-ventral tips of the cephalic furrow (orange and purple).

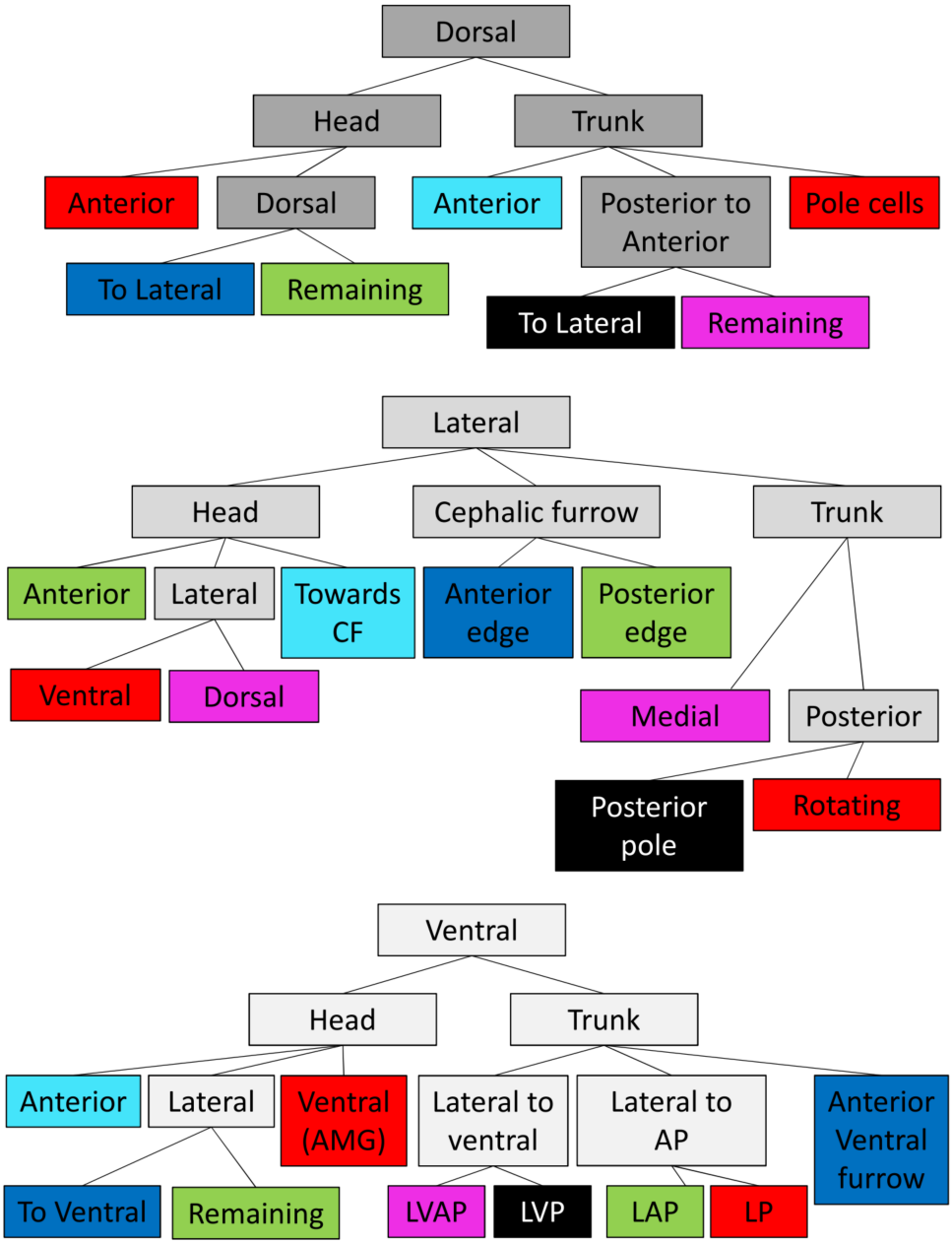


Figure 3.4: **Cell classification scheme** The majority of the tracked cells were classified into 1 of 25 clusters to allow for a more robust analysis. The classification was done using a flow chart like scheme with the following underlying criteria: Side of the embryo, coarse and fine cell origin as well as heading. The colored boxes depict the clusters. The tracks belonging to each cluster can be found in figure 3.5.

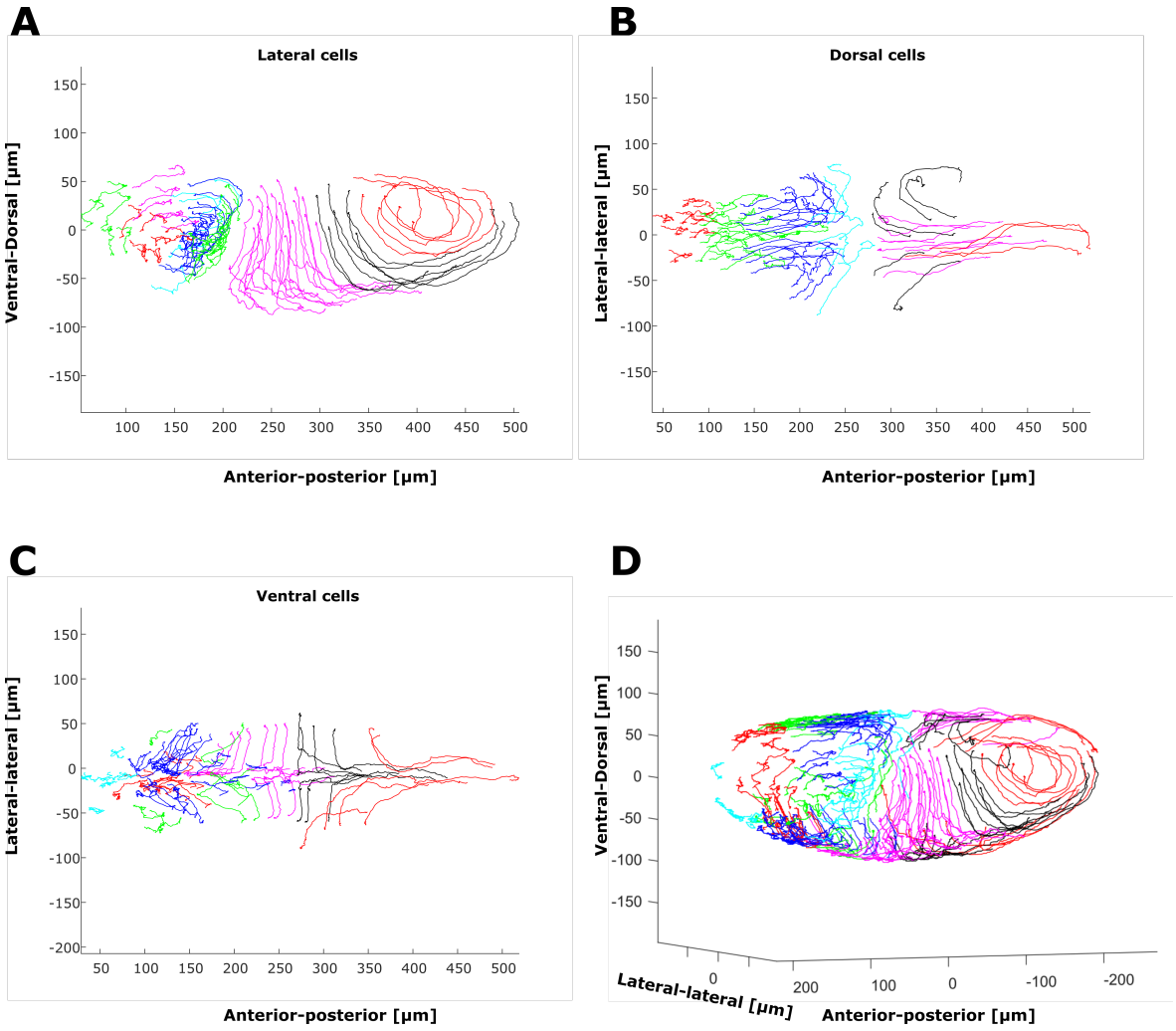


Figure 3.5: **Track clusters** Panels **A-C** depict the tracked cell trajectories of the clusters from figure 3.4 from the **A** lateral, **B** dorsal and **C** ventral side. Panel **D** depicts a 3D view of all clustered cells. Trajectories of one cluster were often parallel **A** or bilateral symmetric **B** and **C**.

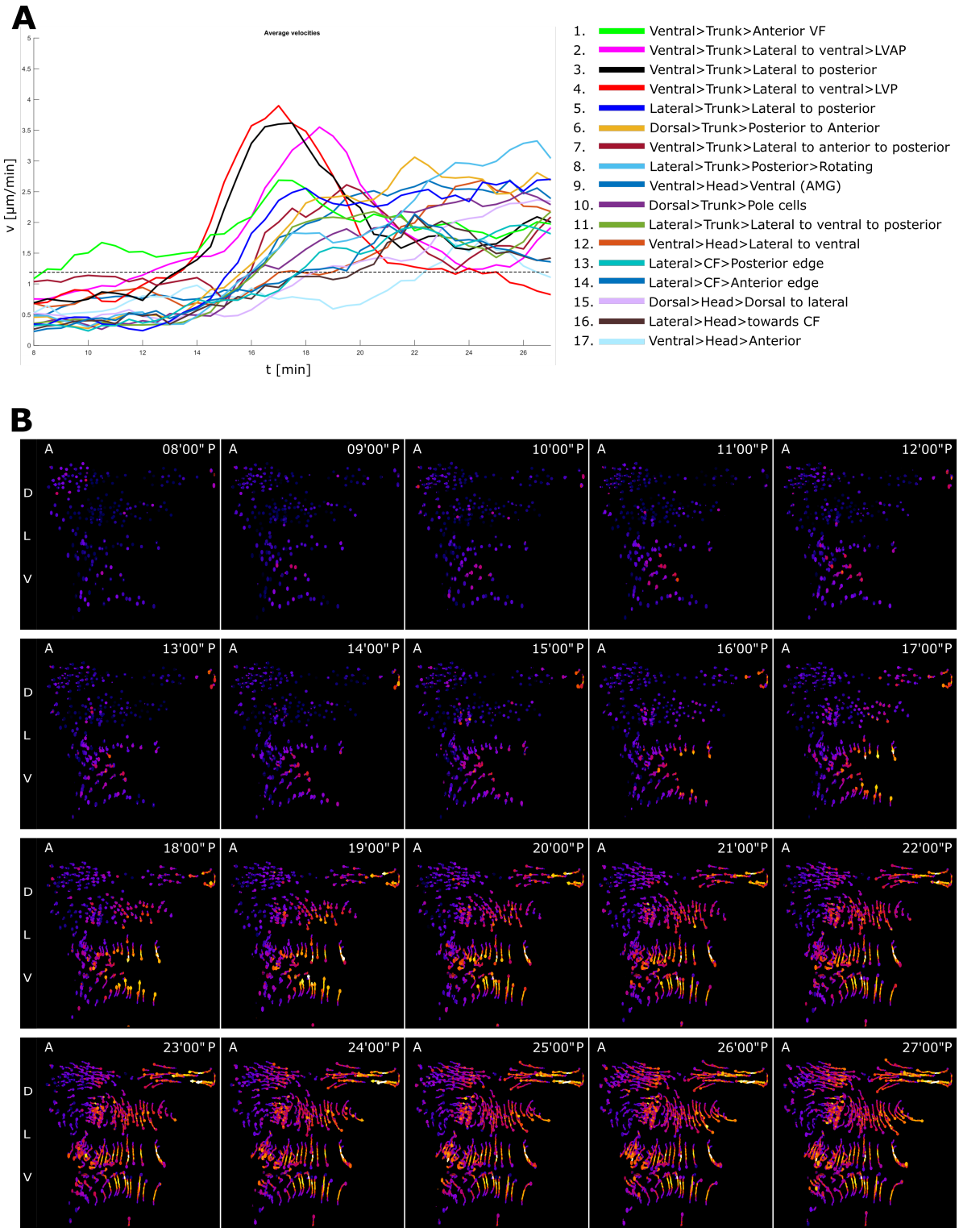


Figure 3.6: **Chronological gastrulation movements** Panel **A** depicts the cell velocities of selected clusters in the initial minutes of gastrulation. The legend next to the chart depicts a chronological list of these observed movements.

Figure 3.6: To separate relevant motion from background noise, a threshold was defined at 1.1 $\mu m/min$. At first the ventral furrow invaginated, and was soon followed by surrounding ventro-lateral cells. After the ventral cells, the first lateral (5) and dorsal cells (6) followed. After these cells almost the whole embryo started to move. The head region was slightly delayed, and started its motion with the invagination of the anterior midgut (9). Panel **B** depicts the motion of the single cell tracks, using the same unrolled heatmap visualization as shown already in figure 3.2 A. Dark colors correspond to little or no velocities, whereas high velocities ($>$ $10\mu m/min$) are labeled white.

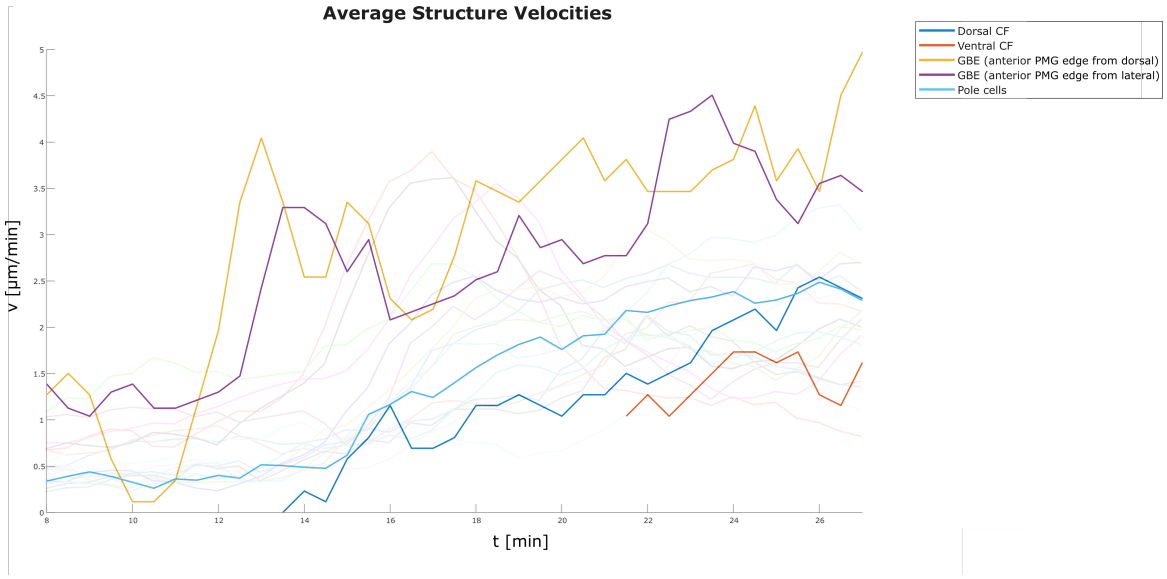


Figure 3.7: **Structure velocities** This figure depicts the structure velocities in the foreground, as well as the cell cluster velocities from figure 3.6 in the background. The progress of germband extension (measured at the anterior posterior midgut invagination edge) was tracked from the lateral and dorsal side (purple and yellow graphs). The dorsal and ventral cephalic furrow sides are depicted in blue and red. The pole cells are shown in light blue. The posterior midgut was the first structure showing motion and started to move directly after the first ventral furrow invagination movements. However, this movement decreased shortly afterwards, and in a subsequent step all structures, except the ventral cephalic furrow started accelerate at the same time when most of the cell clusters accelerated.

by 3 clusters of ventrally located cells. The cells of these clusters moved from their initial ventro-lateral positions, towards the ventral side and then to the anterior end with a consecutive u-turn towards the posterior end of the embryo (2), directly to the posterior (3) or first to the ventral side and then to the posterior end, skipping the anterior turn (4) (figure 3.6 A). After these ventral cells, with a delay of roughly 2 minutes, the first lateral cells, located at $60\% - 70\%$ EL started moving in ventral and posterior direction (5) (figure 3.6 A). Almost immediately afterwards, cells on the posterior dorsal side of the embryo moved in the anterior direction (6). These cells depicted the first movement in only the AP-direction, a clear mark for the beginning of GBE. This movement was after less than half a minute followed by 5 groups of cells starting to move almost at the very same time: The first group of those 5 was located on the ventral side, directly posterior of the developing CF, and moved first in the anterior-ventral, and after turning around in the posterior-ventral and posterior direction (7) (figure 3.6 A). The second group of the 5 was made up of cells on the lateral side in the very posterior area of the embryo. These cells completed a full rotation, similar to the motion suggested in the modeled flow fields from [Dicko et al., 2017] (8) (figure 3.6 A). These rotating cells were followed by cells which invaginated in the AMG invagination, marking the first head movement (9). After the cells from the AMG invagination, the pole cells accelerated (10). However, the heatmap in figure 3.6 B shows that they already started moving very early albeit at a lower velocity. The last group of the 5 was a large group of cells in the medial lateral area, moving towards the ventral side and later towards the posterior end of the embryo (11) (figure 3.6 A). After 1-2 minutes, the next 4 groups of cells joined the already almost global motion. At first, cells from the ventro-lateral head area moved towards the ventral side, to join or possibly replace the cells from the AMG invagination (12). Directly afterwards, the cells in the anterior and posterior CF started to move with almost the same velocity (13+14) (figure 3.6 A). As last of this 4 groups, cells in the dorsal head region moved in posterior direction towards the CF and then continued in lateral direction (15). This movement was followed after half a minute by cells located in the lateral head, which moved towards the CF (16) and after further 2 minutes

by cells in the ventral anterior tip of the head which moved slightly dorsal (17) (figure 3.6). Figure 3.6 B shows the same data using heatmap colored tracks in an unrolled embryo, similar to figure 3.2. Using the insights gained from the previous analysis, I was able to address the above raised question concerning the "motors" of gastrulation, mainly GBE, a question that has received quite some attention in recent years. My data showed that the first movement in the gastrulating embryo, the VF invagination, was first followed by ventro-lateral cells moving in the ventral direction, and only later by cells moving in the AP direction (group 3 from figure 3.6 also depicts a significant ventral motion vector, see figure 3.5). These cells were then followed by lateral cells (5) switching from an initial DV movement towards an AP directed movement. Finally with group (6) cells only moving in the AP direction started to accelerate. This AP-movement was likely propelled by the PMG invagination, which is, like the VF invagination, an apical constriction dependent invagination, producing extremely strong forces [Sweeton et al., 1991, Collinet et al., 2015]. However, another possible hypothesis could be that cells were propelled by the rotation of the CF. To confirm or reject the above raised hypothesis, I used the structure tracks, of the following interest points (figure 3.7)

- Dorsal cephalic furrow
- Ventral cephalic furrow
- Posterior midgut invagination / Germband extension at the anterior PMG edge (dorsally tracked)
- Posterior midgut invagination / Germband extension at the anterior PMG edge (laterally tracked)

To complement these structure tracks, I added the tracks from the pole cells (figure 3.7). The onset of PMG invagination came very early during gastrulation, with a first visible movement at 12 minutes after onset of gastrulation, lasting for 4 minutes. This motion started around 4 minutes after the first motion of mesoderm invagination and 2 minutes before the first ventro-lateral cells followed the initial

mesoderm cells (figure 3.7). At 16 minutes past onset of gastrulation the velocity of GBE started to increase with a constant acceleration (pitch of the plot) together with multiple cell clusters moving in the AP and DV direction, including the pole cells. Around the same time, the dorsal CF also started to move in posterior direction (figure 3.7). All these movements suggested that gastrulation became global at this time point. However, one process was still missing: the ventral CF, which was not even developed yet, started to move only 5 minutes later (figure 3.7). This finding excluded the CF as motor of GBE, but even more it put into question the often described picture of the CF as a planar and stable structure.

These results in context with published data suggested that the initial GBE movements in the trunk depended on 2 processes involving apical constriction: VF and PMG invagination. However as GBE seemed to be, at least in the initial phase, bidirectional, and the pull of the PMG invagination was in the posterior direction an anterior directed force was still missing. Interestingly the first movement in the head I observed was the AMG invagination. This brought me to the conclusion that the AMG invagination might be responsible for the pull of the germband in the anterior direction, just as the PMG invagination is responsible for the pull in the posterior direction [Collinet et al., 2015].

3.3.3 Epithelial integrity in the cephalic furrow is maintained

I showed above the motion of cells in context with surrounding structures on the surface of the embryo, however, the question for the behavior of cells inside these structures still remained open. I did not know yet whether cells inside a structure like the CF remained part of the surface epithelium or whether their properties changed as soon as they were inside, leading to a behavior as one object / furrow independent of the epithelium. In figure 3.3 I showed that most of the tracks were parallel suggesting a very high level of epithelial integrity on the surface of the embryo. However to assess the difference between epithelial integrity on the surface and in the CF I required a quantitative analysis. Therefore I used a custom made solution based on triangles. These triangles were assigned at the first tracked time point (5 minutes past onset of gastrulation) and monitored over the entire

track length for 2 parameters: the triangle area (A) and the circumscribed circle diameter (CCD) (figure 3.8). Both parameters, and especially their combination reveal shape and size changes of the triangles without being limited to a single point like an angle. The assignment of the triangles was done via a visual inspection of the tracks, based on objective parameters like nearest neighbor and triangle shape, but also on subjective parameters like experience.

I used the triangles on cells from both the anterior and posterior edge of the CF. As ground truth for a surface epithelium, I used on the one hand the lateral medial region of the embryo's trunk. These cells moved first in the ventral and later in the posterior direction, therefore accounting for the direction change of the CF cells, which moved first in the AP and later in the DV direction. On the other hand I used cells from the dorsal head region, which moved shortly in the AP direction, but remained in their original region. These dorsal head cells were part of a mitotic domain, and accounted for the mitosis present in the CF.

Over the tracked 65 minutes, the average area of the triangles of all regions roughly doubled (figure 3.8 C). The same was true for the CCD (figure 3.8 D). Both of the parameters, the area and the CCD proved to be very noisy, and did not show any significant differences between the surface and the CF. These findings suggested that the CF remained the whole time part of the epithelium, as already suggested by the fact that the CF unfolds during later embryogenesis without leaving any traces behind [Campos-Ortega and Hartenstein, 1997].

My analyses of cell behavior supports the current understanding that the epithelium inside the CF is maintained. They also showed that the CF is probably not a motor of GBE. I could not detect evidence for a specific CF-surface interaction nor any indication for involvement in the speed up of GBE, suggesting that the CF does not act as a motor of GBE.

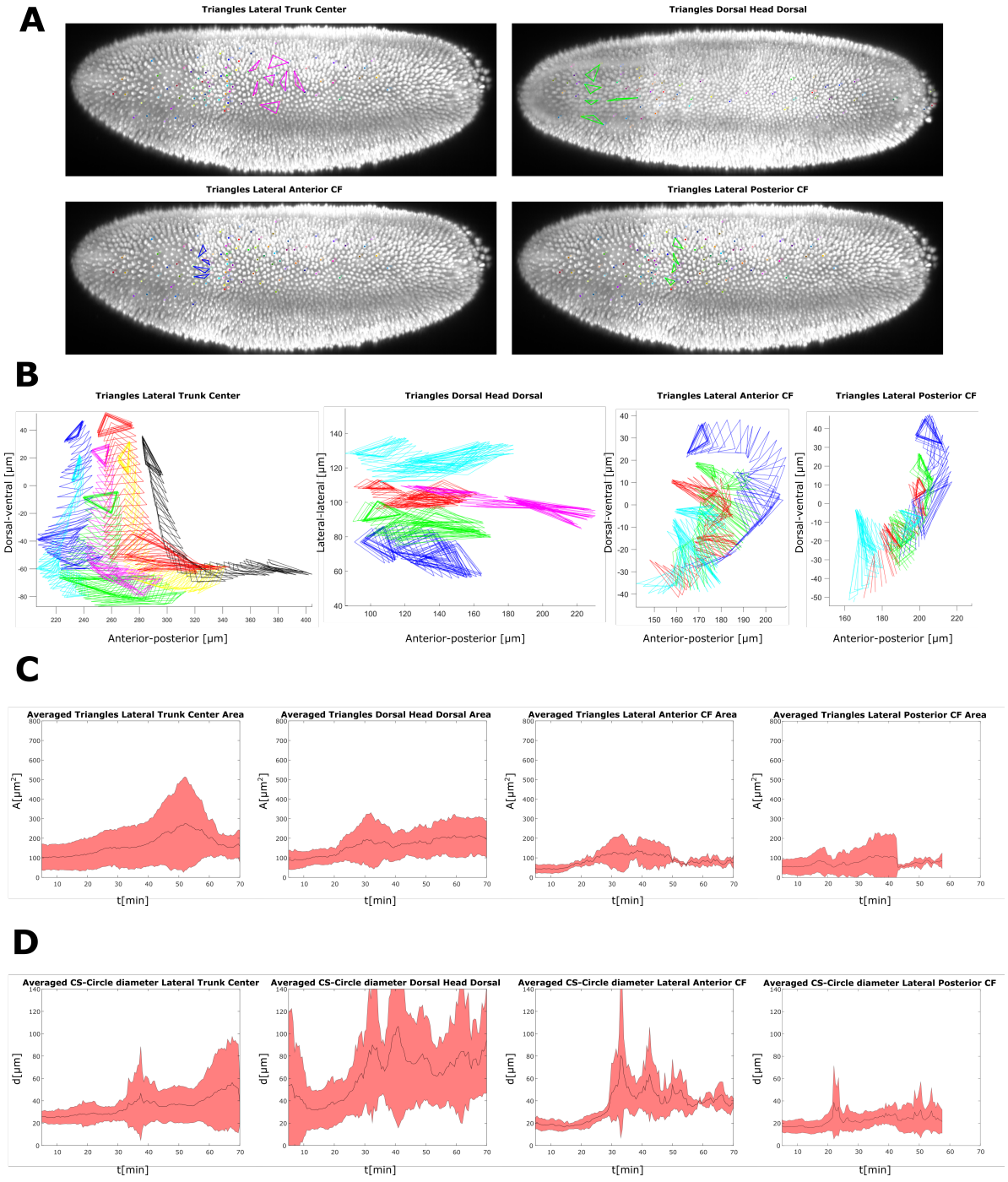


Figure 3.8: **Epithelial integrity was maintained in the cephalic furrow** Panel **A** depicts the chosen triangles for each of the 4 analyzed tracking clusters on a lateral maximum intensity projection. Two of them were surface clusters, one from the lateral trunk area, and one from the dorsal head region, the other two were the anterior and posterior cephalic furrow edges, invaginating in the embryo.

Figure 3.8: Panel **B** shows the location of the triangles over time. All triangle trajectories showed a high order, there was no overtaking and the trajectories were mainly parallel. In panel **C** and **D** the area and the circumscribed circle diameter are depicted. Besides the high noise shown by these parameters, no obvious differences were found, showing that epithelial integrity was maintained on, as well as below, the surface.

3.4 The cephalic furrow in context of germband extension

As epithelial fold, the CF maintained epithelial properties. While not surprising, this finding highlighted a challenge for embryo-wide morphogenesis. The CF is formed in an area where anterior vortex-like cell movement and poster straight cell movement were located in close proximity. Here, straight and circular motion vectors could be found next to each other. Like in other contexts, the coinciding but opposing movements are likely to produce shearing forces. Furthermore, these shearing forces are likely to increase in case GBE velocities are as high as it is the case in *D. melanogaster*. My observation of very high epithelial integrity in spite of predicted shearing forces, indicated the existence of a tension-release mechanism that prevented the tissue from otherwise inevitable rupture.

3.4.1 Defining the cephalic furrow as object

To test whether the CF was a structure that had the potential to handle increased shearing forces stemming from high-speed GBE, I first defined and analyzed CF properties as a single multicellular object. For the handling of such shearing forces the CF will require certain reproducible features. As its proposed function is of mechanical nature, certain characteristics concerning mechanical robustness will have to be present. To test whether these requirements were fulfilled, I selected a set of features that were most suitable to reflect the mechanical properties and robustness of a furrow:

1. **Depth:** The depth of a furrow seems to be directly linked to rigidity, having

a role similar to the material thickness of a piece of metal.

2. **Deformation (planarity):** The deformation of a furrow is a direct measure for the above mentioned rigidity. In the case of the CF which resembles a conic section, the deformation can be measured via its planarity.

3. **Translocation (rotation angle):** The larger or deeper the furrow the more force is likely required to translocate its position. This is however a more indirect measure, as some translocational process such as rotation might be more favorable than others like linear displacements.

4. **Extension (circumference):** The extension of a furrow is a similar measure to its depth, just in another dimension. For the CF the completion in % of the embryo circumference is another measure for its rigidity.

I analyzed the first 3 features in fixed embryos which allowed for a higher sample number than SPIM to account for reproducibility. It appeared, that the CF was still growing deeper during the early phase of germband extension. To be able to use the results despite this issue I used GBE progress as time marker. I found that the depth of the CF increased proportional with GBE progress (figure 3.9 A). The angle between the CF and the AP axis was between 65° and 75° before or shortly after the onset of GBE. During GBE this angle dropped to a value between 50° and 60° (figure 3.9 B). The planarity of the CF remained at a very good level at the onset of GBE, but later dropped to a slightly lower level (see spread in y-direction, figure 3.9 C or S-shape, figure 3.1 A). Important to note was the consistency of this data among the individuals. The small number of outliers (especially figure 3.9 C) can be explained by issues fitting the plane.

As closely related Drosophila species and even populations of *D. melanogaster* can be found in a variety of habitats with varying environmental conditions [Kuntz and Eisen, 2014], I tested whether these features were robust against global stress. I moved eggs after deposition to 29° C and analyzed the embryos for the same parameters. I found a slight, but rather irrelevant change in all parameters (3.9 A'-C'). These results showed that the CF was quite robust and did not show too much variation, even at 29° C.

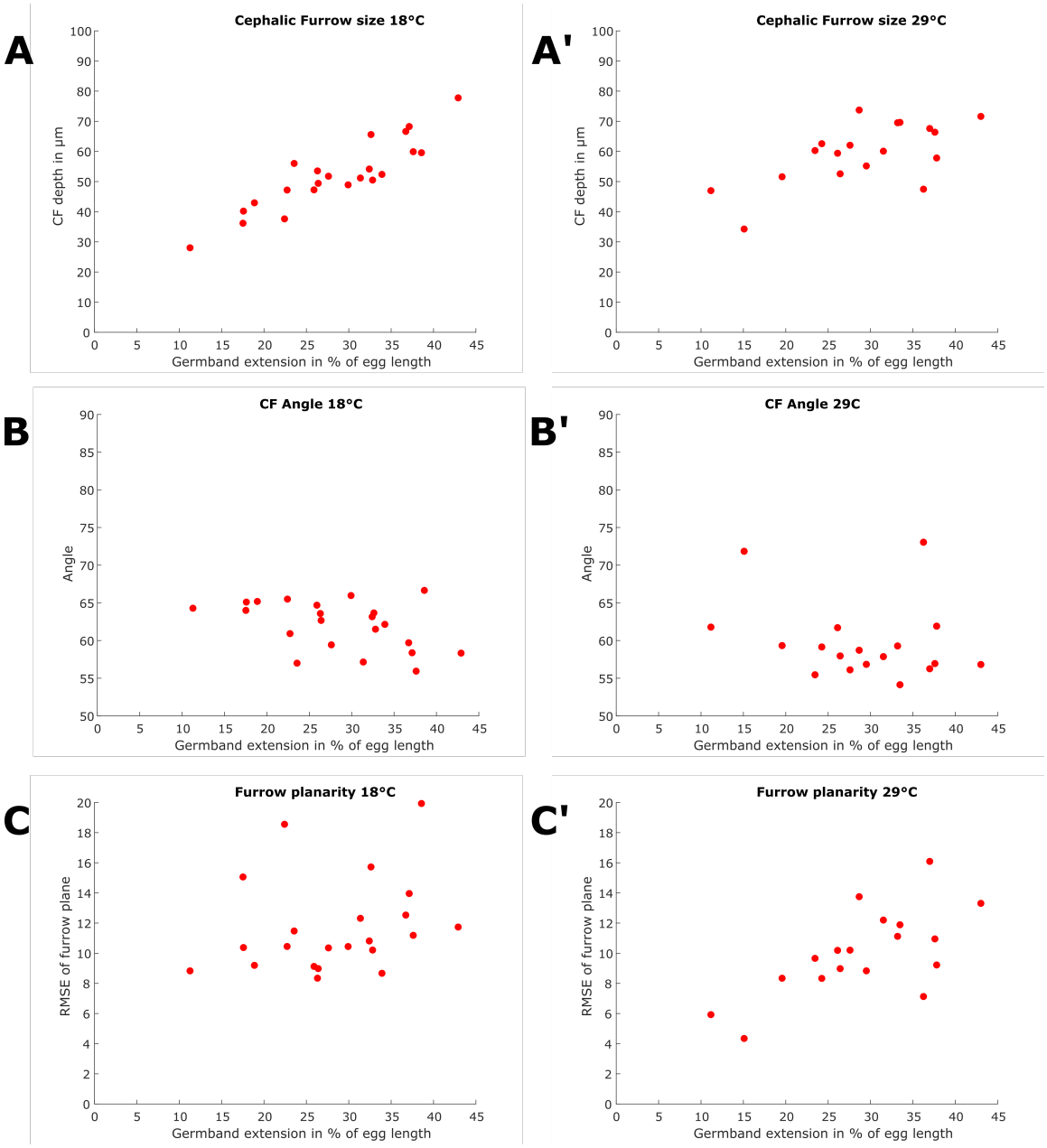


Figure 3.9: **Cephalic furrow parameters** Panels **A-C** show depth, the angle (between cephalic furrow and anterior-posterior axis) and the planarity of the cephalic furrow for embryos developed at 18° C and 29° C. All these 3 parameters were dependent on the progress of germband extension. Their temperature dependence was however low, indicating a temperature independence of the cephalic furrow.

3.4.2 The difference between the inner and outer cephalic furrow

To find out whether the CF could handle the shearing forces occuring during germband extension between the head and trunk region I needed to get a more detailed picture of it. My previous analyses showed that the cells in the CF were highly motile. I required a detailed analysis of this motion to find similarities, pinpointing epithelial integrity, and differences, pinpointing shear and associated mechanisms, between the anterior and posterior edges. Because the CF did not show distortion at its surface, I speculated that the solution is rather hidden in the inner structure of the CF. This assumption was in particular based on three reasons: 1. As epithelial integrity was maintained in 2D, its limitations were also valid for 2 dimensions, but might not be valid in a third dimension. 2. During tracking I had the impression that I could observe tissue faults deep inside the CF. This observation was consistent with the shear handling hypothesis. 3. Any features and the structure of the CF close to the surface had already been quite well investigated without major findings [Turner and Mahowald, 1977, Vincent et al., 1997, Campos-Ortega and Hartenstein, 1997, Spencer et al., 2015].

In a first experiment I analyzed the velocities of the cells in the the anterior and posterior edge of the CF. The overall velocity (scalar) did not show any significant differences (figure 3.10 A); the DV velocity (vector) showed minor differences at the beginning of GBE (figure 3.10 B). However this can be explained by the rotating movement in the CF area and of the CF itself, as vectors on a circular track are highly position dependent. From these results, it looked like the anterior and posterior CF edges were tightly coupled, as expected for an epithelium. It seemed like the shear force handling was not velocity based.

However, considering the internal faults, I observed deep in the CF, shear stress handling could well be based on a 3 dimensional structure invisible from the outside. To test this possibility I had to do a full analysis of invagination, including anterior and posterior velocity vectors in detail, and the structure of the CF over the course of GBE. Furthermore, as shear stress handling required flexibility in the

AP- and DV-direction, I required an analysis which could show motion in these two directions simultaneously and in the best case analyze epithelial integrity at the same time, to confirm that shear stress handling is executed by the mechanisms found, and not a sudden change in epithelial integrity.

I used the triangle analysis I introduced above (figure 3.8), as it fulfilled all these criteria. This time I defined the triangles either between one anterior, and two posterior CF nuclei or the other way round between two anterior and one posterior nucleus (figure 3.11 A). The triangle parameters, the area and the CCD showed neither any significant increase over time, nor much noise (figure 3.11 C). This finding was very interesting, as it seemed to show that even between the anterior and posterior sides of the CF the epithelium was strongly conserved . However, the triangles themselves showed a very different behavior; all of them turned from their initial parallel surface orientation to a perpendicular orientation towards the surface, where the posterior cells were located deeper inside the embryo than the anterior cells (figure 3.11 B). These results showed that the CF in the tested embryo invaginated asymmetrically. In addition, the triangle positions were not conserved in the DV-direction, as the cells of some triangles, which were originally more dorsally located, ended up more ventrally than the cells of other more ventrally located triangles. This loss of DV-positioning was possibly a result of the internal tissue faults. The triangles also made those internal faults visible, as they were represented in the trajectories of the triangles.

To sum up, I found that invagination in the CF was asymmetric in the AP-direction and the original DV-positioning was not maintained, possibly because of internal faults. These findings illustrated that the properties valid for 2D epithelia might not necessarily be valid for a 3D epithelium. These results were consistent with my hypothesis, as the CF seemed to provide the flexibility in both AP- and DV-direction necessary for the handling of shearing forces.

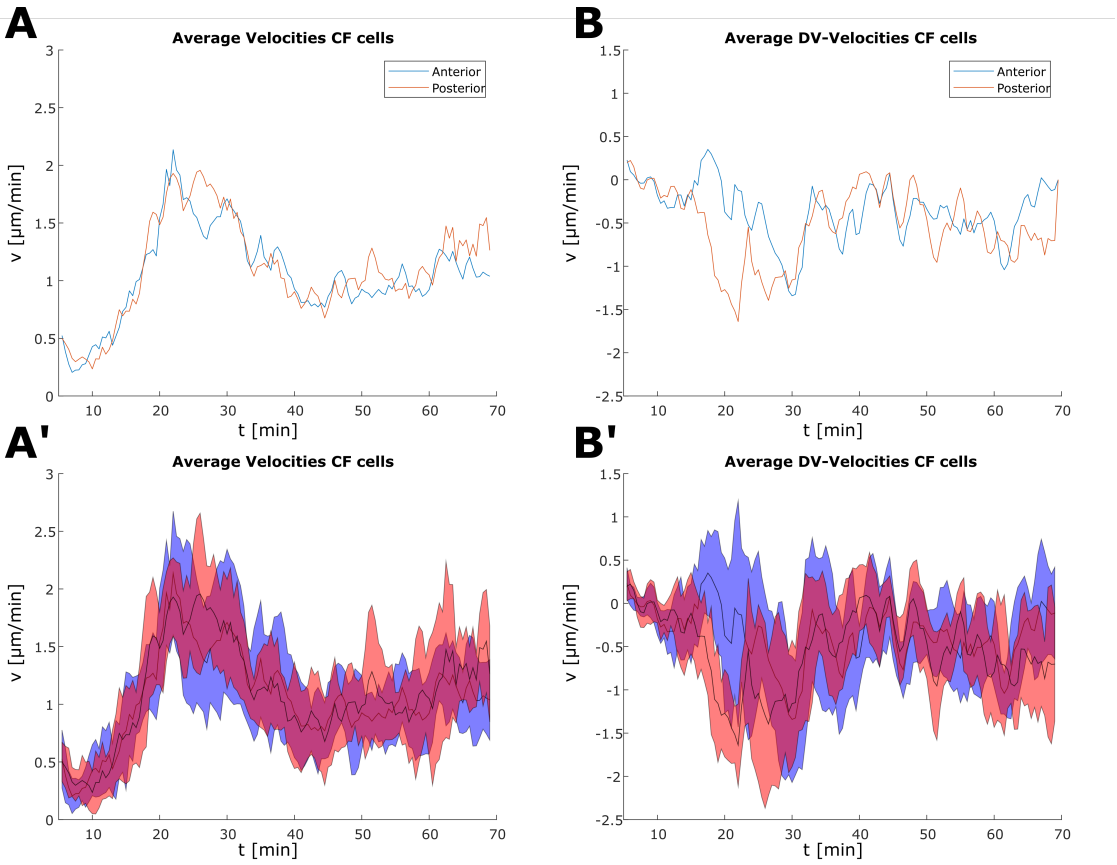


Figure 3.10: **Cephalic edge cell velocities** The velocities of the anterior and posterior cephalic furrow edges are shown here for **A** and **A'** all directions as scalar, and **B** and **B'** in dorso-ventral directions as vector (negative values: ventral direction, positive values: dorsal direction). Except for a larger difference at the onset of germband extension where the anterior cells moved more in the posterior direction, while the posterior cells moved already in the ventral direction, the velocities were rather similar.

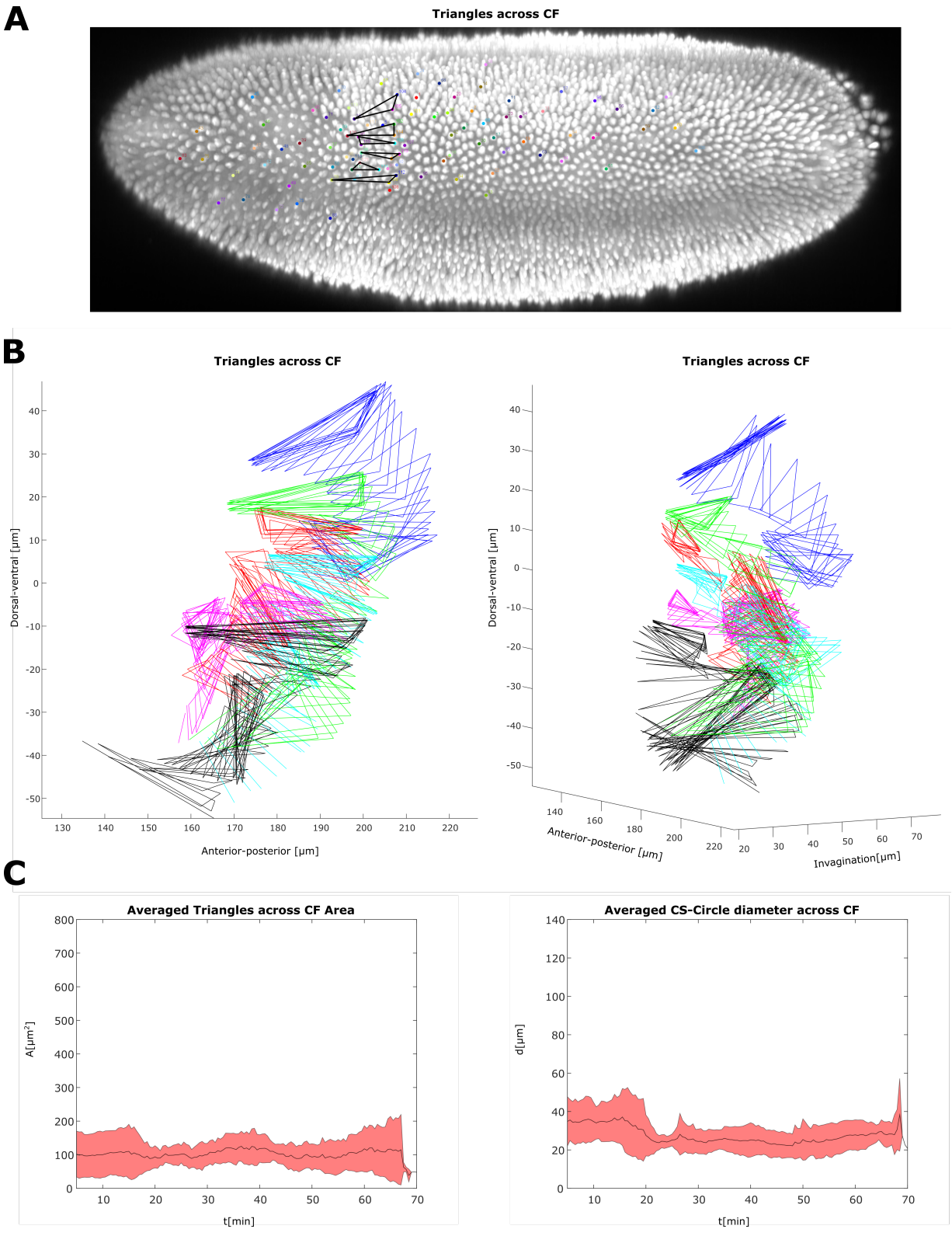

A
Triangles across CF
B
Triangles across CF
Dorsal-ventral [µm]
Anterior-posterior [µm]
Triangles across CF
Dorsal-ventral [µm]
Anterior-posterior [µm]
Invagination[µm]
C
Averaged Triangles across CF Area
A[µm²]
t[min]
Averaged CS-Circle diameter across CF
d[µm]
t[min]

Figure 3.11: **The cephalic furrow modulated epithelial integrity**
Here, the triangle analysis was repeated for triangles connecting cells from the anterior and posterior cephalic furrow edges. Panel **A** depicts the defined triangles on the embryo surface. In Panel **B** 3D plots of the triangles over time are shown. In contrast to the previous triangle analysis, the cells of some triangles passed by the other ones although the triangle parameters were very steady over the whole time **C**. In addition the triangles turned by 90° as the posterior cells moved below the anterior cells **B**. This behavior fitted my observation of internal faults deep inside the cephalic furrow and suggested a certain flexibility in the anterior-posterior and dorso-ventral direction. It appeared that the limits of a 2D epithelium could be overcome by the cephalic furrow via creating a 3D epithelium.

3.5 The posterior dorsal fold in context of germband extension

Besides the CF another deep transient fold exists in the Drosophila embryo. The posterior DF is located at 66% EL and forms during GBE [Turner and Mahowald, 1977, Campos-Ortega and Hartenstein, 1997]. During GBE it deepens rapidly [Wang et al., 2013] and its tips extend symmetrically on a curved trajectory into the direction of the germband fold [Campos-Ortega and Hartenstein, 1997, Ingham, 1994]. The function of the posterior DF is also unknown. Interestingly my tracks (figure 3.3) showed that the posterior DF was located in between cells whose trajectory was mainly dorso-ventral and lateral-posterior, and cells whose trajectory was a complete circle very much like the vortex proposed in the model from [Dicko et al., 2017] and similar to the region where the CF is located. Such a region where straight moving cells and cells in a vortex are next to each other, raises again the question how the different velocity vectors are handled with respect to shear and epithelial integrity. Could the posterior DF handle the shear associated with such cell movement by extending the epithelium into a third dimension, similar to what I proposed and observed for the CF?

3.5.1 Defining the posterior dorsal fold as object

Before I answered this question I asked the same question as for the CF before: Is the posterior DF mechanically stable enough to execute a given plausible mechanical function? Again I used fixed embryos to increase my sample numbers and used 18° C and 29° C as temperature conditions, to account for normal and more extreme living conditions. I measured the depth and extension of the posterior DF. It did not make sense to measure the angle or planarity, as the posterior DF did not represent a plane. The size of the posterior DF was similar to that of the CF, though there was a slight increase in size (depth and extension) at 29° C (figure 3.9 and figure 3.14).

Judging from the parameters it looked like the posterior DF was robust enough for a functional consideration. To see whether the cell movement in the posterior DF could provide hints for its function, I tracked cells from the dorsal midline area of the embryo, following them through the posterior DF. I did this experiment in a first attempt in the tracked embryo shown in the above sections, however, I realized very soon, that the invagination speed of the posterior DF was extremely high, which made me lose all tracked cells as soon as they were invaginated. In order to complete this experiment, I required a second *D. melanogaster* embryo, with superior imaging quality, as a better imaging quality can make up for a lack in temporal resolution. I used a nuclei marker emitting light in the infrared spectrum, which allowed for high quality imaging deep inside the embryo (Hufnagel, unpublished).

Cells which invaginated in the posterior DF on the dorsal side of the embryo resurfaced and left the posterior DF or came at least very close to the surface. These cells were at a first glance indistinguishable from cells which moved in the same direction, but stayed on the surface the whole time (figure 3.13). The posterior DF looked from this experiment very much like a measure for the evacuation of the dorsal epithelium, which was consistent with previous findings [Turner and Mahowald, 1977]. However, on the other hand I observed the same phenomenon for the CF. This brought up 2 possible hypotheses:

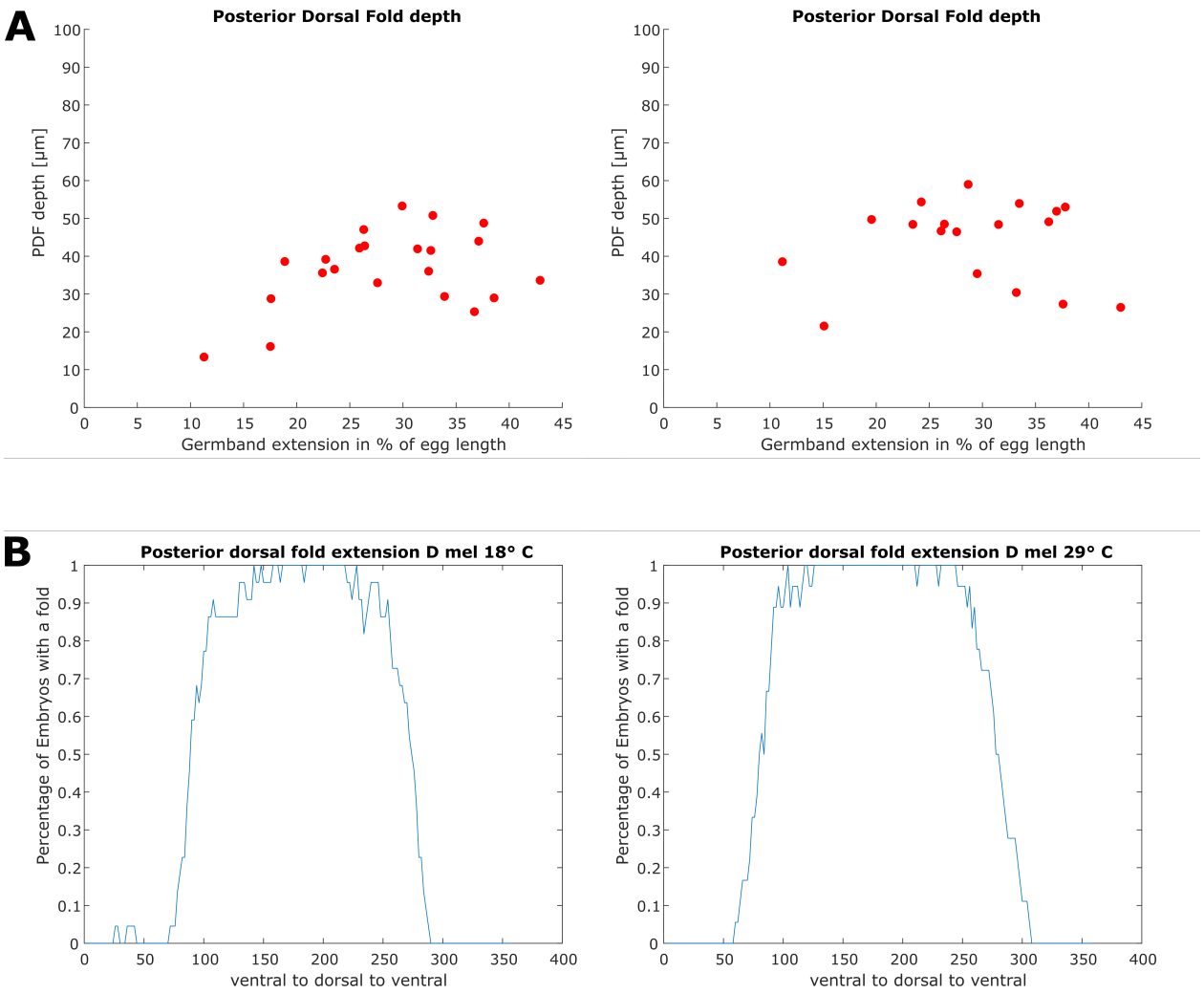


Figure 3.12: **The posterior dorsal fold**
Panels **A** and **B** depict the depth and extension of the posterior dorsal fold, similar parameters to those shown already for the cephalic furrow. It can be seen that at a higher temperature the posterior dorsal fold increased by a small amount in extension, while the depth remained almost unchanged. In general the posterior fold seemed to be robust, even at high temperature differences.

1. The posterior DF acts as a way for cells to exit from the cramped dorsal midline to the lateral sides, thereby making space for the extending germband, a loss of this transport capacity could lead to increased resistance on the dorsal midline. As a result GBE might be impaired.

2. The posterior DF acts as mechanical clutch to handle shear stress in a way similar to that which I hypothesized for the CF, a reduction in fold depth would likely limit these capabilities. This could possibly lead to ruptures in the epithelium or other phenotypes promoting the release of tension.

3.5.2 Germband extension is disturbed in embryos with a shallow posterior dorsal fold

Since my tracks showed that the posterior DF seemed to be tightly integrated into the cellular motion during GBE (figure 3.13), I wanted to test whether the motion dynamics during GBE got disturbed when the posterior DF was altered. Luckily, it has been shown recently, that a knockout of the GTPase activating protein Rapgap1 reduces the depth of the posterior DF in Drosophila to the depth of the anterior DF [Wang et al., 2013]. Using these *D. melanogaster* Rapgap1 knockout embryos, I had a simple way to test my hypothesis. At first I had to prove the consistency of the posterior DF size for the *D. melanogaster* Rapgap1 mutant, using the same analysis as already used for *D. melanogaster*. To reduce the work load (all furrows had to be tracked manually = 2h per embryo) I decided to stick with 29° *C* as mechanical perturbations were more likely to occur at higher developmental speeds, present at higher temperatures [Kuntz and Eisen, 2014]. As expected, the furrow depth and extension of the posterior DF were significantly reduced in the *D. melanogaster* Rapgap1 mutant.

While I analyzed (tracked) the furrows of these embryos it soon became obvious, that the *D. melanogaster* Rapgap1 mutant embryos had severe problems with maintaining a straight GBE. Their germbands deviated from the ventral and dorsal midlines, or in very severe cases twisted around the embryo. As a result I obtained embryos without clearly identifiable dorsal and ventral sides (figure

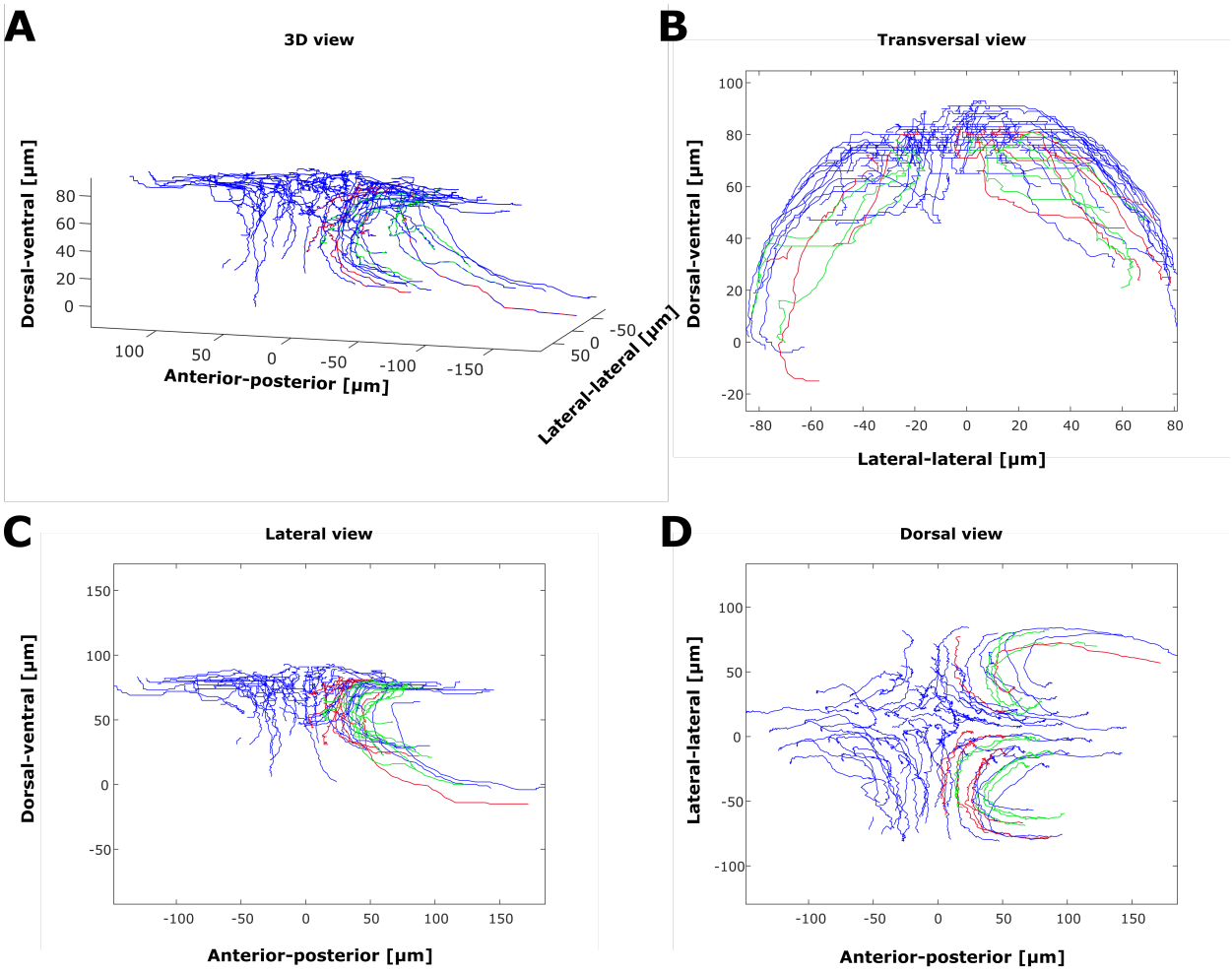


Figure 3.13: **Tracks in the posterior dorsal fold**
Cells in the posterior dorsal fold and surrounding areas were tracked in a second embryo using the same methods as described above. Panel **A** shows a 3D view of all tracked cells. Anterior edge cells were kept in red, posterior edge cells in green, and surrounding cells were labeled in blue. Panels **B-D** show the same cells in a transversal, lateral and dorsal view. Cells in or outside the posterior dorsal fold were indistinguishable at a first glance, suggesting that cells in the dorsal folds behave in a very similar way as surface cells.

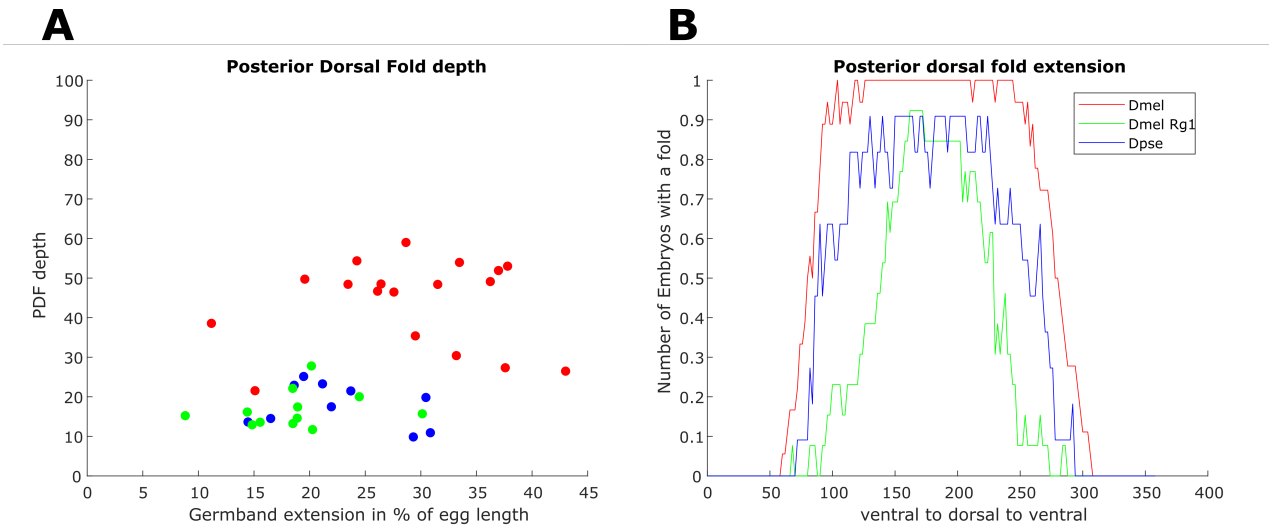


Figure 3.14: **Posterior dorsal fold sizes**
At 29° *C* the posterior dorsal fold in *D. melanogaster* wildtype embryos was twice as deep as in *D. melanogaster* Rapgap1 mutant and *D. pseudoobscura* wildtype embryos **A**. The fold depths of *D. melanogaster* Rapgap1 mutant and *D. pseudoobscura* wildtype embryos were very similar **A**.. The posterior dorsal fold was most extended in *D. melanogaster* wildtype embryos and least extended in *D. melanogaster* Rapgap1 mutant embryos. However *D. melanogaster* Rapgap1 mutant and *D. pseudoobscura* wildtype embryos did not correlate as well as for the depth **B**.

3.15). I quantified the amount of twisted GBE in all embryos using a custom made method (see section 5.2.2). I found that embryos which developed a shallow posterior DF, namely *D. melanogaster* Rapgap1 embryos, had significantly more, and more severe, twists (figure 3.15 C+D). It seemed that the manipulation of a fold with a possible mechanical function could severely disturb embryo development. However it was difficult to judge where the observed effect of twisted GBE came from. The twisted embryos did not reveal how this twist originated.

To confirm that the observed effect was the result of a reduced depth of the posterior DF, I decided to analyze *D. pseudoobscura* embryos which also have a small posterior DF (personal communication in the lab). This allowed me to check the effect of a reduced posterior DF depth, using 2 independent approaches. I did the same consistency analysis as before to check for the variability of the posterior DF size (figure 3.14). I found that the posterior DF in *D. pseudoobscura* embryos was very similar to the posterior DF in *D. melanogaster* Rapgap1 mutant embryos. My next question was whether this reduced fold depth also had the same effect on GBE. Indeed, I observed very similar effects considering quality and quantity of the twisted germbands 3.15 C+D). Interestingly, I had the impression (without considering the twisted germbands), that the *D. melanogaster* Rapgap1 knockout embryos looked more similar to *D. pseudoobscura* embryos, than to *D. melanogaster* wildtype embryos. To check whether this impression was true or not, I measured several parameters in all embryos. I found that not only the depth of the posterior DF was altered in Rapgap1 knockout embryos, but also their posterior midgut changed to a *D. pseudoobscura* like phenotype, which was the reason why they looked like *D. pseudoobscura* embryos (figure 3.16).

To conclude, I found that a shallow posterior DF promoted twisted GBE. I was however not able to see exactly how this twisted GBE originated. Independent of the exact mechanism, this finding showed that the alteration of a fold in the Drosophila embryo was able to induce mechanical perturbations, which is consistent with my hypothesis of tension release as role of the transient folds. However,the question for the exact mechanism and role of the posterior DF remained open.

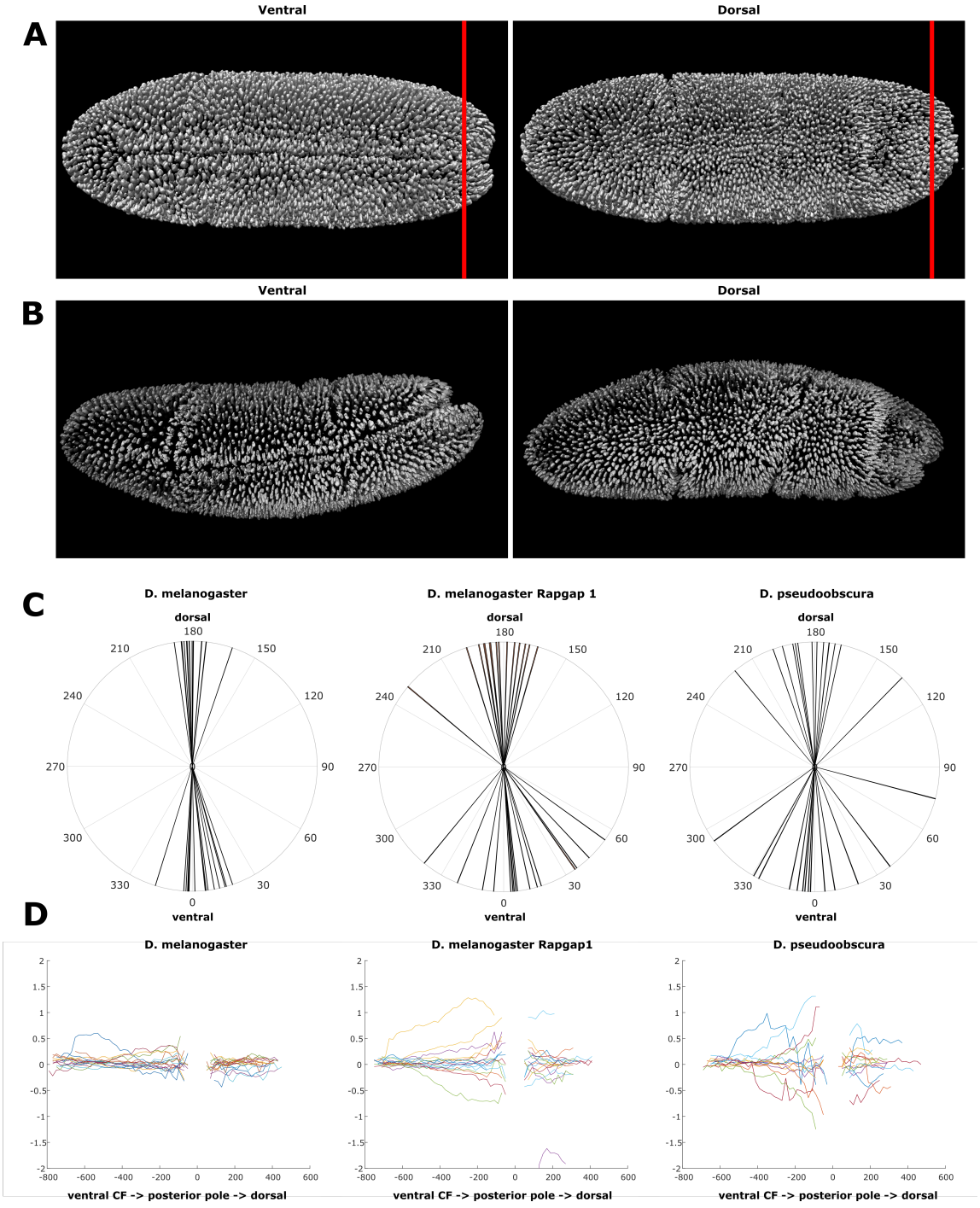


Figure 3.15: **Twisted germband extension in Drosophila**
Germband extension in *D. melanogaster* embryos usually appeared (and is also described) as shown in panel **A**.

Figure 3.15: Certain embryos showed a different phenotype which I refer to as twisted germband extension **B**. At 29° *C* *D. melanogaster* Rapgap1 mutant embryos and *D. pseudoobscura* wildtype embryos showed an increased number of germband twists, some of them being very severe, than *D. melanogaster* wildtype embryos **C** and **D**. Panel **C** depicts these twists in a transversal cross section close to the posterior pole (red line in **A**) of the embryo where 0° and 180° mark the ventral and dorsal midlines. Panel **D** depicts these twists in a tracking along the anterior-posterior axis (see 5.2.2) where 0 on the y-axis corresponds again to the midlines and the x-axis represents the total germband length from the ventral cephalic furrow to the anterior edge of the germband on the dorsal side of the embryo.

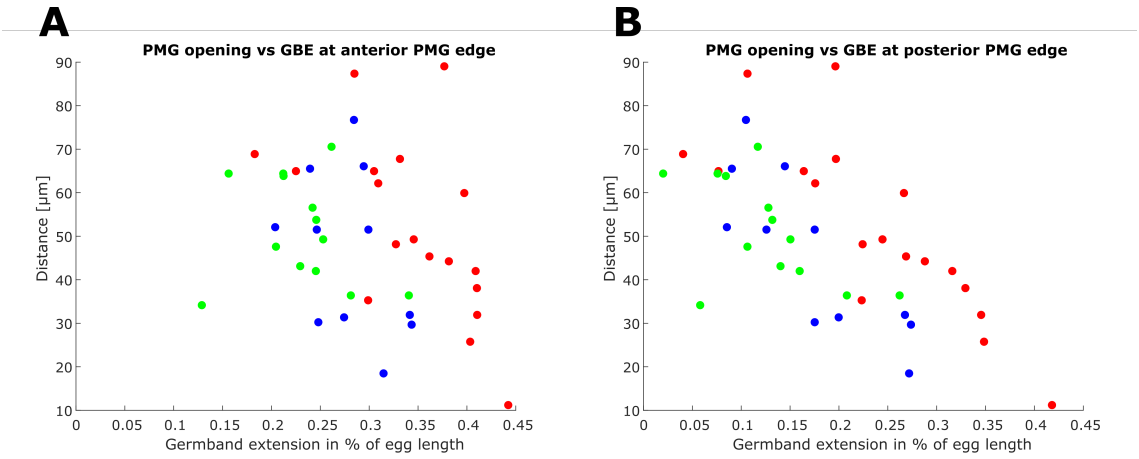


Figure 3.16: **The posterior midgut was affected in a Rapgap1 knockout** Posterior midgut invagination in Rapgap1 mutant *D. melanogaster* embryos was more similar to *D. pseudoobscura* embryos than to *D. melanogaster* wildtypes. There was no difference between an anterior **A** or posterior **B** measurement of germband extension.

4 | Discussion

I showed that the movement in Drosophila embryos appears highly coordinated. The cephalic furrow (CF) and the posterior dorsal fold (DF) were not only present but seemed to be functionally involved in this coordinated movement, possibly for specific tension relaxation. Interestingly both of them originated somewhere in the 250 million years which separate *Drosophila melanogaster* (*D. melanogaster*) and *Chironomus riparius* (*C. riparius*), which might suggest a link between them. However, currently I do not have any evidence for this due to methodological limitations. For example, I would like to knockout the CF, but there is no available way to do this without completely disrupting the whole embryo. In the following I shall discuss the current state, other options, problems, and ideas I have about the cell dynamics and especially fold dynamics in the Drosophila embryo.

4.1 *In toto* analysis

4.1.1 The *in toto* approach requires appropriate imaging technology

The analysis of a single piece of any functional system can reveal the properties of this piece as well as its possible functions. However, the reason why this piece is important and its real function need to be analyzed in the context of the whole, and in the best case running, system. In the case of *D. melanogaster*, a rather global analysis connecting different processes used to be done in earlier days [Turner

and Mahowald, 1977, Wieschaus et al., 1984]. This is not really surprising, as the technology, especially imaging technology, in those days was limited. Although this work was global, it contained rather descriptions of knockout phenotypes and no detailed information about how and why these phenotypes originated, as high speed imaging technology for the entire embryo was simply not available. To get mechanistic insights, this problem was overcome by limiting the region of interest [Rauzi et al., 2010, Wang et al., 2012, Spencer et al., 2015]. However, since the technology of lightsheet microscopy has become available, both approaches were more and more combined [Keller et al., 2008, Tomer et al., 2012, Rauzi et al., 2015]. At the same time, the light sheet microscope technology was also significantly improved, leading to a spatial resolution not far behind confocal resolution [Huisken et al., 2004, Krzic et al., 2012, Liu et al., 2018, Migliori et al., 2018]. This means that by now we are able to have a look at the complete embryo, in a sufficient spatial and temporal resolution to tackle the above raised challenges.

4.1.2 The hardware is outrunning the software

For the reasons listed above, the hardware does not give rise to the the main current challenges. The post processing (registration and fusion of the multiview image data) is one of the current major problems, as the algorithms used either take a very long time or do not produce satisfying results. However, even after the successful registration and fusion of a multiview data set, the challenges do not stop, because simply looking at a movie does not produce any data. Instead the images have to be analyzed. At this point it becomes interesting, as none of the current lighsheet microscopes comes with an out of the box analysis pipeline like confocal microscopes (Volocity, LAS X) and several open source tools including multiple ImageJ plugins either cannot handle 3D+t data, or are simply overstrained by the excessive amount of data. This is one of the main reasons why this thesis focused not only on the biological question, but also on finding suitable ways for the analysis of time lapse lightsheet microscope data.

Current analysis pipelines

For the analysis of cell motion, but also for the analysis of the topology of structures, there are in general two approaches, a manual one and an automated one, and maybe as third option with increased popularity in recent years, a semi-automated one which increases its accuracy via manual data curation [Sommer et al., 2011, Amat et al., 2014]. Automated analysis pipelines are usually fast, and can handle increasing amounts of data, but many of them still lack the required accuracy [Amat et al., 2014, Matula et al., 2015]. This is why a lot of data is still analyzed using labor intensive manual or semi-automated pipelines. For my work I used Particle Image Velocimetry (PIV) as automated analysis and manual cell and furrow tracking as manual analysis. In contrast to tracking pipelines, PIV is not object based, that is to say, it does not work with cells, but only with pixels, voxels or clusters of them, so called supervoxels [Achanta et al., 2012]. This approach circumvents the tracking accuracy problem, but at the cost of lacking cell lineages. Using PIV I was able to show the general velocity and direction for multiple regions in the embryo (figure 3.2). However, Brownian motion like movements of the nuclei prevented a higher resolution which made the analysis of differences in motion vectors of neighboring cells impossible. In addition it was not possible to show the whole trajectory of cells as PIV only considers 2 time points at a time. Of course one can try to overcome the problem of the missing tracks via storing positional and temporal information in the voxels of an image to plot not only the current transition from t_n to t_{n-1} but also plot the last 5 or 10 time points. This solution produces a visualization depicting cells with tails which imitates real tracks. However, this is not accurate and is suitable only for visualization, rather than for analysis purposes. The reason why this is not accurate, is that the cells have to be segmented automatically to enable a PIV analysis in the first place. Current segmentation algorithms usually consider only one image/stack at a time. The main segmentation criterion is still intensity, mainly due to the lack of other criteria. Intensity is however not ideal, as it changes the whole time, for both biological and physical reasons. On the one hand, the fluorescent protein in the nucleus is constantly bleached and regenerated which produces fluctuations in the

amount of emitted light, and on the other hand the amount of received light by the camera depends on the distance of the nucleus to the objective, as well as on the surrounding tissue. As a result the size of segmented nuclei is subject to constant change. Just in theory, if the segmented nuclei had the same shape and size the whole time PIV would allow for cell lineaging. However in this case also automated cell tracking would be 100% accurate, simply because the likelihood of having 2 identical nuclei next to each other is close to zero (mitosis being omitted in this thought experiment).

As I wanted to see the motion dynamics in the whole embryo, and also in the structures and folds (for example, the CF), an automated segmentation dependent analysis was just not good enough, as below the surface of the embryo most cells are lost, whereas close to the microscope objective, nuclei are often aggregated. The only solution to circumvent automated segmentation is a manual annotation of cells, commonly known as manual tracking. Manual tracking or in general manual annotation of cells is also not error free, which is why often several "trackers" work on the same data set, or the manual approach is combined with an automated approach [Muthinja et al., 2017]. As the previous methods were not available, I stuck to certain criteria which were defined based on my tracking experience (see section 3.3). These criteria minimize the error rate, but they can of course not exclude tracking errors like selecting a neighboring cell. These errors likely occur in the region of tissue folds, especially with high invagination speeds. However, tracking errors are often not as bad as they sound, as directly neighboring cells have often very parallel paths (figure 3.3) and end up very close to each other. In general manual tracking allows the tracking of almost every cell in the embryo. Cells that can not be tracked manually will likely never be tracked automatically, as the segmentation done by the human eye and brain is still by magnitudes superior to an automated segmentation, especially if the user is experienced.

Scalability of current analysis pipelines

Although manual tracking sounds from the above paragraph like the best solution to use for cell tracking, there is one major drawback: The workload of manual

tracking is very high. While a surface cell which does not move much, can be tracked in a few minutes, difficult, fast moving cells in regions of low intensity can take 15 minutes and more. From my own experience, I have to admit that there is also a time limit of 2-3 hours a day, especially for difficult cells. This means that it took not weeks, but months, to track the 700 cells for this thesis in 2 embryos. As the embryo has around 6000 cells at blastoderm stage, I tracked only around 5% per embryo for only 2 embryos. Any large scale, high throughput, or statistical analysis of variation between species or in a species, is not possible with these numbers. However, as long as there is no other solution with high accuracy and a universal ability to track high and low intensity regions, manual tracking will still be the option of choice as long as highly accurate data is required.

Enhancements and future analysis

As stated in the above paragraph, there is no ideal solution for cell tracking at the moment. Although semi-automated tracking with guided manual data curation has become very popular [Amat et al., 2014, Haubold et al., 2016], it still is not the solution as it relies on current segmentation algorithms. "We do not have a tracking problem, we rather have a segmentation problem",(unknown) was a sentence I heard once at a conference, which partially explains the problem we are currently facing. However, I would go even further, the problem is not limited to segmentation, it rather includes all the steps which lead to the data set supposed to be segmented or, to put it in general terms, analyzed. If we really want to deliver all the promises that are always associated with SPIM, like a complete *in toto* analysis, all parts of the pipeline leading to the fused image have to be improved:

- Imaging markers have to be improved, for example using new fly lines with markers in the infrared spectrum which have a higher penetration depth and less scattering (personal communication with Dimitri Kromm from the Hufnagel laboratory at EMBL Heidelberg).

- The imaging technology has to be improved to reduce diffraction to achieve

better signals as already done via the use of adaptive optics [Liu et al., 2018] (and personal communication with Dimitri Kromm). This will allow for a better imaging of internalized tissue.

- The segmentation algorithms need to be improved using machine-learning algorithms [Zhang et al., 2014, Amat et al., 2015, Li et al., 2017] to allow for the simultaneous segmentation of very differently labeled cells.

I assume that in a few years the progress in these 3 fields will allow us to reach almost the accuracy level of manual tracking using automated segmentation and (manually curated) tracking algorithms. There are, however, still limits, as cells which a scientist can not even see will always be classified as artifacts although the algorithms might be accurate enough to segment and track them at some point.

4.2 Epithelial integrity

The cells in the epithelium of the Drosophila embryo seemed to move in a very coordinated fashion as a single epithelial sheet without cells passing by their neighbors. This suggests that the fluid properties of an epithelium are, at least for Drosophila, rather to be seen as representation of the whole sheet. The observed order of the single cell tracks and the integrity of the triangles suggest that the cells of the epithelium were strongly attached to each other, rather similar to molecules in a solid object instead of a fluid one. This seemed to be true even in areas of induced neighbor exchange like the ventral area. On the other hand, the observed velocity distributions seemed to fit very much to fluid like properties, as I observed the highest germband extension (GBE) velocities at the posterior pole where the cell flow had to adapt to a reduced diameter. These velocities could be a representation of Bernoulli's law, but they also could simply be the effect of the very strong posterior midgut (PMG) invagination [Collinet et al., 2015]. It is therefore still unclear how similarly to a fluid the Drosophila embryo epithelium acts.

Having an epithelium in the embryo with properties of a complex fluid (liquid with particles) was always a matter of intense discussion in the lab, as it could have

explained tension release in areas of opposing or shearing flow. However, as my data suggested a rather rigid epithelium, it was still unclear how tensions in areas of opposing or shearing flow can be relieved to prevent rupture. In theory the release of tension is possible via cell division [George et al., 2017], however cell division in the gastrulating Drosophila embryo is limited to certain domains [Foe, 1989] and therefore likely of limited influence. Possibly the epithelium, commonly also referred to as an epithelial sheet, could be compared with a sheet of paper, or even better with a sheet of an elastic material to represent the partly elastic properties of an epithelium. Such a sheet could possibly release motion induced tension using its elastic behavior and its ability to fold. Of course this comparison is not ideal, as a sheet of paper or even of an elastic material is probably too rigid to move on the surface of an ellipsoid. The real properties of the Drosophila epithelium might likely be somewhere in between the suggested models. Interestingly, two of the transient folds were in areas where high shear between the moving cells was likely to occur, which is a detail that should be observed more closely.

4.3 The cephalic furrow

The CF is a transient fold which appears at the onset of gastrulation together with the ventral furrow (VF) [Campos-Ortega and Hartenstein, 1997]. However, in contrast to the VF the CF remains part of the epithelium for its whole lifetime, and vanishes without a trace in later development [Ingham, 1994]. In chapter 3.4 I had a closer look at the CF, especially the dynamics in the inside, and pointed out a possible function. Using my triangle analysis (figure 3.11), I was able to investigate the epithelial integrity in the Drosophila embryo (discussed above). I found that certain epithelial regions seemed to be able to move past each other deep inside the CF while the epithelium remained intact. During the tracking of these cells I observed that the epithelium seemed to undergo several internal folding processes leading to tissue faults, which fitted to my other observations, where I saw that the CF had various shapes in different regions of the embryo. These inside folding events could explain why different regions of the epithelium can pass

others while keeping epithelial integrity. This might be one of possibly multiple mechanisms to absorb tension occuring during development or temporarily decouple two tissue regions. In addition, I observed an asymmetric invagination of the CF where the posterior tissue moved below the anterior tissue, possibly creating a certain flexibility in the anterior-posterior (AP) direction, similar to the bumpers of train wagons. It looked like the tissue in the CF had certain folding mediated capabilities, however, it did not become clear whether they were a side effect of the CF, or whether that structure evolved to provide exactly these functions. It might however help to change the point of view a little bit, away from object capabilities to a more needs-focused point of view.

The Drosophila embryo has an ellipsoidal shape; when tissue moves from an anterior or posterior region towards the center of the embryo, the curvature and size of the surface changes. This could likely produce tension. Another phenomenon which likely produces tension is the presence of different velocity vectors in close proximity to one another. In most regions of the embryo the sum of velocity vectors over time, i.e. the trajectory of a cell, changes only very slightly between neighboring cells. However, between the head and the trunk, cells which undergo a rotation or rotation like movement and cells with an almost straight dorso-ventral movement are in almost direct contact. The same is true close to the posterior pole in the region of the posterior DF (figure 3.4). As the epithelium has only a limited ability to reduce these tensions via the remodeling of junctions, which, based on the triangle analysis, seemed not to be extensively used, the embryo could possibly need additional mechanisms to release tension.

The literature does not reveal much about the CF, and nothing about its function. The lack of data facilitates the speculation, considering everything from no function, through a cell storage function, to a rigid barrier preventing the "bad" trunk from entering the head. During my thesis I considered also various ideas, including a function as a ventral or dorsal barrier to prevent the germband from entering the head. As *D. melanogaster* is a well studied model organism, possibly many people have also tried to find the function of the CF, but did not come up with clear results. But why is this the case? Surely one reason is the very lim-

ited genetic background knowledge about the CF, as both *even-skipped* (*eve*) and *buttonhead* (*btd*) are too far upstream and their knockouts produce too many side effects to interpret the results of their knockouts for a possible function of the CF. However, one could knockout the CF using a partial *eve* knockout which either targets only *eve*-stripe 1, or rescues stripes 2-7. This knockout line would allow the analysis of an embryo lacking only the CF without major effects on GBE. As a counterpart, one could analyze embryos lacking *eve*-stripes 2-7, which abolishes GBE but keeps the CF intact. These mutants are currently under development in the laboratory of Yu-Chiun Wang. However, due to a delay in their development, they could not be used for this thesis. Another reason might be the structure of the CF, in which it possibly hides its function. I have shown that deep inside the CF the planar surface structure was lost, and internal folding reshuffled cell clusters. These observations would have not been possible without *in toto* image data with high temporal and spatial resolution with high penetration depth, i.e. not possible without Selective plane illumination microscopy (SPIM) or a similar technology. However, even using this technology, I was not able to get down to the root of the CF. I still lost too many cells while tracking and could not reach the deepest cells. I am however confident that the function of the CF will be solved in the next years, using advanced imaging technology and the above described mutants.

4.4 The posterior dorsal fold

I have shown that, considering the the overall cell motion, the posterior DF is situated in a location very similar to that of the CF. Due to technical reasons, I could not analyze the posterior DF in the same way as the CF, so I could not perform the triangle analysis. However, I could show that cells enter the posterior DF on the dorsal side, and leave it again on the lateral sides of the embryo. Furthermore I could show that a shallow posterior DF coincides with twisted GBE, independent of whether this shallow fold originates from a gene knockout in *D. melanogaster* or as the wild type phenotype of another Drosophila

species. However, with even fewer cells than I had for the CF it is rather difficult to point out the function of the posterior DF. The fact that my data showed a link between Rapgap1 and the PMG activity, and not only to the depth of the posterior DF, causes even more problems, as I could not pinpoint whether the twisted germbands were a result of the altered PMG invagination, or a result of the altered posterior DF. Under the assumption that the twisted germbands originated from the change in posterior DF depth, one could hypothesize that either a possible mechanical decoupling or absorbing function is disturbed, or that a possible pipeline function is disturbed, leading to a cramped dorsal surface and as a result to too much resistance for the extending germband. Using significantly more cell tracks one could possibly determine which of these two directions is correct. However at the same time a way has to be found to analyze whether the twisted germbands were PMG or posterior DF derived. This could be done using mutants which alter PMG invagination. On the basis of the current data set any statement about the function of the posterior DF is pure speculation. Based on experience from the roughly 10 live movies, and the more than 100 fixed images I observed and/or analyzed during my thesis, I assume that the posterior DF is necessary to enable a robust GBE, however I have no clue about the underlying mechanism. The problems described above, likely propelled the fact that the posterior DF is in a very similar situation as the CF. Its function is not known and open for the same speculation as the function of the CF, however with a stronger focus on a possible lack of function. In the lack of function hypothesis the DFs are usually explained as being artefacts of the rather strong forces occuring during GBE [Sonnenblick, 1950]. The 3 transient folds (the anterior DF is not part of this thesis, as it proved to be highly variable and often also absent) have in common that they invaginate during early gastrulation, remain part of the epithelium and vanish in later development without any trace [Campos-Ortega and Hartenstein, 1997]. In addition, the position, initiation and to some extent also invagination of all of them is genetically determined [Vincent et al., 1997, Wang et al., 2012, Wang et al., 2013]

4.5 Proposing a connected system of transient folds

As already mentioned in this thesis, although there are differences between the CF and the posterior DF, they also have much in common which is a reason why I considered the possibility that these folds form a system and might be dependent on each other. One of the main underlying reasons for this assumption was the fact that *C. riparius* does not have any transient fold [Ritter, 1889], while *D. melanogaster* does have them. An interesting difference between those two species is the velocity of GBE, which is around $3 - 4\mu m$ in *C. riparius* [Urbansky, 2016] (and personal communication), while it can reach $12 - 15\mu m$ in Drosophila. Considering the data from my experiments, it seems like fast GBE needs transient folds. This might sound logical in the first place, but there is a difference between the scalar and the direction of a vector. A scalar velocity change would not change the directions of the motion vectors, which would not really change the dynamics of the system as long as the value of the velocity is increased or decreased equally. Conversely, this suggests that not only the value of the velocity changed, but also the direction of the vectors, possibly changed in GBE-driving processes. Indeed looking at SPIM movies from both species seems to reveal some differences. However, as I explained above looking at movies does not help, simply because the amount of information is too high, to be understandable. I suggest that the same analysis done here, should also be done in *C. riparius* to understand the differences in GBE between them. Once we understood those, we can get a better idea of the function of transient folds, and how they ensure robust and fast GBE in Drosophila.

5 | Materials & Methods

5.1 Materials

5.1.1 Flies

LF 034	*Drosophila pseudoobscura*	wt
LF 030	*Drosophila melanogaster*	Dmely[1],oc[R3.2];Gr22b[1],Gr22d[1],cn[1],bw[1],sp[1], LysC[1],lab[R4.2],MstProx[1],GstD5[1],Rh6[1]
LF 145	*Drosophila melanogaster*	Rapgap1 [22]

5.1.2 Reagents

DanKlorix Reiniger Original	DanKlorix 02400170
Ethanol 100%	(EtOH) Sigma, 32212
Formaldehyde	Sigma Aldrich 252549
Glycerol	Sigma-Aldrich 54997
Heptane	Roth, 8654.3
Methanol 100%	Sigma Aldrich 322415
Tween 20	Sigma Aldrich P21379
Triton X-100	Sigma Aldrich 34021300900
Propidium Iodide	Life technologies 1719154

5.1.3 Enzymes

RNAse A	Thermo Scientific EN0531

5.1.4 Buffers, solutions and media

Apple juice plates	Agar Agar	24 g
	Apple juice	250 ml
	H_2O	740 ml
	Saccharose	25 g
Glycerol 50%	Glycerol	25 ml
	PBS	25 ml
Glycerol 75%	Glycerol	37.5 ml
	PBS	12.5 ml
$NaCl$ 40x 28%	H_2O	$1x$
	$NaCl$	280 g/l
PBS (10X) (pH 7.4)	$NaCl$	137 mM
	KCl	2.7 mM
	KH_2PO_4	240 mg/l
	Na_2HPO_4	1.44 g/l
	H_2O	1 l
PBT	PBS	1×
	Tween 20	0.1 %
Propidium Iodide staining solution	PI	20 μl
	H_2O	980 μl
Triton X-100 5%	H_2O	95 ml
	Triton X-100	5 ml

5.1.5 Lab materials

Cover slips	Marienfeld Laboratory Glassware 24x60mm
Cover slips	Roth 18x18mm Article 0657

5.1.6 Provided image data

The two Selective plane illumination microscopy (SPIM) data sets were kindly provided by Lars Hufnagel (EMBL, Heidelberg) and his PhD student Dimitri Kromm. The data set used in all chapters, except for the dorsal fold (DF) analysis (section 3.5) was imaged during early gastrulation of a *Drosophila melanogaster* (*D. melanogaster*) embryo, using a histone coupled mCherry marker [Krzic et al., 2012]. The fused image data shown here is a product of the registration and fusion of 4 individual views [Krzic et al., 2012]. The second data set shows also early gastrulation and was imaged using a nuclear marker in the infrared spectrum (Hufnagel laboratory, unpublished). It consists of 2 opposing views and was fused non-isotropically using the Luxendo fusion and registration pipeline (see section 5.1.7).

5.1.7 Software

Matlab

Matlab (developed by Mathworks) is a high level programming language and Integrated Development Environment (IDE) specialized for numerical computation and matrix operations. It is very powerful as many mathematical operations and algorithms are already integrated, or can be found in open source libraries on the web. For these reasons it was used as main analysis tool for all the numerical data gained from multiple other programs.

Geom3d

Geom3D is a library for 2- and 3 dimensional geometrical calculations in Matlab [Legland, 2016]. It was used to fit a plane to the cephalic furrow (CF), as the native Matlab functions did not allow plane fitting in matrices containing not a number (NaN) values.

ImageJ / Fiji

ImageJ is an open source image analysis software. It was developed by Wayne Rasband at the National Institute of Health (NIH) in the USA. ImageJ was released in 1997 and is based on NIH Image. More than just a tool, ImageJ is a versatile library for image processing and can be extended by a huge number of available third party plugins and macros, making it one of the most powerful tools for scientific image analysis. Fiji is a distribution of ImageJ, which contains a selected and curated set of plugins [Schindelin et al., 2012]. Fiji was used for most of the figures where actual image data are shown. Furthermore it was used for the visualization of optical flow data, and to draw tracks and planes onto image data.

Excel

Excel is a spreadsheet developed by Microsoft. It was used to convert mixed alphanumeric tables, which mainly derived from cell and furrow tracking, into numeric tables, as it provided a more convenient way to do so than Matlab. It was not used for further analysis, as for that purpose Matlab is not only more convenient, but also more powerful.

LAS X

LAS X is a software platform, developed by Leica Microsystems, for acquiring, processing and analyzing confocal microscopy data. It was used to acquire confocal 3D data of fly embryos which were later proecssed in Fiji. In another experiment it was used to acquire and fuse a tile scan from multiple embryos on one slide.

Notepad++

Notepad++ is a free text and source code editor which is able to handle large text files. It was developed by Don Ho. Notepad was especially useful for debugging analysis pipelines of optical flow and tracking data.

Multiview Reconstruction Plugin

The Multiview Reconstruction Plugin is an ImageJ plugin developed by Stephan Preibisch in the lab of Pavel Tomancak [Preibisch et al., 2010]. It is a versatile tool to register and fuse acquisitions of the same object from different angles. The registration can be done using any given interest points, for example fluorescent microspheres or even nuclei.

Luxendo Registration and Fusion software

The Luxendo software for registration and fusion of images is a pipeline based on a software originally developed in the lab of Lars Hufnagel [Krzic et al., 2012]. It allows for bead based, as well as content based or external transformation matrix based registration and fusion of multiple views into one 3D+t data set. To my knowledge it is currently the only pipeline, allowing for a reliable content based opposing fusion of SPIM data sets. It differs from the directly above described ImageJ plugin by the ability to perform a real content based registration, instead of a nucleus base object one. This is an essential advantage, as the nuclei in SPIM data sets often do not have the required quality, to use them as template for an object based registration.

Optical Flow

For determination of the optical flow in the data from the MuVi-SPIM, the optical flow software from the Keller lab was used [Amat et al., 2013]. The advantages were high scalability concerning data size and computing power, as well as robustness.

MTrackJ

MTrackJ is a powerful manual tracking plugin for ImageJ. It was developed by Erik Meijering [Meijering et al., 2012]. Its advantages are full 4D-compatibility, and a structured output of the tracking data, which can be adjusted using the *Set Scale...* and *Properties..* menus of ImageJ. All manual cell tracking data in this

thesis was obtained using MTrackJ. Output analysis was done using Matlab and Excel.

TransformJ and ImageScience packages

This thesis involved large 3D data sets, which need to be looked at from different angles. This can either be done using a 3D viewer such as Chimera or by rotating the image stack. TransformJ and the underlying libraries from the ImageScience package are capable of rotating any type of image data by any angle [Meijering et al., 2001]. The only limit is the available memory of the computer used. Both packages were used on an almost daily basis especially since they are required for furrow tracking.

UCSF Chimera

Originally developed for the visualization of large molecules, mainly proteins, Chimera is also one of the best 3D viewers for image stacks. Chimera is developed by the Resource for Biocomputing, Visualization, and Informatics at the University of California, San Francisco [Pettersen et al., 2004]. Chimera is valuable and was on of the most important programs for this thesis for several reasons:

- Its 3D rendering is very precise, which allows for a good visual inspection of furrows.
- It supports several 3D vision technologies.
- It supports moving inside a hollow object, such as the fly embryo.
- It is currently the only software in this field supporting a *3Dconnexion Space Navigator* which allows for precise 3D navigation by the use of 6 axis.

Own custom developed software

Except for the following *Automated furrow tracking Plugin* (which was not used to produce results for this thesis), my own software can be found in the Appendix B. The detailed functions are described in the Method section.

Automated furrow tracking plugin (did not pass the prototype stage)

Manual tracking of furrows such as the CF or the DFs provides a very high accuracy, but is also very time consuming. However, if this very high accuracy is not needed, an automated approach can perform the same task in a much shorter time. I developed a program which is implemented as an ImageJ plugin and can at least mark the CF with a high enough accuracy for the fit of a plane. The underlying principle of the program is to find stacked cells, a typical characteristic of a furrow in the mainly single layered embryo. These stacked cells are searched by line scans, reaching from one edge of the image to the other edge or only to the center of the embryo. The line scans work with a counter which is increased for every foreground pixel. Background pixels in between foreground pixels, which is a typical sign of stacked cells, generate boni. This technique leads to an exponential increase of the counter in the region of a furrow, when several cells are above each other. These line scans are done in multiple directions, but always on cross sections of the embryo, which is rotated around its anterior-posterior (AP)-axis in 1° steps. In addition to the line scans which detect furrow edges, another line scan is created to detect the continuity of the surface, this is helpful to detect small furrows at the very beginning of gastrulation.

The cross sections are either dorsal ventral, or lateral, depending on the rotation of the embryo. This technique was already used in the manual furrow tracking macro as described above. The rotation was done using java classes from the *imagescience* package [Meijering et al., 2001, Meijering et al., 2012]. As output the plugin marked areas which likely contain a furrow edge. The coordinates of the marked pixels could have been used as template for a plane fit, representing for example the CF. However, as the plugin never reached the accuracy of manually annotated furrow data, its development was truncated.

5.1.8 Hardware

The analysis of large data sets, especially data from a multiview selective plane illumination microscope (MuVi-SPIM), which can have a size of 3 TB can quickly

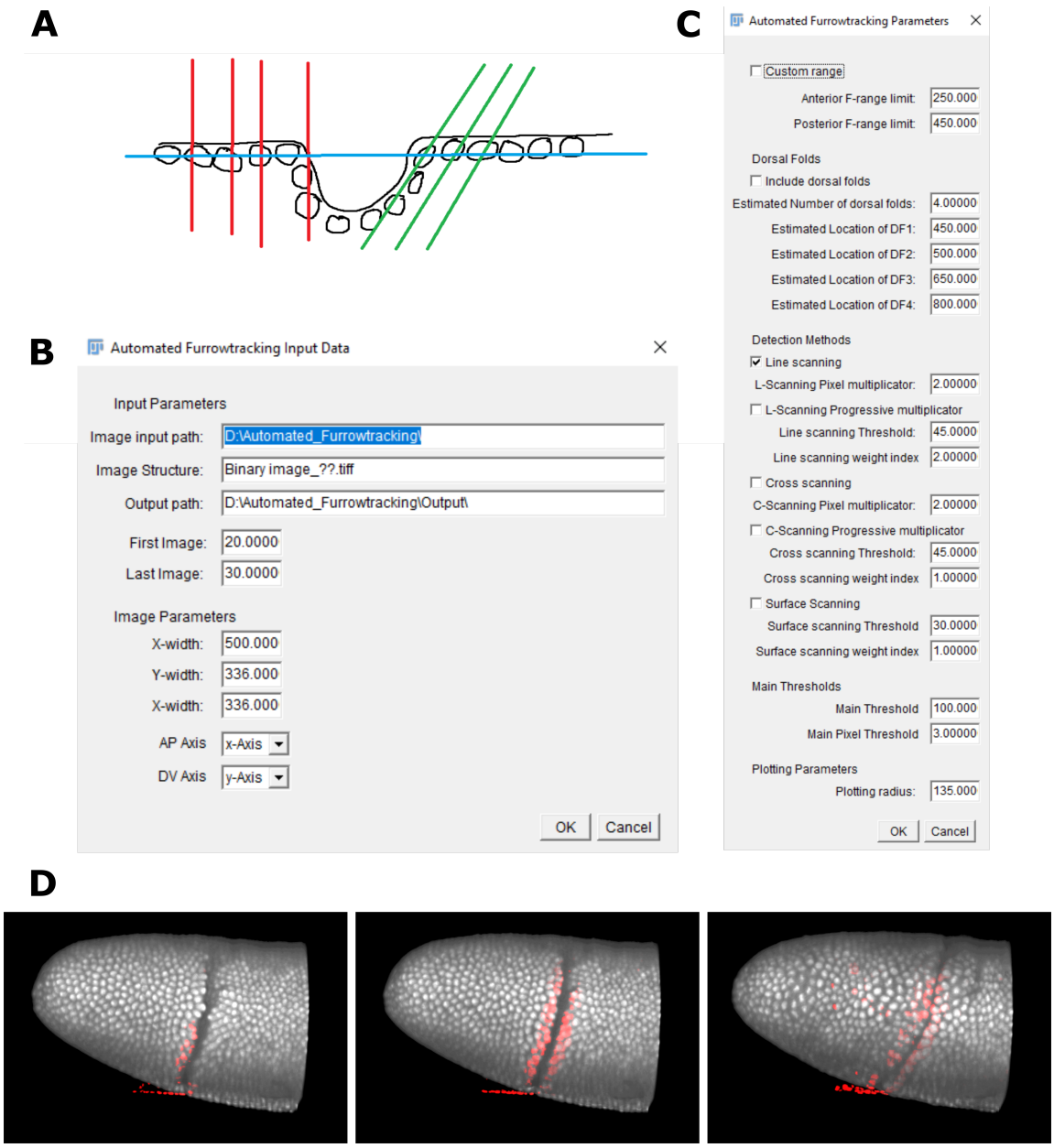


Figure 5.1: **Automated furrow tracking** The ImageJ plugin for automated furrow tracking is based on the principle shown in panel **A**. Line scanners scan a 3D data set with segmented nuclei in multiple directions. Each nucleus within one line increases the scanner score (red and green lines). Stacked nuclei, as present in furrow edges lead to high scores. An additional scanner on the surface finds gaps by comparing distances between the surface nuclei (blue line). The ImageJ plugin consists of 2 windows: in the first window **B**, basic image parameters like paths and dimension have to be entered. In the second window **C** the scanners and their respective parameters have to be defined. The output is shown in panel **D**, the red marks show regions where the likelyhood of a furrow edge was above the chosen threshold.

reach the limits of common IT environments. Also smaller data sets with 30 GB to 50 GB can reach these limits when they have to be loaded at once into the memory required, for example for tracking or 3D-visualization. In addition all this data has to be stored which can range from 10-20 TB for each scientific project, sometimes even more. To overcome these problems a suitable IT environment is necessary. We solved this issue using several machines with different processor, memory and graphic hardware configurations which further provided a total storage space of more than 160 TB distributed on 4 separate RAID-6 systems.

System 1 aka Wolverine

- Operating system: Windows 7
- CPU: 2x Intel Xeon E5620, 4 cores, 2.4 GHz
- Memory: 192 GB
- GPU1: Nvidia Tesla K40c
- GPU2: Nvidia Quadro K620
- Storage: 24 TB RAID 6

System 2 aka Mystique

- Operating system: Ubuntu 14.04 LTS
- CPU: 2x Intel Xeon E5620, 4 cores, 2.4 GHz
- Memory: 96 GB
- GPU: Nvidia Geforce GTX Titan Black
- Storage: 24 TB RAID 6

System 3 aka The Punisher

- Operating system: CentOS 7
- CPU: 2x Intel Xeon E5-2699 v3, 18 cores, 2.3 GHz
- Memory: 256 GB
- GPU: Nvidia Geforce GTX Titan X 12 GB
- Storage: 60 TB RAID 6

System 4 aka Storm

- Operating system: CentOS 7
- CPU: 2x Intel Xeon E5-2699 v4, 22 cores, 2.2 GHz
- Memory: 512 GB
- GPU: Nvidia Geforce GTS 240
- Storage: 56 TB RAID 6

5.1.9 Devices and microscopes

Leica SP8

For imaging fixed samples I used a Leica SP8 confocal microscope with a 20x multi-immersion objective. I used glycerol as the immersion medium.

Leica SP5

To image gastrulation of multiple fly embryos at a time, I used a Leica SP5 with a heating chamber to be able to choose the development temperature.

Multiview selective plane illumination microscope

Currently, the only way to image a fly embryo *in toto* with the spatial and temporal resolution necessary for this project is using a lightsheet microscope as the Multiview selective plane illumination microscope (MuVi-SPIM) [Krzic et al., 2012]. Its design allows for a temporal resolution of below 30 s, while imaging the embryo from four sides. The four individual stacks are later registered and fused into a single 3D volume.

Other devices

Heating block	BIOER Mixing block MA-102
Binocular	Zeiss Stemi 2000-C
Light source	Zeiss CL 1500 Eco

5.2 Methods

5.2.1 Two side imaging of fixed embryos for *in toto* 3D reconstructions

Deposition

Apple Juice plates were taken from the fridge and put on the bench for 30 to 60 minutes, so they could warm up to room temperature. A drop of yeast was added to the center of each plate. When the light switched off in the incubator, the old plate of each fly cage was exchanged with one of the new plates prepared with yeast for deposition. The flies were given 2 hours for deposition, after which the plates were exchanged again with apple juice plates without yeast.

Fixation

The plates which contained the eggs from deposition were briefly cleaned with a brush to remove excessive yeast. 100% bleach was added for roughly 30 s. While bleaching, the embryos were stirred with a brush, and after 30 seconds, they were

collected in a sieve and washed for 3-4 minutes with tap water. Using a brush, the embryos were transferred to a 50 ml falcon tube (one per species or condition). In a second falcon tube the fixation solution was prepared by adding 500 μl of $NaCl$ 40×, 200 μl of a 5% Triton X-100 solution and 19.5 ml H_2O for each set of embryos. The fixation solution was poured into a glass beaker and heated in the microwave. 20 ml of the hot solution was poured over the embryos in each falcon tube. The lids of the tubes were closed, and the tubes repeatedly inverted for 30 s, after which 30 ml of water (room temperature (RT)) was poured in each tube, and the embryos were given time to settle down. The supernatant was removed, and the embryos were collected and transferred to 1.5 ml reaction tubes. The supernatant was removed again, and 500 μl of heptane and subsequently 500 μl of methanol 100% were added to the tube. The lid was closed, and the tube was shaken hard for around 30 s. After shaking the embryos, were given a few minutes to settle down, then the supernatant was removed, and the embryos were washed 3× with methanol 100%, by repeatedly adding 1 ml of methanol and removing it again. After washing a second fixation solution containing 643 μl PBS, 216 μl methanol 100% and 135 μl formaldehyde 37%, was added to the embryos. The tubes were then incubated for 25 minutes at RT on an electric rocking plate. After that, the embryos were again washed 3× with methanol 100% and stored in 1 ml of methanol at $-20°$.

Staining

The Propidium Iodide (PI) staining was based on a protocol from the laboratory of Robert Duronio (University of North Carolina). The mounting part was replaced by the mounting procedure from the DAPI-staining protocol used in the laboratory of Steffen Lemke. The embryos were removed from the freezer, and the methanol covering them was removed. They were washed 3× with 1 ml PBT. Next, the embryos were incubated with RNAse A at a concentration of 400 $\mu g/ml$ (96 μl PBT + 4 μl RNAse A) for 40 minutes at 37° on the shaker at 300 rounds per minute (RPM). After incubation, the embryos were washed 3× with 1 ml PBT. The embryos were then incubated with 1 $\mu g/ml$ PI (25 μl PI staining solution + 475 μl

PBT) for 20 min on an electric rocking plate. From now on, the reaction tube was covered with aluminum foil to prevent the dye from bleaching. Afterwards, the embryos were again washed 3× with 1 ml PBT, however, the last washing step took place for 30 minutes on the electric rocking table. After the last washing step, the PBT was removed, and 1 ml of 50% glycerol was added to the embryos. After 10 minutes when the embryos settled, they were transferred into a new tube with 75% glycerol and stored at 4°.

Mounting and Imaging

Mounting and imaging of fly embryos of the size of Drosophila embryos is challenging because of their size. The light traveling through the embryo is scattered by its matter, which prevents imaging of a whole embryo just from one side. In a lightsheet microscope this problem is efficiently solved by mounting the sample on a rotation capable stage. However, a confocal microscope is not designed for a rotating stage. That is why the rotation has to be done manually, the solution is a symmetrical slide which can be imaged from 2 sides, by simply turning it around with the scientist's hands.

For mounting, symmetrical slides that could be imaged from both sides had to be prepared. These slides were made of 2 24×60 mm cover slips, and 4 18×18 mm cover slips. A drop of superglue was added $9 - 10$ mm inside of both of the short edges of one 24×60 mm cover slip. A 18×18 mm cover slip was placed on each drop, carefully aligning it with the edges of the large cover slip underneath. After the glue fixed the cover slips in position, another drop was added in the middle of each of the 2 18×18 mm cover slips fixed to the 24×60 mm cover slip. The remaining 2 18×18 mm cover slips were now glued on top, again carefully aligning the edges. A drop of $20 - 30$ μl of the glycerol solution was pipetted into the middle of the cover slip construct. It was important to make sure the drop did not contain too many embryos, to avoid any difficulties in finding embryos again when imaging the second side. A maximum number of 50 embryos proved to be good, which meant that not more than 10 embryos were in the right stage for imaging. After the drop was placed, another drop of superglue was added to each

of the 2 upper 18×18 mm cover slips, and the remaining 24×60 mm cover slip was put on top. Due to the viscosity of the glycerol, it was not necessary to seal the open edges of this cover slip construct. They were stored at 4° in the fridge.

For imaging, one of the cover slip constructs was placed on the mounting tray of the microscope, using a 20× multi-immersion objective with glycerol as immersion medium. The resolution was set to 1024×512, which was a good fit for the embryo, and prevented the acquisition of excessive background. Each embryo was fitted into this rectangle using the *field of view rotation* option. The *scan speed* was set to 200 to create high quality images with clearly shaped nuclei (most important for the later image registration process). The resolution of the microscope in the xy direction according to these settings was 0.568 μm, so the z step size was set to 0.56 μm. This way an oversampling in z was performed, which was beneficial for further processing steps. A stack was defined which had to cover around 2/3 of the embryo to provide enough overlap for the registration. For excitation of the dye (PI) the 488-laser was used with depth adjusted intensity (common values: 3% at the surface to 15%-20% deep inside) to provide a similar signal over all slices of the embryo. After the acquisition of the stack, certain parameters of the embryo were noted down, using the following scheme. This was necessary to find the embryo again when imaging from the other side of the slide:

- Stage position at the beginning of the stack in μm
- Stage position at the end of the stack in μm
- Location of the embryo on the slide according to a 360 degree scale (000 = top, 090 = right, 180 = down, 270 = left).
- Radius of the embryo's location according to the upper scale (inside = center of slide, medium = between center and edge, outside = edge of the glycerol drop).
- Orientation of the embryo according to its axes.
- Number of neighbors.

- Characteristics of neighbours (Developmental state, injuries, unfertilized).

This procedure allowed for the imaging of up to 10 embryos per slide, without any trouble in refinding them. Once all the embryos were imaged from one side, the slide was very carefully removed from the slide holder and gently cleaned using a microfiber wipe to remove the glycerol from the outside of the slide. It was very important to reduce any mechanical manipulation of the slide to the least possible amount, to prevent destruction or motion of the mounted embryos. The slide was quickly turned by 180° around its long axis and placed on the slide holder again. The embryos were given 5-10 minutes to sink down again. The progress of this process was checked from time to time in the live mode, when the image did not change anymore, the embryos had reached their final position. Each embryo was imaged again using the above described settings, afterwards the slides were marked as finished. The data was saved a lif file, one per imaged slide.

5.2.2 Computational analysis of fixed embryos for *in toto* 3D reconstructions

Further processing and fusion

The lif files were opended with Fiji [Schindelin et al., 2012] and sorted. One of the 2 stacks of each embryo was turned by 180° in the x or y axis so that the anterior posterior orientation of the embryos in both images matched. This lead to an opposite orientation in the z axis, so that in one image the surface of the embryo was located at the beginning of the z axis and in the other one at the end of the z axis. This process reconstructed a natural 180° rotation. In case background embryos were present in the images, they were removed, as they can disturb the registration process. For each embryo, the 2 images were saved in a separate folder. To register and fuse the data, the Multiview Reconstruction [Preibisch et al., 2010] plugin was used performing the following steps (for detailed parameters see A.0.1):

1. **Define Multi-View Dataset:** This step sets up a new session and gathers the necessary information for further processing

2. **Detect Interest Points for Registration:** The type of image registration used here requires interest points, e.g. fluorescent microspheres or nuclei. In this case nuclei are used, this step sets the parameters and segments the interest points.

3. **Register Dataset based on Interest Points:** In this step the 2 images are registered in 3 dimensional space using a RANSAC algorithm to match the interest points of the 2 images defined in the step before.

4. **Fuse/Deconvolve Dataset** In this step the 2 images are fused into one image using the information from the registration step.

Usually the stitching was seamless and no stitching artifacts were observed. However in up to 10% of the embryos, the fusion failed, which made them unusable for any further analysis. Furthermore, the fusion seemed to be orientation independent, which was beneficial as embryos could be imaged in any orientation.

Fixed furrow tracking

The furrow tracking in the fixed embryos was done using an ImageJ macro originally developed during the time of my master thesis [Schütz, 2014], which was extended to enable tracking of the CF and the anterior and posterior DF (see B.1.1. It uses 7 points per furrow which have to be set manually by the user:

- Points 1 and 7 mark the surface next to the furrow.
- Points 2 and 6 mark the outside edge of the furrow.
- Points 3 and 5 mark the lower edges of the furrow.
- Point 4 marks the deepest spot of the furrow.

Before the first furrow is defined, the following 7 interest points have to be marked once per embryo on a cross section (dorsal to ventral midline and anterior to posterior pole):

- Posterior pole

- Posterior edge of posterior midgut (PMG) invagination
- Anterior edge of PMG invagination
- Posterior dorsal fold
- Anterior dorsal fold
- Cephalic furrow
- Anterior pole

Afterwards the furrows can be tracked in cross sections cutting the embryo into two halves along the AP-axis. After each cross section, the embryo is rotated with a fixed angle. The rotation angle was adjusted to 2°, as this is still high enough. If a furrow was not present at a certain stage, it was skipped.

Analysis of fixed furrows

The output provided from the furrow tracking macro (see 5.2.2 and B.1.1) was a table which was imported into Matlab for further analysis. Using a custom made script in Matlab (see B.2.1), the following parameters were analyzed:

1. Depth of the CF measured from the surface
2. Depth of the CF measured along the edges
3. Opening of the CF
4. Depth of the DFs measured from the surface
5. Depth of the DFs measured along the edges
6. Opening of the DFs
7. Extension of the DFs

Tracking and analysis of the germband

Tracking of the germband for fixed embryos was done using a custom made ImageJ macro and a consecutive Matlab script. In the ImageJ macro (see B.1.2), the AP-axis of the embryo was aligned with the z-axis of the image. Tracking started at the posterior pole in transversal cross sections. The macro automatically fitted a circle to the currently displayed cross section. The user than had to set a point in the center of the dorsal germband (upper image half) and one in the center of the ventral germband (lower image half). The output consisted of a text file for each embryo, which was then read into the Matlab script for germband extension (GBE) analysis (see B.2.3), where the cartesian coordinates from the image were transformed into a cylindrical coordinate system, which allowed for the visualization of the deviation of GBE from the ventral and dorsal midlines.

Embryo parameter analysis

The furrow tracking generated very accurate tracking data which contained significantly more information than was used in the analysis of the furrows. This data was used in the Matlab script for embryo parameter analysis (see B.2.2). The purpose of this script was to compare distances and positions of structures in the embryo using all the available data. The script depended on the sorted matrices from the furrow tracking macro, which had to be executed first. Using these matrices, the following parameters (additional to the ones from the furrow analysis) were calculated for each embryo:

- Furrow extension in dorsal-ventral (DV)-direction for all transient furrows
- Furrow extension in AP-direction for all transient furrows
- Fitted plane
- Furrow planarity
- Furrow angle vs AP-axis

- Single distances between the anterior pole, the posterior pole, the PMG and all transient folds
- Width of the PMG invagination

5.2.3 Analysis of live data

Optical flow analysis

For the analysis of global cell movement I used Optical Flow (OF) (as described above), and 2 custom made ImageJ plugins for visualization of absolute value (scalar velocity) and direction of the generated vectors. The velocity visualization was done using a heatmap which displayed the absolute value of the vectors. This value was multiplied by a constant to fit the range of $0 - 255$ of the 8-bit *Fire* lookup table (LUT) from ImageJ (figure 5.2 C). In order to show only motion along the AP-axis of the embryo, the plugin contained a possibility to limit the axes considered for absolute value calculation. The second plugin used the Spectrum LUT from ImageJ and assigned colors to each direction via vectors in a cylindrical coordinate system fitted to the embryo (figure 5.2 B). Both of these plugins used a stretched cylindrical projection with a given radius to unroll the embryo (figure 5.2 A). This unrolling feature enables a full view on the embryo, displaying all sides simultaneously

Cell tracking

The cell tracking was done in a 4D stack using ImageJ and MtrackJ (see above) [Meijering et al., 2012]. The stack was aligned with the edges of the image and rotated in a way that the z-axis of the image corresponded to the DV axis of the embryo. Cells were then tracked forward or backward in time, some also in both directions. Whether to track forward or backward was usually a question of the analyzed domain of cells. Technically the direction does not matter at all. In each time point, the 3 dimensional center of the nucleus of a cell was marked. Cell divisions were also annotated, but played a minor role, as they usually appeared towards the end of the analyzed time period.

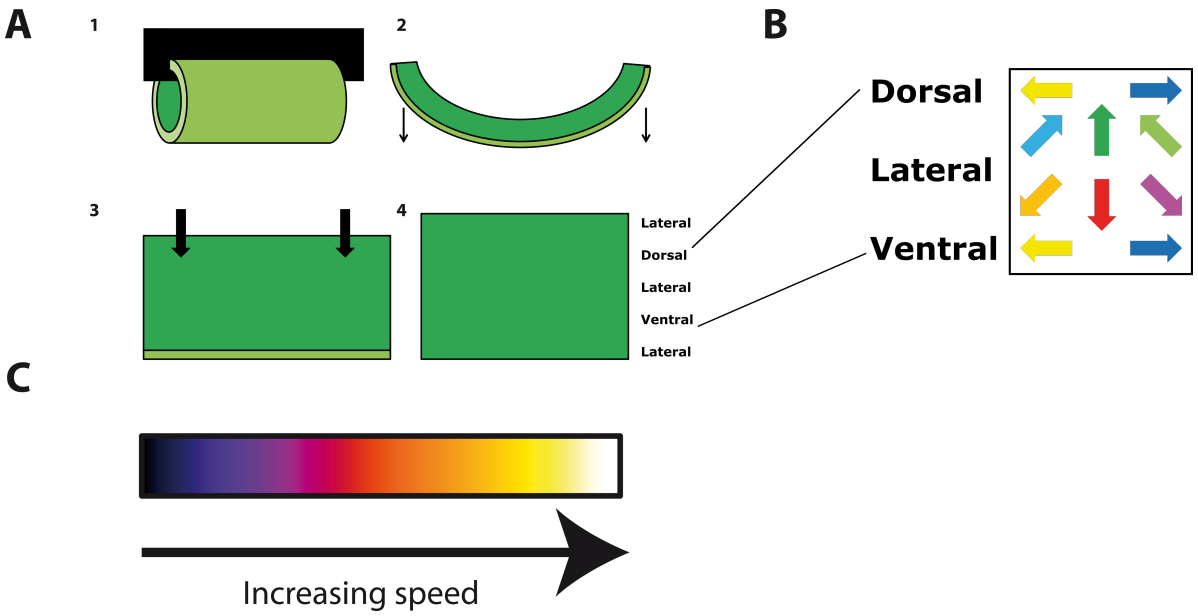


Figure 5.2: **Optical flow visualization methods** The two ImageJ plugins for optical flow visualization unroll the embryo by opening it on one lateral side and using a stretched (fixed radius) cylindrical projection **A**. In a second step either the direction **B**, using a color spectrum look up table, or the speed of the cells using a heatmap look up table are labeled **C**.

Structure tracking

The structure tracking in live data (CF position at dorsal and ventral midline, CF ventral closing and GBE progress) was done with the same pipeline as the cell tracking. However, the information of the z-axis / DV-axis was discarded. Depending on the size and shape of the structure, multiple positions in the DV-axis were used for tracking the same structure.

Track clustering

The cell tracks from the live cell tracking were clustered into groups which allowed the analysis of average motion vectors and parameters. The clustering was done only for parent cells and was based on a visual inspection of origin, trajectory and final location of the cell. Some cells were omitted, as they could not be assigned with enough confidence to a single cluster. This was the case either when they were located in between 2 clusters, or when they were "lost" during tracking

Triangle definition

The triangles were always defined among cells from one cluster, except for the triangles across the CF, which contained 2 cells from the anterior CF cluster and one cell from the posterior one, or the other way round. The triangle definition was done visually, using 3 main criteria: 1. The cell origins should be close to each other 2. The cell tracks should be rather parallel 3. The definition should allow for as many triangles as possible. However, each cell only belonged to one triangle, as an overdefinition just increased the noise, without showing valuable data.

Tracking analysis of live cell tracks in Matlab

The tracking data sets from MTrackJ were exported as text files using μm as length unit for the underlying cartesian coordinate system, as the imaged voxels were, at least for the second embryo, non-isotropic. In a subsequent step, the non-numeric characters were removed in Excel, which drastically simplified the Matlab pipeline. The processed text files were read by a custom made Matlab script (see

B.2.4) and processed in the following way: At first, the tracks were sorted into 3D-matrices along the following dimensions:

1. Time
2. Parameters
3. Cell or structure IDs

In a second step, additional parameters that had not been obtained from the tracking software were calculated.

- Step length
- Velocity (scalar)
- Average velocity over 3 timepoints (scalar)
- Acceleration
- x-Velocity (vector)
- Averaged x-Velocity over 3 timepoints (vector)
- x-Acceleration
- Averaged x-Acceleration over 3 timepoints
- dv-Velocity (scalar)
- Averaged dv-Velocity over 3 timepoints (scalar)
- dv-Velocity (vector)
- Averaged dv-Velocity over 3 timepoints (vector)
- Averaged acceleration over 3 timepoints

After these parameters were calculated, the tracked points of each cell were sorted according to time, to allow for the calculation of average and triangle parameters. The triangles' matrices were defined by setting the points A,B,C for each time point. Based on these points all other triangle parameters, plus their averages among all triangles of one cluster for each timepoint were calculated:

- $|a|$
- $|b|$
- $|c|$
- α
- β
- γ
- height a
- area A
- circumscribed circle diameter

In the following step averages for selected parameters were calculated for each time point among all cells of one cluster:

- Average velocity (scalar)
- Average AP-velocity (vector)
- Average DV-velocity (scalar)
- Average DV-velocity (vector)
- Average track length

After the calculation of these parameters, the remaining task of the script was plotting.

Contributions

Fly work: The maintenance of fly cultures and the preparation of cages for deposition was done by Maike Wosch. Sometimes the plates for deposition were also changed by her.
Stock solution and materials: Stock solutions and common materials such as apple juice plates were prepared by Maike Wosch.
Lightsheet Microscopy: The *D. melanogaster* SPIM movies were provided by the laboratory of Lars Hufnagel at the European Molecular Biology Laboratory (EMBL). The second Drosophila data set was imaged by Dimitri Kromm.
***D. melanogaster* Rapgap1 knockout line:** The *D. melanogaster* Rapgap1 knockout line was kindly provided by the laboratory of Yu-Chiun Wang from the Riken Center for Developmental Bioloogy, Kobe, Japan.

Bibliography

[Achanta et al., 2012] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2281.

[Allena and Aubry, 2012] Allena, R. and Aubry, D. (2012). An extensive numerical simulation of the cephalic furrow formation in Drosophila embryo. *Computer Methods in Biomechanics and Biomedical Engineering*, 15(5):445–455.

[Allena et al., 2010] Allena, R., Mouronval, A.-S., and Aubry, D. (2010). Simulation of multiple morphogenetic movements in the Drosophila embryo by a single 3d finite element model. *Journal of the Mechanical Behavior of Biomedical Materials*, 3(4):313–323.

[Amat et al., 2015] Amat, F., Höckendorf, B., Wan, Y., Lemon, W. C., McDole, K., and Keller, P. J. (2015). Efficient processing and analysis of large-scale light-sheet microscopy data. *Nature Protocols*, 10(11):1679–1696.

[Amat et al., 2014] Amat, F., Lemon, W., Mossing, D. P., McDole, K., Wan, Y., Branson, K., Myers, E. W., and Keller, P. J. (2014). Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data. *Nature Methods*, 11(9):951–958.

[Amat et al., 2013] Amat, F., Myers, E. W., and Keller, P. J. (2013). Fast and

robust optical flow for time-lapse microscopy using super-voxels. *Bioinformatics*, 29(3):373–380.

[Bertet et al., 2004] Bertet, C., Sulak, L., and Lecuit, T. (2004). Myosin-dependent junction remodelling controls planar cell intercalation and axis elongation. *Nature*, 429(6992):nature02590.

[Blanch-Mercader et al., 2017] Blanch-Mercader, C., Vincent, R., Bazellières, E., Serra-Picamal, X., Trepat, X., and Casademunt, J. (2017). Effective viscosity and dynamics of spreading epithelia: a solvable model. *Soft Matter*, 13(6):1235–1243.

[Blankenship et al., 2006] Blankenship, J. T., Backovic, S. T., Sanny, J., Weitz, O., and Zallen, J. A. (2006). Multicellular rosette formation links planar cell polarity to tissue morphogenesis. *Developmental Cell*, 11(4):459–470.

[Butler et al., 2009] Butler, L. C., Blanchard, G. B., Kabla, A. J., Lawrence, N. J., Welchman, D. P., Mahadevan, L., Adams, R. J., and Sanson, B. (2009). Cell shape changes indicate a role for extrinsic tensile forces in Drosophila germ-band extension. *Nature Cell Biology*, 11(7):859–864.

[Campos-Ortega and Hartenstein, 1997] Campos-Ortega, J. A. and Hartenstein, V. (1997). The embryonic development of Drosophila melanogaster.

[Chen et al., 1997] Chen, F., Barkett, M., Ram, K. T., Quintanilla, A., and Hariharan, I. K. (1997). Biological characterization of Drosophila Rapgap1, a GTPase activating protein for Rap1. *Proceedings of the National Academy of Sciences*, 94(23):12485–12490.

[Clément et al., 2017] Clément, R., Dehapiot, B., Collinet, C., Lecuit, T., and Lenne, P.-F. (2017). Viscoelastic dissipation stabilizes cell shape changes during tissue morphogenesis. *Current Biology*, 27(20):3132–3142.e4.

[Collinet et al., 2015] Collinet, C., Rauzi, M., Lenne, P.-F., and Lecuit, T. (2015). Local and tissue-scale forces drive oriented junction growth during tissue extension. *Nature Cell Biology*, 17(10):ncb3226.

[Curran et al., 2017] Curran, S., Strandkvist, C., Bathmann, J., Gennes, M. d., Kabla, A., Salbreux, G., and Baum, B. (2017). Myosin ii controls junction fluctuations to guide epithelial tissue ordering. *Developmental Cell*.

[Delile et al., 2017] Delile, J., Herrmann, M., Peyriéras, N., and Doursat, R. (2017). A cell-based computational model of early embryogenesis coupling mechanical behaviour and gene regulation. *Nature Communications*, 8:13929.

[Desprat et al., 2008] Desprat, N., Supatto, W., Pouille, P.-A., Beaurepaire, E., and Farge, E. (2008). Tissue deformation modulates twist expression to determine anterior midgut differentiation in Drosophila embryos. *Developmental Cell*, 15(3):470–477.

[Dicko et al., 2017] Dicko, M., Saramito, P., Blanchard, G. B., Lye, C. M., Sanson, B., and Étienne, J. (2017). Geometry can provide long-range mechanical guidance for embryogenesis. *PLOS Computational Biology*, 13(3):e1005443.

[Foe, 1989] Foe, V. E. (1989). Mitotic domains reveal early commitment of cells in Drosophila embryos. *Development (Cambridge, England)*, 107(1):1–22.

[Foe and Alberts, 1983] Foe, V. E. and Alberts, B. M. (1983). Studies of nuclear and cytoplasmic behaviour during the five mitotic cycles that precede gastrulation in Drosophila embryogenesis. *Journal of cell science*, 61:31–70.

[George et al., 2017] George, M., Bullo, F., and Campàs, O. (2017). Connecting individual to collective cell migration. *Scientific Reports*, 7(1):9720.

[Gilbert and Barresi, 2016] Gilbert, S. F. and Barresi, M. J. F. (2016). *Developmental biology*. Sinauer Associates, Inc., Publishers, Sunderland, Massachusetts U.S.A., eleventh edition edition.

[Goldman-Levi et al., 1996] Goldman-Levi, R., Miller, C., Greenberg, G., Gabai, E., and Zak, N. B. (1996). Cellular pathways acting along the germband and in the amnioserosa may participate in germband retraction of the Drosophila melanogaster embryo. *The International journal of developmental biology*, 40(5):1043–51.

[Grimaldi and Engel, 2005] Grimaldi, D. and Engel, M. S. (2005). *Evolution of the Insects*.

[Gumbiner, 1996] Gumbiner, B. M. (1996). Cell adhesion: The molecular basis of tissue architecture and morphogenesis. *Cell*, 84(3):345–357.

[Haubold et al., 2016] Haubold, C., Schiegg, M., Kreshuk, A., Berg, S., Koethe, U., and Hamprecht, F. A. (2016). *Focus on Bio-Image Informatics*, volume 219. Advances in anatomy, embryology, and cell biology.

[He et al., 2014] He, B., Doubrovinski, K., Polyakov, O., and Wieschaus, E. (2014). Apical constriction drives tissue-scale hydrodynamic flow to mediate cell elongation. *Nature*, 508(7496):392–396.

[Heisenberg, 2009] Heisenberg, C. (2009). Dorsal closure in Drosophila: cells cannot get out of the tight spot. *BioEssays*, 31(12):1284–1287.

[Hughes and Kaufman, 2002] Hughes, C. L. and Kaufman, T. C. (2002). Hox genes and the evolution of the arthropod body plan. *Evolution and Development*, 4(6):459–499.

[Huisken et al., 2004] Huisken, J., Swoger, J., Bene, F. D., Wittbrodt, J., and Stelzer, E. H. K. (2004). Optical sectioning deep inside live embryos by selective plane illumination microscopy. *Science*, 305(5686):1007–1009.

[Ingham, 1994] Ingham, P. (1994). *The development of Drosophila melanogaster. Edited by Michael Bate and Alfonso Martinez Arias*, volume 10. Trends in Genetics.

[Irvine and Wieschaus, 1994] Irvine, K. D. and Wieschaus, E. (1994). Cell intercalation during Drosophila germband extension and its regulation by pair-rule segmentation genes. *Development (Cambridge, England)*, 120(4):827–41.

[Jacinto et al., 2002] Jacinto, A., Woolner, S., and Martin, P. (2002). Dynamic analysis of dorsal closure in Drosophila: From genetics to cell biology. *Developmental Cell*, 3(1):9–19.

[Jennings, 2011] Jennings, B. H. (2011). Drosophila - a versatile model in biology and medicine. *Materials Today*, 14(5):190–195.

[Kam et al., 1991] Kam, Z., Minden, J. S., Agard, D. A., Sedat, J. W., and Leptin, M. (1991). Drosophila gastrulation: analysis of cell shape changes in living embryos by three-dimensional fluorescence microscopy. *Development (Cambridge, England)*, 112(2):365–70.

[Keller et al., 2008] Keller, P. J., Schmidt, A. D., Wittbrodt, J., and Stelzer, E. H. (2008). Reconstruction of Zebrafish early embryonic development by scanned light sheet microscopy. *Science*, 322(5904):1065–1069.

[Kong et al., 2017] Kong, D., Wolf, F., and Großhans, J. (2017). Forces directing germ-band extension in Drosophila embryos. *Mechanisms of Development*, 144:11–22.

[Krzic et al., 2012] Krzic, U., Gunther, S., Saunders, T. E., Streichan, S. J., and Hufnagel, L. (2012). Multiview light-sheet microscope for rapid in toto imaging. *Nature Methods*, 9(7):730–733.

[Kuntz and Eisen, 2014] Kuntz, S. G. and Eisen, M. B. (2014). Drosophila embryogenesis scales uniformly across temperature in developmentally diverse species. *PLoS Genetics*, 10(4):e1004293.

[Lacy and Hutson, 2016] Lacy, M. E. and Hutson, M. S. (2016). Amnioserosa development and function in Drosophila embryogenesis: Critical mechanical roles for an extraembryonic tissue. *Developmental Dynamics*, 245(5):558–568.

[Legland, 2016] Legland, D. (2016). geom3d: A Matlab library to handle 3d geometric primitives: create, intersect, display, and make basic computations.

[Leptin et al., 1992] Leptin, M., Casal, J., Grunewald, B., and Reuter, R. (1992). Mechanisms of early Drosophila mesoderm formation. *Development (Cambridge, England). Supplement*, pages 23–31.

[Leptin and Grunewald, 1990] Leptin, M. and Grunewald, B. (1990). Cell shape changes during gastrulation in Drosophila. *Development (Cambridge, England)*, 110(1):73–84.

[Li et al., 2017] Li, R., Zeng, T., Peng, H., and Ji, S. (2017). Deep learning segmentation of optical microscopy images improves 3-d neuron reconstruction. *IEEE Transactions on Medical Imaging*, 36(7):1533–1541.

[Liu et al., 2018] Liu, T.-L., Upadhyayula, S., Milkie, D. E., Singh, V., Wang, K., Swinburne, I. A., Mosaliganti, K. R., Collins, Z. M., Hiscock, T. W., Shea, J., Kohrman, A. Q., Medwig, T. N., Dambournet, D., Forster, R., Cunniff, B., Ruan, Y., Yashiro, H., Scholpp, S., Meyerowitz, E. M., Hockemeyer, D., Drubin, D. G., Martin, B. L., Matus, D. Q., Koyama, M., Megason, S. G., Kirchhausen, T., and Betzig, E. (2018). Observing the cell in its native state: Imaging subcellular dynamics in multicellular organisms. *Science*, 360(6386):eaaq1392.

[Lye et al., 2015] Lye, C. M., Blanchard, G. B., Naylor, H. W., Muresan, L., Huisken, J., Adams, R. J., and Sanson, B. (2015). Mechanical coupling between endoderm invagination and axis extension in Drosophila. *PLOS Biology*, 13(11):e1002292.

[Lye and Sanson, 2011] Lye, C. M. and Sanson, B. (2011). Current topics in developmental biology. *Current topics in developmental biology*, 95:145–187.

[Martin et al., 2008] Martin, A. C., Kaschube, M., and Wieschaus, E. F. (2008). Pulsed contractions of an actin-myosin network drive apical constriction. *Nature*, 457(7228):495–499.

[Matula et al., 2015] Matula, P., Maška, M., Sorokin, D. V., Matula, P., Ortiz-de Solórzano, C., and Kozubek, M. (2015). Cell tracking accuracy measurement based on comparison of acyclic oriented graphs. *PLOS ONE*, 10(12):e0144959.

[Meijering et al., 2012] Meijering, E., Dzyubachyk, O., and Smal, I. (2012). Methods in enzymology. *Methods in enzymology*, 504:183–200.

[Meijering et al., 2001] Meijering, E. H., Niessen, W. J., and Viergever, M. A. (2001). Quantitative evaluation of convolution-based methods for medical image interpolation. *Medical Image Analysis*, 5(2):111–126.

[Migliori et al., 2018] Migliori, B., Datta, M. S., Dupre, C., Apak, M. C., Asano, S., Gao, R., Boyden, E. S., Hermanson, O., Yuste, R., and Tomer, R. (2018). Light sheet theta microscopy for rapid high-resolution imaging of large biological samples. *BMC Biology*, 16(1):57.

[Momen-Roknabadi et al., 2016] Momen-Roknabadi, A., Di Talia, S., and Wieschaus, E. (2016). Transcriptional timers regulating mitosis in early Drosophila embryos. *Cell Reports*, 16(11):2793–2801.

[Muthinja et al., 2017] Muthinja, M. J., Ripp, J., Hellmann, J. K., Haraszti, T., Dahan, N., Lemgruber, L., Battista, A., Schütz, L., Fackler, O. T., Schwarz, U. S., Spatz, J. P., and Frischknecht, F. (2017). Microstructured blood vessel surrogates reveal structural tropism of motile malaria parasites. *Advanced Healthcare Materials*, 6(6):1601178.

[Parks and Wieschaus, 1991] Parks, S. and Wieschaus, E. (1991). The Drosophila gastrulation gene Concertina encodes a Gα-like protein. *Cell*, 64(2):447–458.

[Pettersen et al., 2004] Pettersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C., and Ferrin, T. E. (2004). UCSF Chimera-a visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 25(13):1605–1612.

[Pope and Harris, 2008] Pope, K. L. and Harris, T. J. C. (2008). Control of cell flattening and junctional remodeling during squamous epithelial morphogenesis in Drosophila. *Development*, 135(13):2227–2238.

[Poulson, 1950] Poulson, D. F. (1950). Histogenesis, organogenesis, and differentiation in the embryo of Drosophila melanogaster. *Biology of Drosophila*, pages 168–274.

[Preibisch et al., 2010] Preibisch, S., Saalfeld, S., Schindelin, J., and Tomancak, P. (2010). Software for bead-based registration of selective plane illumination microscopy data. *Nature Methods*, 7(6):nmeth0610–418.

[Rabinowitz, 1941] Rabinowitz, M. (1941). Studies on the cytology and early embryology of the egg of Drosophila melanogaster. *Journal of Morphology*, 69(1):1–49.

[Rauzi et al., 2015] Rauzi, M., Krzic, U., Saunders, T. E., Krajnc, M., Ziherl, P., Hufnagel, L., and Leptin, M. (2015). Embryo-scale tissue mechanics during Drosophila gastrulation movements. *Nature Communications*, 6:ncomms9677.

[Rauzi et al., 2010] Rauzi, M., Lenne, P.-F., and Lecuit, T. (2010). Planar polarized actomyosin contractile flows control epithelial junction remodelling. *Nature*, 468(7327):1110–1114.

[Ritter, 1889] Ritter, R. (1889). Die Entwicklung der Geschlechtsorgane und des Darmes bei Chironomus.

[Schindelin et al., 2012] Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J.-Y., White, D. J., Hartenstein, V., Eliceiri, K., Tomancak, P., and Cardona, A. (2012). Fiji: an open-source platform for biological-image analysis. *Nature Methods*, 9(7):676–682.

[Schmidt-Ott et al., 2010] Schmidt-Ott, U., Rafiqi, A. M., and Lemke, S. (2010). Hox3/zen and the evolution of extraembryonic epithelia in insects. *Advances in experimental medicine and biology*, 689:133–44.

[Schöck and Perrimon, 2002] Schöck, F. and Perrimon, N. (2002). Cellular processes associated with germ band retraction in Drosophila. *Developmental Biology*, 248(1):29–39.

[Schütz, 2014] Schütz, L. C. (2014). Analysis of cephalic furrow movement during germband extension in the Drosophila pseudoobscura embryo.

[Schweisguth et al., 1991] Schweisguth, F., Vincent, A., and Lepesant, J. (1991). Genetic analysis of the cellularization of the Drosophila embryo. *Biology of the Cell*, 72(1-2):15–23.

[Sommer et al., 2011] Sommer, C., Straehle, C., Kothe, U., and Hamprecht, F. A. (2011). Ilastik: Interactive learning and segmentation toolkit. *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 1:230–233.

[Sonnenblick, 1950] Sonnenblick, B. P. (1950). The early embryology of Drosophila melanogaster. *Biology of Drosophila*, pages 62–167.

[Spencer et al., 2015] Spencer, A. K., Siddiqui, B. A., and Thomas, J. H. (2015). Cell shape change and invagination of the cephalic furrow involves reorganization of f-actin. *Developmental Biology*, 402(2):192–207.

[Sweeton et al., 1991] Sweeton, D., Parks, S., Costa, M., and Wieschaus, E. (1991). Gastrulation in Drosophila: the formation of the ventral furrow and posterior midgut invaginations. *Development (Cambridge, England)*, 112(3):775–89.

[Takeichi, 1991] Takeichi, M. (1991). Cadherin cell adhesion receptors as a morphogenetic regulator. *Science*, 251(5000):1451–1455.

[Technau and Campos-Ortega, 1985] Technau, G. M. and Campos-Ortega, J. A. (1985). Fate-mapping in wild-type Drosophila melanogaster. *Wilhelm Roux's Archives of Developmental Biology*, 194(4):196–212.

[Thiery et al., 1988] Thiery, J. P., Boyer, B., Tucker, G., Gavrilovic, J., and Valles, A. M. (1988). Adhesion mechanisms in embryogenesis and in cancer invasion and metastasis. *Ciba Foundation symposium*, 141:48–74.

[Thisse et al., 1988] Thisse, B., Stoetzel, C., Gorostiza-Thisse, C., and Perrin-Schmitt, F. (1988). Sequence of the twist gene and nuclear localization of its protein in endomesodermal cells of early Drosophila embryos. *The EMBO journal*, 7(7):2175–83.

[Tomer et al., 2012] Tomer, R., Khairy, K., Amat, F., and Keller, P. J. (2012). Quantitative high-speed imaging of entire developing embryos with simultaneous multiview light-sheet microscopy. *Nature Methods*, 9(7):nmeth.2062.

[Turner and Mahowald, 1977] Turner, F. and Mahowald, A. P. (1977). Scanning electron microscopy of Drosophila melanogaster embryogenesis ii. gastrulation and segmentation. *Developmental Biology*, 57(2):403–416.

[Underwood et al., 1980] Underwood, E., Turner, F., and Mahowald, A. (1980). Analysis of cell movements and fate mapping during early embryogenesis in Drosophila melanogaster. *Developmental Biology*, 74(2):286–301.

[Urbansky, 2016] Urbansky, S. (2016). Turning the switch: fog and t48 control the mode of mesoderm internalization in flies.

[Vincent et al., 1997] Vincent, A., Blankenship, J. T., and Wieschaus, E. (1997). Integration of the head and trunk segmentation systems controls cephalic furrow formation in drosophila. *Development (Cambridge, England)*, 124(19):3747–54.

[Wang et al., 2012] Wang, Y.-C., Khan, Z., Kaschube, M., and Wieschaus, E. F. (2012). Differential positioning of adherens junctions is associated with initiation of epithelial folding. *Nature*, 484(7394):390–393.

[Wang et al., 2013] Wang, Y.-C., Khan, Z., and Wieschaus, E. (2013). Distinct Rap1 activity states control the extent of epithelial invagination via α-catenin. *Developmental Cell*, 25(3):299–309.

[Warn and Robert-Nicoud, 1990] Warn, R. M. and Robert-Nicoud, M. (1990). F-actin organization during the cellularization of the Drosophila embryo as revealed with a confocal laser scanning microscope. *Journal of cell science*, 96 (Pt 1):35–42.

[Wieschaus et al., 1984] Wieschaus, E., Nüsslein-Volhard, C., and Kluding, H. (1984). Krüppel, a gene whose activity is required early in the zygotic genome for normal embryonic segmentation. *Developmental Biology*, 104(1):172–186.

[Zallen and Blankenship, 2008] Zallen, J. A. and Blankenship, J. T. (2008). Multicellular dynamics during epithelial elongation. *Seminars in Cell and Developmental Biology*, 19(3):263–270.

[Zalokar and Erk, 1976] Zalokar, M. and Erk, I. (1976). Division and migration of nuclei during early embryogenesis of Drosophila melanogaster. *Journal de Microscopie et de Biologie Cellulaire*, 25(2):97–108.

[Zhang et al., 2014] Zhang, C., Yarkony, J., and Hamprecht, F. A. (2014). Cell detection and segmentation using correlation clustering. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 17(Pt 1):9–16.

[Zusman and Wieschaus, 1985] Zusman, S. B. and Wieschaus, E. F. (1985). Requirements for zygotic gene activity during gastrulation in Drosophila melanogaster. *Developmental Biology*, 111(2):359–371.

List of Figures

Appendices

A | Protocols

A.0.1 Parameters for registration and fusion

This protocol contains the detailed parameters and steps that worked to fuse 90% of the Drosophila embryos:

- Select *Plugins >Multiview Reconstruction >Batch processing >Define Multi-View Dataset*
- Select *Image Stacks (ImageJ Opener)* as type of data set.
- Name the data set and click OK
- In the next window, select
  - Multiple Angles: Yes (One file per angle).
- In the next dialog, enter the directory of your files, and the filename, the angle has to be substituted by aaa
- Enter 0,180 as Acquisition Angles
- Proceed with the following settings and click OK:
  - Calibration Type: Same voxel-size for all views
  - Calibration Definition: User define voxel-size(s)
  - ImagLib2 data container: ArrayImg (faster)
  - Tick Show list of filenames

- If all files are recognized, continue with the next step, if not find the error, and repeat the procedure.
- Enter the voxel size in x,y,z direction.
- Go to Plugins>Multiview Reconstruction>Batch Processing>Detect Interest Points for Registration
- Click OK in the open dialog
- In the next dialog, change the Type of interest point detection to Difference-of-Mean
- Type nuclei in the field Label interest points and click OK.
- In the next window, change the field Interest point specification to AdvancedâĂę and click on OK
- In the next dialog, set Radius 1 to 3 and Radius 2 to 10 and click OK
- The software should find between 3000 and 6000 interest points per image.
- If this step was successful continue with the next step, otherwise try to change the settings.
- Go to Plugins>Multiview reconstruction>Batch Processing>Register Dataset based on Interest Points
- In the first window, make sure the Select XML text field shwos the right path to your XML file, then click OK.
- In the window Basic Registration Parameters, make sure that Registration algorithm is set to Fast 3d geometric hashing (rotation invariant) and click OK.
- The next window shows more detailed registration parameters, for the first run, it is not recommended to change anything in this window. However, a non-successful registration can make changes necessary. Click on OK.

- The same hold true for the next window, where the parameter Lambda has to be set.

- Changes might only be necessary, if the registration was not successful. Click OK to start the registration.

- The log window shows the progress of the registration process (<5 s). Once the registration is completed successfully, scroll up in the Log window to find the following line: (Wed Dec 09 15:18:56 CET 2015): [TP=0 angle=0, ch=0, illum=0 »> TP=0 angle=180, ch=0, illum=0]: Remaining inliers after RANSAC: 104 of 113 (92%) with average error 1.3099617057866355

- The remaining inliers after RANSAC should be more than 30, the software needs only 12, however, anything below 30 is likely to give strange results during the fusion.

- If the fusion was not successful, check the orientation of the embryos (180Âř rotation), or change the fusion parameters, however changing the fusion parameters often just allows for a higher error tolerance in registration, resulting in a positive registration, but in an akward fusion.

- If the fusion was successful continue with Plugins>Multiview Reconstruction>Batch Processing>Fuse/ Deconvolve dataset.

- In the first window, make sure the Select XML text field shows the right path to your XML file, then click OK.

- The window Image Fusion pops up. Select Weighted-average fusion as Type of image fusion; select Define manually for Bounding Box and Display using ImageJ for Fused image, then click OK.

- In the next window move all Minimal and Maximal sliders to the respective minimum or maximum.

- The slider Downsample fused dataset should be set to 1.

- The Pixel type can be set to 16-bit unsigned integer, as 32 bits are not necessary, and not as convenient as 16 bits.
- The other settings do not need to be changed,default values should be:
- IMGLib2 container: CellImg (large images).
- Process views in parallel: All.
- Blend images smoothly: True.
- Conten-based fusion: False.
- Interpolation: Linear Interpolation.
- Start the fusion by clicking OK.

B | Scripts

The scripts used in this thesis were either written in Java, the ImageJ macro language or in Matlab, depending on their purpose. In general Matlab scripts were used for numerical analysis and visualizations of graphs, while ImageJ macros or plugins (Java) were used for image visualization. Scripts above 1000 lines were left out for size reasons, but can be found on the attached CD.

B.1 ImageJ plugins and macros

B.1.1 Furrow tracking macro

Filename: *CF-DF-Tracking_Intoto_Fixation.ijm* This ImageJ macro served as tracking program for the CF and the anterior as well as posterior DF. The way it works is explained in 5.2.2.

```
1  //************************Furrow macro*****************************************************
2  /*
3   * Set 7 Points per Furrow for a maximum of 3 Furrows
4   on both sides.
5   * ************--1----2           6----7--
6   *                   -              -
7   *                   -              -
8   *                   -              -
9   *                   3-          -5
10  *                       ----- -----
11  *                            -4-
12  * Use "alt" to skip a furrow and "space" to continue with the next timepoint (only if not all
        42 points are set)
13  * Use "shift" for deleting the last point set.
14  *
15  * The center of all (set) previous (t-1) tracked furrows will be indicated by a dot to keep
        track of volatile furrows.
16 */
17
```

```
18 //************** Variable initialization ********************************
19 flag = 0;
20 setColor(0,255,0);
21
22 dmX = newArray(7);
23 dmY = newArray(7);
24 dmZ = newArray(7);
25
26 X = newArray(42);
27 Y = newArray(42);
28 Z = newArray(42);
29 PFx = newArray(10);
30 PFy = newArray(10);
31 Array.fill(X,NaN);
32 Array.fill(Y,NaN);
33 Array.fill(Z,NaN);
34 Array.fill(PFx,NaN);
35 Array.fill(PFy,NaN);
36 i = 0;
37 //zSlice = 209;
38 //z = 209;
39 key = false;
40 embryo = "2015-12-10_Dmel_E08_Fusion_TRACK.tif";
41 angle = 0;
42 dmCounter = 0;
43
44 getDimensions(nX,nY,channels,nZ,frames);
45
46 //print("nX: "+nX+" nY: "+nY);
47
48 //*****************************************************************************
49
50 title = embryo;
51
52
53 if (angle==0) print("Started embryo "+embryo);
54
55 //************************ Part 1 ******************************************
56 //Track dorsal midline
57 if (angle==0){
58 print("Track the dorsal midline by setting a total number of 7 points in the lateral cross
       section: in the following order.");
59 print(" 1. Posterior pole 2. PMG/Germband posterior edge 3. PMG/Germband anterior edge 4.
       Posterior dorsal 5. Anterior dorsal fold 6. Cephalic furrow 7. Anterior pole");
60
61 setSlice(nSlices/2);
62 run("Enhance Contrast", "saturated=0.35");
63 run("Duplicate...", "duplicate");
64 run("RGB Color");
65
66 while (dmCounter<7){
67
68  getCursorLoc(x,y,z,flag);
69
70  if (flag == 16){
71
72   setColor(250,200,0);
73   fillOval(x-1,y-1,3,3);
74   dmX[dmCounter] = x;
75   dmY[dmCounter] = y;
76   dmZ[dmCounter] = z;
77   wait(300);
78   dmCounter++;
79
```

```
80      flag = false;
81    }
82  }
83
84  print("Ppole:␣"+dmX[0]+"␣"+dmY[0]+"␣"+dmZ[0]+"␣Germband␣1:␣"+dmX[1]+"␣"+dmY[1]+"␣"+dmZ[1]+"␣
        Germband␣2:␣"+dmX[2]+"␣"+dmY[2]+"␣"+dmZ[2]+"␣PDF:␣"+dmX[3]+"␣"+dmY[3]+"␣"+dmZ[3]+"␣ADF:␣"+
        dmX[4]+"␣"+dmY[4]+"␣"+dmZ[4]+"␣CF:␣"+dmX[5]+"␣"+dmY[5]+"␣"+dmZ[5]+"␣Apole:␣"+dmX[6]+"␣"+
        dmY[6]+"␣"+dmZ[6]);
85  run("Close");
86  }
87  //**********Part
        2************************************************************
88
89  while (flag!=4 && angle<180){
90   selectWindow(title);
91   run("TransformJ␣Rotate", "z-angle=0.0␣y-angle=0.0␣x-angle="+angle);
92   selectWindow(title+"␣rotated");
93   setSlice(nSlices/2);
94   run("Enhance␣Contrast", "saturated=0.35");
95   run("RGB␣Color");
96   getCursorLoc(x,y,z,flag);
97   setSlice(nSlices/2);
98
99
100  run("Enhance␣Contrast", "saturated=0.35");
101  run("Z␣Project...", "start="+(nSlices()/2-5)+"␣stop="+(nSlices()/2+5)+"␣projection=[Max␣
         Intensity]");
102  selectWindow(title+"␣rotated");
103  run("Set...␣", "zoom=150␣x=565␣y=336");
104
105
106
107  while (i<42 && !key){
108   getCursorLoc(x,y,z,flag);
109   key = false;
110   if (flag == 16){
111
112    if (i<21 && y>168){
113
114     if (i==0){
115      setColor(0,255,0);
116      fillRect(0,0,10,10);
117      i+=7;
118     }
119     if (i==7){
120      setColor(0,255,0);
121      fillRect(247,0,10,10);
122      i+=7;
123     }
124     if (i==14){
125      setColor(0,255,0);
126      fillRect(500,0,10,10);
127      i+=7;
128     }
129
130    }
131    setColor(0,255,0);
132    fillOval(x-1,y-1,3,3);
133    X[i] = x;
134    Y[i] = y;
135    Z[i] = z;
136    wait(300);
137    i++;
138   }
```

```
139     key = isKeyDown("space");
140     skip = isKeyDown("alt");
141     delete = isKeyDown("shift");
142     if (skip && (i==0 || i==7 || i==14 || i==21 || i==28 || i==35)){
143      for (k=i; k<i+7; k++){
144       X[k] = NaN;
145       Y[k] = NaN;
146       Z[k] = NaN;
147      }
148      if (i==0){
149       setColor(0,255,0);
150       fillRect(0,0,10,10);
151      }
152      else if (i==7){
153       setColor(0,255,0);
154       fillRect((nX/2),0,10,10);
155      }
156      else if (i==14){
157       setColor(0,255,0);
158       fillRect((nX-10),0,10,10);
159      }
160      else if (i==21){
161       setColor(0,255,0);
162       fillRect(0,(nY-10),10,10);
163      }
164      else if (i==28){
165       setColor(0,255,0);
166       fillRect((nX/2),(nY-10),10,10);
167      }
168      else if (i==35){
169       setColor(0,255,0);
170       fillRect((nX-10),(nY-10),10,10);
171      }
172
173
174      i = i+7;
175      wait(500);
176      skip = false;
177     }
178     if (delete){
179
180      delete = false;
181      setColor(255,0,0);
182      i--;
183      fillOval(X[i]-1,Y[i]-1,3,3);
184      setColor(0,255,0);
185      wait(500);
186
187     }
188    }
189
190    for (j=0; j<7; j++){
191     print("Angle␣"+angle+"␣CF-1␣"+X[j]+"␣"+Y[j]+"␣"+Z[j]+"␣ADF-1␣"+X[j+7]+"␣"+Y[j+7]+"␣"+Z[j+7]+"
           ␣PDF-1␣"+X[j+14]+"␣"+Y[j+14]+"␣"+Z[j+14]+"␣CF-2␣"+X[j+21]+"␣"+Y[j+21]+"␣"+Z[j+21]+"␣ADF
           -2␣"+X[j+28]+"␣"+Y[j+28]+"␣"+Z[j+28]+"␣PDF-2␣"+X[j+35]+"␣"+Y[j+35]+"␣"+Z[j+35]);
192    }
193
194    // Previous Furrow
195
196    for (k=0; k<20; k+=2){
197
198     if (X[k]>0){
199      PFx[k/2] = (X[k]+X[k+1])/2;
200      PFy[k/2] = (Y[k]+Y[k+1])/2;
```

```
201     }
202    }
203    Array.fill(X,NaN);
204    Array.fill(Y,NaN);
205    Array.fill(Z,NaN);
206    selectWindow(title+" rotated");
207    run("Enhance Contrast", "saturated=0.35");
208    run("Close");
209    selectWindow("MAX_"+title+" rotated");
210    run("Close");
211    selectWindow(title);
212    angle+=2;
213
214    //run("RGB Color");
215    i = 0;
216    zSlice = z;
217    key = false;
218    wait(500);
219   }
220   print("Timepoint-completed");
```

B.1.2 GBE tracking macro

Filename: *Advanced_ GBE_ Tracking.ijm* This ImageJ macro served as tracking program for the GBE trajectory. The way it works is explained in 5.2.2.

```
1   //Accurate GBE Tracking Macro including left right assymetry of Posterior Pole
2
3
4   year = "2015";
5   month = "12";
6   day = "03";
7
8   species = "Dpse";
9
10  temperature = 18;
11
12  eNumber = 36;
13  dualNum = IJ.pad(eNumber,2);
14
15  base = "F:\\Lucas_SPE_Data\\Double_sided\\Pseudoobscura_18C\\";
16
17  pRadius = 2;
18  flag = 0;
19  dorsal = true;
20
21  TOP1 = 1000;
22  TOP2 = 1000;
23  TOP3 = 1000;
24
25
26
27  // 1. Part Left-Right Assymetry
28
29  open(base+year+"-"+month+"-"+day+"_Embryo_"+eNumber+"\\Tracking\\"+year+"-"+month+"-"+day+"_"+
        species+"_E"+dualNum+"_Fusion_TRACK.tif");
30
31  run("TransformJ Turn", "z-angle=0 y-angle=0 x-angle=90");
32  run("Z Project...", "stop="+(nSlices()/2)+" projection=[Max Intensity]");
33  //setColor(0xFF0000);
```

```
34
35 while(flag == 0){  //upper (image) posterior pole edge
36
37  getCursorLoc(x,y,z,flag);
38  if (flag == 16){
39
40   print(x+" "+y+" "+z);
41   fillRect(x-pRadius,y-pRadius,pRadius*2,pRadius*2);
42  }
43 }
44 wait(300);
45 getCursorLoc(x,y,z,flag);
46 while(flag == 0){  //lower (image) posterior pole edge
47
48  getCursorLoc(x,y,z,flag);
49  if (flag == 16){
50
51   print(x+" "+y+" "+z);
52   fillRect(x-pRadius,y-pRadius,pRadius*2,pRadius*2);
53  }
54 }
55 wait(300);
56 getCursorLoc(x,y,z,flag);
57 while(flag == 0){  //anterior pole edge
58
59  getCursorLoc(x,y,z,flag);
60  if (flag == 16){
61
62   print(x+" "+y+" "+z);
63   fillRect(x-pRadius,y-pRadius,pRadius*2,pRadius*2);
64  }
65 }
66
67 //run("Close All");
68
69 // 2. Part Salami Slices
70
71 run("Image Sequence...", "open="+base+year+"-"+month+"-"+day+"_Embryo_"+eNumber+"\\
      Salami_projections\\Projection_1_to_20.tif convert_to_rgb sort");
72 setColor(0xFF0000);
73 n = 1;
74 Stack.getDimensions(nX, nY, nC, nZ, nT);
75
76 fixP1X = 0;
77 fixP1Y = nY/3*2;
78
79 fixP2X = nX/2;
80 fixP2Y = 0;
81
82 fixP3X = nX-1;
83 fixP3Y = nY/3*2;
84
85 while (n<nSlices()){
86
87  space = isKeyDown("Space");
88  shift = isKeyDown("Shift");
89  if (space){
90   break;
91  }
92  getCursorLoc(x,y,z,flag);
93
94  if (flag == 16){
95   //print("NaN NaN NaN "+x+" "+y+" "+z+" "+dorsal);
96   fillRect(x-pRadius,y-pRadius,pRadius*2,pRadius*2);
```

```
97      if (dorsal){
98       dorsal = false;
99       print("NaN NaN NaN "+x+" "+y+" "+z+" "+dorsal+" NaN NaN NaN");
100     }
101     else{
102      dorsal = true;
103
104      //Define center and radius
105
106      for (yCount=0; yCount<nY; yCount++){
107       for (xCount=0; xCount<nX; xCount++){
108
109        if (getPixel(xCount,yCount)>-10000000 && getPixel(xCount,yCount)!=0xFF0000){
110         vec1 = sqrt(pow((xCount-fixP1X),2)+pow((yCount-fixP1Y),2));
111         vec2 = sqrt(pow((xCount-fixP2X),2)+pow((yCount-fixP2Y),2));
112         vec3 = sqrt(pow((xCount-fixP3X),2)+pow((yCount-fixP3Y),2));
113
114         if (vec1<TOP1){
115          TOP1 = vec1;
116          xTOP1 = xCount;
117          yTOP1 = yCount;
118         }
119         else if (vec2<TOP2){
120          TOP2 = vec2;
121          xTOP2 = xCount;
122          yTOP2 = yCount;
123         }
124         else if (vec3<TOP3){
125          TOP3 = vec3;
126          xTOP3 = xCount;
127          yTOP3 = yCount;
128         }
129        }
130       }
131      }
132      setColor(0x00FF00);
133      fillRect(xTOP1-pRadius,yTOP1-pRadius,pRadius*2,pRadius*2);
134      fillRect(xTOP2-pRadius,yTOP2-pRadius,pRadius*2,pRadius*2);
135      fillRect(xTOP3-pRadius,yTOP3-pRadius,pRadius*2,pRadius*2);
136      setColor(0xFF0000);
137      quotient = 2*(xTOP1*(yTOP2-yTOP3)+xTOP2*(yTOP3-yTOP1)+xTOP3*(yTOP1-yTOP2));
138               xM = ((pow(xTOP1,2)+pow(yTOP1,2))*(yTOP2-yTOP3)+(pow(xTOP2,2)+pow(yTOP2,2))*(yTOP3-
                      yTOP1)+(pow(xTOP3,2)+pow(yTOP3,2))*(yTOP1-yTOP2))/quotient;
139               yM = ((pow(xTOP1,2)+pow(yTOP1,2))*(xTOP3-xTOP2)+(pow(xTOP2,2)+pow(yTOP2,2))*(xTOP1-
                      xTOP3)+(pow(xTOP3,2)+pow(yTOP3,2))*(xTOP2-xTOP1))/quotient;
140      circleRadius = ((sqrt(pow((xM-xTOP1),2)+pow((yM-yTOP1),2))+sqrt(pow((xM-xTOP2),2)+pow((yM-
             yTOP2),2))+sqrt(pow((xM-xTOP3),2)+pow((yM-yTOP3),2)))/3);
141
142      print("NaN NaN NaN "+x+" "+y+" "+z+" "+dorsal+" "+xM+" "+yM+" "+circleRadius);
143      n++;
144      setSlice(n);
145      TOP1 = 1000;
146      TOP2 = 1000;
147      TOP3 = 1000;
148     }
149     wait(250);
150    }
151    if (shift){
152     print("NaN NaN NaN NaN NaN NaN "+dorsal+" NaN NaN NaN");
153     if (dorsal) dorsal = false;
154     else{
155      dorsal = true;
156      n++;
157      setSlice(n);
```

```
158     }
159     wait(300);
160    }
161
162  }
163
164
165  //selectWindow("Log");
166  //run("Text...", "save="+base+year+"-"+month+"-"+day+"_Embryo_"+eNumber+"\\Tracking\\DV_GBELog_
         "+species+"_E"+eNumber+"_"+temperature+"C.txt");
167  //run("Close All");
```

B.1.3 Velocity visualization plugin

Filename: *Heat_map_stretched_AP_Dmel.java* This ImageJ plugin used the vectors from the Particle Image Velocimetry (PIV) pipeline and converted them into visual information using colors, and the absolute value of the vectors. The way it works is explained in section 5.2.3.

```
1   import ij.*;
2   import ij.process.*;
3   import ij.gui.*;
4   //import java.awt.*;
5   import ij.plugin.*;
6   import ij.plugin.frame.*;
7   import ij.ImagePlus.*;
8   import java.lang.*;
9   import java.util.*;
10  import java.io.*;
11  import ij.io.*;
12  import ij.gui.GenericDialog.*;
13
14  public class Heat_map_stretched_AP_Dmel implements PlugIn {
15
16   public String color_t1;
17   public String color_t0;
18
19   public void run(String arg) {
20
21    int nx, ny, nz, k, sFrame, eFrame;
22    boolean Frame1, smooth, seg, fForm, flatMap, ventralSide, dorsalSide, elongated;
23    boolean alreadyInside=false;
24    boolean furrow=false;
25    String inputPath, outputPath, zone;
26    double psiEntryThreshold, psiMainThreshold, axisEntryThreshold, axisMainThreshold,
          psiSwitchThreshold, axisSwitchThreshold;
27    double xCenter, yCenter, zCenter, xCenterS, yCenterS, zCenterS, quotient, xM, yM, xP1circ,
          yP1circ, xP2circ, yP2circ, xP3circ, yP3circ;
28          double circleRadius, radiusThreshold, distP1, distP2, distP3, topP1, topP2, topP3,
                radDiff;
29          double xP1, yP1, xP2, yP2, xP3, yP3;
30          int x_flat, y_flat;
31
32
33    GenericDialog inputParams = new GenericDialog("Conditional 4D Cell Flow Visualization");
34    inputParams.addStringField("Enter text file directory: ", "/media/Storm_RAID/Old_SPIM_data/
          Dmel1/Turned_Raw_Data_and_Flow/",40);
```

```
35  inputParams.addStringField("Enter output directory: ", "/media/Storm_RAID/Lucas/Dmel1/
        Speed_Heatmap/AP_Speed/AP_single/stretched/Run_4/", 40);
36  inputParams.addMessage("----------Dimensions----------");
37  inputParams.addNumericField("X-width: ", 400, 10);
38  inputParams.addNumericField("Y-width: ", 400, 10);
39  inputParams.addNumericField("Z-slices: ", 1024, 10);
40  inputParams.addNumericField("X-center of embryo: ", 200, 10);
41  inputParams.addNumericField("Y-center of embryo: ", 200, 10);
42  inputParams.addNumericField("Z-center of embryo: ", 512, 10);
43  inputParams.addNumericField("Starting Frame: ", 50, 10);
44  inputParams.addNumericField("Last Frame: ", 150, 10);
45  inputParams.addMessage("----------Visualization parameters----------");
46  inputParams.addCheckbox("Create flat map (needs furrow labeling)", false);
47  inputParams.addCheckbox("Cut at ventral side (needs flat map)", false);
48  inputParams.addCheckbox("Cut at dorsal side (needs flat map)", false);
49  inputParams.addCheckbox("Show dorsal side as elongation", false);
50
51  inputParams.showDialog();
52
53  if (inputParams.wasCanceled()){
54   return;
55  }
56
57  inputPath= inputParams.getNextString();
58  outputPath= inputParams.getNextString();
59  nx= (int) inputParams.getNextNumber();
60  ny= (int) inputParams.getNextNumber();
61  nz= (int) inputParams.getNextNumber();
62  xCenterS= inputParams.getNextNumber();
63  yCenterS= inputParams.getNextNumber();
64  zCenterS= inputParams.getNextNumber();
65  sFrame= (int) inputParams.getNextNumber();
66  eFrame= (int) inputParams.getNextNumber();
67  flatMap = inputParams.getNextBoolean();
68  ventralSide= inputParams.getNextBoolean();
69  dorsalSide= inputParams.getNextBoolean();
70  elongated= inputParams.getNextBoolean();
71
72  xP1 = 200;
73  yP1 = 0;
74     xP2 = 0;
75     yP2 = 300;
76     xP3 = 400;
77     yP3 = 300;
78
79     new File(outputPath+"3D_Embryo/").mkdirs();
80     new File(outputPath+"Ventral_split/").mkdirs();
81        new File(outputPath+"Dorsal_split/").mkdirs();
82        new File(outputPath+"Elongated/").mkdirs();
83
84  for (int j=sFrame; j<=eFrame; j++){
85
86  k=j-1;
87  ImagePlus Frame1_imp = IJ.createImage("Frame_"+j, "RGB black", nx, ny, nz);
88  IJ.run(Frame1_imp, "8-bit Color", "number=256");
89  //IJ.run("Threshold...");
90  IJ.setThreshold(Frame1_imp, 0, 255);
91  //IJ.run("Brightness/Contrast...");
92  IJ.run(Frame1_imp, "Apply LUT", "stack");
93  IJ.run(Frame1_imp, "Fire", "");
94
95  ImagePlus flatFrameV_imp = IJ.createImage("flatFrameV_"+j, "RGB black", 1200, 300, nz);
```

```
96    IJ.run(flatFrameV_imp, "8-bit Color", "number=256");
97    //IJ.run("Threshold...");
98    IJ.setThreshold(flatFrameV_imp, 0, 255);
99    //IJ.run("Brightness/Contrast...");
100   IJ.run(flatFrameV_imp, "Apply LUT", "stack");
101   IJ.run(flatFrameV_imp, "Fire", "");
102
103   ImagePlus flatFrameD_imp = IJ.createImage("flatFrameD_"+j, "RGB black", 1200, 300, nz);
104   IJ.run(flatFrameD_imp, "8-bit Color", "number=256");
105   //IJ.run("Threshold...");
106   IJ.setThreshold(flatFrameD_imp, 0, 255);
107   //IJ.run("Brightness/Contrast...");
108   IJ.run(flatFrameD_imp, "Apply LUT", "stack");
109   IJ.run(flatFrameD_imp, "Fire", "");
110
111   ImagePlus flatFrameEl_imp = IJ.createImage("flatFrameEl_"+j, "RGB black", 1200, 300, 2*nz);
112   IJ.run(flatFrameEl_imp, "8-bit Color", "number=256");
113   //IJ.run("Threshold...");
114   IJ.setThreshold(flatFrameEl_imp, 0, 255);
115   //IJ.run("Brightness/Contrast...");
116   IJ.run(flatFrameEl_imp, "Apply LUT", "stack");
117   IJ.run(flatFrameEl_imp, "Fire", "");
118
119   ImageStack Frame1_is= Frame1_imp.getStack();
120
121   ImageStack flatFrameV_is= flatFrameV_imp.getStack();
122   ImageStack flatFrameD_is= flatFrameD_imp.getStack();
123   ImageStack flatFrameEl_is= flatFrameEl_imp.getStack();
124
125   File file = new File(inputPath+"Timepoint_t0_"+j+"/opticalFlow.txt");
126   Scanner filestring = null;
127   try {
128    filestring= new Scanner(file);
129   } catch (Exception ex) {
130   }
131   List<String> lines = new ArrayList<String>();
132
133   int xt1;
134   int yt1;
135   int zt1;
136   int val_t1;
137
138   xP1circ = 0;
139   yP1circ = 0;
140   xP2circ = 0;
141   yP2circ = 0;
142   xP3circ = 0;
143   yP3circ = 0;
144   radiusThreshold = 0;
145  xM=0;
146  yM=0;
147   xCenter=xCenterS;
148   yCenter=yCenterS;
149   zCenter=zCenterS;
150   topP1 = nx;
151   topP2 = nx;
152   topP3 = nx;
153   circleRadius=1;
154
155   int i=0;
156   for (int z=1; z<=nz; z++){
157
158    quotient = 2*(xP1circ*(yP2circ-yP3circ)+xP2circ*(yP3circ-yP1circ)+xP3circ*(yP1circ-yP2circ))
          ;
```

```
159     xM = ((Math.pow(xP1circ,2)+Math.pow(yP1circ,2))*(yP2circ-yP3circ)+(Math.pow(xP2circ,2)+Math.
            pow(yP2circ,2))*(yP3circ-yP1circ)+(Math.pow(xP3circ,2)+Math.pow(yP3circ,2))*(yP1circ-
            yP2circ))/quotient;
160     yM = ((Math.pow(xP1circ,2)+Math.pow(yP1circ,2))*(xP3circ-xP2circ)+(Math.pow(xP2circ,2)+Math.
            pow(yP2circ,2))*(xP1circ-xP3circ)+(Math.pow(xP3circ,2)+Math.pow(yP3circ,2))*(xP2circ-
            xP1circ))/quotient;
161     circleRadius = Math.sqrt(Math.pow((xM-xP1circ),2)+Math.pow((yM-yP1circ),2));
162     radiusThreshold = circleRadius-20;
163     topP1 = nx;
164     topP2 = nx;
165     topP3 = nx;
166
167     for (int y=1; y<=ny; y++){
168
169      for (int x=1; x<=nx; x++){
170
171       String rows =filestring.nextLine();
172       String[] columns = rows.split("␣");
173       double dx=Double.parseDouble(columns[0]);
174       double dy=Double.parseDouble(columns[1]);
175       double dz=Double.parseDouble(columns[2]);
176
177       if (dx!=0 || dy!=0 || dz!=0){
178
179         xt1 = x + (int) Math.round(dx);
180         yt1 = y + (int) Math.round(dy);
181         zt1 = z + (int) Math.round(dz);
182
183        //xCenter=xM;
184        //yCenter=yM;
185        xCenter=xCenterS;
186              yCenter=yCenterS;
187              zCenter=zCenterS;
188        double u=x-xCenter;
189        double v=y-yCenter;
190        double w=z-zCenter;
191
192        double dist_t0=Math.sqrt(Math.pow(v,2)+Math.pow(u,2));
193        double dist_t1=Math.sqrt(Math.pow((v+dy),2)+Math.pow((u+dx),2));
194        double ddist=dist_t1-dist_t0;
195
196        double psi_t0=Math.atan2(v,u);
197        double psi_t1=Math.atan2((v+dy),(u+dx));
198        double dpsi=Math.abs(psi_t1)-Math.abs(psi_t0);
199
200        double daxis=-dz;
201
202        //double slope=Math.abs(ddist/daxis);
203
204                              //Furrow visualization with radii for each cross section
205
206                                  distP1=Math.sqrt(Math.pow((xt1-xP1),2)+Math.pow((yt1-yP1),2));
207                                  distP2=Math.sqrt(Math.pow((xt1-xP2),2)+Math.pow((yt1-yP2),2));
208                                  distP3=Math.sqrt(Math.pow((xt1-xP3),2)+Math.pow((yt1-yP3),2));
209
210
211                                  if (distP1<topP1){
212                                      topP1 = distP1;
213                                      xP1circ = xt1;
214                                      yP1circ = yt1;
215                                  }
216                                  else if (distP2<topP2) {
217                                      topP2 = distP2;
218                                      xP2circ = xt1;
```

```
219                                 yP2circ = yt1;
220                             }
221                             else if (distP3<topP3){
222                                 topP3 = distP3;
223                                 xP3circ = xt1;
224                                 yP3circ = yt1;
225                             }
226
227         //Visualization start
228
229         if (dz>0){
230          val_t1= (int) (40*Math.sqrt(Math.pow(dz,2)));
231         }
232         else {
233          val_t1 = 0;
234         }
235
236         if (val_t1>254) {
237                              val_t1=254;
238                             }
239                             Frame1_is.setVoxel(xt1,yt1,zt1,val_t1);
240
241                             //Unrolling of embryo
242         psi_t1 = psi_t1 + Math.PI/2;
243         if(psi_t1 > (Math.PI)){
244          psi_t1 = psi_t1 - 2 * Math.PI;
245         }
246         if (flatMap && ventralSide){
247          x_flat = 600 + (int) ((psi_t1)*157);
248          radDiff = circleRadius-dist_t1;
249          y_flat = 100 + (int) (radDiff);
250          flatFrameV_is.setVoxel(x_flat, y_flat, zt1, val_t1);
251         }
252         psi_t1 = psi_t1 - Math.PI/2;
253         if(psi_t1 < (-Math.PI)){
254          psi_t1 = psi_t1 + 2 * Math.PI;
255         }
256         if (flatMap && dorsalSide){
257          if (psi_t1<0) x_flat = 600 + (int) ((-Math.PI-psi_t1)*157);
258          else x_flat = 600 + (int) ((Math.PI-psi_t1)*157);
259          radDiff = circleRadius-dist_t1;
260          y_flat = 100 + (int) (radDiff);
261          flatFrameD_is.setVoxel(x_flat, y_flat, zt1, val_t1);
262         }
263         if (flatMap && elongated){
264          if (psi_t1<0) x_flat = 600 + (int) ((-Math.PI-psi_t1)*157);
265          else x_flat = 600 + (int) ((Math.PI-psi_t1)*157);
266          radDiff = circleRadius-dist_t1;
267          y_flat = 100 + (int) (radDiff);
268          flatFrameEl_is.setVoxel(x_flat, y_flat, zt1, val_t1);
269
270          if (x>168) {
271           x_flat = 600 + (int) ((psi_t1)*157);
272           radDiff = circleRadius-dist_t1;
273           y_flat = 100 + (int) (radDiff);
274           flatFrameEl_is.setVoxel(x_flat, y_flat, 1850-zt1, val_t1);
275          }
276         }
277        }
278       }
279      }
280     }
281
282
```

```
283     IJ.saveAs(Frame1_imp, "Tiff", outputPath+"3D_Embryo/Frame_"+j+".tif");
284
285     if (flatMap && ventralSide){
286      IJ.saveAs(flatFrameV_imp, "Tiff", outputPath+"Ventral_split/flatFrameV_"+j+".tif");
287     }
288
289     if (flatMap && dorsalSide){
290      IJ.saveAs(flatFrameD_imp, "Tiff", outputPath+"Dorsal_split/flatFrameD_"+j+".tif");
291     }
292     if (flatMap && elongated){
293      IJ.saveAs(flatFrameEl_imp, "Tiff", outputPath+"Elongated/flatFrameEl_"+j+".tif");
294     }
295
296     }
297    }
298   }
```

B.1.4 Direction visualization plugin

Filename: *Conditional_Spectrum.java* This ImageJ plugin used the vectors from the PIV pipeline and converted them into visual information using colors, and the main direction of the vectors. The way it works is explained in section 5.2.3.

```
1   import ij.*;
2   import ij.process.*;
3   import ij.gui.*;
4   //import java.awt.*;
5   import ij.plugin.*;
6   import ij.plugin.frame.*;
7   import ij.ImagePlus.*;
8   import java.lang.*;
9   import java.util.*;
10  import java.io.*;
11  import ij.io.*;
12  import ij.gui.GenericDialog.*;
13
14  public class Conditional_Spectrum implements PlugIn {
15
16   public String color_t1;
17   public String color_t0;
18
19   public void run(String arg) {
20
21    int nx, ny, nz, k, sFrame, eFrame;
22    boolean Frame1, fForm, flatMap, ventralSide, dorsalSide, elongated;
23    boolean alreadyInside=false;
24    boolean furrow=false;
25    String inputPath, outputPath, zone;
26    double psiEntryThreshold, psiMainThreshold, axisEntryThreshold, axisMainThreshold,
          psiSwitchThreshold, axisSwitchThreshold;
27    double xCenter, yCenter, zCenter, xCenterS, yCenterS, zCenterS, quotient, xM, yM, xP1circ,
          yP1circ, xP2circ, yP2circ, xP3circ, yP3circ;
28          double circleRadius, radiusThreshold, distP1, distP2, distP3, topP1, topP2, topP3,
                radDiff;
29          double xP1, yP1, xP2, yP2, xP3, yP3;
30          int x_flat, y_flat;
31          int colorID_t0, colorID_t1, preColorID_t0, postColorID_t1, colorCounter_t0,
                colorCounter_t1;
32          double colorSecValue_t1, colorSecValue_t0;
```

```
33
34    GenericDialog inputParams = new GenericDialog("Conditional 4D Cell Flow Visualization");
35    inputParams.addStringField("Enter text file directory: ", "/media/AG_Lemke3/Dmel1/
          Turned_Raw_Data_and_Flow/",40);
36    inputParams.addStringField("Enter output directory: ", "/media/Volume/Dmel/Dmel_Flow/Run_2/",
           40);
37    inputParams.addMessage("----------------------Dimensions-----------------------------------")
          ;
38    inputParams.addNumericField("X-width: ", 400, 10);
39    inputParams.addNumericField("Y-width: ", 400, 10);
40    inputParams.addNumericField("Z-slices: ", 1024, 10);
41    inputParams.addNumericField("X-center of embryo: ", 200, 10);
42    inputParams.addNumericField("Y-center of embryo: ", 200, 10);
43    inputParams.addNumericField("Z-center of embryo: ", 512, 10);
44    inputParams.addNumericField("Starting Frame: ", 50, 10);
45    inputParams.addNumericField("Last Frame: ", 150, 10);
46    inputParams.addCheckbox("First Frame", true);
47    inputParams.addMessage("----------------------Visualization parameters-----------------------")
          ;
48    inputParams.addNumericField("Delta psi entry Threshold: ", 0.005, 10);
49    inputParams.addNumericField("Delta psi main Threshold: ", 0.002, 10);
50    inputParams.addNumericField("Delta axis entry Threshold: ", 0.5, 10);
51    inputParams.addNumericField("Delta axis main Threshold: ", 0.2, 10);
52    inputParams.addNumericField("Delta psi switch color Threshold: ", 0.005, 10);
53    inputParams.addNumericField("Delta axis switch color Threshold: ", 0.5, 10);
54    inputParams.addCheckbox("Enable furrow labeling", false);
55    inputParams.addCheckbox("Create flat map (needs furrow labeling)", false);
56    inputParams.addCheckbox("Cut at ventral side (needs flat map)", false);
57    inputParams.addCheckbox("Cut at dorsal side (needs flat map)", false);
58    inputParams.addCheckbox("Show dorsal side as elongation", false);
59
60    inputParams.showDialog();
61
62    if (inputParams.wasCanceled()){
63     return;
64    }
65
66    inputPath= inputParams.getNextString();
67    outputPath= inputParams.getNextString();
68    nx= (int) inputParams.getNextNumber();
69    ny= (int) inputParams.getNextNumber();
70    nz= (int) inputParams.getNextNumber();
71    xCenterS= inputParams.getNextNumber();
72    yCenterS= inputParams.getNextNumber();
73    zCenterS= inputParams.getNextNumber();
74    sFrame= (int) inputParams.getNextNumber();
75    eFrame= (int) inputParams.getNextNumber();
76    Frame1= inputParams.getNextBoolean();
77    psiEntryThreshold= inputParams.getNextNumber();
78    psiMainThreshold= inputParams.getNextNumber();
79    axisEntryThreshold= inputParams.getNextNumber();
80    axisMainThreshold= inputParams.getNextNumber();
81    psiSwitchThreshold= inputParams.getNextNumber();
82    axisSwitchThreshold= inputParams.getNextNumber();
83    fForm= inputParams.getNextBoolean();
84    flatMap = inputParams.getNextBoolean();
85    ventralSide= inputParams.getNextBoolean();
86    dorsalSide= inputParams.getNextBoolean();
87    elongated= inputParams.getNextBoolean();
88
89    xP1 = 200;
90       yP1 = 0;
91       xP2 = 0;
92       yP2 = 279;
```

```
93          xP3 = 336;
94          yP3 = 279;
95
96          new File(outputPath+"3D_Embryo/").mkdirs();
97          new File(outputPath+"Ventral_split/").mkdirs();
98              new File(outputPath+"Dorsal_split/").mkdirs();
99              new File(outputPath+"Elongated/").mkdirs();
100
101     for (int j=sFrame; j<=eFrame; j++){
102
103     k=j-1;
104     ImagePlus Frame1_imp = IJ.createImage("Frame_"+j, "RGB black", nx, ny, nz);
105     IJ.run(Frame1_imp, "8-bit Color", "number=256");
106     //IJ.run("Threshold...");
107     IJ.setThreshold(Frame1_imp, 0, 255);
108     //IJ.run("Brightness/Contrast...");
109     IJ.run(Frame1_imp, "Apply LUT", "stack");
110     IJ.run(Frame1_imp, "12_colors_ramp", "");
111
112     ImagePlus flatFrameV_imp = IJ.createImage("flatFrameV_"+j, "RGB black", 1300, 200, nz);
113     IJ.run(flatFrameV_imp, "8-bit Color", "number=256");
114     //IJ.run("Threshold...");
115     IJ.setThreshold(flatFrameV_imp, 0, 255);
116     //IJ.run("Brightness/Contrast...");
117     IJ.run(flatFrameV_imp, "Apply LUT", "stack");
118     IJ.run(flatFrameV_imp, "12_colors_ramp", "");
119
120     ImagePlus flatFrameD_imp = IJ.createImage("flatFrameD_"+j, "RGB black", 1300, 200, nz);
121     IJ.run(flatFrameD_imp, "8-bit Color", "number=256");
122     //IJ.run("Threshold...");
123     IJ.setThreshold(flatFrameD_imp, 0, 255);
124     //IJ.run("Brightness/Contrast...");
125     IJ.run(flatFrameD_imp, "Apply LUT", "stack");
126     IJ.run(flatFrameD_imp, "12_colors_ramp", "");
127
128     ImagePlus flatFrameEl_imp = IJ.createImage("flatFrameEl_"+j, "RGB black", 1300, 200, 2*nz);
129     IJ.run(flatFrameEl_imp, "8-bit Color", "number=256");
130     //IJ.run("Threshold...");
131     IJ.setThreshold(flatFrameEl_imp, 0, 255);
132     //IJ.run("Brightness/Contrast...");
133     IJ.run(flatFrameEl_imp, "Apply LUT", "stack");
134     IJ.run(flatFrameEl_imp, "12_colors_ramp", "");
135
136     ImageStack Frame1_is= Frame1_imp.getStack();
137
138     ImageStack flatFrameV_is= flatFrameV_imp.getStack();
139     ImageStack flatFrameD_is= flatFrameD_imp.getStack();
140     ImageStack flatFrameEl_is= flatFrameEl_imp.getStack();
141
142     //int[] pixels_t1=(int[])Frame1_is.getPixels();
143
144     String path = (outputPath+"3D_Embryo/Frame_"+k+".tif");
145     ImagePlus imp = (Frame1 == false) ? IJ.openImage(path) : null;
146     ImageStack Frame0_is = null;
147     try {
148      Frame0_is = imp.getImageStack();
149     } catch (NullPointerException ex) {
150      IJ.log("Couldn't open file at" + path);
151     }
152     //ImagePlus Frame0_imp=IJ.openImage(path);
153     //ImageStack Frame0_is=Frame0_imp.getStack();
154      //int pixels_t0[]=(int[])Frame0_is.getPixels();
155     //add 8bitC procedure if necessary
156
```

```
157     //File file = new File(inputPath+"Timepoint_t0_"+j+"/opticalFlow.txt");
158     File file = new File(inputPath+"Timepoint_t0_"+j+"/opticalFlow.txt");
159     Scanner filestring = null;
160     try {
161      filestring= new Scanner(file);
162     } catch (Exception ex) {
163     }
164     List<String> lines = new ArrayList<String>();
165
166     int xt1;
167     int yt1;
168     int zt1;
169     int val_t1;
170
171     xP1circ = 0;
172             yP1circ = 0;
173             xP2circ = 0;
174             yP2circ = 0;
175             xP3circ = 0;
176             yP3circ = 0;
177             radiusThreshold = 0;
178             xM=0;
179             yM=0;
180     xCenter=xCenterS;
181             yCenter=yCenterS;
182             zCenter=zCenterS;
183     topP1 = nx;
184             topP2 = nx;
185             topP3 = nx;
186             circleRadius=1;
187
188     int i=0;
189     for (int z=1; z<=nz; z++){
190
191                     quotient = 2*(xP1circ*(yP2circ-yP3circ)+xP2circ*(yP3circ-yP1circ)+xP3circ*(
                            yP1circ-yP2circ));
192                     xM = ((Math.pow(xP1circ,2)+Math.pow(yP1circ,2))*(yP2circ-yP3circ)+(Math.pow(
                            xP2circ,2)+Math.pow(yP2circ,2))*(yP3circ-yP1circ)+(Math.pow(xP3circ,2)+
                            Math.pow(yP3circ,2))*(yP1circ-yP2circ))/quotient;
193                     yM = ((Math.pow(xP1circ,2)+Math.pow(yP1circ,2))*(xP3circ-xP2circ)+(Math.pow(
                            xP2circ,2)+Math.pow(yP2circ,2))*(xP1circ-xP3circ)+(Math.pow(xP3circ,2)+
                            Math.pow(yP3circ,2))*(xP2circ-xP1circ))/quotient;
194
195                     circleRadius = Math.sqrt(Math.pow((xM-xP1circ),2)+Math.pow((yM-yP1circ),2));
196                     radiusThreshold = circleRadius-20;
197
198                     topP1 = nx;
199             topP2 = nx;
200             topP3 = nx;
201
202      for (int y=1; y<=ny; y++){
203
204       for (int x=1; x<=nx; x++){
205
206
207
208         String rows =filestring.nextLine();
209         String[] columns = rows.split("␣");
210         double dx=Double.parseDouble(columns[0]);
211         double dy=Double.parseDouble(columns[1]);
212         double dz=Double.parseDouble(columns[2]);
213
214         if (dx!=0 || dy!=0 || dz!=0){
215
```

```
216          xt1 = x + (int) Math.round(dx);
217          yt1 = y + (int) Math.round(dy);
218          zt1 = z + (int) Math.round(dz);
219
220         xCenter=xM;
221         yCenter=yM;
222
223         //IJ.log("dX: "+dx+" dY: "+dy+" dZ: "+dz+" Z: "+z);
224
225         double u=x-xCenter;
226         double v=y-yCenter;
227         double w=z-zCenter;
228
229         double dist_t0=Math.sqrt(Math.pow(v,2)+Math.pow(u,2));
230         double dist_t1=Math.sqrt(Math.pow((v+dy),2)+Math.pow((u+dx),2));
231         double ddist=dist_t1-dist_t0;
232
233         double psi_t0=Math.atan2(v,u);
234         double psi_t1=Math.atan2((v+dy),(u+dx));
235         double dpsi=Math.abs(psi_t1)-Math.abs(psi_t0);
236
237         double daxis=-dz;
238
239         double circleTracklength = dpsi * (dist_t0+dist_t1)/2;
240         double colorSector = Math.atan2(-circleTracklength,-daxis);
241
242         //double slope=Math.abs(ddist/daxis);
243
244                             //Furrow visualization with radii for each cross section
245
246
247                                 distP1=Math.sqrt(Math.pow((xt1-xP1),2)+Math.pow((yt1-yP1),2));
248                                 distP2=Math.sqrt(Math.pow((xt1-xP2),2)+Math.pow((yt1-yP2),2));
249                                 distP3=Math.sqrt(Math.pow((xt1-xP3),2)+Math.pow((yt1-yP3),2));
250
251
252                                 if (distP1<topP1){
253                                     topP1 = distP1;
254                                     xP1circ = xt1;
255                                     yP1circ = yt1;
256                                 }
257                                 else if (distP2<topP2) {
258                                     topP2 = distP2;
259                                     xP2circ = xt1;
260                                     yP2circ = yt1;
261                                 }
262                                 else if (distP3<topP3){
263                                     topP3 = distP3;
264                                     xP3circ = xt1;
265                                     yP3circ = yt1;
266                                 }
267
268
269         //Visualization start
270
271         val_t1 = 0;
272
273
274         String color_t1 = null;
275         String color_t0 = null;
276         if (Frame1==false){
277
278          //Frame0_is.setSlice(z);
279          int val_t0 = (int) Frame0_is.getVoxel(x,y,z);
```

```
280
281         //Color table
282
283
284         preColorID_t0 = (int)Math.floor(val_t0/16);
285         colorSecValue_t0 = 0;
286
287         switch (preColorID_t0) {
288
289          case 0: colorID_t0 = 0;
290
291           break;
292          case 1: colorID_t0 = 0;
293
294           break;
295          case 2: colorID_t0 = 1;
296           colorSecValue_t0 = 0;
297           break;
298          case 3: colorID_t0 = 0;
299
300           break;
301          case 4: colorID_t0 = 2;
302           colorSecValue_t0 = Math.PI*1/6;
303           break;
304          case 5: colorID_t0 = 3;
305           colorSecValue_t0 = Math.PI*2/6;
306           break;
307          case 6: colorID_t0 = 4;
308           colorSecValue_t0 = Math.PI*3/6;
309           break;
310          case 7: colorID_t0 = 6;
311           colorSecValue_t0 = Math.PI*5/6;
312           break;
313          case 8: colorID_t0 = 7;
314           colorSecValue_t0 = -Math.PI;
315           break;
316          case 9: colorID_t0 = 5;
317           colorSecValue_t0 = Math.PI*4/6;
318           break;
319          case 10: colorID_t0 = 8;
320           colorSecValue_t0 = -Math.PI*5/6;
321           break;
322          case 11: colorID_t0 = 9;
323           colorSecValue_t0 = -Math.PI*4/6;
324           break;
325          case 12: colorID_t0 = 10;
326           colorSecValue_t0 = -Math.PI*3/6;
327           break;
328          case 13: colorID_t0 = 11;
329           colorSecValue_t0 = -Math.PI*2/6;
330           break;
331          case 14: colorID_t0 = 12;
332           colorSecValue_t0 = -Math.PI*1/6;
333           break;
334          case 15: colorID_t0 = 13;
335
336           break;
337
338          default: colorID_t0 = 0;
339           colorSecValue_t0 = 0;
340
341          break;
342
343         }
```

```
344
345         colorCounter_t0 = val_t0%16;
346
347           boolean chigh1=false;
348           boolean chigh2=false;
349           boolean penalty=false;
350           boolean assignment=false;
351
352           //check and assign preliminary color tag for t1
353           colorID_t1 = 13;
354           colorCounter_t1 = 0;
355
356           if (Math.abs(dpsi)>psiEntryThreshold || Math.abs(daxis)>axisEntryThreshold ||
                  colorCounter_t0>=3){
357
358            if ((colorSector>=-Math.PI && colorSector<-Math.PI*11/12) || (colorSector<=Math.PI &&
                     colorSector>=Math.PI*11/12)){
359             colorID_t1 = 7;
360             colorSecValue_t1 = -Math.PI;
361            }
362            else if (colorSector>=-Math.PI*11/12 && colorSector<-Math.PI*9/12){
363             colorID_t1 = 8;
364             colorSecValue_t1 = -Math.PI*5/6;
365            }
366            else if (colorSector>=-Math.PI*9/12 && colorSector<-Math.PI*7/12){
367             colorID_t1 = 9;
368             colorSecValue_t1 = -Math.PI*4/6;
369            }
370            else if (colorSector>=-Math.PI*7/12 && colorSector<-Math.PI*5/12){
371             colorID_t1 = 10;
372             colorSecValue_t1 = -Math.PI*3/6;
373            }
374            else if (colorSector>=-Math.PI*5/12 && colorSector<-Math.PI*3/12){
375             colorID_t1 = 11;
376             colorSecValue_t1 = -Math.PI*2/6;
377            }
378            else if (colorSector>=-Math.PI*3/12 && colorSector<-Math.PI*1/12){
379             colorID_t1 = 12;
380             colorSecValue_t1 = -Math.PI*1/6;
381            }
382            else if (colorSector>=-Math.PI*1/12 && colorSector<Math.PI*1/12){
383             colorID_t1 = 1;
384             colorSecValue_t1 = 0;
385            }
386            else if (colorSector>=Math.PI*1/12 && colorSector<Math.PI*3/12){
387             colorID_t1 = 2;
388             colorSecValue_t1 = Math.PI*1/6;
389            }
390            else if (colorSector>=Math.PI*3/12 && colorSector<Math.PI*5/12){
391             colorID_t1 = 3;
392             colorSecValue_t1 = Math.PI*2/6;
393            }
394            else if (colorSector>=Math.PI*5/12 && colorSector<Math.PI*7/12){
395             colorID_t1 = 4;
396             colorSecValue_t1 = Math.PI*3/6;
397            }
398            else if (colorSector>=Math.PI*7/12 && colorSector<Math.PI*9/12){
399             colorID_t1 = 5;
400             colorSecValue_t1 = Math.PI*4/6;
401            }
402            else if (colorSector>=Math.PI*9/12 && colorSector<Math.PI*11/12){
403             colorID_t1 = 6;
404             colorSecValue_t1 = Math.PI*5/6;
405            }
```

```
406
407             //Comparison with t0 and assignement of final color tag
408
409             if (colorID_t0 == colorID_t1){
410
411              colorCounter_t1 = colorCounter_t0 + 1;
412              if (colorCounter_t1>=5) colorCounter_t1 = 5;
413             }
414             else if (colorID_t0 == 0 || colorID_t0 == 13){
415
416              colorCounter_t1 = 0;
417             }
418             else if (colorID_t1 != colorID_t0){
419
420            if ((Math.abs(colorID_t1-colorID_t0) == 1 || Math.abs(colorID_t1-colorID_t0) == 11) &&
                   colorID_t0 !=7){
421
422             if (colorSecValue_t0 - 0.43633 < colorSector && colorSector <  colorSecValue_t0 +
                   0.43633){
423
424              colorID_t1 = colorID_t0;
425              colorCounter_t1 = colorCounter_t0;
426             }
427             else if (colorCounter_t0 >= 5){
428
429              colorID_t1 = colorID_t0;
430              colorCounter_t1 = 0;
431             }
432             else{
433              colorID_t1 = 13;
434              colorCounter_t1 = 0;
435             }
436            }
437            else if (Math.abs(colorID_t1-colorID_t0) == 1 && colorID_t0 == 7){
438
439             if ((-Math.PI <= colorSector && colorSector <-Math.PI + 0.43633) || (Math.PI -
                   0.43633 < colorSector && colorSector <= Math.PI)){
440
441              colorID_t1 = colorID_t0;
442              colorCounter_t1 = colorCounter_t0;
443             }
444             else if (colorCounter_t0 >= 5){
445
446              colorID_t1 = colorID_t0;
447              colorCounter_t1 = 0;
448             }
449             else{
450              colorID_t1 = 13;
451              colorCounter_t1 = 0;
452             }
453            }
454             }
455            }
456            else {
457             colorID_t1 = 13;
458             colorCounter_t1 =0;
459            }
460        }
461        else {
462         colorID_t1 = 13;
463         if (Math.abs(dpsi)>psiEntryThreshold || Math.abs(daxis)>axisEntryThreshold){
464
465            colorCounter_t1 = 0;
466
```

```
467             if ((colorSector>=-Math.PI && colorSector<-Math.PI*11/12) || (colorSector<=Math.PI &&
                    colorSector>=Math.PI*11/12)){
468              colorID_t1 = 7;
469              colorSecValue_t1 = -Math.PI;
470             }
471             else if (colorSector>=-Math.PI*11/12 && colorSector<-Math.PI*9/12){
472              colorID_t1 = 8;
473              colorSecValue_t1 = -Math.PI*5/6;
474             }
475             else if (colorSector>=-Math.PI*9/12 && colorSector<-Math.PI*7/12){
476              colorID_t1 = 9;
477              colorSecValue_t1 = -Math.PI*4/6;
478             }
479             else if (colorSector>=-Math.PI*7/12 && colorSector<-Math.PI*5/12){
480              colorID_t1 = 10;
481              colorSecValue_t1 = -Math.PI*3/6;
482             }
483             else if (colorSector>=-Math.PI*5/12 && colorSector<-Math.PI*3/12){
484              colorID_t1 = 11;
485              colorSecValue_t1 = -Math.PI*2/6;
486             }
487             else if (colorSector>=-Math.PI*3/12 && colorSector<-Math.PI*1/12){
488              colorID_t1 = 12;
489              colorSecValue_t1 = -Math.PI*1/6;
490             }
491             else if (colorSector>=-Math.PI*1/12 && colorSector<Math.PI*1/12){
492              colorID_t1 = 1;
493              colorSecValue_t1 = 0;
494             }
495             else if (colorSector>=Math.PI*1/12 && colorSector<Math.PI*3/12){
496              colorID_t1 = 2;
497              colorSecValue_t1 = Math.PI*1/6;
498             }
499             else if (colorSector>=Math.PI*3/12 && colorSector<Math.PI*5/12){
500              colorID_t1 = 3;
501              colorSecValue_t1 = Math.PI*2/6;
502             }
503             else if (colorSector>=Math.PI*5/12 && colorSector<Math.PI*7/12){
504              colorID_t1 = 4;
505              colorSecValue_t1 = Math.PI*3/6;
506             }
507             else if (colorSector>=Math.PI*7/12 && colorSector<Math.PI*9/12){
508              colorID_t1 = 5;
509              colorSecValue_t1 = Math.PI*4/6;
510             }
511             else if (colorSector>=Math.PI*9/12 && colorSector<Math.PI*11/12){
512              colorID_t1 = 6;
513              colorSecValue_t1 = Math.PI*5/6;
514             }
515         }
516         else {
517          colorCounter_t1 = 0;
518          colorID_t1 = 13;
519         }
520         }
521          //Convert to LUT
522
523            switch (colorID_t1){
524
525             case 1: postColorID_t1 = 32;
526              break;
527             case 2: postColorID_t1 = 64;
528              break;
529             case 3: postColorID_t1 = 80;
```

```
530             break;
531            case 4: postColorID_t1 = 96;
532             break;
533            case 5: postColorID_t1 = 144;
534             break;
535            case 6: postColorID_t1 = 112;
536             break;
537            case 7: postColorID_t1 = 128;
538             break;
539            case 8: postColorID_t1 = 160;
540             break;
541            case 9: postColorID_t1 = 176;
542             break;
543            case 10: postColorID_t1 = 192;
544             break;
545            case 11: postColorID_t1 = 208;
546             break;
547            case 12: postColorID_t1 = 224;
548             break;
549            case 13: postColorID_t1 = 240;
550             break;
551
552            default: postColorID_t1 = 0;
553            break;
554           }
555           val_t1 = postColorID_t1 + colorCounter_t1;
556
557                       Frame1_is.setVoxel(xt1,yt1,zt1,val_t1);
558
559                       //Unrolling of embryo
560                       /*if (flatMap && ventralSide){
561                        x_flat = 650 + (int) ((psi_t1)*circleRadius);
562                        radDiff = circleRadius-dist_t1;
563                        y_flat = 100 + (int) (radDiff);
564                        flatFrameV_is.setVoxel(x_flat, y_flat, zt1, val_t1);
565                       }*/
566
567                       if (flatMap && dorsalSide){
568                        if (psi_t1<0) x_flat = 650 + (int) ((-Math.PI-psi_t1)*circleRadius);
569                        else x_flat = 650 + (int) ((Math.PI-psi_t1)*circleRadius);
570                        radDiff = circleRadius-dist_t1;
571                        y_flat = 100 + (int) (radDiff);
572                        flatFrameD_is.setVoxel(x_flat, y_flat, zt1, val_t1);
573
574                       }
575                       if (flatMap && elongated){
576
577
578                        if (psi_t1<0) x_flat = 650 + (int) ((-Math.PI-psi_t1)*circleRadius);
579                        else x_flat = 650 + (int) ((Math.PI-psi_t1)*circleRadius);
580                        radDiff = circleRadius-dist_t1;
581                        y_flat = 100 + (int) (radDiff);
582                        flatFrameEl_is.setVoxel(x_flat, y_flat, zt1, val_t1);
583
584                        if (x>168) {
585                         x_flat = 650 + (int) ((psi_t1)*circleRadius);
586                         radDiff = circleRadius-dist_t1;
587                         y_flat = 100 + (int) (radDiff);
588                         flatFrameEl_is.setVoxel(x_flat, y_flat, 1850-zt1, val_t1);
589                        }
590                       }
591         }
592
593     }
```

```
594     }
595
596   }
597   //pixels_t1[i]=(int) val_t1;
598   //Frame1_is.setSlice(zt1);
599
600   IJ.saveAs(Frame1_imp, "Tiff", outputPath+"3D_Embryo/Frame_"+j+".tif");
601
602   if (flatMap && ventralSide){
603    IJ.saveAs(flatFrameV_imp, "Tiff", outputPath+"Ventral_split/flatFrameV_"+j+".tif");
604   }
605
606   if (flatMap && dorsalSide){
607    IJ.saveAs(flatFrameD_imp, "Tiff", outputPath+"Dorsal_split/flatFrameD_"+j+".tif");
608   }
609   if (flatMap && elongated){
610    //IJ.saveAs(flatFrameEl_imp, "Tiff", outputPath+"Elongated/flatFrameEl_"+j+".tif");
611   }
612   Frame1=false;
613    }
614  }
615  }
```

B.1.5 Track visualization plugin

B.2 Matlab scripts

The following section contains the Matlab scripts used in this thesis. Some of these scripts make use of the geom3d [Legland, 2016].

B.2.1 Furrow analysis

To be found on the CD (>1000 lines). The furrow tracking analysis script was used to match the tracked points of each furrow back to the embryo and calculate the required parameters. The main steps are listed in the method section.

B.2.2 Embryo parameters

The Embryo parameters script made further use of the furrow tracking data, was used to calculate distances between several interest points, as well as size and extension of certain structures.

```
1  % Systemic Furrow system analysis
2  %
3  %This software is used to do a step by step analysis of the network
4  %underlying the furrow system, to unravel the hidden connections.
```

```
5  %
6  %Input: Furrow Arrays from Furrowdepth fixed
7  %
8  %
9  %
10 %Output: CF-Angle, DF-AP-Extension, Fold-depth, Fold position
11
12 aBegDF = 85
13
14 aEndDF = 95
15
16 aBegCF = 30
17
18 aEndCF = 50
19
20
21 %************************Dmel*************************************************
22
23
24 %Simple Depth
25
26 adfArrayMel(:,23,:) = sqrt(((adfArrayMel(:,5,:)+adfArrayMel(:,17,:))/2-adfArrayMel(:,11,:)).^2
       +  ((adfArrayMel(:,6,:)+adfArrayMel(:,18,:))/2-adfArrayMel(:,12,:)).^2 + ((adfArrayMel
       (:,7,:)+adfArrayMel(:,19,:))/2-adfArrayMel(:,13,:)).^2);
27 pdfArrayMel(:,23,:) = sqrt(((pdfArrayMel(:,5,:)+pdfArrayMel(:,17,:))/2-pdfArrayMel(:,11,:)).^2
       +  ((pdfArrayMel(:,6,:)+pdfArrayMel(:,18,:))/2-pdfArrayMel(:,12,:)).^2 + ((pdfArrayMel
       (:,7,:)+pdfArrayMel(:,19,:))/2-pdfArrayMel(:,13,:)).^2);
28 cfArrayMel(:,23,:) = sqrt(((cfArrayMel(:,5,:)+cfArrayMel(:,17,:))/2-cfArrayMel(:,11,:)).^2 +
       ((cfArrayMel(:,6,:)+cfArrayMel(:,18,:))/2-cfArrayMel(:,12,:)).^2 + ((cfArrayMel(:,7,:)+
       cfArrayMel(:,19,:))/2-cfArrayMel(:,13,:)).^2);
29
30
31 %Advanced Depth
32
33 adfArrayMel(:,24,:) = sqrt((adfArrayMel(:,8,:)-adfArrayMel(:,5,:)).^2+(adfArrayMel(:,9,:)-
       adfArrayMel(:,6,:)).^2+(adfArrayMel(:,10,:)-adfArrayMel(:,7,:)).^2) + sqrt ((adfArrayMel
       (:,11,:)-adfArrayMel(:,8,:)).^2+(adfArrayMel(:,12,:)-adfArrayMel(:,9,:)).^2+(adfArrayMel
       (:,13,:)-adfArrayMel(:,10,:)).^2);
34 adfArrayMel(:,25,:) = sqrt((adfArrayMel(:,17,:)-adfArrayMel(:,14,:)).^2+(adfArrayMel(:,18,:)-
       adfArrayMel(:,15,:)).^2+(adfArrayMel(:,19,:)-adfArrayMel(:,16,:)).^2) + sqrt ((adfArrayMel
       (:,14,:)-adfArrayMel(:,11,:)).^2+(adfArrayMel(:,15,:)-adfArrayMel(:,12,:)).^2+(adfArrayMel
       (:,16,:)-adfArrayMel(:,13,:)).^2);
35 adfArrayMel(:,26,:) = (adfArrayMel(:,24,:)+adfArrayMel(:,24,:))/2;
36
37 pdfArrayMel(:,24,:) = sqrt((pdfArrayMel(:,8,:)-pdfArrayMel(:,5,:)).^2+(pdfArrayMel(:,9,:)-
       pdfArrayMel(:,6,:)).^2+(pdfArrayMel(:,10,:)-pdfArrayMel(:,7,:)).^2) + sqrt ((pdfArrayMel
       (:,11,:)-pdfArrayMel(:,8,:)).^2+(pdfArrayMel(:,12,:)-pdfArrayMel(:,9,:)).^2+(pdfArrayMel
       (:,13,:)-pdfArrayMel(:,10,:)).^2);
38 pdfArrayMel(:,25,:) = sqrt((pdfArrayMel(:,17,:)-pdfArrayMel(:,14,:)).^2+(pdfArrayMel(:,18,:)-
       pdfArrayMel(:,15,:)).^2+(pdfArrayMel(:,19,:)-pdfArrayMel(:,16,:)).^2) + sqrt ((pdfArrayMel
       (:,14,:)-pdfArrayMel(:,11,:)).^2+(pdfArrayMel(:,15,:)-pdfArrayMel(:,12,:)).^2+(pdfArrayMel
       (:,16,:)-pdfArrayMel(:,13,:)).^2);
39 pdfArrayMel(:,26,:) = (pdfArrayMel(:,24,:)+pdfArrayMel(:,24,:))/2;
40
41 cfArrayMel(:,24,:) = sqrt((cfArrayMel(:,8,:)-cfArrayMel(:,5,:)).^2+(cfArrayMel(:,9,:)-
       cfArrayMel(:,6,:)).^2+(cfArrayMel(:,10,:)-cfArrayMel(:,7,:)).^2) + sqrt ((cfArrayMel
       (:,11,:)-cfArrayMel(:,8,:)).^2+(cfArrayMel(:,12,:)-cfArrayMel(:,9,:)).^2+(cfArrayMel
       (:,13,:)-cfArrayMel(:,10,:)).^2);
42 cfArrayMel(:,25,:) = sqrt((cfArrayMel(:,17,:)-cfArrayMel(:,14,:)).^2+(cfArrayMel(:,18,:)-
       cfArrayMel(:,15,:)).^2+(cfArrayMel(:,19,:)-cfArrayMel(:,16,:)).^2) + sqrt ((cfArrayMel
       (:,14,:)-cfArrayMel(:,11,:)).^2+(cfArrayMel(:,15,:)-cfArrayMel(:,12,:)).^2+(cfArrayMel
       (:,16,:)-cfArrayMel(:,13,:)).^2);
43 cfArrayMel(:,26,:) = (cfArrayMel(:,24,:)+cfArrayMel(:,24,:))/2;
44
```

```
45
46  %Furrow Opening
47  adfArrayMel(:,27,:) = sqrt((adfArrayMel(:,17,:)-adfArrayMel(:,5,:)).^2 + (adfArrayMel(:,18,:)-
        adfArrayMel(:,6,:)).^2 + (adfArrayMel(:,19,:)-adfArrayMel(:,7,:)).^2);
48  pdfArrayMel(:,27,:) = sqrt((pdfArrayMel(:,17,:)-pdfArrayMel(:,5,:)).^2 + (pdfArrayMel(:,18,:)-
        pdfArrayMel(:,6,:)).^2 + (pdfArrayMel(:,19,:)-pdfArrayMel(:,7,:)).^2);
49  cfArrayMel(:,27,:) = sqrt((cfArrayMel(:,17,:)-cfArrayMel(:,5,:)).^2 + (cfArrayMel(:,18,:)-
        cfArrayMel(:,6,:)).^2 + (cfArrayMel(:,19,:)-cfArrayMel(:,7,:)).^2);
50
51  %DV Furrow Extension
52  dvExtArrayADFMel(:,1) = adfArrayMel(:,1,1);
53  dvExtArrayPDFMel(:,1) = pdfArrayMel(:,1,1);
54  dvExtArrayCFMel(:,1) = cfArrayMel(:,1,1);
55
56  for j = 1:nEmbryoMel
57
58      dvExtArrayADFMel(find(adfArrayMel(:,2,j)>0),j+1) = 1;
59      dvExtArrayPDFMel(find(pdfArrayMel(:,2,j)>0),j+1) = 1;
60      dvExtArrayCFMel(find(cfArrayMel(:,2,j)>0),j+1) = 1;
61
62  end
63  dvExtArrayADFMel(:,(nEmbryoMel+2)) = sum(dvExtArrayADFMel(:,2:(nEmbryoMel+1)),2);
64  dvExtArrayPDFMel(:,(nEmbryoMel+2)) = sum(dvExtArrayPDFMel(:,2:(nEmbryoMel+1)),2);
65  dvExtArrayCFMel(:,(nEmbryoMel+2)) = sum(dvExtArrayCFMel(:,2:(nEmbryoMel+1)),2);
66
67  %AP Furrow Extension
68
69  apExtArrayADFMel(:,1) = min(((adfArrayMel(:,5,:)+adfArrayMel(:,17,:))./2),[],1);
70  apExtArrayADFMel(:,2) = max(((adfArrayMel(:,5,:)+adfArrayMel(:,17,:))./2),[],1);
71  apExtArrayPDFMel(:,1) = min(((pdfArrayMel(:,5,:)+pdfArrayMel(:,17,:))./2),[],1);
72  apExtArrayPDFMel(:,2) = max(((pdfArrayMel(:,5,:)+pdfArrayMel(:,17,:))./2),[],1);
73  apExtArrayCFMel(:,1) = min(((cfArrayMel(:,5,:)+cfArrayMel(:,17,:))./2),[],1);
74  apExtArrayCFMel(:,2) = max(((cfArrayMel(:,5,:)+cfArrayMel(:,17,:))./2),[],1);
75
76  %Furrow Area
77  adfArrayMel(:,28,:) = polyarea(adfArrayMel(:,5:3:17,:),adfArrayMel(:,6:3:18,:),2);
78  pdfArrayMel(:,28,:) = polyarea(pdfArrayMel(:,5:3:17,:),pdfArrayMel(:,6:3:18,:),2);
79  cfArrayMel(:,28,:) = polyarea(cfArrayMel(:,5:3:17,:),cfArrayMel(:,6:3:18,:),2);
80
81  %Furrow Volume
82
83  %Furrow Plane
84
85  adfOnEmbryoArrayMel(:,2,:) = adfArrayMel(:,5,:);
86  adfOnEmbryoArrayMel(:,3,:) = adfArrayMel(:,7,:)+(-cosd(adfArrayMel(:,1,:)).*abs(adfArrayMel
        (:,6,:)-adfArrayMel(:,7,:)));
87  adfOnEmbryoArrayMel(:,4,:) = adfArrayMel(:,7,:)+(sind(adfArrayMel(:,1,:)).*abs(adfArrayMel
        (:,6,:)-adfArrayMel(:,7,:)));
88
89  adfOnEmbryoArrayMel(:,5,:) = adfArrayMel(:,17,:);
90  adfOnEmbryoArrayMel(:,6,:) = adfArrayMel(:,19,:)+(-cosd(adfArrayMel(:,1,:)).*abs(adfArrayMel
        (:,18,:)-adfArrayMel(:,19,:)));
91  adfOnEmbryoArrayMel(:,7,:) = adfArrayMel(:,19,:)+(sind(adfArrayMel(:,1,:)).*abs(adfArrayMel
        (:,18,:)-adfArrayMel(:,19,:)));
92
93  pdfOnEmbryoArrayMel(:,2,:) = pdfArrayMel(:,5,:);
94  pdfOnEmbryoArrayMel(:,3,:) = pdfArrayMel(:,7,:)+(-cosd(pdfArrayMel(:,1,:)).*abs(pdfArrayMel
        (:,6,:)-pdfArrayMel(:,7,:)));
95  pdfOnEmbryoArrayMel(:,4,:) = pdfArrayMel(:,7,:)+(sind(pdfArrayMel(:,1,:)).*abs(pdfArrayMel
        (:,6,:)-pdfArrayMel(:,7,:)));
96
97  pdfOnEmbryoArrayMel(:,5,:) = pdfArrayMel(:,17,:);
98  pdfOnEmbryoArrayMel(:,6,:) = pdfArrayMel(:,19,:)+(-cosd(pdfArrayMel(:,1,:)).*abs(pdfArrayMel
        (:,18,:)-pdfArrayMel(:,19,:)));
```

```
99  pdfOnEmbryoArrayMel(:,7,:) = pdfArrayMel(:,19,:)+(sind(pdfArrayMel(:,1,:)).*abs(pdfArrayMel
        (:,18,:)-pdfArrayMel(:,19,:)));
100
101 cfOnEmbryoArrayMel(:,2,:) = cfArrayMel(:,5,:);
102 cfOnEmbryoArrayMel(:,3,:) = cfArrayMel(:,7,:)+(-cosd(cfArrayMel(:,1,:)).*abs(cfArrayMel(:,6,:)-
        cfArrayMel(:,7,:)));
103 cfOnEmbryoArrayMel(:,4,:) = cfArrayMel(:,7,:)+(sind(cfArrayMel(:,1,:)).*abs(cfArrayMel(:,6,:)-
        cfArrayMel(:,7,:)));
104
105 cfOnEmbryoArrayMel(:,5,:) = cfArrayMel(:,17,:);
106 cfOnEmbryoArrayMel(:,6,:) = cfArrayMel(:,19,:)+(-cosd(cfArrayMel(:,1,:)).*abs(cfArrayMel
        (:,18,:)-cfArrayMel(:,19,:)));
107 cfOnEmbryoArrayMel(:,7,:) = cfArrayMel(:,19,:)+(sind(cfArrayMel(:,1,:)).*abs(cfArrayMel(:,18,:)
        -cfArrayMel(:,19,:)));
108
109
110
111 for sigCounter = 1:nEmbryoMel
112     adfFirstSig1Mel = squeeze(find(abs(adfArrayMel(:,5,sigCounter))'>0,1,'first'));
113     adfLastSig1Mel = squeeze(find(abs(adfArrayMel(:,5,sigCounter))'>0,1,'last'));
114
115     pdfFirstSig1Mel = squeeze(find(abs(pdfArrayMel(:,5,sigCounter))'>0,1,'first'));
116     pdfLastSig1Mel = squeeze(find(abs(pdfArrayMel(:,5,sigCounter))'>0,1,'last'));
117
118     cfFirstSig1Mel = squeeze(find(abs(cfArrayMel(:,5,sigCounter))'>0,1,'first'));
119     cfLastSig1Mel = squeeze(find(abs(cfArrayMel(:,5,sigCounter))'>0,1,'last'));
120
121     if adfFirstSig1Mel>0
122         adfPlaneMel(sigCounter,:) = nanFitPlane(cat(1,adfOnEmbryoArrayMel(adfFirstSig1Mel:
                adfLastSig1Mel,2:4,sigCounter),adfOnEmbryoArrayMel(adfFirstSig1Mel:adfLastSig1Mel
                ,5:7,sigCounter)));
123     end
124     if pdfFirstSig1Mel>0
125         pdfPlaneMel(sigCounter,:) = nanFitPlane(cat(1,pdfOnEmbryoArrayMel(pdfFirstSig1Mel:
                pdfLastSig1Mel,2:4,sigCounter),pdfOnEmbryoArrayMel(pdfFirstSig1Mel:pdfLastSig1Mel
                ,5:7,sigCounter)));
126     end
127     if cfFirstSig1Mel>0
128         cfPlaneMel(sigCounter,:) = nanFitPlane(cat(1,cfOnEmbryoArrayMel(cfFirstSig1Mel:
                cfLastSig1Mel,2:4,sigCounter),cfOnEmbryoArrayMel(cfFirstSig1Mel:cfLastSig1Mel,5:7,
                sigCounter)));
129
130     end
131
132
133
134 %Furrow planarity
135  if adfFirstSig1Mel>0
136     adfOnEmbryoArrayMel(:,8,sigCounter) = distancePointPlane(adfOnEmbryoArrayMel(:,2:4,
            sigCounter), adfPlaneMel(sigCounter,:));
137     adfOnEmbryoArrayMel(:,9,sigCounter) = distancePointPlane(adfOnEmbryoArrayMel(:,5:7,
            sigCounter), adfPlaneMel(sigCounter,:));
138     adfOnEmbryoArrayMel(:,10,sigCounter) = nansum(nansum(adfOnEmbryoArrayMel(:,8:9,sigCounter))
            );
139  end
140
141  if pdfFirstSig1Mel>0
142     pdfOnEmbryoArrayMel(:,8,sigCounter) = distancePointPlane(pdfOnEmbryoArrayMel(:,2:4,
            sigCounter), pdfPlaneMel(sigCounter,:));
143     pdfOnEmbryoArrayMel(:,9,sigCounter) = distancePointPlane(pdfOnEmbryoArrayMel(:,5:7,
            sigCounter), pdfPlaneMel(sigCounter,:));
144     pdfOnEmbryoArrayMel(:,10,sigCounter) = nansum(nansum(pdfOnEmbryoArrayMel(:,8:9,sigCounter))
            );
145  end
```

```
146
147  if cfFirstSig1Mel>0
148      cfOnEmbryoArrayMel(:,8,sigCounter) = distancePointPlane(cfOnEmbryoArrayMel(:,2:4,sigCounter
             ), cfPlaneMel(sigCounter,:));
149      cfOnEmbryoArrayMel(:,9,sigCounter) = distancePointPlane(cfOnEmbryoArrayMel(:,5:7,sigCounter
             ), cfPlaneMel(sigCounter,:));
150      cfOnEmbryoArrayMel(:,10,sigCounter) = nansum(nansum(cfOnEmbryoArrayMel(:,8:9,sigCounter)));
151  end
152
153 end
154
155 meanPlaneDistMel = sqrt(nanmean(((cfOnEmbryoArrayMel(:,8,:)+cfOnEmbryoArrayMel(:,9,:))/2).^2,3)
        );
156
157 cfMeanMel(:,1,:) = (cfOnEmbryoArrayMel(:,8,:) + cfOnEmbryoArrayMel(:,9,:))/2;
158 cfMeanMel(:,2,:) = (sqrt((cfOnEmbryoArrayMel(:,2,:)).^2 + (cfOnEmbryoArrayMel(:,3,:)).^2 + (
        cfOnEmbryoArrayMel(:,4,:)).^2) + sqrt((cfOnEmbryoArrayMel(:,5,:)).^2 + (cfOnEmbryoArrayMel
        (:,6,:)).^2 + (cfOnEmbryoArrayMel(:,7,:)).^2))/2;
159
160 %for R2counter = 1:31
161     rmse = sqrt(nanmean(abs((cfOnEmbryoArrayMel(:,8,:)+cfOnEmbryoArrayMel(:,9,:))./2)).^2);
162     rsquare = 1 - nansum(cfMeanMel(:,1,:).^2)./nansum(cfMeanMel(:,2,:).^2);
163
164
165 %Furrow Angle
166 adfAnglesMel(:,1:3) = planeNormal(adfPlaneMel);
167 adfAnglesMel(:,4) = atand(adfAnglesMel(:,2)./adfAnglesMel(:,1))+90;
168
169 pdfAnglesMel(:,1:3) = planeNormal(pdfPlaneMel);
170 pdfAnglesMel(:,4) = atand(pdfAnglesMel(:,2)./pdfAnglesMel(:,1))+90;
171
172 cfAnglesMel(:,1:3) = planeNormal(cfPlaneMel);
173 cfAnglesMel(:,4) = atand(cfAnglesMel(:,2)./cfAnglesMel(:,1))+90;
174
175 %GBE Progress
176 for gbeAfillCount = 1:180
177
178     gbeProgMel(gbeAfillCount,1:21,:) = (permute(axisInputMel,[3,2,1]));
179 end
180
181 gbeProgMel(:,22,:) = (gbeProgMel(:,1,:)-gbeProgMel(:,4,:))./(gbeProgMel(:,1,:)-gbeProgMel
        (:,19,:));
182 gbeProgMel(:,23,:) = (gbeProgMel(:,1,:)-gbeProgMel(:,7,:))./(gbeProgMel(:,1,:)-gbeProgMel
        (:,19,:));
183 gbeProgMel(:,24,:) = abs((gbeProgMel(:,1,:)-(gbeProgMel(:,7,:)+gbeProgMel(:,4,:))./2)./(
        gbeProgMel(:,1,:)-gbeProgMel(:,19,:)));
184
185
186 %Inter Fold distance
187 foldDistanceMel(:,1,:) = (cfArrayMel(:,5,:)+cfArrayMel(:,17,:))./2 - gbeProgMel(:,19,:);
188 foldDistanceMel(:,2,:) = (adfArrayMel(:,5,:)+adfArrayMel(:,17,:))./2 - (cfArrayMel(:,5,:)+
        cfArrayMel(:,17,:))./2;
189 foldDistanceMel(:,3,:) = (pdfArrayMel(:,5,:)+pdfArrayMel(:,17,:))./2 - (adfArrayMel(:,5,:)+
        adfArrayMel(:,17,:))./2;
190 foldDistanceMel(:,4,:) = gbeProgMel(:,7,:) - (pdfArrayMel(:,5,:)+pdfArrayMel(:,17,:))./2;
191 foldDistanceMel(:,5,:) = gbeProgMel(:,1,:) - (pdfArrayMel(:,5,:)+pdfArrayMel(:,17,:))./2;
192 foldDistanceMel(:,6,:) = (pdfArrayMel(:,5,:)+pdfArrayMel(:,17,:))./2 - (cfArrayMel(:,5,:)+
        cfArrayMel(:,17,:))./2;
193 foldDistanceMel(90,7,:) = [3 15 16 9 6 13 9 NaN 8 6 11 7 NaN 9 8 6 8 NaN]'
194 foldDistanceMel(90,8,:) = [4 17 12 7 4 11 7 NaN 8 5 8 6 NaN 9 9 2 5 NaN]'
195
196
197 %Results
198
```

```
199 averageADFSimpleSizeMel = squeeze(nanmean(adfArrayMel(aBegDF:aEndDF,23,:)))
200 averageADFAdvancedSizeMel = squeeze(nanmean(adfArrayMel(aBegDF:aEndDF,26,:)))
201 averageADFOpeningMel = squeeze(nanmean(adfArrayMel(aBegDF:aEndDF,27,:)))
202
203 averagePDFSimpleSizeMel = squeeze(nanmean(pdfArrayMel(aBegDF:aEndDF,23,:)))
204 averagePDFAdvancedSizeMel = squeeze(nanmean(pdfArrayMel(aBegDF:aEndDF,26,:)))
205 averagePDFOpeningMel = squeeze(nanmean(pdfArrayMel(aBegDF:aEndDF,27,:)))
206
207 averageCFSimpleSizeMel = squeeze((nanmean(cfArrayMel(aBegCF:aEndCF,23,:))+nanmean(cfArrayMel((
      aBegCF+90):(aEndCF+90),23,:)))/2)
208 averageCFAdvancedSizeMel = squeeze((nanmean(cfArrayMel(aBegCF:aEndCF,26,:))+nanmean(cfArrayMel
      ((aBegCF+90):(aEndCF+90),26,:)))/2)
209 averageCFOpeningMel = squeeze((nanmean(cfArrayMel(aBegCF:aEndCF,27,:))+nanmean(cfArrayMel((
      aBegCF+90):(aEndCF+90),27,:)))/2)
210
211 averageFoldDistanceMel = squeeze(nanmean(foldDistanceMel(:,1:5,:)))';
212
213 %**************************************End Dmel************************
214
215
216 %**************************************Dpse*******************************
217
218
219
220 %Simple Depth
221
222 adfArrayPse(:,23,:) = sqrt(((adfArrayPse(:,5,:)+adfArrayPse(:,17,:))/2-adfArrayPse(:,11,:)).^2
      +  ((adfArrayPse(:,6,:)+adfArrayPse(:,18,:))/2-adfArrayPse(:,12,:)).^2 + ((adfArrayPse
      (:,7,:)+adfArrayPse(:,19,:))/2-adfArrayPse(:,13,:)).^2);
223 pdfArrayPse(:,23,:) = sqrt(((pdfArrayPse(:,5,:)+pdfArrayPse(:,17,:))/2-pdfArrayPse(:,11,:)).^2
      +  ((pdfArrayPse(:,6,:)+pdfArrayPse(:,18,:))/2-pdfArrayPse(:,12,:)).^2 + ((pdfArrayPse
      (:,7,:)+pdfArrayPse(:,19,:))/2-pdfArrayPse(:,13,:)).^2);
224 cfArrayPse(:,23,:) = sqrt(((cfArrayPse(:,5,:)+cfArrayPse(:,17,:))/2-cfArrayPse(:,11,:)).^2 +
      ((cfArrayPse(:,6,:)+cfArrayPse(:,18,:))/2-cfArrayPse(:,12,:)).^2 + ((cfArrayPse(:,7,:)+
      cfArrayPse(:,19,:))/2-cfArrayPse(:,13,:)).^2);
225
226
227 %Advanced Depth
228
229 adfArrayPse(:,24,:) = sqrt((adfArrayPse(:,8,:)-adfArrayPse(:,5,:)).^2+(adfArrayPse(:,9,:)-
      adfArrayPse(:,6,:)).^2+(adfArrayPse(:,10,:)-adfArrayPse(:,7,:)).^2) + sqrt ((adfArrayPse
      (:,11,:)-adfArrayPse(:,8,:)).^2+(adfArrayPse(:,12,:)-adfArrayPse(:,9,:)).^2+(adfArrayPse
      (:,13,:)-adfArrayPse(:,10,:)).^2);
230 adfArrayPse(:,25,:) = sqrt((adfArrayPse(:,17,:)-adfArrayPse(:,14,:)).^2+(adfArrayPse(:,18,:)-
      adfArrayPse(:,15,:)).^2+(adfArrayPse(:,19,:)-adfArrayPse(:,16,:)).^2) + sqrt ((adfArrayPse
      (:,14,:)-adfArrayPse(:,11,:)).^2+(adfArrayPse(:,15,:)-adfArrayPse(:,12,:)).^2+(adfArrayPse
      (:,16,:)-adfArrayPse(:,13,:)).^2);
231 adfArrayPse(:,26,:) = (adfArrayPse(:,24,:)+adfArrayPse(:,24,:))/2;
232
233 pdfArrayPse(:,24,:) = sqrt((pdfArrayPse(:,8,:)-pdfArrayPse(:,5,:)).^2+(pdfArrayPse(:,9,:)-
      pdfArrayPse(:,6,:)).^2+(pdfArrayPse(:,10,:)-pdfArrayPse(:,7,:)).^2) + sqrt ((pdfArrayPse
      (:,11,:)-pdfArrayPse(:,8,:)).^2+(pdfArrayPse(:,12,:)-pdfArrayPse(:,9,:)).^2+(pdfArrayPse
      (:,13,:)-pdfArrayPse(:,10,:)).^2);
234 pdfArrayPse(:,25,:) = sqrt((pdfArrayPse(:,17,:)-pdfArrayPse(:,14,:)).^2+(pdfArrayPse(:,18,:)-
      pdfArrayPse(:,15,:)).^2+(pdfArrayPse(:,19,:)-pdfArrayPse(:,16,:)).^2) + sqrt ((pdfArrayPse
      (:,14,:)-pdfArrayPse(:,11,:)).^2+(pdfArrayPse(:,15,:)-pdfArrayPse(:,12,:)).^2+(pdfArrayPse
      (:,16,:)-pdfArrayPse(:,13,:)).^2);
235 pdfArrayPse(:,26,:) = (pdfArrayPse(:,24,:)+pdfArrayPse(:,24,:))/2;
236
237 cfArrayPse(:,24,:) = sqrt((cfArrayPse(:,8,:)-cfArrayPse(:,5,:)).^2+(cfArrayPse(:,9,:)-
      cfArrayPse(:,6,:)).^2+(cfArrayPse(:,10,:)-cfArrayPse(:,7,:)).^2) + sqrt ((cfArrayPse
      (:,11,:)-cfArrayPse(:,8,:)).^2+(cfArrayPse(:,12,:)-cfArrayPse(:,9,:)).^2+(cfArrayPse
      (:,13,:)-cfArrayPse(:,10,:)).^2);
```

```
238 cfArrayPse(:,25,:) = sqrt((cfArrayPse(:,17,:)-cfArrayPse(:,14,:)).^2+(cfArrayPse(:,18,:)-
        cfArrayPse(:,15,:)).^2+(cfArrayPse(:,19,:)-cfArrayPse(:,16,:)).^2) + sqrt ((cfArrayPse
        (:,14,:)-cfArrayPse(:,11,:)).^2+(cfArrayPse(:,15,:)-cfArrayPse(:,12,:)).^2+(cfArrayPse
        (:,16,:)-cfArrayPse(:,13,:)).^2);
239 cfArrayPse(:,26,:) = (cfArrayPse(:,24,:)+cfArrayPse(:,24,:))/2;
240
241 %Furrow Opening
242 adfArrayPse(:,27,:) = sqrt((adfArrayPse(:,17,:)-adfArrayPse(:,5,:)).^2 + (adfArrayPse(:,18,:)-
        adfArrayPse(:,6,:)).^2 + (adfArrayPse(:,19,:)-adfArrayPse(:,7,:)).^2);
243 pdfArrayPse(:,27,:) = sqrt((pdfArrayPse(:,17,:)-pdfArrayPse(:,5,:)).^2 + (pdfArrayPse(:,18,:)-
        pdfArrayPse(:,6,:)).^2 + (pdfArrayPse(:,19,:)-pdfArrayPse(:,7,:)).^2);
244 cfArrayPse(:,27,:) = sqrt((cfArrayPse(:,17,:)-cfArrayPse(:,5,:)).^2 + (cfArrayPse(:,18,:)-
        cfArrayPse(:,6,:)).^2 + (cfArrayPse(:,19,:)-cfArrayPse(:,7,:)).^2);
245
246 %DV Furrow Extension
247 dvExtArrayADFPse(:,1) = adfArrayPse(:,1,1);
248 dvExtArrayPDFPse(:,1) = pdfArrayPse(:,1,1);
249 dvExtArrayCFPse(:,1) = cfArrayPse(:,1,1);
250
251 for j = 1:nEmbryoPse
252
253     dvExtArrayADFPse(find(adfArrayPse(:,2,j)>0),j+1) = 1;
254     dvExtArrayPDFPse(find(pdfArrayPse(:,2,j)>0),j+1) = 1;
255     dvExtArrayCFPse(find(cfArrayPse(:,2,j)>0),j+1) = 1;
256
257 end
258 dvExtArrayADFPse(:,(nEmbryoPse+2)) = sum(dvExtArrayADFPse(:,2:(nEmbryoPse+1)),2);
259 dvExtArrayPDFPse(:,(nEmbryoPse+2)) = sum(dvExtArrayPDFPse(:,2:(nEmbryoPse+1)),2);
260 dvExtArrayCFPse(:,(nEmbryoPse+2)) = sum(dvExtArrayCFPse(:,2:(nEmbryoPse+1)),2);
261
262 %AP Furrow Extension
263
264 apExtArrayADFPse(:,1) = min(((adfArrayPse(:,5,:)+adfArrayPse(:,17,:))./2),[],1);
265 apExtArrayADFPse(:,2) = max(((adfArrayPse(:,5,:)+adfArrayPse(:,17,:))./2),[],1);
266 apExtArrayPDFPse(:,1) = min(((pdfArrayPse(:,5,:)+pdfArrayPse(:,17,:))./2),[],1);
267 apExtArrayPDFPse(:,2) = max(((pdfArrayPse(:,5,:)+pdfArrayPse(:,17,:))./2),[],1);
268 apExtArrayCFPse(:,1) = min(((cfArrayPse(:,5,:)+cfArrayPse(:,17,:))./2),[],1);
269 apExtArrayCFPse(:,2) = max(((cfArrayPse(:,5,:)+cfArrayPse(:,17,:))./2),[],1);
270
271 %Furrow Area
272 adfArrayPse(:,28,:) = polyarea(adfArrayPse(:,5:3:17,:),adfArrayPse(:,6:3:18,:),2);
273 pdfArrayPse(:,28,:) = polyarea(pdfArrayPse(:,5:3:17,:),pdfArrayPse(:,6:3:18,:),2);
274 cfArrayPse(:,28,:) = polyarea(cfArrayPse(:,5:3:17,:),cfArrayPse(:,6:3:18,:),2);
275
276 %Furrow Volume
277
278 %Furrow Plane
279
280 adfOnEmbryoArrayPse(:,2,:) = adfArrayPse(:,5,:);
281 adfOnEmbryoArrayPse(:,3,:) = adfArrayPse(:,7,:)+(-cosd(adfArrayPse(:,1,:)).*abs(adfArrayPse
        (:,6,:)-adfArrayPse(:,7,:)));
282 adfOnEmbryoArrayPse(:,4,:) = adfArrayPse(:,7,:)+(sind(adfArrayPse(:,1,:)).*abs(adfArrayPse
        (:,6,:)-adfArrayPse(:,7,:)));
283
284 adfOnEmbryoArrayPse(:,5,:) = adfArrayPse(:,17,:);
285 adfOnEmbryoArrayPse(:,6,:) = adfArrayPse(:,19,:)+(-cosd(adfArrayPse(:,1,:)).*abs(adfArrayPse
        (:,18,:)-adfArrayPse(:,19,:)));
286 adfOnEmbryoArrayPse(:,7,:) = adfArrayPse(:,19,:)+(sind(adfArrayPse(:,1,:)).*abs(adfArrayPse
        (:,18,:)-adfArrayPse(:,19,:)));
287
288 pdfOnEmbryoArrayPse(:,2,:) = pdfArrayPse(:,5,:);
289 pdfOnEmbryoArrayPse(:,3,:) = pdfArrayPse(:,7,:)+(-cosd(pdfArrayPse(:,1,:)).*abs(pdfArrayPse
        (:,6,:)-pdfArrayPse(:,7,:)));
```

```
290 pdfOnEmbryoArrayPse(:,4,:) = pdfArrayPse(:,7,:)+(sind(pdfArrayPse(:,1,:)).*abs(pdfArrayPse
      (:,6,:)-pdfArrayPse(:,7,:)));
291
292 pdfOnEmbryoArrayPse(:,5,:) = pdfArrayPse(:,17,:);
293 pdfOnEmbryoArrayPse(:,6,:) = pdfArrayPse(:,19,:)+(-cosd(pdfArrayPse(:,1,:)).*abs(pdfArrayPse
      (:,18,:)-pdfArrayPse(:,19,:)));
294 pdfOnEmbryoArrayPse(:,7,:) = pdfArrayPse(:,19,:)+(sind(pdfArrayPse(:,1,:)).*abs(pdfArrayPse
      (:,18,:)-pdfArrayPse(:,19,:)));
295
296 cfOnEmbryoArrayPse(:,2,:) = cfArrayPse(:,5,:);
297 cfOnEmbryoArrayPse(:,3,:) = cfArrayPse(:,7,:)+(-cosd(cfArrayPse(:,1,:)).*abs(cfArrayPse(:,6,:)-
      cfArrayPse(:,7,:)));
298 cfOnEmbryoArrayPse(:,4,:) = cfArrayPse(:,7,:)+(sind(cfArrayPse(:,1,:)).*abs(cfArrayPse(:,6,:)-
      cfArrayPse(:,7,:)));
299
300 cfOnEmbryoArrayPse(:,5,:) = cfArrayPse(:,17,:);
301 cfOnEmbryoArrayPse(:,6,:) = cfArrayPse(:,19,:)+(-cosd(cfArrayPse(:,1,:)).*abs(cfArrayPse
      (:,18,:)-cfArrayPse(:,19,:)));
302 cfOnEmbryoArrayPse(:,7,:) = cfArrayPse(:,19,:)+(sind(cfArrayPse(:,1,:)).*abs(cfArrayPse(:,18,:)
      -cfArrayPse(:,19,:)));
303
304
305
306 for sigCounter = 1:nEmbryoPse
307     adfFirstSig1Pse = squeeze(find(abs(adfArrayPse(:,5,sigCounter))'>0,1,'first'));
308     adfLastSig1Pse = squeeze(find(abs(adfArrayPse(:,5,sigCounter))'>0,1,'last'));
309
310     pdfFirstSig1Pse = squeeze(find(abs(pdfArrayPse(:,5,sigCounter))'>0,1,'first'));
311     pdfLastSig1Pse = squeeze(find(abs(pdfArrayPse(:,5,sigCounter))'>0,1,'last'));
312
313     cfFirstSig1Pse = squeeze(find(abs(cfArrayPse(:,5,sigCounter))'>0,1,'first'));
314     cfLastSig1Pse = squeeze(find(abs(cfArrayPse(:,5,sigCounter))'>0,1,'last'));
315
316     if adfFirstSig1Pse>0
317         adfPlanePse(sigCounter,:) = nanFitPlane(cat(1,adfOnEmbryoArrayPse(adfFirstSig1Pse:
              adfLastSig1Pse,2:4,sigCounter),adfOnEmbryoArrayPse(adfFirstSig1Pse:adfLastSig1Pse
              ,5:7,sigCounter)));
318     end
319     if pdfFirstSig1Pse>0
320         pdfPlanePse(sigCounter,:) = nanFitPlane(cat(1,pdfOnEmbryoArrayPse(pdfFirstSig1Pse:
              pdfLastSig1Pse,2:4,sigCounter),pdfOnEmbryoArrayPse(pdfFirstSig1Pse:pdfLastSig1Pse
              ,5:7,sigCounter)));
321     end
322     if cfFirstSig1Pse>0
323         cfPlanePse(sigCounter,:) = nanFitPlane(cat(1,cfOnEmbryoArrayPse(cfFirstSig1Pse:
              cfLastSig1Pse,2:4,sigCounter),cfOnEmbryoArrayPse(cfFirstSig1Pse:cfLastSig1Pse,5:7,
              sigCounter)));
324
325     end
326
327
328
329 %Furrow planarity
330  if adfFirstSig1Pse>0
331     adfOnEmbryoArrayPse(:,8,sigCounter) = distancePointPlane(adfOnEmbryoArrayPse(:,2:4,
          sigCounter), adfPlanePse(sigCounter,:));
332     adfOnEmbryoArrayPse(:,9,sigCounter) = distancePointPlane(adfOnEmbryoArrayPse(:,5:7,
          sigCounter), adfPlanePse(sigCounter,:));
333     adfOnEmbryoArrayPse(:,10,sigCounter) = nansum(nansum(adfOnEmbryoArrayPse(:,8:9,sigCounter))
          );
334  end
335
336  if pdfFirstSig1Pse>0
```

```
337       pdfOnEmbryoArrayPse(:,8,sigCounter) = distancePointPlane(pdfOnEmbryoArrayPse(:,2:4,
              sigCounter), pdfPlanePse(sigCounter,:));
338       pdfOnEmbryoArrayPse(:,9,sigCounter) = distancePointPlane(pdfOnEmbryoArrayPse(:,5:7,
              sigCounter), pdfPlanePse(sigCounter,:));
339       pdfOnEmbryoArrayPse(:,10,sigCounter) = nansum(nansum(pdfOnEmbryoArrayPse(:,8:9,sigCounter))
              );
340    end
341
342    if cfFirstSig1Pse>0
343       cfOnEmbryoArrayPse(:,8,sigCounter) = distancePointPlane(cfOnEmbryoArrayPse(:,2:4,sigCounter
              ), cfPlanePse(sigCounter,:));
344       cfOnEmbryoArrayPse(:,9,sigCounter) = distancePointPlane(cfOnEmbryoArrayPse(:,5:7,sigCounter
              ), cfPlanePse(sigCounter,:));
345       cfOnEmbryoArrayPse(:,10,sigCounter) = nansum(nansum(cfOnEmbryoArrayPse(:,8:9,sigCounter)));
346    end
347
348 end
349
350
351 %Furrow Angle
352 adfAnglesPse(:,1:3) = planeNormal(adfPlanePse);
353 adfAnglesPse(:,4) = atand(adfAnglesPse(:,2)./adfAnglesPse(:,1))+90;
354
355 pdfAnglesPse(:,1:3) = planeNormal(pdfPlanePse);
356 pdfAnglesPse(:,4) = atand(pdfAnglesPse(:,2)./pdfAnglesPse(:,1))+90;
357
358 cfAnglesPse(:,1:3) = planeNormal(cfPlanePse);
359 cfAnglesPse(:,4) = atand(cfAnglesPse(:,2)./cfAnglesPse(:,1))+90;
360
361 %GBE Progress
362 for gbeAfillCount = 1:180
363
364     gbeProgPse(gbeAfillCount,1:21,:) = (permute(axisInputPse,[3,2,1]));
365 end
366
367 gbeProgPse(:,22,:) = (gbeProgPse(:,1,:)-gbeProgPse(:,4,:))./(gbeProgPse(:,1,:)-gbeProgPse
        (:,19,:));
368 gbeProgPse(:,23,:) = (gbeProgPse(:,1,:)-gbeProgPse(:,7,:))./(gbeProgPse(:,1,:)-gbeProgPse
        (:,19,:));
369 gbeProgPse(:,24,:) = abs((gbeProgPse(:,1,:)-(gbeProgPse(:,7,:)+gbeProgPse(:,4,:))./2)./(
        gbeProgPse(:,1,:)-gbeProgPse(:,19,:)));
370
371
372 %Inter Fold distance
373 foldDistancePse(:,1,:) = (cfArrayPse(:,5,:)+cfArrayPse(:,17,:))./2 - gbeProgPse(:,19,:);
374 foldDistancePse(:,2,:) = (adfArrayPse(:,5,:)+adfArrayPse(:,17,:))./2 - (cfArrayPse(:,5,:)+
        cfArrayPse(:,17,:))./2;
375 foldDistancePse(:,3,:) = (pdfArrayPse(:,5,:)+pdfArrayPse(:,17,:))./2 - (adfArrayPse(:,5,:)+
        adfArrayPse(:,17,:))./2;
376 foldDistancePse(:,4,:) = gbeProgPse(:,7,:) - (pdfArrayPse(:,5,:)+pdfArrayPse(:,17,:))./2;
377 foldDistancePse(:,5,:) = gbeProgPse(:,1,:) - (pdfArrayPse(:,5,:)+pdfArrayPse(:,17,:))./2;
378 foldDistancePse(:,6,:) = (pdfArrayPse(:,5,:)+pdfArrayPse(:,17,:))./2 - (cfArrayPse(:,5,:)+
        cfArrayPse(:,17,:))./2;
379 foldDistancePse(90,7,:) = [14 17 10 8 NaN NaN 7 6 11 24 9]'
380 foldDistancePse(90,8,:) = [10 11 11 12 NaN NaN 5 7 10 17 6]'
381
382 %Results
383
384 averageADFSimpleSizePse = squeeze(nanmean(adfArrayPse(aBegDF:aEndDF,23,:)))
385 averageADFAdvancedSizePse = squeeze(nanmean(adfArrayPse(aBegDF:aEndDF,26,:)))
386 averageADFOpeningPse = squeeze(nanmean(adfArrayPse(aBegDF:aEndDF,27,:)))
387
388 averagePDFSimpleSizePse = squeeze(nanmean(pdfArrayPse(aBegDF:aEndDF,23,:)))
389 averagePDFAdvancedSizePse = squeeze(nanmean(pdfArrayPse(aBegDF:aEndDF,26,:)))
```

```
390 averagePDFOpeningPse = squeeze(nanmean(pdfArrayPse(aBegDF:aEndDF,27,:)))
391
392 averageCFSimpleSizePse = squeeze((nanmean(cfArrayPse(aBegCF:aEndCF,23,:))+nanmean(cfArrayPse((
        aBegCF+90):(aEndCF+90),23,:)))/2)
393 averageCFAdvancedSizePse = squeeze((nanmean(cfArrayPse(aBegCF:aEndCF,26,:))+nanmean(cfArrayPse
        ((aBegCF+90):(aEndCF+90),26,:)))/2)
394 averageCFOpeningPse = squeeze((nanmean(cfArrayPse(aBegCF:aEndCF,27,:))+nanmean(cfArrayPse((
        aBegCF+90):(aEndCF+90),27,:)))/2)
395
396 averageFoldDistancePse = squeeze(nanmean(foldDistancePse(:,1:5,:)))';
397
398 %**************************End Dpse************************************
399
400
401 %************************Dmel Rapgap1*************************************************
402
403
404 %Simple Depth
405
406 adfArrayMelRG1(:,23,:) = sqrt(((adfArrayMelRG1(:,5,:)+adfArrayMelRG1(:,17,:))/2-adfArrayMelRG1
        (:,11,:)).^2 + ((adfArrayMelRG1(:,6,:)+adfArrayMelRG1(:,18,:))/2-adfArrayMelRG1(:,12,:))
        .^2 + ((adfArrayMelRG1(:,7,:)+adfArrayMelRG1(:,19,:))/2-adfArrayMelRG1(:,13,:)).^2);
407 pdfArrayMelRG1(:,23,:) = sqrt(((pdfArrayMelRG1(:,5,:)+pdfArrayMelRG1(:,17,:))/2-pdfArrayMelRG1
        (:,11,:)).^2 + ((pdfArrayMelRG1(:,6,:)+pdfArrayMelRG1(:,18,:))/2-pdfArrayMelRG1(:,12,:))
        .^2 + ((pdfArrayMelRG1(:,7,:)+pdfArrayMelRG1(:,19,:))/2-pdfArrayMelRG1(:,13,:)).^2);
408 cfArrayMelRG1(:,23,:) = sqrt(((cfArrayMelRG1(:,5,:)+cfArrayMelRG1(:,17,:))/2-cfArrayMelRG1
        (:,11,:)).^2 + ((cfArrayMelRG1(:,6,:)+cfArrayMelRG1(:,18,:))/2-cfArrayMelRG1(:,12,:)).^2
        + ((cfArrayMelRG1(:,7,:)+cfArrayMelRG1(:,19,:))/2-cfArrayMelRG1(:,13,:)).^2);
409
410
411 %Advanced Depth
412
413 adfArrayMelRG1(:,24,:) = sqrt((adfArrayMelRG1(:,8,:)-adfArrayMelRG1(:,5,:)).^2+(adfArrayMelRG1
        (:,9,:)-adfArrayMelRG1(:,6,:)).^2+(adfArrayMelRG1(:,10,:)-adfArrayMelRG1(:,7,:)).^2) +
        sqrt ((adfArrayMelRG1(:,11,:)-adfArrayMelRG1(:,8,:)).^2+(adfArrayMelRG1(:,12,:)-
        adfArrayMelRG1(:,9,:)).^2+(adfArrayMelRG1(:,13,:)-adfArrayMelRG1(:,10,:)).^2);
414 adfArrayMelRG1(:,25,:) = sqrt((adfArrayMelRG1(:,17,:)-adfArrayMelRG1(:,14,:)).^2+(
        adfArrayMelRG1(:,18,:)-adfArrayMelRG1(:,15,:)).^2+(adfArrayMelRG1(:,19,:)-adfArrayMelRG1
        (:,16,:)).^2) + sqrt ((adfArrayMelRG1(:,14,:)-adfArrayMelRG1(:,11,:)).^2+(adfArrayMelRG1
        (:,15,:)-adfArrayMelRG1(:,12,:)).^2+(adfArrayMelRG1(:,16,:)-adfArrayMelRG1(:,13,:)).^2);
415 adfArrayMelRG1(:,26,:) = (adfArrayMelRG1(:,24,:)+adfArrayMelRG1(:,24,:))/2;
416
417 pdfArrayMelRG1(:,24,:) = sqrt((pdfArrayMelRG1(:,8,:)-pdfArrayMelRG1(:,5,:)).^2+(pdfArrayMelRG1
        (:,9,:)-pdfArrayMelRG1(:,6,:)).^2+(pdfArrayMelRG1(:,10,:)-pdfArrayMelRG1(:,7,:)).^2) +
        sqrt ((pdfArrayMelRG1(:,11,:)-pdfArrayMelRG1(:,8,:)).^2+(pdfArrayMelRG1(:,12,:)-
        pdfArrayMelRG1(:,9,:)).^2+(pdfArrayMelRG1(:,13,:)-pdfArrayMelRG1(:,10,:)).^2);
418 pdfArrayMelRG1(:,25,:) = sqrt((pdfArrayMelRG1(:,17,:)-pdfArrayMelRG1(:,14,:)).^2+(
        pdfArrayMelRG1(:,18,:)-pdfArrayMelRG1(:,15,:)).^2+(pdfArrayMelRG1(:,19,:)-pdfArrayMelRG1
        (:,16,:)).^2) + sqrt ((pdfArrayMelRG1(:,14,:)-pdfArrayMelRG1(:,11,:)).^2+(pdfArrayMelRG1
        (:,15,:)-pdfArrayMelRG1(:,12,:)).^2+(pdfArrayMelRG1(:,16,:)-pdfArrayMelRG1(:,13,:)).^2);
419 pdfArrayMelRG1(:,26,:) = (pdfArrayMelRG1(:,24,:)+pdfArrayMelRG1(:,24,:))/2;
420
421 cfArrayMelRG1(:,24,:) = sqrt((cfArrayMelRG1(:,8,:)-cfArrayMelRG1(:,5,:)).^2+(cfArrayMelRG1
        (:,9,:)-cfArrayMelRG1(:,6,:)).^2+(cfArrayMelRG1(:,10,:)-cfArrayMelRG1(:,7,:)).^2) + sqrt
        ((cfArrayMelRG1(:,11,:)-cfArrayMelRG1(:,8,:)).^2+(cfArrayMelRG1(:,12,:)-cfArrayMelRG1
        (:,9,:)).^2+(cfArrayMelRG1(:,13,:)-cfArrayMelRG1(:,10,:)).^2);
422 cfArrayMelRG1(:,25,:) = sqrt((cfArrayMelRG1(:,17,:)-cfArrayMelRG1(:,14,:)).^2+(cfArrayMelRG1
        (:,18,:)-cfArrayMelRG1(:,15,:)).^2+(cfArrayMelRG1(:,19,:)-cfArrayMelRG1(:,16,:)).^2) +
        sqrt ((cfArrayMelRG1(:,14,:)-cfArrayMelRG1(:,11,:)).^2+(cfArrayMelRG1(:,15,:)-
        cfArrayMelRG1(:,12,:)).^2+(cfArrayMelRG1(:,16,:)-cfArrayMelRG1(:,13,:)).^2);
423 cfArrayMelRG1(:,26,:) = (cfArrayMelRG1(:,24,:)+cfArrayMelRG1(:,24,:))/2;
424
425 %Simple Depth
426
```

```
427 adfArrayMelRG1(:,23,:) = sqrt(((adfArrayMelRG1(:,5,:)+adfArrayMelRG1(:,17,:))/2-adfArrayMelRG1
      (:,11,:)).^2 + ((adfArrayMelRG1(:,6,:)+adfArrayMelRG1(:,18,:))/2-adfArrayMelRG1(:,12,:))
      .^2 + ((adfArrayMelRG1(:,7,:)+adfArrayMelRG1(:,19,:))/2-adfArrayMelRG1(:,13,:)).^2);
428 pdfArrayMelRG1(:,23,:) = sqrt(((pdfArrayMelRG1(:,5,:)+pdfArrayMelRG1(:,17,:))/2-pdfArrayMelRG1
      (:,11,:)).^2 + ((pdfArrayMelRG1(:,6,:)+pdfArrayMelRG1(:,18,:))/2-pdfArrayMelRG1(:,12,:))
      .^2 + ((pdfArrayMelRG1(:,7,:)+pdfArrayMelRG1(:,19,:))/2-pdfArrayMelRG1(:,13,:)).^2);
429 cfArrayMelRG1(:,23,:) = sqrt(((cfArrayMelRG1(:,5,:)+cfArrayMelRG1(:,17,:))/2-cfArrayMelRG1
      (:,11,:)).^2 + ((cfArrayMelRG1(:,6,:)+cfArrayMelRG1(:,18,:))/2-cfArrayMelRG1(:,12,:)).^2
      + ((cfArrayMelRG1(:,7,:)+cfArrayMelRG1(:,19,:))/2-cfArrayMelRG1(:,13,:)).^2);
430
431
432 %Advanced Depth
433
434 adfArrayMelRG1(:,24,:) = sqrt((adfArrayMelRG1(:,8,:)-adfArrayMelRG1(:,5,:)).^2+(adfArrayMelRG1
      (:,9,:)-adfArrayMelRG1(:,6,:)).^2+(adfArrayMelRG1(:,10,:)-adfArrayMelRG1(:,7,:)).^2) +
      sqrt ((adfArrayMelRG1(:,11,:)-adfArrayMelRG1(:,8,:)).^2+(adfArrayMelRG1(:,12,:)-
      adfArrayMelRG1(:,9,:)).^2+(adfArrayMelRG1(:,13,:)-adfArrayMelRG1(:,10,:)).^2);
435 adfArrayMelRG1(:,25,:) = sqrt((adfArrayMelRG1(:,17,:)-adfArrayMelRG1(:,14,:)).^2+(
      adfArrayMelRG1(:,18,:)-adfArrayMelRG1(:,15,:)).^2+(adfArrayMelRG1(:,19,:)-adfArrayMelRG1
      (:,16,:)).^2) + sqrt ((adfArrayMelRG1(:,14,:)-adfArrayMelRG1(:,11,:)).^2+(adfArrayMelRG1
      (:,15,:)-adfArrayMelRG1(:,12,:)).^2+(adfArrayMelRG1(:,16,:)-adfArrayMelRG1(:,13,:)).^2);
436 adfArrayMelRG1(:,26,:) = (adfArrayMelRG1(:,24,:)+adfArrayMelRG1(:,24,:))/2;
437
438 pdfArrayMelRG1(:,24,:) = sqrt((pdfArrayMelRG1(:,8,:)-pdfArrayMelRG1(:,5,:)).^2+(pdfArrayMelRG1
      (:,9,:)-pdfArrayMelRG1(:,6,:)).^2+(pdfArrayMelRG1(:,10,:)-pdfArrayMelRG1(:,7,:)).^2) +
      sqrt ((pdfArrayMelRG1(:,11,:)-pdfArrayMelRG1(:,8,:)).^2+(pdfArrayMelRG1(:,12,:)-
      pdfArrayMelRG1(:,9,:)).^2+(pdfArrayMelRG1(:,13,:)-pdfArrayMelRG1(:,10,:)).^2);
439 pdfArrayMelRG1(:,25,:) = sqrt((pdfArrayMelRG1(:,17,:)-pdfArrayMelRG1(:,14,:)).^2+(
      pdfArrayMelRG1(:,18,:)-pdfArrayMelRG1(:,15,:)).^2+(pdfArrayMelRG1(:,19,:)-pdfArrayMelRG1
      (:,16,:)).^2) + sqrt ((pdfArrayMelRG1(:,14,:)-pdfArrayMelRG1(:,11,:)).^2+(pdfArrayMelRG1
      (:,15,:)-pdfArrayMelRG1(:,12,:)).^2+(pdfArrayMelRG1(:,16,:)-pdfArrayMelRG1(:,13,:)).^2);
440 pdfArrayMelRG1(:,26,:) = (pdfArrayMelRG1(:,24,:)+pdfArrayMelRG1(:,24,:))/2;
441
442 cfArrayMelRG1(:,24,:) = sqrt((cfArrayMelRG1(:,8,:)-cfArrayMelRG1(:,5,:)).^2+(cfArrayMelRG1
      (:,9,:)-cfArrayMelRG1(:,6,:)).^2+(cfArrayMelRG1(:,10,:)-cfArrayMelRG1(:,7,:)).^2) + sqrt
      ((cfArrayMelRG1(:,11,:)-cfArrayMelRG1(:,8,:)).^2+(cfArrayMelRG1(:,12,:)-cfArrayMelRG1
      (:,9,:)).^2+(cfArrayMelRG1(:,13,:)-cfArrayMelRG1(:,10,:)).^2);
443 cfArrayMelRG1(:,25,:) = sqrt((cfArrayMelRG1(:,17,:)-cfArrayMelRG1(:,14,:)).^2+(cfArrayMelRG1
      (:,18,:)-cfArrayMelRG1(:,15,:)).^2+(cfArrayMelRG1(:,19,:)-cfArrayMelRG1(:,16,:)).^2) +
      sqrt ((cfArrayMelRG1(:,14,:)-cfArrayMelRG1(:,11,:)).^2+(cfArrayMelRG1(:,15,:)-
      cfArrayMelRG1(:,12,:)).^2+(cfArrayMelRG1(:,16,:)-cfArrayMelRG1(:,13,:)).^2);
444 cfArrayMelRG1(:,26,:) = (cfArrayMelRG1(:,24,:)+cfArrayMelRG1(:,24,:))/2
445
446 %Furrow Opening
447 adfArrayMelRG1(:,27,:) = sqrt((adfArrayMelRG1(:,17,:)-adfArrayMelRG1(:,5,:)).^2 + (
      adfArrayMelRG1(:,18,:)-adfArrayMelRG1(:,6,:)).^2 + (adfArrayMelRG1(:,19,:)-adfArrayMelRG1
      (:,7,:)).^2);
448 pdfArrayMelRG1(:,27,:) = sqrt((pdfArrayMelRG1(:,17,:)-pdfArrayMelRG1(:,5,:)).^2 + (
      pdfArrayMelRG1(:,18,:)-pdfArrayMelRG1(:,6,:)).^2 + (pdfArrayMelRG1(:,19,:)-pdfArrayMelRG1
      (:,7,:)).^2);
449 cfArrayMelRG1(:,27,:) = sqrt((cfArrayMelRG1(:,17,:)-cfArrayMelRG1(:,5,:)).^2 + (cfArrayMelRG1
      (:,18,:)-cfArrayMelRG1(:,6,:)).^2 + (cfArrayMelRG1(:,19,:)-cfArrayMelRG1(:,7,:)).^2);
450
451 %DV Furrow Extension
452 dvExtArrayADFMelRG1(:,1) = adfArrayMelRG1(:,1,1);
453 dvExtArrayPDFMelRG1(:,1) = pdfArrayMelRG1(:,1,1);
454 dvExtArrayCFMelRG1(:,1) = cfArrayMelRG1(:,1,1);
455
456 for j = 1:nEmbryoMelRG1
457
458     dvExtArrayADFMelRG1(find(adfArrayMelRG1(:,2,j)>0),j+1) = 1;
459     dvExtArrayPDFMelRG1(find(pdfArrayMelRG1(:,2,j)>0),j+1) = 1;
460     dvExtArrayCFMelRG1(find(cfArrayMelRG1(:,2,j)>0),j+1) = 1;
461
```

```
462 end
463 dvExtArrayADFMelRG1(:,(nEmbryoMelRG1+2)) = sum(dvExtArrayADFMelRG1(:,2:(nEmbryoMelRG1+1)),2);
464 dvExtArrayPDFMelRG1(:,(nEmbryoMelRG1+2)) = sum(dvExtArrayPDFMelRG1(:,2:(nEmbryoMelRG1+1)),2);
465 dvExtArrayCFMelRG1(:,(nEmbryoMelRG1+2)) = sum(dvExtArrayCFMelRG1(:,2:(nEmbryoMelRG1+1)),2);
466
467 %AP Furrow Extension
468
469 apExtArrayADFMelRG1(:,1) = min(((adfArrayMelRG1(:,5,:)+adfArrayMelRG1(:,17,:))./2),[],1);
470 apExtArrayADFMelRG1(:,2) = max(((adfArrayMelRG1(:,5,:)+adfArrayMelRG1(:,17,:))./2),[],1);
471 apExtArrayPDFMelRG1(:,1) = min(((pdfArrayMelRG1(:,5,:)+pdfArrayMelRG1(:,17,:))./2),[],1);
472 apExtArrayPDFMelRG1(:,2) = max(((pdfArrayMelRG1(:,5,:)+pdfArrayMelRG1(:,17,:))./2),[],1);
473 apExtArrayCFMelRG1(:,1) = min(((cfArrayMelRG1(:,5,:)+cfArrayMelRG1(:,17,:))./2),[],1);
474 apExtArrayCFMelRG1(:,2) = max(((cfArrayMelRG1(:,5,:)+cfArrayMelRG1(:,17,:))./2),[],1);
475
476 %Furrow Area
477 adfArrayMelRG1(:,28,:) = polyarea(adfArrayMelRG1(:,5:3:17,:),adfArrayMelRG1(:,6:3:18,:),2);
478 pdfArrayMelRG1(:,28,:) = polyarea(pdfArrayMelRG1(:,5:3:17,:),pdfArrayMelRG1(:,6:3:18,:),2);
479 cfArrayMelRG1(:,28,:) = polyarea(cfArrayMelRG1(:,5:3:17,:),cfArrayMelRG1(:,6:3:18,:),2);
480
481 %Furrow Volume
482
483 %Furrow Plane
484
485 adfOnEmbryoArrayMelRG1(:,2,:) = adfArrayMelRG1(:,5,:);
486 adfOnEmbryoArrayMelRG1(:,3,:) = adfArrayMelRG1(:,7,:)+(-cosd(adfArrayMelRG1(:,1,:)).*abs(
        adfArrayMelRG1(:,6,:)-adfArrayMelRG1(:,7,:)));
487 adfOnEmbryoArrayMelRG1(:,4,:) = adfArrayMelRG1(:,7,:)+(sind(adfArrayMelRG1(:,1,:)).*abs(
        adfArrayMelRG1(:,6,:)-adfArrayMelRG1(:,7,:)));
488
489 adfOnEmbryoArrayMelRG1(:,5,:) = adfArrayMelRG1(:,17,:);
490 adfOnEmbryoArrayMelRG1(:,6,:) = adfArrayMelRG1(:,19,:)+(-cosd(adfArrayMelRG1(:,1,:)).*abs(
        adfArrayMelRG1(:,18,:)-adfArrayMelRG1(:,19,:)));
491 adfOnEmbryoArrayMelRG1(:,7,:) = adfArrayMelRG1(:,19,:)+(sind(adfArrayMelRG1(:,1,:)).*abs(
        adfArrayMelRG1(:,18,:)-adfArrayMelRG1(:,19,:)));
492
493 pdfOnEmbryoArrayMelRG1(:,2,:) = pdfArrayMelRG1(:,5,:);
494 pdfOnEmbryoArrayMelRG1(:,3,:) = pdfArrayMelRG1(:,7,:)+(-cosd(pdfArrayMelRG1(:,1,:)).*abs(
        pdfArrayMelRG1(:,6,:)-pdfArrayMelRG1(:,7,:)));
495 pdfOnEmbryoArrayMelRG1(:,4,:) = pdfArrayMelRG1(:,7,:)+(sind(pdfArrayMelRG1(:,1,:)).*abs(
        pdfArrayMelRG1(:,6,:)-pdfArrayMelRG1(:,7,:)));
496
497 pdfOnEmbryoArrayMelRG1(:,5,:) = pdfArrayMelRG1(:,17,:);
498 pdfOnEmbryoArrayMelRG1(:,6,:) = pdfArrayMelRG1(:,19,:)+(-cosd(pdfArrayMelRG1(:,1,:)).*abs(
        pdfArrayMelRG1(:,18,:)-pdfArrayMelRG1(:,19,:)));
499 pdfOnEmbryoArrayMelRG1(:,7,:) = pdfArrayMelRG1(:,19,:)+(sind(pdfArrayMelRG1(:,1,:)).*abs(
        pdfArrayMelRG1(:,18,:)-pdfArrayMelRG1(:,19,:)));
500
501 cfOnEmbryoArrayMelRG1(:,2,:) = cfArrayMelRG1(:,5,:);
502 cfOnEmbryoArrayMelRG1(:,3,:) = cfArrayMelRG1(:,7,:)+(-cosd(cfArrayMelRG1(:,1,:)).*abs(
        cfArrayMelRG1(:,6,:)-cfArrayMelRG1(:,7,:)));
503 cfOnEmbryoArrayMelRG1(:,4,:) = cfArrayMelRG1(:,7,:)+(sind(cfArrayMelRG1(:,1,:)).*abs(
        cfArrayMelRG1(:,6,:)-cfArrayMelRG1(:,7,:)));
504
505 cfOnEmbryoArrayMelRG1(:,5,:) = cfArrayMelRG1(:,17,:);
506 cfOnEmbryoArrayMelRG1(:,6,:) = cfArrayMelRG1(:,19,:)+(-cosd(cfArrayMelRG1(:,1,:)).*abs(
        cfArrayMelRG1(:,18,:)-cfArrayMelRG1(:,19,:)));
507 cfOnEmbryoArrayMelRG1(:,7,:) = cfArrayMelRG1(:,19,:)+(sind(cfArrayMelRG1(:,1,:)).*abs(
        cfArrayMelRG1(:,18,:)-cfArrayMelRG1(:,19,:)));
508
509
510
511 for sigCounter = 1:nEmbryoMelRG1
512     adfFirstSig1MelRG1 = squeeze(find(abs(adfArrayMelRG1(:,5,sigCounter))'>0,1,'first'));
513     adfLastSig1MelRG1 = squeeze(find(abs(adfArrayMelRG1(:,5,sigCounter))'>0,1,'last'));
```

```
514
515     pdfFirstSig1MelRG1 = squeeze(find(abs(pdfArrayMelRG1(:,5,sigCounter))'>0,1,'first'));
516     pdfLastSig1MelRG1 = squeeze(find(abs(pdfArrayMelRG1(:,5,sigCounter))'>0,1,'last'));
517
518     cfFirstSig1MelRG1 = squeeze(find(abs(cfArrayMelRG1(:,5,sigCounter))'>0,1,'first'));
519     cfLastSig1MelRG1 = squeeze(find(abs(cfArrayMelRG1(:,5,sigCounter))'>0,1,'last'));
520
521     if adfFirstSig1MelRG1>0
522         adfPlaneMelRG1(sigCounter,:) = nanFitPlane(cat(1,adfOnEmbryoArrayMelRG1(
                adfFirstSig1MelRG1:adfLastSig1MelRG1,2:4,sigCounter),adfOnEmbryoArrayMelRG1(
                adfFirstSig1MelRG1:adfLastSig1MelRG1,5:7,sigCounter)));
523     end
524     if pdfFirstSig1MelRG1>0
525         pdfPlaneMelRG1(sigCounter,:) = nanFitPlane(cat(1,pdfOnEmbryoArrayMelRG1(
                pdfFirstSig1MelRG1:pdfLastSig1MelRG1,2:4,sigCounter),pdfOnEmbryoArrayMelRG1(
                pdfFirstSig1MelRG1:pdfLastSig1MelRG1,5:7,sigCounter)));
526     end
527     if cfFirstSig1MelRG1>0
528         cfPlaneMelRG1(sigCounter,:) = nanFitPlane(cat(1,cfOnEmbryoArrayMelRG1(cfFirstSig1MelRG1
                :cfLastSig1MelRG1,2:4,sigCounter),cfOnEmbryoArrayMelRG1(cfFirstSig1MelRG1:
                cfLastSig1MelRG1,5:7,sigCounter)));
529
530     end
531
532
533
534 %Furrow planarity
535  if adfFirstSig1MelRG1>0
536     adfOnEmbryoArrayMelRG1(:,8,sigCounter) = distancePointPlane(adfOnEmbryoArrayMelRG1(:,2:4,
           sigCounter), adfPlaneMelRG1(sigCounter,:));
537     adfOnEmbryoArrayMelRG1(:,9,sigCounter) = distancePointPlane(adfOnEmbryoArrayMelRG1(:,5:7,
           sigCounter), adfPlaneMelRG1(sigCounter,:));
538     adfOnEmbryoArrayMelRG1(:,10,sigCounter) = nansum(nansum(adfOnEmbryoArrayMelRG1(:,8:9,
           sigCounter)));
539  end
540
541  if pdfFirstSig1MelRG1>0
542     pdfOnEmbryoArrayMelRG1(:,8,sigCounter) = distancePointPlane(pdfOnEmbryoArrayMelRG1(:,2:4,
           sigCounter), pdfPlaneMelRG1(sigCounter,:));
543     pdfOnEmbryoArrayMelRG1(:,9,sigCounter) = distancePointPlane(pdfOnEmbryoArrayMelRG1(:,5:7,
           sigCounter), pdfPlaneMelRG1(sigCounter,:));
544     pdfOnEmbryoArrayMelRG1(:,10,sigCounter) = nansum(nansum(pdfOnEmbryoArrayMelRG1(:,8:9,
           sigCounter)));
545  end
546
547  if cfFirstSig1MelRG1>0
548     cfOnEmbryoArrayMelRG1(:,8,sigCounter) = distancePointPlane(cfOnEmbryoArrayMelRG1(:,2:4,
           sigCounter), cfPlaneMelRG1(sigCounter,:));
549     cfOnEmbryoArrayMelRG1(:,9,sigCounter) = distancePointPlane(cfOnEmbryoArrayMelRG1(:,5:7,
           sigCounter), cfPlaneMelRG1(sigCounter,:));
550     cfOnEmbryoArrayMelRG1(:,10,sigCounter) = nansum(nansum(cfOnEmbryoArrayMelRG1(:,8:9,
           sigCounter)));
551  end
552
553 end
554
555
556 %Furrow Angle
557 adfAnglesMelRG1(:,1:3) = planeNormal(adfPlaneMelRG1);
558 adfAnglesMelRG1(:,4) = atand(adfAnglesMelRG1(:,2)./adfAnglesMelRG1(:,1))+90;
559
560 pdfAnglesMelRG1(:,1:3) = planeNormal(pdfPlaneMelRG1);
561 pdfAnglesMelRG1(:,4) = atand(pdfAnglesMelRG1(:,2)./pdfAnglesMelRG1(:,1))+90;
562
```

```
563 cfAnglesMelRG1(:,1:3) = planeNormal(cfPlaneMelRG1);
564 cfAnglesMelRG1(:,4) = atand(cfAnglesMelRG1(:,2)./cfAnglesMelRG1(:,1))+90;
565
566 %GBE Progress
567 for gbeAfillCount = 1:180
568
569     gbeProgMelRG1(gbeAfillCount,1:21,:) = (permute(axisInputMelRG1,[3,2,1]));
570 end
571
572 gbeProgMelRG1(:,22,:) = (gbeProgMelRG1(:,1,:)-gbeProgMelRG1(:,4,:))./(gbeProgMelRG1(:,1,:)-
        gbeProgMelRG1(:,19,:));
573 gbeProgMelRG1(:,23,:) = (gbeProgMelRG1(:,1,:)-gbeProgMelRG1(:,7,:))./(gbeProgMelRG1(:,1,:)-
        gbeProgMelRG1(:,19,:));
574 gbeProgMelRG1(:,24,:) = abs((gbeProgMelRG1(:,1,:)-(gbeProgMelRG1(:,7,:)+gbeProgMelRG1(:,4,:))
        ./2)./(gbeProgMelRG1(:,1,:)-gbeProgMelRG1(:,19,:)));
575
576
577 %Inter Fold distance
578 foldDistanceMelRG1(:,1,:) = (cfArrayMelRG1(:,5,:)+cfArrayMelRG1(:,17,:))./2 - gbeProgMelRG1
        (:,19,:);
579 foldDistanceMelRG1(:,2,:) = (adfArrayMelRG1(:,5,:)+adfArrayMelRG1(:,17,:))./2 - (cfArrayMelRG1
        (:,5,:)+cfArrayMelRG1(:,17,:))./2;
580 foldDistanceMelRG1(:,3,:) = (pdfArrayMelRG1(:,5,:)+pdfArrayMelRG1(:,17,:))./2 - (adfArrayMelRG1
        (:,5,:)+adfArrayMelRG1(:,17,:))./2;
581 foldDistanceMelRG1(:,4,:) = gbeProgMelRG1(:,7,:) - (pdfArrayMelRG1(:,5,:)+pdfArrayMelRG1
        (:,17,:))./2;
582 foldDistanceMelRG1(:,5,:) = gbeProgMelRG1(:,1,:) - (pdfArrayMelRG1(:,5,:)+pdfArrayMelRG1
        (:,17,:))./2;
583 foldDistanceMelRG1(:,6,:) = (pdfArrayMelRG1(:,5,:)+pdfArrayMelRG1(:,17,:))./2 - (cfArrayMelRG1
        (:,5,:)+cfArrayMelRG1(:,17,:))./2;
584 foldDistanceMelRG1(90,7,:) = [13 NaN NaN 19 16 21 8 NaN NaN 9 NaN NaN NaN]'
585 foldDistanceMelRG1(90,8,:) = [12 NaN 12 17 16 17 10 NaN NaN 11 NaN NaN NaN]'
586
587 %Results
588
589 averageADFSimpleSizeMelRG1 = squeeze(nanmean(adfArrayMelRG1(aBegDF:aEndDF,23,:)))
590 averageADFAdvancedSizeMelRG1 = squeeze(nanmean(adfArrayMelRG1(aBegDF:aEndDF,26,:)))
591 averageADFOpeningMelRG1 = squeeze(nanmean(adfArrayMelRG1(aBegDF:aEndDF,27,:)))
592
593 averagePDFSimpleSizeMelRG1 = squeeze(nanmean(pdfArrayMelRG1(aBegDF:aEndDF,23,:)))
594 averagePDFAdvancedSizeMelRG1 = squeeze(nanmean(pdfArrayMelRG1(aBegDF:aEndDF,26,:)))
595 averagePDFOpeningMelRG1 = squeeze(nanmean(pdfArrayMelRG1(aBegDF:aEndDF,27,:)))
596
597 averageCFSimpleSizeMelRG1 = squeeze((nanmean(cfArrayMelRG1(aBegCF:aEndCF,23,:))+nanmean(
        cfArrayMelRG1((aBegCF+90):(aEndCF+90),23,:)))/2)
598 averageCFAdvancedSizeMelRG1 = squeeze((nanmean(cfArrayMelRG1(aBegCF:aEndCF,26,:))+nanmean(
        cfArrayMelRG1((aBegCF+90):(aEndCF+90),26,:)))/2)
599 averageCFOpeningMelRG1 = squeeze((nanmean(cfArrayMelRG1(aBegCF:aEndCF,27,:))+nanmean(
        cfArrayMelRG1((aBegCF+90):(aEndCF+90),27,:)))/2)
600
601 averageFoldDistanceMelRG1 = squeeze(nanmean(foldDistanceMelRG1(:,1:5,:)))';
602
603 %****************************************End Dmel RapGap1**********************
604
605 figure(38)
606
607 hold on
608 scatter(squeeze(gbeProgMel(1,22,:)),0.56*(squeeze(gbeProgMel(1,4,:))-squeeze(gbeProgMel(1,7,:))
        ),'red','filled');
609 scatter(squeeze(gbeProgPse(1,22,:)),0.56*(squeeze(gbeProgPse(1,4,:))-squeeze(gbeProgPse(1,7,:))
        ),'blue','filled');
610 scatter(squeeze(gbeProgMelRG1(1,22,:)),0.56*(squeeze(gbeProgMelRG1(1,4,:))-squeeze(
        gbeProgMelRG1(1,7,:))),'green','filled');
611
```

```
612 title('PMG opening vs GBE at posterior PMG edge');
613 xlabel('Germband extension in % of egg length')
614 ylabel('Distance [µm]')
615 xlim([0 0.45]);
616 %ylim([280 460]);
617
618 hold off
619
620 figure(39)
621
622 hold on
623 scatter(squeeze(gbeProgMel(1,23,:)),0.56*(squeeze(gbeProgMel(1,4,:))-squeeze(gbeProgMel(1,7,:))
      ),'red','filled');
624 scatter(squeeze(gbeProgPse(1,23,:)),0.56*(squeeze(gbeProgPse(1,4,:))-squeeze(gbeProgPse(1,7,:))
      ),'blue','filled');
625 scatter(squeeze(gbeProgMelRG1(1,23,:)),0.56*(squeeze(gbeProgMelRG1(1,4,:))-squeeze(
      gbeProgMelRG1(1,7,:))),'green','filled');
626
627 title('PMG opening vs GBE at anterior PMG edge');
628 xlabel('Germband extension in % of egg length')
629 ylabel('Distance [µm]')
630 xlim([0 0.45]);
631 %ylim([280 460]);
632
633 hold off
634
635 figure(40)
636
637 hold on
638 scatter(squeeze(gbeProgMel(1,24,:)),0.56*(squeeze(gbeProgMel(1,4,:))-squeeze(gbeProgMel(1,7,:))
      ),'red','filled');
639 scatter(squeeze(gbeProgPse(1,24,:)),0.56*(squeeze(gbeProgPse(1,4,:))-squeeze(gbeProgPse(1,7,:))
      ),'blue','filled');
640 scatter(squeeze(gbeProgMelRG1(1,24,:)),0.56*(squeeze(gbeProgMelRG1(1,4,:))-squeeze(
      gbeProgMelRG1(1,7,:))),'green','filled');
641
642 title('PMG opening vs GBE at PMG center');
643 xlabel('Germband extension in % of egg length')
644 ylabel('Distance [µm]')
645 xlim([0 0.45]);
646 %ylim([280 460]);
647
648 hold off
```

B.2.3 Germband extension analysis

The germband analysis script was used to analyze the data obtained from manual germband tracking using a custom made macro. It transformed the Cartesian coordinates from the tracked image into cylindrical coordinates, to visualize the deviation of the germband from the dorsal and ventral midlines as angle.

```
1
2 %%GBE analysis for 'Advanced_GBE_Tracking'-Macro
3
4 clear all
5
6 %Variable declaration
7
```

```
8
9   inputGBEMelRapGap129C = NaN(100,15,17);
10  inputGBEMel29C = NaN(100,15,18);
11  inputGBEPse29C = NaN(100,15,14);
12
13  %Read txt files
14
15  melRG129CNumberArray = [2 3 4 5 6 7 8 9 11 12 13 16 17 18 19 20 21];
16
17  mel29CNumberArray = [1 3 4 5 6 7 8 10 11 12 13 14 15 16 17 18 19 20];
18
19  pse29CNumberArray = [1 3 5 6 7 8 10 11 14 16 17 18 19 20];
20
21  for i = 1:18
22
23      if i < 18
24          inputM29R = dlmread(['D:\Lucas_SPE_Data\Double_sided\Melanogaster_RapGap1_29C\
                Trackingdata\Advanced_GBE_Tracklog\Dmel_RapGap1_29C_E',num2str(
                melRG129CNumberArray(i)),'_Advanced_GBE_Tracklog.txt'],'␣');
25          inputGBEMelRapGap129C(1:(length(inputM29R)-3),1:7,i) = inputM29R(4:end,4:10);
26          polesMelRapGap129C(i,1:3) = inputM29R(1,1:3);
27          polesMelRapGap129C(i,4:6) = inputM29R(2,1:3);
28          polesMelRapGap129C(i,7:9) = inputM29R(3,1:3);
29          % axisInputMelRG1(i,:) = load(['F:\Lucas_SPE_Data\Double_sided\Melanogaster_RapGap1_29C\
                Trackingdata\Folds\TRACKLogAxis_Dmel_RapGap1_E',num2str(melRG1NumberArray(i)),'_Mat
                .txt']);
30      end
31
32      if i < 19
33          inputM29 = dlmread(['D:\Lucas_SPE_Data\Double_sided\Melanogaster_29C\Trackingdata\
                Advanced_GBE_Tracklog\Dmel_29C_E',num2str(mel29CNumberArray(i)),'
                _Advanced_GBE_Tracklog.txt'],'␣');
34          inputGBEMel29C(1:(length(inputM29)-3),1:7,i) = inputM29(4:end,4:10);
35          polesMel29C(i,1:3) = inputM29(1,1:3);
36          polesMel29C(i,4:6) = inputM29(2,1:3);
37          polesMel29C(i,7:9) = inputM29(3,1:3);
38          % axisInputMel(i,:) = load(['F:\Lucas_SPE_Data\Double_sided\Melanogaster_29C\
                Trackingdata\Folds\TRACKLogAxis_Dmel_E',num2str(melNumberArray(i)),'_Mat.txt']);
39      end
40      if i < 15
41          inputP29 = dlmread(['D:\Lucas_SPE_Data\Double_sided\Pseudoobscura_29C\Trackingdata\
                Advanced_GBE_Tracklog\Dpse_29C_E',num2str(pse29CNumberArray(i)),'
                _Advanced_GBE_Tracklog.txt'],'␣');
42          inputGBEPse29C(1:(length(inputP29)-3),1:7,i) = inputP29(4:end,4:10);
43          polesPse29C(i,1:3) = inputP29(1,1:3);
44          polesPse29C(i,4:6) = inputP29(2,1:3);
45          polesPse29C(i,7:9) = inputP29(3,1:3);
46          %axisInputPse(i,:) = load(['F:\Lucas_SPE_Data\Double_sided\Pseudoobscura_29C\
                Trackingdata\Folds\TRACKLogAxis_Dpse_E',num2str(pseNumberArray(i)),'_Mat.txt']);
47      end
48  end
49
50  %Sort into new table
51
52  calcGBEMelRapGap129C(:,1:3,1:17) = inputGBEMelRapGap129C(2:2:end,1:3,:); %Ventral points cols
        1:3, 4:6 reserved for calculations
53  calcGBEMelRapGap129C(:,11:13,1:17) = inputGBEMelRapGap129C(1:2:end,1:3,:); %Dorsal points cols
        7:9, 10:12 reserved for calculations
54  calcGBEMelRapGap129C(:,21:23,1:17) = inputGBEMelRapGap129C(2:2:end,5:7,:); %Circle params cols
        13:15
55
56  calcGBEMel29C(:,1:3,1:18) = inputGBEMel29C(2:2:end,1:3,:); %Ventral points cols 1:3, 4:6
        reserved for calculations
```

```
57 calcGBEMel29C(:,11:13,1:18) = inputGBEMel29C(1:2:end,1:3,:); %Dorsal points cols 7:9, 10:12
       reserved for calculations
58 calcGBEMel29C(:,21:23,1:18) = inputGBEMel29C(2:2:end,5:7,:); %Circle params cols 13:15
59
60 calcGBEPse29C(:,1:3,1:14) = inputGBEPse29C(2:2:end,1:3,:); %Ventral points cols 1:3, 4:6
       reserved for calculations
61 calcGBEPse29C(:,11:13,1:14) = inputGBEPse29C(1:2:end,1:3,:); %Dorsal points cols 7:9, 10:12
       reserved for calculations
62 calcGBEPse29C(:,21:23,1:14) = inputGBEPse29C(2:2:end,5:7,:); %Circle params cols 13:15
63
64 % Do calculations
65
66 %Melanogaster RapGap1
67 calcGBEMelRapGap129C(:,4,1:17) = calcGBEMelRapGap129C(:,1,1:17) - calcGBEMelRapGap129C
       (:,21,1:17);                                    %xCirc
68 calcGBEMelRapGap129C(:,5,1:17) = calcGBEMelRapGap129C(:,2,1:17) - calcGBEMelRapGap129C
       (:,22,1:17);                                    %yCirc
69 calcGBEMelRapGap129C(:,6,1:17) = sqrt((calcGBEMelRapGap129C(:,4,1:17)).^2 + (
       calcGBEMelRapGap129C(:,5,1:17)).^2);            %Radius
70 calcGBEMelRapGap129C(:,7,1:17) = atan2(calcGBEMelRapGap129C(:,4,1:17),calcGBEMelRapGap129C
       (:,5,1:17));                                    %Angle
71
72 calcGBEMelRapGap129C(:,4:7,18) = nanmean(abs(calcGBEMelRapGap129C(:,4:7,1:17)),3);
                                                       %Calculate Means
73 calcGBEMelRapGap129C(:,9:10,18) = nanvar(calcGBEMelRapGap129C(:,6:7,1:17),[],3);
                                                       %Calculate Variance
74
75 calcGBEMelRapGap129C(:,14,1:17) = calcGBEMelRapGap129C(:,11,1:17) - calcGBEMelRapGap129C
       (:,21,1:17);                                    %xCirc
76 calcGBEMelRapGap129C(:,15,1:17) = calcGBEMelRapGap129C(:,12,1:17) - calcGBEMelRapGap129C
       (:,22,1:17);                                    %yCirc
77 calcGBEMelRapGap129C(:,16,1:17) = sqrt((calcGBEMelRapGap129C(:,14,1:17)).^2 + (
       calcGBEMelRapGap129C(:,15,1:17)).^2);           %Radius
78 calcGBEMelRapGap129C(:,17,1:17) = atan2(calcGBEMelRapGap129C(:,14,1:17),-calcGBEMelRapGap129C
       (:,15,1:17));                                   %Angle
79
80 calcGBEMelRapGap129C(:,14:17,18) = nanmean(abs(calcGBEMelRapGap129C(:,14:17,1:17)),3);
                                                       %Calculate Means
81 calcGBEMelRapGap129C(:,19:20,18) = nanvar(calcGBEMelRapGap129C(:,16:17,1:17),[],3);
                                                       %Calculate Variance
82
83 calcGBEMelRapGap129C(:,24,1:17) = calcGBEMelRapGap129C(:,3,1:17) * 20 + 10;
                                                       %Axis
84
85 %Melanogaster
86 calcGBEMel29C(:,4,1:18) = calcGBEMel29C(:,1,1:18) - calcGBEMel29C(:,21,1:18);
                                %xCirc
87 calcGBEMel29C(:,5,1:18) = calcGBEMel29C(:,2,1:18) - calcGBEMel29C(:,22,1:18);
                                %yCirc
88 calcGBEMel29C(:,6,1:18) = sqrt((calcGBEMel29C(:,4,1:18)).^2 + (calcGBEMel29C(:,5,1:18)).^2);
                  %Radius
89 calcGBEMel29C(:,7,1:18) = atan2(calcGBEMel29C(:,4,1:18),calcGBEMel29C(:,5,1:18));
                             %Angle
90
91 calcGBEMel29C(:,4:7,19) = nanmean(abs(calcGBEMel29C(:,4:7,1:18)),3);
                                   %Calculate Means
92 calcGBEMel29C(:,9:10,19) = nanvar(calcGBEMel29C(:,6:7,1:18),[],3);
       %Calculate Variance
93
94 calcGBEMel29C(:,14,1:18) = calcGBEMel29C(:,11,1:18) - calcGBEMel29C(:,21,1:18);
                                %xCirc
95 calcGBEMel29C(:,15,1:18) = calcGBEMel29C(:,12,1:18) - calcGBEMel29C(:,22,1:18);
                                %yCirc
```

```
96  calcGBEMel29C(:,16,1:18) = sqrt((calcGBEMel29C(:,14,1:18)).^2 + (calcGBEMel29C(:,15,1:18)).^2);
                %Radius
97  calcGBEMel29C(:,17,1:18) = atan2(calcGBEMel29C(:,14,1:18),-calcGBEMel29C(:,15,1:18));
                          %Angle
98
99  calcGBEMel29C(:,14:17,19) = nanmean(abs(calcGBEMel29C(:,14:17,1:18)),3);
                                    %Calculate Means
100 calcGBEMel29C(:,19:20,19) = nanvar(calcGBEMel29C(:,16:17,1:18),[],3);              %Calculate
      Variance
101
102 calcGBEMel29C(:,24,1:18) = calcGBEMel29C(:,3,1:18) * 20 + 10;
                                               %Axis
103
104 %Pseudoobscura
105 calcGBEPse29C(:,4,1:14) = calcGBEPse29C(:,1,1:14) - calcGBEPse29C(:,21,1:14);
                                    %xCirc
106 calcGBEPse29C(:,5,1:14) = calcGBEPse29C(:,2,1:14) - calcGBEPse29C(:,22,1:14);
                                    %yCirc
107 calcGBEPse29C(:,6,1:14) = sqrt((calcGBEPse29C(:,4,1:14)).^2 + (calcGBEPse29C(:,5,1:14)).^2);
                %Radius
108 calcGBEPse29C(:,7,1:14) = atan2(calcGBEPse29C(:,4,1:14),calcGBEPse29C(:,5,1:14));
                              %Angle
109
110 calcGBEPse29C(:,4:7,15) = nanmean(abs(calcGBEPse29C(:,4:7,1:14)),3);
                                      %Calculate Means
111 calcGBEPse29C(:,9:10,15) = nanvar(calcGBEPse29C(:,6:7,1:14),[],3);
      %Calculate Variance
112
113 calcGBEPse29C(:,14,1:14) = calcGBEPse29C(:,11,1:14) - calcGBEPse29C(:,21,1:14);
                                %xCirc
114 calcGBEPse29C(:,15,1:14) = calcGBEPse29C(:,12,1:14) - calcGBEPse29C(:,22,1:14);
                                %yCirc
115 calcGBEPse29C(:,16,1:14) = sqrt((calcGBEPse29C(:,14,1:14)).^2 + (calcGBEPse29C(:,15,1:14)).^2);
                %Radius
116 calcGBEPse29C(:,17,1:14) = atan2(calcGBEPse29C(:,14,1:14),-calcGBEPse29C(:,15,1:14));
                          %Angle
117
118 calcGBEPse29C(:,14:17,15) = nanmean(abs(calcGBEPse29C(:,14:17,1:14)),3);
                                      %Calculate Means
119 calcGBEPse29C(:,19:20,15) = nanvar(calcGBEPse29C(:,16:17,1:14),[],3);
                                    %Calculate Variance
120
121 calcGBEPse29C(:,24,:) = calcGBEPse29C(:,3,:) * 20 + 10;
      %Axis
122
123 %Plotting
124
125 %%Polar histogramm plot
126
127 figure(60)
128 pax = polaraxes('rlim',[0 6]);
129 hold on
130 polarhistogram(calcGBEMel29C(6,7,1:18),700);
131 polarhistogram((calcGBEMel29C(6,17,1:18)+pi),700);
132 hold off
133 title('Melanogaster␣29C');
134 pax.ThetaZeroLocation = 'bottom';
135 pax.RLim = [0 1];
136 pax.RTick = [0 1];
137
138 figure(61)
139 pax = polaraxes('rlim',[0 6]);
140 hold on
141 polarhistogram(calcGBEMelRapGap129C(6,7,1:17),700);
```

```
142 polarhistogram((calcGBEMelRapGap129C(6,17,1:17)+pi),700);
143 hold off
144 title('Melanogaster Rapgap 1 29C');
145 pax.ThetaZeroLocation = 'bottom';
146 pax.RLim = [0 1];
147 pax.RTick = [0 1];
148
149 figure(62)
150 pax = polaraxes('rlim',[0 6]);
151 hold on
152 polarhistogram(calcGBEPse29C(6,7,1:14),700);
153 polarhistogram((calcGBEPse29C(6,17,1:14)+pi),700);
154 hold off
155 title('Pseudoobscura 29C');
156 pax.ThetaZeroLocation = 'bottom';
157 pax.RLim = [0 1];
158 pax.RTick = [0 1];
159
160 figure(41)
161 title('Melanogaster RapGap1 29C');
162 ylim([-2 2]);
163 hold on
164 for a = 1:17
165     plot(-calcGBEMelRapGap129C(1:50,24,a),calcGBEMelRapGap129C(1:50,7,a));
166     plot(calcGBEMelRapGap129C(1:50,24,a),calcGBEMelRapGap129C(1:50,17,a));
167 end
168 hold off
169
170 figure(42)
171 title('Melanogaster 29C');
172 ylim([-2 2]);
173 hold on
174 for a = 1:18
175     plot(-calcGBEMel29C(1:50,24,a),calcGBEMel29C(1:50,7,a));
176     plot(calcGBEMel29C(1:50,24,a),calcGBEMel29C(1:50,17,a));
177 end
178 hold off
179
180
181 figure(43)
182 title('Pseudoobscura 29C');
183 ylim([-2 2]);
184 hold on
185 for a = 1:14
186     plot(-calcGBEPse29C(1:50,24,a),calcGBEPse29C(1:50,7,a));
187     plot(calcGBEPse29C(1:50,24,a),calcGBEPse29C(1:50,17,a));
188 end
189 hold off
```

B.2.4 Tracking analysis

To be found on the CD (>1000 lines). The tracking analysis script was used to visualize the tracks and calculate and visualize parameters. The main steps are listed in the method section.