NNT : 2017SACLX065

A mathematical approach to embryonic morphogenesis based on spatio-temporal cell lineages

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'École Polytechnique

École doctorale n°573 Interfaces : approches interdisciplinaires, fondements, applications et innovation (Interfaces)
Spécialité de doctorat: Physique

Thèse présentée et soutenue à Palaiseau, le 19 octobre 2017, par

Juan Raphael Diaz Simões

Composition du Jury :

Vincent Fleury Directeur de recherches, Université Paris Diderot	Président
Jeffrey Johnson Professeur, Open University	Rapporteur
Martin Nilsson Jacobi Professeur, Chalmers Universiy of Technology	Rapporteur
Nicolas David Chargé de recherche, École Polytechnique	Examinateur
Dirk Drasdo Directeur de recherches, INRIA Rocquencourt	Examinateur
Denis Grebenkov Chargé de recherche, École Polytechnique	Directeur de thèse
Paul Bourgine Directeur honoraire, Réseau National des Systèmes Complexes	Co-Directeur de thèse
Nadine Peyriéras Directeur de recherches, CNRS Gif-sur-Yvette	Co-Directeur de thèse

Thèse de doctorat

Acknowledgements

This thesis was only possible thanks to the supervision of Denis Grebenkov, Paul Bourgine and Nadine Peyriéras. They gave me all the attention and structure I needed, and their different personal ideas and approaches offered an unique environment, that was essential for the establishment of such a diverse work.

I would also like to thank all the members of the jury for their attention and valuable feedback on my work.

I am very thankful of the patience of Anne-Marie Dujardin and Amparo Ruiz-Vera, which supported me in my uncommon administrative life and allowed me to finish this thesis.

And finally, life was made much easier with the friendship of Rafael, Pedro, Lucho, Mathieu and Igor, the warm reception of Christine, René and Annie, and Anne-Charlotte, who makes everything good.

Contents

Chapter 0

Introduction

This thesis is devoted to the phenomenon of *embryogenesis*. Embryogenesis is the process where an embryo's cells progressively organize themselves, through proliferation, motility and differentiation, into *morphogenetic fields* and compartments, that will carry out specific functions in the organism. Since these compartments have a very specific spatial organization, which is fundamental for the formation of organs and the execution of the animal's physiological functions, the study of these compartments' individuation and morphogenesis is essential to our understanding of embryogenesis.

The formation of these regions gives rise to an intermediary structure of organization between cells and the whole animal. Therefore, this is also an *emergence* phenomenon, with the appearance of a mesoscopic scale between a microscopic one and a macroscopic one. We study this process through a complex systems science approach. Two aspects of this approach are especially significant: the interaction between the parts and the whole, and the mathematical representation of objects.

The first aspect characterizes the way we study the interaction between the individual cells and the whole embryo. In this thesis, models are based on hypotheses that impose either a bottom-up causality or a top-down restriction on states. This allows the study of the embryo to not consist only in the study of individual cells.

The second aspect characterizes the method by which the study is performed, which can be described by the approximation of embryogenesis to the formal sciences. This approximation is done through the representation of the objects of study by mathematical objects, and their manipulation following either formal rules or semantic inferences, similarly to what is done in physics.

This study is made possible by reconstructed spatio-temporal cell lineages. This data is algorithmically extracted from 3D+time microscopy images and is composed by the set of cells present at each moment in time, together with their positions and genealogic relations. Therefore, we can observe how the cell lin-

eage evolves in time and how groups of cells flow to particular configurations. The quantity of cells in each data set is large and we manipulate it computationally.

Our analysis is focused on the zebrafish (*Danio rerio*), a very well studied model animal. More particularly, in a moment of its development when its spatio-temporal cell lineages consist largely on the displacement of cells, which gives a *physical* nature to the data. For this reason, together with biological reasoning, we use established physical theories as the basis for the manipulation and interpretation of the data.

Based on these observations we summarize the point of view of this thesis on embryonic morphogenesis by:

> Biological objects, observed through physics, described by mathematics, manipulated computationally, in the light of biology and physics.

The theoretical accomplishments of this project are a mathematical formalism for the manipulation of spatio-temporal lineages and three independent applications, investigating different aspects of embryogenesis. These studies provide many evidences and results that open the way to a large range of descriptive and predictive studies in morphogenesis.

The mathematical framework is based on the *formal description* of biological data. While many possibilites exist, we claim that the point of view of *measurements over temporal lineages* is the most adapted one for our goals. In this representation, one "forgets" the spatial details of the spatio-temporal lineages and represents them as a measurement over cells at each time step. In order to give a mathematical description to these temporal lineages, we use areas of mathematics that are descriptive and algebraic by nature. This framework is used in three applications.

The applications have been developed following some general guidelines, imposed by the data, which exhibit two main characteristics: the data is abundant, and while structurally homogenous, it is semantically diverse. It is abundant since the experiments producing it are reproducible at will, thus making it possible to have large numbers of analysed embryos. It is also diverse since many different animals are imaged, each one having different singular characteristics that define its embryogenesis. Therefore it is desirable, especially in an incipient domain of study, for the developed methods and theories to have as little bias and manual selection as possible. In that way, one maximizes the initial utility of the developed algorithms, opening the way for more specific and precise ones.

The first application explores the *emergence* of morphogenetic fields through the hypothesis that tissues can be characterized from quantities defined over temporal lineages. For this goal, we propose an algorithm that transforms sets of measurements into groups of cells that are temporally and genealogically

coherent. The algorithm shows in many cases a good matching between the morphology of the animal and the emerged clusters.

The second application explores *deterministic* aspects of embryogenesis, dealing with the estimation of intercellular forces from the trajectories of cells. We do this by finding the set of forces between neighboring cells that fits in the best possible way the accelerations calculated from trajectories. The patterns revealed by this algorithm are well aligned with what is expected from the formation of many supracellular structures.

The last application explores *stochastic* aspects of embryogenesis by performing a statistical study of the deviations of cell pairs. This is done through the comparison of statistical distributions and summarizing measurements of the relative movement of neighboring cells. Evidences have been found that corroborate the intuition that cells behave differently immediately after mitosis and that indicate that the relative movement of cells is not very distinct from a Brownian motion.

We proceed to the discussion of the practical aspects of the thesis. The main material accomplishment of this work is the implementation of `lineageflow`, which is an extensible ecosystem of libraries, executables and interfaces that aim to serve as a basis for a progressive collaboration between biologists, physicists, mathematicians and computer scientists. The implementation of all the mentioned applications has been done in this system. Its design is based on the following idea:

> Interdisciplinary science is made not only from the interaction of concepts but also the interaction of people with different expertise. One can optimize this interaction through the establishment of formal and practical interfaces of interaction.

We put some of its components in perpective in the following.

The application of theories to real data goes through their computational implementation. For this computation to be coherent, it is important to have guarantees that the computational manipulations follow the meaning of the theoretical manipulations. This has been done through the specification of a domain-specific language, which leverages the type system of the programming language for the interactive verification of the coherence of the computation with its mathematical semantics.

As mentioned, we aimed to create applications which are as unbiased as possible. However, while the mechanical analysis of data can be unbiased, the interpretation of the results should not be, and should be accessible to the specialists of the domain. For this purpose, we make use of our mathematical framework and provide an infrastructure allowing non-programmers to use the methods that have been developed by others. This is done through a *declarative* interface for algorithms, which allows the integration of new methods into the present ecosystem and their use through an unified user interface. The results of these

algorithms can be interpreted through their 3D visualization or their statistical plotting.

This can be interpreted as the search for a scientific modularity, allowing each person to work on its domain of expertise, but with means to offer mutual feedback. The common interface allows for this feedback to be more efficient, diminishing the need of learning new systems. Also, since the development of algorithms has a large quantity of technical prerequisites, the possibility of a certain level of autonomy makes it easier for the relation between specialists to be one of collaboration, and not one of dependence.

We summarize our main thesis by:

> The goals of performing integrative studies of multilevel complex systems, and theoretical studies through mathematical representations are not incompatible, and can provide a synergetic relation between all involved scientists.

We proceed now to a more precise description of embryogenesis.

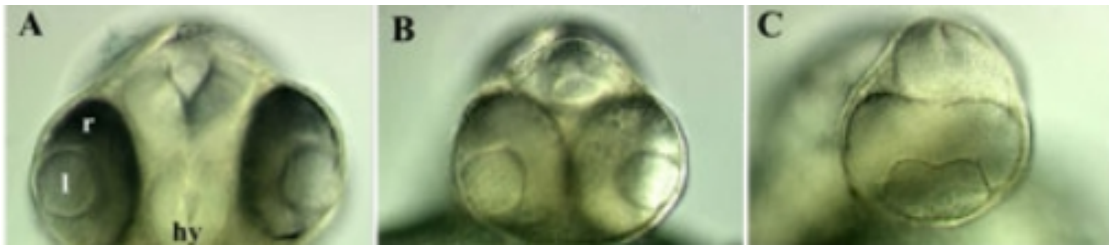

Figure 1: **Example of zebrafish mutants exhibiting various degrees of cyclopia.** Frontal views of a wild-type zebrafish embryo (A) and two embryos (B,C) with mutations affecting the formation of the ventral brain (*hy*) that is normally located between the eyes. The retinae (*r*) are partially (B) or completely (C) fused. Images by Ichiro Masai.

0.1 Embryogenesis

The process of development of an embryo is called *embryogenesis*. During this process the cells progressively organize themselves into tissues and functional compartments (Fagotto, 2014). These compartments have a particular spatial organization, which gives a characteristic *shape* to the animal. This characteristic of embryogenesis makes it an example of *morphogenesis*, the development of shapes.

The consolidation of the compartments occurs through the formation of *morphogenetic fields* (Alberts, Johnson, Lewis, & others, 2002). A morphogenetic field is a group of cells that behaves in a coherent way, leading eventually to the formation of the presumptive organs. Therefore, central to the study of embryogenesis is the study of cell behavior.

Biochemical processes inside a cell are performed and regulated by *proteins* or their complex interactions and assemblages. Some examples of the functions they perform are catalysis, cell signaling, ligand binding, active transport and structural constitution. The production of proteins is regulated by the expression of *genes*.

Since the composition of a cell, including its membranes and the elements that go through it, result from its internal processes, the genetic activity also regulates indirectly the *interaction* between cells. This has a counterpart, since external interactions can also alter the expression of genes. The factors that regulate the genetic expression of a cell can be either *genetic* or *epigenetic*.

The genetic factors are those that come from the genetic *composition* of the cell. A practical example of the effect of the genetic composition of a cell in embryogenesis is given by *mutants*. The Figure 1 shows an one-eyed pinhead (*oep*) zebrafish mutant (Zhang, Talbot, & Schier, 1998). A mutation in the *oep* gene of these specimens results in cyclopia and other formation abnormalities.

Non-genetic factors that change the activity of genes are called *epigenetic* (Holli-

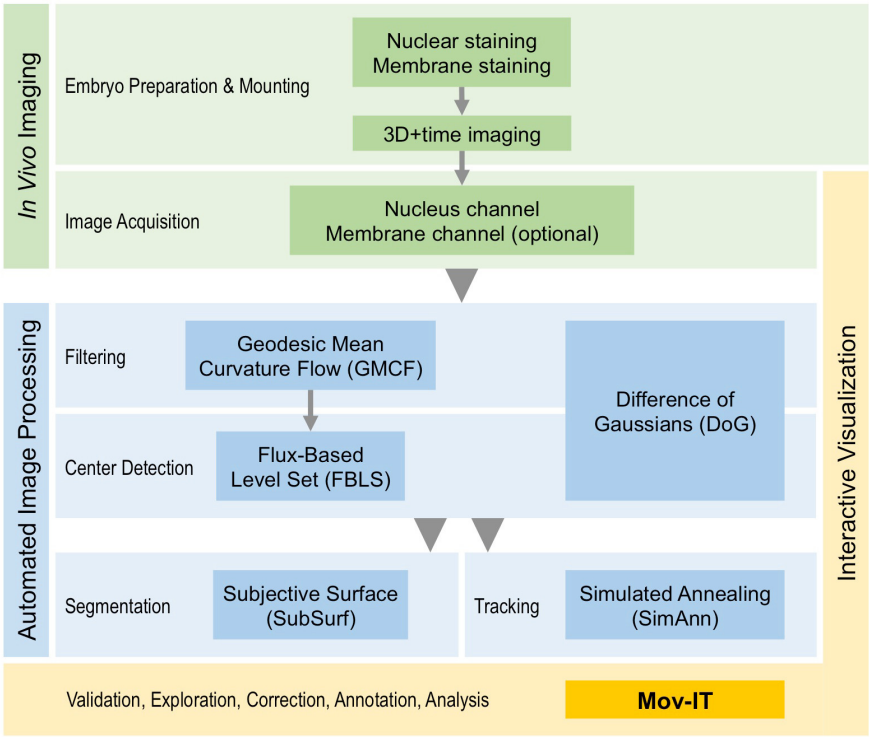

Figure 2: **BioEmergences workflow for the reconstruction of cell lineages.** The scheme representing the different phases of the reconstruction of cell lineages from 3D+time imaging data, going through nuclei center identification and cell tracking

day, 1990). There are many possible origins for these changes in genetic activity, most of which are chemical and molecular in nature. However, there are factors that are physical in nature. An example is the observed effect of mechanical forces on genetic expression (Chien, Li, & Shyy, 1998). Another example is the role of temperature on the rate of division of cells in many species (Begasse, Leaver, Vazquez, Grill, & Hyman, 2015).

In the following we discuss the data that we use for the study of embryogenesis.

Imaging data and processing workflow

The data used in this thesis comes from *in-vivo* microscopy images of several embryos analysed through the BioEmergences workflow (Faure et al., 2016). We describe here the steps performed to obtain the reconstruction of embryonic spatio-temporal cell lineages. A schema of this process is shown in Fig. 2.

The process starts with the preparation of the embryo and the staining of some biological structures, in order to enhance contrast in images. Typically, nuclei and optionally membranes are stained, with reagents that emit different frequencies that can be captured independently by the microscope.

The embryo is then imaged using techniques like multiphoton laser scanning microscopy (MLSM) and selective plane illumination microscopy (SPIM) (Fischer, Wu, Kanchanawong, Shroff, & Waterman, 2011). These techniques explore the imaging of planar sections of the embryo in order to reconstruct the 3D volume. This process is then repeated successively in order to obtain the temporal evolution of the imaged object, in the form of 3D+time images. An example of the obtained images is shown in Fig. 3. From this point, one can perform a visual exploration of the development of the imaged embryo, which is an essential part of the work of the biologist.

From these images one can algorithmically identify the positions of cell nuclei at each time step. One option is filtering the image using a geodesic mean curvature flow algorithm (Bourgine et al., 2010) and analysing the filtered image using a flux-based level set algorithm (Frolkovič & Mikula, 2007). Another is analysing the image directly using the difference of Gaussians algorithm (Rizzi & Sarti, 2009). The choice between these options depend on the species being analysed and the nature or the images. In the following chapters we use the term *cell center* for these identified points. The rate of identification of cell centers is very close to 100%, this rate being calculated in comparison to manually corrected identifications.

The final step of the analysis is the reconstruction of spatio-temporal lineages through the *tracking* of cells. This is done through the identification of the movement of cells between different time steps and the identification of mitosis. This can be done algorithmically from the cell centers identified previously using a simulated annealing algorithm (Melani et al., 2007). The precision rate of the tracking of cells is around 98% and that of mitosis identification around 50%, this rate being calculated in comparison to manually corrected identifications.

The practice of biologists involve the visualization of the embryo, augmented by the capacity of following cells and their lineages in time. This is done through a visualization software named Mov-IT (Savy, 2007). However, in this thesis, all the work is done through the algorithmical manipulation of cell centers and their tracking.

Zebrafish

We use the *zebrafish* as a model for the analysis of early embryogenesis. It can be seen in its adult phase in Fig. 4.

This species is widely studied in developmental biology due to some characteristics that are advantageous for research. Its genome has been fully sequenced, its embryonic development is very rapid and its embryos are relatively small, robust, transparent and able to develop outside the mother. Furthermore, its embryo has a fairly constant size, and is amenable to long-term imaging with nuclei and membranes staining through the expression of fluorescent proteins. A full body image of the early development of the zebrafish is shown in Fig. 5.

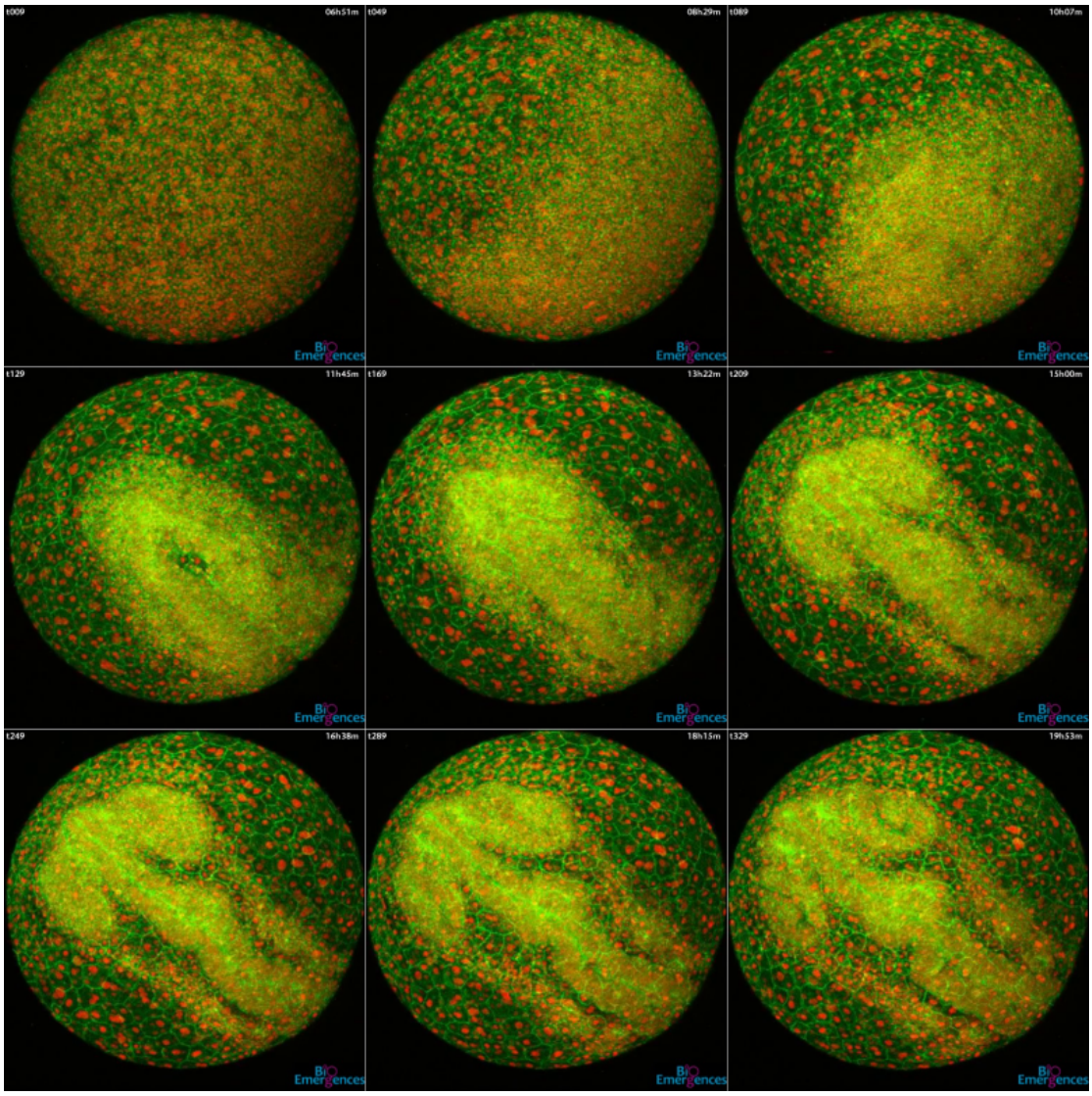

Figure 3: **Example of time lapse imaging of a developing zebrafish embryo.** These images give an example of a perspective view of the 3D volumes that compose the 3D+time images obtained from the microscopy of a wild-type zebrafish. In these images one can see the fluorescent signal emitted by nuclei (orange) and membranes (green). This embryo is imaged from 6h51 to 19h53 hours post fertilization, and the time evolution goes from left to right, from top to bottom. This data set contains around 9000 cells per frame and has a volume size of $705 \times 705 \times 241$ μm.

Figure 4: **A zebrafish adult specimen.** An adult specimen has around 4cm in length. Image by Mike Norén.

Figure 5: **Full body image of a zebrafish in its chorion.** In this image we can see a specimen 36 hours post fertilization. The zebrafish embryo develops fast and its tissues are transparent. Image by University of Guelph, Department of Integrative Biology.

The phases of the development of the zebrafish are well described (C. B. Kimmel, Ballard, Kimmel, Ullmann, & Schilling, 1995). We reproduce the table present in this article as a reference in Fig. 6.

In the data sets used in this thesis we have roughly the following phases:

- at the end of blastulation, epiboly, which consists on the spreading of cells over the yolk;
- during gastrulation, convergence and extension movements elongating the antero-posterior axis;
- in early segmentation, formation of different compartments and beginning of organogenesis.

The thickness of this embryo makes difficult to image it in full in a single volume. However, a very significant part of its body is present in the field of view of the microscope, as it can be seen in Fig. 3. Typical zebrafish data sets that are used in this thesis have:

- up to 9500 cells per time step, with cells having sizes about 15μm;
- a time interval between two 3D images of around 2min30sec, with a voxel size about 1.3μm;
- an imaged volume with dimensions about 700μm $\times$ 700 μm $\times$ 250μm and a total imaging time of 14h.

Considerations on the data

As the presented numbers show, we have to manipulare a fairly large number of cells. Besides the computational complexity that decurs from the quantity of cells, we are also confronted to *statistical* problems. Since embryogenesis is characterized by the progressive differentiation of cell behaviors, it would be unwise to treat the embryo as an homogenous whole. The data lacks the explicit information of homogeneous groups of cells, meaning that we are left with a large number of cells whose relation to each other is, *a priori*, unknown. Therefore, much care must be taken when integrating the cells into the whole for applications, which we will discuss in more detail in the following sections.

Another important limitation of the data is the information it gives on cells. This information consists of their genealogies and trajectories, which gives a kinematical nature to the data. This makes a *physical* approach to the dynamics of cells almost a necessity. For this reason, we discuss some physical aspects of embryogenesis in the following section.

Period	h	Description
Zygote	0	The newly fertilized egg through the completion of the first zygotic cell cycle
Cleavage	¾	Cell cycles 2 through 7 occur rapidly and synchronously
Blastula	2¼	Rapid, metasynchronous cell cycles (8, 9) give way to lengthened, asynchronous ones at the midblastula transition; epiboly then begins
Gastrula	5¼	Morphogenetic movements of involution, convergence, and extension form the epiblast, hypoblast, and embryonic axis; through the end of epiboly
Segmentation	10	Somites, pharyngeal arch primordia, and neuromeres develop; primary organogenesis; earliest movements; the tail appears
Pharyngula	24	Phylotypic-stage embryo; body axis straightens from its early curvature about the yolk sac; circulation, pigmentation, and fins begin development
Hatching	48	Completion of rapid morphogenesis of primary organ systems; cartilage development in head and pectoral fin; hatching occurs asynchronously
Early larva	72	Swim bladder inflates; food-seeking and active avoidance behaviors

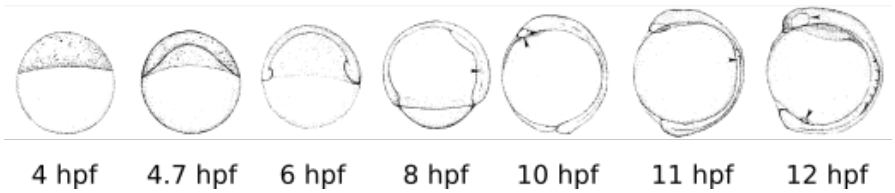

Figure 6: **Periods of early development of the zebrafish.** The times in the second column indicates the beginning of the period and is measured in hours post fertilization at 28.5°C. Our analysis is focuses the interval of time between the end of blastula up to the beginning of the segmentation period. Reproduced from (C. B. Kimmel et al., 1995).

0.2 Physical aspects of embryogenesis

Here we expose some approaches to the physical description of cell behaviors and embryogenesis, emphasizing the movement of cells and its origins. We start from the interior of the cell and increase the scale progressively.

As commented before, most activities of the cell are intermediated by proteins. We give special attention to a number of different protein assemblages here. The first are motor proteins, which are capable of converting chemical energy into mechanical work. Their study is done through the interaction of thermodynamics and hydrodynamics, exploring the chemicals origins of the energy using it for moving in a crowded environment like the interior of a cell (Howard, 2002). These structures are at the origin of the internal active transport in cells and the active movement of cells.

The propagation of local forces produced by molecular motors and external forces is done through a fiber network lying inside cells. This network, which includes the cytoskeleton, acts as a *skeleton* for the cell, providing a certain structure and rigidity. The study of these structures is mechanical by nature, dealing with the complex topologies of these networks and their composition (Ronceray, 2016). It can be seen as one of the intermediates between cell interaction and the global propagation of forces.

We proceed by the discussion of membranes and *cell adhesion*. Adhesion forces arise from a very complex interplay of ligands and receptors present in the membrane. A quantitative analysis of this process has been done in (Sackmann & Bruinsma, 2002) through the study of vesicles with reconstituted receptors and ligands that mimick the key elements of a cell surface. There one sees indications that the adhesion between the adherent shell and a target cell or tissue can be decribed by a double-well potential, each well having a different depth, whose minima correspond to weak and strong attachment. Furthermore, the interplay between membrane elasticity and attachment energy has a strong influence on the deformation of membranes during the process of adhesion.

This interplay between adhesion and elasticity and its influence in the shape of cells has been studied in (Käfer, Hayashi, Marée, Carthew, & Graner, 2007). The basis of the study is the resemblance of some epithelial tissues to packed soap bubbles. Using the analogy, the authors have been able to create a surface dependent model that reproduces the cell shapes of packs of cells in the eye of Drosophila.

Cell intercalations and the rate of change of cell shapes are important aspects of the morphogenesis of the embryo. However, they are difficult to access due to the level of detail demanded on the images for a proper analysis of these phenomena. In (Blanchard et al., 2009), the authors defined continuum measurements that correspond to cell intercalations and local change of shape for planar tissues. The establishment of these quantities allowed to show the rela-

tive importance of each process to the formation of different tissues in several animals.

The non-invasive estimation of forces between cells is an open problem in general. In (Ishihara et al., 2013), the authors compare a number of methods for the inference of forces that use the analysis of the shapes of cell membranes in planar tissues. These methods suppose the interfaces between cells to be straight lines and explore the angles of the triple points of contact.

There are studies that investigate embryogenesis using a direct fluid dynamics approach (Fleury, 2011). In this approach one identifies the local velocities of the fluid flow using particle imaging velocimetry tracking on the images of the focused region of the embryo. The produced velocity fields can be then interpreted and used for quantitative analysis. In (Fleury, 2012) these fields have been used for explaining and justifying some of the global movements of cells that induce the formation of morphological structures.

We conclude with the fact that there are many techniques for measuring cell-generated forces (Polacheck & Chen, 2016) in very different contexts. We proceed now to the approach used in this thesis for the physical exploration of embryogenesis.

0.3 Exploration of in-toto embryogenesis

The distinguishing aspect of the data used in this thesis is its position in the tradeoff between volume and detail. It contains a large volume of the embryo and consequently a large quantity of cells, constrained by the algorithmical identifiability of these cells. However, this large volume implies losses in details. Some features like membranes, cell volumes and contacts, though visible, are not algorithmically extractible at this moment and their manual identification is impractical. For this reason we cannot use many of the finer details that are known about the interaction between cells exposed in the previous sections.

We can however take advantage of the interaction between two properties of out data: cells are identified and present in large number, covering a large region of the embryo. Together with some general facts on biological data, we are able to explore the interactions between the local and global scales, in the form of bottom-up causalities and top down restrictions. We will discuss some aspects of this interaction.

We start with the fact that morphogenetic fields are composed by cells whose behaviors are similar. Let us assume the reductionist interpretation that these fields are *just* sets of cells. Under this hypothesis, we can say that the dynamics of morphogenetic fields is *determined* by that of cells. This is explored in **Chapter 5**, where we propose an algorithm that transforms sets of measurements over cells, which are meant to represent their behavior, into a segmentation of the embryo into groups of cells where these measurements are homogeneous. The algorithm explores the genetic similarity of cells belonging to the same lineage and the role of genetics in morphogenesis. This *emergence* of artificially produced morphogenetic fields allows the study of their dynamics and comparison with the morphology of the animal.

We proceed with the fact that we can estimate accelerations from the displacements of cells. Let us assume that these accelerations are originated by intercellular forces, which are local by nature. Therefore, the trajectories of cells *restrict* the possible allocations of forces between cells. As the number of cells approach the complete embryo, this restriction becomes stronger. We explore this reasoning in **Chapter 6**, where we estimate intercellular forces and stresses from the accelerations calculated from cell trajectories.

We finish with the fact that the sampling of a random variable restricts in probability its possible distributions. The larger is the sampling, the stronger is the restriction. Furthermore, one may consider a set of single samples coming from a set of identical, independent random variables as equivalent to a set of independent samples of a single random variable. We explore this idea in **Chapter 7** where we compare three different populations of pairs of cells. This comparison explores the local homogeneity of displacement fields in general, due to the existence of morphogenetic fields. This reasoning allows the inference of differences between types of pairs of cells from the sampled distributions.

We proceed by discussing the relevance of this kind of data. Its biological relevance is evident, as it provides visual aspects that are identifiable in both the local and global scales. We now argue that it is also extremely relevant to the mathematical and physical exploration of embryogenesis. Spatio-temporal lineages are the minimal observables that offer:

- the recognition of cells as *individual entities*;
- their life span of cells, which allow to identify the *coexistence* of cells;
- their genealogical relations, which allow to identify the *persistence* of the lineage;
- their positions, which allow to disambiguate cells, since two cells cannot share the same point in space.

While this set may lack details like membranes, it is structurally rich enough to allow for the *representation* of these features as measurements. This is done through the definition of a *temporal lineage*, which is the object that consists on the temporal evolution of the cell lineage. Using this construct, in the same way that one can represent positions as the measurement of vectors over the temporal lineage, one may represent membranes and cell contacts using the tools from algebraic topology, as the measurement of simplicial manifolds over the temporal lineage (Bott & Tu, 1982). That is, these limitations are contextual to the experimental and algorithmic capabilities of the moment and not to the fundamental structure of the data. We provide a deeper discussion of the subject and present the mathematical formulation of temporal lineages and measurements in **Chapter 2**.

We conclude this section by discussing the role of this thesis in the context of the BioEmergences workflow (Fig. 7). All the work done here proceeds from the phenomenological reconstruction of spatio-temporal lineages, produced by the cell identification and tracking algorithms, which can be seen as *statistics* calculated from the raw data. The algorithms presented here consist only of the transformations of these statistics or measurements into other statistics. The only algorithms that contain extra hypotheses in their formulation, being classified as theoretical reconstructions, are those presented in the application chapters **5**, **6** and **7**. These hypotheses are presented explicitly in the corresponding introductions. No simulation allowing comparisons with the raw data is done, even if we produce the means for doing so. For this reason, all the exploration of measurements is done visually and statistically.

We proceed now to the practical aspects of this thesis.

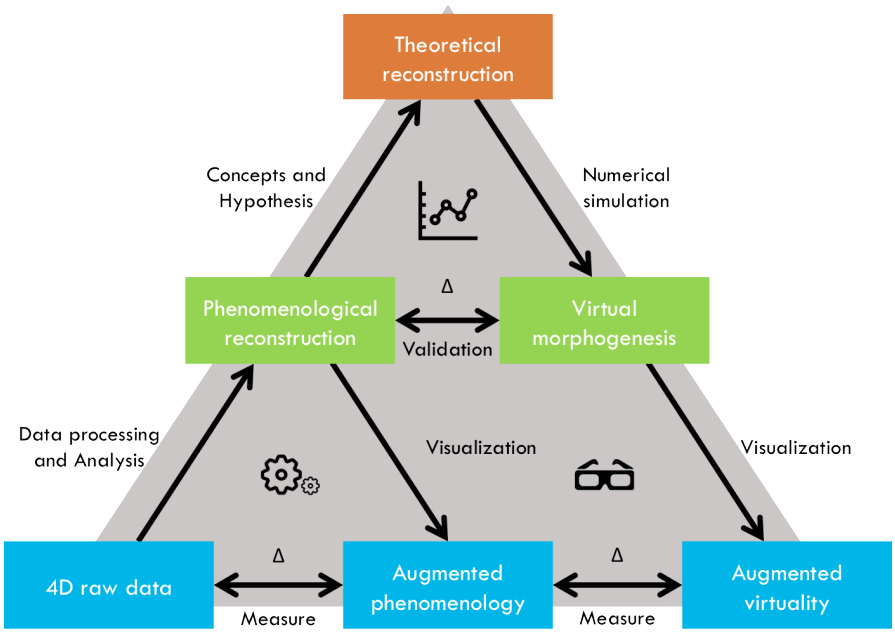

Figure 7: **Epistemological triangle summarizing the BioEmergences workflow.** This diagram organizes the procedures of reconstruction, modeling, simulation and exploration performed by the laboratory. This thesis is placed on the upper part of the triangle, performing the manipulations of statistics and the inference of observables.

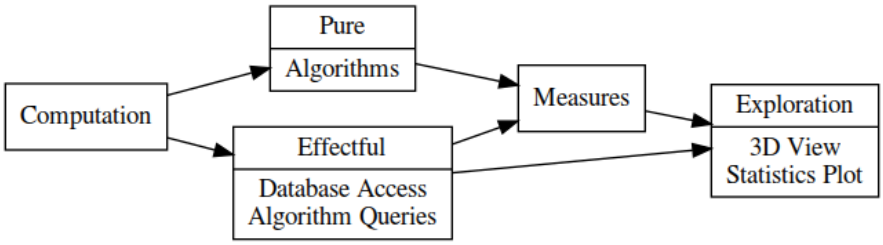

Figure 8: **Decomposition of computation.** Computation is divided in two types: pure and effectful. The interactions of these two types of computation generate measurements that can be explored visually or statistically.

0.4 Practical aspects and collaboration

We now discuss some practical aspects of the manipulation of data. This includes both the computational transformation of data and the interaction of the user with the system which performs the computation.

The manipulations and algorithms presented in this thesis are based only on the structure of the data. We also avoid making too many assumptions related to species, period of morphogenesis, etc. This is the case for two reasons:

- this field of study is recent, therefore, we prefer a strategy of progressive specification of methods to one of progressive generalization;
- the methods developed here can be applied to multiple species, which can have very different dynamics during embryogenesis.

These choices make the applications developed here prone to be used repeatedly with different data sets and by different people.

Based on this consideration, a system named `lineageflow` has been created, with the goal of accommodating users who are not programmers and being extensible by new algorithms. The ease of development, deployment and use of new methods are essential for a progressive development of the field, since they allow a more agile feedback loop between developers and embryologists and therefore, all these aspects have been considered.

We discuss here some engineering aspects of this system. We use the guiding principle of *separation of concerns* and expose the different components of this system, which are schematized in Fig. 8.

The first component of the system is a library defining an embedded domain-specific language containing the essential types and manipulations on temporal lineages and measurements. These types are derived from the mathematical formulation discussed previously, which guarantees a certain degree of coherence between theory and computation. The development of algorithms, as transformations of sets of measurements into other sets of measurements is performed in this component of the system. The algorithms are organized as

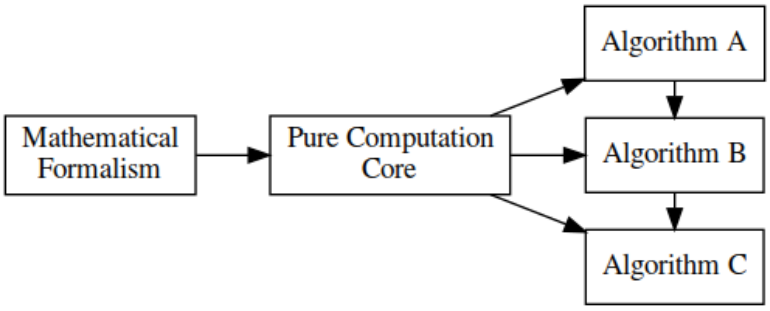

Figure 9: **Mathematical formalism and pure computational core.** The mathematical formalism is used as a specification of types and manipulations. Algorithms are defined in a declarative manner as methods that transform sets of measurements. These algorithms, due to their declarative nature, can form an ecosystem.

libraries so they can form an *ecosystem* of reusable parts, allowing the reuse of existing algorithms and the reduction of duplicated efforts. This is schematized in Fig. 9.

The second component consists of a declarative layer for the definition of algorithms. Based on these declarative definitions, the system can generate the necessary database interactions, as well as the user interface for algorithms. This makes it possible to have also an ecosystem of *executables*, which can be used directly by non-programmers. All practical results are generated from the interaction of algorithms and this component, allowing transformations to be applied to actual data.

The third component of the system is composed of an integrated 3D+time measurement viewer and a statistical plot generator. This component allows the visual and statistical exploration of measurements with no need for programming.

Finally, there is an unified graphical user interface, which integrates all algorithms and exploration tools. The automated interfaces and integrated exploration tools facilitate the communication between algorithm developers and biologists, making collaboration easier. This is schematized in Fig. 10.

This infrastucture is implemented in `lineageflow`, an ecosystem of libraries and executables, which is described in detail in **Chapter 3** and released as free software.

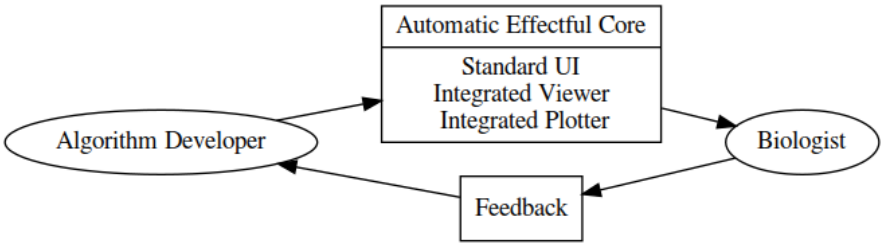

Figure 10: **Standardized effectful inteface and interaction with biologists.** The user interface for algorithms is automatically generated from their definition. This uniformization allows a faster feedback loop between algorithm developers and biologists.

0.5 Reading guide

This thesis is divided in the following parts:

- Mathematical prerequisites and formalism: **Chapters 1** and **2**;
- Computational infrastructure: **Chapter 3**;
- Reference algorithms and examples: **Chapter 4**;
- Applications: **Chapters 5**, **6** and **7**;
- General discussion: **Chapter 8**.

All the application chapters have the same structure:

- *Introduction*: general ideas and description of the study;
- *Formulation*: mathematical formulation of the procedure;
- *Implementation*: details of the computational implementation of the algorithms;
- *Results*: results obtained through the application of the algorithms to real data;
- *Discussion*: general discussion of the significance of the results and possible directions for the future.

The text is organized so that a reader may read the introduction and jump directly to the results and discussion. This is done in order to make the text more accessible for an interdisciplinary readership.

The only prerequisite for the introduction section of all chapters is this introduction. The formulation sections of applications have a prerequisite on the introduction of **Chapter 2** and the implementation sections have a weak prerequisite on **Section 3.1**. Therefore, this thesis can be read in a non-linear way.

We proceed to the description of the subject of each section of all chapters.

Chapter 1 and 2: Mathematical formalism

In the first chapter we present relations, categories and dependent types. These structures are a requisite for **Chapter 2**.

In the second chapter we introduce a mathematical formalism that has as fundamental entities *cells*, connected by their lineage and *time steps*, connected by the flow of time. These entities are used to present two equivalent point of view over the embryo:

- the temporal point of view, which is represented by sets of cells that evolve in time;
- the cellular point of view, which is represented by the time intervals where each cell is present.

These complementary points of view define a *temporal lineage*. The formal presentation of these concepts and an proof of the equivalence of these points of view is the subject of **Section 2.1**.

We proceed by the definition of measurements as functions whose domain is composed by times and cells. The classification of measurements by their domains and codomains is very useful for the implementation of the declarative interface for algorithms. This is the subject of **Section 2.2**.

Finally, we explore the algebraic structure of *cell trajectories*. The formulation of this structure is important for the development of the applications. This is the subject of **Section 2.3**.

Final considerations and possible directions of research are presented in **Section 2.4**.

Chapter 3: Computational implementation

The structure of this chapter is based on the discussion presented in the previous section.

A computation is *pure* when it consists of a transformation of data described by its inputs and outputs. Relevant to this thesis are the algorithms exposed in the application chapters, which tranform measurements or statistics into other statistics. Since we aim to represent manipulations of measurements on the embryo, we must be able to represent them computationally. This representation of measurements is derived from the mathematical formulation through *denotations*. This is the subject of **Section 3.1**.

In contrast there is an *effectful* computation, which besides its input and output, has an implicit interaction with the environment. Relevant to this thesis are database access and data visualization. The definition of measurements in the pure computational core allows a declarative interface for algorithms. This interface makes possible the automation of the interaction with the database of measurements and an universal user interface for the use of algorithms. This is the subject of **Section 3.2**.

The interaction with the database gives an easy interface for the 3D+time visualization of measurements. This is important since it allows to immediately grasp patterns in a way that is coherent with the spatio-temporal organization of the animal. Also, even if measurements are semantically very diverse, they can be represented in very similar ways. We take advantage of this structural similarity to offer a range of standard statistics without the need for programming. These capabilities are the subject of **Section 3.3**.

Final considerations and possible directions of research are presented in **Section 3.4**.

Chapter 4: Reference manipulations

When analysing measurements over the embryo, two operations are very common. The first one is the calculation of the rate of change of a measurement. This is done through *differentiation* which is the subject of **Section 4.1**.

The second one is the control of the characteristic time and space scales in order to better visualize patterns. For this purpose we should use a process which reduces the local variations of a measurement. This is done through *homogenization* which is the subject of **Section 4.2**.

While these two operations are conceptually simple, there are some details that are subtle, due to the discrete nature of cells and time steps. Furthermore, these operations serve as an example for the foundations established in the previous chapters.

Chapter 5: Clustering cell trajectories

The morphogenetic fields of an embryo are composed of cells. It is reasonable to expect that there should be a certain set of observables or measurements over cells that define the formation of these structures.

In this chapter we expose a method that serves as a tool for the search of such set of observables. This is done through the transformation of measurements over the temporal lineage into groups of cells that share similar measurements. For the exposition of the method we explore the groups of cells generated from the measurement of positions and velocities, which are shown to have a good match with the morphology of the animal.

The application of the algorithm to different sets of measures allows the transformation of a multidimensional set of measurements, each changing constantly in time, into groups of cells that are temporally and genealogically persistent. Also, this transformation makes the visualization and interpretation of the set of measurements much easier.

Chapter 6: Estimating forces from trajectories

In this chapter we explore some deterministic aspects of the development of the embryo. This is done through the prism of Newtonian mechanics.

The general idea is that from the trajectories of cells we can obtain accelerations, which are determined by *forces* in classical mechanics. We aim to find the set of forces between neighboring cells that reproduces as well as possible the observed trajectories.

While this model is too restricted for biological data, we are able to obtain interesting results and a good match with the expectations given by the formation

of tissues. Furthermore, since forces are easy to interpret, as compared to other more abstract physical concepts, they offer a good bridge between physicists and biologists.

Chapter 7: Cell trajectory deviations

In this chapter we explore some stochastic aspects of the movement of cells. This is done through the study of the relative positions of cells sharing similar environments.

Using this analysis we have been able to identify differences between the behavior of cells after mitosis, and cells in general. Also, some general summarizing statistics of the relative movement have been measured, giving strong evidences of the similarity of this process to a Brownian motion.

The results of this chapter provide some strong restrictions on possible models of noise on cell trajectories.

Chapter 8: Discussion

In this chapter we discuss some general aspects of this project, summarize the main results and propose future directions of research.

Chapter 1

Mathematical prerequisites

This chapter exposes the mathematical prerequisites for the next two chapters, which constitute the foundations used for applications. It is recommended for mathematically inclined people that are interested in the inner workings of the mathematical foundations and the denotations of computations. For the basic aspects of the mathematical representation one can read **Section 2.1** directly. Otherwise, it is not essential for the reader who is only interested in the practical results.

Since we aim to describe objects mathematically, it is natural that we use descriptive, algebraic and constructive aspects of mathematics. The sections, which are independent of each other, are organized as follows.

Section 1 is devoted to *relations*. As the name suggests, they are an abstraction of valued relations between objects. In particular, they represent the concepts of graphs and paths in graphs, which are strongly related to cell lineages and time.

In **Section 2** we proceed to the presentation of some concepts of *category theory*. Categories are suited for the definition of structured objects and transformations preserving this structure. In the next chapters we will be interested by categories whose objects are cell lineages and times.

Finally in **Section 3** we present some *dependent types*. Dependent types are suited for the definition of sets that depend on values. They are used for the representation of the evolution of the set of cells with time and the difference on the life spans of cells.

1.1 Relations

Relations are suited for the representation of graphs, which are the basis for the representation of cell lineages and times in the next chapters. The definition used here is taken from (Jost, 2015) and is slightly more general than usual. In this section we define relations and their composition, and provide some examples.

Monoids and semirings

We start by defining some algebraic structures (Golan, 1999).

Definition A *monoid* is composed of a set M, a function $\wedge : M \times M \to M$ and an element $\top \in M$ such that:

- $\wedge$ is associative: $a \wedge (b \wedge c) = (a \wedge b) \wedge c$;
- $\top$ is an unit for $\wedge$: for all $a \in M$ we have that $\top \wedge a = a \wedge \top = a$.

Definition A *semiring* is composed of a set M, two functions $\wedge, \vee : M \times M \to M$ and two elements $\top, \bot : M$ such that $(M, \wedge, \top)$ and $(M, \vee, \bot)$ are monoids and satisfy:

- $\vee$ is commutative: $a \vee b = b \vee a$;
- for all $a \in M$ we have $\bot \wedge a = a \wedge \bot = \bot$;
- the following distributive laws hold:
  - $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$;
  - $(b \vee c) \wedge a = (b \wedge a) \vee (c \wedge a)$.

We present some examples of semirings that are used in the following chapters. The semiring that is used the most in this thesis is the *logical semiring*. It is composed of the booleans $\Sigma = \{\top, \bot\}$, whose elements are spelled "true" and "false" respectively, together with the operations "and" ($\wedge$) and "or" ($\vee$). The notation above has been chosen in order to well accomodate this example.

Another important example is the *numerical semiring*. It is composed of the real numbers $\mathbb{R}$ and the operations $\wedge = \times$ and $\vee = +$, whose identities are respectively 1 and 0. While this set can be given more structure, this is not needed for the following.

The last example is the *tropical semiring*. It is composed of the extended real numbers $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$ and the operations $\wedge = +$ and $\vee = \min$, whose identities are respectively 0 and ∞.

Relations

Definition A *relation* between finite sets A and B with values in a set M is a function:

$$r : A \times B \to M$$

The set of such relations is denoted $A \overset{M}{\mapsto} B$. Every such relation has a corresponding *transpose* $r^\dagger : B \overset{M}{\mapsto} A$ defined by $r^\dagger(b, a) = r(a, b)$. A relation r is said to be *symmetric* if $r^\dagger = r$.

Definition Let $s : A \overset{M}{\mapsto} X$ and $r : X \overset{M}{\mapsto} B$, where M is a semiring. We define the *composed relation* $r \circ s : A \overset{M}{\mapsto} B$ by

$$[r \circ s](a, b) = \bigvee_{x:X} [s(a, x) \wedge r(x, b)]$$

From the associativity of $\wedge$, commutativity of $\vee$ and the distributive properties, it is possible to prove that the composition of relations is associative: $r \circ (s \circ t) = (r \circ s) \circ t$. Furthermore, by defining $\mathrm{Id}_A : A \overset{M}{\mapsto} A$ by

$$\mathrm{Id}_A(x, y) = \begin{cases} \top & \text{if } x = y \\ \bot & \text{otherwise} \end{cases}$$

we have that $r \circ \mathrm{Id}_A = r$ and $\mathrm{Id}_A \circ s = s$.

Definition Let $r : A \overset{M}{\mapsto} B$, $f : A' \to A$ and $g : B' \to B$. We define the pullback $(f, g)_* r : A' \overset{M}{\mapsto} B'$ by:

$$[(f, g)_* r](a, b) = r(f(a), f(b))$$

and it is easy to prove that $(g, f)_* \circ (\dagger) = (\dagger) \circ (f, g)_*$.

Examples of relations

Logical relations are based on the logical semiring. Typically, a logical relation $r : A \overset{\Omega}{\mapsto} B$ is interpreted as the subset of $A \times B$ such that $r(a, b) = \top$. The composition of two logical relations $s : A \overset{\Omega}{\mapsto} B$ and $r : B \overset{\Omega}{\mapsto} C$ can be rewritten as:

$$[r \circ s](a, c) = \exists b : B \Rightarrow s(a, b) \wedge r(b, c)$$

which is the same as the usual definition of the composition of relations. Furthermore, the identity logical relation corresponds to propositional equality: $\mathrm{Id}_A = \equiv_A$.

Numerical relations are based on the numerical semiring. The composition of two numerical relations $s : I \overset{\mathbb{R}}{\mapsto} J$ and $r : J \overset{\mathbb{R}}{\mapsto} K$ is given by:

$$[r \circ s](i,k) = \sum_{j:J} s(i,j) \times r(j,k)$$

which makes it clear that numerical relations behave as matrices on the free vector spaces (Fletcher, 1973) generated by the sets I, J and K. We see that composition corresponds to matrix multiplication and that the identity relation corresponds to the identity matrix.

Finally, tropical relations are based on the tropical semiring. A symmetric tropical relation is called a *dissimilarity*. A relation $r : X \overset{\overline{\mathbb{R}}}{\mapsto} X$ can be seen as a weighted graph, with the composition $r \circ r$:

$$[r \circ r](a,c) = \min_{b:X} \{r(a,b) + r(b,c)\}$$

seen as the cost of the smallest two-step path between two points.

Graphs

We now focus on logical relations from a set to itself, which can be interpreted as *graphs*. That is, the relation $r : X \overset{\Omega}{\mapsto} X$ represents the fact that an element is linked to another one by an edge. For this reason, when the relation is implicit, we will use the notation $x \to y$ to denote $r(x,y) = \top$.

Definition A *graph morphism* between graphs (X,r) and (Y,s) is a function $f : X \to Y$ such that:

$$r(x,y) \Rightarrow s(f(x), f(y))$$

Definition Let $r : X \overset{\Omega}{\mapsto} X$ be a graph. We define its *graph of paths* $\hat{r} : X \overset{\Omega}{\mapsto} X$ by:

$$\hat{r}(x,y) = \bigvee_{n=1}^{\infty} r^{\circ n}(x,y)$$

where $r^{\circ n}$ represents a relation composed with itself $n-1$ times.

Two elements $x, y : X$ are said to be *related* if either $\hat{r}(x,y)$ or $\hat{r}(y,x)$, that is, if they are connected by a path. If this is not true, we say that they are *unrelated*.

Finally, we say that a graph is *acyclic* if $\hat{r}(x,x) = \bot$.

1.2 Category theory

In this section we present some of the categorical concepts needed in the following chapters. Category theory (Barr & Wells, 2005) can be seen as the study of structured objects and transformations that preserve this structure. We will be interested in the next chapters in the structures of *times* and *cell lineages*.

Categories

Definition A *category* $\mathbf{C}$ is a collection of *objects* $O(\mathbf{C})$ and for every pair of objects $A, B : O(\mathbf{C})$, a set $\mathbf{C}(A, B)$ of *morphisms* such that:

- for every object A, there is a morphism $id_A \in \mathbf{C}(A, A)$ called the *identity*;
- for all objects A, B, C, there is a function $\circ_{ABC} : \mathbf{C}(A, B) \times \mathbf{C}(B, C) \to \mathbf{C}(A, C)$ called *composition*.

When indices can be inferred from the context, they will be omitted. These morphisms must obey the following properties:

- *identity*: $f \circ id = id \circ f = f$;
- *associativity*: $(f \circ g) \circ h = f \circ (g \circ h)$.

Some examples of categories are:

- **Set**, whose objects are sets and whose morphisms are functions;
- **Top**, whose objects are topological spaces and whose morphisms are continuous functions;
- **Graph**, whose objects are graphs and whose morphisms are graph morphisms, as defined in the previous section.

Definition Every category $\mathbf{C}$ has an *opposite category* called $\mathbf{C}^{op}$, which has the same objects but whose morphisms are inverted. That is:

- objects are preserved: $O(\mathbf{C}^{op}) = O(\mathbf{C})$;
- morphisms are inverted: $\mathbf{C}^{op}(A, B) = \mathbf{C}(B, A)$;
- composition is inverted: $f \circ g$ in $\mathbf{C}^{op}$ is equal to $g \circ f$ in $\mathbf{C}$;
- identities are preserved.

Definition An *isomorphism* is a morphism f for which there is another morphism f^{-1} satisfying both $f \circ f^{-1} = id$ and $f^{-1} \circ f = id$. If there is an isomorphism between two objects A and B we say that they are *isomorphic* and denote that by $A \cong B$.

Definition For categories $\mathbf{C}$ and $\mathbf{D}$ we can define the *product category* $\mathbf{C} \times \mathbf{D}$ whose objects are pairs of objects and whose morphisms are pairs of morphisms. That is:

- objects are pairs of objects: $O(\mathbf{C} \times \mathbf{D}) = O(\mathbf{C}) \times O(\mathbf{D})$;

- morphisms are pairs of morphisms: $[\mathbf{C}\times\mathbf{D}]((A_1, B_1), (A_2, B_2)) = \mathbf{C}(A_1, A_2)\times \mathbf{D}(B_1, B_2)$;
- identities are pairs of identities: $id_{A\times B} = (id_A, id_B)$;
- composition is done coordinatewise: $(f_1, g_1) \circ (f_2, g_2) = (f_1 \circ f_2, g_1 \circ g_2)$.

Functors

Definition A *functor* is a function between collections of objects that preserve the category structure. Formally, a functor $\mathcal{F}$ between categories $\mathbf{C}$ and $\mathbf{D}$ is:

- a function $\mathcal{F} : O(\mathbf{C}) \to O(\mathbf{C})$;
- for all A, B, a function $\mathcal{F}_{AB} : \mathbf{C}(A, B) \to \mathbf{D}(\mathcal{F}(A), \mathcal{F}(B))$.

When indices can be inferred from the context, they will be omitted. These functions must satisfy:

- $\mathcal{F}(id_A) = id_{\mathcal{F}(A)}$;
- $\mathcal{F}(f \circ g) = \mathcal{F}(f) \circ \mathcal{F}(g)$.

The simplest example of a functor is the identity functor $\mathrm{Id}_{\mathbf{C}} : \mathbf{C} \to \mathbf{C}$ which leaves both objects and morphisms unchanged. Another important example is the powerset functor $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$ which:

- transforms any set A into the set of its subsets $\mathcal{P}(A)$;
- transforms the function $f : A \to B$ into the function $\mathcal{P}(f) : \mathcal{P}(A) \to \mathcal{P}(B)$ which applies the function f elementwise.

Definition A *contravariant functor* $\mathcal{F}$ from $\mathbf{C}$ to $\mathbf{D}$ is:

- either a functor $\mathbf{C}^{op} \to \mathbf{D}$;
- or a functor $\mathbf{C} \to \mathbf{D}^{op}$.

Which means that the composition property for contravariant functors becomes:

$$\mathcal{F}(f \circ g) = \mathcal{F}(g) \circ \mathcal{F}(f)$$

The powerset functor can also be given the structure of a contravariant functor. In this case, it:

- transforms any set A into the set of its subsets $\mathcal{P}(A)$;
- transforms the function $f : A \to B$ into the function $\mathcal{P}(f) : \mathcal{P}(B) \to \mathcal{P}(A)$ which maps every set subset X of B into its preimage $f^{-1}(X)$, which is a subset of A.

Natural Transformations

Definition A natural transformation λ between two functors $\mathcal{F}, \mathcal{G} : \mathbf{C} \to \mathbf{D}$ is, for every object A, a morphism $\lambda_A : \mathbf{D}(\mathcal{F}(A), \mathcal{G}(A))$ such that the following

diagram commutes:

$$\begin{tikzcd} \mathcal{F}(A) \arrow[r, "\mathcal{F}(f)"] \arrow[d, "\lambda_A"'] & \mathcal{F}(B) \arrow[d, "\lambda_B"] \\ \mathcal{G}(A) \arrow[r, "\mathcal{G}(f)"'] & \mathcal{G}(B) \end{tikzcd}$$

that is, $\lambda_B \circ \mathcal{F}(f) = \mathcal{G}(f) \circ \lambda_A$.

An example of natural transformation is the singleton transformation $\eta : Id_{\mathbf{Set}} \to \mathcal{P}$ whose functions $\eta_A : A \to \mathcal{P}(A)$ map every element $x \in A$ to the singleton set $\{x\}$ in $\mathcal{P}(A)$.

A natural transformation λ is called a *natural isomorphism* if for every A, λ_A is an isomorphism.

Adjunctions

Let $\mathcal{F} : \mathbf{C} \to \mathbf{D}$ and $\mathcal{G} : \mathbf{D} \to \mathbf{C}$ be functors. In order to define an adjunction between them, we must first define two other functors derived from them.

The first is $\mathbf{D}(\mathcal{F}-, -) : \mathbf{C}^{op} \times \mathbf{D} \to \mathbf{Set}$ that:

- maps the pair of objects $A \times B$ to the set $\mathbf{D}(\mathcal{F}(A), B)$;
- maps the pair of morphisms (f, g) to the function $h \to g \circ h \circ \mathcal{F}(f)$.

The second is $\mathbf{C}(-, \mathcal{G}-) : \mathbf{C}^{op} \times \mathbf{D} \to \mathbf{Set}$ that:

- maps the pair of objects $A \times B$ to the set $\mathbf{C}(A, \mathcal{F}(B)$;
- maps the pair of morphisms (f, g) to the function $h \to \mathcal{G}(g) \circ h \circ f$.

Definition An *adjunction* between $\mathcal{F}$ and $\mathcal{G}$ is a natural isomorphism between $\mathbf{D}(\mathcal{F}-, -)$ and $\mathbf{C}(-, \mathcal{G}-)$. It is denoted by $\mathcal{F} \vdash \mathcal{G}$. In this case, we say that $\mathcal{F}$ is the *left adjoint* of $\mathcal{G}$, which is the *right adjoint* of $\mathcal{F}$.

Two examples of adjunctions can be given using graphs. There is a forgetful functor $\mathcal{F}_{\mathbf{Graph}} : \mathbf{Graph} \to \mathbf{Set}$ which maps every graph into its set of nodes. There are two simple ways of defining functors that generate graphs from sets.

The first is $\mathcal{F}_d : \mathbf{Set} \to \mathbf{Graph}$ which generates from the set X the *discrete graph* $d_X : X \overset{\Omega}{\mapsto} X$ defined by:

$$d_X(x, y) = \perp$$

The second one is $\mathcal{F}_f : \mathbf{Set} \to \mathbf{Graph}$ which generates from the set X the *full graph* $f_X : X \overset{\Omega}{\mapsto} X$ defined by:

$$f_X(x, y) = \top$$

It is not hard to prove that $\mathcal{F}_d \vdash \mathcal{F}_{\mathbf{Graph}}$ and that $\mathcal{F}_{\mathbf{Graph}} \vdash \mathcal{F}_f$. In general, we call a left adjoint of a forgetful functor a *free functor* and a right adjoint of a forgetful functor a *cofree functor*.

Contravariant right adjunctions

Definition A *contravariant right adjunction* between two contravariants functors $\mathcal{F} : \mathbf{C} \to \mathbf{D}$ and $\mathcal{G} : \mathbf{D} \to \mathbf{C}$ is a natural isomorphism:

$$\mathbf{D}(B, \mathcal{F}A) \cong \mathbf{C}(A, \mathcal{G}B)$$

where A varies in $\mathbf{C}$ and B varies in $\mathbf{D}$. Both are contravariant functors $\mathbf{C} \times \mathbf{D} \to \mathbf{Set}$. This concept can be reduced to an adjunction by considering the opposite category $\mathbf{D}^{op}$ in:

$$\mathbf{D}^{op}(\mathcal{F}A, B) \cong \mathbf{C}(A, \mathcal{G}B)$$

This concept is essential for the next chapter.

1.3 Dependent sums and products

As discussed previously, the number of cells in the data set change over time. Cells divide, enter inside the field of view and leave the field of view. In order to describe a structure that models suitably these characteristics, we use dependent sums and products. These structures are generalizations of products and functions and their study belongs to *type theory* (Martin-Löf, 1985).

The objects of study of type theory are called *types*, which are an alternative to sets in mathematical foundations. The judgement that an element a belongs to a type A is written $a : A$. When reasoning informally, one can interchange the words "type" and "set" freely. In order to avoid confusion, we do not use the *notation* typical of type theory for their presentation. However, we present types using the following pattern, which is typical to the domain:

- *formation*: how to create a new type;
- *introduction*: how to construct elements of the type;
- *elimination*: how to extract information from an arbitrary element of the type;
- *computation*: how introduction and elimination interact.

which we examplify below.

Examples and lambda notation

We expose the presentation pattern in two simple examples. The first one is the *product* of two types.

Definition In the presence of two types A and B, we can form their *product*, named $A \times B$. An element of this product can be constructed using an element $a : A$ and an element $b : B$, generating $(a, b) : A \times B$. We can extract information from pairs using the projections:

- p_1, which produces an element of type A from an element of type $A \times B$, by the computation rule $p_1(a, b) = a$;
- p_2, which produces an element of type B from an element of type $A \times B$, by the computation rule $p_2(a, b) = b$.

The second example is the type of *functions* between two types.

Definition In the presence of two types A and B, we can form the type of functions between them, named $A \to B$. An element of this type can be constructed with a rule that takes an $a : A$ and produces a $f(a) : B$. We represent anonymous functions using λ notation, giving:

$$\lambda x.\, f(x)$$

We can eliminate a function $A \to B$ using an element $a : A$, using function application:

$$(\lambda x.\, f(x))(a) = f(a)$$

As it can be seen, these correspond to the product of sets and functions between two sets. We proceed now to the generalization of these concepts.

Dependent sums

The word *dependent* is used when types can depend on values. In this case, we deal with a type of product which allows the type of the second element of a tuple to depend on the first one.

Definition In the presence of a type A and a rule that for every $a : A$ gives a type B_a, we can form their *dependent sum* $\sum_{a:A} B_a$. We can construct an element of this type, called a *dependent pair*, with an $a : A$ and a $b : B_a$, generating $(a, b) : \sum_{a:A} B_a$. We can extract information from dependent pairs using the projections:

- p_1, which produces an element of type A from an element of type $\sum_{a:A} B_a$, by the computation rule $p_1(a, b) = a$;
- p_2, which produces an element of type $B_{p_1(x)}$ from a $x : \sum_{a:A} B_a$, by the computation rule $p_2(a, b) = b$.

In the case where $B_a = B$ for every a we have that $\sum_{a:A} B = A \times B$.

Dependent products

Dependent products are a generalization of function spaces which allows the codomain of the function to depend on the element in the domain.

Definition In the presence of a type A and a rule that for every $a : A$ gives a type B_a, we can form their dependent product $\prod_{a:A} B_a$. We can construct an element of this type, called a *dependent function*, with an $a : A$ and a rule that returns for every such a a $b(a) : B_a$, generating:

$$\lambda x.\, b(x) : \prod_{a:A} B_a$$

We can eliminate a function $\prod_{a:A} B_a$ using an element $a : A$, by dependent function application:

$$(\lambda x.\, b(x))(a) = b(a) : B_a$$

In the case where $B_a = B$ for every a we have that $\prod_{a:A} B = A \to B$.

Dependent Currying

Currying is a relation between function and product types, which gives an equivalence:

$$A \times B \to C \approx A \to (B \to C)$$

that:

- from left to right, takes f to $\lambda a \,.\, [\lambda b \,.\, f(a,b)]$;
- from right to left, takes f to $\lambda(a,b) \,.\, f(a)(b)$.

This relation can be extended to the dependent case with:

$$\prod_{(a,b):\sum_{a:A} B_a} C_{(a,b)} \approx \prod_{a:A} \prod_{b:B_a} C_{(a,b)}$$

that:

- from left to right, takes f to $\lambda(a:A) \,.\, [\lambda(b:B_a) \,.\, f(a,b)]$;
- from right to left, takes f to $\lambda((a,b):\sum_{a:A} B_a) \,.\, f(a)(b)$.

which reduces to the more typical:

$$\sum_{a:A} B_a \to C \approx \prod_{a:A} (B_a \to C)$$

when the codomain is constant.

Lifted Application

Suppose we have for every $a : A$ an associated type B_a. A family of functions:

$$f_a : B_a \to C$$

can be assembled in order to define a function $\sum_{a:A} f_a : \sum_{a:A} B_a \to C$, defined by:

$$\left[\sum_{a:A} f_a\right](a,b) = f_a(b)$$

when the context makes it evident, we will just say Σf.

Chapter 2

Mathematical formalism

This thesis is centered around physical measurements taken over the embryo during its development. While much of the formal representation and manipulation of these measurements can be derived from the practice in physics, there are many differences. This is due to the fact that biological data are more structured and less regular at the same time. The data has an inherent structure that comes from the cell lineage and the connection between mother and children cells. But at the same time, the lifetime of cells is finite, which means we have to deal with the appearance and disappearance of cells in the data set. Furthermore, the fact that the images do not comprise the animal *in toto* gives the possibility of entrance and exit of cells from the field of view.

For these reasons, one needs to model:

- cells and cellular lineages;
- time and the lifetime of cells;
- measurements over combinations of these concepts.

This chapter presents a mathematical formalism that incorporates these features in a way that is adapted to the data, its computational implementation and applications.

All sections of this chapter are independent of each other.

Cells and lineages, time steps and time

We first have to define want we mean by a *cell*, in the sense of a mathematical concept that corresponds to a biological cell. It has the following properties.

- Cells are identifiable and unambiguous. That is, we can always tell with certainty if a cell is inside a data set or not. Furthermore, there is no doubt about the identity of each cell;
- The life of a cell is uninterrupted in time and defined by its beginning and end. After a cell quits the data set, it cannot come back;

- Cells can be created in two ways:
  - Entrance in the field of view creates a new cell;
  - Cellular division creates two new cells, called *children cells*.
- Cells can be destroyed in two ways:
  - Exit from the field of view destroys the given cell;
  - Cellular division destroys one cell, called the *mother cell*.

These properties define a number of cells that is always larger or equal to the real number of cells. This is due to the fact that the exit and reentrance of the same cell in the field of view cannot be determined from the information contained in the data set. For this reason, we are obliged to consider it as two separate entities. The set of cells, together with the relation that exists between mother and children cells is called a *lineage*.

The structure we give to *time* is simpler. In the work developed in this thesis, it is always a discrete, finite sequence of *time steps*, together with the relation of succession between them.

Measurements and temporal lineages

In this thesis, the notion of measurement implies an attribution of values to objects. This includes both measurements coming from experiences, like positions, and *statistics*, which are functions applied to experimental samples. A simple measurement that can be performed in the embryo, and is accessible to us from the spatio-temporal lineages, is the determination of the position of cells. Measuring position means that for all time steps and for all cells there is an associated position.

This measurement can be seen from two equivalent points of view:

- The *temporal point of view*: at each time step, there is a set of cells and for each of these cells, there is an associated position;
- The *cellular point of view*: each cell has an associated interval of time, corresponding to its lifetime, and for each time step in this interval, there is an associated position.

Physicists are familiar with this kind of duality. In fluid mechanics there are two different approaches for the study of fluid flow (Landau & Lifshitz, 2013):

- In the *Euler approach* measurements are written as functions on time and space;
- In the *Lagrange approach* measurements are written following a fluid element in time.

Another similar duality happens between the Heisenberg and Schrödinger pictures in quantum mechanics (Sakurai, 1994). In the first case the states are dependent of time while in the second one the operators change in time.

We denote through all this thesis the set of time steps by $\mathcal{T}$ and the set of cells in the whole data set by $\mathcal{C}$. Also, we denote by $\mathcal{C}_t$ the set of cells present at the time step t and by $\mathcal{T}_c$ the set of time steps where the cell c is present. Using this notation we represent the temporal point of view by the set:

$$\sum_{t:\mathcal{T}} \mathcal{C}_t$$

whose elements have the form (t, c), where $t \in \mathcal{T}$ and $c \in \mathcal{C}_t$. Similarly, we represent the cellular point of view by the set:

$$\sum_{c:\mathcal{C}} \mathcal{T}_c$$

whose elements have the form (c, t), where $c \in \mathcal{C}$ and $t \in \mathcal{T}_c$. Therefore, we can represent the duality of these points of view as an isomorphism:

$$\sum_{t:\mathcal{T}} \mathcal{C}_t \cong \sum_{c:\mathcal{C}} \mathcal{T}_c$$

We call this structure, which can be viewed as composed of cells-on-time-steps or time-steps-on-cells, a *temporal lineage*, which is the main subject of **Section 2.1**. This section is recommended only for mathematically inclined readers, interested in the formal interpretation of these objects and proofs of coherence of the duality.

A *measurement* is any function whose domain is composed of times and cells. In the example of positions, we can see that the domain is a temporal lineage while the codomain is the set of vectors in three dimensions. However, there are many other possible domains occuring in practice, which are exemplified in the application chapters. There are measurements taken over pairs of cells, pairs of cells in a given time step, etc...

The possibility of describing measurements in this way take us closer to physics, as it gives us mathematical objects to manipulate. Furthermore, the *domain* and *codomain* of measurements can be used for their classification, which is a fundamental component of the computational system discussed in the next chapter. Some generalities about these measurements are covered in **Section 2.2** of this chapter, which is recommended to anyone who aims to study or develop models for embryogenesis.

While the temporal point of view is more natural for visualization and manipulation, the cellular point of view has a rich algebraic structure. From this point of view, cell trajectories are the main objects of our study and this is the subject of **Section 2.3**. This structure is particularly important for readers interested on the mathematical formulation of **Chapter 5**.

In **Section 2.4** we provide some final considerations and propose directions of future research.

2.1 Times and cell lineages

In this section we formulate the concepts of times and cell lineages using the language of relations and categories, presented in **Chapter 1**. Using these concepts we give the formal definition of a temporal lineage and the proof of the isomophism of points of view. We finish by the proof that this isomorphism is preserved by restrictions of times and cells lineages. The results presented here can be seen as theoretical evidences of the adaptation of temporal lineages to the intuitions we impose over it.

It is important to note that all the definitions done here are made towards the goal of the natural isomorphism of points of view. The fact that the microscope images do not comprise the whole embryo complicates the theory significantly. This is due to the possibility of cells appearing without a mother, which allows the existence of cell lineages with multiple independent branches. While this is not a problem in the cellular point of view, the structure of the temporal point of view becomes more convoluted.

Finally, all relations used in this chapter are logical and for this reason we write $X \mapsto Y$ instead of $X \overset{\Omega}{\mapsto} Y$.

Times and time intervals

We model the concept of time as a relation between elements of a *time set*. The elements of a time set are called *time steps*.

Definition A *time* is defined by a relation $s : \mathcal{T} \mapsto \mathcal{T}$, called *succession*, with the following properties:

- it contains no cycles;
- there is only one connected component;
- every time step has at most one successor: the cardinality of $s(t)$ is either 0 or 1;
- every time step has at most one antecessor: the cardinality of $s^\dagger(t)$ is either 0 or 1.

When applicable, we call the element in $s(t)$ the *successor* of t an the element in $s^\dagger(t)$ the *antecessor* of t.

Definition A *time morphism* $f : s' \to s$ between two times is a graph morphism between times whose underlying function is injective.

Semantically, a time morphism corresponds to the renaming and inclusion of an interval of time into a possibly larger one. This inclusion must preserve successors and antecessors.

The category that is composed of times and time morphisms is called **Time**. Since every time is a graph and every time morphism is a graph morphism,

there is a forgetful functor $\mathcal{F}_{\textbf{Time}} : \textbf{Time} \to \textbf{Graph}$.

We proceed now to the description of time intervals.

Definition For all $u \leq v$, the *time interval* $[u, v]$ in $\mathcal{T}$ is composed of the elements t such that $u \leq t \leq v$. A *generalized time interval* is either a time interval or one of the three following elements:

- the past, denoted ∞_-; this is the time interval associated to the antecessors of the cells in the data set;
- the future, denoted ∞_+; this is the time interval associated to the successors of the cells in the data set;
- the undefined, denoted $\mathbb{U}$; this is the time interval associated to cells whose lineage does not appear at all in the data set.

The set of generalized intervals in $\mathcal{T}$ is called $\mathcal{I}\mathcal{T}$.

We can give a graph structure to the set of generalized time intervals.

Definition Let $s : \mathcal{T} \mapsto \mathcal{T}$ be a time, whose minimum is t_- and whose maximum is t_+. We define the relation $i_s : \mathcal{I}\mathcal{T} \mapsto \mathcal{I}\mathcal{T}$ that connects intervals whose extremities are connected. Pasts and futures are connected to themselves accordingly, and undefined is isolated, connected only to itself. That is:

- $[u_1, v_1] \to [u_2, v_2]$ if $v_1 \to u_2$;
- $\infty_- \to [u, v]$ if $u \equiv t_-$;
- $[u, v] \to \infty_+$ if $v \equiv t_+$;
- $\infty_- \to \infty_-$;
- $\infty_+ \to \infty_+$;
- $\mathbb{U} \to \mathbb{U}$.

The function that transforms every time into the graph of its intervals can be given the structure of a contravariant functor.

Proposition Let $f : \mathcal{T}' \to \mathcal{T}$ be a time morphism. We define $\mathcal{I}f : \mathcal{I}\mathcal{T} \to \mathcal{I}\mathcal{T}'$ by:

- $[u, v]$ is mapped to its preimage $f^{-1}[u, v]$. If this set is empty, either:
  - v is smaller than the image of f and the interval is mapped to ∞_-;
  - u is larger than the image of f and the interval is mapped to ∞_+;
- ∞_- is mapped to ∞_-.
- ∞_+ is mapped to ∞_+.
- $\mathbb{U}$ is mapped to $\mathbb{U}$.

Then $\mathcal{I}f$ is a graph morphism and satisfies $\mathcal{I}(f \circ g) = \mathcal{I}(g) \circ \mathcal{I}(f)$. This defines a contravariant functor $\mathcal{I} : \textbf{Time} \to \textbf{Graph}$.

We skip the proof of this proposition. ■

Lineages and tagged cell slices

We model the concept of *lineage* as a relation between elements of a *cell set*.

Definition A *cell lineage* is a relation $c : \mathcal{C} \mapsto \mathcal{C}$, called *children* with the following properties:

- it contains no cycles;
- every cell has either two or no children: the cardinality of $c(t)$ is either 0 or 2;
- every cell has at most one mother: the cardinality of $c^\dagger(t)$ is either 0 or 1.

It represents a binary forest whose nodes are *cells* and whose edges connect mothers to daughers. When applicable, we call the elements in $c(a)$ the *daughters* or *children* of a and the element in $c^\dagger(a)$ the *mother* of a.

Definition A *lineage morphism* $f : c' \to c$ is a graph morphism between lineages whose underlying function is injective.

Semantically, a lineage morphism corresponds to renaming and inclusion of a lineage into another that has been possibly extended in both directions. This inclusion must preserve daughters and mothers.

The category composed of lineages and lineage morphisms is called **Lin**. Since every lineage is a graph, there is a forgetful functor $\mathcal{F}_{\mathbf{Lin}} : \mathbf{Lin} \to \mathbf{Graph}$.

We proceed now to the description of tagged cell slices.

Definition A *cell slice* is a subset of the cell set composed of pairwise unrelated cells. A *tagged cell slice* is a pair (S, H) where S is a cell slice and H is a subset of the cell set, called the *history* of S, satisfying:

- if $a \in S$ and $b < a$, then $b \in H$;
- if $a \in S$ and $b \geq a$, then $b \notin H$;
- if $a \in H$ and $a \to b$ then $b \in S \cup H$.

The set of tagged cell slices in $\mathcal{C}$ is called $\mathcal{S}\mathcal{C}$.

We can give a graph structure to the set of tagged cell slices. There is a connection between two tagged slices if this transition is coherent with the lineage, while incrementing the history in a suitable way.

Definition Let $c : \mathcal{C} \to \mathcal{C}$ be a lineage. We define the relation $n_c : \mathcal{S}\mathcal{C} \to \mathcal{S}\mathcal{C}$ where $n_c((S_1, H_1), (S_2, H_2))$ if:

- for all $a \in S_1$, either:
  - $a \in S_2$;
  - $c(a) \subset S_2$ and $a \in H_2$.
- for all $a \in S_2$, either:
  - $a \in S_1$;
  - $c^\dagger(a) \subset S_1$ and $a \notin H_1$.
- $H_1 \subset H_2$.

The function that transforms every lineage into its graph of tagged cell slices can be given the structure of a contravariant functor.

Proposition Let $f : \mathcal{C}' \to \mathcal{C}$ be a lineage morphism. We define $\mathcal{S}f : \mathcal{S}\mathcal{C} \to \mathcal{S}\mathcal{C}'$ by (X, Y) to $(f^{-1}(X), f^{-1}(Y))$ where f^{-1} denotes the preimage. Since the inverse lineage morphism preserves the *relatedness* of cells and the presence of the past of a cell, this is a tagged cell slice. Then $\mathcal{S}f$ is a graph morphism and satisfies $\mathcal{S}(f \circ g) = \mathcal{S}(g) \circ \mathcal{S}(f)$. This defines a contravariant functor $\mathcal{S} : \mathbf{Lin} \to \mathbf{Graph}$.

We skip the proof of this proposition. ■

Temporal lineages and points of views

A *temporal lineage* is an object consisting of a cell lineage evolving in time. We can structure this objects in two different ways.

From the *temporal point of view* we think about the cells present at each time step, with the transitions happening between these time steps. Since the cells present at a given time step must be unrelated, this corresponds to a function:

$$\mathcal{T} \to \mathcal{S}\mathcal{C}$$

Since the transitions between the sets of cells should obey the transitions in the lineage, this function must be a graph morphism. That is, it must be an element of $\mathbf{Graph}(\mathcal{T}, \mathcal{S}\mathcal{C})$.

From the *cellular point of view* we think about the time steps where each cell is present, and the connection between these time intervals. Since the lifetime of a cell is contiguous, this corresponds to a function:

$$\mathcal{C} \to \mathcal{I}\mathcal{T}$$

Since the transition from mother to children happens between successive time steps, this function must be a graph morphism. That is, it must be an element of $\mathbf{Graph}(\mathcal{C}, \mathcal{I}\mathcal{T})$.

Both views are equivalent, and this is what we will formalize in the following. We start by proving that there is a *bijection* between both sets. This guarantees that for every temporal lineage viewed in one way, there is an equivalent one viewed in the other way. We proceed then to prove that this equivalence continues to hold even in the presence of transformations via morphisms.

Before starting, we give some intuition on this equivalence. The underlying set of temporal lineages can be seen as a relation of type:

$$\mathcal{T} \times \mathcal{C} \to \Omega$$

which can be transformed in two equivalent ways:

- $\mathcal{T} \to (\mathcal{C} \to \Omega)$, which corresponds to the temporal view;
- $\mathcal{C} \to (\mathcal{T} \to \Omega)$, which corresponds to the cellular view.

Since both time and cell sets occur in negative position (before the arrow), this shows the *contravariant* nature of the transformation. Both time and lineage morphisms correspond to the inclusion of lineages or times into larger ones they can also be seen as a *restriction* of the corresponding times and lineages. This leads to the intuitive conclusion that the restriction of times and lineages leads to a restriction of the temporal lineage.

We expect that this restriction is the same independently of the point of view over the temporal lineage. This can be represented similarly to a *contravariant right adjunction*, as the natural isomorphism:

$$\mathbf{Graph}(\mathcal{F}_{\mathbf{Time}}-, \mathcal{S}-) \cong \mathbf{Graph}(\mathcal{F}_{\mathbf{Lin}}-, \mathcal{I}-)$$

of the contravariant functors $\mathbf{Time} \times \mathbf{Lin} \to \mathbf{Set}$.

We proceed now to the proofs of these propositions.

Proofs of bijection and natural isomorphism

Before starting, we notice three facts.

For sets A and B, every function $f : A \to (B \to \Omega)$ has a *transpose* called $f^\dagger : B \to (A \to \Omega)$ defined by:

$$f^\dagger(b) = \{a \in A \,|\, b \in f(a)\}$$

and this operation is an involution, that is, $(f^\dagger)^\dagger = f$. This is the curried equivalent of the transposition of relations.

Every morphism $f : \mathbf{Graph}(\mathcal{T}, \mathcal{S}\,\mathcal{C})$ defines three functions of type $\mathcal{T} \to (\mathcal{C} \to \Omega)$:

- f_0, which maps each time step to its corresponding slice;
- f_-, which maps each time step to its corresponding history;
- f_+, which maps each time step to the set of cells that are yet to appear;

and these three functions completely determine the morphism, since they contain both the slice and its history.

Every morphism $g : \mathbf{Graph}(\mathcal{C}, \mathcal{I}\,\mathcal{T})$ defines three functions of type $\mathcal{C} \to (\mathcal{T} \to \Omega)$:

- g_0, which maps the cell to the set of time steps corresponding to its interval;

- g_-, which maps the cell to the set of time steps corresponding to its past lineage;
- g_+, which maps the cell to the set of time steps corresponding to its future lineage;

and these three functions completely determine the morphism, since the emptiness of the image of g_- and g_+ determine the classification of an empty interval as ∞_-, ∞_+ or $\mathbb{U}$.

Proposition There is a bijection between **Graph**$(\mathcal{T}, \mathcal{SC})$ and **Graph**$(\mathcal{C}, \mathcal{IT})$.

We prove this proposition in three steps.

Lemma Every f : **Graph**$(\mathcal{T}, \mathcal{SC})$ defines a morphism g : **Graph**$(\mathcal{C}, \mathcal{IT})$.

We define $g_0 = f_0^\dagger$, $g_- = f_+^\dagger$ and $g_+ = f_-^\dagger$.

We prove that for any $x : \mathcal{C}$, if $g_0(x)$ is non-empty, then it is an interval. Let t^- and t^+ be the lower and upper bounds of the set $g_0(x)$. Since $x \in f_0(t^+)$ then x is not present in this set's history, and consequently, in that of any previous time step. Also, since $x \in f_0(t^-)$ and the disappearance of x implies its appearance of the history, it means that x is present in all time steps between t^- and t^+. Therefore, $g_0(x)$ is an interval.

Then we define the function $g : \mathcal{C} \to \mathcal{IT}$ by:

- If $g_0(x)$ is non-empty, then $g(x) = g_0(x)$;
- Otherwise:
  - If $g_-(x)$ is non-empty, then $g(x) = \infty_+$;
  - If $g_+(x)$ is non-empty, then $g(x) = \infty_-$;
  - Otherwise $g(x) = \mathbb{U}$.

In order to prove that g is a morphism, we must prove that if $x \to x'$ then $g(x) \to g(x')$.

If $g(x)$ and $g(x')$ are both empty, it is clear that they have the same category and therefore are connected by the identity arrow.

If $g(x)$ is empty and $g(x')$ not, then $g(x)$ must be ∞_- and the smallest element of $g(x')$ must be the smallest element of $\mathcal{T}$, otherwise f would not be a morphism. Therefore they are connected by the arrow coming from ∞_-.

If $g(x)$ is non-empty and $g(x')$ is empty, we use the same reasoning as before, with $g(x')$ being ∞_+. Therefore they are connected by the arrow targeting ∞_+.

If both are non-empty, let t_+ be the last time step of $g(x)$ and t_- the last first step of $g(x')$. Since f is a morphism and $x \in f(t_+)$, then $c(x) \subset f(s(t_+))$ and consequently $s(t_+) \in g(x')$. Since $g(x')$ is an interval, $s(t_+)$ must be the smallest element of the interval and $g(x) \to g(x')$. ■

Lemma Every g : **Graph**$(\mathcal{C}, \mathcal{IT})$ defines a f : **Graph**$(\mathcal{T}, \mathcal{SC})$.

We define $f_0 = g_0^\dagger$ and $f_- = g_{++}^\dagger$ where

$$g_{++}(x) = \{u : \mathcal{T} \mid y \geq g_0(x)\}$$

that is, the set of time steps following the disappearance of the cell.

We prove that for any $t : \mathcal{T}$ the set $f_0(t)$ is a slice. Let x and x' be two cells in $f(t)$. Then $g(x)$ and $g(x')$ have an nonempty intersection, since both contain t. Since related cells always have nonintersecting life intervals, x and x' cannot be related. Therefore $f(t)$ is a slice.

We prove that $f_-(t)$ is a valid history for $f_0(t)$. These sets do not intersect since for all $x : \mathcal{C}$ the sets $g_0(x)$ and $g_+(x)$ do not intersect. The assertion that $x \in f_-(t)$ and $y \leq x$ implies $y \in f_-(t)$ comes from the fact that if $t \in g_+(x)$ and $s \geq t$ then $s \in g_+(x)$. The assertion that $x \in f_-(t)$ and $x \to y$ implies $y \in f_-(t) \cup f_0(t)$ comes from the fact that $t \in g_+(x)$ and $s \to t$ then $s \in g_+(x) \cup g_0(x)$.

We define $f(t) = (f_0(t), f_-(t))$. In order to prove that f is a morphism, we must prove that if $t \to t'$ then $f(t) \to f(t')$.

The fact that $f_-(t) \subset f_-(t')$ derives from the observation that for any $x : \mathcal{C}$ and $y \leq x$ it is true that $g_+(x) \subset g_+(t)$.

If both $f(t)$ and $f(t')$ are empty, then all the conditions for the morphism are trivial.

If $f(t)$ is non-empty, let x be such that $t \in g_0(x)$. If t is the maximum $g_0(x)$, since g is a morphism, we have that $c(x) \subset g_0(t')$. If t is not the maximum of this interval, then t' is in this interval and $x \in f(t)$.

If $f(t')$ is non-empty, let x be such that $t' \in g_0(x')$. If t' is the minimum of $g(x)$, since g is a morphism, we have that $c^\dagger(x) \subset f(t)$. If t' is not the minimum of this interval, then t is in this interval and $x \in f(t')$.

Therefore the transitions between slices are valid and f is a morphism. ■

Lemma The transformations defined above are inverses.

This follows from the fact that † is an involution. ■

Natural isomorphism of points of view

Proposition The contravariant functors $\mathbf{Graph}(\mathcal{F}_{\mathbf{Time}}-, \mathcal{S}-)$ and $\mathbf{Graph}(\mathcal{F}_{\mathbf{Lin}}-, \mathcal{I}-)$ are naturally isomorphic.

We hide the forgetful functors for simplicity. Let $f : \mathcal{T} \to \mathcal{T}'$ and $g : \mathcal{C} \to \mathcal{C}'$. We must prove the commutation of the following diagram:

$$\begin{tikzcd}
\mathbf{Graph}(\mathcal{T}', \mathcal{SC}') \arrow[r, "\mathbf{Graph}(f,\mathcal{S}g)"] \arrow[d, "I_{\mathcal{T}'\mathcal{C}'}"'] & \mathbf{Graph}(\mathcal{T}, \mathcal{SC}) \arrow[d, "I_{\mathcal{T}\mathcal{C}}"] \\
\mathbf{Graph}(\mathcal{C}', \mathcal{IT}') \arrow[r, "\mathbf{Graph}(g,\mathcal{I}f)"'] & \mathbf{Graph}(\mathcal{C}, \mathcal{IT})
\end{tikzcd}$$

The proof of bijection explores the expression of the transformations as involutions using †. Using the same expression, the natural isomorphism is equivalent to:

$$\mathbf{Graph}(g, \mathcal{I}f) \circ (\dagger) = (\dagger) \circ \mathbf{Graph}(f, \mathcal{S}g)$$

Since this equation can be translated into the language of relations as:

$$(g, f)_* \circ (\dagger) = (\dagger) \circ (f, g)_*$$

which is true, the diagram commutes and the functors are naturally isomorphic. ■

2.2 Measurements

In this section we discuss the concept of a *measurement*. Intuitively, a measurement is the attribution of a value to some object. For example, one can measure velocity by the attribution of a vectorial value to all cells at all time steps. We use this term for both measurements coming from experiences, like positions, and *statistics*, which are functions applied over the experimental samples.

We define a measurement to be any function that has a defined domain and codomain, with the domain being composed of cells and times. This description is vague since there is no obvious way of determining all the possible ways of composing these concepts in a semantically compatible way. Therefore we proceed by showing examples of measurements, domains and codomains that are relevant to applications. As remarked before, the possibility of denoting measurements as mathematical objects is the one of the things that take us closer to physics.

This section depends on **Section 1.3**.

Points of view and domains

We can represent a temporal lineage using the relational representation

$$f : \mathcal{T} \times \mathcal{C} \to \Omega$$

that defines a subset of $\mathcal{T} \times \mathcal{C}$, whose elements are the (t, c) such that the cell c is present in the data set at the time t. This subset is denoted

$$\sum_{t:\mathcal{T}} \mathcal{C}_t$$

where $\mathcal{C}_t = f(t)$. We can also use the symmetric representation

$$f^\dagger : \mathcal{C} \times \mathcal{T} \to \Omega$$

that defines a subset of $\mathcal{T} \times \mathcal{C}$, whose elements are the (c, t) such that t is a time step where c is alive. This subset is denoted

$$\sum_{c:\mathcal{C}} \mathcal{T}_c$$

where $\mathcal{T}_c = f^\dagger(c)$. Since the transposition of relations is an isomorphism, we have that:

$$\sum_{t:\mathcal{T}} \mathcal{C}_t \cong \sum_{c:\mathcal{C}} \mathcal{T}_c$$

Temporal lineages are the fundamental domains from which other domains for measurements are derived.

Relative and absolute times and slices

When we talk about $\mathcal{T}$ and $\mathcal{C}$ we are implicitly attributing an identifier to each time and cell, which is their corresponding element in these sets. However in the implementation of algorithms and in some applications, it is useful to have identifiers that are local to a time step or cell and defined by integers. In that way, the elements become indices in an array.

For any time $\mathcal{T}$ we can define another time called its *relative time* $\mathcal{T}'$, which is composed by integers and has the following properties:

- $\mathcal{T}$ and $\mathcal{T}'$ are isomorphic;
- the smallest element of $\mathcal{T}'$ is 0.

That is, the relative time of the time interval $\mathcal{T}_c$ of a cell c is a measurement of the time elapsed since the appearance of this cell in the data set. Also, the measurement of time is done in time steps.

Similarly, for any slice $\mathcal{C}_t$ we can define another set, its *relative slice* $\mathcal{C}'_t$ representing the local identifiers for cells at the time step t. Since slices have no defined order, we can define it directly as $\mathcal{C}'_t = [0, |\mathcal{C}_t| - 1]$.

Using these representations, the isomorphisms of point of view becomes:

$$\sum_{t:\mathcal{T}} \mathcal{C}'_t \cong \sum_{c:\mathcal{C}} \mathcal{T}'_c$$

which is the isomorphism used in the computational implementation.

Typing measurements

In this section we give examples of the types to which certain measurements belong, which corresponds to an attribution of a domain and a codomain. The first example is the positions of cells in space. This can be represented by a function whose domain is a temporal lineage and whose codomain is a vector is $\mathbb{R}^3$, having the type:

$$\sum_{c:\mathcal{C}} \mathcal{T}_c \to \mathbb{R}^3$$

which is isomorphic to:

$$\sum_{t:\mathcal{T}} \mathcal{C}_t \to \mathbb{R}^3$$

and in applications:

$$\sum_{c:\mathcal{C}} \mathcal{T}'_c \to \mathbb{R}^3$$

Domains can be structurally simple. For example the measure of the average speed of each cell, or other global cell statistics, corresponds to:

$$\mathcal{C} \to \mathbb{R}$$

The measure of the center of mass of the embryo at each time step, or other global temporal statistics, corresponds to:

$$\mathcal{T} \to \mathbb{R}^3$$

In **Chapter 5** we calculate dissimilarities between cells, which corresponds to:

$$\mathcal{C} \times \mathcal{C} \to \overline{\mathbb{R}}$$

The distance between cells at each time step corresponds to:

$$\sum_{t:\mathcal{T}} \mathcal{C}_t \times \mathcal{C}_t \to \mathbb{R}$$

Finally, if we align trajectories with relation to their relative time, their position correspond to:

$$\sum_{t':\mathcal{T}'} \mathcal{C}_{t'} \to \mathbb{R}$$

Manipulation of measurements

We give an example of manipulation of measurements using the basic operations on domains and codomains. This is useful both for the mathematical formulation and for programming, since the steps below can be seen as type signatures for functions.

We assume the existence of some algorithm for the calculation of speed, which for any time set $\mathcal{T}$, has the type $(\mathcal{T} \to \mathbb{R}^3) \to (\mathcal{T} \to \mathbb{R})$. In order to obtain velocities for all cells at all times in temporal view, from the position of all cells at all times in temporal view we execute the following steps:

1. to start with position:
$$\sum_{t:\mathcal{T}} \mathcal{C}_t \to \mathbb{R}^3$$
2. to apply the isomorphism between temporal view and cellular view:
$$\sum_{c:\mathcal{C}} \mathcal{T}_c \to \mathbb{R}^3$$
3. to apply dependent currying:
$$\prod_{c:\mathcal{C}} (\mathcal{T}_c \to \mathbb{R}^3)$$
4. to use the lifted application of speed:
$$\prod_{c:\mathcal{C}} (\mathcal{T}_c \to \mathbb{R})$$
5. to apply dependent uncurrying:
$$\sum_{c:\mathcal{C}} \mathcal{T}_c \to \mathbb{R}$$
6. to apply the isomorphism between cellular view and temporal view:
$$\sum_{t:\mathcal{T}} \mathcal{C}_t \to \mathbb{R}$$

2.3 Trajectories

In this section we describe an algebraic structure for *cell trajectories*. It is composed of the operations of *concatenation* and *intersection* and is essential for the devolopment of **Chapter 5**.

Defining cell trajectories

Let $\mathcal{T}$ be a time set and S a state set for measurements. We define a *trajectory* to be a dependent pair (I, x), where $I = [I_-, I_+]$ is an interval in $\mathcal{T}$ and $x : I \to S$ is a function into the state space. We call the space of trajectories in time $\mathcal{T}$ and state S by $Tr(\mathcal{T}, S)$.

Possibly undefined values

We define a *possibly undefined value of type* X to be an element of the set $\overline{X} = X \cup \{\mathbb{U}\}$ where the *undefined* value $\mathbb{U}$ has been added. Any function $f : X \to Y$ can be lifted to a function $\overline{f} : \overline{X} \to \overline{Y}$ by

$$\begin{aligned}\overline{f}(\mathbb{U}) &= \mathbb{U}\\ \overline{f}(x) &= f(x)\end{aligned}$$

It can be verified that this operation defines a functor **Set** $\to$ **Set**. We identify semantically the set $\overline{\mathbb{R}}$ with the extended reals, where $\mathbb{U} = \infty$.

An algebra for trajectories

Here we define the operations of intersection and concatenation of trajectories. Since the operation of intersection in time does not always have a defined result, we will have to formalize this notion and deal with these cases.

Let $X = (I, x)$ and $Y = (J, y)$ be trajectories. If $I_+ \to J_-$, we can define the *trajectory concatenation* $\vee$ by:

$$X \vee Y = ([I_-, J_+], x \vee y)$$

where

$$[x \vee y](t) = \begin{cases} x(t) & \text{if } t \in I \\ y(t) & \text{if } t \in J \end{cases}$$

This operation can be lifted to $\overline{Tr(T,S)}$, by making $\mathbb{U}$ a unity:

$$X\overline{\vee}Y = \begin{cases} Y & \text{if } X = \mathbb{U} \\ X & \text{if } Y = \mathbb{U} \\ X \vee Y & \text{otherwise} \end{cases}$$

We define moreover *trajectory intersection* $\wedge : Tr(T,S) \times Tr(T,S) \to \overline{Tr(T,S^2)}$ by:

$$X \wedge Y = \begin{cases} (I \cap J, (x,y)|_{I\cap J}) & \text{if } I \cap J \neq \emptyset \\ \mathbb{U} & \text{otherwise} \end{cases}$$

This operation can be lifted to $\overline{Tr(T,S)}$ by defining $\mathbb{U}$ to be a zero

$$X\overline{\wedge}Y = \begin{cases} \mathbb{U} & \text{if } X = \mathbb{U} \\ \mathbb{U} & \text{if } Y = \mathbb{U} \\ X \wedge Y & \text{otherwise} \end{cases}$$

Using these definitions, it is not hard to prove the distributivity of intersection over concatenation

$$Z\overline{\wedge}(X\overline{\vee}Y) = (Z\overline{\wedge}X)\overline{\vee}(Z\overline{\wedge}Y)$$

In the following chapters we hide the overline and write $\wedge$ and $\vee$ for the lifted operations.

2.4 Discussion

This chapter offers a mathematical representation for the objects studied in this thesis. In particular, we can represent the data, composed by spatio-temporal lineages as a function:

$$f : \sum_{t:\mathcal{T}} \mathcal{C}_t \to \mathbb{R}^3$$

from which one can derive a variety of other measurement spaces.

The possibility of this representation makes our approach closer to physics. Indeed, in the cellular point of view these positions are represented by:

$$f : \sum_{c:\mathcal{C}} \mathcal{T}_c \to \mathbb{R}^3$$

which can be seen as the equivalent of the fluid flow in the Lagrange approach:

$$f : \mathbb{R}^3 \times \mathcal{T} \to \mathbb{R}^3$$

where $f(x_0, t)$ corresponds to the position of the fluid element x_0 at time t. This does not make the equivalence without difficulties, due to limited life span of cells, which does not happen with fluid elements.

The fact that temporal lineages have a more complicated structure than the spaces in fluid mechanics makes their manipulation more convoluted. While one can manipulate these spaces intuitively, it is interesting to have formal guarantees of their validity. The theorems proven in **Section 2.1** ensures the coherence of points of view with relation to some simple transformations, like restriction of time intervals and lineages. However, as it will be seen in the application chapters, these manipulations are not enough for the practice on the domain. In the following we propose some next steps in order to improve this situation.

Research directions

As commented previously, the structure given to time intervals and lineages is significantly complicated by the fact that cells enter and exit the field of view. However, we did not explicitly developed the theory of *in-toto* cell lineages. An interesting development would to expose this simplified theory, as compared to the case of partial lineages, and show how both categorical structures interact. That is, what is the mathematical construct that represents the more convoluted structure as due to a loss of information.

The categorical interpretation given in this chapter is limited to restrictions of time intervals to other time intervals. Another temporal transformation that is practically significant is the change of *sampling rates*. Physically, it is relevant due to its relation with the temporal scales of the visualized measurements. The choice of a sampling rate that is small enough can imply the change of the shape of lineages, making the connection of a mother only to two children not true anymore. Therefore one would have to generalize the structure given here.

We can also explore this subject in an aesthetical way and remark the similarity of the natural isomophism of points of view presented here with an adjunction. Therefore, one can proceed to a generalization of both the space of times and cells so that the natural isomorphism is an actual adjunction, while preserving the semantic content on the structures we consider to be actual biological cell lineages.

The mathematical framework presented here is not adequate for the manipulation of multiple embryos. In order to be better adapted it would have to be able to deal at least with different sampling velocities. Furthermore, the direct use of temporal lineages as the means of comparison between different embryos is inappropriated, since it excludes all the material factors that make embryogenesis be the phenomenon it is. Instead, it is more appropriate to compare embryos through common sets of measurements. A naive way of doing it using a similar framework to that used here would be by the definition of a 2-category whose arrows are measurements and whose 2-arrows represent a difference between measurements. In this context, the problem of best matching between the cells of different embryos can be formulated as a *Kan extension* (Lehner, 2014).

Finally, we called a measurement any function whose domain is a combination of cells and times. This definition is very vague and we question if this can be formalized. This could be done by the establishment of a collection of primitive manipulations, which starting from a temporal lineage, can generate all the possible domains that are semantically valid. This thesis offers a large range of domains that serve as a practical constraint for this exploration, and a series of manipulations that produces valid domains, like isomorphism of points of view, projections, liftings, products, etc, which serves as a good starting point. The obtention of a description of this kind would allow the mathematical formalization of manipulations to be derived from the formalization of the set of building blocks.

Chapter 3

Implementation and design

This chapter discusses the engineering aspects that involve the development of `lineageflow`, a computational system for the manipulation of measurements, with the goal of creating a collaboration ground for biologists, physicists, mathematicians and computer scientists. The implementation of such a system has to cover many practical aspects like:

- development workflow;
- exploration of measurements;
- reuse of components;
- deployment of algorithms;
- usage by a non-programmer.

We give an overview of the approach used to deal with each of these problems in the following. The organization of the sections of this chapter is based on the decomposition of computation in *pure* and *effectful* components.

A computation is pure when it is fully described by its inputs and outputs. This is the case of algorithms, which transform sets of measurements into other sets of measurements. This aspect of the system is discussed in **Section 3.1**.

In opposition, a computation is effectful when it has an observable interaction with its environment. This is the case of the interaction with a database of measurements, plotting and others. This aspect of the system is discussed in **Section 3.2**.

We give some special emphasis to the tools for visual and statistical exploration of data. These are essential for the interaction between researchers in an interdisciplinary domain. These tools are discussed in **Section 3.3**.

All the sections are recommended for readers who aim to develop algorithms within the system.

Algorithm development and pure computation

A type system is a language used by the programmer to describe data and functions. The process of type checking allows a real-time interaction of the developer with the compiler, which makes it possible to catch errors before the program is run. Most importantly, a type system gives a declarative interface for functions, using its inputs and outputs.

One of the factors that distinguish the different type systems used in programming languages is their *expressivity*. Languages range from untyped flavors of ASM, going through C, Java, up to dependently typed languages like Coq and Agda, whose type system is as expressive as the value level programming language itself. The implementation of `lineageflow` is made in Haskell, a functional language that offers a good ground between these two extremes. It is not object-oriented, which makes it different from Java and Python, for example. It is based on the Hindley-Milner (Milner, 1978) type system, being able to express algebraic data types and higher order polymorphism, and having the unique property of *type inference*, meaning that the compiler can infer the type of any correctly formed expression. Furthermore, the language has a good ecosystem and a foreign function interface (FFI) to interact with other languages.

Using the type system we can encode aspects of the domain of study in a manner that prevents errors. As an example, it is very reasonable to encode the identifiers of both time and cells using integers. However, a function:

```
cellSpeed :: Int -> Double
```

accepts any integer as input, which can lead to involuntary errors, like the input of a time identifier instead of a cell identifier. But a function:

```
cellSpeed :: Cell -> Double
```

will only accept values that have been defined to be cell identifiers, and the compiler will prevent you to do it otherwise. In **Section 3.1** we expose how the primitive concepts and manipulations of the study of temporal lineages and measurements have been encoded computationally, and the guarantees that are given by this encoding.

Another practical aspect that must be considered in the development of a system like this is code reuse and modularity. A recognized enemy of modularity is global state (Raymond, 2003), since its use makes the use of functions context dependent. Ideally, functions shoud be fully described by their inputs and outputs. While this can be done in any language, the use of a functional language enforces this behavior.

Database and effectful computation

The precise definition and encoding of our objects of study allows a great deal of automation on recurrent tasks. In this case, it allowed the automation of the database interaction and the generation of the user interface. This automation comes from the declaration of algorithms in the format:

```
algorithm :: Parameters -> Input -> Output
```

and of programs, which are collections of related algorithms, in the format:

```
data Program =
  Algorithm1 ProgramInput |
  Algorithm2 ProgramInput |
  Algorithm3 ProgramInput
```

where `|` represents an union in the type system. Besides of the positive aspect of explicit declaration, it allows the programmer to take less decisions when developing, and thus focus on the algorithm.

This interface is only possible due to the developments of **Chapter 2**, which allow the classification of measurements by their domain and codomain. This classification makes possible to give only appropriated inputs to algorithms, and to use the produced outputs in the correct way.

The deployment of executables is also declarative, using Nix (Dolstra, Löh, & Pierron, 2010). As soon as the necessary infrastucture has been provided for the initial deployment, the cost of adding new executables is close to zero. Fast deployment lowers the barrier for interaction between developers and biologists.

Finally, it is very common for the developer to explore measurements and statistics in a dynamical way, adjusting calculations in real time, visualizing plots, etc. While Haskell, like most compiled languages, is not perfect in this sense, the knowledge of the domain allowed the development of a library that gives an infrastructure offering an experience à la Matlab/Python using the REPL. As a positive byproduct, environments of exploration are declarative, allowing their sharing and reproduction.

The effectful infrastructure is explained in depth in **Section 3.2**.

User interface and exploration

Another product of the declarative nature of algorithms is an unified graphical user interface. A graphical user interface facilitates the interaction with algorithms, eliminating the need of programming knowledge, which gives *autonomy* to specialists who only wish to apply methods and interpret the results. The fact that it is unified allows a one-time learning experience, with the marginal cost of learning how to use a new algorithm much lower.

The same is done for the exploration of measurements. The infrastructure for 3D+time visualization of measurements and statistical plotting are connected to the graphical user interface, making their use easier. This is exposed further in **Section 3.3**.

3.1 Computational core

We present here the encoding of temporal lineages and measurements in Haskell, together with its denotations. This section has a weak prerequisite in **Section 1.3** and **Section 2.2**.

Notation

We present here some Haskell notations (Lipovaca, 2011), which are necessary for the exposition of the computational terms. Lines preceded by a > represent input commands in the REPL, and the following lines the output.

The type of a value is declared with `(::)`. Function types are declared with `->`.

```
> :type sin
sin :: Double -> Double
```

Function application is done with separation by whitespace:

```
> sin 0
0
```

Functions with multiple inputs are usually declared in curried form:

```
> :type plus
Int -> Int -> Int
> plus 1 2
3
```

Records or product types can be named or unnamed:

```
-- a record with two unnamed integers
-- the first 'Pair' is the name of the type
-- while the second 'Pair' is the name of the constructor
data Pair = Pair Int Int

-- a record with two named integers
data Pair = Pair
  { fst :: Int
  , snd :: Int }
```

Records with only one element can be newtyped, which incurs no performance penalties with relation to the base type:

```
-- NewInt is distinct from Int, but has the same runtime
-- representation
newtype NewInt = NewInt Int
```

Sum types are represented as:

```
data Name =
  Option1 Var1 |
  Option2 Var2 Var3
```

In the same way that values have types, types have *kinds*. Basic types like `Int` and `Double` have kind `Type`. Type constructors like `Maybe`:

```
data Maybe a =
  Nothing |
  Just a
```

which represents possibly undefined values, have kind `Type -> Type`.

Domain specific types

We present here the types and constructs that model temporal lineages, measurements and their derived structures. It is important to remark that the encoding of this domain of study is limited to the scope of Haskell's type system, which is based on Hindley-Milner (Milner, 1978) and for this reason, comes with similar limitations. The most fundamental one is that we cannot encode arbitrary propositions in the type system. Under the equivalence of categories and types (Taylor, 1999), this means that we cannot talk about subobjects in this language. As an example, we can express "the set of cells", but not "the set of cells that go through mitosis". That being said, we start presenting the fundamental types.

The fundamental types are:

- `newtype Time = Time Int`, which represents an absolute identifier of time;
- `newtype Cell = Cell Int`, which represents an absolute identifier of a cell;
- `newtype Dep a b = Dep b`, which represents `b` in a manner that is dependent of `a`. For example, the local identifier of a cell, at a given time step, is represented by `Dep Time Cell`;
- `S2 a`, `S3 a`, `S4 a`, which represent simplexes (non-repeating lists) with vertices of type `a`. For example, cell contacts at a given time step are represented by `S2 (Dep Time Cell)`.

Synonyms for `Scalar`, `Vector` and `Tensor` are also defined:

- `type Scalar = Double`;
- `type Vector = V3 Scalar`, where `data V3 a = V3 a a a`;
- `type Tensor = V3 Vector`.

Using these fundamental types, we can define measurements. Measurements are always presented in the form `f a` where `f` is a type constructor of kind `Type -> Type` and `a` is a type of kind `Type`. The type `a` represents the set

where the values of the measurement are taken, or the *codomain* of the measurement. The type `f` represents the object over which we are performing a measurement, or the *domain* of the measurement.

As it can be seen, domains are not represented by their corresponding types. Instead, they are represented in Yoneda form, where the type `d` is represented by `d ->`, the type of functions whose domain is `d`.

For efficiency reasons, this representation is not done using the function constructor `(->)`, but using indexed arrays `Array :: Type -> (Type -> Type)` instead. This representation of measurements as arrays is possible since measurements are functions over finite domains, whose values can be enumerated. The values corresponding to the domain subset can be recovered using an identity map.

Therefore, for a given domain `d` and codomain `a`, the type `Array d a`, represents the set of measures with values in `a` over the domain `d`. As an example, the measurement attributing to every cell its first time step in the data set would have the type `Array Cell Time`.

Using this method, we can construct more complicated domains, namely domains that have more than one component. For example, position is a measurement that gives for every time step, for every cell, a vector. So we may be tempted to say it has the type

```
Array Time (Array (Dep Time Cell) Vector)
```

While this is semantically correct, this does not follow the idea that the domain-image decomposition is a measurement over the embryo. In this case, the domain is time and the image is a measurement over cells in that time step. We solve this situation by the use of the combinator

```
Compose :: (Type -> Type) -> (Type -> Type) -> Type -> Type
```

which composes two type constructors into a single one. Here, this corresponds to the construction of a domain from two others, giving the type:

```
Compose (Array Time) (Array (Dep Time Cell)) Vector
```

Since this pattern happens frequently, we define the synonym:

```
type DSumMap t c a = Compose (Array t) (Array (Dep t c)) a
```

which simplifies the type to `DSumMap Time Cell Vector`. The name means "map from dependent sum". Using this synonym we can also talk about the isomorphisms between the time-view and the cell-view:

```
tc :: DSumMap Time Cell (Cell, Dep Cell Time)
ct :: DSumMap Cell Time (Time, Dep Time Cell)
```

which can be used to transform measurements between the two points of view.

Manipulation example

We give here an example of a typical manipulation of measures. Using a function `speed :: Array (Dep Time Cell) Vector -> Array (Dep Time Cell) Scalar`, which calculates the velocity for a single cell, we will calculate the velocity for every cell, in time-view, using positions in time-view.

The isomorphisms between the time and cell views correspond to elements `tc` and `ct` as before. Isomorphism application is:

```
dSumMapTranspose ::
  forall i j a.
  DSumMap i j (j, Dep j i) -> DSumMap j i a -> DSumMap i j a
```

that means that global speed can be calculated as:

```
globalSpeed ::
  DSumMap Time Cell Vector -> DSumMap Time Cell Scalar
globalSpeed =
  dSumMapTranspose tc . -- conversion to time-view
  Compose . -- uncurrying
  fmap speed . -- lifted application
  getCompose . -- currying
  dSumMapTranpose ct -- conversion to cell-view
```

where `.` denotes the composition of functions, and which must be read from bottom to top, like mathematical notation.

Denotations

Denotational semantics (Stoy, 1977) is a method of specifying the meaning of languages using a mathematical model as a reference. For every set of syntactic constructs C we define:

- a mathematical model $[\![C]\!]$ of meanings
- a semantic function $[\![\cdot]\!] : C \to [\![C]\!]$

The idea has been created by Scott and Strachey (Scott & Strachey, 1971) to model computer programs in aspects like termination and non-determinism. We use this method as a *guide* for the implementation of the system, that is, the computational implementation is directed by the desired semantics. In the general case, there is no exact rule on how to execute this procedure, but a good way of verifying that the denotation is coherent is through the *equivalence of properties* (Elliott, 2009). This principle says that the "manipulation of the meaning must follow the meaning of the manipulation", which is the criterium used for the implementation. We list some of these denotations in the following, which are implemented in the package `lineageflow-core`.

Type level denotations

We start with type level denotations of the fundamental types. These denotations take *types* and return *sets*. We use the notation exposed in **Chapter 2** for the exposition.

The basic building blocks are the set of time steps:

$$[\![\texttt{Time}]\!] = \mathcal{T}$$

and the set of cells:

$$[\![\texttt{Cell}]\!] = \mathcal{C}$$

The function `Dep` constructs relative times and cells:

$$[\![\texttt{Dep Cell Time}]\!] = \mathcal{T}'_c$$

$$[\![\texttt{Dep Time Cell}]\!] = \mathcal{C}'_t$$

and as such, pairs in which second element depends of the first one are mapped to dependent sums:

$$[\![\texttt{(a, Dep a b)}]\!] = \sum_{a:[\![\texttt{a}]\!]} [\![\texttt{b}]\!]_a$$

Measurements are defined by their domain and codomain. Indexed arrays represent functions:

$$[\![\texttt{Array a b}]\!] = [\![a]\!] \to [\![b]\!]$$

so that, for example, a function that gives the first time step of each cell has the type and meaning:

$$[\![\texttt{Array Cell Time}]\!] = \mathcal{C} \to \mathcal{T}$$

Composed arrays denote maps over dependent sums:

$$[\![\texttt{DSumMap i j a}]\!] = \sum_{i:[\![\texttt{i}]\!]} [\![\texttt{j}]\!]'_i \to [\![\texttt{a}]\!]$$

and as an example, the isomorphism between time-view and cell-view is given by an element with type and meaning:

$$[\![\texttt{DSumMap Time Cell (Cell, Dep Cell Time)}]\!] = \sum_{t:\mathcal{T}} \mathcal{C}_t \to \sum_{c:\mathcal{C}} \mathcal{T}_c$$

Finally, the `Compose` constructor corresponds to uncurrying:

$$\begin{aligned}[\![\texttt{Compose}]\!] &: [\![\texttt{Array i (Array (Dep i j) a)}]\!] \to [\![\texttt{DSumMap i j a}]\!] \\ &: \prod_{i:[\![\texttt{i}]\!]} ([\![\texttt{j}]\!]'_i \to [\![\texttt{a}]\!]) \to (\sum_{i:[\![\texttt{i}]\!]} [\![\texttt{j}]\!]'_i \to [\![\texttt{a}]\!])\end{aligned}$$

Value level denotations

Now we proceed with the value level denotations. Following (Elliott, 2009), these are done using Haskell's type classes. These type classes have laws that have to be followed, which are the properties that guide the implementation.

Since arrays are denoted by functions, the equivalent of `Functor`'s `fmap`:

```
fmap :: Functor f => (a -> b) -> f a -> f b
```

which applies a function to every element in the array, is composition:

$$[\![\texttt{fmap f m}]\!] = f \circ [\![\texttt{m}]\!]$$

The same cannot be done with the domain since we have no access to its exact elements only from the type, as discussed before.

The function denotation implies the behavior of `Apply`'s `zipA`:

```
zipA :: Apply f => f a -> f b -> f (a,b)
```

as a product of measurements:

$$[\![\texttt{zipA m1 m2}]\!] = [\![\texttt{m1}]\!] \times [\![\texttt{m2}]\!]$$

Since arrays correspond to functions, indexing of arrays

```
(!) :: Array i a -> i -> a
```

corresponds to function evaluation:

$$[\![\texttt{m!i}]\!] = [\![\texttt{m}]\!]([\![\texttt{i}]\!])$$

but this must be done with attention, once more because of the subobject indetermination.

Since arrays are functions, and functions can be composed through evaluation, arrays can be composed with

```
comp :: Array a b -> Array b c -> Array a c
comp f g = fmap (\b -> g ! b) f
```

which gives

$$[\![\texttt{comp}]\!] = \circ$$

3.2 Effectful core

Here we expose the effectful core, which automatizes some aspects of the development of algorithms, namely the database access and the user interface. This automation achieved through a declarative interface for executables and algorithms. By declaring algorithms and executables following a certain structure, the details of implementation are taken care of.

Measurement definition

Inside the type system of the programming language, measurements are structurally defined by their domain and codomain. While this is enough for the development of algorithms, this is not enough to guarantee a database that has good properties. For this goal, we define the criteria that define the classification of measurements, with examples being given by the measurement that attributes to each cell its mother, if it exists. They are:

- *species*, the scientific name of the animal. Ex. *Danio rerio*, the zebrafish;
- *specimen*, an identifier of the experiment that generated the data. Ex. *141108aF*, using the BioEmergences identifiers;
- *tracking*, an identifier of the tracking algorithm used to generate the temporal lineage. Ex. *SimAnn*;
- *domain*, the type of the domain over which the measurement is taken. Ex. *cell*;
- *domain name*, the identifier of the subset of the domain which is used. Ex. *all*;
- *image* or *codomain*, the set in which the measurements are taken. Ex. *maybe cell*;
- *name*, the identifier of this measurement. Ex. *mother*.

This classification leads to a database scheme which defines the elements the user has to provide as input to the algorithm. A property of this scheme is that if two measurements share all elements, except the identifier, then they correspond to measurements over the same "real" object, with values in the same set. In particular, the arrays containing the measurement values have the same lengths for both.

While the criteria of classification are defined, the way each entry is to be written must be based on some convention. Domains and codomains are simple, since they can be serialized directly from the programming language. However, while there are conventions for the scientific names of species, this is not true for the identifications of trackings, domain subsets or measurement names, or even the scheme of identification of a specimen. Therefore, the exact contents of the database scheme is something that must be decided by a community.

The access to the database is given by a *measurement query*, which consists of a

JSON serialization of the classification of a measure. This uniformization of the access to measurements is one of the ingredients for the automation of the user interface.

Algorithm definition

Algorithms must be declared matching the pattern `Parameter -> Input -> Output`. In this form, both inputs and outputs must be composed of measurements, while parameters must be values that can be interpreted and fed to the algorithm.

Each component of the parameters, inputs and ouputs have a *cardinality*. The cardinality can be either `Single`, corresponding to a single value, or `Many`, corresponding to a list of values.

The conditions for each component of the algorithm are:

- Parameters must be records of readable quantities, with a given cardinality.
- Inputs are records of measurements with a given cardinality, whose domain and image are *readable*.
- Output are records of measurements with a given cardinality, whose domain and image are *writeable*.

Both *readable* and *writeable* are properties that are defined internally, with relation to the database and this is checked at compile time.

An example of algorithm declaration is given here:

```
module Derivative

data Parameter = Parameter
  { parameter_size
      :: Int -- type
      :% Single -- cardinality
      :? "Size of the filter to be used for derivation."
  } deriving Generic

data Input = Input
  { input_measurement
      :: DSumMap Cell Time Vector
      :% Single
      :? "Vectorial measurement to be derived."
  } deriving Generic

data Output = Output
  { output_measurement
      :: DSumMap Cell Time Vector
```

```
      :% Single
      :? "Derived measurement."
  } deriving Generic

algorithm :: Parameter -> Input -> Output
algorithm
  (Parameter (S s))
  (Input (S m)) =
  (Output (S d))
  where
    d = deriveV (simpleForward s) d
```

Given this declaration, the system derives automatically the way to access the database for each one of these components. This is done using GHC Generics (Lämmel & Peyton Jones, 2003), which decompose the records into components and processes each one of them separately.

The interface between the user and algorithms is mediated by *algorithm queries*. These queries are JSON serializations of the algorithm declaration. Since the access to measurements is standardized, and the interaction of an algorithm with the database is done through measurements, this means that such queries can be standardized. This is another ingredient for the automation of the user interface.

Executable definition

Executables must be declared as collections of algorithms, from which the interface of the program is automatically derived. The generated executable:

- generates a template of the algorithm query to be filled by the user;
- feeds the filled query to the algorithm, which is run.

The executable definition is merely a mechanical operation, since the building blocks are already prepared. An example is given here:

```
import qualified Derivative1
import qualified Derivative2

data Program =
  Derivative1 ProgramInput |
  Derivative2 ProgramInput
  deriving Generic

runner :: Program -> IO ()
runner = \case
  Derivative1 input ->
```

```
    runProgramWith csvDatabase input Derivative1.algorithm
  Derivative2 input ->
    runProgramWith csvDatabase input Derivative2.algorithm

main :: IO ()
main =
  genericMain
    "lf-derivatives" -- name of the executable
    "Description of the executable."
    runner
```

This definition would result in an executable `lf-derivatives` with the algorithms `derivative-1` and `derivative-2`. Every algorithm has two command-line options:

- `gen` generates an algorithm query adapted to the algorithm, that just needs to be filled, manually or by the graphical user interface.
- `run` takes as inputs the database location and the algorithm query and runs.

which interact with the two functions of the `runner` functions.

Unified graphical user interface

Given the aforementioned infrastructure, all that is needed is a friendly representation of measurement and algorithm queries. This is done through the graphical user interface, which is composed of two parts: a server and a client.

The server intermediates the relation between the user interface and the executables by:

- transforming requests into executable commands;
- returning the generated query templates;
- running the executables and checking the running state of algorithms.

The client:

- sends the requests for specific algorithms;
- receives the query templates and renders them graphically;
- guarantees that inputs to algorithms are present in the database.

Since inputs can be fed to the algorithm using the current contents of the database, the requests are less prone to errors than hand-written ones. All these functionalities are implemented in the packages:

- `lineageflow-database`;
- `lineageflow-database-meta`;
- `lineageflow-database-csv`;
- `lineageflow-server`;

- `lineageflow-client`.

3.3 Measurement exploration

The effectful infrastructure offers a standard interface for the access to databases and the execution of algorithms. We make use of this structure in order to provide easier interfaces for the visual, statistical and numerical exploration of measurements.

3D visualization

The time-view for temporal lineages corresponds directly to the way images are generated by the microscope, as the evolution in time of a specimen. Therefore, we may visualize directly measurements over temporal lineages through the superposition of the value attributed to a cell to its position. Using this method, we developed the package `lineageflow-view`, where we visualize some types of measurements in 3D, namely:

- Scalars, which can be visualized using a color code;
- Vectors, which can be visualized using arrows, colored by the norm of the vector norm using a color code;
- The symmetric part of tensors can be visualized as ellipsoids whose axes are its eigenvectors. There are different ways to color this ellipsoid, but the simplest one is using the norm of the tensor.
- The antisymmetric part of tensors can be visualized as vectors. For a given tensor M, it is the vector m that satisfies for every vector $v \in \mathbb{R}^3$ the equation $Mv = m \times v$. This gives the plane over which rotations are performed.

The visualization of these measurements can lead to important information about the correlation between the measurements and the morphology of the animal.

We think it is particularly important to relate the visualization of these measurements to the visualization of graphs. While this is a distinct way of interpreting numerical measures, it is similar to the way a physicist interprets a graph by evaluating its convexity, concavity and etc. That is, the absolute identification of values is not as important as the identification of recognizable patterns. Since these patterns have three dimensions, they are more variable and sophisticated than one or two dimensional ones.

Integrated statistical plotting

Even if there are many different types of domains and codomain, structurally, there are not many possibilities. Domains are either composed of simple sets like time or cells or composed of a dependent sum using these types. As showed in the previous section, these types of measurements are represented as arrays or arrays of arrays. Therefore, we can take advantage of these similarities in

order to offer a variety of statistics that only use this structure, with no need of programming knowledge. All graphs in this thesis have been produced using this infrastructure, which shows its versatility.

In general, measurements that correspond to evolutions in time can be represented as graphs or videos, and measurements that correspond to values on cells are better represented as distributions. For example, for any scalar measurement over a temporal lineage, we may plot:

- the graph of the time evolution of means and standard deviations of the measures on the set of cells at each time step;
- the histogram of the global distribution of a measurement;
- the video (or frame sequence) showing the evolution of the histograms corresponding to the local distribution of each time step.

All these features are implemented in the package `lineageflow-plot`.

Interactive REPL

Very often, algorithm developers want to experiment with different ways of plotting and generating measurements, or with partial implementations of algorithms, and this is an essential part of research. The possibility of this interactive experience is one of the main reasons scientists use languages like Python and MatLab, with this kind of interaction with programming being generaly linked to dynamically typed and interpreted languages. While Haskell has a REPL (read-eval-print loop), it is currently not as adapted to persistent environments as those of the mentioned languages. However, given the state of definition of this domain of study, exposed in the previous sessions, this has been overcome. The technical details not being very interesting, we present here the use of this infrastructure, implemented in the package `lineageflow-play`.

In order to use the interactive infrastructure, one has to create a folder, which we will name `project`. Inside this folder, we must have a file where the desired measures are fetched, named `Vars.hs`:

```
velocity5 <- getVector "velocity-5"
velocity10 <- getVector "velocity-10"
```

and another one where the function definitions are made, named `Defs.hs`:

```
import LineageFlow.Prelude
import LineageFlow.Playground

getVector :: Text -> IO (DSumMap Vector)
getVector name =
  retrieveMeasure domain image $
    MeasureQuery
      "species"
```

```
      "specimen"
      "tracking"
      (domainName domain)
      "all"
      (imageName image)
      name

plotNorm :: DSumMap Vector -> IO ()
plotNorm = showPlot . graph . meanWith norm
...
```

and then the REPL can be used in the same way as it is used for other languages, with reloading and redefinition of variables being done with the command `:lf project`.

3.4 Discussion

In this chapter we approached the design of a framework for the manipulation of spatio-temporal lineages with a separation of concerns in mind. We have been able to separate to some degree the development of algorithms, the interaction with the database and the exploration of measurements. Central to this decomposition are the mathematical developments of **Chapter 2**, which allowed the definition of formal *interfaces* between algorithms through the classification of measurements.

We can extend the interface metaphor to the different components of the system, as their points of interaction serve as interfaces between fields of expertise. One can make local improvements based on purely technical concerns and have these improvements propagate through the whole system. This reasoning is very similar to the one that is made in the software development community. However, in this case we are dealing with the interaction of people, which is intermediated only in part by a computational system. We will discover the outcomes of this approach with practice, in the future.

We will discuss further the application of this paradigm to other domains in the general conclusion, so we focus on local improvements in the following.

Research directions

The denotations defined in **Section 3.1** give a guide for the implementation of many manipulations. However, it would be interesting to have a set of denotations that can be considered complete or at least very encompassing. That is, that defines as much as possible the semantics of the possible manipulations on measures. Since these denotations are defined mathematically, this is related to the definition of a set of primite manipulations on domains, proposed in **Section 2.4**.

As discussed previously, there are limitations to the encoding of domain subsets. However, these limitations do not mean that one has to abdicate completely to their identification. A possibility is to recur, for example, to explicit type tags, as it is done in the database for the classification of measurements. The improvement of this identification can be especially important when one will start to compare measurements over multiple embryos.

The database scheme as defined here has a certain degree of arbitrarity. It has been created in this form in order to adapt well to the degree of granularity of the analysis present in the laboratory, but it is not the only possible one. A more thorough analysis of the possible schemes and their advantages would be positive for the future development of the system, and is related to the extension of this methodology to other domains, as proposed in **Chapter 8**.

Finally, the visualization infrastructure can have many improvements. A limitation that is evident in this thesis (**Section 6.3**) is the representation of tensors in 3D, defined by the way we color ellipsoids. The development of a technique that allows the representation of the signals and intensities of the different eigenvalues of the tensor would vastly improve the interpretation of images. In particular, that would make easier the visual distinction between compression and decompression from stress tensors.

Chapter 4

Reference manipulations

In this chapter we discuss two operations that arise often in the application chapters, *homogenization* and *differentiation*. This chapter has no original content, but it has two main purposes:

- to serve as a reference for the following chapters;
- to provide concrete examples of the mathematical structures and types developed in the previous chapters.

Also, it discusses some of the problems that arise on the manipulation of structures that are fundamentally discrete. It is recommended for readers interested on the mathematical formulation of the following chapters.

The first section is devoted to homogenization. While homogenization is not very common in continuum physics, it is essential for the manipulation of discrete structures. This is due to the fact that it allows the *control* of the spatial and temporal scales of measurements. As a consequence, we can control the spatial and temporal resolution of 3D visualizations. Since the interaction of homogenization with non-linear combinations of measurements is non-trivial, visual criteria are used for the decision on how and how much to homogenize. The presentation will focus on the homogenization of the quantities appearing in the application chapters.

The second section is devoted to differentiation. The study of physical measurements like velocities and accelerations make the evaluation of rates of change a necessity. The difficulty we face is that the definition of a derivative is not unique in the discrete case, and each option has different advantages and drawbacks. We will present some of these methods of differentiation and their properties.

Before starting, we discuss *convolution*, since both differentiation and homogenization in time are defined using it.

Convolution

Let $x : \mathbb{N} \to \mathbb{R}$ be a discrete function, n an integer and $f : [-n, n] \to \mathbb{R}$ a filter. The convolution $f * x : \mathbb{N} \to \mathbb{R}$ between them is defined by:

$$[f * x](t) = \sum_{k=-n}^{n} f(k)x(t-k)$$

From this definition one can see the difficulty of the definition of the operation in the case where x is defined in a finite interval, since $x(t-k)$ is not always defined. There are two main techniques to overcome this problem:

- We define time to be cyclic, which eliminates the boundaries. This is inappropriate to this application since the behavior of cells is variable in time.
- We restrict the domain of the convolution only to the points where the formula is well defined, ignoring the boudaries. This is inappropriate to this application since the domain of measurements must remain unchanged in order to correspond to the same object.

Our solution is to define the value of the convolution to be an *undefined* value at the boundary points. This definition preserves the time interval of the measurement at the cost of changing its codomain. We formalize this definition. Let:

- $\mathcal{T} = [t^-, t^+]$ be a discrete time interval;
- S be a semiring with operations $+$ and $\times$ and $\overline{S}$ its extension with an undefined value;

For all times $t \in \mathcal{T}$ we define t_n to be the element that is the n-th successor (or antecessor, for negative n) of t. Given an integer n, a filter $f : [-n, n] \to S$, and a function $\mathcal{T} \to S$ we define the convolution $f * x : \mathcal{T} \to \overline{S}$ to be:

$$[f * x](t) = \begin{cases} \sum_{k=-n}^{n} f(k)x(t_k) & \text{for } t \in [t_n^-, t_{-n}^+] \\ \mathbb{U} & \text{otherwise} \end{cases}$$

In order to be able to apply convolutions to the obtained result, we must give a semiring structure to $\overline{S}$. This can be done with:

$$\mathbb{U} + a = a + \mathbb{U} = \mathbb{U}$$
$$\mathbb{U} \times a = a \times \mathbb{U} = \mathbb{U}$$

The application of multiple convolutions can lead to the proliferation of undefined elements to the codomain. This problem can be solved using *flatten-*

ing. Let $\mathbb{U}$ and $\mathbb{U}'$ be the added elements in $\overline{S}$ and $\overline{\overline{S}}$. We can define a function $\mu : \overline{\overline{S}} \to \overline{S}$ by:

$$\mu(\mathbb{U}') = \mathbb{U}$$
$$\mu(x) = x$$

This allows the convolutions to be applied without leaving the first layer of undefined values.

The implementation of this method can incur into efficiency problems if done without care. For this reason, we define $\overline{\texttt{Double}} = \texttt{Double}$, with $\mathbb{U}$ being an IEEE NaN, which stands for "not a number". It is not difficult to show that this choice obeys both the semiring structure and the flattening of undefined values.

Since `Double` is the representation $\mathbb{R}$, for any filter f of any size, we will consider that the operation $x \to f * x$ has the type $(\mathcal{T} \to \mathbb{R}) \to (\mathcal{T} \to \mathbb{R})$.

4.1 Homogenization

Intuitively, homogenization is a process that reduces the differences between objects, following a given criterion. As commented before, the function of homogenization in this thesis is to control the *scale* of measures, which is evaluated through the visualization of measurements in 3D+time. This same goal has been studied in the context of 2D graphs in domains like signal processing (Antoniou, 2000) and analytical chemistry (Savitzky & Golay, 1964).

In this section we cover the methods of homogenization used in this thesis, namely:

- temporal homogenization of scalar, vectors and tensors;
- spatial homogenization of scalar, vectors and tensors;
- temporal homogenization of cell contacts.

Since the homogenization of vectors and tensors is done coordinate by coordinate, we focus on the homogenization of scalars.

Temporal homogenization

In the temporal setting, the homogenization of scalars is done by a convolution with a filter $f : [-n, n] \to \mathbb{R}$ such that:

- it is symmetric: $f(-k) = f(k)$;
- its total sum is one: $\sum_{k=-n}^{n} f(n) = 1$.

The *naive homogenization* is given by the flat filter $f^n : [-n, n] \to \mathbb{R}$ which for every $k \in [-n, n]$ is given by:

$$f^n(k) = \frac{1}{2n+1}$$

More interesting results are found by the study of continuous functions that have been contaminated by noise. Let $\hat{x} = x + \epsilon$ be a function defined on integers, where ϵ is a noise with no time correlation and unit variance, and $f : [-n, n] \to \mathbb{R}$ a convolution filter. Then one can show that:

$$\text{Var}(f * \hat{x}) = \sum_{k=-n}^{n} f_k^2$$

The minimization of this variance, bound by the moment conditions up to order p:

$$\sum_{k=-n}^{n} k^l f_k = \begin{cases} l = 0 & \Rightarrow 1 \\ l = 1, 2, ..., p & \Rightarrow 0 \end{cases}$$

defines the *Savitzky-Golay filters* (Savitzky & Golay, 1964). A suitable choice of the parameters n and p allows analytical solutions to be found.

Spatial homogenization

The most common way of homogenizing values defined in a set of points in space is through spatial convolution (Goldhirsch & Goldenberg, 2002). There are two types of homogenization, normalized and non-normalized.

Let $\phi : \mathbb{R}^3 \to \mathbb{R}$ be a non-negative function whose integral is equal to 1, and f a real function defined in a set of points $\{x_n\}_{n=0}^N$ in $\mathbb{R}^3$. The non-normalized homogenization of f using ϕ is defined by:

$$\phi(f)(x) = \sum_{n=0}^{N} f(x_n)\phi(x - x_n)$$

Since it conserves the total sum of the function, it is adapted to measurements like mass, whose total sum has to be kept constant.

The normalized homogenization of f using ϕ is defined by:

$$\tilde{\phi}(f)(x) = \frac{\sum_{n=0}^{N} f(x_n)\phi(x - x_n)}{\sum_{n=0}^{N} \phi(x - x_n)}$$

Since it preserves constant functions, it is well adapted to measurements like velocity.

Typically, the function ϕ is a Gaussian function, whose variance controls the width of the homogenization.

Simplicial homogenization

Suppose we have a set of cell contacts, subset of $\sum_{t:\mathcal{T}} \mathcal{C}_t \times \mathcal{C}_t$, which changes often in time and we want to obtain contacts that are less variable. This is an inherently discrete problem, but we can get a continuous equivalent that is easier to deal with.

The first step is to transform the set of contacts into its characteristic function:

$$c : \sum_{t:\mathcal{T}} [(\mathcal{C}_t \times \mathcal{C}_t) \to \mathbb{R}]$$

where $c_t(x, y)$ equals 1 if the cells x and y are in contact in time t, and 0 otherwise. Since every $\mathcal{C}_t$ is a subset of $\mathcal{C}$, we can extend each of these functions with zeroes to $\mathcal{C} \times \mathcal{C}$:

$$\hat{c} : \sum_{t:\mathcal{T}} [(\mathcal{C} \times \mathcal{C}) \to \mathbb{R}]$$

This set is isomorphic to $\mathcal{C} \times \mathcal{C} \to (\mathcal{T} \to \mathbb{R})$, which we can interpret as functions in time for every pair of cells. Now we can use any method of scalar homogenization for each of these function, giving a family of functions $\tilde{c}$ of same type.

The final result is obtained by restricting at each time step the function to the present cells. For each time step t, we denote the restriction of the function $\tilde{c}_t : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ to the domain $\mathcal{C}_t \times \mathcal{C}_t$ by $\bar{c}_t$. The final result is given by the lifted application:

$$\sum_{t:\mathcal{T}} \bar{c}_t : \sum_{t:\mathcal{T}} [(\mathcal{C}_t \times \mathcal{C}_t) \to \mathbb{R}]$$

This result can be manipulated directly, as a weighted contact matrix for each time step, or other methods can be used for the extraction of contacts.

Module interface

In the package `lineageflow-homogenization` there is a type `Homogenizer`, defined as:

```
newtype Homogenizer =
  Homogenizer (Array i Scalar -> Array i Scalar)
```

that can be applied using:

```
homogenizeS :: Homogenizer -> Array i Scalar -> Array i Scalar
homogenizeV :: Homogenizer -> Array i Vector -> Array i Vector
homogenizeT :: Homogenizer -> Array i Tensor -> Array i Tensor
```

Each method presented above is implemented using this interface in the modules `LineageFlow.Homogenization.Time` and `LineageFlow.Homogenization.Space`.

4.2 Differentiation

The concept of differentiation is used widely in physics for the calculation of rates of change. Its continuous definition is given by:

$$f'(x) = \lim_{x' \to x} \frac{f(x') - f(x)}{x' - x}$$

However, its direct discrete translation:

$$\Delta f(n) = \frac{f(n + \Delta n) - f(n)}{\Delta n}$$

does not have good properties for general use.

All the methods used here consist of a convolution with a filter which is called a *differentiator*. In the following we present different differentors and their properties.

Since the differentiation of vectors and tensors is done coordinate by coordinate, we focus on the differentiation of scalars.

Naive differentiators

The forward differentiator of size n is given by:

$$f(k) = \begin{cases} k = 0 & \Rightarrow -1/n \\ k = n & \Rightarrow 1/n \\ \text{otherwise} & \Rightarrow 0 \end{cases}$$

giving:

$$[f * x](k) = \frac{x(k + n) - x(k)}{n}$$

As commented before, this is not a good choice in general, as it is greatly amplifies noise. However, when noise is the quantity to be studied, like in **Chapter 7**, it is well adapted.

Lanczos differentiators

Lanczos differentiators follows from an integral definition of derivation in the continuous case:

$$f'(x) = \lim_{h \to 0} \frac{3}{2h^3} \int_{-h}^{h} y f(x+y) dy$$

They are optimal in the case where the sampling corresponds to a smooth function that has been disturbed by uncorrelated noise (McDevitt, 2012). For this reason, they are closely related to the Savitzky-Golay filters presented in the previous section (Savitzky & Golay, 1964). Their derivation is done in a similar way.

Let $\hat{x} = x + \epsilon$ be a function defined on integers, where ϵ is a noise with no time correlation and unit variance, and $f : [-n, n] \to \mathbb{R}$ a convolution filter. Then one can show that:

$$\text{Var}(f * \hat{x}) = \sum_{k=-n}^{n} f_k^2$$

The minimization of this variance, bound by the moment conditions up to order p, which are:

$$\sum_{k=-n}^{n} k^l f_k = \begin{cases} l = 1 & \Rightarrow 1 \\ l = 0, 2, ..., p & \Rightarrow 0 \end{cases}$$

defines the *Lanczos differentiators*. A suitable choice of the parameters n and p allows analytical solutions to be found.

Holoborodko differentiators

Holoborodko differentors aim to approximate low-pass filters in using filters of finite size (Holoborodko, 2008). This approximation is done by the study of the power spectrum of the filter, which is chosen in order to reproduce the derivative at low frequencies, with decaying power at higher frequencies. For this reason, these filters are adapted to cases where one wants to suppress high frequencies or where one knows that a certain range of frequencies is infected.

Let $f : [-n, n] \to \mathbb{R}$ be an antisymmetric convolution filter. Its frequency response in the Fourier spectrum is:

$$\tilde{f}(\omega) = 2i \sum_{k=1}^{n} f(k) \sin(k\omega)$$

Since the frequency response of the regular derivative, whose corresponding filter is called d, is $\tilde{d} = i\omega$, we define the tangency system of order i and j:

$$\begin{cases} \frac{\partial^k \tilde{f}}{\partial^k \omega}(0) = \frac{\partial^k \tilde{d}}{\partial^k \omega}(0) & \text{for } k = 0, ..., i \\ \frac{\partial^k \tilde{f}}{\partial^k \omega}(\pi) = 0 & \text{for } k = 0, ..., j \end{cases}$$

A suitable choice of the parameters n, i and j allows analytical solutions to be found.

Module interface

In the package `lineageflow-derivatives` there is a type `Deriver`, defined as:

```
newtype Deriver =
  Deriver (Array i Scalar -> Array i Scalar)
```

that can be applied using:

```
deriveS :: Deriver -> Array i Scalar -> Array i Scalar
deriveV :: Deriver -> Array i Vector -> Array i Vector
deriveT :: Deriver -> Array i Tensor -> Array i Tensor
```

Each differentiator presented above is implemented using this interface in the modules `LineageFlow.Derivatives.Naive`, `LineageFlow.Derivatives.Lanczos` and `LineageFlow.Derivatives.Holoborodko`.

Chapter 5

Clustering cell trajectories

One of the great transversal subjects in complex system science is multi-scale dynamics, the study of the relation between the dynamics of microscopic and macroscopic observables. Many approaches to this subject exist, differing both in formalism and application, ranging from thermodynamics (Callen, 1960) and statistical mechanics (Grabert, 1982), to dynamical systems theory (Jacobi, 2009) and to automata and multi-agent systems (Doolen, 1991).

A requisite for this kind of study is being able to describe the objects that compose the different scales. This may be a problem, for example, in the social sciences, where the identification and recognition of differents groups of people can be very difficult. In embryogenesis however, there is a large amount of work dealing with both scales that are studied in this thesis (Alberts et al., 2002). Microscopically there are cells and their observed dynamics and macroscopically, morphogenetic fields and tissues, which are composed by groups of cells. However, we have no automatic way of identifying such fields.

In general, the description of morphogenetic fields and tissues in the biological literature is verbal, visual and schematic, often taking the form of an *atlas* or *fate map*, which is a mapping of territories over the embryo. An example of such representation is given in Fig. 5.1. However, in this thesis we are interested in numerical measurements taken over the embryo and their relation to the morphology of the animal. The goal of this chapter is to build a bridge between the two languages, through the transformation of numerical measurements into groups of cells that share similarities. Besides offering insights on the formation of tissues, this transformation can offer a middle ground for the collaboration between biologists and physicists, making the interpretation of some measurements easier.

Before introducing the algorithm, we discuss the implications of such a transformation. Morphogenetic fields can be seen as groups of cells that behave similarly. Therefore, the interpretation of the generated cells groups as morphogenetic fields implies the interpretation of the set of measurements provided to the algorithm as a *quantification* of the behavior of cells. This leads to two com-

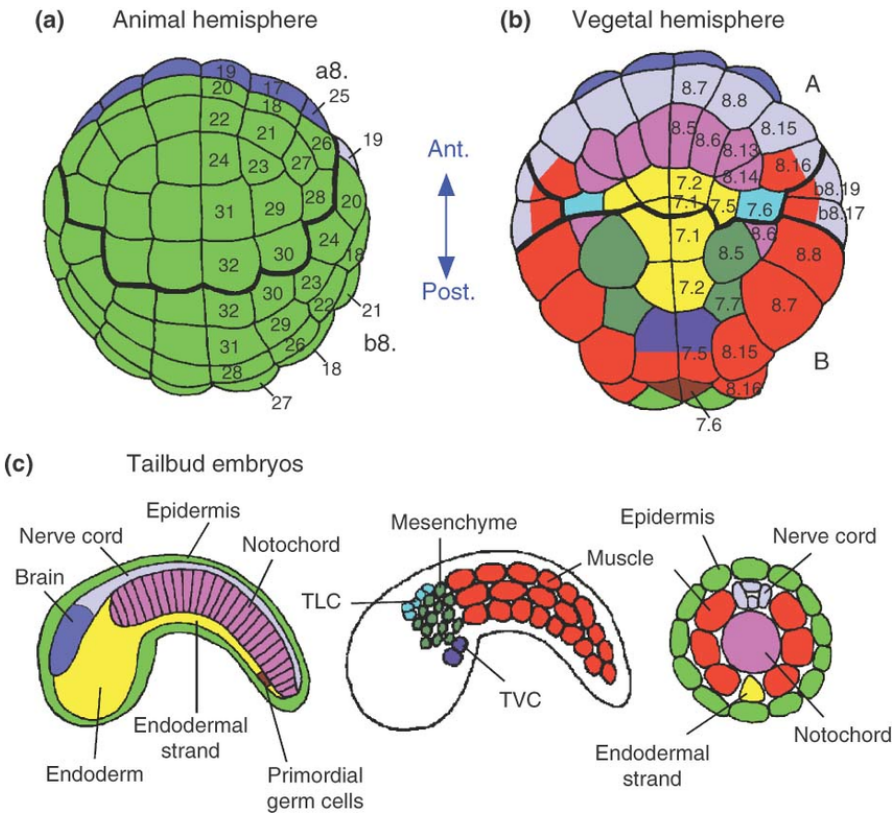

Figure 5.1: **Example of a fate map of the ascidian** (Makabe & Nishida, 2012). In this image we can see the distinctive aspects of the representation of tissues. The regions are represented graphically, in superposition to the animal morphology and referred to by name. We aim to translate numerical measurements into a similar representation.

plementary views on the transformation.

On one hand, the algorithm can be applied to any set of measurements. Therefore, the mechanical transformation of these into homogeneous groups of cells can be seen as an *induced emergence* of macroscopic observables. In this case, the induced observables can be interpreted as *artificial* morphogenetic fields, which take into account only the chosen measurements.

On the other hand, we expect that a diligent choice of a number of descriptive measurements can be used to represent the behavior of cells to a good degree of accuracy. We expect this ideal set of measurements to be transformed into the actual morphogenetic fields of the animal. This judgement provides a search strategy for the best choice of descriptive measurements, whose corresponding artificial morphogenetic fields match as well as possible the morphological ones. Given the different morphologies of animals, the best set of measurements is bound to be highly dependent on the species and the period of the embryogenesis being studied.

We proceed now to the criteria used for the determination of morphogenetic fields. Independently of species or moment in time, the substract of embryogenesis is the same: cells interacting in space. Given this fact, we have a number of expectations on the groups that are to be found.

- *Spatial contiguity.* As the fate maps show, morphogenetic fields are connected in space. This connectivity is explained by the necessity of communication and coherence on the morphogenetic fields for the execution of morphological functions.

- *Temporal persistence.* The formation of compartments is a progressive phenomenon and these compartments do not disappear. We expect the existence of similarities between cell behaviors at early stages and their consolidation while the groups are formed.

- *Permanence of the lineage into the group.* Given the strong role of genetic expression on the behavior of cells, we expect cells to remain inside a given morphogenetic field, and cell interchange between them to be an exception. Furthermore, since there is a persistence of expression even after mitosis, the same is expected from the children of cells in a given group.

Our goal is to find a method to transform a set of measures into groups that obey more or less closely these properties. The strategy used here is to define a *dissimilarity* between cells through the comparison of measurements at each time step. The use of comparisons that are local in time only is justified by the fact that the global dynamics of the embryo changes very strongly during the development of the animal. The method used for the composition of this dissimilarity is inspired from path integrals in quantum mechanics and stochastic processes (Feynman, 1965).

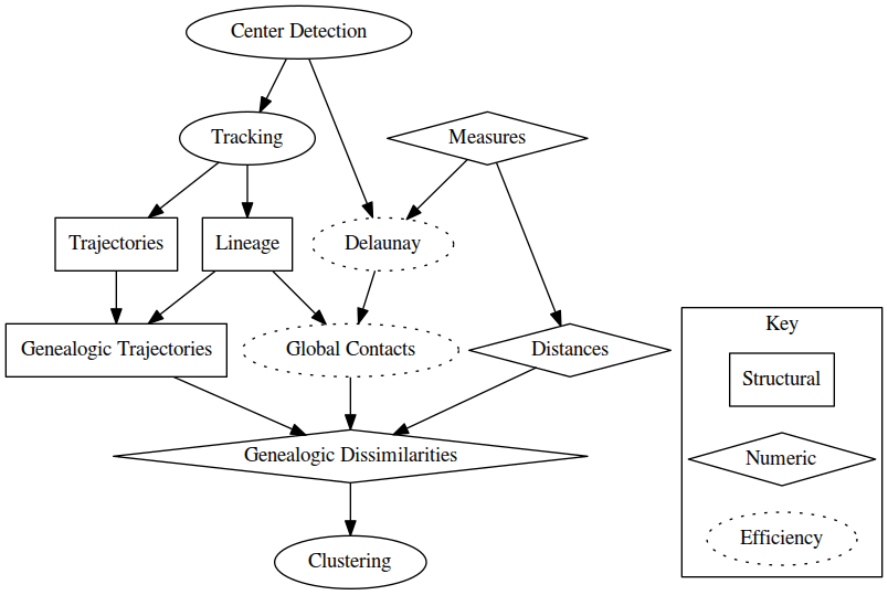

Figure 5.2: **Scheme describing the algorithm steps.** Boxes are classified as structural, numeric, or efficiency. Structural elements are the ones related only to temporal lineages. Numeric elements are measurements taken over the temporal lineage. Efficiency elements are optional steps that can be used in order to reduce the computational cost of the algorithm.

Description

The algorithm can be decomposed into the following steps, represented in Fig. 5.2:

1. To select the measures to be used for clustering. In this chapter we chose cell positions coming from the cell identification and velocities calculated using the methods described in **Chapter 4**. This choice is justified by the contiguity requirement for morphogenetic fields and the coherent movement of cells towards the same region.

2. To calculate *genealogic dissimilarities* between cell trajectories, as described in the following sections;

3. Optionally, to calculate these dissimilarities only between a set of *neighbors*. This step reduces the computational complexity of the task and is crucial for the application of the algorithm to full zebrafish data sets;

4. To use these dissimilarities as an input for a clustering algorithm.

We now proceed to the description of these steps.

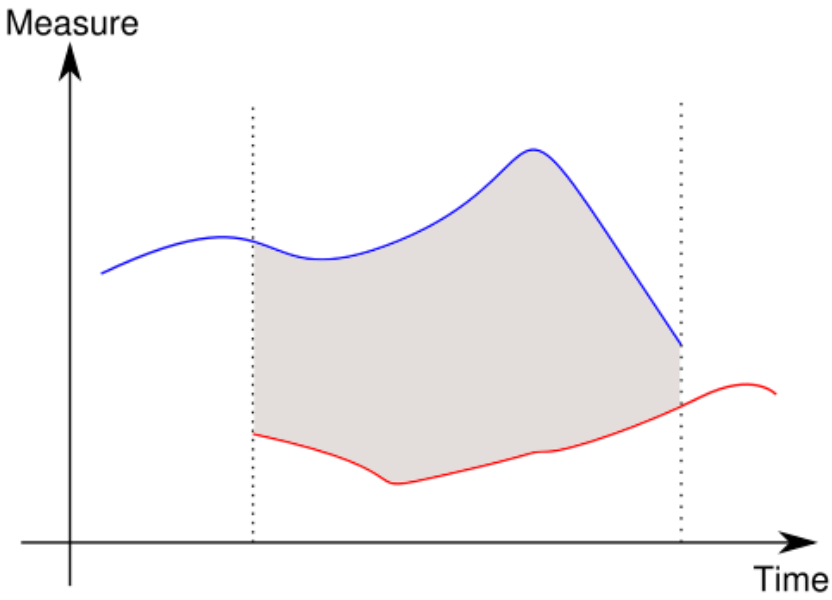

Figure 5.3: **Example of dissimilarity calculation.** The lines represent the evolution in time of the value of a measurement for two cells (blue and red). The dissimilarity is calculated as the average dissimilarity between the two values in the time period where both cells are present (grey region between dotted lines).

Genealogic dissimilarity

A *dissimilarity* is a function that takes two similar inputs and gives a non-negative value as output. Some examples of dissimilarities between cells at a given time step are:

- their distance;
- the absolute value of their velocity difference;
- the absolute value of the difference of intensities of some marker.

In general, the absolute value of the difference of the values of any measurement is a dissimilarity. This means that any numerical measurement offers a criterion of comparison between cells.

As mentioned before, the comparison between cells is done only at fixed moments in time. In order to build a global dissimilarity between cells, we must combine the dissimilarities calculated at each time step into a single value. This is done using the *average value* of the dissimilarities calculated in the time steps where both cells are present. If there is no moment in time where both cells are present, this value is left undetermined. In Fig. 5.3 we see an example of this process.

This method has a clear shortcoming: we cannot compare mothers to children, since the appearance of one implies the disappearance of the other. In order to

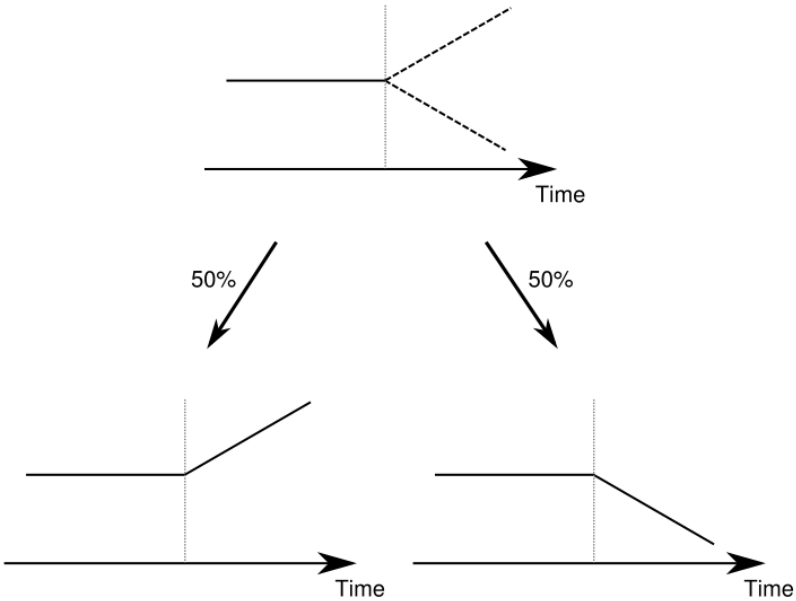

Figure 5.4: **Example of genealogic trajectory.** In the upper part of the image we see the trajectory of a cell (bold line) and the trajectory of its children (dashed lines). There is 50% of chance for the trajectory of the mother to be extended by that of either of its children, resulting in the two trajectories (bold lines) in the lower part of the image.

overcome this problem, we introduce the concept of a *genealogic trajectory*. The genealogic trajectory of a cell is given by its own trajectory, extended by the trajectory of its children, which in turn are extended by their children and so on. This extension is done in a probabilistic manner. If the cell divides, there is a 50% of chance for its trajectory to be extended by one of its children. If the cell has no children, the genealogic trajectory is just the regular one. This process is exemplified in Fig. 5.4.

This construction may remind physicists of *branching processes* (Grimmett & Stirzaker, 2001). However, the construction of genealogic trajectories is more related to a *decision tree*, as it consists on the choice of branches on a preexisting structure, while branching processes study the creation of these structures.

We proceed to the calculation of the dissimilarities between these genealogic trajectories. Since they have been extended, we can now compare mothers and children. However, since these trajectories are random varibles, the dissimilarities themselves are random variables. In order to transform these variables into numbers, we take the average value of each dissimilarity given the possible trajectory choices. The resulting value is called the *genealogic dissimilarity*, which is the value that will be used for clustering. In particular, the genealogic dissimilarity between a mother and one of its children is half the dissimilarity between its children.

In order to use multiple measurements for the clustering, we should mix the multiple dissimilarity calculations into a single one. This is done using a weighted mean, which gives a number of *mixing parameters* that have to be provided by an user.

Efficiency

The computation time and memory space needed for the calculation of these dissimilarities are very high. In order to lower the requirements, we can retrict the calculation only to *neighboring* cells. We say that two cells are neighbors for a given measurement if their respective values are neighbors in the image of the measurement, at some moment in time. This definition assures the choice of pairs of trajectories that have the greater probability of belonging to the same group.

Clustering

For the clustering of trajectories, we may use any algorithm that accepts as input a matrix of dissimilarities. Our method of choice the *spectral clustering* algorithm exposed in (Ng, Jordan, & Weiss, 2001). It is adapted to this application since it can be calculated using sparse dissimilarity matrices in a very efficient way (Luxburg, 2007). This efficiency is essential since it allows the experimentation of different mixing parameters. Besides the matrix of dissimilarities, a desired number of clusters must be provided as input.

5.1 Formulation

We present here the mathematical formulation of genealogic trajectories and dissimilarities. The definitions are based on the algebraic definition of *intersection* and *concatenation* presented in **Section 2.3**.

Genealogic Trajectories

A *genealogic trajectory* is a cell trajectory that has been extended probabilistically by the trajectories of its children. For every cell division, one of the children is chosen at random and its trajectory is concatenated with that of its mother. This process is repeated for each children.

The formal definition is recursive. For any trajectory X we define i_X to be an equiprobable zero-one random variable. The genealogical trajectory $\widetilde{X}$ is a trajectory-valued random variable such that

- If X has no siblings, $\widetilde{X} = X$;
- If X has siblings Y_0 and Y_1 then $\widetilde{X} = X \vee \widetilde{Y}_{i_X}$.

We impose the condition that for two different cells X and Y, i_X is independent of i_Y. This allows the calculation of joint probabilities using multiple cells.

Genealogic Dissimilarity

In the following, $\mathcal{C}$ is the set of cells and $\mathcal{T}$ the set of time steps. For any set X, we represent an undetermined value of type X by $\overline{X} = X \cup \{\mathbb{U}\}$.

A dissimilarity in $\mathcal{C}$ is a symmetric non-negative function $d : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$. Given

- two trajectories X and Y with values in a set S, whose intersection is $X \wedge Y = (K, z)$;
- a dissimilarity d in S;

we define the *partial dissimilarity* P_d between two trajectories as

$$P_d(X, Y) = \left(|K|, \int_K [d \circ z](t)\, dt \right)$$

The set $\mathbb{R}^2$ has a natural monoidal structure given by the sum of coordinates represented by $\oplus$. By giving the following monoidal structure to $\overline{\mathbb{R}}^2$

$$a \oplus b = \begin{cases} b & \text{if } a = \mathbb{U} \\ a & \text{if } b = \mathbb{U} \\ a \oplus b & \text{otherwise} \end{cases}$$

we can prove that P_d satisfies the following

$$P_d(X \vee Y) = P_d(X) \oplus P_d(Y)$$

We define the *trajectory dissimilarity* $D_d : Tr(T,S) \times Tr(T,S) \to \overline{\mathbb{R}}$ as

$$D_d(X,Y) = \overline{\text{div}}\, P_d(X \wedge Y)$$

where div $(m,s) = s/m$, considering $0/0 = \mathbb{U}$.

The *genealogical dissimilarity* between two trajectories is defined as

$$\widetilde{D}_d(X,Y) = \mathbb{E}[D_d(\widetilde{X},\widetilde{Y})]$$

where $\mathbb{E}$ denotes expectation.

Finally, we define the *similarity* $S_{d,\sigma} : Tr(T,S) \times Tr(T,S) \to \mathbb{R}$ as

$$S_{d,\sigma}(X,Y) = \exp\left[-\frac{\widetilde{D}_d(X,Y)^2}{2\sigma^2}\right]$$

where σ is a scale factor. This is the measurement that is used as input to the clustering algorithm.

Properties

The distributive law for $\vee$ and $\wedge$ gives a way to decompose genealogical dissimilarities into regular dissimilarities. Genealogical dissimilarities can be calculated from the matrix of pairwise partial dissimilarities:

$$P_d(Z \wedge (X \vee Y)) = P_d(Z \wedge X) \oplus P_d(Z \wedge Y) \tag{5.1}$$

Let us consider a simple lineage with a cell M, two daughters C_1 and C_2 and a cell N with no daughter. If M and N do not coexist at any moment in time, Eq (5.1) states that

$$\widetilde{D}_d(M,N) = \frac{\widetilde{D}_d(C_1,N) + \widetilde{D}_d(C_2,N)}{2}$$

In particular

$$\widetilde{D}_d(M, C_1) = \frac{\widetilde{D}_d(C_1, C_2)}{2} = \widetilde{D}_d(M, C_2)$$

This equation shows that the mother is equidistant to its daughters and that if both daughters stay close to each other, their mother stays close to both of them, leading to a degree of coherence between their clustering allocations.

Another important property is given by the application of Eq. 5.1 and the addition of a point in time for Z. This can be written as

$$P_d(X \wedge (Z \vee Z')) = P_d(X \wedge Z) \oplus P_d(X \wedge Z')$$

This is an important property that allows the incremental calculation of partial dissimilarities from past partial dissimilarities and new data points.

Derived dissimilarities

A *derived dissimilarity* allows one to use a measurement as the criterion for the measure of dissimilarity between cells. From any measurement $m : \sum_{t:\mathcal{T}} \mathcal{C}_t \to M$ where M is a metric space with a distance d, we can derive a family of dissimilarities $d_m : \sum_{t:\mathcal{T}} \mathcal{C}_t \times \mathcal{C}_t \to \mathbb{R}$ given by:

$$d_m(t, c_1, c_2) = d(m(t, c_1), m(t, c_2))$$

Since there is no guarantee that $d_m(c_1, c_2) = 0$ implies that $c_1 \equiv_{\mathcal{C}_t} c_2$, then we cannot say this is a metric. The triangular inequality is preserved, but it is lost in the genealogic dissimilarity.

This is the procedure we use for the generation of dissimilarities. In this context, we can speak about the clusters generated by a set of measurements, leaving the dissimilarities implicit.

5.2 Implementation

Generation of genealogic trajectories

As explained before, the genealogic trajectory of a cell is a trajectory-valued random variable and the dissimilarity is calculated as an expectation over pairs of these variables. Since we calculate this expectation explicitly, we must generate the full set of pairs of trajectories.

The fact that the probabilities of branching between divisions are independent allows the calculation of this expectation. However, much care must be taken when comparing two genealogic trajectories. Notably, if two cells are in the same lineage, their trajectories must coincide from some point in time.

The simplest example of this situation is given by a single cell M that divides into C_1 and C_2. The dissimilarity $\widetilde{D}(M, M)$ is given by:

$$\frac{D(M \vee C_1, M \vee C_1) + D(M \vee C_2, M \vee C_2)}{2}$$

that is, there are no mixed terms involving both C_1 and C_2.

Since the algorithm for the generation of this set is not so simple, we describe its general lines below in Haskell-like pseudocode. We encode a lineage recursively as:

```
data Lineage =
  Last Cell |
  Divided Cell Depth Lineage Lineage
```

that is, a lineage may be composed of just a cell (identified by its ID number), or a cell that divides, identified by its ID number, the depth of its tree and the lineage of its children. The depth of the tree is zero at the end of the lineage, and grows for every antecessor. We suppose that we already transformed each cell into its this representation.

We aim to obtain a function that returns a set of pairs of trajectories from a pair of lineages, named `pairs`. A trajectory is just a sequence of cells. In order to define this function, we have to study three cases.

Both cells do not divide

The only possible pair of trajectories is the pair of cells themselves.

```
pairs (Last id1) (Last id2) = {([id1],[id2])}
```

Only one of the cells divides

In this case we are sure that the cells are different, and we can recurse on both branches of the cell that divides.

```
pairs (Last id1) (Divided id2 _ lineage_left lineage_right) =
  add id2 to the right of pairs in
    pairs (Last id1) lineage_left ++
    pairs (Last id1) lineage_right
```

Both cells divide

In this case we use the depth of the lineage. If the lineages have the same depth, we compare their `id`. If the `id`'s are different, we continue as if the cells had different depth. If the `id`'s are equal, we recurse over each side separately:

```
pairs (Divided id _ lineage_left lineage_right)
      (Divided id _ lineage_left lineage_right) =
  add id to both sides of pairs in
    pairs lineage_left lineage_left +
    pairs lineage_right lineage_right
```

If they have different depths, we recurse over the branches of the deepest one:

```
pairs lineage_short
      (Divided id_long _ lineage_long_left lineage_long_right) =
  add id_long to the right set in
    pairs lineage_short lineage_long_left +
    pairs lineage_short lineage_long_right
```

Clustering and mixing measures

The clustering algorithm uses a single similarity as input (Luxburg, 2007). In order to be able to use this technique for multiple measures we need to be able to mix different similarities into one.

For every measure m we have a dissimilarity d_m and consequently a genealogic dissimilarity and a similarity between cells $S_m : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$. We give a weight w_m to every measure and by defining the standard deviation of the entries of the genealogic similarity matrix by σ_m we define the *mixed similarity* by:

$$S_w(c_1, c_2) = \frac{\sum_m w_m \frac{S_m(c_1,c_2)}{\sigma_m}}{\sum_m w_m}$$

This allows to mix different similarities in a more natural way, giving them weights that are independent on the unit we use to measure the different values. That is, if we use two measures and give them the same weight, we expect the global similarity to be composed in equal proportions of each similarity.

Neighborhood and sparsity

The calculation of the matrix of dissimilarities has size and running time of order $O(C^2)$ in the number of cells. While this is possible to calculate for subsets of the embryo, it becomes prohibitive for the full data sets.

In order to decrease the total running time and needed memory for the algorithm, we only calculate dissimilarity between *neighbors*. We say that two cells are neighbors if at some moment in time they are neighbors for the local Delaunay tesselation in the image of the measurement. This guarantees the proximity of cells that are going to be used for clustering. Since the number of neighbors of a cell does not change when expanding the spatial volume of a data set, the complexity grows more slowly.

5.3 Results

For the exposition of the method, we will use the following data sets, shown in Table 1, composed of three wild types and two *oep* mutants.

Table 5.1: **Data sets used for the analysis**. Three wild-types (WT) and two $oep^{tz57/tz57}$ (cyclopic) mutants have been studied. The voxel and timestep size entries correspond to the parameters used by the microscope. The starting and ending time of each data set has been chosen in order to obtain similar development phases at the same time step.

ID	Genotype	Voxel size (μm)	Time step size	Start	End
071226a	*WT*	1.37	2min33sec	7h31	13h03
140523aF	*WT*	1.26	2min28sec	7h52	13h13
141108aF	*WT*	1.38	2min26sec	6h54	12h10
081018a	$oep^{tz57/tz57}$	1.31	2min31sec	8h15	13h42
120914aF	$oep^{tz57/tz57}$	1.51	2min31sec	6h45	12h12

In all applications of the method, we used the genealogic dissimilarities derived from positions and velocities. These dissimilarities have been mixed with a parameter λ, ranging from 0 to 1. When λ is 0 all weight is given to velocity and when λ is 1 all weight is given to position. In most cases, $\lambda = 0.5$ is used. We denote the number of clusters provided to the algorithm by k. When evolution in time is showed, we use the end shield, bud stage and 6-10 somites stages of the zebrafish development (C. B. Kimmel et al., 1995).

The application of the algorithm to the wild type 141108aF (Fig. 5.5) has some interesting aspects. The first and most important one is that clusters are persistent. That is, from the moment that a cluster appears in the data set, it stays until the end. This shows that this method is able to satisfy the criterion of temporal persistence of the identified clusters.

An interesting point is that for a number of clusters equal to 2, the embryo is segmented through its bilateral symmetry axis. This emergence is not imposed by the method, but a natural emergence from the data. Furthermore, as the number of clusters grows, their bounds roughly coincide with the visual contours of the embryo. This shows that these groups respect to some extent the morphology of the animal.

Another notable observation is the appearance of *layers* of cells in the latest stages of development. This superposition of groups of cells is due to the use of velocity for the generation of clusters and the relative drift of these groups.

The results of the algorithm depend on the persistence of cells inside the field

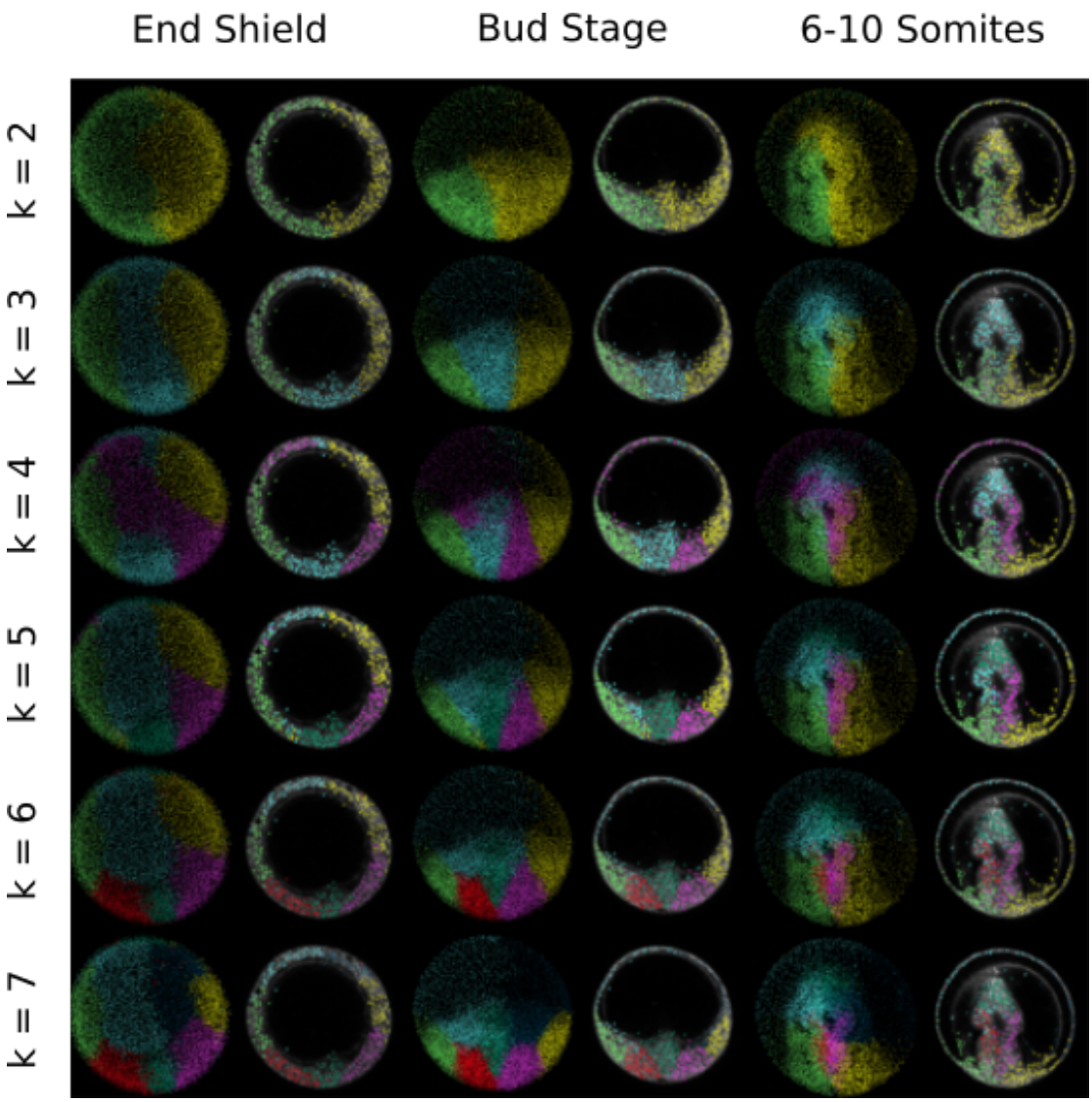

Figure 5.5: **Coherence of patterns revealed by the clustering algorithm.** Results of the application of the algorithm for the wild type 141108aF with a fixed $\lambda = 0.5$. The result is represented by the 3D rendering of the embryo with different colors representing different clusters. The choice of colors has no special signification, but is chosen in order to help the visualization of the evolution of clusters with the change of parameters. Each row corresponds to the number of clusters given as parameter to the algorithm, ranging from 2 to 7. In each line we see the corresponding view from the animal pole and an orthoslice at 6h54 (End Shield), 9h31 (Bud Stage) and 12h10 (6-10 Somites) hpf (hours post fertilization).

of view, as shown by the comparison between the wild types 071226a, 140526aF and 141108aF (Fig. 5.6). As it can be seen, the algorithm does not completely eliminates the possibility of temporal segmentation. This is due to the partiality of the data set and the flow of cells through the boundary of the field of view. This limitation is not absolute, as 141108aF shows full time persistence of clusters, even if there is some flow through the boundaries. However, we can see similar patterns emerging at comparable moments of the development.

The comparison of results of the application of the algorithm between wild types and *oep* mutants (Fig. 5.7), shows fairly similar patterns that have some differences. The observation that the frontal part of the head of the wild type is segmented in two symmetrical parts, while the *oep*'s has a central distinguished region can correspond to the cyclopy of this specimen. However, we have no sufficient evidence to prove this explanation.

Parameter exploration

An exploration on the parameters λ and k shows how much they change the resulting patterns in Fig. 5.8 and Fig. 5.9. Going from the left to right, we increase the importance that is given to position on the calculation of clusters. Comparing the first column with the last one, we see that clusters from last column have more visible boundaries and are less spread through the embryo. This is very important, since that means that our intuition about the measurements is translated well into the clusters.

In the exploration for 120914aF (Fig. 5.9) we see that for $\lambda = 0$ and $k = 2$ the algorithm does not identify bilateral symmetry. This shows that bilateral symmetry is a decurrence of the weight of position in the algorithm.

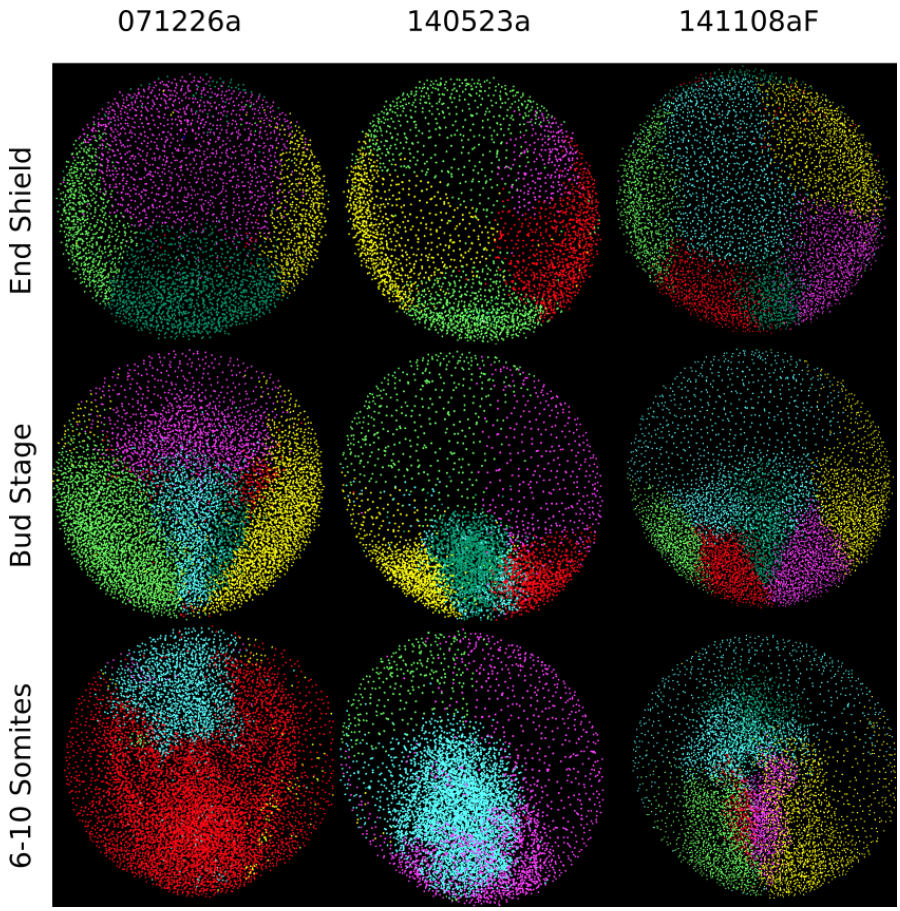

Figure 5.6: **Comparison of patterns on different wild types.** Results of the application of the algorithm with fixed $\lambda = 0.5$ and $k = 6$ to embryos 071226a (left), 140523aF (middle) and 141108aF (right) at three different stages of embryogenesis. All the embryos are represented from the animal pole in vertical position. Different colors represent different clusters for each embryo. The choice of colors is arbitrary and has no specific meaning. In both left and middle specimens we see the influence of the entrance and exit of cells from the field of view for the algorithm, which breaks the temporal coherence of the cell lineages and consequently the temporal robustness of the method.

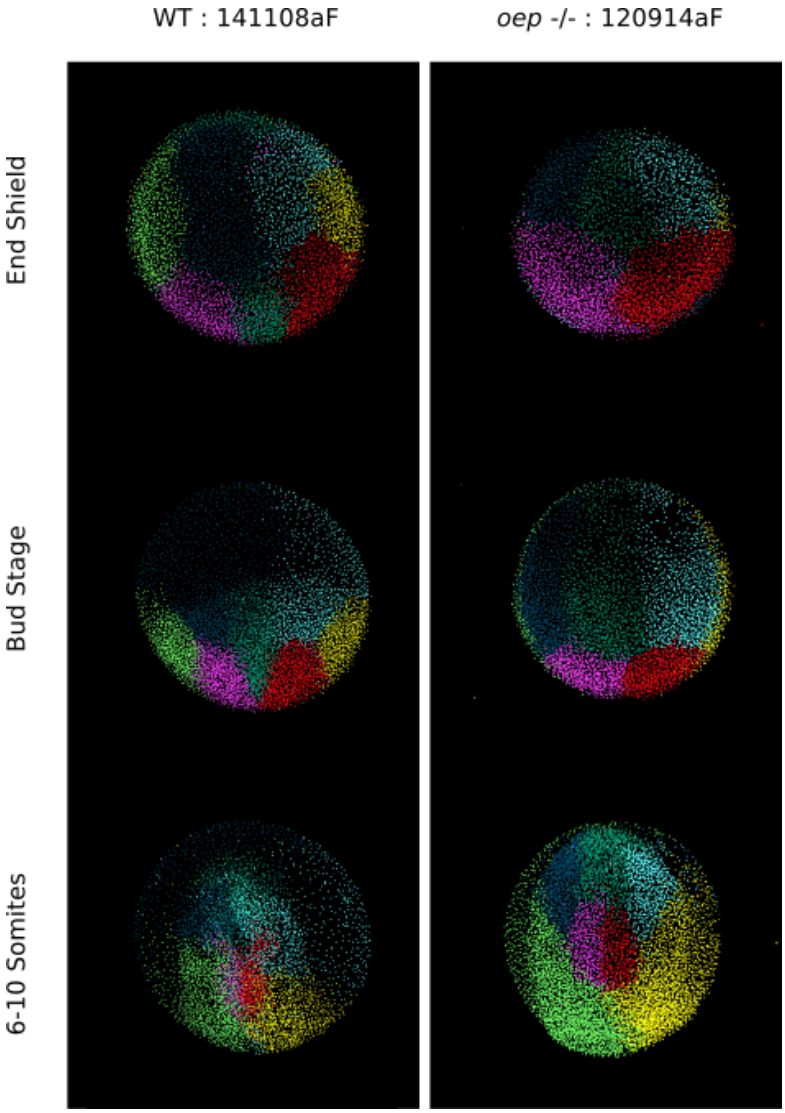

Figure 5.7: **Comparison of patterns between wild type and *oep*.** Comparison of the application of the algorithm for the wildtype 141108aF (left) the *oep* 120914aF (right) for $\lambda = 0.25$ and $k = 7$ at three different stages of embryogenesis. The application of the algorithm shows similar distributions of clusters in both embryos, but with different characteristics corresponding to their different natures.

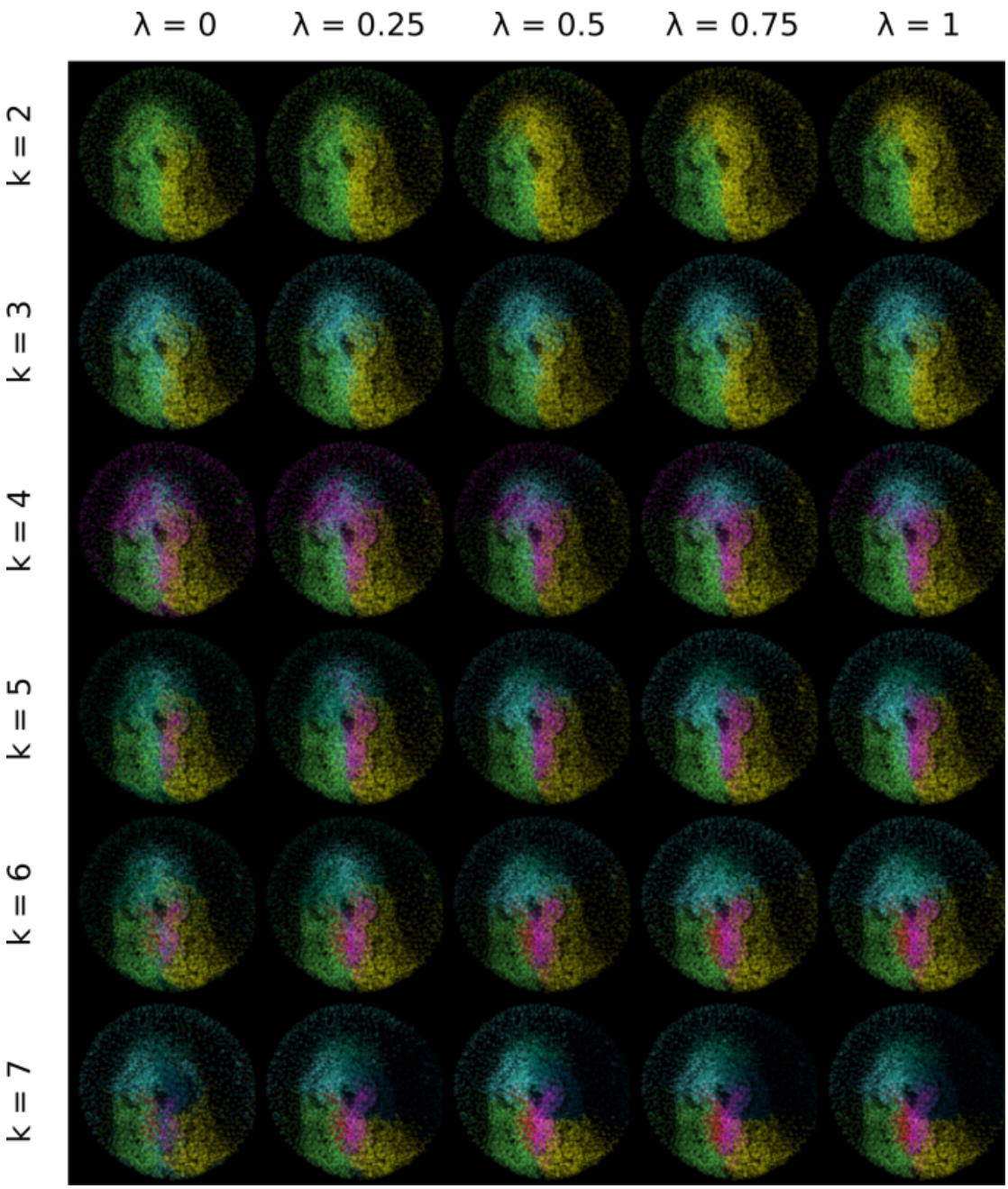

Figure 5.8: **Parameter exploration for the wildtype 141108aF.** This figure shows the results of the clustering algorithm on the same embryo for various parameters. Each row corresponds to a number of clusters (varying from 2 to 7) and each columns to a λ, the value that interpolates the weights of positions and velocities. The importance of position grows from left to right. It can be seen that the clusters on the right side of the table have the tendency to be less spread through the embryo. This agrees with the intuition of the parameter giving more importance to position than velocity.

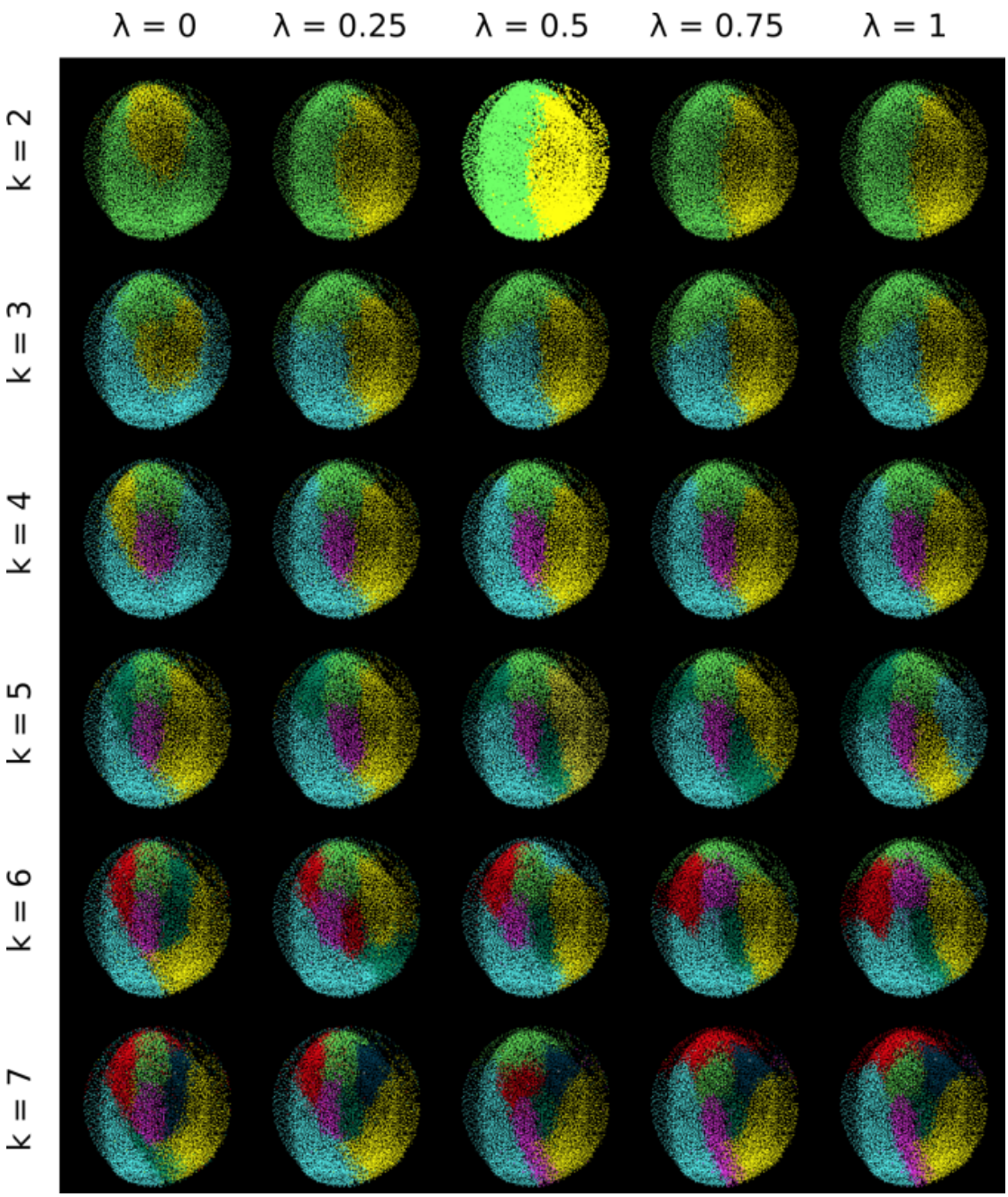

Figure 5.9: **Parameter exploration for the wildtype 120914aF.** This table shows the results of the clustering algorithm on the same embryo for various parameters. The parameters and disposition are the same as the ones used in Fig. 5.8. The upper left image shows that bilateral symmetry is not always found when $k = 2$.

5.4 Discussion

The goal of the development of this algorithm was the transformation of sets of measurements into a partition of the embryo in a way that is coherent with the interpretation given to these measurements, possibly matching the morphology of the animal. We discuss here our accomplishments and shortcomings.

In the cases where no temporal segmentation of the embryo was generated, the results are remarkably positive. All clusters are visible, spatially localized and evolve in time in way that is visually natural. Furthermore, we can see coherent domains that reveal some morphological landmarks, most notably, bilateral symmetry. Also, the way groups evolve with the change of weights seem coherent with the interpretation given to these measures. In particular, greater weights given to position is translated into more compact clusters.

Some results were partially negative. The temporal persistence of clusters depends on the flux of cells through the boundary of the field of view, which may cause the generation of ephemeral clusters. Ideally, having full data sets would eliminate this problem as all the lineages would be completely connected and every pair of cells would be comparable. However, we note that the temporal segmentation happens at a specific moment, before which the clusters are persistent.

The algorithm in its present form can take into account an arbitrary number of measurements. This is positive, as one can refine the set of clusters by progressively adding new measurements that seem relevant for the morphologic process being studied. However, the way we combine these measurements is arbitrary, which augments the space of uncertainties of the algorithm.

Finally, one of the features this algorithm offers is the possibility of *augmented observation*. Using this method one can visualize multidimensional set of measurements through their generated clusters, which are much easier to interpret. This is already visible in the results, where one can observe *layers* of cells due to the influence of velocity in the clustering. However, given the nature of observation, only the experience of systematically applying the method and verifying the results will bring more examples of concrete applications of this idea to light.

Research directions

There are several ways of expanding the study done in this chapter.

The first one is the experimentation with different species. As said before, the use of the algorithm in embryos with different characteristics will show more clearly the effects of the choices of measurement made here. For one part, the embryogenesis of some animals like the sea urchin exhibits cellular divisions

which are more proeminent than that of the zebrafish. Also, others animals like the ascidian have fate maps that are more well known, making it possible to perform better comparisons with the expected results. Furthermore, since the zebrafish has a relatively large embryo, the task will be computationally less demanding.

Another related next step is the use of different measurements as input. Of great relevance are the values coming from gene expression markers. Given the body of knowledge of the relation between the expression of certain genes and the formation of embryos, these would offer good criteria for the quality of the results.

Finally, we can create control groups for the application of the algorithm. This can be done through the generation of data by some controlled model where there are clearly distinct groups with known relevant parameters. The application of the method to this data would offer a valuable reference on its limitations.

The output of this algorithm is given in the form of groups that are homogeneous by some criterion. This property can be used for other studies, such as methods that use the uniformity of groups for the definition of parameters. For example, one may want to study a model of adhesion between cells whose parameters of adhesion depend only on the morphogenetic fields each cell belongs to. Therefore, one may use the output of this algorithm as input for the model.

Chapter 6

Estimating forces from trajectories

In this chapter we will explore some *deterministic* aspects of embryogenesis. A system is deterministic when complete knowledge of the state of the system fully determines its future states. Here, we study this component of the *movement* of cells.

We use one of the simplest and oldest forms of describing movement, Newtonian mechanics, as a reference (Goldstein, Jr., & Safko, 2001). Newtonian mechanics puts *forces* at the origin of movement, relating them to a change of velocity:

> The sum of the forces applied to a body equals its mass times its acceleration.

Furthermore, when bodies are interacting, the applied forces are not arbitrary:

> When one body exerts a force on a second body, the second body simultaneously exerts a force equal in magnitude and opposite in direction on the first body.

This constraint is fundamental, as it allows us to integrate the local information of cells into the whole embryo, by creating an interdependence between them.

Our goal in this chapter is not to create a model of forces between cells, but rather to *infer* possible forces from the cells movement. More precisely, we aim at finding the set of forces between cells that reproduce as well as possible the cells' trajectories. The problem can formulated as:

> Given the constraint of Newton's laws, what are the forces between neighboring cells that match as well as possible the accelerations calculated from the tracking data?

The mechanical interactions between cells happens through the contact of their membranes. Currently, we have no means to extract precisely the cell membranes from the microscope images, which is due to the resolution of the images and the nature of the membranes of the cells of the zebrafish.

Since there is no obvious way to a give default set of contacts for every cell, we use instead the Delaunay tesselation (Delaunay, 1934) of the center of cells. This method provides contacts that respect the proximity of cells. However, this algorithm is adapted to *convex* sets of points, which is not the case for the zebrafish embryo. A consequence of this is the production of long-distance contacts between cells. We overcome this problem by removing contacts that are farther away than a certain threshold, compatible with the typical distances between cells.

The reader probably noted that we did not mention masses. Once more, this is related to the absence of membranes, which would offer a way to estimate the cells' masses through their volumes. In this particular situation we prefer to not use indirect methods like dual triangulations (Bott & Tu, 1982). This is due to the fact that these methods tend to give unstable results and depend very strongly on the set of contacts being used. For this reason, we use the simplifying assumption that every cell has the same mass. It is important to stress that this is a limitation related to the data, not to the method.

Finally, we do not expect to be able to find a perfect matching between forces and accelerations. The first reason is that forces can be non-Newtonian in nature, which is the case of dissipative forces and friction. The second reason is that we ignore completely non-cellular elements, including the environment and the elements that are beyond the boundary of the field of view. We will talk more about these limitations in the discussion.

Controlling time and space scales

We aim at having results that are visually interpretable, since this is essential for the interplay between physicists and biologists. Therefore, we must be able to control the characteristic scales of forces in both time and space in order to obtain results that are smooth in these dimensions. Since noisy inputs lead to noisy outputs in general, we present the methods used to smooth out the irregularities of accelerations and contacts.

Accelerations are notably difficult to calculate with good quality. Even small fluctuations on a smooth curve are greatly amplified by the process of differentiation. For this reason we use the robust differentiators described in **Section 4.2**, that are able to provide better results through the use of a larger number of time steps. Afterwards, the obtained accelerations are homogenized in space, in order to obtain more meaningful spatial patterns.

The Delaunay triangulation is very unstable with relation to changes in cell positions. At two close time steps one can have very different triangulations, which gives an extra factor of discontinuity for the estimation of forces. For this reason we homogenize the cell contacts as described in **Section 4.1**. This procedure provides a set of contacts coming from an interval of time around the current time step, with each contact having a *connexity* factor. This connexity

factor represents the proportion of time steps in the time interval where the contact is present. Therefore, a high connexity factor represents a contact that is persistent in time, while a low one represents a contact that is present only eventually.

Since the quantity of contacts between cells is one of the largest contributors for the running time of the algorithm, we remove those that have small connexity. This allows the reduction of the computational requirements to the calculation, while keeping the most important contributors to the interaction between cells.

Visualization and stress tensors

Forces between pairs can be visualized as lines between their positions in space, whose color corresponds to the intensity of the force. While this may seem natural, our experience shows that the resulting images are typically very difficult to interpret, due to the large number of cells. Furthermore, it is difficult to manipulate measurements defined over contacts, as they appear and disappear more often than cells, in a less structured way.

In order to overcome this problem, we introduce the concept of a *stress tensor* (Landau, Lifshitz, Kosevich, & Pitaevski, 1986). Stress tensors are the analogous of forces between pairs of cells in the context of continuous mechanics. Instead of providing directly the forces between the neighbors, it shows how forces are applied at *every* direction around the point it is calculated. By integrating the application of this tensor to a surface we obtain the total force applied on it.

For a given discretization of space, we can recover forces between pairs of cells from the application of the tensor over the surfaces of contact. It is worth noting that the cells themselvels are a natural discretization of space. By the use of this process, we lose the continuous details of the stress tensor.

Stress tensors can be recovered from discrete forces between pairs of cells using the method of *coarse graining* (Goldhirsch & Goldenberg, 2002). This method uses a convolution in order to "spread" the values defined at the discrete points into space. This generates continuous fields, whose evolution equation can be derived from intercellular forces. However, by using this procedure, we lose the precise information of the forces between each pair of cells.

The representation using stress tensors has three advantages:

- It is simple to homogenize temporally and spatially, giving greater control of scales;
- Its visualization is less noisy and more intuitive. We can represent its symmetric part as an ellipsoid, whose main axes are the principal directions of stress. This leads to visual alignments that are easier to grasp;
- Its representation is independent of the choice of cell contacts used for the calculation of forces. That means that it is a possible intermediate

representation for the comparison of calculations using different sets of contacts.

Taking that into account, after the calculation of the stress tensor, it is homogenized in time and space.

6.1 Formulation

The equation connecting forces to accelerations, setting all masses to 1, can be written as:

$$\sum_j F_{ij} = a_i$$

where F_{ij} is the force applied by cell j over cell i.

We aim to find F_{ij} that match as well as possible the calculated accelerations. The solution for this problem is well defined in the language of linear algebra. Here, we explain formally the intermediate steps that are necessary for the appropriate estimation of forces.

Connexities

In order to obtain less noisy results we homogenize the cell contacts in time (**Section 4.1**). This homogenization attributes to each contact a *connexity* value, which is a real number between 0 and 1. By denoting the connexity between cells i and j by c_{ij}, we introduce these connexities in the equation for forces by:

$$\sum_j c_{ij} F_{ij} = a_i$$

which means that the effective force between two cells that is to be calculated is $c_{ij}F_{ij}$. This corresponds to the attribution of greater weights to the forces between contacts that are more consistent in time.

Linearization

We proceed to the transformation of the previous equation into a matrix formula. For that, we take into account the third of Newton's laws:

$$F_{ij} = -F_{ij}$$

and keep as variables only half of the forces. Let F be the vector of F_{ij} for $i > j$ and A the vector of accelerations. We define the *contact matrix* $N : \mathcal{C} \times \mathcal{C} \overset{\mathbb{R}}{\mapsto} \mathcal{C}$ by:

$$N_{i,jk} = \begin{cases} c_{jk} & \text{for } i = j \\ -c_{jk} & \text{for } i = k \\ 0 & \text{otherwise} \end{cases}$$

This definition makes the equation equivalent to $NF = A$.

In the implementation, we must order pairs of indexes into a contiguous set of numbers. We use an encoding similar to Cantor's pairing function applied to the lower diagonal:

$$(i, j) \to \frac{i(i-1)}{2} + j$$

For the inverse, we define $i = \left\lfloor \sqrt{\frac{1+8n}{2}} \right\rfloor$ in:

$$n \to \left(i, n - \frac{i(i-1)}{2} \right)$$

Pseudosolution

The forces we are looking for should reproduce as close as possible the accelerations, while staying as small as possible. This description matches well the solution given by the *Moore-Penrose pseudoinverse* (Penrose & Todd, 1955). For every matrix M, its Moore-Penrose pseudoinverse M^+ always exists, being uniquely determined by the following properties:

- $MM^+M = M$
- $M^+MM^+ = M^+$
- $(MM^+)^* = MM^+$
- $(M^+M)^* = M^+M$

where M^* is the transpose of M. For some special cases it is simple to calculate:

- If M is invertible then $M^+ = M^{-1}$
- If M^*M is invertible then $M^+ = (M^*M)^{-1}M^*$
- If MM^* is invertible then $M^+ = M^*(MM^*)^{-1}$

From the properties:

- MM^+ is the orthogonal projection onto the range of M
- M^+M is the orthogonal projection onto the range of M^*

we can see this solution as a least-squares approximation for the system of equations, with the projection happening both in the domain and the codomain. The solution given by the pseudoinverse is the one whose image is the closest to the

vector provided, while being the one which has the smallest quadratic norm. This means that the solution we are looking for can be written as

$$F = N^+ A$$

which can be transformed into forces by multiplication, contact by contact, with connexities.

The explicit calculation of the pseudoinverse is not necessary, since the result can be obtained using explicit least squares minimization. However, this formulation is simpler to manipulate formally. As an example, it is useful to reduce the dimensions of the problem in order to lower the computational complexity of the problem. Namely, the pseudo inverse has the following useful property:

$$N^+ = N^*(NN^*)^+$$

which reduces the calculation of the pseudoinverse of a matrix of size $|\mathcal{C}|^2 \times |\mathcal{C}|$ to one of size $|\mathcal{C}| \times |\mathcal{C}|$.

6.2 Implementation

Here we describe some details of the algorithms implemented in the package `lineageflow-forces`.

Least Squares Algorithm

The forces are calculated using a least squares approximation of the resulting accelerations. This approximation is calculated using the algorithm `SparseQR` from the `Eigen` library (Guennebaud, Jacob, & others, 2010). This choice come from the test of multiple algorithms and the comparison with the explicit pseudosolution in test cases with a small number of cells. Furthermore, the documentation of this algorithm states explicitly that it is optimized for least square problems.

Complexity

The main source of computational cost in this problem is the size of the matrices involved. In particular, the bottleneck of this algorithm is the size in RAM of the contact matrix. For this reason, much care has to be taken when choosing which neighbors to use. In particular, the most demanding step is the calculation of the least squares solution, which has large memory requirements. The reduction of the pseudoinverse dimension using the properties in the previous section is useful for reducing these requirements.

The main quantity that controls the running time and RAM usage is the number of cell contacts. Filtering these contacts by connexity allows the reduction of an almost arbitrary quantity of contacts while retaining as much stability as possible. By filtering the contacts whose connexities are the smallest, we guarantee to keep the most relevant terms in the least squares equation.

Stress Tensor

By defining σ to be the three dimensional stress tensor, the explicit formula for its elements $\sigma_{\alpha\beta}$ using forces between pairs of cells is (Goldhirsch & Goldenberg, 2002):

$$\begin{aligned}\sigma_{\alpha\beta}(r_k) = &-\sum_{i>j} F_{ij\alpha} r_{ij\beta} \int_0^1 N_\sigma[r_{ki} + s r_{ij}] ds \\ &- \sum_i \Delta v_{i\alpha}(r_k) \Delta v_{i\beta}(r_k) N_\sigma(r_{ki})\end{aligned}$$

where:

- r_k is the position of cell k;
- $r_{ij} = r_i - r_j$ is the difference between the positions of cells i and j
- $\Delta v_i(r) = v_i - V(r)$ is the difference between the calculated velocity of a cell and its Gaussian spatial homogenization, as defined in **Section 4.1**.

and N_σ is the *normal distribution* of a three-dimensional vector v with standard deviation σ:

$$N_\sigma(v) = \frac{1}{\sigma^3(2\pi)^{3/2}} e^{-\frac{|v|^2}{\sigma^2}}$$

Since the integral term in this equation is very costly to calculate for every single contact between cells, we stablish bounds for this term in order to find the most relevant ones. The integral term can be approximated as:

$$\begin{aligned}\int_0^1 N_\sigma(a+tb)dt &\approx \frac{\exp\left(-\frac{|a|^2}{\sigma^2} + \frac{(a\cdot b)^2}{\sigma^2|b|^2}\right)}{4\sqrt{2}\pi\sigma^2|b|}\left[\text{erf}\left(\frac{|b|}{\sigma} + \frac{a\cdot b}{\sigma|b|}\right) - \text{erf}\left(\frac{a\cdot b}{\sigma|b|}\right)\right] \\ &\leq \frac{1}{2\sqrt{2}\pi\sigma^2|b|}\end{aligned}$$

So that by putting an upper bound in $|b|$ we bound the smallest value this term can have. By using the mean value theorem, we know that there is some $0 < s < 1$ for which:

$$\begin{aligned}\int_0^1 N_\sigma(a+tb)dt &= \frac{\exp\left(-\frac{|a|^2}{\sigma^2} + \frac{(a\cdot b)^2}{\sigma^2|b|^2}\right)}{\sigma^3(2\pi)^{3/2}} \exp\left[-\left(\frac{a\cdot b}{\sigma|b|} + s\frac{|b|}{\sigma}\right)^2\right] \\ &= \frac{1}{\sigma^3(2\pi)^{3/2}} \exp\left(-\frac{|a|^2 + 2s(a\cdot b) + s^2|b|^2}{\sigma^2}\right) \\ &\leq \frac{1}{\sigma^3(2\pi)^{3/2}} \exp\left(-\frac{(|a| - s|b|)^2}{\sigma^2}\right)\end{aligned}$$

This implies that with an upper bound in $|b|$, we may establish an upper bound in $|a|$, which gives the smallest value this term can have. These two approximations together give the largest bounds for terms which have both low $|a|$ and $|b|$.

We can write the equation in matrix form as:

$$\Sigma = -\Psi(f \otimes r) - \Phi(\Delta v \otimes \Delta v)$$

where:

- Σ_k is the stress tensor at the position of cell k;
- $\Psi_{ij,k} = \int_0^1 N_\sigma[r_{ki} + sr_{ij}]ds$;
- $\Phi_{ik} = N_\sigma(r_{ik})$.

Given the previous remarks, $\Psi_{ij,k}$ has the largest bounds for values that have both small $|r_{ki}|$ and $|r_{ij}|$. It means that a good approximation is to consider only the terms for pairs of indices that are neighbors.

Types

We introduce here the main types for the estimation of forces at a given time step. Introducing the synonyms:

```
type DArray i j a = Array (Dep i j) a
type DS2 i j = S2 (Dep i j)
```

we have the following types for the set of contacts, which is identified by an identity map:

```
DArray Time (DS2 Time Cell) (DS2 Time Cell)
```

accelerations in the temporal-view:

```
DArray Time Vector
```

forces as vectors for every contact:

```
DArray Time (DS2 Time Cell) Vector
```

and stress tensors in the temporal-view:

```
DArray Time Tensor
```

Resources

As a reference, the requirements for the results calculated in the following section are approximately:

- RAM: 8GB;
- Time: 14h on a Intel(R) Core(TM) i5-2450M CPU @ 2.50GHz.

6.3 Results

We present here the results obtained by the application of the force estimation method to the zebrafish embryo 141108aF. This data set has around:

- 340 time steps;
- between 5000 and 8000 cells per time step;
- 100000 cells globally.

The voxel size is 1.38μm and the time steps are spaced by 2min33sec. Space units are represented in voxel size and time in time steps. This is due to the fact that for the conversion to metric units we would need precise values for masses, which we do not have at the moment.

Scalars, vector and tensor norms are represented with a color code using a gradient on the sequence blue, green, yellow and red, which is represented in each image. Scalars over contacts are represented as lines between pairs of cells, colored by the value of the scalar. Vector are represented as lines outcoming from the position of the cell, in the direction of the vector and colored by its norm. The symmetric part of tensors is represented as ellipsoids whose eigenvectors are used as axis, colored by the norm of the matrix. The asymmetric part of tensors is represented as the vector that corresponds to the antisymmetric matrix.

Accelerations

We calculate accelerations using:

- a Lanczos derivative of order 2, using a filter size of 20 time steps;
- a Gaussian spatial homogenization of size 20 voxels.

both procedures are described in **Chapter 4**. These values have been chosen by trial and error, in order to obtain time and space scales that are appropriate for visual interpretation. The results are shown in Fig. 6.1.

Contacts and Connexities

The cell contacts are calculated using a Delaunay triangulation on the set of cell centers. From this we obtain around 50000 contacts per time step.

We proceed by removing all the cell contacts whose lengths are larger than 15 voxels. This results in around 40000 contacts per time step.

We proceed to the homogenization of the contacts, using a flat homogenization of scalars, with a filter size of 10 time steps. This procedure gives a measure of temporal connexity for each contact and increases the number of contacts to around 100000 per time step. We recall that the increase of the quantity of

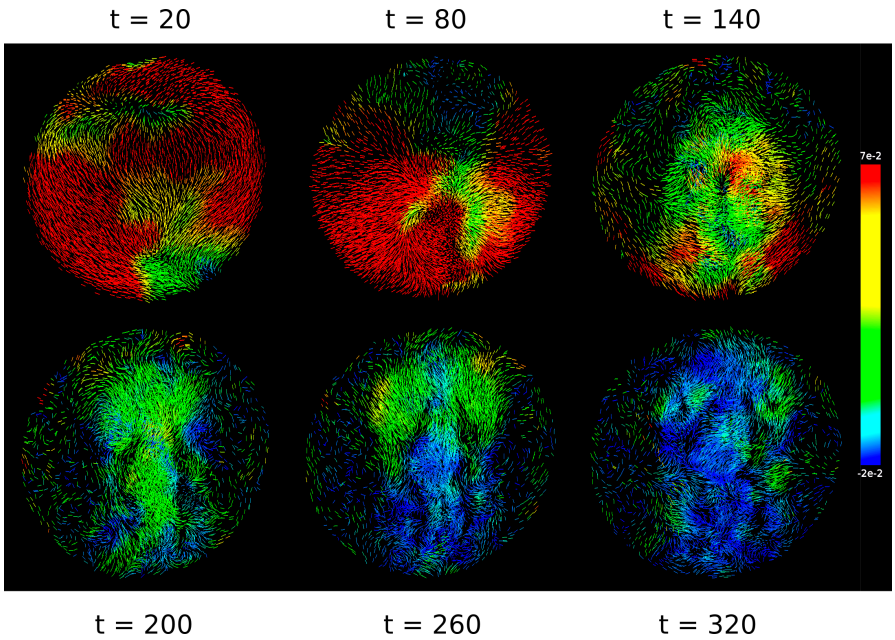

Figure 6.1: **Accelerations calculated with Lanczos derivative.** The filter size and homogenization radius are chosen in order to obtain adequate scales for visual interpretation.

contacts is due to the collection of all contacts in a window of time around each time step.

Finally, we remove all the contacts whose temporal connexity is smaller than 0.6. This reduces the quantity of contacts to around 20000 per time step. The connexity threshold has been chosen in order to make possible the estimation of forces given the computational constraints. These connexities are shown in Fig. 6.2. As it can be seen, there are no clear patterns and the result is noisy. This is not surprising given the nature of the cell center identification and the inherent instability of the Delaunay triangulation.

Forces

We calculate the forces using the method described previously, using the acceleration and connexities exposed in the previous sections. The representation of the norms of forces are shown in Fig. 6.3. While the results seem more ordered than the connexities, no clear spatial patterns are visible. However, we are able to see that the global intensity of forces is variable in time.

We explore further the change of the global intensity of forces through the distribution of the their norms in Fig. 6.4. As it can be seen, the distribution is not stationary and change very significantly also in shape with time. The global intensity of forces decreases in the last time steps, which is compatible with the stabilization of the positions of cells.

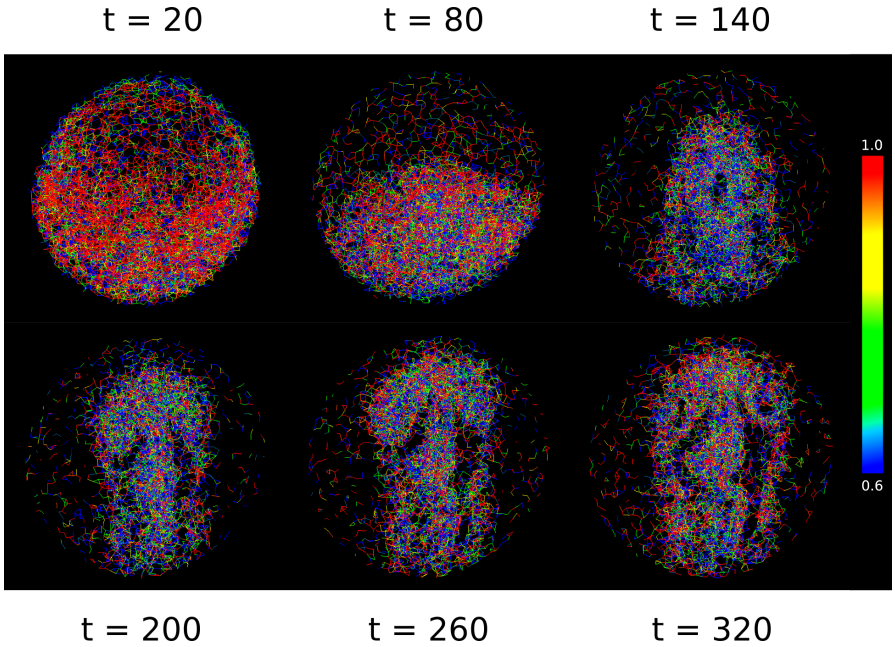

Figure 6.2: **Connexities of the filtered set of contacts.** The filtered set of contacts contain only those whose connexity is larger than 60%. No clear patterns are found in these results.

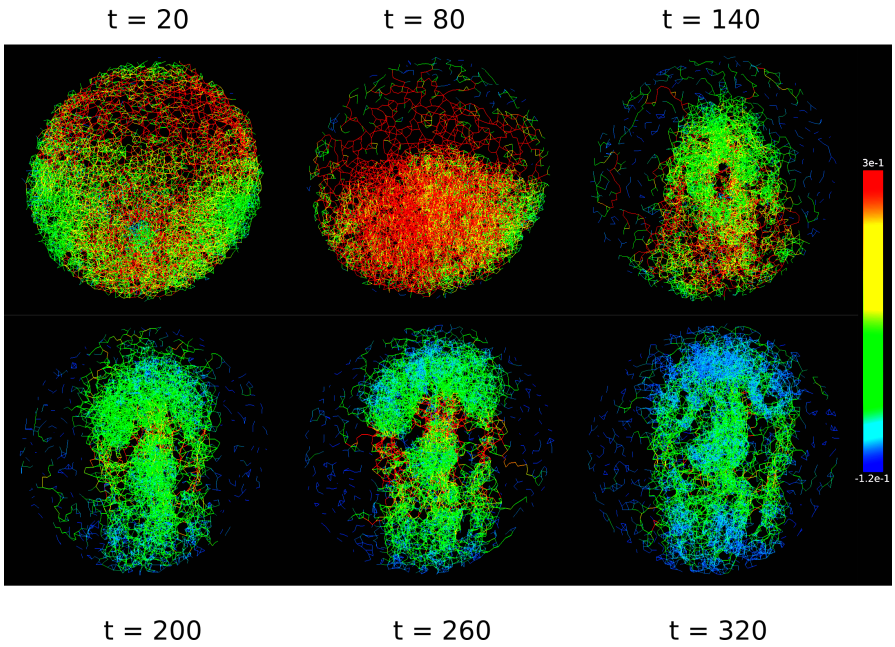

Figure 6.3: **Norm of forces calculated using the proposed method.** While some patterns are identifiable, they are short lived and hard to interpret. However, one can see that the global intensity of forces change in time.

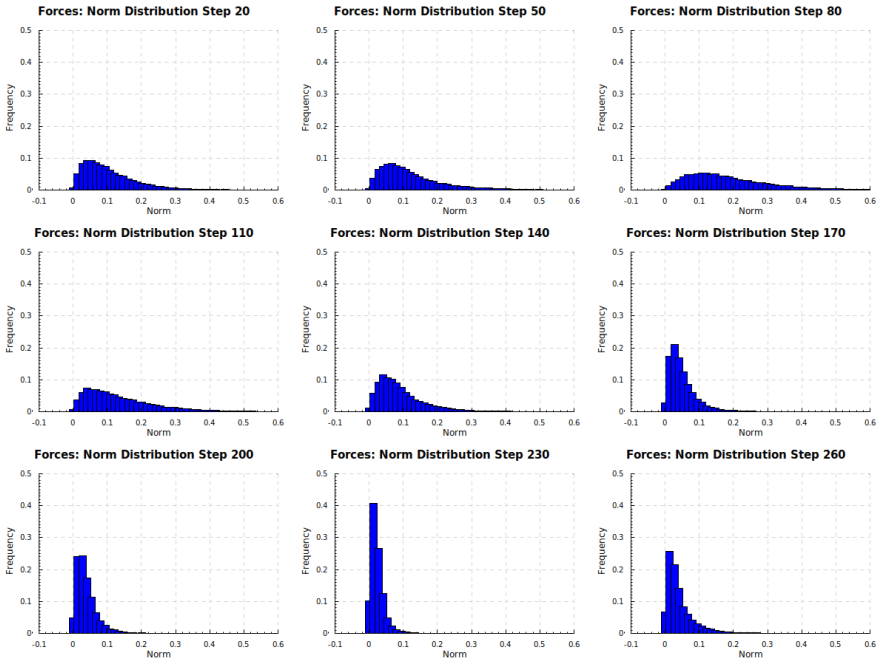

Figure 6.4: **Distribution of the norm of estimated forces.** The distributions change very significantly in time, both in intensity and shape. The global distribution is more concentrated in the lower intensities in the last time steps.

Stress Tensor

The stress tensors are calculated using the method cited previously (Goldhirsch & Goldenberg, 2002). The radius used for the spatial coarse graining is 20 voxels. Furthermore, a flat temporal homogenization with size 10 and a Gaussian spatial homogenization of size 20 is applied. These parameters have been chosen by visual inspection.

The symmetric part of these stress tensors is shown in Fig. 6.5. Due to the homogenization, we are able to see color patterns and tensor alignments that are interpretable. We recall that red represents regions with high stress, while blue represent low intensities. We are able to identify:

- The large red region with vertically aligned tensors in the first time steps, denoting the upward movement of cells towards the head;
- The red region of horizontally aligned cells in the following time steps, converging towards the location of the spine;
- The two red regions in the later time steps, coinciding with the formation of the two eyes;
- The stabilization of stresses in the last time steps, coinciding with the stabilization of shapes.

The distribution of pressures derived from the trace of the stress tensor is presented in Fig. 6.6. It is interesting to note that the time evolution shows multi-

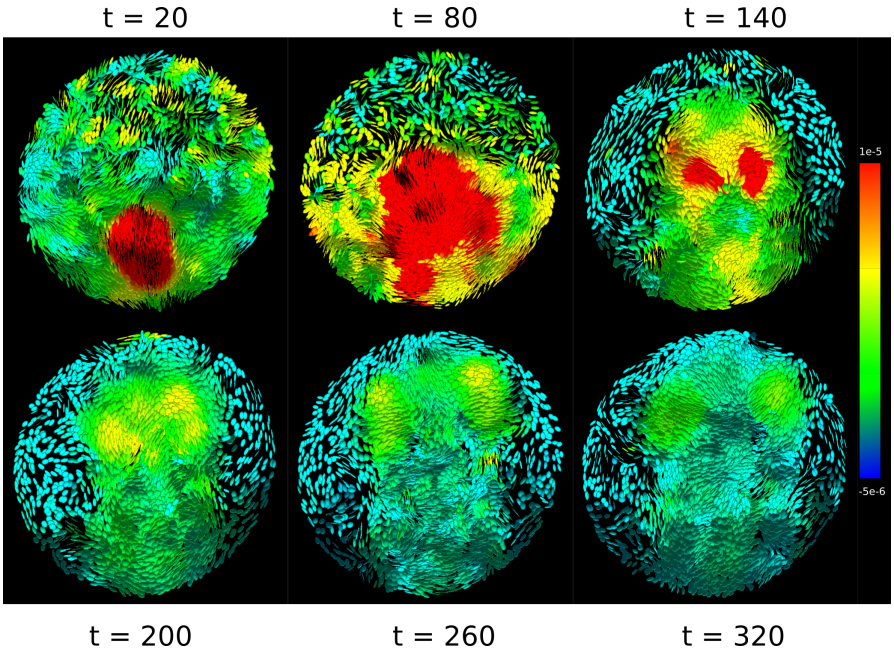

Figure 6.5: **Homogenized stress tensors.** Here we see the emergence of patterns both in intensities and the alignment of tensors. We are able to identify the movement towards the head, the formation of the spine and the formation of the eyes.

ple simultaneous peaks in the distribution. This can be interpreted as different regions which have stresses which are homogeneous locally but different between these regions.

We decompose the stress tensors in two parts, a symmetric and an antisymmetric one. The symmetric part corresponds to the one that preserves the angular momentum, while the antisymmetric part transforms it. The asymmetry of the stress tensor is often linked to anomalies in the material and diverges from the classical formulation of continuum mechanics (Willam & Iordache, 2001). We present the vectors corresponding to these antisymmetric tensors in Fig. 6.7. The most notable feature of this component is that it has a norm that is an order of magnitude smaller than the symmetric part, which means that the stress tensor are almost symmetric.

We present the statistical distribution of these quantities now. We start with the norm of the symmetric part of the stress tensor in Fig. 6.8. It is interesting to note that these norms have a very similar distribution to that of the pressure, with similar values and peaks. In opposition, the antisymmetric part of the tensor (Fig. 6.9) has values that are an order of magnitude smaller, and with a different profile, with single peaks and which is more uniform in time than pressures.

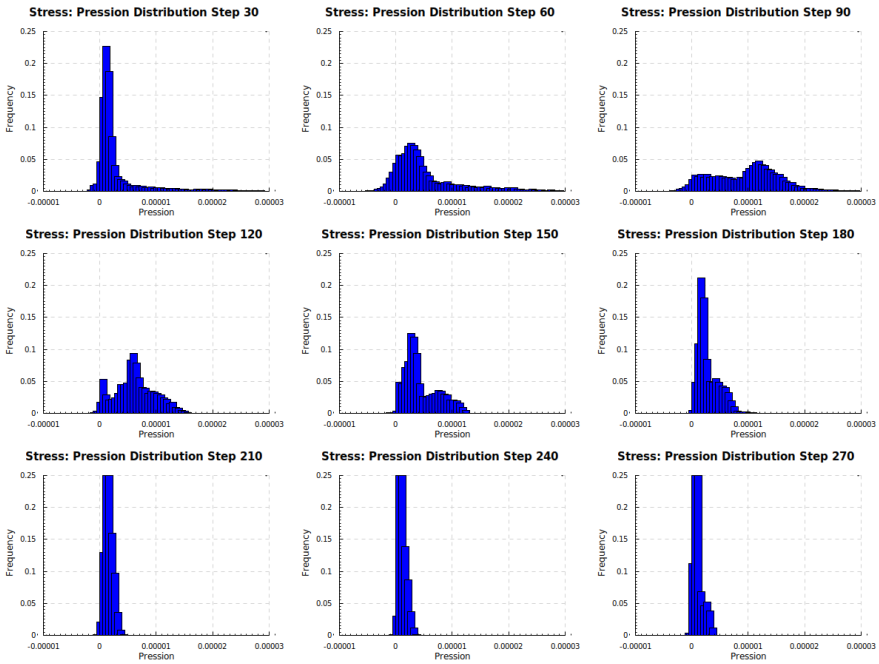

Figure 6.6: **Pressures derived from stress tensors.** The distribution of pressures shows simultaneous peaks in different locations at some time steps. This shows the local nature of the interaction between cells.

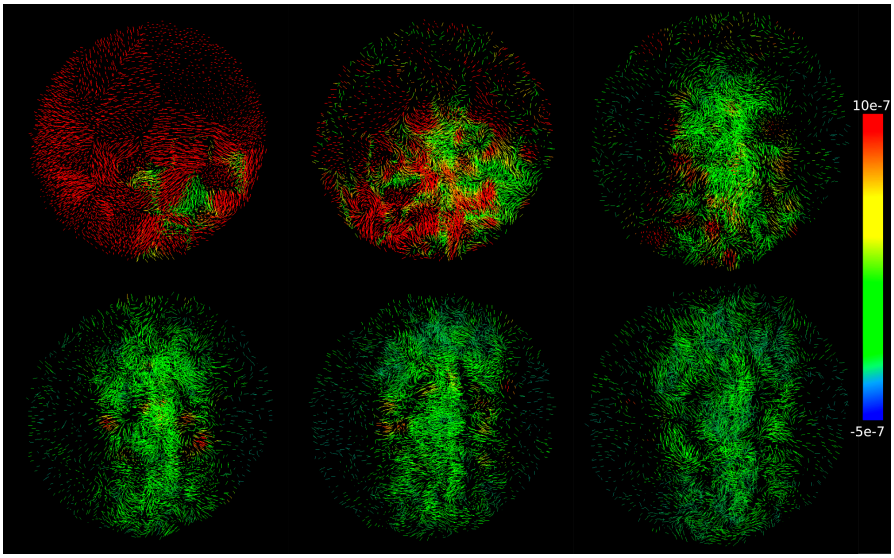

Figure 6.7: **Antisymmetric part of homogenized stress tensors.** This component is responsible for the transformation of angular momentum. As it can be seen, it is less intense than the symmetric part and no spatial patterns can be identified.

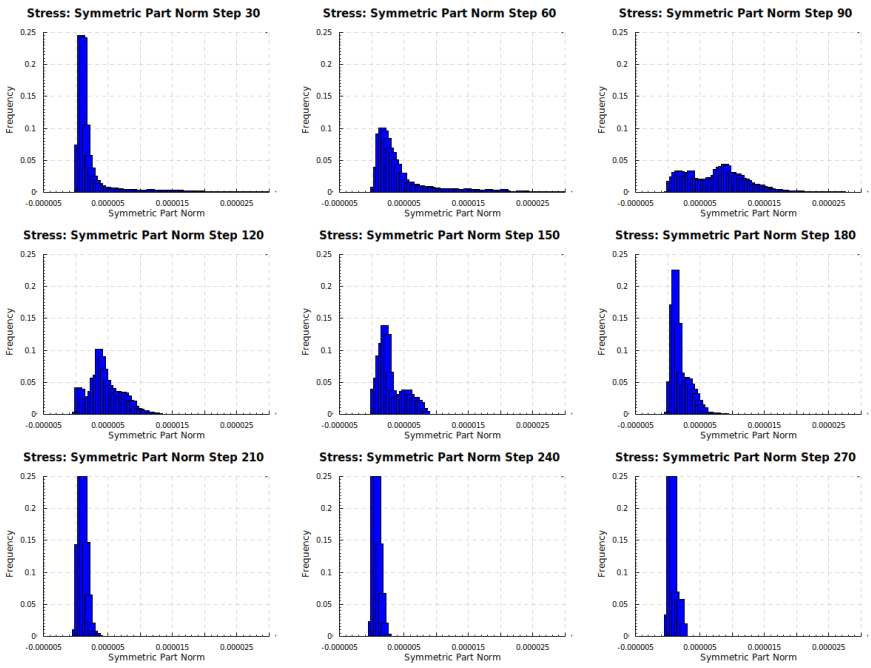

Figure 6.8: **Norm of the symmetric part of the stress tensor.** The distribution of this quantity is very similar to the one of pressures, both in intensity and shape.

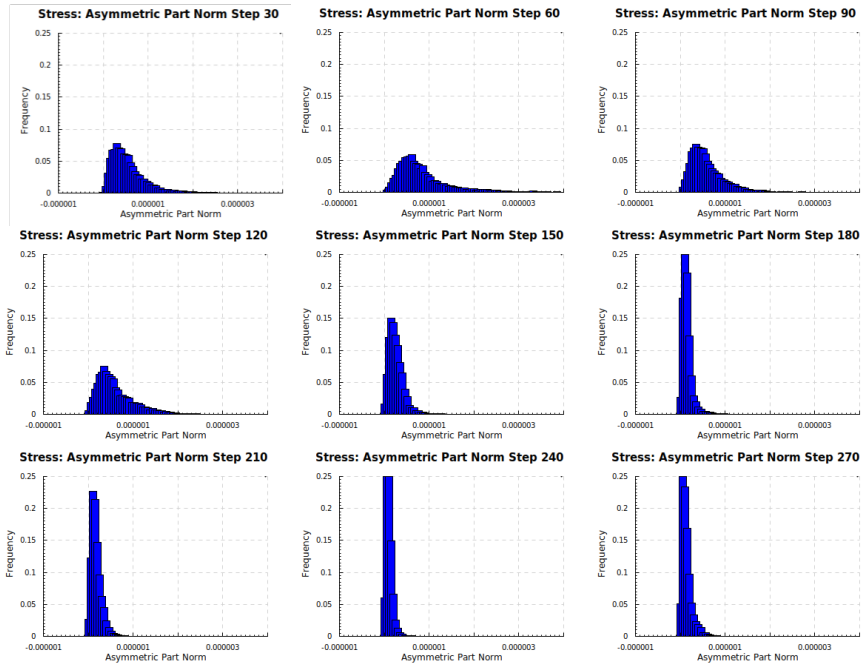

Figure 6.9: **Norm of the antisymmetric part of the stress tensor.** The values of this component is an order of magnitude smaller than the symmetric part. It differs also in the profile of the distribution, having a single peak at each time step.

6.4 Discussion

We start by noting that the movement of cells is not the only relevant component for the morphogenesis of the embryo. In the presence of cell division and the change in shape of cells, the importance of forces depends very strongly on the species being studied. As an example, while the zebrafish development is very rich in cell displacement, the development of the sea urchin is dominated by cellular divisions.

That being said, forces provide a good conceptual bridge between physicists and biologists. Since it is a common concept that is easy to understand and interpret, the mutual feedback about the models is simpler. This was shown in the results, where the alignment of tensors and color patterns were simple to interpret with some knowledge of the morphology of the zebrafish.

These alignments and patterns are remarkable, as they show the coherence of the calculated tensors with the morphogenetic processes happening during the development of the embryo. This is especially important in a method that is composed of such a large pipeline of transformations, each one capable of compromising the results. We attribute this success to the judicious homogenization of measurements.

Those tensors shed light on processes that are not completely obvious. As an example, the formation of the eyes is visible as large region, with strong intensity and horizontally align tensors. While the visual distinction of tissues is perhaps not surprising, the intensity and alignment of these forces are not evident and clarify the nature of a morphologically relevant process.

This leads to the question of the correctness of these forces. As discussed before, the model we use here is very restrictive and ignores many possible contributions to the dynamics of cells. However, it is difficult to access the precision of the method, since it is impractical to measure forces in the embryo globally. On the other hand, many experimental methods for the local measurement of forces exist (Polacheck & Chen, 2016). Therefore, one can theoretically validate these stresses using the locally measured forces as reference points. However, in order to do so one confronts the problem of combining the microscopy methods being used with these particular techniques, which can be a problem.

Directions of research

A first possible direction of research is the *prediction* of trajectories through stresses. Using the estimated stresses, one may simulate the movement of cells and compare the results with the actual trajectories. Also, this is another test that can be used for the verification of the correctness of forces.

By assuming some degree of correctness of these stresses, one may study them

directly. Since the algorithm produces large quantities of data, this offers good statistics. This can be done through the study of its material properties, like elasticity, or the creation of models, which can then be fitted to the data.

This model goes in a different direction of many others, which consider forces to be purely frictional. One can verify the compatibility of both assumptions through the use of the calculated stresses. From the application of Cauchy's equation (Landau et al., 1986) one can derive the local forces, which can be compared directly to the velocities calculated from trajectories.

An important characteristic of this method is its extensibility by more precise models. Finer-grained assumptions about the nature of the cell interactions can lead to parametric models that are more meaningful and whose results are more informative. This parametric model would change the shape of the terms in the least squares equation, but the general procedure is still valid.

One of the shortcomings of this method is the lack of intuition on its limitations. This can be overcome with the study of its application to systems whose dynamics is explicit. In particular, the application to systems where there is dissipation of energy and friction can give important insights on this matter. This method can also be applied to better understand the effect of the partiality of the data set on the results.

The knowledge of the real cell contacts can improve the results. This would remove one of the arbitrary choices of the method, where many parameters have to be given, namely the Delaunay tesselations, averagings and thresholds. Also, that would give the means of comparing both results through the stress tensors in order to evaluate the dependency of the results on the choice of contacts.

Finally, one can use these contacts for the recalculation of forces. Since we can see well defined patterns on stresses, due to the homogenization procedures, we ought to see them in forces too. This can be done through the application of the stress tensor to the interfaces between cells, which can be calculated as dual meshes from the cell contacts. We expect these calculated forces to be as well behaved as stresses, and allow the patterns to be seen in view of the direct interaction between cells.

Chapter 7

Cell trajectory deviations

In opposition to determinism there is stochasticity. A process is *stochastic* when the knowledge of the state of the system does not determine its future, this behavior being modeled by a *random* process. This opposition comes also in form of complementarity. The idea is that a process can be decomposed into two components: one deterministic and one stochastic. In the context of this decomposition, the deterministic component represents the main tendance and the stochastic one fluctuations around this tendance.

In this chapter, we study the stochastic component of the movement of cells. This is done through the study of the *relative position* of pairs of cell, that is, their difference in position.

It is known (Cadart, Zlotek-Zlotkiewicz, Berre, Piel, & Matthews, 2014) that immediately after mitosis the just divided cells are rounder and less attached to its neighbors than regular cells. We expect to be able to identify this difference in attachment by analysing the statistics of relative positions. For this purpose, we will study three different groups of cells:

- pairs of cells that are within a certain distance to each other, which will be called *neighbors*;
- pairs of cells after division, which will be called *sisters*;
- pairs composed of one cell after division and another cell within a certain distance of the first one, which will be called *divided neighbors*.

The reference distance between neighbors is chosen in order to have a similar size to typical distances between sisters.

We now discuss some conditions under which the study of relative positions can lead to the stochastic component of the cells' movement. For this goal, we decompose the factors that contribute to the movement of cells into:

- *external*: environmental constraints and forces between cells;
- *internal*: autonomous self-directed dynamics;

- *fluctuations*: divergences from the main tendance of external and internal factors.

and make assumptions about these factors:

- External forces and constraints are continuous in space. It means that cells that are close to each experience similar environmental effects. This assumption is reasonable given the coherent movement of morphogenetic fields;
- The self-directed forces that lead to autonomous movement are statistically negligeable. It means that taking away the external factors, we keep only the fluctuations around the main tendances. This assumption is reasonable since the number of cells which are autonomous is much smaller than the global number of cells;
- The additivity of these different factors. It means that the manipulation of the different factors can be done algebraically.

Since two neighboring cells share similar external factors and their autonomy is to be ignored, we expect their relative movement to be determined by fluctuations. Thus, we claim that:

> If two cells are close to each other at a given moment in time, then the dynamics of their relative position at short times is stochastic.

We limit ourselves to claim at short times since the closeness between cells is eventually broken due to displacement. However, we also study the long time evolution of the relative position, as it also offers insights on the dynamics of neighbors.

Measurements to be studied

We divide the study of the relative position of cells in two parts, one focusing on its short time dynamics and another focusing on its long time dynamics.

For the study of short time dynamics we will focus on the statistics of relative positions. We analyse the statistical distribution of relative increments, that is, the changes in relative positions for each of the tree groups of cells.

For the study of long time dynamics we will focus on resuming statistics of the relative positions and increments. Since the increase of distance between cells has the tendance of increasing the difference between their environments, we expect extra drift effects.

We analyse the following measurements:

- *Mean squared relative displacement*, whose behavior gives indications on the nature of the stochastic process and on how cells drift away from each other in time. The interpretation of this measurement is done through the

comparison with *Brownian motion*, whose mean squared displacement is linear in time.

- *Relative increment autocorrelation*, which quantifies the relation between relative increments at different moments in time. For example, positive correlations between two different points in time shows that the displacements tend to go to the same direction, while negative correlations shows that they tend to go to opposite directions.

Both measurements compare different moments in time and are presented in initial and averaged form:

- The *initial* form compares all the different time steps with the initial point in time;
- The *averaged* form averages all the comparisons over intervals of time of the same size. This can lead to better statistics and give indications about the stationarity of the process.

7.1 Formulation

We collect here the definition of statistical terms and measurements that have been studied. All these are adapted for discrete and finite time sets.

Definition of statistical terms

Let V be a $\mathbb{R}$-vector space whose inner product is represented by $\cdot$.

The *average* of a function $X : \mathcal{C} \to V$, denoted by $\mathbb{E}_{\mathcal{C}}[X] : V$ is defined by:

$$\mathbb{E}_{\mathcal{C}}[X] = \frac{1}{|\mathcal{C}|} \sum_{c:\mathcal{C}} X(c)$$

The *covariance* of two functions $X, Y : \mathcal{C} \to V$, denoted by $\mathbb{C}_{\mathcal{C}}[X, Y] : \mathbb{R}$ is defined by:

$$\mathbb{C}_{\mathcal{C}}[X, Y] = \mathbb{E}_{\mathcal{C}}[(X - \mathbb{E}_{\mathcal{C}}[X]) \cdot (Y - \mathbb{E}_{\mathcal{C}}[Y])]$$

The *autocovariance* of a function $X : \sum_{t:\mathcal{T}} \mathcal{C} \to V$, denoted by $\mathbb{A}_X : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$ is defined by:

$$\mathbb{A}_X(s, t) = \mathbb{C}_{C_{t+s} \cap C_t}[X(t + s), X(t)]$$

Relative Position

For a given cell i, we denote by $X_i(t) \in \mathbb{R}^3$ its position at the moment $t \in \mathcal{T}$. We define the *relative position* of two cells i and j by

$$X_{ij}(t) = X_i(t) - X_j(t)$$

This quantity is well defined for all t in the time interval $I_{ij} = I_i \cap I_j$.

Relative Displacement

We recall that times with an apostrophe are relative times (**Section 2.2**), which measure the time elapsed from the moment of appearance of the cell in the data set. We define the *relative displacement* by

$$D_{ij}(s', t') = X_{ij}(t' + s') - X_{ij}(t')$$

This quantity is well defined for pairs of cells in the set $\mathcal{C}_{s't'} = \mathcal{C}_{t'+s'} \cap \mathcal{C}_{t'}$. The initial relative displacement is

$$D^0_{ij}(s') = D_{ij}(s', 0')$$

and the averaged relative displacement is

$$\widetilde{D}_{ij}(s') = \mathbb{E}_{I'_{ij}}\left[D_{ij}(s')\right]$$

Relative Increment

We define the *relative increment* or *relative velocity* between two cells i and j to be:

$$V_{ij}(t') = D_{ij}(t'+1) - D_{ij}(t')$$

The initial autocovariance of these increments is

$$\gamma^0_{ij}(s') = \mathbb{A}_{V_{ij}}(s', 0')$$

and the averaged autocovariance of these increments is

$$\widetilde{\gamma}_{ij}(s') = \mathbb{E}_{I'_{ij}}\left[\mathbb{A}_{V_{ij}}(s')\right]$$

Mean squared relative displacement

The initial mean squared relative displacement is defined by

$$\delta^0(s') = \mathbb{E}_{C_{s'}}\left[|D^0_{ij}(s')|^2\right]$$

and the time averaged mean squared relative displacement is

$$\widetilde{\delta}(s') = \mathbb{E}_{C_{s'}}\left[|\widetilde{D}_{ij}(s')|^2\right]$$

Relative increment autocorrelation

The normalized initial relative increment autocorrelation is defined by:

$$\gamma^0(s') = \frac{\mathbb{E}_{C_{s'}}\left[\gamma^0_{ij}(s')\right]}{\mathbb{E}_{C_{0'}}\left[\gamma^0_{ij}(0')\right]}$$

and the normalized averaged relative increment autocorrelation is

$$\widetilde{\gamma}(s') = \frac{\mathbb{E}_{C_{s'}}\left[\widetilde{\gamma}_{ij}(s')\right]}{\mathbb{E}_{C_{0'}}\left[\widetilde{\gamma}_{ij}(0')\right]}$$

7.2 Implementation

Measurements

The use of higher-order functions greatly simplify the presentation of measurements. Both the mean square deviation and velocity autocorrelation are instances of the same functions:

```
shiftWith :: (a -> a -> Scalar) -> Array t a -> Array t Scalar
shiftWith0 :: (a -> a -> Scalar) -> Array t a -> Array t Scalar

msd = shiftWith (\v1 v2 -> square (v1 - v2))
auto = shiftWith (\v1 v2 -> v1 !.! v2)
```

where `square` denotes the squared norm and `!.!` denotes the inner product. The two versions of `shiftWith` correspond to different implementations corresponding to the averaged and initial (suffixed by `0`) versions.

Computational complexity

Any cell at any time step is susceptible to be taken as a reference for the calculation of deviations between neighbors. However, we always take these cells at the first time step of their appearance on the data set. This choice maximizes the time steps contributing to statistics and reduces redundant calculations.

The computational complexity of the calculation depends mostly on the number of reference cells and radius of calculation. In practice:

- Every pair of sisters is used since they are few and statistics is more scarce;
- For every cell that just divided, we take all the neighbors in a given radius;
- In the case of neighbors in general, for a given proportion of all cells, we take all the neighbors in a given radius. This restriction of the number of cells reduces the running time. Experimentation shows that the statistics found using all cells or just a proportion of them do not differ substantially.

Types

For each pair of cells, there is a sequence of vectors representing their relative position. This is represented by the type:

```
DSumMap (Cell,Cell) Time Vector
```

It is important to note that since the subset of the domain is not identified in the type system, the type is the same for all the groups of pairs. This representation of relative positions is equivalent to the cellular point of view of the data set.

Since the alignment of trajectories for statistics is given with relation to relative times, the transposed type is:

```
DSumMap (Dep (Cell,Cell) Time) (Cell,Cell) Vector
```

We note that this is different from the alignment with relation to absolute time, which would give:

```
DSumMap Time (Cell,Cell) Vector
```

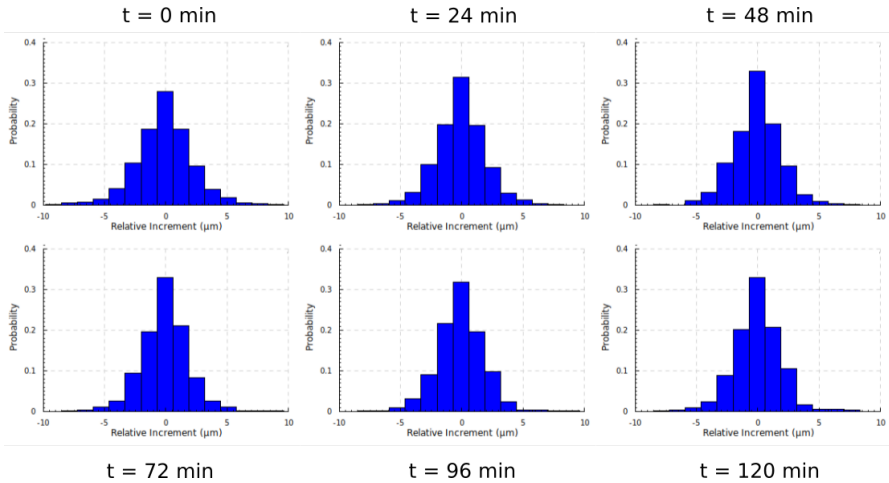

Figure 7.1: **Distribution of relative increments: sisters.** Distribution of the coordinate x of relative increments over time for pairs of sister cells of the embryo 141108aF.

7.3 Results

We present here the results of the calculations applied to the embryo 141108aF. This data set has approximately:

- 340 time steps;
- between 5000 and 8000 cells per time step;
- 100000 cells globally.

Around 10% of the cells in the data set divide. This means that we have around 10000 pairs of sisters cells, which implies around 20000 cells generated after division.

The typical distance between sisters after division is around 10 voxels, or 15 μm. Inside a sphere with a radius of this size, there are around 10 cells.

Distribution of relative increments

We plot the distributions of relative increments for sisters (Fig. 7.1), neighbors of divided cells (Fig. 7.2) and neighbors of cells in general (Fig. 7.3) respectively. This plotting is done using the coordinate x of the displacement vector, but the result is not substantially different for other coordinates.

These distributions are not visually distinct. In order to obtain better criteria for their distinction, we compare the distributions using the Kolmogorov-Smirnov test (Kolmogorov, 1933). This test compares two distributions using a p-value (Marsaglia, Tsang, & Wang, 2003) and returns if they are compatible. This is a binary test, whose answer is always "yes" or "no".

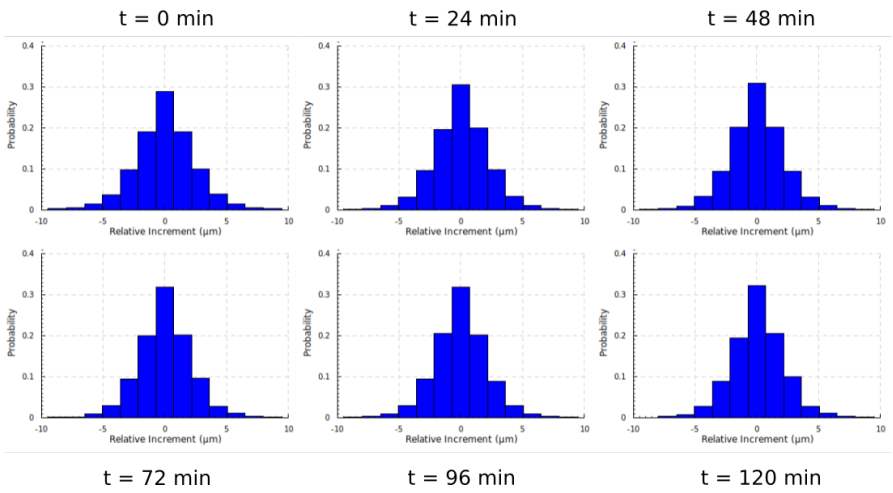

Figure 7.2: **Distribution of relative increments: divided neighbors.** Distribution of the coordinate x of relative increments over time for neighbors of divided cells of the embryo 141108aF.

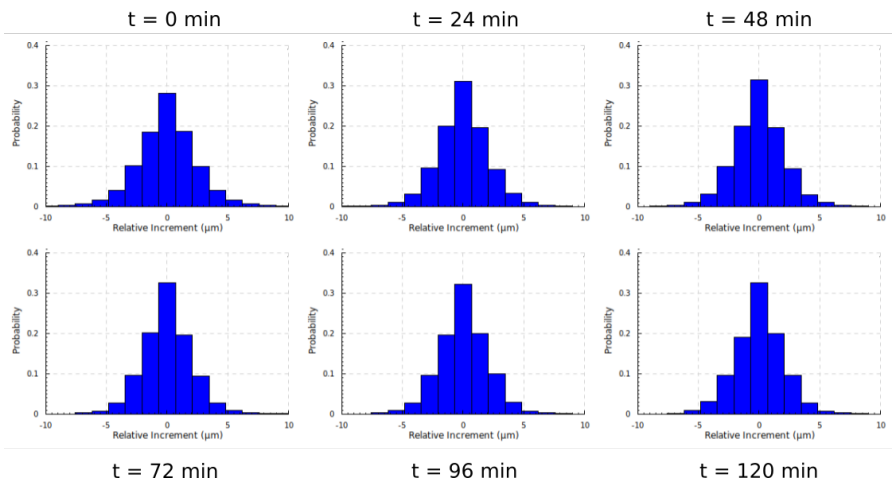

Figure 7.3: **Distribution of relative increments: neighbors.** Distribution of the coordinate x of relative increments over time for neighboring cells of the embryo 141108aF.

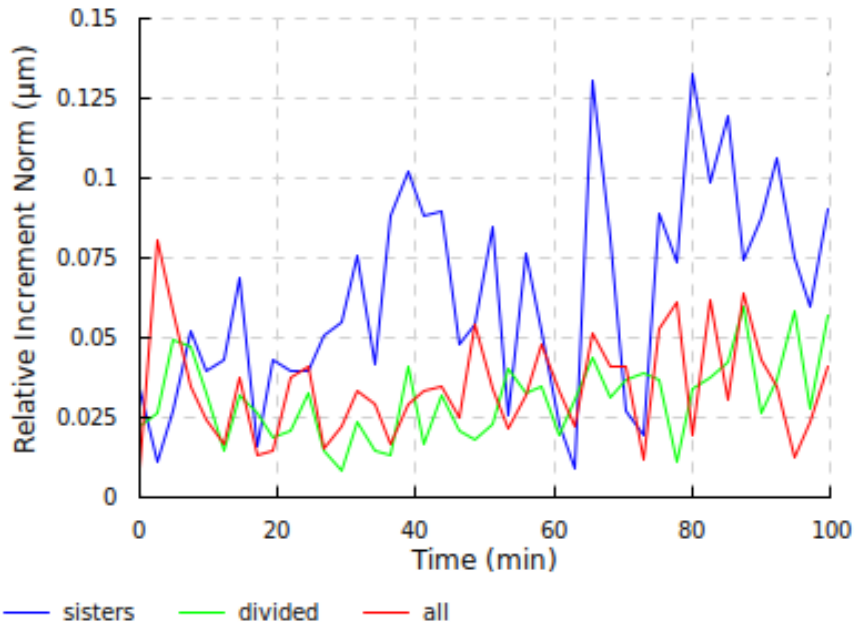

Figure 7.4: **Evolution of the norm of the average of relative increments on the embryo 141108aF.** The evolution is shown for pairs of sisters, neighbors of divided cells and neighbors in general.

We apply this test to each pair of distributions at each time step. For a p-value of 0.1, we can observe that:

- the distributions of sisters cells and neighbors of divided cells are undistinguishable for almost all time steps;
- the distributions of sisters and neighbors of divided cells are both distinguishable from neighbors in general, for times smaller than 35 minutes.

These results corroborate the idea that a divided cell is attached differently to its environment. Intuitively, since the environment is less relevant for a divided cell, it does not matter if we study its deviation relatively to its sister or a regular cell. However, it is important to note that the statistical difference between distributions disappears very fast.

Properties and non-Gaussianity of the distribution

We study now the typical sizes of relative increments. As shown in the previous section, the distribution of relative increments is centered around zero. We quantify this property using the norm of the average of relative increments, that is shown in Fig. 7.4. While the evolution in time of this quantitity is different for different groups of pairs, the values are an order of magnitude smaller than the size of typical relative increments, which shows that all averages are very close to the null vector.

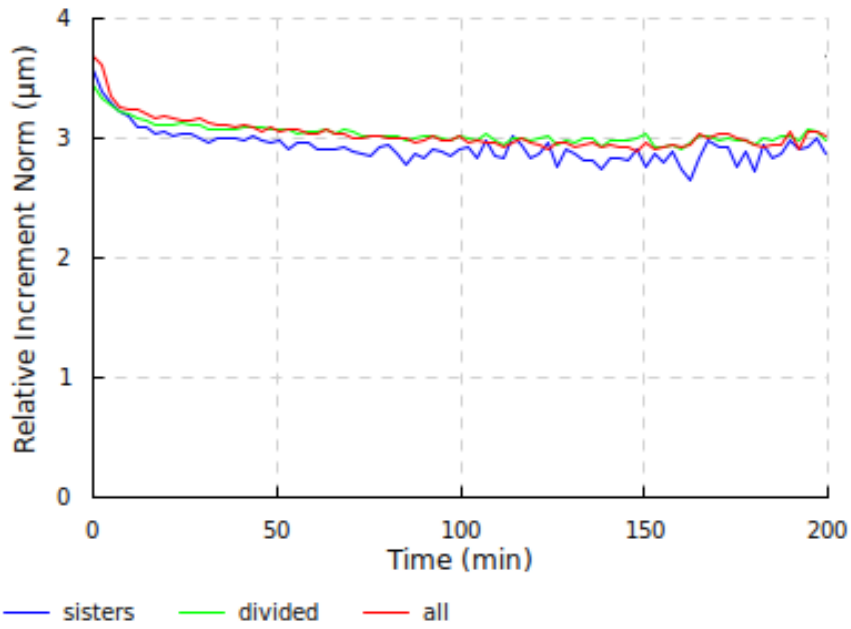

Figure 7.5: **Evolution of the average of the norm of relative increments on the embryo 141108aF.** The evolution is shown for pairs of sisters, neighbors of divided cells and neighbors in general.

The average size of relative displacements is show in Fig. 7.5. We can see that this size is slightly higher in the first time steps and decrease to a stable value in all cases. There is no significant difference between the size of relative displacements between the three groups.

We aim to identify the distribution of these increments. The first natural test is to compare this distribution with a Gaussian one whose mean and standard deviation parameters have been calculated from the sample data. The Kolmogorov-Smirnov test is able to distinguish the two distributions both for $p = 0.05$ and $p = 0.1$. This is expected since the distribution of relative increments is discrete due to the sampling of voxels in the microscope images.

In order to overcome this problem, we sample the Gaussian distribution accordingly. Given a real random variable X, we define the *integer sampled distribution* $\hat{X}$ by the probability distribution:

$$P(\hat{X} = n) = P\left(-\frac{1}{2} \leq X - n < \frac{1}{2}\right)$$

We compare the empirical distribution of relative increments with the integer sampled Gaussian whose mean and standard deviation parameters have been calculated from the sample data. However, the Kolmogorov-Smirnov test is still able to distinguish the two distributions for both $p = 0.05$ and $p = 0.1$. A similar test has been done for the binomial distribution, with negative results.

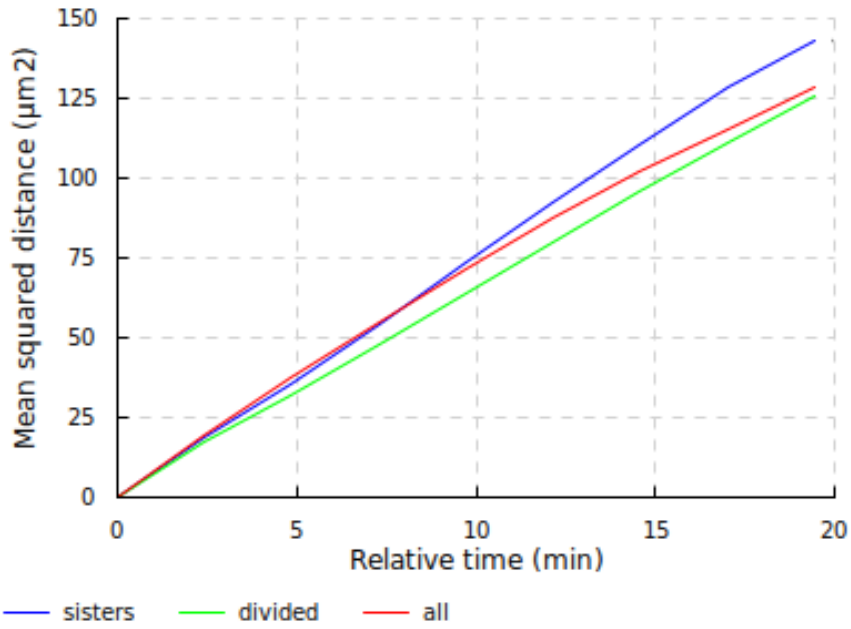

Figure 7.6: **Evolution of the mean squared relative displacement of pairs of cells in initial form, first 25 minutes.** This mean is calculated over pairs of sisters, neighbors of divided cells and neighbors in general in the embryo 141108aF.

In order to obtain different characteristics from a Gaussian while retaining symmetry, we tested q-Gaussian distributions (Tsallis, 2009). The main parameter of this family of distributions is determined by the excess kurtosis of the data. The calculation of the empirical excess kurtosis oscillates in time around zero, from negative to positive. Since the signal of this parameter determines the interpretation given to the distribution, we consider this test negative.

Mean squared relative displacement

We present now the evolution in time of the mean squared relative displacement for each group of cells. We divide this in three parts, up to 10, 50 and 100 time steps, which correspond to about 25 minutes, 2 hours and 4 hours. All the curves are presented in initial and averaged forms, as defined in the previous sections.

In the curve in initial form up to 25 minutes (Fig. 7.6 and Fig. 7.7) we see a slightly concave behavior. The averaged curve is more straight, meaning that this concave behavior is particular to the first time steps only. Furthermore, the displacements are sightly larger in the initial form, which corresponds to the larger sizes of relative increments in the first time steps.

The curves up to 2 hours (Fig. 7.8 and Fig. 7.9) are globally straight in both

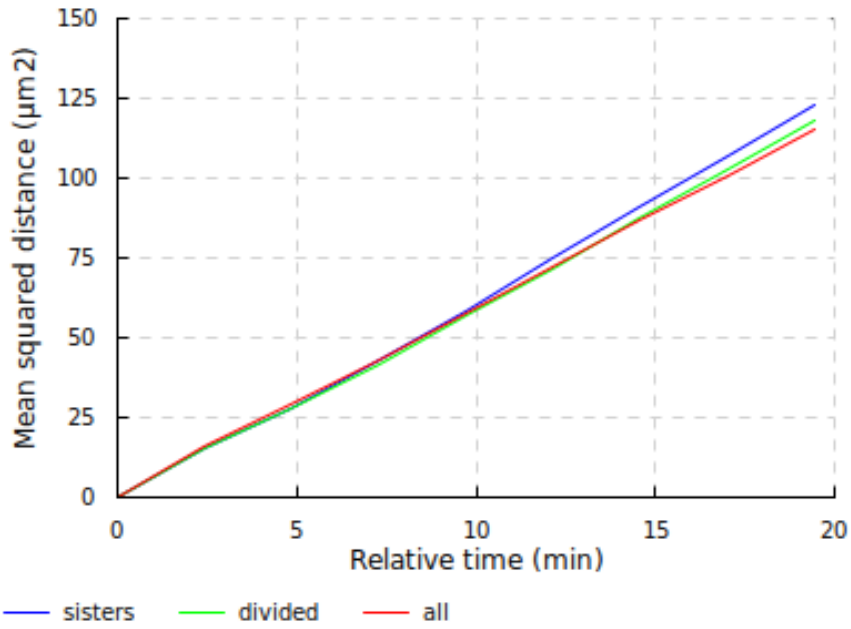

Figure 7.7: **Evolution of the mean squared relative displacement of pairs of cells in averaged form, first 25 minutes.** This mean is calculated over pairs of sisters, neighbors of divided cells and neighbors in general in the embryo 141108aF.

initial and averaged cases. As commented before, this behavior is similar to that of a Brownian motion, and is an indicator of uncorrelated increments. Displacements in both cases have a similar magniture, which shows once more the particularity of the first time steps. Finally, the curves which include divided cells grow slightly faster than the curve for neighbors in general.

The curves up to 4 hours (Fig. 7.10 and Fig. 7.11) have slightly convex behavior at the end. An interpretation of this observation is that as cells separate from each other, we start to see effects of drift due to differences of global flow. It is remarkable that the global linear behavior is observed up to 3h, which shows a very low rate of divergence between neighboring cells in general. The initial and averaged curves are similar in this case due to the lack of extra statistics for large intervals of time.

Autocorrelation of relative increments

We present here the calculated autocorrelation of relative increments, in both initial and averaged forms (Fig. 7.12, Fig. 7.13 and Fig. 7.14). First, there is almost no difference between the calculations in initial and averaged forms. This is an evidence to the stationarity of the process. We stress that this stationarity is related to the fluctuations of relative positions, and not to the whole process of embryogenesis.

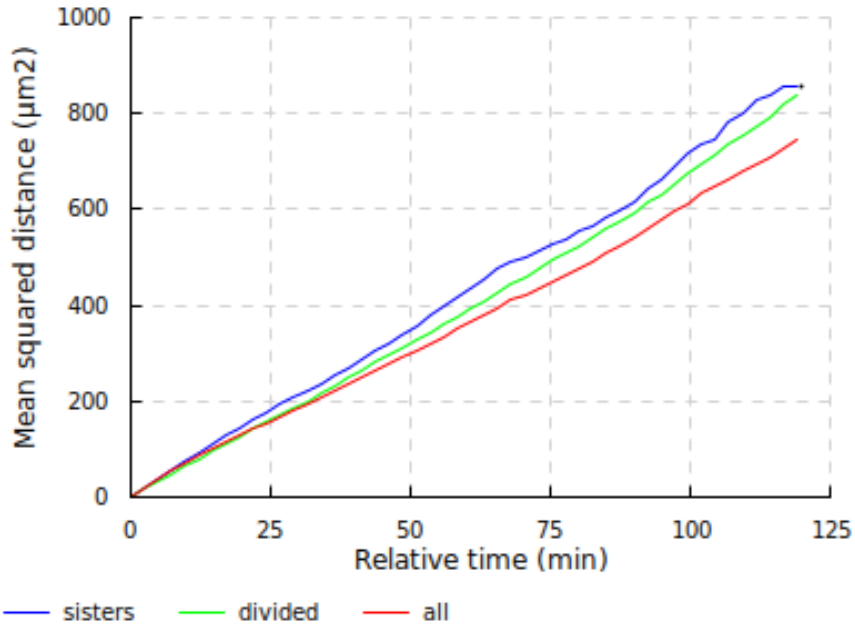

Figure 7.8: **Evolution of the mean squared relative displacement of pairs of cells in initial form, first 2 hours.** This mean is calculated over pairs of sisters, neighbors of divided cells and neighbors in general in the embryo 141108aF.

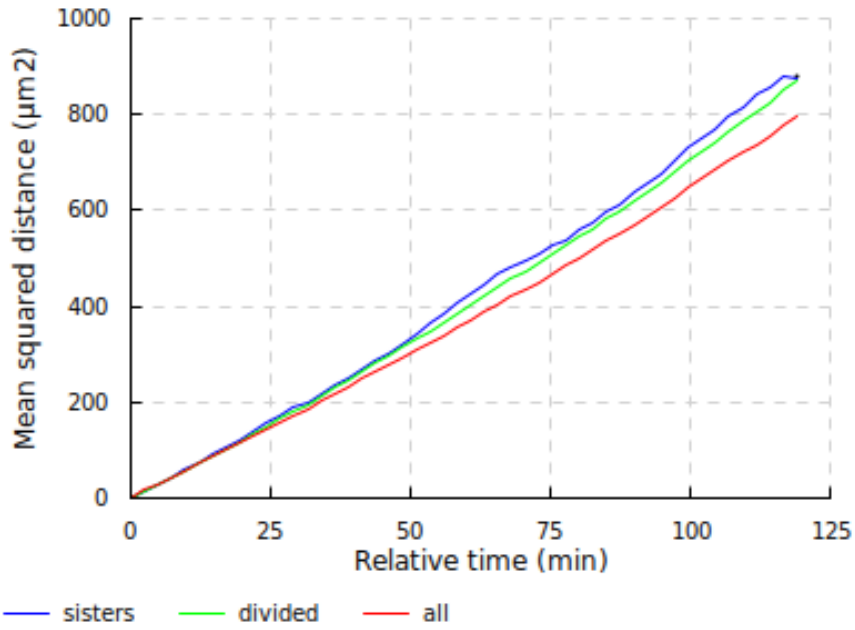

Figure 7.9: **Evolution of the mean squared relative displacement of pairs of cells in averaged form, first 2 hours.** This mean is calculated over pairs of sisters, neighbors of divided cells and neighbors in general in the embryo 141108aF.

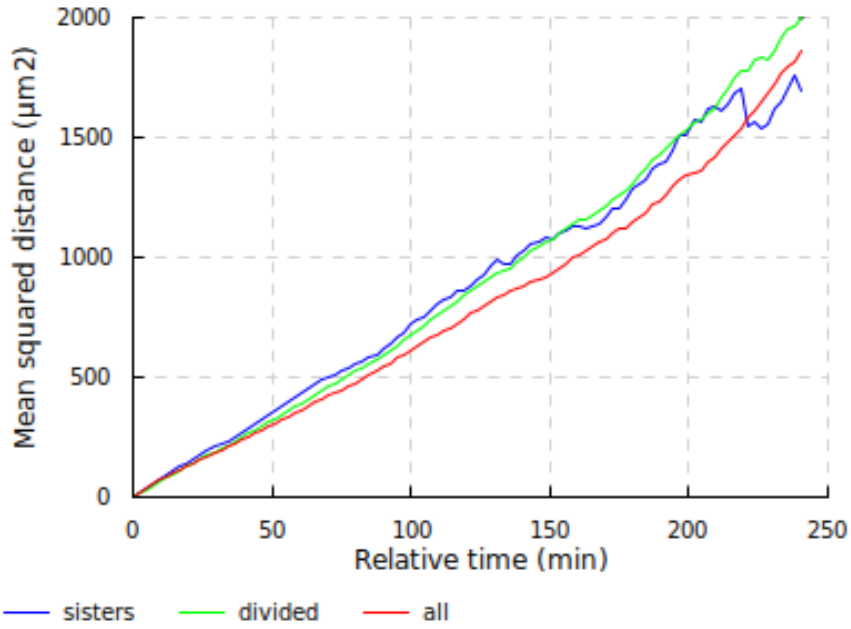

Figure 7.10: **Evolution of the mean squared relative displacement of pairs of cells in initial form, first 4 hours.** This mean is calculated over pairs of sisters, neighbors of divided cells and neighbors in general in the embryo 141108aF.

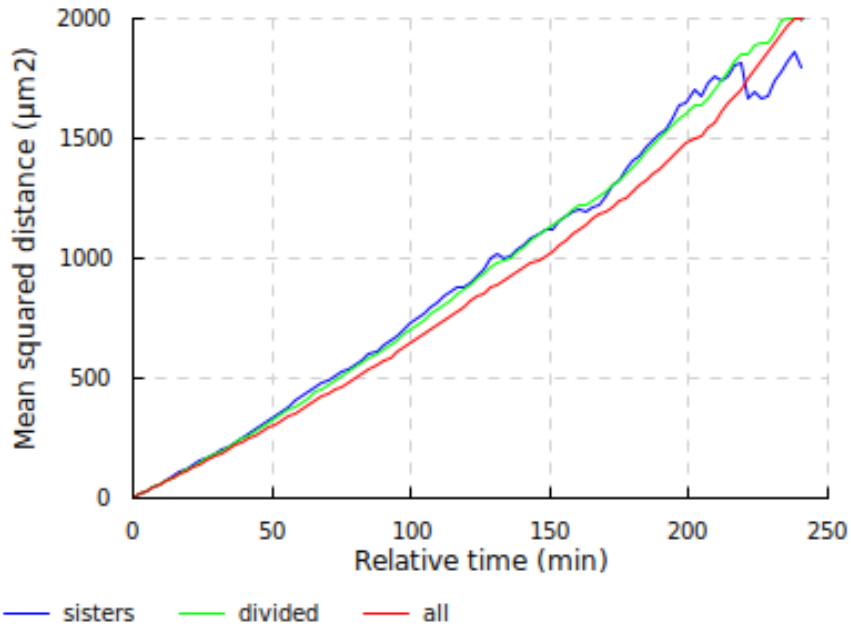

Figure 7.11: **Evolution of the mean squared relative displacement of pairs of cells in averaged form, first 4 hours.** This mean is calculated over pairs of sisters, neighbors of divided cells and neighbors in general in the embryo 141108aF.

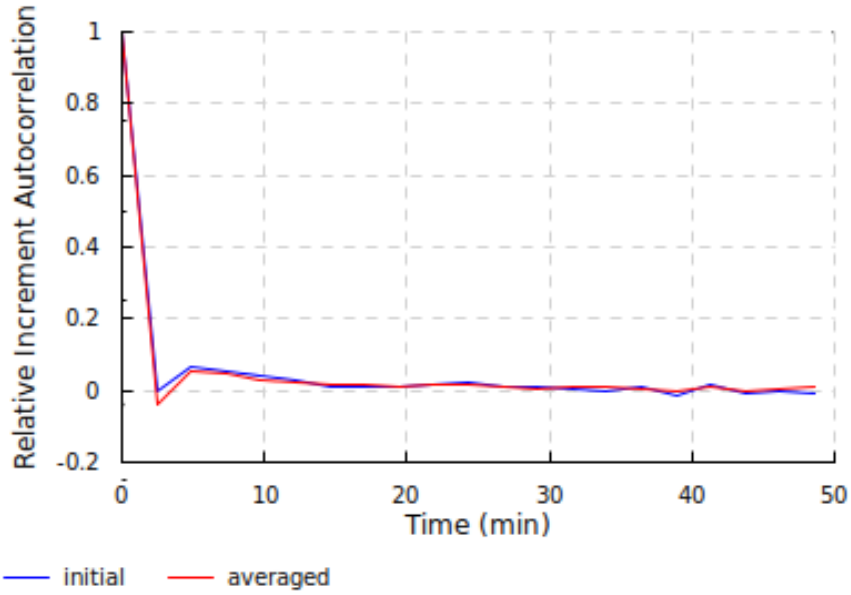

Figure 7.12: **Autocorrelation of the relative increments between sisters.** This autocorrelation is presented in initial and averaged form over the embryo 141108aF.

Furthermore, the curves are very similar between the three groups of cells. In all cases, we see a low negative correlation at the first time step, and very low positive one at second time step. Given the strong similarity of these curves and the slight difference on the previous ones, we posit that the identified difference between pairs which contains sisters and others is caused by a difference in the distribution of increments, and not by a difference in the nature of the process.

Nature of the process

These combined results show that this process is not very different from Brownian motion. However, some differences are important:

- While small, the correlations at time steps 1 and 2 are present in the statistics for all types of pairs, in both initial and averaged form. The fact that this behavior is visible even in the presence of averaging shows that it is probably not just statistical noise;
- The distribution of increments is not Gaussian, as discussed previously;
- The intensity of the process is not constant in time, even if the variations are small, as shown in the figure Fig. 7.5.

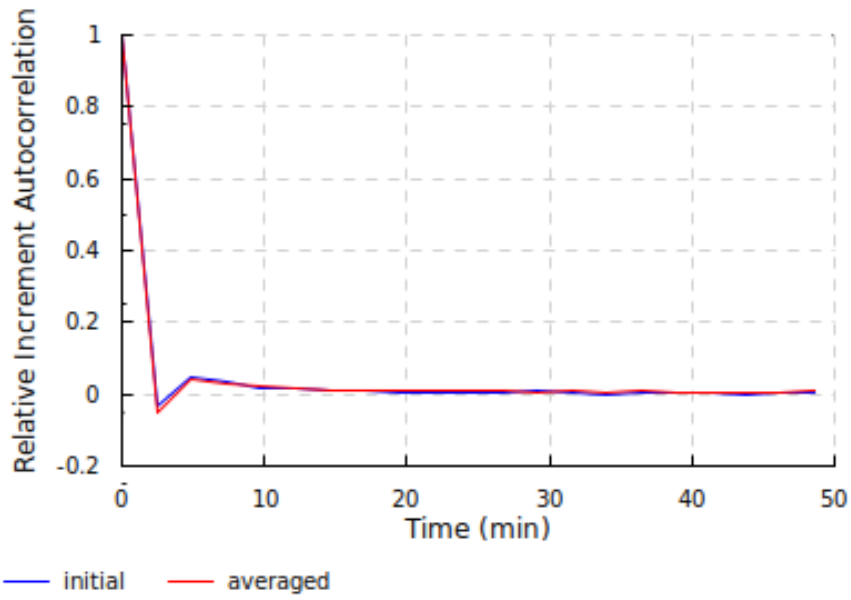

Figure 7.13: **Autocorrelation of the relative increments between divided neighbors.** This autocorrelation is presented in initial and averaged form over the embryo 141108aF.

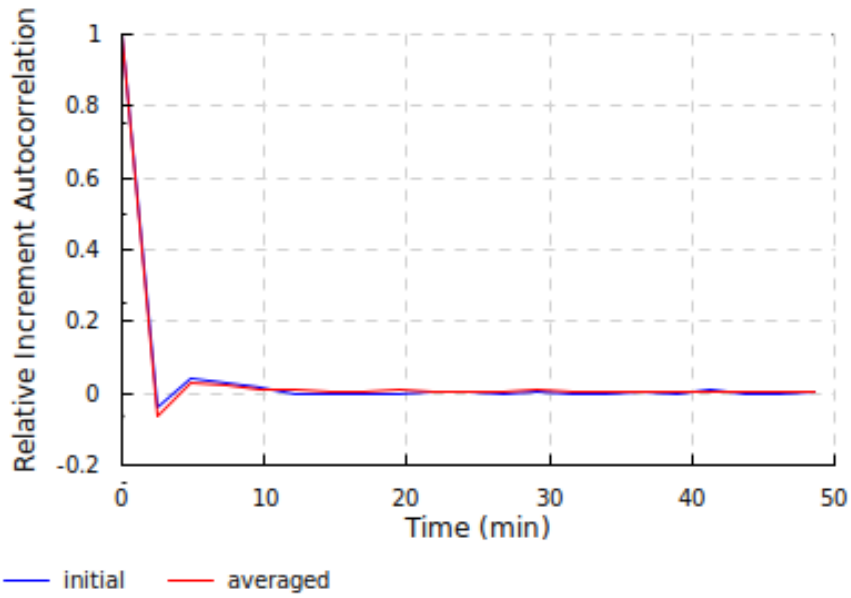

Figure 7.14: **Autocorrelation of the relative increments between neighbors in general.** This autocorrelation is presented in initial and averaged form over the embryo 141108aF.

7.4 Discussion

The numerical results give some insights about the behavior of neighboring cells both at short and long time intervals. There are differences in the short time dynamics of the relative displacement between pairs of cells that contain just divided ones and those which do not contain. We have some statistical evidence for this distinction and a suitable biological explication, the rounding of cells after division. However, since we are not able to distinguish these distributions after 35 minutes this may mean that this difference is not important globally for the morphogenesis of the animal.

These statistical differences could also be a valuable asset for the *identification* of cell divisions from the tracking. For this reason, one would need a more precise description of the distribution of relative increments. A possible way of obtaining better results is through the study of data sets with increased spatial and temporal resolutions in some particular region of the embryo. This would reduce the bias introduced by voxel sampling in the distribution and give a finer temporal information, at the cost of a smaller number of cells to study.

At long time intervals, there are no visible drift effects before 3h of elapsed time. This means that, in general, neighbors stay fairly close to each other, or at least close enough to be in an area that has a constant drift. While it is expected for cells to tend to stay close to their neighbors for the formation of morphogenetic fields, one may question if this behavior is a bias introduced by the tracking algorithm. This question can only be answered by the use of a data set that has been manually corrected to contain no tracking errors.

Finally, we have evidences of the stationarity of the process of deviation between neighboring cells. However, it is important to remark that this stationarity is linked to relative times, and does not imply a stationarity with relation to absolute times. The statistics for relative times involve a mixture of many different moments in time, since cells appear and disappear from the data set very often. This mixture and the consequent delocalization in time of the statistics is a possible reason for the observed stationarity.

Directions of research

As commented before, we can differentiate the behavior of three groups of cells at short time ranges, but cannot differentiate them at long time ranges. This means that a model of the long time dynamics of embryogenesis may make no distinction between these types of groups. In particular, such a model can treat cells after division just like regular cells. This assumption can be broken however in phases of embryogenesis where there are massive waves of simultaneous divisions. In this case, cells immediately after division are not exceptional anymore, which is important since small anomalous behaviors in embryogene-

sis can have permanent effects on the dynamics of the embryo.

The study performed here used an alignment of trajectories following their relative times, since we were interested in the behavior of cells after division. However, this choice makes the study of fluctuations delocalized in time. A similar study using absolute times would shed light on the temporal dynamics of noise, at the cost of having statistics that are irregularly distributed on time. An example of characteristic accessible through this study is the stationarity of deviations.

A natural model for the distribution of relative increments is a sum of two random processes, each one corresponding to a cell of the pair. By attributing a fixed distribution for each kind of cell, after division of not, one can estimate the most appropriate distributions for these types of cells. This estimation is simpler by the use of the assumption that both processes are independent, which permits the explicit calculation of the optimal distribution. However, since neighboring cells interact with each other, a more accurate model would consider the presence of correlations between these distributions. These correlations can also be a reason for the non-Gaussianity of the distributions of relative increments, since even the sum of two correlated Gaussians is not necessarily a Gaussian. A simplifying assumption that can be used in this case is that the distribution attached to each cell is Gaussian, leaving only its parameters and the correlation to be calculated from the data. The obtention of these results is very valuable for the modeling of trajectories.

The fact that autocorrelation of relative increments is very small has other consequences as well. Using the assumption of thermodynamical equilibrium, the term arising from the fluctuation-dissipation theorem (Kubo, 1966) is probably negligeable. This means that the deterministic and stochastic components of the system can be, in this case, completely disentangled and studied in separation.

Chapter 8

Conclusion

We divide this conclusion in three parts. We start by summarizing the main results of this thesis and discuss some global aspects of the project. We proceed by proposing future research directions based on the results and methods developed in this thesis. Finally, we conclude with some final remarks.

8.1 Main results

We divide the results in three parts: mathematical formalism, biological and physical results, and computational system.

Mathematical formalism

We began this thesis with the definition of a *temporal lineage* and its two possible points of view: the temporal one, which observes the temporal evolution of the set of cells composing the embryo, and the cellular one, which observes the points in time in which each cell is present. These points of view are isomorphic through the transformation $(\text{time}, \text{cell}) \to (\text{cell}, \text{time})$:

$$\sum_{t:\mathcal{T}} \mathcal{C}_t \cong \sum_{c:\mathcal{C}} \mathcal{T}_c$$

which allows one to change the representation of data when one is more convenient than the other. For example, the calculation of the spatial homogenization of a measurement is more convenient in the temporal point of view, while the calculation of its derivative is more convenient in the cellular point of view.

The possibility of representing observables over the embryo as functions, whose domain is formally defined, is a central point of this thesis. This representation is essential to approach this subject with the methodology of fundamental sciences like physics. In the same way as is done in these sciences, one does not need to manipulate these mathematical objects only formally: one can also manipulate them intuitively, using the conceptual semantics that is given to them.

In the following, we gave a categorical interpretation to temporal lineages that integrates into them the relations between cells and time steps that come from times and lineages. Under this interpretation, we proved the coherence between the two points of view with relation to the restriction of temporal lineages to time intervals and sublineages. These proofs give formal guarantees on the correctness of simple manipulations that are performed often.

We completed this formalism by the definition of an algebraic structure for trajectories, which allows a convenient and useful method of concatenation and intersection of trajectories.

These results provide a good mathematical foundation for the study of embryogenesis, as shown by the theoretical and computational applications. However, many possible extensions are possible.

Biological and physical results

The results obtained in this thesis are very diverse in their content, but they have in common the fact that they can be interpreted as *observations*. This is due to the fact that the hypothesis used through this thesis aimed to integrate the single cells into the whole embryo, not to offer conditions for *prediction*.

The first result we presented was the use of measurements over temporal lineages for the generation of *clusters* of cells having characteristics compatible with what biologists expect for a morphogenetic field, like temporal and genealogic persistance. The main goal of the creation of this algorithm was to produce a tool for the active search of local observables that are important for the determination of the embryo's morphogenetic fields. Using it, one can progressively enrich the set of measurements that generate the clusters and assess their relevance and compatibility with the morphology of the animal, therefore choosing measurements based on the results and their interpretation.

While we call this algorithm an *inference*, instead of a prediction, there are similarities. In the same way we use differential equations for the prediction of the next time step of a system in the future, this algorithm can be seen as predicting the next step in an organizational level. However, unlike in the case of differential equations, this next step belongs to a completely diverse space, which is not governed by the same rules. Since this space, composed of morphogenetic fields, does not behave in the same way as cells, we are unable to continue using the same technique.

The estimation of intercellular forces goes in the opposite direction. Instead of searching for larger structures, we look for finer dynamical details, which ultimately *determine* the processes underlying the data that we analyse. The use of simplifying hypotheses, like the Newtonian one, allow us to access these details by using an heuristic equivalent to a numerical interpretation of Occam's razor: find the minimal set of forces that reproduce the data. Heuristics like this one are not absolute, but are very common in physics, like in the principle of least action.

The results we obtained are compatible with our intuition of the process of the formation of the embryo. We are able to see localized regions with high stresses when there is a convergence of cells to these regions, and the global stabilization of stresses when morphogenetic movements become less intense. Even if these results are only visible on stress tensors, we expect that forces contain the same information, statistically, since stresses are calculated from them.

The complementary strategy takes us to the stochastic behavior of cells, since they compose the aspects of the dynamics of cells that are completely inaccessible from the scale of organization of the data. The approach that we use for the modelling of these behaviors is through random processes. Intuitively, the deviations from normal trajectories arise from activities internal to the cell, which can be completely intracellular or be related to the cell surface and its interac-

tion with the cell environment.

The study of the deviations between neighboring cells gave results that match very closely a Brownian motion, except for two points. The first one are temporal correlations, which seem to be too small to have an important interference on the global morphogenesis of the embryo. The second is the non-Gaussianity of the process, which we interpret from the point of view that the deviation between cells is produced from a sum of noises for each cell. As commented above, there are interactions between cells happening at microscopic levels. Therefore, a possible reason for the non-Gaussianity of process is this interaction, which introduces correlations between the individual noises and deforms the distribution of relative displacements.

Under the interpretation of these results as observations, we obtained a view of embryonic morphogenesis from many different angles. We have been able to access, within our limitations, the underlying components of movement, deterministic and stochastic. The independent study of these quantities was only possible due to an extensive homogenization of accelerations and the exploration of the local continuity of displacement fields. In addition to this cellular scale, we managed to obtain some details of the dynamics of larger scales of the embryo implied by the dynamics of cells themselves.

These results give a large body of quantitative observations. This is considered valuable in the field of developmental biology, that has been for a long time mainly relying on the visual observation and verbal description of morphogenetic events. Our quantitative results open the way to the measurement of differences between individuals whether normal or mutants or submitted to drug treatments. They are also expected to be the basis of theoretical models reconstructing morphogenetic processes underlying embryogenesis. If these are carefully considered, in the context of the assumptions that generated them, many possible directions of research are possible. We propose some of them in **Section 8.1**.

Computational system

Another outcome of this thesis is `lineageflow`, which is an extensible, user-friendly system for the manipulation of measurements over the embryo. It is architectured as a series of formally defined *interfaces*, which intermediate the relation between users, algorithms and measurements. These interfaces allow the establishment of an ecosystem of reusable parts, which connect themselves in order to produce more sophisticated results.

Since all the results presented in this thesis have been produced with this system, they are highly reproducible. Furthermore, since these results can be produced from a graphical interface, they are also reproducible by non-specialists. After some basic training, an user should be able to produce the results obtained in this thesis, including graphs and 3D visualizations, in around two

days. This estimation comes from a personal use of the system for this thesis, and the average running times of algorithms.

Given the fact that algorithms are going to be used by people that are not necessarily familiar with their implementation, one must take extra care. The expected dynamics between peers is that the developer explains the general ideas behind the method to the user, and how the results are to be interpreted. With these informations, the user runs algorithms on its own data and interprets it. Therefore, it is of utmost importance that the implementation of algorithms follows the semantics given by them. This has been covered to some degree on the implementation of the library that composes the computational core of the system. This library implements the primitive types and operations that are proper to this domain through the use of denotations, that represent these operations as mathematical transformations.

The possibility of having 3D visualizations and statistical plots is also essential in this context, as they are the means we use for the interpretation of data. In this way we can put the data on the hands of the people that can make the best use of the results, and not necessarily on the hand of those capable of producing them.

We see many possibilities for the expansion of this system, which we comment in the next section.

8.2 Future Directions

We propose here directions of research that involve either the interaction between the independent studies developed in this thesis or partially unexplored subjects. Each subsection explores a particular interaction.

Clustering and forces

There are two main applications for the interaction of the clustering algorithm with the estimation of stresses. One is the generation of clusters taking into account the value of the calculated stress tensors. This is a promising approach, since we can already see well defined patterns on the 3D visualization of stress tensors, which would then be summarized in time by the clustering algorithm. Attention has to be made on the notion of distance that is to be used between tensors, as different distances have different semantic contents. The use of this algorithm with stresses presupposes that morphogenetic fields have homogeneous stresses locally, while having a different interaction with morphogenetic fields, which lead to the next possibility.

The supposition of local homogeneity of stresses can lead to parametric models which assume a constant value in certain regions of the embryo. Therefore parameters can be estimated from a set of clusters, obtained through the algorithm, and the stress tensors calculated from the data. Such a model can offer more precise information about the nature of stresses, the material properties of the embryo tissues, etc.

Both applications can also interact. We can execute this process iteratively, by feeding the estimated stresses to the clustering algorithm and repeating the procedure. Ideally, this process converges eventually, giving a set of stresses that are homogeneous by definition and also by inference. This can be considered as a direct fitting of the model to the data, which only uses the estimation process as the starting point.

Clustering and noises

The analysis of noises performed in this thesis is focused on relative times and the appearance of cells in the data set, which makes the statistics delocalized in both time and space. While we already proposed an extension of the study for absolute time, the analysis of the spatial distribution of noise is more complicated, as the set of cells which is to be used for a statistical analysis is not evident to choose.

The clustering algorithm can be a solution for this problem, as it creates groups of cells that are consistently localized in space, and which are composed always by the same cells or cells in the same lineage, making the generation of

statistics easier. Therefore, through the 3D plotting of the resuming statistics over these clusters one can identify the possible correlation between space and fluctuations.

The inverse procedure can also be executed although with some precautions. In order to compare two cells, one must generate a comparable measurement for cells at each time step. Increments are not appropriate for this goal as the particular values they take are not as important as their distribution. We can overcome this problem by generating a local distribution of increments, by considering all the values of the increment in an interval of time around a given time step. From this point, one can compare the distributions using, for example, Kolmogorov-Smirnov statistics, which measure the difference between these distributions. The exclusive use of this statistics, with no mention to position can be especially interesting, as one can verify an unbiased distribution of noises around the embryo.

Forces and noises

Forces and noises complement each other in the composition the dynamics of cells, through its deterministic and stochastic parts. This is particularly true in a setting of thermodynamical equilibrium, where the term arising from the fluctuation-dissipation theorem is not relevant, due to the lack of correlation of noise. Therefore, one can proceed to the simulation of the movement of cell using the calculated stresses and increment distributions. The result can be directly compared with the actual trajectories, which offer yet another method of verification of the precision of these estimated values.

One of the possible origins for the trajectory fluctuations is the internal activity of the cell, which is regulated by genetic expression. Since it is known that mechanical stresses are capable of changing the expression of some genes, we may ask if this has any effect over the fluctuations. This question is also interesting from the mechanical point of view, as higher rates of compression may lead to less possibilities of movement, or fluctuations of a different nature.

There are multiple possibilities for this analysis. One is collecting the average intensity of stress and the variance of fluctuations for each cell, and plotting this in a scatter graph. The principal directions of this graph may provide indications on an interaction between these measurements. Another possibility for this study is done through the interaction of the three algorithms.

Clustering, forces and noises

The aforementioned relation between stresses and fluctuations can be accessed through clustering. Indeed, the clustering algorithm can generate groups of

cells which have homogeneous stresses or pressures, without imposing a particular spatial distribution. Therefore, for each group we will have typical stresses and a distribution of increments. These quantities can be correlated visually, since the number of clusters is much smaller than the total number of cells.

Multiple embryos and independence

The manipulations we presented in this thesis were always limited to a single embryo. The fact of introducing multiple embryos to models introduces new complexities and difficulties, but can lead to rich statistics and new paths of research. More specifically, the analysis of cohorts of specimens is required to assess the variability in a population. Our ability to quantify this variability is also the path to quantify differences between individuals corresponding to different genetic and environmental conditions. These are major issues in developmental biology We sketch some possible strategies in the following.

Microscopy imaging of embryos produces results that can be very different. Even when comparing two specimens of the same species, same genotype and in similar environments, the obtained results will not be exactly the same, as there is always some degree of biological variability, which is amplified when looking at the embryo as a large set of individual cells. These differences are ever larger when imaging the zebrafish, due to its size, as one must also define the field of view, which can change from one embryo to the other with the movements of the embryo during its development.

An important development when dealing with these variabilities is their *quantification*. That is, the definition of a methodology that compares embryos based on their observable characteristics. In the context of this thesis, this ought to be done through the comparison of measurements on temporal lineages. Supposing a temporal alignment of embryos, a possibility is to do this through the measurement of the Hausdorff or Gromov-Hausdorff distance (Gromov, 2007) between the images of these measurements.

Statistically, we can consider the images coming from different embryos as *independent*, since embryos are imaged separately. Supposing that the corresponding images are produced in conditions that are similar enough, they can be considered as different samplings of a same *image valued* distribution. The application of tracking algorithms, or any other method that consists only in the mathematical transformation of objects, does not change this judgement. These considerations give us effectively a set of independent and identically distributed (i.i.d.) random variables, which can be explored statistically.

Of particular interest are the artificial morphogenetic fields produced by the clustering algorithm. Given similar embryos, the algorithm is expected to produce similar clusters. A large enough cohort of comparable embryos would allow to undertand the impact of biological variability in the outputs of the clustering algorithm. This would lead to a clearer interpretation of its results

and in particular of the relations of these clusters with the morphology of the animal.

Morphogenetic fields and exchangeability

As noted before, morphogenetic fields are composed by cells that behave similarly and converge to a common destiny. Given their similarity, one can interpret these cells as a *sampling* of a distribution of behaviors that correspond to the given morphogenetic field. While we cannot say that the cells are independent, given the constraints given by the morphology of the field, we hypothesize that we can still can consider them as *exchangeable*. This statement allows us to take advantage of the De Finetti-Hewitt-Savage theorem (Shiryaev, 1995).

We reproduce this theorem here. Let $\{X_n\}$ be an exchangeable sequence of random variables with values in $\mathcal{X}$, that is:

$$X_1, ...X_n \cong X_{\pi(1)}..X_{\pi(n)}$$

for any permutation π, where $\cong$ represents the equality of distributions. Then there is a probability measure μ of the set of probability measures $P(\mathcal{X})$ such that:

$$P(X_1 \in A, ..., X_n \in A_n) = \int Q(A_1)...Q(A_n)\, \mu(dQ)$$

where both μ and Q can be calculated from the empirical distribution of values. We can summarize it by saying that an exchangeable set of variables can be represented as a mixture of i.i.d probability distributions.

The clustering algorithm presented here generates groups of cells that are uniform based on the criteria of positions and velocities, which are purely kinematic measurements. That is, we can interpret them as artificial morphogenetic fields for these quantities. Furthermore, the relative increments of trajectories and estimated stresses are directly related to dynamics. Therefore, we can posit that the distributions of these measurements are exchangeable and prone to be studied via the De Finetti-Hewitt-Savage theorem, which greatly simplifies the search for statistical models. A large cohort of similar embryos is likely to provide better statistics for this estimation.

In addition, this theorem being a two-way bridge, the exchangeability hypothesis can be statistically tested with a large enough cohort of similar embryos. Through the application of the clustering algorithm we can obtain a cohort of identified morphogenetic fields, which we suppose to be identified in some way that enables the matching of fields in different embryos. Using this data we can:

- find the best i.i.d. probability distribution for each morphogenetic field in each embryo;
- find the best mixture model for each cohort of a given morphogenetic field.

Therefore, we can compare the distribution of i.i.d. laws with the mixture model, and the convergence of both with the growing size of the cohort can be seen as a test of exchangeability.

Collaboration system

During this thesis we created `lineageflow`, which serves as a collaboration system between biologists, mathematicians, physicists and computer scientists. Much of its development is independent of the study of embryogenesis, depending only on certain ingredients that we put in evidence here.

We consider the backbone of this method to be the declarative interface for algorithms in the form

```
algorithm :: Parameter -> Input -> Output
```

which allows them to form an actual ecosystem of reusable parts. This declarative interface is only possible since we can define inputs and outputs to be composed of *measurements*, which have a standard access method in the database, allowing the automation and uniformization of the user interface. On its turn, the declaration of measurements is only done properly due its *mathematical representation*. This representation is what allows the algorithms to be fed the proper kinds of data, and have its outputs used properly.

Therefore, we propose that through a precise mathematical representation of objects of study, one can extend this approach to other domains. Ideally, one can also represent simultaneously multiple domains through this prism, allowing them to constitute a larger sharing community.

We discuss this proposition through the question on how one can *integrate* systems belonging to different domains. We start by noting that some of the uncertainties of the model used here can be traced back to the interaction between domains. While the declarations of domain and codomain of a measurement are unambiguous, there is no predetermined way of declaring specimens, trackings or even the identifiers of measurements, which leads to ambiguities. Under the point of view we are using here, this can be interpreted as a lack of formal representation of these objects.

In order to eliminate these ambiguities, one would need a way to declare the nature of data at each one of the steps it goes through, in a coherent way. One can interpret this problem as the determination of an interdisciplinar *type system* for scientific computing.

A possibility which comes from this metaphor, is to integrate different domains symbolically. In Haskell notation, the object that can describe both embryos and trackings would be

```
data Object = Embryo | Tracking
```

that is, we define objects on the integration of domains as being either objects of one domain or objects of the other. This can be appropriate as long as the objects are conceptually independent. But, for example, it would be incorrect to determine that:

```
data Object = Animal | Vertebrate
```

since there are intersections between these concepts. Our hypothesis and proposition is that the correct object can be obtained through a *colimit* (Barr & Wells, 2005), which is a generalization of a sum, in the correct conceptual category.

Finally, for every conceptual object there must be a concrete one which is composed by *data*, that can be explored and manipulated. Therefore, we propose that one must also define a category of structured data objects, that can be given a desired semantical denotation and manipulated as conceptual ones. These two ingredients would allow one to create an integrative, multidisciplinary system of both practical and theoretical usage.

This problem is evidently very complicated, and can only be progressively clarified by the application of similar ideas to different domains.

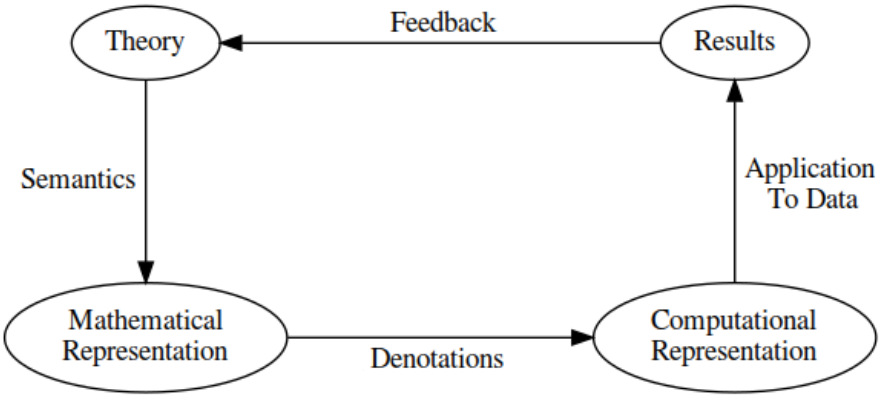

Figure 8.1: **Transdisciplinary feedback loop.** We represent her the interaction between theory and experience, through its intermediary steps: mathematical formulation and computational implementation.

8.3 Concluding remarks

We conclude by revisiting our main thesis, presented in the introduction:

> The goals of performing integrative studies of multilevel complex systems and theoretical studies through mathematical representations are not incompatible, and can provide a synergetic relation between all involved scientists.

The chapters of thesis covered both theoretical and practical aspects of the manipulation of spatio-temporal lineages, passing through a mathematical formalism, a computational ecosystem and physically inspired, biologically relevant applications. Most importantly, this was done in a way that makes these domains complementary to each other, by linking theory to experience.

Biology and physics provide semantics to mathematical objects that, otherwise, can only be manipulated formally. Denotations give mathematical semantics to computational objects that, otherwise, can only be manipulated in arbitrary manners. Together, these interactions allow theories to have a stronger influence over practice.

Conversely, using these semantics, one can transform mathematical manipulations into computational ones, producing results that can be interpreted under the light of biology and physics. This closes the ring, with practical outcomes providing empirical feedback to theory. This allows a continuous improvement of the interaction between biology, physics, mathematics and computer science. We represent this ring in Fig. 8.1.

Due to the possibility of the iterative enhancement of the method, this approach is very flexible. It allowed the exploration of embryogenesis from multiple angles, deterministic, stochastic and emergent, in a coherent manner. Also, it

opens the way for a rich variety of new studies, as demonstrated in the previous section.

However, this approach is confronted to challenges in a multiscale setting, where the emergence of higher scales of organization and the indirect observation of lower ones is of utmost interest. This is particularly evident in the emergence of artificial morphogenetic fields, where we produce objects of a different nature. For this reason, an expansion of the scope of the method is necessary, in order to encompass different objects of study. Such an expansion would require a cohesive effort from diverse groups of scientists, involved in both theoretical and experimental studies.

As shown in this thesis, the formal representation of objects is compatible with the integrative study of complex systems, and a powerful asset for sharing and collaborating. Furthermore, as discussed in the previous section, this method is independent of its substract, being applicable to different subjects, which allows the same interdisciplinary feedback loop to be applied. Therefore, it is hoped that the results exposed here encourage other scientists to make this approach more widespread and a possible organizational and collaborative principle for transdisciplinary science.

Bibliography

Alberts, B., Johnson, A., Lewis, J., & others. (2002). Universal mechanisms of animal development. In *Molecular biology of the cell. 4th edition*. New York: Garland Science.

Antoniou, A. (2000). *Digital filters*. McGraw-Hill Science/Engineering/Math.

Barr, M., & Wells, C. (2005). *Toposes, triples and theories*.

Begasse, M. L., Leaver, M., Vazquez, F., Grill, S. W., & Hyman, A. A. (2015). Temperature dependence of cell division timing accounts for a shift in the thermal limits of c. elegans and c. briggsae. *Cell Reports*, *10*(5), 647–653.

Blanchard, G. B., Kabla, A. J., Schultz, N. L., Butler, L. C., Sanson, B., Gorfinkiel, N., … Adams, R. J. (2009). Tissue tectonics: Morphogenetic strain rates, cell shape change and intercalation. *Nature Methods*, *6*(6), 458–464.

Bott, R., & Tu, L. W. (1982). *Differential forms in algebraic topology*. Springer.

Bourgine, P., Čunderlík, R., Drblíková-Stašová, O., Mikula, K., Remešíková, M., Peyriéras, N., … Sarti, A. (2010). 4D embryogenesis image analysis using pde methods of image processing. *Kybernetika*, *46*(2), 226–259.

Cadart, C., Zlotek-Zlotkiewicz, E., Berre, M. L., Piel, M., & Matthews, H. K. (2014). Exploring the function of cell shape and size during mitosis. *Developmental Cell*, *29*(2), 159–169.

Callen, H. B. (1960). *Thermodynamics: an introduction to the physical theories of equilibrium thermostatics and irreversible thermodynamics*. New York, NY: Wiley.

Chien, S., Li, S., & Shyy, J. Y.-J. (1998). Effects of mechanical forces on signal transduction and gene expression in endothelial cells. *Hypertension*, *31*(1), 162–169.

Delaunay, B. (1934). Sur la sphère vide. a la mémoire de georges voronoï. *Bulletin de L'Académie Des Sciences de L'URSS*, (6), 793–800.

Dolstra, E., Löh, A., & Pierron, N. (2010). NixOS: A purely functional linux distribution. *Journal of Functional Programming*, *20*(5-6), 577–615.

Doolen, G. D. (1991). *Lattice gas methods: Theory, application, and hardware (special*

issues of physica d). A Bradford Book.

Elliott, C. (2009). *Denotational design with type class morphisms (extended version)*. LambdaPix. Retrieved from `http://conal.net/papers/type-class-morphisms`

Fagotto, F. (2014). The cellular basis of tissue separation. *Development 141, 3303-3318*.

Faure, E., Savy, T., Rizzi, B., Melani, C., Stašová, O., Fabrèges, D., ... Bourgine, P. (2016). A workflow to process 3D+time microscopy images of developing organisms and reconstruct their cell lineage. *Nature Communications*, *7*, 8674.

Feynman, A. R., R. P.; Hibbs. (1965). *Quantum mechanics and path integrals*. New York: McGraw-Hill.

Fischer, R. S., Wu, Y., Kanchanawong, P., Shroff, H., & Waterman, C. M. (2011). Microscopy in 3D: A biologists toolbox. *Trends in Cell Biology*, *21*(12), 682–691.

Fletcher, C. R. (1973). Elementary rings and modules, by iain t. adamson. pp 136. 1·50. 1972 (oliver and boyd). *The Mathematical Gazette*, *57*(400), 145.

Fleury, V. (2011). A change in boundary conditions induces a discontinuity of tissue flow in chicken embryos and the formation of the cephalic fold. *The European Physical Journal E*, *34*(7).

Fleury, V. (2012). Clarifying tetrapod embryogenesis by a dorso-ventral analysis of the tissue flows during early stages of chicken development. *Biosystems*, *109*(3), 460–474.

Frolkovič, P., & Mikula, K. (2007). High-resolution flux-based level set method. *SIAM Journal on Scientific Computing*, *29*(2), 579–597.

Golan, J. S. (1999). *Semirings and their applications*. Springer Netherlands.

Goldhirsch, I., & Goldenberg, C. (2002). On the microscopic foundations of elasticity. *The European Physical Journal E - Soft Matter*, *9*(3), 245–251.

Goldstein, H., Jr., C. P., & Safko, J. L. (2001). *Classical mechanics (3rd edition)*. Pearson.

Grabert, H. (1982). *Projection operator techniques in nonequilibrium statistical mechanics*. Springer Berlin Heidelberg.

Grimmett, G. R., & Stirzaker, D. R. (2001). *Probability and random processes*. Oxford University Press.

Gromov, M. (2007). *Metric structures for riemannian and non-riemannian spaces*. Birkhäuser.

Guennebaud, G., Jacob, B., & others. (2010). Eigen v3. http://eigen.tuxfamily.org.

Holliday, R. (1990). DNA methylation and epigenetic inheritance. *Philosophical*

Transactions of the Royal Society B: Biological Sciences, *326*(1235), 329–338.

Holoborodko, P. (2008). Smooth noise robust differentiators. `http://www.holoborodko.com/pavel/numerical-methods/numerical-derivative/smooth-low-noise-differentiators/`.

Howard, J. (2002). Mechanics of motor proteins. In *Physics of biomolecules and cells. les houches, vol 75*. Springer, Berlin, Heidelberg.

Ishihara, S., Sugimura, K., Cox, S. J., Bonnet, I., Bellaïche, Y., & Graner, F. (2013). Comparative study of non-invasive force and stress inference methods in tissue. *The European Physical Journal E*, *36*(4).

Jacobi, M. N. (2009). Hierarchical dynamics. In *Encyclopedia of complexity and systems science* (pp. 4588–4608). Springer New York.

Jost, J. (2015). *Mathematical concepts*. Springer.

Käfer, J., Hayashi, T., Marée, A. F. M., Carthew, R. W., & Graner, F. (2007). Cell adhesion and cortex contractility determine cell patterning in the drosophila retina.

Kimmel, C. B., Ballard, W. W., Kimmel, S. R., Ullmann, B., & Schilling, T. F. (1995). Stages of embryonic development of the zebrafish. *Developmental Dynamics*, *203*(3), 253–310.

Kolmogorov, A. (1933). Sulla determinazione empirica di una legge di distribuzione. *G. Ist. Ital. Attuari.*, (4), 83–91.

Kubo, R. (1966). The fluctuation-dissipation theorem. *Reports on Progress in Physics*, *29*(1), 255.

Landau, L., & Lifshitz, E. (2013). *Fluid mechanics*. Elsevier Science.

Landau, L., Lifshitz, E., Kosevich, A., & Pitaevski, L. (1986). *Theory of elasticity*. Butterworth-Heinemann.

Lämmel, R., & Peyton Jones, S. (2003). Scrap your boilerplate: A practical approach to generic programming. In *SIGPLAN workshop on types in language design and implementation*. ACM Press.

Lehner, M. C. (2014). Concepts are kan extensions": Kan extensions as the most universal of the universal constructions.

Lipovaca, M. (2011). *Learn you a haskell for great good!: A beginner's guide* (1st ed.). San Francisco, CA, USA: No Starch Press.

Luxburg, U. von. (2007). A tutorial on spectral clustering. *Statistics and Computing, 17 (4)*.

Makabe, K. W., & Nishida, H. (2012). Cytoplasmic localization and reorganization in ascidian eggs: Role of postplasmic/PEM RNAs in axis formation and

fate determination. *Wiley Interdisciplinary Reviews: Developmental Biology*, *1*(4), 501–518.

Marsaglia, G., Tsang, W. W., & Wang, J. (2003). Evaluating kolmogorovs distribution. *Journal of Statistical Software*, *8*(18).

Martin-Löf, P. (1985). *Intuitionistic type theory: Notes by giovanni sambin of a series of lectures given in padua, june 1980 (studies in proof theory)*. Prometheus Books.

McDevitt, T. J. (2012). Discrete lanczos derivatives of noisy data. *International Journal of Computer Mathematics, 89:7, 916-931*.

Melani, C., Campana, M., Lombardot, B., Rizzi, B., Veronesi, F., Zanella, C., ... Sarti, A. (2007). Cells tracking in a live zebrafish embryo. In *2007 29th annual international conference of the IEEE engineering in medicine and biology society*. IEEE.

Milner, R. (1978). A theory of type polymorphism in programming. *Journal of Computer and System Sciences*, *17*(3), 348–375.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems* (pp. 849–856). MIT Press.

Penrose, R., & Todd, J. A. (1955). A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, *51*(03), 406.

Polacheck, W. J., & Chen, C. S. (2016). Measuring cell-generated forces: A guide to the available tools. *Nature Methods*, *13*(5), 415–423.

Raymond, E. (2003). *The art of unix programming*. Pearson Education.

Rizzi, B., & Sarti, A. (2009). Region-based PDEs for cells counting and segmentation in 3D+Time images of vertebrate early embryogenesis. *International Journal of Biomedical Imaging*, *2009*, 1–9.

Ronceray, P. (2016, May). *Active contraction in biological fiber networks* (Theses No. 2016SACLS154). Université Paris-Saclay. Retrieved from `https://tel.archives-ouvertes.fr/tel-01359592`

Sackmann, E., & Bruinsma, R. F. (2002). Cell adhesion as wetting transition? *ChemPhysChem*, *3*(3), 262–269.

Sakurai, J. J. (1994). *Modern quantum mechanics; rev. ed.* Reading, MA: Addison-Wesley.

Savitzky, A., & Golay, M. J. E. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, *36*(8), 1627–1639.

Savy, T. (2007). Mov-it. http://bioemergences.eu/bioemergences/openworkflow-movit.php.

Scott, D., & Strachey, C. (1971). *Towards a mathematical semantics for computer*

languages. Univ.Oxford Computing Lab. Programming Research Gp.

Shiryaev, A. N. (1995). *Probability (graduate texts in mathematics) (v. 95)*. Springer.

Stoy, J. E. (1977). *Denotational semantics: The scott-strachey approach to programming language semantics*. MIT Press, Cambridge, Massachusetts.

Taylor, P. (1999). *Practical foundations of mathematics*. Cambridge University Press.

Tsallis, C. (2009). Nonadditive entropy and nonextensive statistical mechanics -an overview after 20 years. *Brazilian Journal of Physics*, *39*(2a), 337–356.

Willam, K. J., & Iordache, M. M. (2001). On the lack of symmetry in materials. In *Proceedings of the international conference trends in computational structural mechanics, 233-242*.

Zhang, J., Talbot, W. S., & Schier, A. F. (1998). Positional cloning identifies zebrafish one-eyed pinhead as a permissive EGF-related ligand required during gastrulation. *Cell*, *92*(2), 241–251.

université PARIS-SACLAY
ÉCOLE DOCTORALE
INTERFACES
Approches Interdisciplinaires : Fondements, Applications et Innovation

Titre : Une approche mathématique de la morphogenèse embryonnaire basée sur des lignages cellulaires spatio-temporels.

Mots clés : Morphogenèse ; Systèmes dynamiques ; Mécanique statistique

Résumé : Cette thèse traite des processus morphogénétiques au cours de l'embryogenèse précoce des vertébrés par le biais d'une étude mathématique et physique des lignages cellulaires spatio-temporels reconstruits à partir d'imagerie 3D+temps *in vivo*.
Notre méthodologie repose sur une représentation de type système complexe de l'embryon avec ses différents niveaux d'organisation en interaction et l'analyse formelle des déplacements cellulaires dans l'espace et dans le temps.
Nous avons conçu et mis en œuvre une méthodologie originale pour identifier dans les lignages cellulaires la formation de compartiments en cohérence avec les repères anatomiques et l'organisation des organes présomptifs.
En outre, nous proposons une stratégie pour inférer les forces biomécaniques sous-jacentes.
Nous délivrons également une interface informatique ergonomique, d'abord déployée pour mettre en œuvre notre méthodologie, mais aussi conçue pour être extensible et versatile.
Ces outils visent à construire une représentation commune pour les biologistes, les mathématiciens, les physiciens et les informaticiens explorant les processus de la morphogenèse des organismes vivants.

Title : A mathematical approach to embryonic morphogenesis based on spatio-temporal cell lineages

Keywords : Morphogenesis ; Dynamical systems ; Statistical mechanics

Abstract : This thesis approaches morphogenetic processes in the early embryogenesis of vertebrates through the mathematical and physical study of spatio-temporal cell lineages reconstructed from *in vivo* 3D+time images.
Our methodology is based on a complex systems representation of the embryo, with the interaction between levels of organization and the formal analysis of cell displacements in space and time.
We designed and implemented an original methodology to identify in cell lineages the formation of compartments in consistency with anatomical landmarks and the organization of presumptive organs.
In addition, we proposed a strategy to infer the underlying biomechanical forces.
We also delivered a user-friendly computer interface, first deployed for using our methodology but also designed to be extensible and versatile, which aims to be a common ground for biologists, mathematicians, physicists and computer scientists investigating morphogenetic processes in living systems.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France