Information processing in biology
A study on signaling and emergent computation

Tiago Ramalho

München 2015

Information processing in biology A study on signaling and emergent computation

Tiago Ramalho

Dissertation
an der Fakultät für Physik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Tiago Ramalho
aus Lissabon

München, den 10. Juni

Erstgutachter: Prof. Dr. Ulrich Gerland

Zweitgutachter: Prof. Dr. Erwin Frey

Datum der müdlichen Prüfung: 23.10.2015

Zusammenfassung

Erfolgreiche Organismen müssen auf eine Vielzahl von Herausforderungen einer dynamischen und unsicheren Umgebung angemessen reagieren. Die Grundmechanismen eines solchen Verhalten können im Allgemeinen als Ein-/Ausgabeeinheiten beschrieben werden. Diese Einheiten bilden die Umweltbedingungen (Eingänge) auf assoziierte Reaktionen (Ausgänge) ab. Vor diesem Hintergrund ist es interessant zu versuchen diese Systeme mit Informationstheorie – eine Theorie entwickelt um mathematisch Ein-/Ausgabesysteme zu beschreiben – zu modellieren.

Aus der Informationstheoretischen Sicht ist das Verhalten eines Organismus vom seinem Repertoire an mglichen Reaktionen unter verschiedenen Umgebungsbedingungen vollständig charakterisiert. Unter dem Gesichtspunkt der natürlichen Auslese ist es berechtigt anzunehmen, dass diese Ein-/Ausgabeabbildung zur Optimierung der Fitness des Organismus optimiert worden ist. Unter dieser Annahme, sollte es möglich sein die mechanistischen Details der Implementierung zu abstrahieren und die zu Fitness führenden Grundprinzipien unter bestimmten Umweltbedingungen zu verstehen. Diese können dann benutzt werden um Hypothesen über die zugrunde liegende Implementierung des Systems zu formulieren sowie um neuartige Reaktionen unter äußeren Störungen vorherzusagen.

In dieser Arbeit wende ich Informationstheorie auf die Frage an, wie biologische Systeme komplexe Ausgaben mit relativ einfachen Mechanismen in einer robusten Weise erzeugen. Insbesondere untersuche ich, wie Kommunikation und verteilte Berechnung zu emergenten Phänomenen führen kann, welche kollektiven Systemen ermöglicht in differenzierteren Weise auf seine Umwelt zu reagieren als ein einzelner Organismus.

Die vorliegende Arbeit ist haupsächlich in zwei Teile unterteilt: der erste Teil beschäftigt sich mit der Musterbildung im frühen Stadium der Embryonenentwicklung, und der zweite mit abstrakteren statistischen Eigenschaften der zellulären Kommunikation und der Berechnung von Netzwerkdynamiken.

Im ersten Teil der vorliegenden Arbeit werden wir Positionsinformation, welche in molekularen Gradienten im Fruchtfliegenembryo kodiert ist, quantifizieren. Zu diesen Zweck wenden wir statistische Instrumente auf experimentell erhaltenen Konzentrationsprofile an. Die erarbeiteten Instrumente können dann angewendet werden, um ein biochemisches Profil dass im statistischen Sinne maximal informativ ist zu bestimmen. Durch die Verwendung von Variationsrechnung erhalten wir eine analytische Lösung dieses Problems und bemerken, dass die optimale Profilform von der Statistik der molekularen Schwankungen abhängt. Obwohl beobachtete Profile maximal informativ zu sein schein, zeigen experimentelle Daten dass die endgültigen räumlichen Muster in der Fruchtfliege offenbar nicht ausschliesslich nur von diesem Profil ereugt werden.

Wir untersuchen weitere mustererzeugende Prozesse in denen sich räumliche Muster aus der Dynamik eines Regelnetzwerkes in einem räumlich verteilten System entwickeln. Durch Nächste-Nachbar Wechselwirkung wird Positionsinformation innerhalb des System übertragen, im Gegensatz zu Systemen in denen, wie im vorangehenden Kapitel, ein globales Organisationssignal direkt ausgelesen wird um den Muster zu erstellen. Um den relativen Einfluss dieser beiden Mechanismen zu quantifizieren, haben wir zwei einfache Modelle erforscht. Wir fangen mit einem zellulären Automaten Modell an, ein einfaches binäres System. Damit können wir verstehen welche Arten von Mustern durch verteilte Regelungsdynamik effizient codiert werden können. Dieses Vorgehen wird erweitert durch ein weiteres Modell, in welchen kontinuierliche Zustandsvariablen verwendet werden. In diesem Szenario modellieren wir die Genexpression mit einem rekurrenten neuronalen Netzwerk und verwenden die Gradientenabstieg Methode, um sie zu trainieren. Die daraus resultierende Dynamik sollte dem Zielmuster entsprechen. Um ein robustes Ergebnis zu gewährleisten, können wir in diesem Modell emergente Dynamik mit einem globalen Konzentrationsprofil kombinieren.

Im zweiten Teil charakterisieren wir den Einfluss von Fluktuationen auf biologische Informationsverarbeitung. Zell-Zell Kommunikationssysteme sind bekannt für ihre Zuverlässigkeit trotz Fluktuationen in der Umgebung. Aus diesem Grund versuchen wir das Rauschen in einem solchen System zu quantifizieren und den Kommunikationsmechanismus genau zu charakterisieren. Im Experiment wurden *E. Coli* Zellen verwendet, die genetisch dahingehend verändert wurden dass sie auf ein kleines, diffusionsfähiges Molekül mit Fluoreszenz reagieren. Um die entsprechenden Reaktionsparameter zu quantifizieren entwickelte ich ein Software-Tool, um automatisch Hellfeldmikroskopie Bilder zu segmentieren und gleichzeitig Fluoreszenzwerte zu messen. Mit Hilfe von maschinellen Lernen erhalten wir ein Datenset von hoher Qualität, welches uns erlaubt den Induktionsmechanismus mit einer höheren Genauigkeit vorangenhende Versuche zu quantifizieren. Diese Daten wurden dann verwendet, um ein synthetisches bakterielles Sender-Empfänger-System zu kalibrieren.

Schließlich entwickeln wir ein numerisches Verfahren, um die stochastische Dynamik von großen regulatorischen Netzwerken effizient zu approximieren. Diese Methode ist nützlich wenn man versucht regulatorische Netzwerkparameter aus experimentellen Daten abzuschätzen: mehrere Hypothesen können schneller ausgewertet werden, um die besten Parameter zu fitten.

Abstract

To survive, organisms must respond appropriately to a variety of challenges posed by a dynamic and uncertain environment. The mechanisms underlying such responses can in general be framed as input-output devices which map environment states (inputs) to associated responses (output). In this light, it is appealing to attempt to model these systems using information theory, a well developed mathematical framework to describe input-output systems.

Under the information theoretical perspective, an organism's behavior is fully characterized by the repertoire of its outputs under different environmental conditions. Due to natural selection, it is reasonable to assume this input-output mapping has been fine tuned in such a way as to maximize the organism's fitness. If that is the case, it should be possible to abstract away the mechanistic implementation details and obtain the general principles that lead to fitness under a certain environment. These can then be used inferentially to both generate hypotheses about the underlying implementation as well as predict novel responses under external perturbations.

In this work I use information theory to address the question of how biological systems generate complex outputs using relatively simple mechanisms in a robust manner. In particular, I will examine how communication and distributed processing can lead to emergent phenomena which allow collective systems to respond in a much richer way than a single organism could.

The thesis is divided into two large parts: one is concerned with pattern formation in the early stage of animal development, and the second is concerned with more abstract statistical properties of cellular communication and processing network dynamics.

In the first part of the current work we will start by quantifying the positional information in molecular gradients in the fruit fly embryo by applying statistical tools to experimentally obtained concentration profiles. The framework used to quantify positional information can then be applied in an optimization context to determine which profiles are optimally informative in a statistical sense. By using variational calculus we obtain an analytic solution to this question and observe that the optimal profile shape depends on the statistics of molecular fluctuations. While observed profiles appear to be optimally informative, experimental data suggests the final spatial patterns in the fruit fly do not appear to be directly encoded by these profiles.

We explore another pattern generating process, whereby spatial patterns are encoded in the dynamics of regulatory networks embedded in a spatially distributed system. Via nearest neighbor interaction, positional information can be transmitted across the system instead of being read out from a global organizing signal. To quantify the relative importance of these two mechanisms we have explored two simple models. We begin with the cellular automaton model, a simple binary system which will allow us to understand which kinds of patterns can be efficiently encoded in distributed regulatory dynamics and how. This approach is extended by using into continuous state variables. In this scenario we model gene expression using a recurrent neural network model and use gradient descent methods to train it to evolve in time towards specified spatial configurations. In this second model, emergent pattern formation can be combined with a global concentration profile providing positional information to ensure a robust outcome.

In the second part we characterize the influence of stochasticity on biological information processing. Cell to cell communication systems are known to be quite reliable in spite of environmental fluctuations, and thus we seek to quantify the noise in such a system and accurately characterize the communication mechanism. The experimental setup consisted of genetically engineered *E. Coli* cells which fluoresce in response to a small diffusible molecule. To quantify their response, I developed a software tool to automatically segment brightfield microscopy images and simultaneously measure fluorescence values. Using machine learning tools we obtained a very high quality dataset which allowed us to quantify the induction mechanism with higher precision than previous attempts. This data was then used to calibrate a synthetic bacterial sender-receiver system.

Finally, we develop a numerical method to efficiently approximate the stochastic dynamics of large regulatory networks. This method is useful when trying to reverse engineer regulatory networks from experimental data: multiple hypothesis can be evaluated more quickly in order to infer the correct parameters which describe the data.

Contents

List of Figures

List of Tables

Acronyms

RNA	Ribonucleic Acid
RNAp	RNA Polymerase
mRNA	Messenger RNA
DNA	Deoxyribonucleic Acid
TF	Transcription Factor
GRN	Gene Regulatory Network
CRN	Chemical Regulatory Network
MCMC	Markov Chain Monte Carlo
MLE	Maximum Likelihood Estimator
MAP	Maximum a Posteriori
MI	Mutual Information
FI	Fisher Information
KL	Kullback-Leibler
CA	Cellular Automaton
ECA	Elementary Cellular Automaton
FPE	Fokker-Plank Equation
SDE	Stochastic Differential Equation
Bcd	Bicoid
AP	Anteroposterior
DV	Dorsoventral

GFP Green Fluorescent Protein

RFP Red Fluorescent Protein

PCA Principal Component Analysis

tsl Torso-like Protein

Hb Hunckback

Gt Giant

Kr Krüppel

Kni Knirps

AHL N-acyl homoserine lactone

IPTG Isopropyl β-D-1-thiogalactopyranoside

Chapter 1

Computation in biological systems

The amount of scientific knowledge about biological systems has exploded in the past few decades. While a clear qualitative picture has emerged for a great number of systems, a complete quantitative understanding still eludes us. By quantitative understanding one refers to being able to – like in physics and engineering – calculate the temporal evolution of a set of observables in the system once the initial conditions and a set of time varying external perturbations are specified. [1]

The recent field of quantitative biology attempts to develop this type of understanding by using tools from mathematics, physics and computer science to model the behavior of simple biological systems [2]. It has been argued that as we probe deeper into the inner workings of biological systems, quantitative modeling will become more and more crucial to understanding [3].

Most living systems can be described as a kind of chemical reactor away from equilibrium. It is the set of chemical reactions taking place within which allow the system to maintain homeostasis, grow and replicate [4]. Indeed, it is helpful to conceive of the networks of interacting chemical reactions as the 'programming' of a biological system [5]: they determine the response to external stimuli. The field of systems biology attempts to reverse engineer this programming by combining experimental data with quantitative modeling of chemical reaction networks in order to statistically infer the topology of the reaction networks as well as any relevant quantitative parameters [6].

Within the systems biology framework, we can consider any biological system to zeroth order as an input-output device [7] (Fig. 1.1). Inside this black box lies a complex and disordered system evolved by natural selection. This agent is inserted in a natural environment which bombards it with inputs. The most important inputs will be various chemical species which interact with our agent, but we should not forget that a biological system is not purely the product of chemical kinetics. Other physical processes (i.e. mechanical [8], electromagnetic [9]) play an important role in various systems.

The goal of this system is not explicitly defined. Instead, it arises naturally out of the process of natural selection [10]: those agents which grow and replicate perpetuate their

[1] For an entertaining perspective on the difference in quantitative thinking between physicists and biologists, refer to [1].

existence; while those which do not cease to exist. By this principle, the agents we see today have evolved to optimally take advantage of the environment in which they are in in order to grow and replicate.

Under this principle, we expect that the outputs for a certain environment will be such that the agent's fitness as defined above will be maximal under the present conditions. As an example, a bacterium which detects the presence of lactose in the environment will produce a series of enzymes necessary to metabolize the lactose [11].

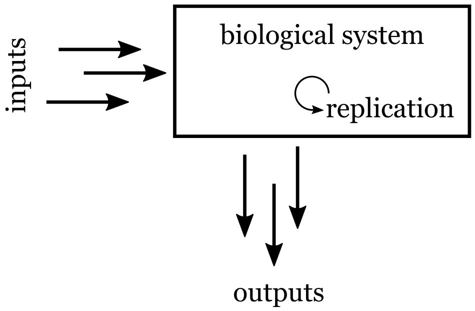


Figure 1.1: A biological system can be viewed as an input-output device: environmental inputs such as chemical concentrations are processed by the various chemical reaction networks operating within the organism and outputs are produced. Given the process of natural selection, I assume the organism's input-output mapping has been optimized to maximize the organism's replication rate.

On its own, the concept of a biological agent as an input-output device is not very helpful. The usefulness of this idea grows when we realize that other agents populate the environment. By producing specific outputs, an agent is able to manipulate the state of the common environment. In this way the various agents are able to interact, either cooperatively or competitively [12]. The nonlinear nature of the input-output relation leads naturally to emergent dynamics within this multi-agent system. [2] Now, because we abstracted away what is inside the black box which composes our agents, we can close off this multi-agent system into a single, coarse-grained, biological system.

This process of coarse-graining lends itself to the construction of a complexity *hierarchy* of biological systems:

1. A **chemical reaction network** is a set of single molecules which interact with other molecules in the environment. The output of the chemical reaction will be one, or more new molecules. Notably, it has been shown that Ribonucleic Acid (RNA) molecules are capable of self-replication [14].

2. A **cell** consists of a lipid membrane enveloping a number of molecules, such that multiple chemical reaction networks can operate in parallel and interact [4]. The emergent behavior of the networks defines the cell's properties.

[2] Emergence means that some properties of the whole system are not found in the individual parts [13].

3. A **multicellular organism** consists of a colony of multiple cells, where often we find specialization (i.e. certain functions are performed only by a subset of cells). Again we find that the organism's characteristics are a function of the individual cells' behavior.

4. An **ecosystem** consists of multiple interacting organisms. The output of this system will be large-scale changes to the natural environment in which the organisms are inserted.

Naturally, the above hierarchy is but an idealization. Eukaryotic cells contain subcompartments, known as *organelles* [4], which specialize in certain chemical reactions; bacterial colonies consist largely of identical cells, but exhibit peculiar emergent behaviors nonetheless; animal bodies are composed of organs, themselves composed of tissues. In spite of this, certain mathematical tools can be applied to each level of the hierarchy largely unchanged and successfully describe quantitative features of the system [15].

This universality suggests that it might be possible to model a system without needing to know the specific details at every level. Drawing from engineering, the idea is to isolate subsystems which perform specific functions within an organism [7]. These subsystems are now described by abstract input-output relations, and we can leverage our previous theoretical knowledge to make quantitative predictions for the system.

Such efforts have had success in smaller systems, such as simple metabolic pathways [11], but it is as of yet unclear whether all systems can be successfully modularized in a simple way. Given the random nature of natural selection we expect that components will be reused in disparate systems. Thus, it is likely that multiple subsystems are deeply interconnected [16]. [3]

The potentially vast web of interdependency among interaction networks complicates the effort of reverse engineering networks from experimental data, since most experimental efforts are based around perturbing individual nodes in the network [18], which cannot be done without downstream repercussions.

With the advent of synthetic biology, interest has turned towards the characterization of small circuits in terms of function and robustness to noise [19, 20]. Being modular and amenable to *de novo* experimental implementation, these circuits are of obvious interest to theoretical modelers as stepping stones to a greater understanding of biological computation.

In the remainder of this chapter I will introduce different biological models which illustrate the usefulness of the concepts above introduced. The language of chemical reaction networks will be pervasive, as they underpin all biological processes.

[3] A striking example of reuse is the conservation of the Hox gene system among vertebrates and invertebrates [17]

1.1 Regulatory networks

A wide array of chemicals are used by cells: lipids are used to build the cell wall; sugars are burned for energy; amino acids are combined to build proteins, which perform most advanced functions in the cell; and nucleotides are used to store genetic information. Since only some proteins from the vast repertoire of possibilities are necessary at any given time, the information necessary to recreate them is stored in long polymeric chains: Deoxyribonucleic Acid (DNA) and RNA.

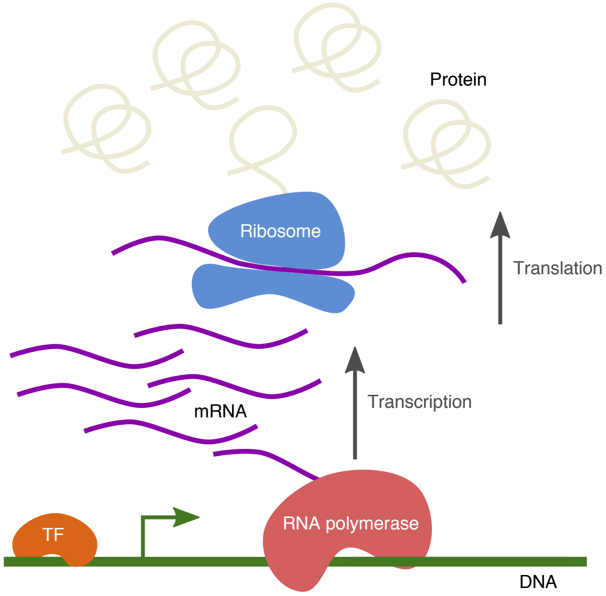


Figure 1.2: The central dogma of biology. All proteins which make up the cell's machinery are encoded as a nucleotide sequence in DNA. To decode this sequence first RNAp binds to DNA in order to copy the nucleotide sequence to a mRNA, which diffuses into the cytoplasm. Once there, it is read by a Ribosome, which converts its code into a protein, a process known as translation. The amount of protein which is produced at a given time is controlled by one or more TFs which bind to non-coding regions of DNA to change the binding affinity of RNAp.

The idea that the information coding for different proteins is stored in DNA and accessed when necessary is known as the *central dogma of molecular biology* [4]: the general idea is that the information encoded as nucleotide basepairs in DNA is copied to a Messenger RNA (mRNA) strand by RNA Polymerase (RNAp) in a process known as **transcription**, and then the mRNA is read by a ribosome which assembles amino acids in the order therein prescribed, a process known as **translation**. Naturally, there are exceptions to this general rule. As an example, an mRNA may undergo post transcriptional modifications prior to translation, which will alter the original information contained in the DNA.

A protein is said to be expressed if the process of transcription and translation is active for its corresponding sequence in the genetic code. Which proteins are expressed or not is determined by the process of gene regulation: protein expression can be up- or down-regulated from its original, basal rate of expression by other proteins which interact with DNA and RNAp. A protein with this function is known as a Transcription Factor (TF). TFs bind to specific regions in DNA, *cis*-regulatory regions, which are upstream of the actual protein coding region and increase or decrease binding affinity for RNAp at the transcription start site – thereby controlling how much of that gene gets transcribed.

Since transcription factors interact with one another and with DNA in complex ways, we can also refer to a set of regulatory interactions as a Gene Regulatory Network (GRN). So while a Chemical Regulatory Network (CRN) is fundamentally defined by the electrostatic interactions between the different molecules present in the system; a GRN is mostly defined by the nucleotide sequence in the genetic code. Clearly a GRN is just a subset of CRN since the interactions between TFs, promotor sites in the genetic code and DNA polymerase are also electrostatic in nature – we just distinguish GRNs due to the additional level of abstraction when compared to direct protein reactions.

The first evidence for regulatory control of protein expression was found with the discovery of the lac operon, which controls the expression of proteins associated with the metabolism of lactose in *E. Coli* [11]. Rather than a simple on-off switch, the lac operon only activates expression of its associated genes when lactose is present and glucose is absent in the environment. This type of combinatorial control provides an advantage as the energetic costs of producing the machinery to digest lactose are only paid when there are no other better sugars available (since glucose is a better source of energy, it is more cost efficient to digest it first).

Besides regulation via TFs, eukaryotic cells can also control gene expression in a myriad other ways. Genes can be repressed via DNA methylation, which can permanently silence a gene. This silencing can even be preserved across generations [4]. Additionally, chromatin structure (i.e. the way DNA is wrapped around proteins called histones) can also influence gene expression [21]. At this point in time, eukaryotic gene expression is still poorly understood when compared to prokaryotic gene expression, and thus most quantitative models are quite abstract.

With the advent of gene sequencing it has been possible to systematically characterize regulatory regions in the DNA [22] and quantitatively measure gene expression dynamics to reverse engineer regulatory networks [23, 24, 25]. Using this knowledge, specific pathways in cellular processes can be precisely targeted and perturbed [26] for therapeutical usage.

This knowledge can also be harnessed to design novel regulatory networks with specific function – synthetic biology. Progress in the last decade has been a veritable engineering tour de force: we have seen implementations of all kinds of circuits, ranging from oscillators [27]; to pattern forming systems [28]; edge detection circuits [9] and even associative memory systems [29].

1.2 Chemical sensing and adaptation

Consider again our picture of a biological system as an input-output device. At the cellular level, the basic input component will be the chemical sensing pathway. For cells to grow and reproduce in a given environment, they must detect which nutrients and toxins are available in the environment and react accordingly [30].

The earliest studies characterizing the accuracy of cellular sensing systems were performed in the context of *Escherichia Coli* chemotaxis. I was observed that *E. Coli* bacteria were able to detect nutrient concentration gradients and move towards the source, thereby maximizing consumption [31]. Initial efforts were focused on characterizing the accuracy of sensing chemicals in a diffusion-limited regime, where it was shown that the performance of simple sensing pathways can reach the limits set by physical constraints [32, 33, 34].

Later the focus turned towards the output: if a cell is to swim up a gradient over several orders of magnitude, its response cannot be linearly proportional to the measured concentration. Instead, its response should be proportional to the detected gradient [35]. Therefore, we expect that the signaling pathway should adapt to the current concentration and only respond proportionally to a detected fold change [35]. [4] It has been suggested that the *E. Coli* chemotaxis system behaves in exactly this way [36, 37].

This phenomenon, known as adaptation, is crucial for various cellular processes, such as maintaining homeostasis in the face of fluctuating external concentrations in yeast [38] and mammals [39]; or even higher-level sensory receptors [40, 41]. Indeed, there appear to be general principles for the design of adaptive systems at the network topology level [42].

The previous discussion was restricted to stationary environments. What happens if the concentrations fluctuate rapidly as a function of time? Interestingly, it appears cells are faced with a tradeoff: they can expend energy to keep a memory of previous concentrations, thereby smoothing the measured time course [43]; or they can exploit the inherent stochasticity [44] in their chemical pathways to respond differently to the same environmental conditions [45].

This last strategy is known as *bet-hedging* [46] and relies on the heterogeneity of the response: because the environment is changing rapidly, some cells' response will be inappropriate, while that of others will turn out to be suited to the environment. Those lucky enough to have chosen the correct response will grow quickly, and thus the population as a whole will benefit from the strategy [47].

1.3 Cell to cell communication: Quorum Sensing

Quorum sensing is a bacterial cell communication mechanism first discovered in *Vibrio fischeri*, a bioluminescent marine bacterium [48]. It is part of a symbiotic relationship with the *Euprymna scolopes* squid, where it accumulates in its light organ providing it with light. In return, the light organ provides a nutrient rich environment for the bacteria to multiply. To preserve energy, the bacteria only begin the process of bioluminescence if the

[4] The fold change is simply the ratio of the later concentration to the earlier one.

cell density in the current colony is high enough (indicating they are in the squid light organ).

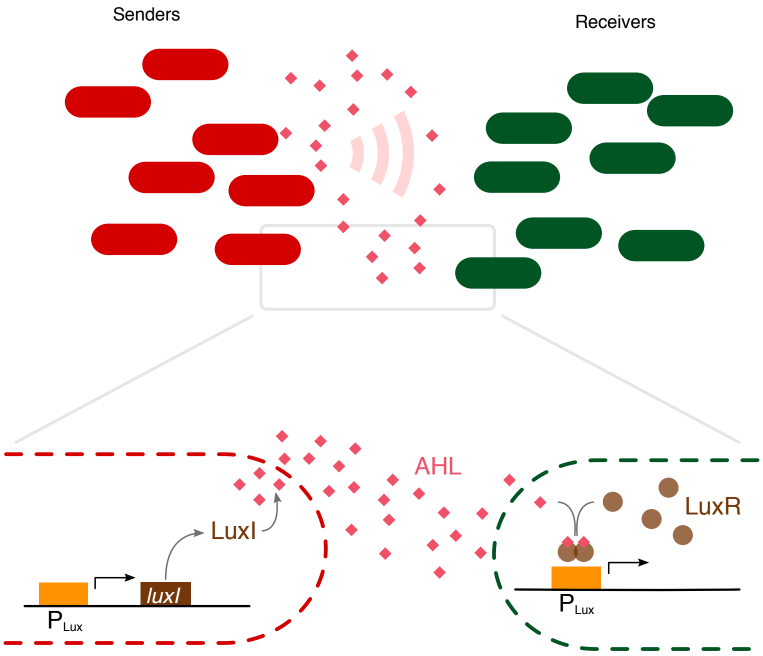


Figure 1.3: To communicate, cells send out a small diffusible molecule into the extracellular environment which is then read by nearby cells. When the signal is used to encode information about the population density, this process is known as *Quorum Sensing*. In the case of *Vibrio fischeri*, all cells are both senders and receivers, but synthetic systems can separate out the sending and receiving modules [49]. In the synthetic system above, the P_{lux} promoter controls the expression of *luxI*, which synthesizes AHL. AHL diffuses out into the environment, where it might permeate through a cellular membrane and bind to *luxR*, the dimer of which will induce the P_{lux} promoter on the receiving cell.

This process is controlled by the *luxICDABE* luciferase operon which codes for a number of proteins, two of which are key to the quorum sensing process: luxI and luxR. LuxI creates a small molecule, AHL which diffuses out of the cell. In a confined environment, extracellular AHL concentration will increase until reaching a threshold, at which point it binds to luxR and activates the luciferase operon. This generates a feedforward loop which amplifies the effect, making sure all cells are induced.

This system can also be artificially implemented in *E. coli* resorting to plasmids, independent pieces of DNA which can be integrated in the cell and can express genes therein encoded. Recent work has quantified the response of this system when separated into two components [49]. Some cells carry a plasmid strain containing the code for luxI (senders) and additional helper proteins, while others carry a plasmid coding for luxR. This technology may allow the development of novel pattern forming systems.

1.4 Fate determination in development

Morphogenesis is the process by which an organism's body acquires its shape. Initially, an embryo is composed only of undifferentiated cells but for a correct phenotype their developmental fate must be determined based on where they lie in the body plan [50].

This process of developmental fate determination is commonly associated with Waddington's epigenetic landscape metaphor [51]: a cell is visualized as a ball rolling downhill in a rugged landscape. Eventually it will be trapped in one of the valleys in this landscape, corresponding to a fixed fate. This landscape is a result of the nonlinear dynamics of regulatory networks, the attractors [52, 53] and bifurcations [54] of which determine the trajectory a cell will take through the landscape and its ultimate fate. Such systems also exhibit properties of excitability [55].

Drosophila melanogaster, the common fruit fly, has been a model system for morphogenesis and fate determination for decades. This is due to a multitude of factors: it is easy to breed and genetically modify; it is a long-germband organism, meaning that the body plan is established simultaneously throughout [5]; and it is possible to acquire quantitative data and do mathematical modeling.

The *Drosophila* fertilized egg consists of a large mass of centrally located yolk containing 256 nuclei (and no cell walls) produced by a series of eight nuclear divisions averaging 8 minutes each – a process known as **superficial cleavage**. These nuclei then start migrating to the periphery of the egg, where they undergo a few more rounds of division (Fig. 1.4). At the end of cycle 13, a cellular membrane begins to form around the nuclei, creating the **cellular blastoderm**, in which all cells are arranged in a single layer in the periphery of the egg.

This stage is termed the *midblastula transition*, and marks the beginning of **gastrulation**. A group of about 1000 cells located in the ventral midline of the embryo creates an invagination known as the **ventral furrow** – later forming the mesoderm. Simultaneously the **cephalic furrow** is formed, separating the head and thoracic region. The cephalic furrow is experimentally useful as a developmental timekeeping mechanism [57]. It is also at cycle 14 that RNA transcription is greatly enhanced and molecular gradients determining the body plan of the future insect are formed.

In the *Drosophila* egg, position is encoded by spatially distributed transcription factor concentration profiles. These TFs are know as **morphogens** [58]. Their function is to unlock certain developmental programs which will ultimately determine a cell's fate. This program of positional specification appears to be modularized in a way – there are separate systems for each of the body's primary axes. The best studied system is the one related to the anterior-posterior (AP) axis, while the dorsal-ventral (DV) system is also well studied. Left-right determination is not very well studied, and it's still unclear how it is achieved [50].

The process of AP patterning begins even before fertilizations: nurse cells present in the egg before fertilization deposit *bicoid* mRNA in the anterior part oocyte while *nanos*

[5] As opposed to short-germband where segments are added to the body sequentially in time [56].

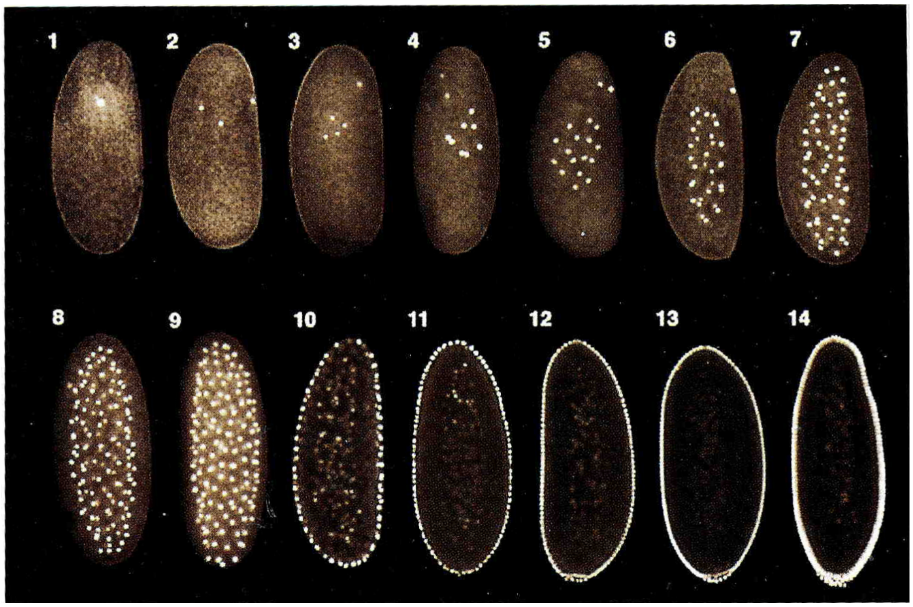


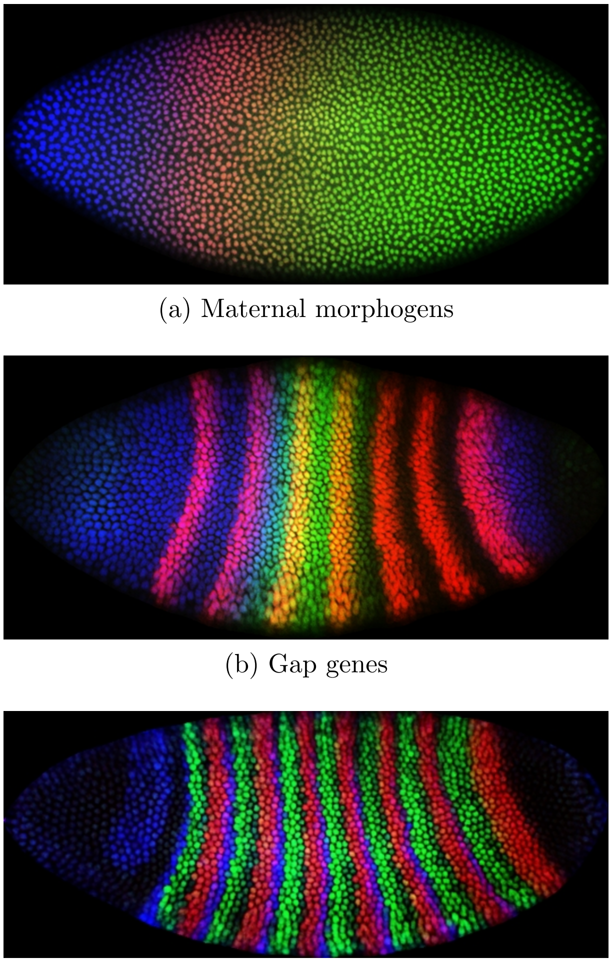
Figure 1.4: The 14 cleavge cycles. In the first cycle a single nucleus is present in the *Drosophila* oocyte, which then duplicates itself each cycle. After cell cycle 9 the nuclei migrate towards the periphery of the egg, and around cycle 13 a membrane begins to form around the nuclei, which will then go on to form mature cells. Adapted from *Gilbert* [50].

mRNA is transported to its posterior. These **maternal morphogens** essentially serve to establish anterior-posterior polarity. The bicoid spatial profile is thought to be established via a process of diffusion and degradation [59, 60], resembling an exponential decay (Fig. 1.5a).

Bicoid regulates downstream morphogens, known as **gap genes**, in a concentration dependent manner [61]. This discovery led to the introduction of the hypothesis of *positional information* [62]: position is encoded by bicoid concentration via an implicit map from a given concentration to a specific position along the AP axis. Bicoid is not solely a source of positional information, however. It is also used to regulate downstream genes in a combinatorial manner, along with the gap genes [63].

One of the gap genes downstream of bicoid is *Hunchback*, which roughly divides the AP axis in two halves (its concentration is high only in the anterior half of the AP axis). Its precise boundary is specified in a concentration dependent manner by bicoid [64]. If one changes the magnitude of bicoid concentration at the midpoint in the AP axis, the sharp Hunchback boundary will shift accordingly [64].

The gap genes *Krüppel*, *Giant* and *Knirps* form the rest of the gap gene system (Fig. 1.5b), an intermediate layer in the AP positional specification network [65]. Their spatial distribution is a product of activation by upstream genes (such as bicoid); self interactions; and diffusion. Once these processes even out, the pattern is established (albeit only for a short while). The interactions amongst gap genes are in general repressive in nature, leading to a process known as *canalization*: their cross repressions establish sharply defined domains [66] and it has been hypothesized that this network architecture buffers against noise [67]. Additionally, each gap gene self-activates, which generates a feed forward loop which further reinforces their activation domains.



(a) Maternal morphogens

(b) Gap genes

(c) Pair rule genes

Figure 1.5: Spatial distribution of morphogens in the *Drosophila* embryo. a) *bicoid* (blue), *even-skipped* (red), *caudal* (green). b) *hunchback* (blue), *even-skipped* (red), *krüppel* (green). c) *sloppy-paired* (blue), *fushi-tarazu* (red), *even-skipped* (green). Images obtained by immunostaining. Source: flyex [68].

The *nanos* maternal gradient is thought to provide additional positional information. It is known to repress hunchback and activate *Caudal*, which has a region of high concentration in roughly the posterior half of the embryo. Its regulatory interactions have not been as well characterized but it is thought to provide input to the gap gene network [65].

The gap gene network regulates a further downstream set of morphogens, the **pair-rule genes**. These segment the AP axis with very high precision, specifying regions with an accuracy of a single cell row (Fig. 1.5c) [69]. Examples of primary pair rule genes are *even-skipped*, *hairy*, and *runt*. Once these are established, later acting secondary pair rule genes such as *odd-skipped*, *fushi-tarazu*, and *odd-paired* are expressed.

A well studied pair-rule gene is *eve* (red in Figure 1.5) [70]. As time progresses, the spatial pattern formed by eve is refined, its boundaries increasingly sharp. These disjoint stripes are achieved thanks to multiple enhancers, which contain *cis*-regulatory regions specific to different morphogens. These morphogens act combinatorially to activate each enhancer in a limited region along the AP axis.

Recently it has been suggested that, similarly to the gap gene network, pair-rule genes are also able to repress each other [71]. This canalization appears to be a plausible mechanism for the observed precision in their stripe patterns, but it is as of yet unclear whether this is a general principle. It is nonetheless appealing to consider whether this type of interaction points towards a more distributed type of pattern forming mechanism [72], as will be discussed later.

Once both primary and secondary pair-rule genes are established, segment polarity genes are activated. At this stage, cells are fully compartmentalized and thus communication takes place via the cell membrane and specialized receptors (i.e. the Wingless protein is secreted from cells and binds to the Frizzled receptor). As their name indicates, these genes help establish divisions within each segment (which has been established by the pair rule genes). Finally the homeotic selector (Hox) genes [73] are activated combinatorially by the upstream morphogens, unlocking the final cell differentiation program.

Chapter 2

Information theory and stochastic modeling

In this chapter I will present an overview of the field of information theory which will be used extensively throughout the thesis to characterize the types of computations performed by organisms. I will then review the core concepts of mathematical modeling of regulatory networks framed in the context of information theory.

Information theory started with the publication of the seminal paper by Claude Shannon "Communication in the presence of noise" [74]. There, he quantified precisely what is meant by information and calculated the maximum rate of transmission of a message through a noisy channel without information loss. This insight then reverberated throughout the computer science community, leading to a better understanding of data compression, cryptography, coding and estimation theory.

The link between information theory and statistical mechanics was immediately recognized by Shannon, naming the basic quantity for the uncertainty in a random variable the 'entropy'. This link was formalized by Jaynes [75, 76, 77], who introduced the concept of maximum entropy and broadened the field to general bayesian inference.

2.1 Information measures

Consider a (discrete) random variable X defined over a set $\mathcal{X}$. The probability distribution $P(x)$ can be used to calculate the probability that x is true. Furthermore denote the conditional probability distribution as $P(x|y)$, which measures the probability of x given y true. [1] The joint distribution $P(x, y)$ measures the probability of both x and y being true. The joint distribution is related to the conditional by Bayes' theorem:

$$P(x|y) = \frac{P(x,y)}{P(y)} = \frac{P(y|x)P(x)}{P(y)} \tag{2.1}$$

[1] For a comprehensive introduction to the bayesian concept of probability refer to [78]

Where the left hand side is known as the posterior distribution, the conditional $P(y|x)$ is known as the likelihood and $P(x)$ is known as the prior. This reflects a common use of bayes' theorem: updating belief. The prior reflects our beliefs about x before receiving data; while the likelihood is the probability of seeing some specific data y if x is true; the updated belief about x given the data y is then stored in the posterior.

The expectation of a random variable shall be denoted as $\langle f(X) \rangle$ with

$$\langle f(X) \rangle = \sum_{x \in \mathcal{X}} f(x) P(x)$$

The above definitions are also valid for the case of continuous variables, in which case P is known as a *measure* and the concept of probability only makes sense under the integral sign. Nonetheless, all identities can be shown to be valid.

2.1.1 Entropy

The **entropy** of a random variable is defined as:

$$h(X) = -\sum_{x \in \mathcal{X}} P(x) \log P(x) \tag{2.2}$$

In the discrete case, since $P(x) \leq 1$ and $\log(P(x)) \leq 0$, we must have $h(X) \geq 0$. Intuitively, we can interpret the entropy as the amount of uncertainty or of missing information. Naturally $h(X) = 0$ only if there is a single state with probability one. In general we can say that the higher the entropy the greater our uncertainty about the next realisation of a variable. Take a Bernoulli distributed random variable:[2] entropy will be highest for a probability of heads/tails of $p = 0.5$ since there is equal probability for each outcome, whereas it will be minimal for $p = 0, 1$ since then we are guaranteed to know what the outcome will be.

The **conditional entropy** measures the uncertainty of a variable when the value of another is known and is defined as:

$$h(X|Y) = -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} P(x, y) \log P(x|y) \tag{2.3}$$

Another quantity related to the entropy is the **Kullback-Leibler divergence**, defined as:

$$D(P; Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)} \tag{2.4}$$

[2] A Bernoulli random variable describes a coin toss with probability p for heads and $1 - p$ for tails

The Kullback-Leibler (KL) divergence represents the amount of information lost when describing a source distribution P by a target distribution Q. It can be shown that $D(P;Q) \geq 0$, which lead to the common use of this quantity as a 'distance' between distributions. Nonetheless it must be noted that it is not a metric due to lack of symmetry. If two distributions are close however, we can look at $D(P;P+\delta P)$ as a metric (known as the Fisher Information, c.f. Sec. 2.2) which is the only natural Riemannian metric on the manifold of probability distributions (up to a constant).

All definitions made in this section can be easily transferred to the case of continuous probability distributions. In general, the same interpretation can be given to the resulting quantities with the caveat that entropy might be negative so it cannot be directly interpretable as the number of bits of uncertainty. Furthermore, it is not invariant to coordinate changes [78], while the KL divergence is.

2.1.2 Mutual Information

The Mutual Information (MI) between two random variables quantifies how many bits of information one variable carries about the other. It is defined as:

$$I(X,Y) = \sum_{x\in\mathcal{X},y\in\mathcal{Y}} P(x,y)\log\frac{P(x,y)}{P(x)P(y)} \tag{2.5}$$

Suppose the variables X and Y are independent. Then, their joint distribution factorizes which makes the ratio inside the logarithm unity and thus the mutual information is zero – this is what we expect from intuition about random variables (Fig. 2.1, right).

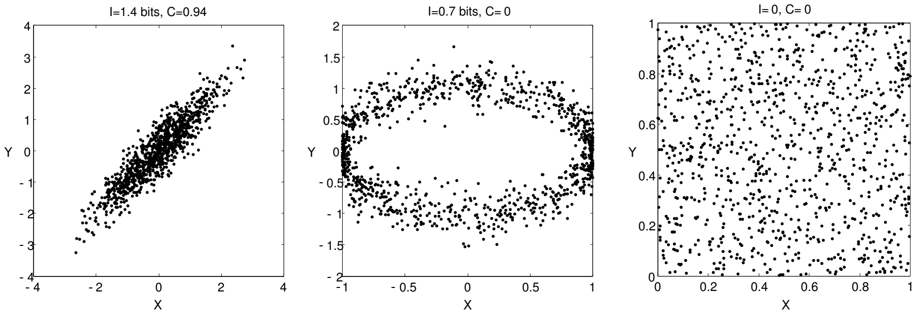


Figure 2.1: MI captures even nonlinear dependence between pairs of variables. In the left panel, linearly correlated variables show high MI and high Pearson correlation. In the middle panel, nonlinearly dependent variables show high MI but zero Pearson correlation. In the case of independent random variables, both MI and Pearson correlation are null. Adapted from *Tkačik et al.* [79]

Conversely, if one variable is a deterministic function of another, knowing one variable reveals everything about the other, and all that remains is the information due to the entropy contribution. This is more easily seen by rewriting the MI using the conditional entropy to show it explicitly as the reduction in uncertainty about one of the variables by the conditional entropy between both.

$$I(X, Y) = h(X) - h(X|Y) = h(Y) - h(Y|X) \tag{2.6}$$

In the deterministic case the second term would be zero, leaving only the entropy term. Another enlightening way to write the mutual information is by the use of the KL divergence:

$$I(X; Y) = D(P(x; y)|P(x)P(y)) \tag{2.7}$$

Here it is possible to directly read out the lack of informativeness in pairs of independent random variables. Most statistical tools used to discover relationships between random variables only look for linear correlation (Fig. 2.1, left) while MI is able to detect any type of nonlinear dependence (Fig. 2.1, middle). This makes it a very powerful, albeit underused statistical tool. Its lack of popularity in the statistics field is due to the fact that it is highly nontrivial to estimate the entropy from a finite set of data points [80].

A final way to look at MI is to write it as the average distance between the probability distribution for one of the variables given that the value of the other is either known (i.e $P(X|Y)$) or unknown ($P(X)$). In that perspective, the equation for MI looks like:

$$I(X; Y) = \langle D(P(Y|X); P(Y)) \rangle_X = \langle D(P(X|Y); P(X)) \rangle_Y \tag{2.8}$$

One deep insight which can be extracted from this information theoretic framework is the **data processing inequality** [81]. It states that given a random variable Y, there is no function or algorithm that can extract more than $I(Y; X)$ bits of information about X. Consider a Markov chain $X \to Y \to Z$ [3] (i.e. $P(x, y, z) = P(z|y)P(y|x)P(x)$), then $I(X; Y) \geq I(X; Z)$. Naturally, if the Markov chain assumption is relaxed, the inequality no longer holds (i.e. take the case of $Z = X + Y$ with X and Y binary coin tosses). The importance of this inequality lies in its application to determining the optimality of a signal processing procedure: once the information extracted has reached $I(X; Y)$ (assuming it can be estimated), the algorithm can be said to be optimal.

Finally it is worthwhile mentioning that the mutual information can be extended beyond pairs of random variables – with applications in time series analysis and multivariate statistics. Naturally the difficulties with its estimation from finite datasets only grow with the number of dimensions: a manifestation of the *curse of dimensionality.* The multivariate mutual information is defined as:

$$I(X_1, X_2, \ldots, X_N; Y) = \sum_{i=1}^{N} I(X_i; Y|X_1, X_2, \ldots, X_{i-1}). \tag{2.9}$$

[3] Meaning that Z is conditionally independent of X given Y

2.2 Fisher information

Coming back to the question of distinguishability of probability distributions, suppose a family of distributions $P(x|\theta)$ where θ is an n-dimensional parameter vector. These distributions form a manifold – a topological space which resembles an Euclidean space at small scales – where each point is a specific distribution $P(x|\theta)$ and the coordinates of which are θ.

How to measure distance between two points in this manifold? To do so, it is necessary to define the infinitesimal distance $d\ell$ between two points θ and $\theta + d\theta$. This distance might depend on *where* in the manifold we are which leads to the necessity of introducing a metric tensor g to account for that.[4]

$$d\ell^2 = g_{ij} d\theta^i d\theta^j \tag{2.10}$$

It can be shown that this metric tensor is unique up to a constant [78]. What is the metric for the probability distribution manifold? The relative difference between two close distributions can be written as:

$$\frac{P(x|\theta + d\theta) - P(x|\theta)}{P(x|\theta)} = \frac{\partial \log P(x|\theta)}{\partial \theta^i} d\theta^i \tag{2.11}$$

The factor before the infinitesimal is also known as the score [81]. Of course the metric should not depend on x, but the average of the score is zero, so consider its variance:

$$\mathcal{F}_{ij} = \int dx \; P(x|\theta) \frac{\partial \log P(x|\theta)}{\partial \theta^i} \frac{\partial \log P(x|\theta)}{\partial \theta^j} \tag{2.12}$$

This is known as the **Fisher Information Matrix**, and can be identified with the metric for the manifold of probability distributions (i.e. $d\ell^2 = F_{ij} d\theta^i d\theta^j$).

The Fisher Information also arises naturally from the concept of distance as defined by the KL divergence. Consider two probability distributions only $d\theta$ apart:

$$D(\theta; \theta + d\theta) = \int dx \; P(x|\theta) \log \frac{P(x|\theta)}{P(x|\theta + d\theta)} \tag{2.13}$$

A taylor expansion will result in:

$$\begin{aligned} D(\theta + d\theta; \theta) = & \underbrace{D(\theta;\theta)}_{=0} + \underbrace{\frac{d}{d\theta} D(\theta;\theta)}_{\langle\text{score}\rangle = 0} d\theta + \\ & \frac{1}{2} \underbrace{\frac{d^2}{d\theta^2} D(\theta;\theta)}_{\text{Fisher info.}} d\theta^2 + \ldots \end{aligned} \tag{2.14}$$

[4] Here the Einstein summation convention is used: repeated indices are to be summed over.

The second term is the average value of the score, which is zero as stated above. The third term is the Fisher Information [82]. In this way, the Fisher Information can be visualised as the curvature of the manifold of probability distributions.

As a final remark, it is worthwhile noting that the Fisher Information (FI) is not reparametrization invariant. Suppose a change of variables $\xi = h(\theta)$ with Jacobian matrix J, then $\mathcal{F}(\theta) = J\,\mathcal{F}(\xi)\,J^T$ [83].

2.3 Estimation theory

Given n i.i.d. measurements[5] $\mathbf{x} = \{x_1, ..., x_n\}$ drawn from $P(x|\theta)$, it is possible to calculate the posterior distribution for θ using Bayes' theorem:

$$P(\theta|\mathbf{x}) = \frac{\mathcal{L}(\mathbf{x}|\theta)P(\theta)}{\int_\Omega d\theta\ \mathcal{L}(\mathbf{x}|\theta)P(\theta)} \tag{2.15}$$

With the likelihood function

$$\mathcal{L}(\mathbf{x}|\theta) = \prod_i^n P(x_i|\theta)$$

Often a single final value for θ is desired, a so called estimate (henceforth denoted by $\hat{\theta}$). Given a posterior, how to compute an estimate?

Decision theory can help in this case. Suppose one needs to choose a certain action a for a certain 'true' parameter θ. Each action will be associated with a penalty incurred by making an error. The function that maps an action to its penalty is called a loss function, and the expected loss (or risk function [83]) is defined by:

$$r(a) = \int d\theta\ l(a|\theta)P(\theta|\mathbf{x}) \tag{2.16}$$

The goal is to find the set of actions a which minimise the expected loss (by finding a stationary point such that $\nabla_a r(a) = 0$). Given different different types of loss functions, different strategies are found for a [83]. In the case of estimation theory, a is a number (or vector) which should be as close to θ as possible. Some possible loss functions are:

Quadratic loss: $l(a|\theta) = (a-\theta)^2$, where small errors are unimportant but large errors are severely penalized. In this case the optimal choice for a is the posterior mean: $a = \langle P(\theta|\mathbf{x})\rangle$.

Linear loss: $l(a|\theta) = |a-\theta|$, where all errors are important, but the closer the better. In this case the optimal choice for a is the posterior median: $P(\theta < a|\mathbf{x}) = P(\theta > a|\mathbf{x}) = 1/2$.

[5] i.i.d.: independent and identically distributed

Delta loss: $l(a|\theta) = -\delta(a-\theta)$, where any error is equally terrible. In this case, the optimal choice is the maximum of the posterior distribution: $a = \text{argmax}_\theta P(\theta|\mathbf{x})$.

The maximum of the posterior distribution is known as the Maximum a Posteriori (MAP) estimate and is the most widely used due to the (relative) simplicity of its calculation. In case the prior $P(\theta)$ is uniform (an uninformative prior, meaning there is no prior knowledge at all), then $P(\theta|\mathbf{x}) \propto \mathcal{L}(\mathbf{x}|\theta)$ and the problem is reduced to simply maximising the likelihood function. This is known as the *maximum likelihood* estimator.[6]

$$\hat{\theta}(\mathbf{x}) = \text{argmax}_\theta \mathcal{L}(\mathbf{x}|\theta) \tag{2.17}$$

This is the most widely used estimator in statistics. It is at the heart of commonly used methods such as the sample mean (Maximum Likelihood Estimator (MLE) for a scalar gaussian mean) or least squares regression (MLE for a gaussian mean which depends on some other variable).

2.3.1 Maximum likelihood estimation

To understand why the MLE is so widely used, it is useful to understand a few of its properties, applicable under the following assumptions [83]:

- The distributions $P(x|\theta)$ are distinct
- The distributions $P(x|\theta)$ have common support
- The observation set $\mathbf{x} = \{x_1, ..., x_n\}$ is composed of i.i.d. x_i with p.d.f. $P(x_i|\theta)$
- The parameter space Ω contains an open set ω of which θ_0 ('true' parameter value) is an interior point

Under these assumptions it can be shown that the probability that the likelihood function is higher at θ_0 than at any other θ goes to 1 as $n \to \infty$ [83]. Combining this result with the assumption that the likelihood is differentiable with respect to θ in ω, then the MLE $\hat{\theta}_n$ is guaranteed to exist and to converge to θ_0 in probability as $n \to \infty$. If, additionally, the likelihood has a unique minimum, the estimator is said to be *consistent*.

Assuming the third derivative of the likelihood is bounded, a consistent estimator will satisfy [83]

$$\sqrt{n}(\hat{\theta}_n - \theta_0) \to \mathcal{N}(0, \mathcal{F}^{-1}(\theta_0))$$

in probability, where $\mathcal{F}$ is the Fisher information (Sec. 2.2).

The MLE also has a geometric interpretation [82]. The log-likelihood function is given by $\log \mathcal{L}(\mathbf{x},\theta) = \sum_i^n \log P(x_i|\theta)$. This can be rewritten as a 'histogram' by creating a new

[6] In fact when calculations are done often $\log \mathcal{L}$ is maximised which produces the same result but simplifies calculation.

sequence $\mathbf{y} = \{y_1, .., y_k\}$ with only unique observation values, and an auxiliary sequence $\mathbf{n} = \{n_1, .., n_k\}$ with the number of occurrences. The likelihood can be rewritten as $\log \mathcal{L}(\mathbf{x}, \theta) = \sum_i^k n_i \log P(y_i|\theta)$. The average log-likelihood is now:

$$\frac{1}{n} \log \mathcal{L}(\mathbf{x}, \theta) = \sum_i^k \frac{n_i}{n} \log P(y_i|\theta) \tag{2.18}$$

The empirical distribution $\hat{P} = \frac{n_k}{n}$ appears in this equation. It is now easy to see that maximizing the log-likelihood is equivalent to minimizing $D(\hat{P}; P(y|\theta))$. Thus, the MLE can be interpreted as a search in model space for the distribution which is closest to the histogram of the results in the sense of the KL distance.

2.3.1.1 Cramér-Rao bound

An *unbiased* estimator has the property $\langle\hat{\theta}\rangle = \theta_0$. The Cramér-Rao bound shows that the variance of an unbiased estimator is bounded by the inverse Fisher information.

$$\mathrm{var}(\hat{\theta}) \geq \frac{1}{\mathcal{F}(\theta)} \tag{2.19}$$

An estimator that saturates this bound is said to be *efficient*. In particular, the MLE is efficient (even though not necessarily unbiased).

2.4 Relationship to statistical physics

2.4.1 Maximum entropy

The principle of maximum entropy has been introduced by Jaynes in 1957 [75, 76, 77] and has seen a resurgence in the past decade with increasing computational resources making bayesian inference more tractable. The basic problem is how to compute a probability distribution for the data without making any model specific assumptions. The solution proposed by the method of maximum entropy is to choose from among all possible probability distributions that agree with the data available the one which reflects maximum ignorance about everything else [78]. The mathematical measure for ignorance is the entropy, so the method entails using variational calculus to maximize the entropy while being subject to the constraints imposed by the available data.

Inspired by thermodynamics where the microscopic variables do not matter so much as the macroscopic variables (i.e. temperature, energy, magnetization), what will be considered here as a data constraint will be some function of all data points. Mathematically the probability distribution should be constrained to have specific expectation values. Consider that the data sets expectation values for a certain set of functions f^k. Then, $p(x)$

should be such that

$$\langle f^k \rangle = \int dx\; p(x) f^k(x)$$

for all k. [7] Then, it is possible to find p which minimizes the functional

$$M[p] = S(p)\; - \alpha \int dx\; p\; - \lambda_k \langle f^k \rangle$$

using variational calculus. The constraint with α is the normalization condition and S is the Shannon entropy – the notation change is employed to make the connection with statistical physics more evident.

Taking the variational derivative and setting it to zero results in the expression

$$\log p + 1 + \alpha + \lambda_k f^k = 0$$

the solution to which is

$$p = \frac{1}{Z} \exp\left(-\lambda_k f^k\right)$$

with

$$Z = \int dx\;\; \exp\left(-\lambda_k f^k\right)$$

the partition function (which comes from setting the normalization constraint). Now the remaining multipliers can be found by solving the system of equations

$$-\frac{\partial \log Z}{\partial \lambda_k} = \langle f^k \rangle$$

Setting no additional expectation values results in a uniform distribution, reflective of maximal ignorance. If the only two expectation values fixed are the data mean and variance the result is a gaussian distribution. The distribution here obtained can be used to describe arbitrary sets of data in a model free setting.

2.4.2 Thermodynamical connection

The above calculation is exactly parallel to the calculation one would do to calculate the partition function for the canonical ensemble in statistical physics. The analogy can be pursued further. Indeed, the analogy goes far beyond the maximum entropy setting and can be extended to any probability distribution. To develop the idea further, consider writing an arbitrary posterior distribution as a Boltzmann distribution:

$$P(\theta|d) = \frac{P(\theta, d)}{P(d)} = \left.\frac{e^{-\beta H(\theta,d)}}{Z}\right|_{\beta=1} \tag{2.20}$$

[7] Let's omit the notation (x) for simplicity.

with $H(\theta, d) = -\log P(\theta, d)/\beta$ and $Z = \int d\theta\ e^{-\beta H(\theta,d)}$. Note that the units are rescaled such that $k_B = 1$ and thus $\beta = 1/T$.

The energy is just the expectation value of the hamiltonian H [84]: [8]

$$E = \langle H \rangle = -\frac{\partial \log Z}{\partial \beta} \tag{2.21}$$

And the entropy is equal to:

$$S = -\int d\theta\ P(\theta|d) \log P(\theta|d) = \beta \langle H \rangle - \log Z \tag{2.22}$$

It is easy to verify that these relations are directly applicable to the above maximum entropy case with $\beta = 1$. The free energy is also easy to define as:

$$F = E - \frac{S}{\beta} = -\frac{1}{\beta} \log Z \tag{2.23}$$

Often Z isn't analytically accessible and it can be approximated via the saddle point (or Laplace's) approximation[9]:

$$Z = \int d\theta\ e^{-\beta H(\theta,d)} \simeq \sqrt{\frac{2\pi}{\beta |H''(\theta^*)|}} e^{-\beta H(\theta^*)} \tag{2.24}$$

where $|H''(\theta^*)|$ is the determinant of the hessian of the hamiltonian and θ^* is such that $H(\theta^*) = \min H(\theta)$. [10] Needless to say this approximation works best for small temperature ($\beta \to \infty$) which might not be close to the correct value at $\beta = 1$. θ^* is equivalent to the MAP estimate. Expectation values can also be approximated in a similar way:

$$\langle f(\theta) \rangle = \int d\theta\ f(\theta) P(\theta|d) \simeq \sqrt{\frac{2\pi}{\beta |H''(\theta^*)|}} f(\theta^*) P(\theta^*|d)$$

So the MAP estimate is defined as $\text{argmax}_\theta P(\theta|d)$. The result won't change by taking the log of the posterior, which leads to a form similar to the entropy:

$$\begin{aligned} \theta_{\text{MAP}} &= \text{argmax}_\theta(-\beta H - \log Z) && (2.25) \\ &= \text{argmax}_\theta(-2\beta H + S) && (2.26) \end{aligned}$$

For infinite temperature ($\beta = 0$) the parameters reflect total lack of knowledge: the entropy is maximized. As we lower the temperature, the energy term contributes more,

[8] Note that the expectation is taken with respect to $P(\theta|d)$.

[9] Usually it isn't even possible to calculate Z numerically for high dimensional problems due to the curse of dimensionality. Markov Chain Monte Carlo (MCMC) calculation is possible but expensive.

[10] Minimum because of the minus sign.

reflecting the information provided by the data, until at temperature zero we would only care about the data contribution and ignore the entropy term [11].

Another cool connection is the fact that the heat capacity is given by [85]:

$$\begin{aligned} C(\beta) &= \beta^2 \langle (\Delta H)^2 \rangle = \beta^2 \langle (H - \langle H \rangle)^2 \rangle \\ &= \beta^2 \frac{\partial^2 \log Z}{\partial \beta^2} = -\beta \frac{\partial S}{\partial \beta} \end{aligned} \tag{2.27}$$

The relation can be used [86] to estimate the entropy by calculating $\langle (\Delta H)^2 \rangle$ by MCMC for various betas and use:

$$S = \int_1^\infty d\beta \, \frac{1}{\beta} C(\beta) \tag{2.28}$$

2.4.3 Markov chain methods

Algorithms developed originally for statistical physics can also be adapted to the probabilistic framework with great success. Suppose our goal is to calculate an expectation $\langle f \rangle$ numerically. It can be approximated by

$$\langle f \rangle = \frac{1}{N} \sum_i^N f(\theta_i)$$

if we have access to samples θ_i from Eq. (2.20).

The easiest would be to do rejection sampling. Draw a random number θ uniformly and reject it with probability $P(\theta|d)$. For high dimensional systems, this would require an astronomical number of samples, so we need to make sure we draw samples likely to get accepted. To do so, it is possible to employ a Markov chain: a stochastic process which moves from one state of the system (i.e. a given configuration of variables) to another arbitrary state with some probability. This method is known as Markov Chain Monte Carlo (MCMC).

Intuitively, if we set up the Markov chain so that it moves preferentially to states close to the one we were in, they are more likely to get accepted. On the other hand, the samples will be correlated, which means we cannot draw all samples generated by the chain, but need to wait some time after drawing a new independent sample (given by the autocorrelation of the chain).

The Markov chain needs to obey certain principles [85]: ergodicity, which means the chain is able to reach all states in the system from any initial state (to guarantee the probability distribution is represented correctly); and detailed balance, which means the probability of going from state A to state B is the same as that of from state B to A, for

[11] This is also the basic idea for the simulated annealing optimization algorithm, where in that case the objective function plays the role of the energy and the algorithm walks around phase space randomly, with jump size proportional to the temperature. The annealing schedule progressively lowers the temperature, restricting the random walk to regions of high objective function value, until it freezes at some point.

all pairs of states in the system.[12] This makes sure the chain doesn't get stuck in a loop where it goes from A to B to C and back to A. Mathematically, the probabilities must obey

$$\frac{p_{A\to B}}{p_{B\to A}} = e^{-\beta(E_B - E_A)}$$

so that the stationary distribution of the Markov chain matches the Boltzmann distribution from Eq. (2.20).

Now, any Markov chain with those properties will converge to the required distribution [85], but we still haven't decided on a concrete transition rule $p_{A\to B}$. The clever part about metropolis hastings comes now. Once a new state B has been proposed, we can actually choose to not transition to it, and instead stay where we are without violating detailed balance. To do so, we define the acceptance ratio A and the proposal distribution g.

The two of them must be balanced such that

$$\frac{p_{A\to B}}{p_{B\to A}} = \frac{g_{A\to B} A_{A\to B}}{g_{B\to A} A_{B\to A}}$$

We can choose a symmetric g for simplicity, such that $g_{A\to B} = g_{B\to A}$, and thus g cancels out. Then

$$\frac{A_{A\to B}}{A_{B\to A}} = e^{-\beta(E_B - E_A)}$$

Now what we want is to accept as many moves as possible, so we can set one of the A's to 1 and the other to the value on the right hand side of the equation. Because the most likely states of the system are the ones with low energy, we generally want to move in that direction. Thus we choose the A's to follow the rule

$$A_{A\to B} = \begin{cases} e^{-\beta(E_B - E_A)} & \text{if } E_B - E_A > 0 \\ 1 & \text{otherwise} \end{cases}$$

Again, this rule actually applies to any probability distribution, since you can go back and forth from the Boltzmann form to an arbitrary distribution.

2.5 Stochastic Processes

Once we introduce temporal dynamics into a model, its probabilistic description becomes slightly more complicated. One could argue that time is just another dimension in the state space of our random variables. This view, while in principle correct, is naive. Time is a privileged dimension in the sense that it inexorably runs in a single direction, which means that it is easier to consider it separately.

Assume the state of our system is described by an N-dimensional random variable X_t at time t. Under the Markov assumption, there exists a well-defined probability for the time

[12] A state is equivalent to a particular realization of θ. So let's introduce the notation $E_A \equiv H(\theta_A, d)$ for comparison with Eq. (2.20).

evolution of the system: $P(x_t|x_{t-1})$. This transition probability is extracted from physical considerations, and is what usually makes up our physical model of the system. Over several time-steps, there are multiple ways to reach one state from an initial condition. The **Chapman-Kolmogorov relation** formalizes this statement [87]:

$$P(x_{t+1}|x_{t-1}) = \int dx_t P(x_{t+1}|x_t)P(x_t|x_{t-1}) \tag{2.29}$$

Assume the time difference is very small. Then, we can convert the Chapman-Kolmogorov relation from an integral formulation to a differential formulation [88]:

$$\begin{aligned}\frac{dP(x_t)}{dt} = &- \sum_i \frac{\partial}{\partial x_i}\left[A_i(x)P(x)\right] + \frac{1}{2}\sum_{i,j}\frac{\partial^2}{\partial x_i \partial x_j}\left[B_{i,j}(x)P(x)\right] \\ &+ \int dx' P(x|x')P(x') - P(x'|x)P(x)\end{aligned} \tag{2.30}$$

Where the subscript t was omitted throughout the left hand side for simplicity of notation. The three terms have simple interpretations: they correspond respectively to drift, diffusion and jump processes. Often drift and diffusion processes are treated separately from jump processes.

For a pure jump process, the differential Chapman-Kolmogorov equation reduces to the **master equation** [87]:

$$\frac{dP(x_t)}{dt} = \int dx' P(x|x')P(x') - P(x'|x)P(x) \tag{2.31}$$

In this form it is clear that the probability evolution consists of a balance between inward and outward probability flows. For a discrete system the integral is replaced by a summation and the transition probabilities form a matrix, the *Markov matrix*.

In the case of a drift and diffusion process, we have what is known as a Fokker-Plank Equation (FPE) [88]:

$$\frac{dP(x_t)}{dt} = - \sum_i \frac{\partial}{\partial x_i}\left[A_i(x)P(x)\right] + \frac{1}{2}\sum_{i,j}\frac{\partial^2}{\partial x_i \partial x_j}\left[B_{i,j}(x)P(x)\right] \tag{2.32}$$

The Fokker-Plank equation can be represented in a different formalism, as a Stochastic Differential Equation (SDE) [88]:

$$dx = A(x,t)dt + \sqrt{B(x,t)}d\mathbf{W}(t) \tag{2.33}$$

Where $d\mathbf{W}(t)$ represents a Wiener process. [13] $\sqrt{B}$ is well defined since B must be positive semidefinite. This form has been particularly successful in more pure mathematics oriented fields, and is also very convenient for numerical simulation [89].

Of particular importance to systems biology are expansion methods to approximate jump processes by drift and diffusion processes, since the latter are more directly comparable to macroscopic observables than the former [90]. The most popular methods are the *Kramers-Moyal* expansion [88] and van Kampen's Ω expansion [87].

Another widely used approximation is the small noise approximation, corresponding to a near-deterministic limit. In this case, the FPE is expanded by the use of scaled variables. The final equation is not particularly helpful *per se*, but it allows one to extract the moments perturbatively. Taking only the first term in the series, one recovers the *linear noise approximation.* This is a Gaussian approximation, since only the mean and covariance are considered. Their evolution will be given by:

$$\frac{dm}{dt} = A(x,t)m(t) \tag{2.34}$$

$$\frac{dC}{dt} = C(t)A^T(x,t) + A(x,t)C(t) + B(x,t) \tag{2.35}$$

With m is the mean and C is the covariance matrix of the approximate distribution for x. This topic will be further elaborated upon in Chapter 6.

2.6 Kolmogorov complexity

Suppose I want to send out a specific message. What is the best encoding strategy to send out this message as compactly as possible? This issue is not solved in general by the information theoretical concepts discussed above. Indeed, it can be proved that this issue cannot be solved in general at all. But a lot can be said about the compressibility of a message and these concepts will have particular importance for certain parts of this thesis.

Let's delve into this topic by first considering a compression heuristic provided by information theory. Consider the message X defined by a number of symbols x_i with $i \in [1, N]$ which can take values from an alphabet $\mathcal{A}$. The obvious way to send out this message is to encode each element of $\mathcal{A}$ as a binary number, and send out the resulting numbers. The message will thus have a size of $N \log_2(S)$, with S the size of the alphabet.

But we do not expect each symbol to appear equally frequently in the message. If we encode the most frequent symbols as a shorter binary string, and the less common symbols as a longer binary string, a long message can be made as small as $N\,H(X)$[14]. This limit can be reached by applying a simple procedure, known as *Huffman coding* [81].

Now suppose that the symbols are correlated in some way. We could now improve the amount of compression by choosing to encode *blocks* of symbols. Those blocks can then be coded with Huffman coding. This method would miss out on correlations within blocks.

[13] A Wiener process is a continuous, zero-mean stochastic process with Gaussian distributed increments.

[14] Note that for a uniform distribution this result reduces to the naive strategy.

We could increase the size of the block, but at some point the block itself will be the size of the message and the compressibility is gone, so there will be an optimal block size. There are many other methods to improve upon this procedure, but they all rely on the same basic core principle: to reduce redundancy in the data [91].

So far, the concept appears straightforward: inspect the data and look for some kind of redundancy. The algorithm which can eliminate the most redundancy from the data should be optimal. However, consider the following two (abbreviated) strings:

$$00100100001111110110101010001000100001011010001100001000\ldots$$
$$0101101110100111011101101001110010110001001100101111010\ldots$$

Both of these strings contain absolutely no redundancy from a statistical point of view (for large N). However, the first is eminently more compressible than the second. To understand why, let us introduce first the concept of a *universal Turing machine*. A **Turing machine** is a hypothetical construct consisting of [92]:

Tape: The tape is of infinite length and divided into cells, each of which contains a discrete symbol drawn from a finite alphabet.

Head: The head reads the currently considered cell and can move the tape left or right one cell at a time.

Finite control: The finite control can be in one of a number of states. At each time step, it can perform one of the following actions[15]:

1. Write a symbol on the current cell;
2. Shift the head to the left or right.

Finally, it moves its current state to a new state. It is possible that the device performs no operation, in which case it is said to *halt*. Its choice depends on the combination of its current state and the current symbol read from the tape. The finite control is known as a *finite state machine*.

Turing proved that this machine can compute *any* computable sequence given a tape of arbitrary length and appropriately defined finite control [93]. There are infinitely many machines which can emulate the behavior of a Turing machine, known as universal Turing machines (i.e. a modern computer architecture) [81]. While physical implementations of these machines have finite memory in practice, they are able to to perform any computation requiring a tape length up to their memory size. Therefore while the Turing machine model was only created as an abstract model of computation and what is computable, in practical terms it is also a useful model for real machines.

Going back to the strings, suppose I want to send a message composed of the first string. This binary string actually corresponds to the fractional part of π expressed in

[15] There are many formally equivalent definitions of a Turing machine, here we follow [92].

binary. I could write a program which computes π and have it display the first N bits of the binary representation of its fractional part. The recipient could run this program on a universal Turing machine and get the answer. For large N the length of this program will be much smaller than any other code for the string. The length of this program is known as the **Kolmogorov complexity** of the string [81]:

$$K_{\mathcal{U}}(X) = \min_{p:\mathcal{U}(p)=X} l(p) \tag{2.36}$$

Where p is a program which computes X on the universal Turing machine $\mathcal{U}$ and l is its length. Note that because $\mathcal{U}$ is universal, it can simulate any other computer $\mathcal{V}$ given some emulation code of length $c_{\mathcal{V}}$. Thus, $K_{\mathcal{U}}(X) \leq K_{\mathcal{V}}(X) + c_{\mathcal{V}}$. So we can't beat the limit set by the Kolmogorov complexity by constructing a very specific computer for some specific string since the overhead will show up in the emulation constant.

The second string is the result of a natural random number generator. Aside from a vanishingly unlikely coincidence, there is no equivalent program which can reproduce this string and therefore it would not be possible to transmit it in much less than N bits (here $S = 2$).

It is simple to understand why most random strings cannot be compressed in this computational way. There are only 2^k programs with length k, and there are 2^N strings with length N. So for a large N only a vanishingly small proportion of the strings will be compressible to length k. Note that this framework is a superset of the information theoretic data compression framework, since all regularities in the strings can be described as programs.

Unfortunately, the **Kolmogorov complexity** is not computable. The best that can be done is to bound it from above by finding a program of length k which can reproduce the string.

A theory of learning has been based upon these ideas. The *minimum description length* principle states that given some data, the shortest hypothesis (in terms of code length) consistent with the data is to be preferred [94, 92]. Since the data has been compressed, it is more likely that future instances of the data will still be well described by the model.

2.7 Mathematical modeling of regulatory networks

A regulatory network is a nonlinear dynamical system composed of a number of variables. The interactions between these variables can be systematised as a network structure, where each dynamic variable is represented by a node and each interaction by an edge. This picture necessarily entails some simplification of reality, since two variables may act in a non-linear, combinatorial way to regulate a third; whereas the network picture implies some sort of independence. The network picture should only be taken seriously as a causality description, where a directed edge merely implies a causal relation between two variables [95].

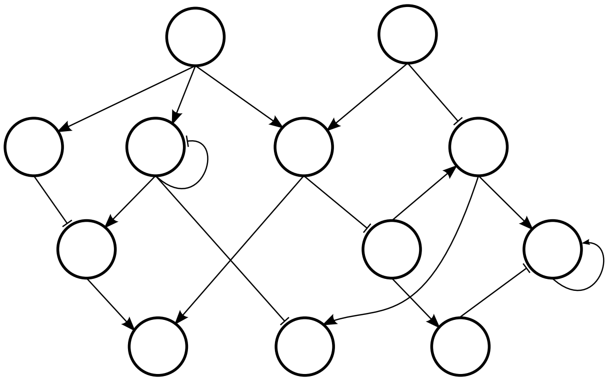


Figure 2.2: Abstract representation of a gene regulatory network. In this framework, a protein (coded by a gene) is represented by a node in the network. The network's edges are specified by the nonzero values of the connection matrix A. These values are determined by the protein's interactions: position a_{ij} will be nonzero if the ith protein acts as a TF for the jth protein with $a_{ij} \geq 0$ (triangular arrows) if it enhances expression and $a_{ij} \leq 0$ (barred arrows) if it inhibits expression. This connection matrix maps directly to a linearized network (cf. (2.46)).

2.7.1 Simple regulatory models

The interactions are modeled by chemical reactions. Following the picture of the central dogma of biology (Fig. 1.2), the most basic regulatory interaction is the basal expression of a protein. Given some degradation constants for mRNA and protein, this will result in a constant steady-state value for the protein.

$$\text{gene} \xrightarrow{\alpha} \text{mRNA} \xrightarrow{\beta} \text{Protein} \tag{2.37}$$

$$\text{mRNA} \xrightarrow{\lambda_m} \emptyset \tag{2.38}$$

$$\text{Protein} \xrightarrow{\lambda_p} \emptyset \tag{2.39}$$

How can we model these chemical reactions? Given the complexity of the systems under consideration, one must choose the relevant time and length scales of the phenomena under consideration and apply some coarse graining procedure to any physical dynamics below these scales.

At the lowest level, it is possible to simulate each atomic interaction (approximating Schrödinger's equation) using molecular dynamics equations. The increasing computational power available has enabled simulations at the scale of 10^6 atoms, including a recent simulation of a whole virus [96, 97].

Looking at the level of interacting chemical reactions, we can simplify away individual atomic interactions and consider only interactions between each individual chemical species (which may contain anywhere from 10^2 to 10^5 atoms). Such a description is made quantitative by the chemical master equation (CME, Eq. (2.31)), which is in general not analytically tractable and thus needs to be simulated numerically [90].

In the case of large reaction networks, it is often computationally more practical to approximate the quantity of each chemical species as a real-valued concentration, in which case the system is described by a set of (stochastic) ODEs [90, 98]. The majority of the following discussion will be developed within the ODE framework.

This model entails a markovian assumption, [16] implying that the intrinsic processes of transcription and translation occur at a much faster timescale than protein and mRNA dynamics. It must be be noted that such an assumption, while necessary, may not be entirely accurate: the time necessary for mRNA to cross the nuclear membrane and being transported into the cytoplasm, where it must diffuse until coming in contact with a ribosome, is not negligible.

2.7.1.1 Stochastic modeling

Due to the molecular nature of these pathways, noise is an inescapable reality which should be reflected in our modeling [99]. Efforts have been made to quantitatively model the noise in gene expression.

The noise is not just a 'blurring' of trajectory spaces but it can lead to quantitative differences in behavior, such as heterogeneity in gene expression [100, 101], used in bet-hedging strategies as discussed above. Other uses are noise-induced oscillations [102] or even reduction of fluctuations [103] (as counter-intuitive it might seem).

The stochastic analog of the ODE model is called the Langevin equation [99]:

$$\frac{dx}{dt} = f(x) + h(x)\eta \tag{2.40}$$

where $f(x)$ is a continuous nonlinear function $h(x)$ has a functional form approximating the noise in the system. η denotes white noise with zero mean and unit variance. In Figure 2.3 we can see a numerical simulation of a Langevin equation which produces heterogeneity as a result.

Intuitively, there are two broad categories of fluctuations in regulatory networks: intrinsic noise, caused by the brownian diffusion of molecules [104]; and extrinsic noise, caused by the variability in the parameters controlling the process [105, 106]. Experimentally, these sources can be distinguished by the use of dual reporter systems [107, 108]. These processes appear to dominate at different expression levels [109]: for low expression levels, intrinsic noise dominates; for high expression levels, extrinsic noise does.

Using a two-step model it is possible to obtain analytical expressions for intrinsic noise in gene expression [110, 111]. At steady state, the noise strength is given by

$$\frac{\sigma^2}{\langle p \rangle} = 1 + b$$

which only depends on the *burst size* b, the average number of proteins synthesised by mRNA transcript.

[16] In a markovian system only concentration values at current t matter, and not the past history

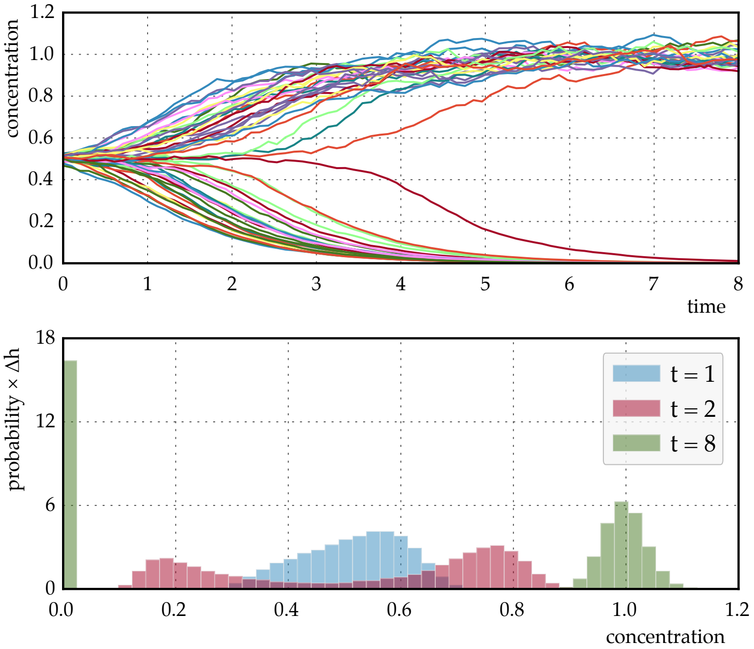


Figure 2.3: Heterogeneity in stochastic dynamics. Top, sample paths from the langevin equation Eq. (2.40) for $f = \frac{x^n}{x^n+K^n}$ with $n = 6$, $K = 0.5$ and $g = x$. Bottom, histogram with bin widths $\delta h = 0.024$ illustrating bistability in the distribution of protein concentrations. Stochastic integration code used to generate this picture is reproduced in B.1.

Extrinsic noise is harder to model, as its source can be any variation in the parameters which determine the functions f and h [107]. A general way to model gene expression distributions in the high expression level regime is to invoke a central limit theorem argument and describe the data as a lognormal distribution [112, 113, 114]. An even simpler argument is based on considering a langevin equation with multiplicative noise, which can be shown analytically to be lognormally distributed [88]. Indeed, experimental data appears to corroborate this approximation [115]. Later in the thesis we will explore several datasets which also follow this distribution.

2.7.1.2 Gene regulatory network dynamics

For a gene regulatory system with N proteins the dynamics describing their evolution will require at least $2N$ equations, corresponding to the transcription and translation steps. In general they are given by the following set of differential equations:

$$\dot{m}_i(t) = f_i(n(t)) + g_i(n(t))\,\eta_i(t) \tag{2.41}$$

$$\dot{n}_i(t) = r_i(m(t)) + h_i(m(t))\,\eta_i(t) \tag{2.42}$$

where m is a vector with mRNA concentration, n is a vector with protein concentration and η is a vector with white noise realizations. The function f must be determined from

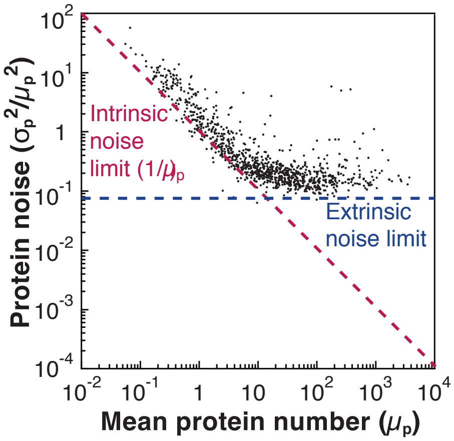


Figure 2.4: Protein noise as a function of mean for 1018 proteins in the *E. Coli* proteome measured using single-molecule fluorescence microscopy. Proteins with low expression levels hit the intrinsic noise limit, while high protein numbers are limited by extrinsic noise. Adapted from *Taniguchi et al.* [109].

biological considerations. Usually the function representing translation (Eq. (2.42)) is just a linear proportionality constant, and thus does not significantly affect the network dynamics [17]. Therefore in many cases the mRNA variables are left out altogether (i.e. a one-step model), and we consider only a single equation for protein evolution in time, with the function f describing the nonlinear regulatory interactions.

2.7.1.3 Hill models

Combinatorial gene regulation in prokaryotes can be quantitatively described using models based on statistical mechanics [116]. The basic idea is to assume transcription activity is proportional to the fraction of promotors occupied by TFs. Then, if TF concentration is denoted by [TF] and promotor effective concentration by [Pr], and we assume cooperative binding of n TFs to a promotor, we have the chemical reaction $[Pr] + n[TF] \rightleftharpoons [PrTF]$ and the activity α is described by:

$$\alpha \propto \frac{[PrTF]}{[Pr] + [PrTF]}$$

By the law of mass action, we have $K' = [PrTF]/[Pr][TF]^n$ which we can replace above to obtain an expression which only depends on K and $[TF]$. This results in the well known hill equation:

$$h = \frac{[TF]^n}{K^n + [TF]^n} \tag{2.43}$$

[17] A significant exception is when a certain time lag between events is important, since the mRNA → protein dynamics introduce a delay between regulation and its effect on protein concentration. In that case one must choose between keeping the mRNA variables or considering a non-markovian process.

This corresponds to an activator. Conversely, transcriptional activity may be inversely proportional to the fraction of promotors occupied by TFs. In that case we would have

$$\alpha \propto \frac{[Pr]}{[Pr] + [PrTF]}$$

This leads to the hill equation for a repressor: [18]

$$h = \frac{K^n}{K^n + [TF]^n} \tag{2.44}$$

The transcription activity α is proportional to a given average number of mRNAs in the cytoplasm which in turn will, as outlined in the first part of this introduction, be translated by ribosomes into the target protein. Therefore, eq. (2.43) will (within a multiplicative factor) describe the fold change in protein concentration for that gene.

It is possible to describe the two parameters – n and K – in terms of their effect on the shape of the hill function. The cooperativity n describes how multiple TFs can have a synergistic effect on the RNAp binding probability. Abstractly, its value is proportional to the smoothness of the transition between a repressed state and an activated state ($n = \infty$ would describe a digital function perfectly switching at K).

The parameter K roughly corresponds to the TF concentration threshold at which we switch from a regime of low expression to one of high expression. K corresponds to an effective microscopic dissociation constant, and is some function of the effective binding energies of the TFs and the RNAp to the DNA. This is tuned by evolution to some precision, as it depends not only on the TF/RNAp molecules themselves, but also on the DNA code at each binding site and adjoining sites and the spatial structure of the DNA.

2.7.2 Towards modeling complex networks

2.7.2.1 Combinatorial logic

In the above discussion, it was assumed that gene regulation was performed by a single chemical species. In case there are multiple proteins acting as TF for a single gene, the situation is harder to derive analytically. In the case of bacteria, statistical mechanics have been used to derive a set of combinatorial logic functions describing the effective regulation for different interactions [117, 118].

In eukaryotes more complex proteins, allosteric interactions and chromatin structure complicate analysis. [19] How to describe combinatorial effects in gene regulation in general remains an open research question [120].

[18] Note that abstractly we can convert an activator into a repressor by setting n negative

[19] One proposed model to partially account for these effects is the Monod-Wyman-Changeux (MWC) model [119].

2.7.2.2 Linearization

One way to solve the combinatorial conundrum is to ignore the nonlinear behavior of f. We can then taylor expand the function f and linearize the equations:

$$f_i(n(t)) \simeq f_i(0) + f_i'(0) \cdot n(t) + \mathcal{O}(n^2) \tag{2.45}$$

The linearized set of differential equations for all proteins can be written as

$$\dot{n}(t) = An(t) + b + \eta(t) \tag{2.46}$$

where b is the basal expression rate (rate of gene expression with no stimulus) and A is a matrix summarizing the responses of each gene to its stimulus. Note that diagonal elements will not be zero as one must at least take into account protein degradation, which will be represented by a term $-\lambda_i$. The individual elements of A represent the different interactions in the network: a 0 representing no interaction; a positive term activation; and a negative term repression.

In many cases, this linear representation of the regulatory network suffices to derive some insight.

2.7.2.3 Equivalence to neural network models

Parallels can be drawn between the analog processing done by gene regulatory networks and neuronal networks. While the physical constraints are vastly different in terms of noise, dynamic range and timescale; [20] abstract models show parallels in terms of the basic ingredients necessary to model nontrivial computations:

- Output is the outcome of a saturating nonlinearity
- Layered network structure
- Cyclical connectivity graph – leading to recurrence and memory
- Distributed representation – the network is not modular

In fact the hill function can be reduced to a logistic sigmoidal function by applying the transformation $y = \log x$ (with $[TF] \equiv x$ in (2.43)):

$$h_s(y) = \frac{1}{1 + e^{n(\log K - y)}} \tag{2.47}$$

The analogy bears insights: we know from the theory of neural networks the conditions necessary for universal computation [122]. The field of neuroscience has decades of experience analyzing neural networks from an information theoretic perspective – in the future it may be possible to leverage this body of work to explain molecular phenomena [79].

[20] Interestingly the issue with combinatorial integration of inputs persists: it is known dendritic trees do more than linearly summing their inputs [121].

Chapter 3

Information processing in fate determination

3.1 Motivation

In 1969 Wolpert introduced the concept of positional information [62]. Under this paradigm, in order for cells to differentiate and acquire specific fates they sense the concentration of a specific chemical, a *morphogen* [123]. Different fates are encoded in the morphogen spatial profile in a concentration-dependent manner [124]. We will henceforth denote the average value of the morphogen profile as[1]

$$\mu(x) \quad \text{with} \quad 0 < x < L \, . \tag{3.1}$$

We have already seen, however, that gene regulation and expression are noisy processes[2]. How can cells reliably read out a noisy morphogen concentration value and robustly acquire the correct fate in the correct position in an organism?

This question has been particularly well studied in the *Drosophila* embryos for the Bcd transcription factor [125, 126, 60, 127]. Bcd acts as a primary regulator for downstream gap genes and pair rule genes [128] along the Anteroposterior (AP) axis. While it has been shown that Bcd precision is theoretically enough to specify cell fates with a resolution of 1 cell along the AP axis [126]; later stage morphological markers such as the cephalic furrow exhibit a robustness in their positional accuracy not completely explained by Bcd readout alone [57].

[1] When appropriate embryo length will be rescaled such that $L = 1$.

[2] Refer to 2.7.1.1 for more.

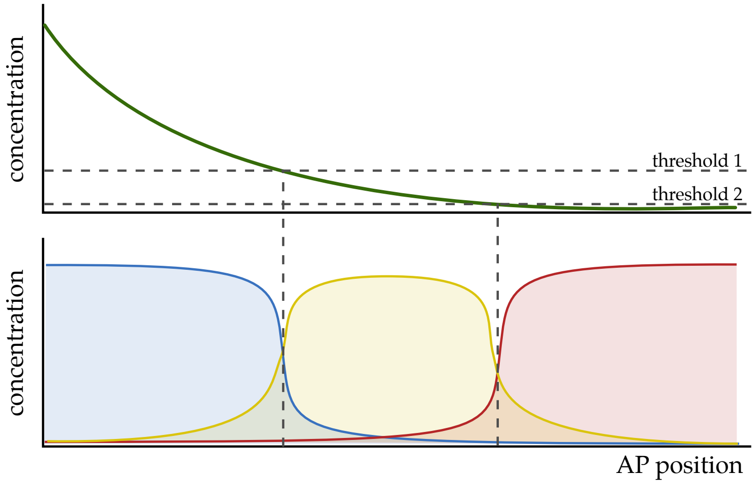


Figure 3.1: Positional information was illustrated by Wolpert [62] with the French Flag analogy. A downstream target gene is expressed when the upstream maternal morphogen concentration is within a certain concentration range. In this way concentration thresholds implicitly define a spatial position at which the boundaries between expression/no expression are located.

3.2 Previous work

Initial work regarding positional information[3] in the Bcd system focused on the careful quantification of the Bcd spatial profile along the AP axis [64, 129]. By carefully measuring noise in expression, one can recover a rough lower bound on positional error estimation via Bcd readout using error propagation.

Simultaneously the question of how the Bcd profile is formed came to prominence. To first order it appears to be an exponentially decaying profile, which hints towards a diffusive process as its origin. Indeed, the leading model is the SDD model [130, 60, 131, 127, 132]: constant anterior protein *synthesis* generates Bcd protein which *diffuses* out along the AP axis and *degrades* uniformly.

Tostevin et al.[133] used the SDD model to compute theoretical upper bounds on positional information

$$\frac{d\rho}{dt} = D\nabla\rho - \lambda\rho + J\delta(x) \tag{3.2}$$

where λ is a degradation rate and J is a production rate. By assuming that the number of proteins in a given space region is a Poissonian distributed random variable they determine

[3] Here, the term positional error is used to denote the standard deviation of an estimator of the position along the AP axis given some concentration measurement. Mathematically, if P(x—c) represents the posterior distribution for the position x given one or more measurements c and if we assume that its expected value corresponds to the "true" value $\langle x\rangle_{P(x|c)} = x_t$, then positional information corresponds to $\sqrt{\langle(x-x_t)^2\rangle_{P(x|c)}}$. The terms positional information and positional accuracy will be used interchangeably to denote the inverse of positional error.

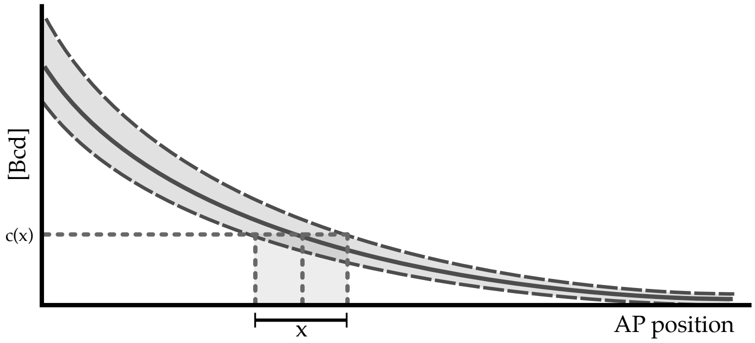


Figure 3.2: Illustration of how a given concentration can be read for a range of positions in a noisy morphogen profile. A lower bound on positional error can be estimated to first order using the error propagation formula $\sigma_x = \left|\frac{d\mu}{dx}\right|^{-1} \sigma_\mu$

readout error as a function of AP position. An upper bound on positional information can then be computed taking into account the possibility of time averaging and multiple readouts. In related work, it has also been hypothesized that pre steady state Bcd readouts can increase the amount of information extracted from the profile [134, 135].

Taking a different perspective, one can consider the SDD model fixed but let its parameters vary. They can then be optimized so that the resulting profile can buffer against intrinsic and extrinsic noise in the system [136]. Under this framework, it has been argued that the main source of positional error is embryo-to-embryo variability [137] (i.e. extrinsic noise).

Diffusion also appears to play a role in increasing positional information by washing out noise arising from the bursting of gene expression [138]. Attempts have been made to theoretically derive an optimal diffusion constant given certain constraints, such as minimizing the amount of morphogen produced for a target positional information amount [139].

An interesting development is the use of mutual information to quantify positional information. In this case, it is not defined in terms of the accuracy of a position estimator, as all previous work, but rather in terms of how much information is conveyed to a target downstream gene [140, 141, 142, 69, 143]. Networks that optimize information transmission can be obtained in this framework, by choosing topologies that maximize mutual information between input and output [144].

Positional information has also been quantified in general as the accuracy of an unbiased estimator via the Fisher Information (Eq. (2.12)) [145, 146]. In this picture, the estimator is formally a map between concentration space (defined by the different morphogens in the system) and physical space. The fisher information can then be used to quantify how a distortion in concentration space corresponds to a change in physical coordinates.

An important question which has only partially been addressed is how robust is positional decoding to changes in concentration. It has been shown that changes to Bcd dosage in the *Drosophila* embryo cause shifts in downstream targets such as Hunchback [64] or the cephalic furrow [57]. Nonetheless, these shifts do not have a proportional response to Bcd concentration changes, implying that Bcd is not the sole source of positional information.

A question related to robustness is that of embryo-to-embryo scale fluctuations, also known as the scaling problem. In a naive setup, where neither the morphogen profile nor the measuring mechanism have any knowledge of of the embryo length, the positional accuracy of the final readout will be degraded. Some authors have not clearly distinguished these two concepts in the literature so it bears reinforcing that one can have good positional accuracy even with a non-scaling profile, as it is theoretically possible to factor out scaling variations via appropriate compensatory mechanisms. Since it has been established that downstream features are correctly positioned in scaled coordinates[64, 147] (i.e. their position scales with embryo length), this robustness must be derived from one of two mechanisms:

- The Bcd gradient scales with the embryo length, such that $c(x/L)$ is independent of L, with L the embryo length, c the Bcd concentration and x the absolute position in the AP axis. In this case we can have a naive readout mechanism.

- The Bcd gradient does not scale with the embryo length, but the readout mechanism can compensate for length and appropriately correct its output.

Some groups have argued that the Bcd gradient scales with embryo length [148, 149, 150]. The proposed mechanisms to achieve this scaling range from cytoplasmic transport [151] to volume-dependent Bcd mRNA deposition [150]. Other groups have evidence for partial scaling (scaling only in the anterior part of the embryo, which is also consistent with a non-scaling model pinned at the anterior end) [152].

Measuring the scaling properties of Bcd gradient is challenging as natural embryos only exhibit a 4% variation in egg length. Most of the above cited studies used artificial selection to create large and small egg lines from which to extract profiles with significant length variance. However, it has recently been shown that such artificial selection processes strongly perturb the whole development process [153], which implies that the conclusions drawn from data analysis in such lines cannot be trivially brought forward to the wild type lines.

As for mechanisms allowing scaling in case the Bcd gradient does not scale, the simplest is to allow for a symmetrically opposite gradient pinned at the posterior side of the embryo (assuming the Bcd gradient is pinned at the anterior) [129]. Other methods involve the use of an intermediary protein released from both ends of the embryo [154, 155] or cross-repressions in the downstream gap genes[4] as an error correction mechanism [156].

3.3 Results

3.3.1 Optimizing morphogen profiles for positional information

The morphogen concentration value measured at each position is a random variable m distributed according to a probability density which depends on the physical distribution

[4] The interesting dynamics of the gap gene system will be addressed in the next chapter.

of morphogen molecules. One parametrization of this distribution $P(m|x)$ is a simple gaussian:

$$P(m|x) = \frac{1}{\sqrt{2\pi\sigma^2(x)}} \exp\left(-\frac{(m-\mu(x))^2}{2\,\sigma^2(x)}\right) \tag{3.3}$$

This form is consistent with a maximum entropy principle [75], since experimental data only provides accurate information about the first two moments of this distribution [126].

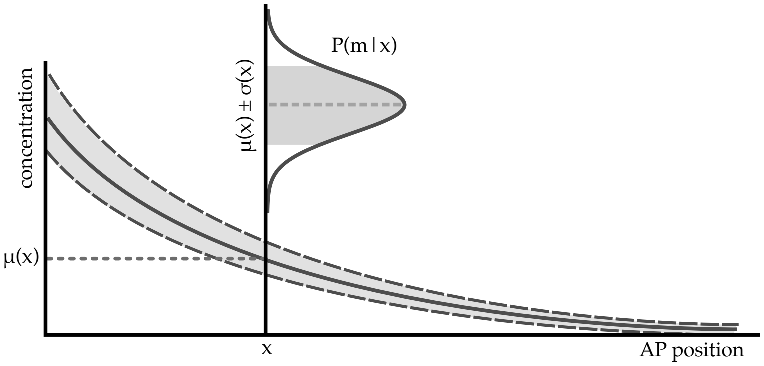


Figure 3.3: The conditional distribution of concentration m given some position x is denoted by $P(m|x)$ and is given by (3.3). It is defined by the average value $\mu(x)$ and the standard deviation $\sigma(x)$

As in previous work, we will take the FI as a measure of positional information (previously discussed in 2.2). Under the gaussian model, the FI can be explicitly calculated as:

$$\mathcal{I}^{-1}(x) = \frac{2\,(\sigma^2)^2}{2\,(\partial_x\mu)^2\sigma^2 + (\partial_x\sigma^2)^2} \tag{3.4}$$

It is clear from (3.4) that when the noise is not spatially dependent, the FI directly corresponds to the error propagation expression. We can interpret the additional term corresponding to the position dependence of the noise as an additional contribution to positional information by taking into account differences in the variance of the distribution of m at different positions, which can be used to constrain the output of an estimator.

Using (3.4) it is possible to quantify positional error in quantitative data of morphogen profiles. Given the discrete nature of experimental data, the data must first be fit to a cubic interpolating spline so that derivatives can be taken. Using the MLE method, two splines corresponding to $\mu(x)$ and $\sigma^2(x)$ were fit such that the likelihood of the model (3.3) was maximized. Results for Bcd and Dorsal profiles are given in Fig. 3.4.

Optimizing positional information Given the easily accessible analytic form of the FI for the Gaussian model, we can ask the question of which average profile $\mu(x)$ minimizes

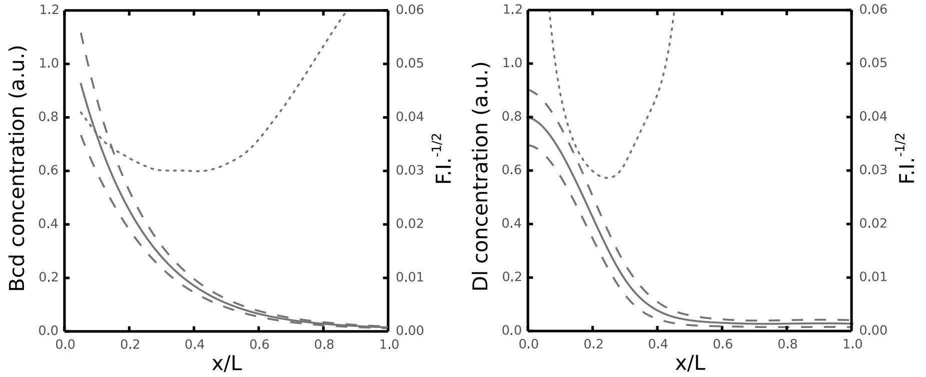


Figure 3.4: Inference of the statistical distribution $P(m|x)$ for the local Bcd concentration m at position x from the data of Gregor *et al.* [57] (left) and for Dorsal from Shvartsman et al [157] (right). The mean profile $\mu(x)$ (solid line) and the standard deviation $\pm\sigma(x)$ (dashed line) are inferred from the experimental data via maximum likelihood estimation based on Eq.3.3 as a function of normalized anteroposterior axis position. The dashed bars are $\pm\sigma_\mu$. Overlaid (dotted) is the calculated minimal σ_x by saturating the Cramér-Rao bound 2.19 using expression 3.4.

the positional error in some spatial range $[a, b]$. To answer this question we can use the language of variational calculus [158], by defining a cost functional:

$$\mathcal{J}[\mu(x)] = \int_a^b dx \, \frac{1}{\mathcal{I}(x)} \tag{3.5}$$

The logic for choosing this functional is that to minimize the error of an unbiased estimator, we need to first saturate the Cramér-Rao bound (defined in 2.3.1.1). Assuming this condition is fulfilled, then the error of an estimator is given by $\mathcal{I}^{-1}$.

This functional does not depend exclusively on the average profile $\mu(x)$ but also on the noise profile σ^2. Given the prior discussion of intrinsic and extrinsic noise in 2.7.1.1 we know that the noise profile often depends on the mean value. Therefore σ^2 can be parametrized as a function of the average profile:

$$\sigma^2(x) = \sigma_0 \cdot \left[\mu(x) + p\,\mu^2(x)\right] \equiv \varsigma \tag{3.6}$$

Here, σ_0 parameterizes absolute noise strength and p quantifies relative weight of extrinsic versus intrinsic noise.

Let us define[5]

$$\mathcal{I} = \frac{\mu'^2(2\varsigma + (\partial_\mu \varsigma)^2)}{2\varsigma^2} = \mu'^2 f(\mu) \; . \tag{3.7}$$

[5] Note that the explicit x dependency is dropped for notational convenience and therefore $\mu' \equiv \frac{d\mu(x)}{dx}$

We can now solve the euler lagrange equations for $\mathcal{L} = 1/\mathcal{I}$:

$$\frac{d\mathcal{L}}{d\mu} - \frac{d}{dx}\frac{d\mathcal{L}}{d\mu'} = 0 \tag{3.8}$$

$$\frac{d\mathcal{L}}{d\mu} = -\frac{1}{\mu'^2 f^2(\mu)}\frac{df(\mu)}{d\mu} \tag{3.9}$$

$$\frac{d}{dx}\frac{d\mathcal{L}}{d\mu'} = \mu''\frac{6}{\mu'^4 f(\mu)} + 2\frac{2}{\mu'^3 f^2(\mu)}\frac{df(\mu)}{d\mu}\frac{\mu}{dx} \tag{3.10}$$

$$\mu'' = \frac{-\mu'^2\frac{df(\mu)}{d\mu}}{2f(\mu)} \tag{3.11}$$

Plugging in the noise expression $\varsigma = \sigma_0(p\mu^2 + \mu)$ into $f(\mu)$:

$$f(\mu) = \frac{2\varsigma + (\partial_\mu \varsigma)^2}{2\varsigma^2} = \frac{\sigma_0 + 2\mu(1+p\mu)(1+2p\sigma_0)}{2\mu^2(1+p\mu)^2\sigma_0}\ , \tag{3.12}$$

and replacing into 3.11 we obtain:

$$\mu''(x) = \mu'^2\frac{(1+2p\mu)(\sigma_0 + \mu(1+p\mu)(1+2p\sigma_0))}{\mu(1+p\mu)(\sigma_0 + 2\mu(1+p\mu)(1+2p\sigma_0))} \tag{3.13}$$

Henceforth we shall consider the boundary values of the morphogen profile fixed: the concentration value is normalized to the left ($\mu(0) = 1$) and the right value will be set to a small number ($\mu(1) = \nu$), the relevance of which shall be discussed later. This differential equation can be exactly solved for different noise scaling limits[6]:

Table 3.1: Analytic solution of Eq. (3.13) for different noise scaling behaviors.

noise model	ς	DE	result
constant	σ_0	$\mu'' = 0$	$(\nu - 1)x + 1$
intrinsic	$\sigma_0\mu$	$\mu'' = \frac{\mu'^2}{2\mu}$	$((\sqrt{\nu} - 1)x + 1)^2$
extrinsic	$\sigma_0\mu^2$	$\mu'' = \frac{\mu'^2}{\mu}$	$e^{x \log \nu}$

[6] For the intrinsic case the DE is actually

$$\mu''(x) = \frac{(\sigma_0 + \mu(x))\mu'(x)^2}{\mu(x)(\sigma_0 + 2\mu(x))}$$

but we use the small noise limit where $\sigma_0 << \mu$ in the interest of analytic tractability.

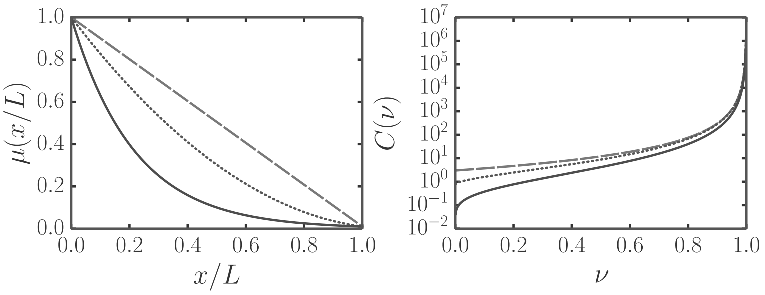


Figure 3.5: Left: Theoretical optimal profile $\mu(x)$ for the cases of constant (dashed), intrinsic (dotted) and extrinsic noise (full). Right: $C(\nu)$ for the same cases.

Conservation law Because $\mathcal{I}$ does not explicitly depend on x, we know there is a conservation law in the system. It is completely analogous to the conservation of energy in classical mechanics [159]:

$$\mathcal{L} - \mu' \frac{d\mathcal{L}}{d\mu'} = C \tag{3.14}$$

Replacing the functional 3.7, we obtain

$$\frac{6\varsigma^2}{\mu'^2(2\varsigma + (\partial_\mu \varsigma)^2)} = C \ , \tag{3.15}$$

which means $3/\mathcal{I} = C$ since

$$\mathcal{I}^{-1} = \frac{2\varsigma^2}{\mu'^2(2\varsigma + (\partial_\mu \varsigma)^2)} \tag{3.16}$$

C can be calculated for the various noise limiting cases, summarized in Table 3.2[7]. This conservation law essentially shows how the minimal error depends on two variables: on the one hand, the noise amplification factor σ_0 and on the other, the dynamic range of the signal.

The dynamic range is inversely proportional to ν, since $\mu(x)$ is normalized to 1 at the anterior boundary and it is monotonous decreasing. As makes intuitive sense, the error can be decreased by either increasing the dynamic range of the system (at the expense of energy) or by decreasing the noise amplification (e.g. via temporal or spatial averaging, again at the expense of energy).

[7] For the intrinsic noise case, we have to taylor expand in σ_0, which results in

$$\frac{3\sigma}{4\left(\sqrt{\nu}-1\right)^2} + O\left(\sigma^2\right)$$

Table 3.2: Conservation law (3.15) for the various noise limiting cases.

noise model	μ	ς	C
constant	$(\nu-1)x+1$	σ_0	$\frac{3\sigma_0}{(1-\nu)^2}$
intrinsic	$((\sqrt{\nu}-1)x+1)^2$	$\sigma_0((\sqrt{\nu}-1)x+1)^2$	$\frac{3\sigma_0}{4\left(\sqrt{\nu}-1\right)^2}$
extrinsic	$e^{x\log\nu}$	$\sigma_0 e^{2x\log\nu}$	$\frac{6\sigma_0}{(\sigma_0+2)\log^2(\nu)}$

Lognormal gene expression From experimental data we find that noise in Bcd is mostly of the extrinsic type. As we have seen in 2.7.1.1, a simple and general model which explains extrinsic type fluctuations is a lognormal distribution[8]:

$$P(m|x) = \frac{1}{m\sqrt{2\pi}\sigma}\ \exp\left(-\frac{\left(\log m-\mu(x)\right)^2}{2\sigma^2}\right) \tag{3.17}$$

It is clear that for a change of variables with $y = \log m$ we recover a Gaussian distribution. In this model we consider σ^2 constant and then go on to solve the minimization problem (3.5). After some calculation, we recover the same differential equation as for the constant Gaussian case, the solution of which is a linear profile for y (refer to Table 3.1), which translates to an exponential profile for m. This is consistent with the extrinsic Gaussian result.

Comparison to data In [57], enormous amounts of quantitative data were acquired to precisely measure the Bcd profile's statistical properties. The experimental setup consists of transgenic fly lines where endogenous Bcd is replaced with fully functional EGFP-Bcd fusion protein. Using live fluorescence imaging, expression of this protein can be quantitatively measured. The acquired images are processed to extract fluorescence in each nuclei along the egg wall, which results in a measurement consisting of a pair $\{\mathrm{AP}, \mathrm{fl}\}$.

These measurements are binned by AP position into 100 bins and the mean and variance of each is calculated (Fig. 3.6, left). The histogram of the binned measurements can be calculated and fit to a lognormal (Fig. 3.6, right). Since the mean profile of Bcd is well described by an exponential distribution [126], the observation that the data appears to follow a lognormal distribution is consistent with the results obtained in the section above. Whether by accident or optimization, it appears that the natural Bcd profile is optimally informative.

3.3.2 Morphogen profile scaling

Bicoid scaling The question of scaling naturally must begin with the Bcd gradient, as it is the earliest spatially distributed TF to appear in the *Drosophila* embryo (considering

[8] For a lognormal, $\sigma_m^2 \propto \langle m\rangle^2$.

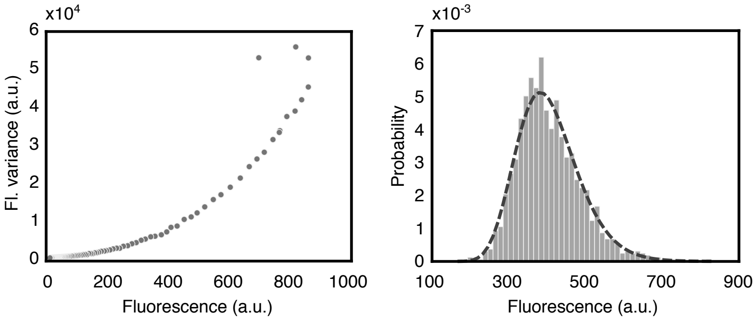


Figure 3.6: Left: variance as function of mean GFP fluorescence in embryos where the Bcd promotor is fused to a GFP construct. The scaling appears $\propto \langle m \rangle^2$. Left, GFP fluorescence distribution for $AP = 0.25L$ and respective MLE lognormal fit (dashed). Data from [57].

only the AP axis). We will leverage the *Liu et al.* dataset [57] to determine whether scaling is found at this level or whether it emerges later.

The simplest approach is to perform an exponential fit to the data of each embryo in the dataset and calculate the statistical properties of its two parameters. Then their correlation coefficient with the embryo length can be determined.

In Fig. 3.7 a summary of the statistics for the parameters A and λ is displayed. As would be expected from the SDD model, higher dosages imply higher decay lengths in the profile. The amplitude appears to be log-normally distributed, while the decay length appears to be normal.

These parameters can then be correlated to the length of each embryo. We can see in Fig. 3.8 (left) that both parameters show essentially non-existing correlation with L. The effect size, as quantified by R^2, is approximately zero. Some groups have suggested that Bcd is deposited in the anterior portion of the embryo in a volume-dependent manner, which would translate to higher amplitudes for larger embryos [148].

If that were the case we would roughly expect $\kappa \times L = A$ with κ a constant independent of of embryo length. While Fig. 3.8 (upper left) appears to support this hypothesis, as the effect size was reduced by an order of magnitude, the effect was already so small that this difference would appear to have no practical implication in a biological setting.

If we were to perform the exponential fit in a rescaled profile, such that $\xi = x/L$, then we would obtain the same value for A, but we would obtain a rescaled value $\hat{\lambda} = \lambda \times L$. This rescaled value will be correlated with L in case the original λ isn't (3.8, lower right). This observation will be useful later on.

Principal component analysis To compare the results for the Bcd profile with downstream gap gene profiles, we need to develop a profile agnostic method of determining scaling. Toward this goal, we shall employ PCA to reduce the dimensionality of the quantitative profiles. In Fig. 3.9 I show that such low dimensional features will map the the exponential fit parameters in the case of Bcd.

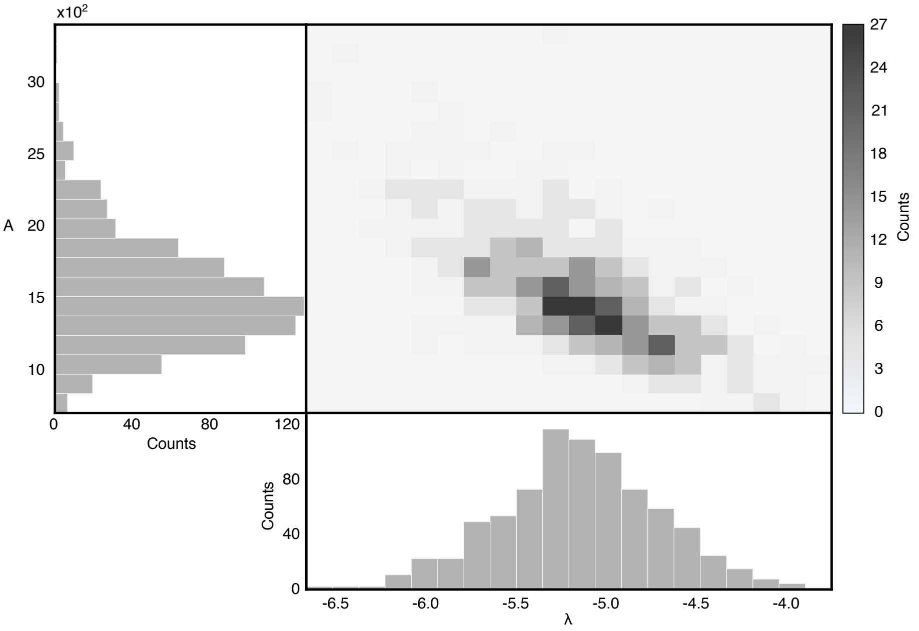


Figure 3.7: Parameters resulting from a fit to the expression $y = A\exp(\lambda x)$. Given that each profile is a sample from a random distribution, the parameter values form themselves a probability distribution $P(A, \lambda)$. In the middle, the histogram for this distribution shows that the two parameters are not independent, which could have implications for scaling. On the left and bottom, we can see the marginalized distributions for the parameters.

Conceptually the PCA method is quite simple. Suppose our data consists of p samples of an n-dimensional data vector $\mathbf{x}$. After mean subtraction, the sample covariance matrix of the data is proportional to

$$C = \sum_{j=1}^{p} \mathbf{x}_j^T \mathbf{x}_j \ .$$

The eigendecomposition of this matrix $C = W^T \Lambda W$ defines a linear transformation to a new space via W: $\lambda = \mathbf{x}W$. Because this is the natural space for the covariance of the data, some of the dimensions will explain most of the variation in the data, while others will show only marginal variation. This hierarchy in dimensions is quantified by the eigenvalues Λ. To perform dimensionality reduction, we can keep only the k dimensions with the highest eigenvalues (which explain most of the variation of the data) by considering only the corresponding high-ranking eigenvectors in the linear transformation matrix W_k. These dimensions are called the *principal components*.

To apply PCA, we need all profiles to be measured at the same AP position such that all dimensions are correctly aligned. To achieve this, the procedure is applied to the rescaled profiles (i.e. with coordinates $\xi = x/L$). The experimental data has been discretized such that $\xi \in \{1, \ldots, 100\}$ and therefore the full profile will be represented by a vector

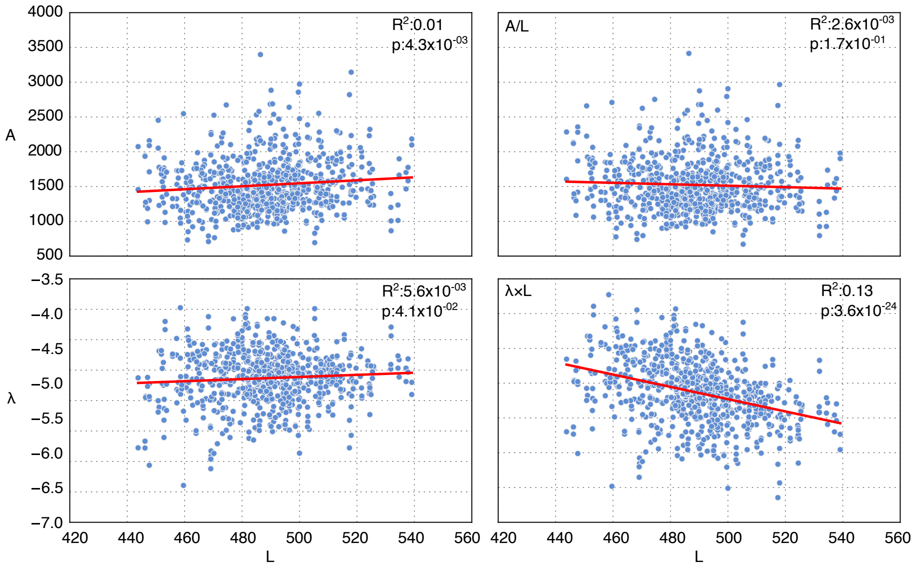


Figure 3.8: Parameters resulting from a fit to the expression $y = A\exp(\lambda x)$ correlated to the embryo length L of each profile. On the left, we see the fit to the raw parameters, which suggests that λ is not correlated to L at all, while there might be a minor correlation for A. This correlation is destroyed when we calculate $\kappa = A/L$ on the right, which provides marginal support of the volume-dependent deposition hypothesis. However, the effect size is too small to draw definite conclusions.

$\mathbf{m} = \{m_1, \ldots, m_{100}\}$. Given this, a profile which scales perfectly will be such that:

$$P(\mathbf{m}|L) = P(\mathbf{m}) \, , \tag{3.18}$$

as this is just the definition of scaling.

The ith feature calculated with PCA shall be denoted as λ_i and will follow:

$$\begin{aligned} P(\lambda_i|L) &= \int d\mathbf{m}\; P(\lambda_i|\mathbf{m})P(\mathbf{m}|L) \\ &= \int dm\; \delta(\lambda_i - w_i.\mathbf{m})P(\mathbf{m}) \end{aligned} \tag{3.19}$$

with w_i the ith eigenvector of the covariance matrix of $\mathbf{m}$[9]. Therefore, any principal component of a perfectly scaling profile should be perfectly uncorrelated with L and we can easily test this using hypothesis testing. The hypothesis under consideration is whether the correlation coefficient between L and λ_i is zero. If this hypothesis is rejected for any i, we can reject scaling with statistical significance.

After establishing that PCA can successfully extract a low dimensional representation of the profiles in terms of descriptive features, we can confidently apply this method to the downstream gap genes. Using data from Thomas Gregor's lab (unpublished), we applied

[9] This is just the definition of PCA

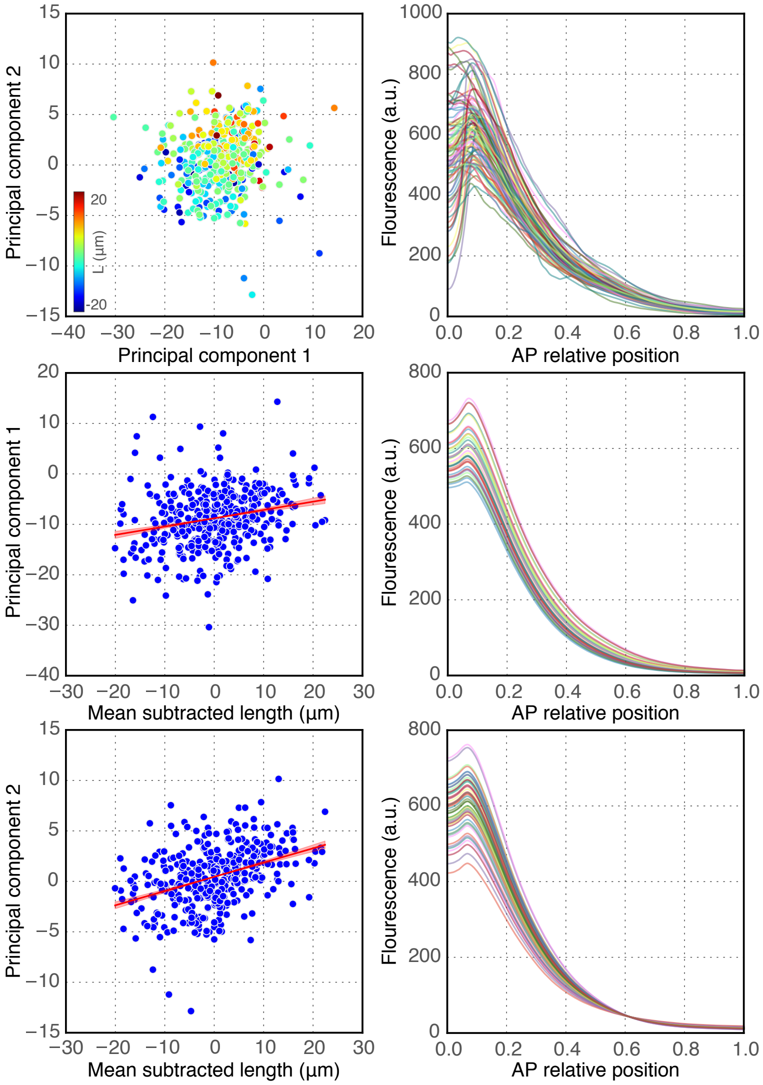


Figure 3.9: Bcd PCA dimensionality reduction for the 2XA dataset from [57]. Top column, left: Bcd profiles represented with only the two largest principal components, colored by embryo length; right: raw data. Lower columns, left: correlation between each of the two principal components and embryo length; right: samples drawn from each independent principal component, illustrating the variation in data space induced by moving along that coordinate.

this procedure to quantitative data of *Drosophila* gap gene profiles for wild type and mutant embryos.

The mutant embryos have the Bcd or Torso-like Protein (tsl) gene deleted. Since these maternal genes influence gene expression at the anterior [61] and posterior [160, 161] ends respectively, we hypothesized that their deletion might negatively impact scaling of the gap genes.

A lot can be learned from Fig. 3.10. To start with, let's focus on the wild type embryos denoted by Bcd2X (meaning that two alleles coding for Bicoid are present). In this case it can be seen that for all gap genes the effect size of scaling dependence reduces from early to late time stages [10]. This is in agreement with the hypothesis that wild type gap gene network dynamics compensate for lack of scaling in the Bcd gradient.

If Bcd is deleted (denoted by bcdE1 in the dataset) some scaling is lost in the late time stage as compared to wild type embryos for the case of Krüppel (Kr), Knirps (Kni) and Hunckback (Hb). This is not true for Giant (Gt) but in this case the gene expression profile has been seriously under-expressed due to the mutation as can be seen in the x-axis of Fig. 3.10, upper right. Note that scaling cannot be correctly determined for genes with too low expression levels, since a flat spatial profile corresponding to basal expression scales trivially.

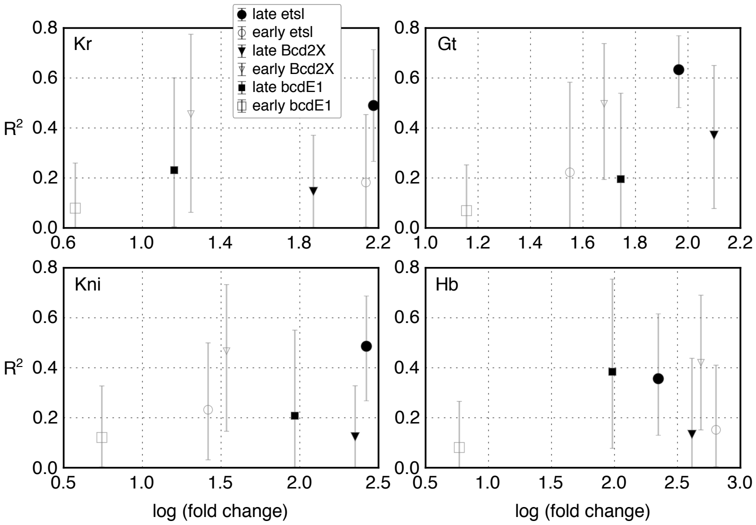


Figure 3.10: Summary of PCA analysis for gap gene data in different fly strains. The wild type strain is represented by *Bcd2X* (inverted triangles), Bcd deletion is represented by *bcdE1* (squares), and the tsl deletion dataset is represented by *etsl* (circles). The most strongly rejected PCA component was taken for each of the data points. The x-axis represents profile fold change, as some deletions generate mostly flat profiles, for which scaling cannot be determined. Therefore, we only should consider points to the right of the x-axis as representative. The y-axis represents the effect size, with points further up exhibiting an apparent rejection of scaling.

In the case of tsl deletion (denoted by etsl), we again find lack of scaling when compared to wild type. Due to the amount of data available, scaling can be rejected at the 5% significance level for all gap genes in the late stage. This appears to confirm the hypothesis that two (non-scaling) gradients pinned at opposite poles of the embryo are necessary for scaling compensation.

[10] Early corresponds to time interval of 15 to 25 minutes into NC14 and late to the interval of 40 to 50 minutes into NC14

3.3.3 Error correction schemes for scaling

Using Monte Carlo sampling, I can simulate the probability distributions arising from scaling and non-scaling systems to assess estimator performance using different error correction strategies. The basic idea is to sample the joint distribution $P(m, x)$ under different conditions and then use $P(x|m)$ to calculate the optimal estimator in the bayesian sense, the performance of which is later assessed.

Drawing samples from $P(m, x)$ is straightforward in the scaling case, since one can draw samples of $P(x)$ by sampling from a uniform distribution with range $[0, 1]$ and m for each x is such that:

$$\log m = \lambda x + \log A + \eta \ , \tag{3.20}$$

with η a zero-mean gaussian with preset $\sigma = 0.1$.

Given the samples, it is quite simple to build the posterior distribution $P(x|m)$: 100 equally spaced bins b_i for m are generated, and the empirical distribution $P(x|b_i < m < b_{i+1})$ is used as a proxy for the posterior. From this distribution, an estimator for x in each concentration bin can be derived using the rules from estimation theory (Section 2.3). Using this procedure we get a mapping from $\bar{m} \rightarrow x_{est}$ defined at 100 different points – with $\bar{m}$ the average m in each concentration bin. Since the distribution of m is continuous, we need a mapping for the other possible values of m. To do so, we use the existing 100 points to create a linear interpolation function $x_{est}(m)$ (Fig. 3.11B).

Applying this estimator mapping to all sampled pairs (m, x) we get the pairs $(x_{est}(m), x)$ which can be used to assess estimator quality via squared deviation:

$$\sigma_k^2 = \sum_{i:k/N<x_i<(k+1)/N} (x_{est}(m_i) - x_i)^2 \tag{3.21}$$

I experimented with two estimators: the posterior mean and the MAP estimate. Given that we are using the squared deviation loss function (3.21), the mean estimator is the theoretical optimum, which we can see in 3.11A, which shows the error for both a perfectly scaling and a non-scaling profile. Nonetheless the MAP estimator, which is more biologically feasible, fares almost as well. It can also be seen from Fig. 3.11A that in the fixed length case the optimum predicted by the FI functional is recovered, with some slight deviations at the boundaries. In case of embryo length fluctuations the non-scaling profile entails a large error towards the end of the embryo.

Extending the procedure to use two profiles, one pinned to the anterior end and one to the posterior end, we obtain a joint distribution $P(m_1, m_2, x)$ and can create an estimator $s(m_1, m_2)$, visualized in Fig. 3.11D. Measuring from these 2 profiles results in an improved error of the order of $1/\sqrt{2}$ for the fixed length case, as expected. But in the fluctuating length case, we see a much greater improvement, provided by the complementary gradient measurements.

A second way to correct for scaling in the case of a non-scaling input gradient is the introduction of a diffusion process with fixed boundary conditions at the anterior and

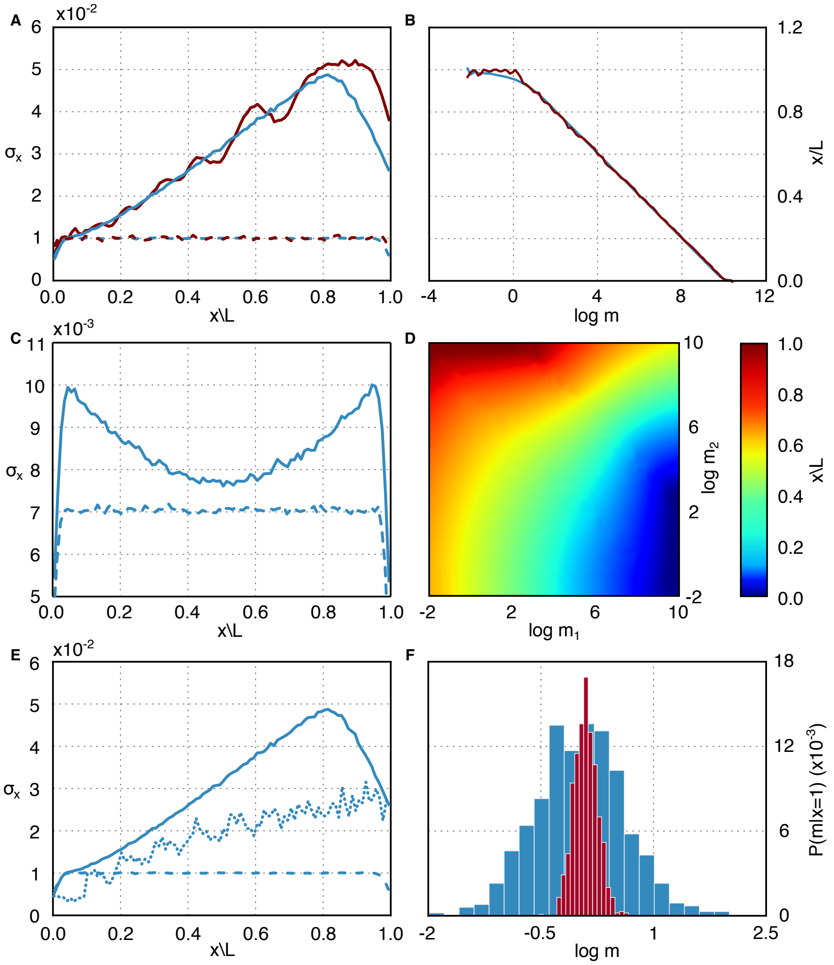


Figure 3.11: A) Estimated error with a single non-scaling profile using the mean (blue) or MAP (red) estimator in the case of no embryo length fluctuations (dashed) or with gaussian distributed embryo length fluctuations with 16% variation around the mean (solid). B) Interpolated estimator curves. C) Estimated error with two profiles pinned at opposite ends using the mean estimator. Dashed and solid lines as before. D) Interpolated 2D estimator surface. E) Estimated error with one profile with pinned diffusion. Dotted line estimated positional error from (3.21) using smoothed profile h. F) Distribution of concentrations m at posterior end of the embryo read out from profile h (3.22) with fixed boundary conditions (red) and without (blue).

posterior ends of the embryo. This generates a pinning effect which can correct for embryo length fluctuations. Mathematically, suppose the profile h is established by the following diffusion process with input from a non-scaling profile μ:

$$\begin{aligned} \frac{dh}{dt} =& D\nabla h + \mu(x) + \eta \\ h|_{x=0} =& h_{\max} \\ h|_{x=1} =& h_{\min} \end{aligned} \tag{3.22}$$

Even with additional noise added to the process, the final positional error by reading out h is smaller than that of directly reading off μ (Fig. 3.11E). This hints at the possibility of information transmission from the boundaries via diffusion, which will be dealt with in depth in the next chapter.

3.4 Conclusion and outlook

In this project we investigated in-depth the positional information properties of maternal morphogens in nature and how they can be affected by external factors. The FI provided a way to quantitatively characterize positional information in arbitrary morphogen profiles without the need to make any assumptions on the mechanism of the profile's generation. We used variational calculus to determine optimally informative profile shapes for various fluctuation types, and provided more evidence to corroborate the hypothesis that the maternal profile Bcd appears to be optimized for positional information.

We investigated how embryo to embryo length fluctuations might hamper positional information and what are the methods which can be employed to compensate for this scaling problem, given that the maternal profiles are not robust to scaling variations. It is plausible that *Drosophila* uses a combination of readout from symmetrically opposing gradients combined with diffusion at the lower levels to create scaling profiles at the gap gene level.

In the future it would be interesting to explore experimentally how scaling is achieved by the gap genes: it is as of yet unclear whether a combinatorial readout from maternal genes is employed or whether an emergent effect from network dynamics provides scalability to the final pattern.

Chapter 4

Emergent computation by reaction networks

4.1 Motivation

In the previous chapter we discussed how developmental systems might use boundary information to compensate for fluctuations in a direct readout of a morphogenetic input. This information eventually reaches all cells in the lattice via the dynamics of the regulatory network, as the final output is both robust and reproducible. Here, we wish to examine how maternal information is combined with network dynamics to produce these robust patterns under the lens of the theory of distributed computation.

The idea of distributed computation has been treated in various, disparate ways in the literature. One can easily get confused by various closely related and somewhat overlapping in definition terms which are used in conjunction with, or as an alternative to, distributed computation. By distributed computation in this thesis I understand an input-output mapping where the output depends on communication between multiple identical compute-nodes. Distributed computing is not necessarily emergent. In an emergent system the final state cannot be deduced by inspecting the behavior of a single agent [13]. Nonetheless, most natural distributed computing systems we will study are emergent.

Hermann Haken, who founded the whole field of *Synergetics* in the 70s [162]. The notion of *Synergetics* encompasses the use of non-equilibrium thermodynamics, nonlinear dynamics and stochastic processes in order to characterize self-organizing processes in nature. The notion of self-organizing is rather vague: a self-organizing system should reach some sort of attractor without external control, but what is internal and external to the system is left to the modeller's choice of boundaries [13]. Naturally, it is possible to consider the Turing mechanism as self-organizing.

In the field of synthetic biology there has been progress when it comes to implementing simple distributed algorithms in genetically engineered yeast [163] and *E. Coli* [49]. These approaches employ parallel computation of simple functions combined with averaging to overcome stochasticity in the output [164]. Simple pattern forming systems have also been implemented, both with [165] and without [28] external organizing signals.

The most general definitions and studies of distributed computing are found – perhaps unsurprisingly – in the theoretical computer science literature. A lot of work [166, 167] has focused on designing biology-inspired distributed algorithms in order to reproduce some of biology's features: robustness, energy efficiency, etc. We can even find some attempts to create distributed pattern-formation systems [168] based on multi-agent interactions. These algorithms do differ from those found in nature, since they are modular [169] (and necessarily not emergent). Nonetheless, some work appears to mimic natural behaviors, such as the well known *boids* algorithm for bird flocking [170]. If algorithms for a specific problem are universal it might be possible to translate solutions between biology and computer science [171].

The origins of the field as applied to biology lie in Alan Turing's seminal paper "The chemical basis of morphogenesis" [172]. Turing develops therein a theory of pattern formation based on *reaction-diffusion equations*.[1] The key insight is that a locally linearly stable system may be driven to non-homogeneous spatial distribution via diffusion (known as a **Turing instability**).

The local kinetics for the Turing mechanism are simple: one chemical species A self-activates and activates another species B which represses A. We also need B to diffuse faster than A. Intuitively this leads to a short range activation and long range inhibition of A, leading to localized stripes where the concentration of A is high. The 'wavelength' of these stripes is determined by the chemical parameters only [173]. Recent experiments have shown that some morphogenetic processes are indeed controlled by a Turing mechanism [174, 175].

Reaction-diffusion models raise interesting theoretical questions applicable to more than pattern formation: they establish that a distributed system of many cells performing the exact same computation can produce a global result which is not explicitly encoded in the local algorithm – known as **emergent computation** [176]. It can be shown these systems are Turing complete[2] [177, 178], and thus they suffer from the *undecidability* of the halting problem: given an algorithm [3] and some inputs it is not even possible to know if the program finishes running (i. e.returns the desired value and performs no further computation) [179].

Another model useful for the study of distributed pattern formation is the CA. A cellular automaton is essentially a discretization of a reaction-diffusion model: instead of continuous dynamic variables, space and time; we have discrete states, a discrete spatial lattice and incremental time steps. Mathematically it is described by a *transition table* which describes which state a cell will acquire in the next time step given its current state and that of its neighbors. Further mathematical details will be given in Chapter 4.

It was thought that cellular automata would provide a 'toy model' for emergent com-

[1] A reaction-diffusion equation is generally $\dot{x} = f(x) + D\nabla^2 x$

[2] A Turing complete system can simulate a Turing machine, which can in turn in principle compute any arbitrary algorithm.

[3] In our case an algorithm is a reaction function describing all the chemical interactions; the inputs are the initial conditions

putation – initial progress was made with Elementary Cellular Automaton (ECA) [180] by categorizing them into distinct 'complexity classes'; and with 2D totalistic rules[4] such as Conway's 'Game of Life' by showing their Turing completeness. Progress has stalled since then, with fundamental obstacles such as the halting problem preventing the development of a full mechanistic understanding of the dynamics.

The interesting features of these models rely on the interaction of a nonlinear function with some sort of a communication mechanism: in reaction-diffusion models, diffusion plays this role [173]; while in cellular automata, it is the neighborhood state which conveys nonlocal information [181].

The question we are interested in here is how to design networks to reproduce certain patterns.

4.2 Previous work

As previously discussed in 1.4, the principal model for pattern formation where the regulatory networks involved are relatively well known is *Drosophila*. It has been suggested that the final gap gene pattern at a later stage in development is a product of gap gene network dynamics, a process known as canalization [67, 156]. Recent experiments have shown that removal of one or more elements of the gap gene networks alters the patterns of the remaining gap genes, suggesting that they are not completely determined by maternal readout [182]. An investigation of the *cis*-regulatory mechanisms for the lower-level pair rule genes suggest that their dynamics also follow a canalization principle [71].

In higher level organisms, there is further evidence for pattern formation via a Turing type mechanism. In mice, digits are formed by a stripe like pattern emerging from repressive interactions [174]; while in Zebrafish skin pigmentation is formed by cell-cell interactions [183]. Mechanical interactions and growth also play a role in defining emergent patters [184] via sensors measuring intracellular forces and cell-cell adhesion [185, 8, 186, 187, 188].

Formally these patterns can be considered to be attractors of the dynamics of the particular gene regulatory network [52, 67]. However, it is not clear how to map network dynamics to final pattern features, so even theoretical research has focused mostly on reverse engineering networks for given patterns [189, 190, 191, 192].

Of course, we should keep in mind that network architecture alone does not completely determine the final pattern. In the gap gene system, deleting one posterior maternal morphogen will disrupt the final pattern [193]. Aside from direct positional positional information from maternal morphogens, localized boundary inputs also help determine the final pattern [72, 194].

Under this lens, we can consider the network as defining fixed attraction basins in a high dimensional space, with external inputs driving the dynamics to a certain attractor [195]. Significant progress has been made in the understanding of how inputs drive the system

[4] A totalistic rule is a simplified cellular automaton which looks only at the sum of the states of its neighbors

to different attractors using the language of **control theory** [196]. Numerical simulation is essential to this understanding as the control input sequence is often nonintuitive [197].

4.3 Results

4.3.1 Boundary control in cellular automata

To begin our investigation we use CA rules to locally determine the temporal evolution of the configuration of a spatially distributed system [198]. Even the simplest case of CA, a 1D row of cells with only two distinguishable states, can generate a wide range of patterns [180]. Furthermore it has been shown that CA can be used to coarse-grain the dynamics of complex systems and recover large-scale features with relatively simple models [199].

Given that CA are discrete and finite systems, it is possible to definitely determine if a pattern can be produced in this framework and in what conditions. To do so, we need to look at the system dynamics (in this case, set by the CA rules) and external control inputs. Here, we will focus on inputs at the boundary of the system for simplicity's sake.

Cellular automaton definition A CA system is defined in a discrete lattice, each point of which contains a number (the *state*). Here we will focus on ECA, meaning that the lattice is 1D and only two states are allowed: 0 or 1. The temporal evolution of these states is defined by the CA rule: a transition table which defines the next state as a function of the current state and the state of its neighbors.

In the ECA, only next nearest neighbors are considered. This means that there are 2^3 possible initial states and 2 possible final states which translates to a total of 256 district rules. Following Wolfram [180], these rules are numbered according to the bit string of possible final outcomes (i. e.rule 110 has outputs 01101110 as in Table 4.1, which is the binary expression for 110).

Table 4.1: Transition table for CA rule 110.

x_{i-1}^t	x_i^t	x_{i+1}^t	x_i^{t+1}	x_{i-1}^t	x_i^t	x_{i+1}^t	x_i^{t+1}
0	0	0	0	1	0	0	0
0	0	1	1	1	0	1	1
0	1	0	1	1	1	0	1
0	1	1	1	1	1	1	0

For the values of $i = 1$ and $i = L$ a choice must be made regarding the extremal values. The usually studied choice is that of periodic boundary conditions [180], where the leftmost state for x_1 is set to the state of x_L and the rightmost state of x_L is set to x_1. Here, we will focus on a different setup. The boundary values $i = 0$ and $i = L + 1$ will be defined by an external source, and we will seek a sequence of boundary values (x_0^t, x_{L+1}^t) such that a target pattern is attained.

4.3.1.1 Fast Control

As a first exercise we considered the case where the boundary values are allowed to change at every time step, and the final target configuration only needs to be attained at a particular time point. This setup is particularly tractable since for a fixed lattice length we can construct what we called a *fast control graph*.

As illustrated in Fig. 4.1, the fast control describes all possible transitions between every configuration in the lattice. For a lattice of length L we will have 2^L possible configurations, so this graph can only be generated for relatively small lattice lengths. Nonetheless, the graph generation process can be trivially parallelized. It is only necessary to iterate over all configurations and calculate the following configuration for all 4 boundary conditions. Then, a connection is added joining the initial configuration and final configuration containing a data structure where the corresponding boundary condition is stored.

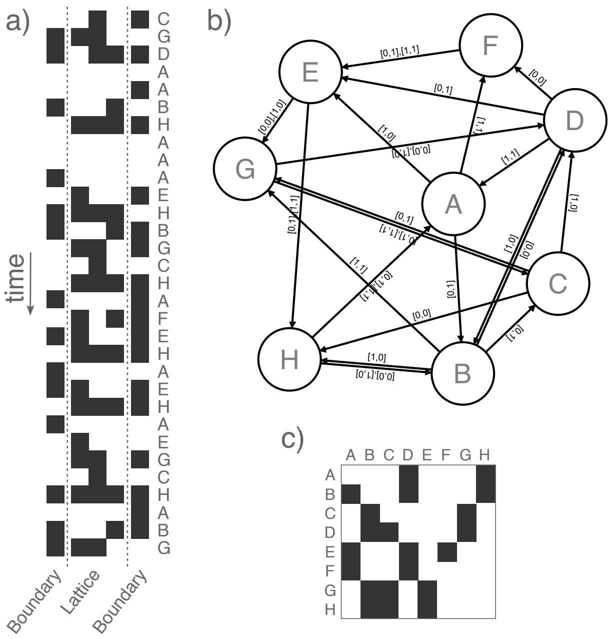


Figure 4.1: Fast control, taking CA rule 86 and lattice length L = 3. a) Temporal evolution of a system with varying boundary input, where time flows top to bottom. The first and last cell in each row correspond to boundary input. 0 = □, 1 = ■. b) Fast control graph as described in the text. The boundaries which induce a given transition are labeled on the corresponding arrows. c) The fast control graph can be described with a connectivity matrix.

This fast control graph can be queried for the fastest trajectory from one configuration to another. So if the system is to start under a homogeneous configuration, we can query the trajectory from the null configuration to the target configuration. By iterating over all possible 256 rules, it is possible to determine the optimal rule / control sequence for a given pattern.

We also investigated how controllable each rule is by calculating a property of the fast control graph: if it is strongly connected, it means every configuration can be reached from any other by using an appropriate boundary control sequence. For the 88 unique ECA rules[5], we observed that the number of rules with strongly connected graphs decreases as the lattice length L increases until only 10 remain: rule 15, 30, 45, 60, 90, 105, 106, 150, 154, 170. In the paper [239] we show that these 10 rules remain controllable for all lattice lengths.

4.3.1.2 Slow Control

In this section we will consider a system where the control inputs change only after a long time compared to the system's intrinsic timescale. This means that we wait for the system to reach a steady state (if one can be achieved under the current update rule) before updating the control input. We do not consider attractors with period > 1 as steady states. Since each rule will only have a small amount of attractors, we will also allow for multiple rules in a single control protocol.

The model system consists – as previously – of an ECA evolving in a lattice with a fixed size L. A control protocol is associated with a start and an end configuration and consists of an ordered sequence of *instructions*. One instruction consists of a specific ECA rule (now all 256 rules must be considered, since specific transitions between configurations are not symmetrically invariant) and an associated pair of boundary conditions. At each step, an initial configuration is evolved under the rule and boundary conditions specified by the instruction until the steady state is reached. Then, the next instruction is selected using the previous attractor configuration as initial condition. This procedure is repeated until all instructions have been applied and the final steady state (corresponding to the final configuration) has been reached.

A direct search for a control protocol with given start and finishing configuration would entail an exhaustive search protocol, which requires exponentially increasing resources and is therefore infeasible.

We will detail a more efficient method of obtaining control protocols below. To do so we need to construct the fast control graph for each ECA rule (same as Fig. 4.1b). A directed edge connects two nodes X and Y if $Y = F(X)$. A node with a self connection corresponds to an attractor configuration. We calculate $F(X)$ for all configurations X in a lattice of fixed width.

Once these auxiliary graphs are constructed and the attractors in each have been determined, we calculate an attraction basin for each attractor. The attraction basin is composed of all configurations in a specific auxiliary graph from which the given attractor is reachable (Fig. 4.2a). At this stage we will need another graph, which we shall call the

[5] Out of the 256 rules, only 88 remain unique after considering the mirror (left-right) symmetry and the state ($0 \rightarrow 1$, $1 \rightarrow 0$) symmetry.

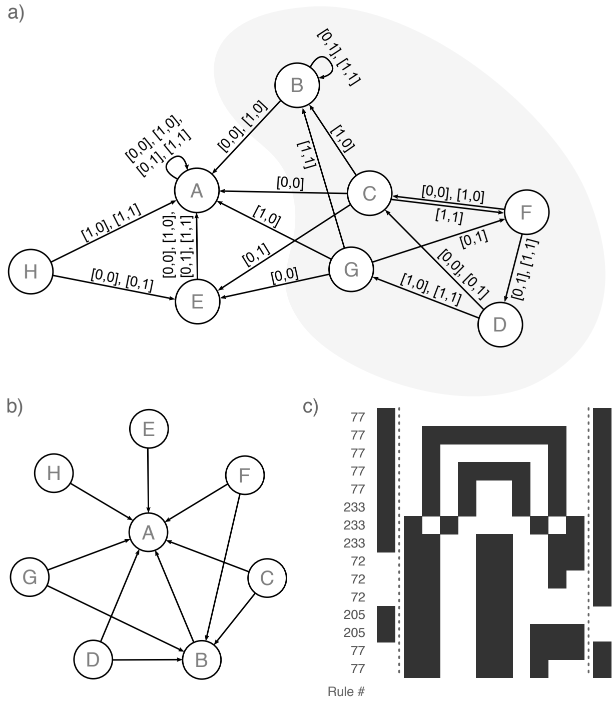


Figure 4.2: a) The basin of attraction of configuration A in this fast control graph for rule 40 is the whole state space, while the basin of attraction for B is the grey shaded region. b) These basins of attraction are mapped to a slow control graph. c) By querying the slow control graph we obtain dynamics.

slow control graph, where again each node is a configuration. Now, for each configuration X in the attraction basin of the configuration Y^* we add a directed edge $X \to Y^*$ to the control graph (Fig. 4.2b). This edge will contain as auxiliary information the rule and boundary configuration of the auxiliary graph with which it is associated.

After iterating through all auxiliary graphs (in the case of the ECA rules there are 1024: 4 boundary configurations × 256 rules) the control graph can be used to extract the control protocol for any pair of configurations. Many configurations will have multiple possible paths joining them, making it necessary to define an objective function which will ascertain the quality of a given function. Here we focus on requiring as few instructions as possible, corresponding to a desire for the shortest 'program length' to obtain a given configuration. The optimal control for this objective is obtained by determining the shortest path between the two target configurations on the control graph. Other objective functions are conceivable such as minimizing the total sum of the number of iterations necessary to reach an attractor for each instruction.

In the following analysis we will restrict ourselves to analyzing the behavior of the system starting from 0, a homogeneous initial condition. This corresponds to a maximally uninformative initial condition: there are no 'seeds' which could simplify the generation of some complex patterns by providing some information. From this idea, we could define the complexity of a pattern by how many instructions it takes to reach starting from 0.

This idea is inspired by the concept of the Kolmogorov complexity of a string, defined by the length of the shortest program running on a Turing machine which outputs that string and halts [81]. The Kolmogorov complexity depends on the computing machine used to generate the string. Suppose $\mathcal{T}$ is a Turing machine, and $\mathcal{A}$ is another kind of computing machine (for example, our CA). Then the Kolmogorov complexity K is such that:

$$K_{\mathcal{T}} \leq K_{\mathcal{A}} + c_{\mathcal{A}} \tag{4.1}$$

with $c_{\mathcal{A}}$ describing the code necessary to emulate $\mathcal{A}$ on $\mathcal{T}$. This means that we can construct some machine which requires very little input to describe a particular string, but that does not make its Kolmogorov complexity low if such a machine is complicated to emulate on a universal Turing machine. Since we are dealing with finite strings, the value of this constant will be important and therefore the complexity of strings can only be compared within the framework of bounded ECA. Nonetheless, assuming a setup where two agents implement an ECA model, a sender could efficiently compress a pattern by transmitting only the control protocol instead of the full description of the configuration.

In this sense, a pattern's complexity is rigorously defined by how much information must be transmitted to reproduce it. On one extreme of this scale will be strings which only require one instruction. There we find strings with visually simple patterns: alternating zeros and ones, domains separated by a wall. These are the strings we would expect to be most compressible. One noticeable absence however is the pattern where all states on one half of the lattice are set to zero and all states on the other half to one (in the case of $L = 8$ this would be configuration 15); it requires two instructions.

On the other end of the scale we find patterns which cannot be reached at all starting from zero. This only occurs for lattice lengths $L > 8$. Not being reachable means that this configuration is disconnected from the whole 'reachability' basin connected to zero. Thus to create these patterns a carefully crafted initial condition is necessary. In fact for the studied lattice lengths all these configurations are isolated nodes (rather than forming their own connected subgraph). The only way to reach these isolated nodes is if the initial condition already matches the target pattern – no compressibility at all.

There does not appear to be a way to predict which patterns are unreachable from first principles. The asymmetry does not appear to play a role, as similarly irregular patterns exist which are reachable from zero. One might expect that these unreachable patterns are only a steady state for the identity rule (rule 204). While this is the case for most unreachable patterns, a significant number are steady states under other ECA rules as well.

By querying the control graph for the shortest path to all other configurations from 0, we can get an idea of how the control behavior scales with L. Contrary to what one might expect, the maximum control protocol length for reachable configurations does not scale in any predictable fashion (Fig 4.3c). Empirically it appears that larger lattices simply can contain patterns with more irregular features (i. e.less compressible) which require longer control protocols to reach. However, those patterns are not at all typical, as we can see in 4.3b. The distribution of instruction lengths appears to have a fat tail towards the right, with most of the patterns having a relatively small instruction length compared to the lattice length. In fact from our simulations it appears that the average number of instructions necessary to generate a (reachable) patterns follows the line $L/2 - 2$.

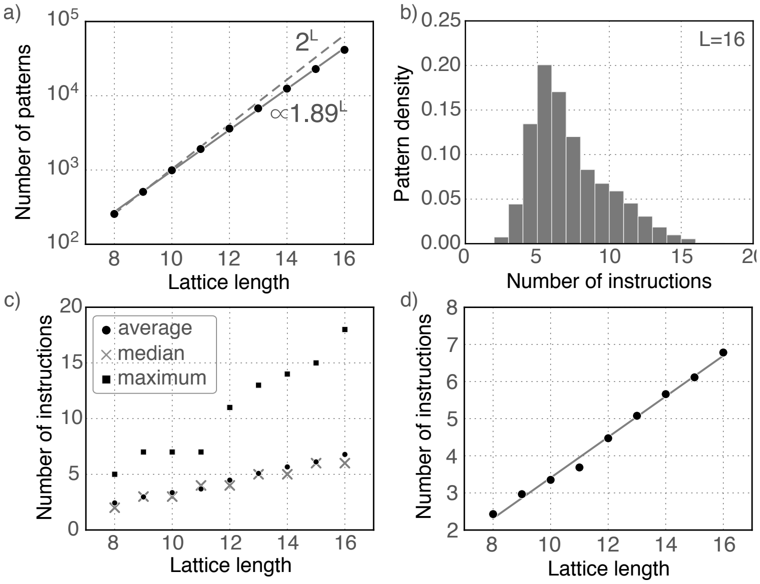


Figure 4.3: a) Number of reachable patterns via slow control as a function of lattice length calculated from slow control graph. The solid line represents a linear fit to the log of the data, while the dashed line is the total number of possible patterns. b) Histogram of number of instructions necessary to generate all reachable patterns for $L = 16$. c) Statistics for the distribution in b) for all L. d) Scaling of average number of instructions with lattice length ($\sim L/2 - 2$).

Comparing the number of reachable states to the total number of states as a function of lattice size for various lengths (Fig 4.3a) it appears that the fraction of patterns reachable from zero becomes negligible (although their absolute number does grow). Previous work [200] has established a procedure for analytically determining the number of attractors for a bounded ECA. According to our observation it appears that the number of reachable patterns scales $\propto 1.89^L$.

We hypothesize that the subset of reachable patterns in the slow control regime might correspond to the set of compressible patterns in the sense of Kolmogorov complexity. As a comparison, consider the general way to generate a stable pattern without using the CA's emergent properties: we would use an external source of positional information (like a morphogen) which would code each position uniquely and then read the appropriate state for that position from a look-up table. It is easy to show that in that case the program length would be $L\,(\log_2 L + 1)$ bits[6]. This procedure is, however, able to generate any

[6] For each lattice position, we need to store the morphogen code with $\log_2 L$ bits, and the final state

pattern, unlike slow control. On the other hand, the average program length using slow control is $10\,(L/2-2)$ bits[7]. This is much cheaper than the general procedure.

Given our analysis, we conclude that slow control can be used to reliably generate a subset of all patterns, which we termed the reachable patterns and which appear to be compressible in an algorithmic sense. The members of this set are determined by the properties of the ECA rules themselves, and therefore are model-dependent. However, since the ECA is a minimal model for algorithmic generative models, we expect that the complexity of the patterns as determined herein is a good upper bound on their universal complexity [81].

4.3.2 Reverse engineering emergent network dynamics

While the fact that the state space of elementary CA is finite allows for tractable results via enumeration, the possible patterns achievable are limited. For a more realistic model, we would like to simulate reaction diffusion equations of the type:

$$\dot{x} = f(x) + D\nabla^2 x \tag{4.2}$$

The number and kind of achievable patterns will depend on the choice made for the nonlinear function f. Since we are looking to simulate regulatory networks, a natural choice for f would thus be a network of hill functions as described in 2.7.1.3. Such a choice entails several practical drawbacks, however. Firstly, in the case of multiple inputs it is not clear how TFs interact in order to regulate a target gene combinatorially. Second, the natural unit is concentration which as we have seen before is log-normally distributed. This means that network parameter determination via optimization cannot use the typical least-squares objective. Finally the expression for the derivative of this function with respect to its parameters is not computationally efficient to calculate.

A natural simplification is to work in log-concentration units, which transforms the nonlinear function into a simple sigmoidal function with straightforward derivative and results in a normally-distributed random variable. The combinatorial issue is thornier, and unfortunately to proceed we must abandon the requirement that the network structure should represent directly the gene regulatory network topology. Instead, we shall let only the inputs and outputs directly map to actual protein (log-) concentrations and the inner layers of the network shall represent only an effective description of the dynamics.

With this in mind, we can then build an efficient computational model for reaction diffusion systems. The resulting model shall map directly into an artificial neural network model, which will then allow us to leverage recent advances in computer science to optimize it efficiently. The basic features of the model are described in Fig. 4.4. The system itself is described by a 1D lattice where the ith cell contains an n-dimensional real vector x_i, corresponding to the log-concentration of n proteins which will form the final

with 1 bit.

[7] The average number of instructions is $L/2-2$. Each instruction is 10 bit large: 2 bits for the boundaries and 8 bits to specify the rule.

pattern. Furthermore at each cell there is also defined an m-dimensional vector with the log-concentration of any external inputs y_i

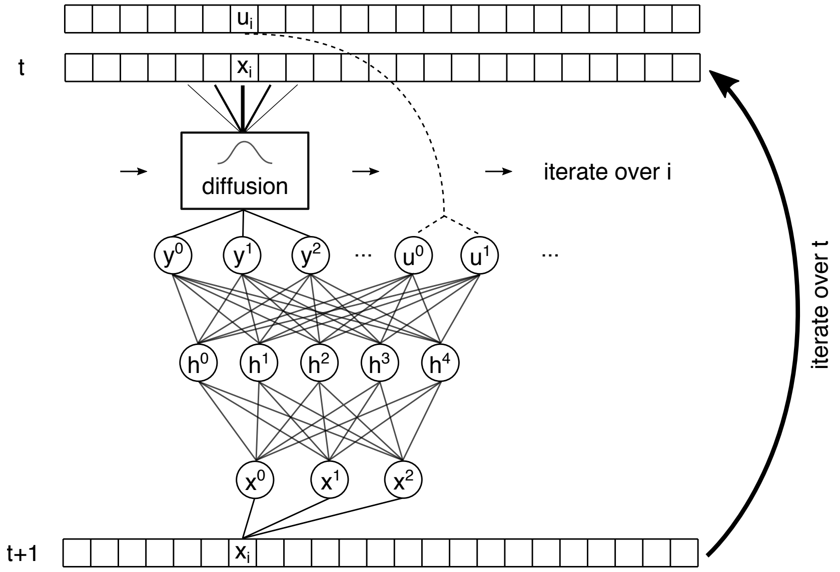


Figure 4.4: Recurrent neural network for reaction-diffusion equation modeling. The reaction variables are passed by a diffusion kernel which smoothes their values along the lattice. Then, the various smoothed species are combined with the external input u to form the concatenated vector $y \| u$. This vector is fed into a hidden neural network layer, and then this intermediate computation is fed into the output layer, which computes $x_i(t+1)$. This process is iterated T times.

At each time step, two operations must be performed for each cell in the lattice: diffusion and reaction. These two are combined into a single network as outlined in Fig. 4.4. The first layer in this network is a convolutional layer, with outputs

$$y_i^j = \sum_{k \in \mathcal{N}(i)} x_k^j K(i-k) \tag{4.3}$$

with K a gaussian kernel with a certain width σ and zero mean. The neighborhood $\mathcal{N}$ is defined purely for computational purposes: in principle the kernel would account for every cell in the lattice but if the weight of far away elements is small we can choose to cut off the computation after a fixed number of neighbors, defined by $\mathcal{N}$. K is normalized to account for this.

This convolution operation will require some attention at the boundaries, since for $\mathcal{N}/2 > i$ or $i > \mathcal{N}/2$ some values in the sum will be undefined. In the following, two types of boundary condition shall be considered: a Von Neumann type boundary condition $x'(1) = x'(L) = 0$ (no flux); and a Dirichlet boundary condition $x(i < 1) = x(i > L) = 0$ (absorbing). These conditions are easily implemented numerically by augmenting the lattice to both sides by $\mathcal{N}/2$ positions and setting those values to $x(1)$ and $x(L)$ on each side in the no flux case, or setting all values to 0 in the absorbing case.

Secondly we have the nonlinear reaction step. This is modeled with two feedforward layers which can represent any nonlinear function depending on the number of hidden units H [122]. The first layer is represented by [8]:

$$h^j = \left[1 + \exp\left(\sum_k w_h^{kj} (y\|u)^k + b_h^j\right)\right]^{-1} \tag{4.4}$$

where $y\|u$ represents the concatenation of these two vectors. The final output is given by another layer of the form above, with h as the argument and with different weight vectors:

$$f^j = \left[1 + \exp\left(\sum_k w_f^{kj} h^k + b_f^j\right)\right]^{-1} \tag{4.5}$$

The values of x at the next time step are set to f, i. e.$x(t+1) = f$. This model has the following parameters: $(n+m) \times H$ weights plus H biases in the hidden layer; $H \times n$ weights plus n biases in the output layer[9]. Note that the weights do not depend on the lattice index i: this is crucial, the network must be *equal at each lattice position.* Otherwise the dynamics learned would be trivial. Denote the concatenation of all these parameters as λ, then the network's action can be described as $f(x, u, \lambda)$ and therefore our reaction-diffusion model is written as [10]:

$$x(t+1) = f(x(t), u(t), \lambda) \tag{4.6}$$

Now, the goal shall be to find weights and biases such that the system evolves towards a given target pattern starting from a homogeneous initial condition after a number of time steps. The final pattern is thus given by:

$$p(\lambda) \equiv x(T) = f(f(...f(x(0), u(0), \lambda))) \tag{4.7}$$

Using algorithmic differentiation methods [201], it is quite simple to obtain the total derivative

$$\begin{aligned} \frac{dp}{d\lambda} &= \frac{\partial x(T)}{\partial \lambda} + \frac{\partial x(T)}{\partial x(T-1)} \frac{dx(T-1)}{d\lambda} \\ &= \frac{\partial x(T)}{\partial \lambda} + \ldots + \left(\prod_{t\in[2,T]} \frac{\partial x(t)}{\partial x(t-1)}\right) \frac{dx(1)}{d\lambda} \end{aligned} \tag{4.8}$$

[8] The i superscript is omitted henceforth.

[9] We will consider σ and $\mathcal{N}$ fixed

[10] In the artificial neural network language, we are dealing with a recurrent network

where the error is propagated across the different time intervals using the chain rule.

Given a target pattern ρ, we can use stochastic gradient descent to minimize the objective function

$$\begin{aligned}&\sum_i (p_i - \rho_i)^2 + \omega_{L_1}|\lambda|_1 \\ &+ \omega_{L_2}|\lambda|_2 + \frac{\omega_s}{T}\sum_{t\in[2,T]}(x(t)-x(t-1))^2\end{aligned} \tag{4.9}$$

where the ω parameters define the weight of the different penalties: the L_1 penalty tries to set weights to 0, while the L_2 penalty makes sure weights don't get large (intuitively large weights means that there is a strong reliance on the input data)[11]. The final smoothness penalty can be set to a small value and was implemented largely to prevent oscillating solutions.

This high-dimensional minimization is tricky, due to two factors. The first is that the parameter space might have vastly different length scales for different parameters. This problem can be solved using recent adaptive gradient descent methods which take the geometry of the parameter space into account [202, 203, 204]. The second problem is that due to the accumulation of error through the different time steps in (4.8), the error gradient contributions coming from earlier time steps can tend towards zero or infinity [203]. While there are strategies to mitigate these problems, they were not necessary in this case as T was low enough that training always completed successfully.

French flag model As an initial test to the network, we train a network to divide the embryo in 2 or 3 equal segments without an explicit input providing positional information. In the case of 2 segments, an input is only provided at the boundary. The system is able to find the middle via a traveling wave solution, where the boundary information is propagated at a precise speed such that the middle is attained at $t = T$. Naturally in this case the pattern is not a steady state, so some sort of clock mechanism is necessary to read out the profile at the correct time.

A french flag model is obtained when we attempt to divide the system into 3 segments. In this case, the input was a profile dividing the embryo in two equal segments (i. e.the output of the previous system). Again a traveling wave solution is observed. An interesting extension to the model would be to add a term to the cost function requiring that $\frac{dx}{dt} = 0$ to ensure that the final pattern would be a steady state. Such an extension was implemented for the current model but convergence was not achieved. I hypothesize this may be due to the fact that there must be some parameter specifying the length scale of the features: either explicitly via positional information from the inputs, via the diffusion constant (as we will see later), or via the system clock.

Gap gene model As a second test, let's reproduce the gap gene pattern as observed in *Drosophila*. As inputs, the Bcd, *Caudal* and *tailless* were provided. The data was

[11] The L_p norm is defined by $|x|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{\frac{1}{p}}$

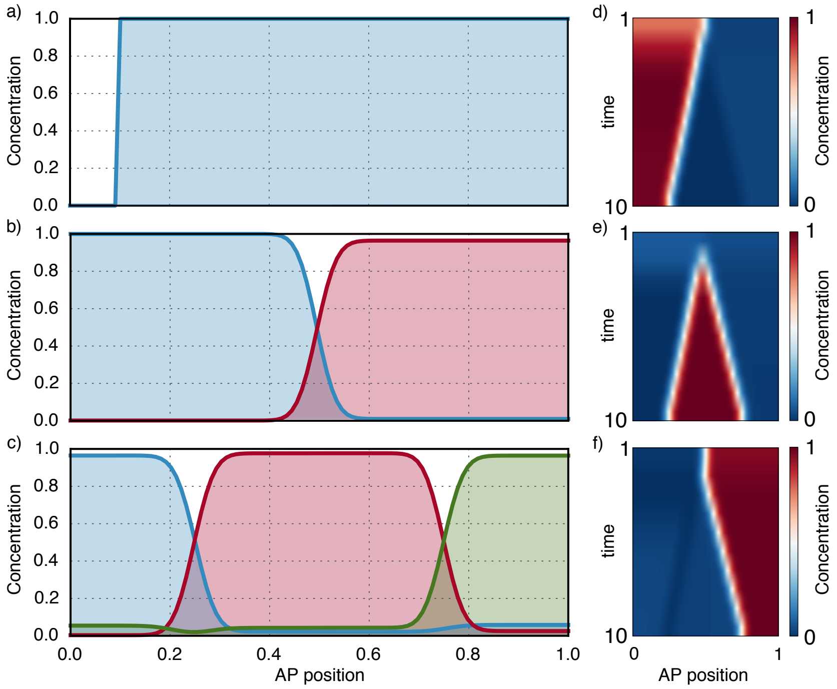


Figure 4.5: Optimization results for a segmentation network without explicit positional information input. a) Symmetry breaking input for the 2-segment problem b) Optimized output for the 2-segment problem. The simplest network which solved this problem had 3 hidden units and reflecting (no-flux) boundary conditions. The diffusion kernel was set with $\sigma = 3$ and $\mathcal{N} = 21$. Lattice size $L = 100$. c) Optimization result for the 3-segment problem, taking the 2-segment pattern as input. Network meta-parameters were the same as in b), the input was a single profile dividing the embryo in two different halves. d), e), f) Temporal evolution of the 3 proteins in c) for the optimized result. Traveling wave solution starting from a half-half division is visible.

obtained from Flyex [68]. The optimized network reproduces the target gap genes faithfully, and the dynamics resemble the expectation of a system with the features associated with canalization.

It bears noting that running the optimized network again with $\sigma << 1$ (which effectively removes communication via diffusion) destroys the obtained pattern. This implies that the network is not just directly reading its inputs at each position. Nonetheless it is possible to train a similar network with no diffusion and obtain a similarly good looking result (at the cost of a larger L_2 norm for the weights).

So how complex a pattern can be encoded in the network dynamics? To explore this question to some extent, let's try to reproduce the gap gene pattern with only a simple input dividing the embryo in two halves. In Figure 4.7 we can observe that the network is able to reproduce the large scale domains via cross repressive interactions, the length scale of which is determined by the diffusion coefficient (indeed in this case the diffusion

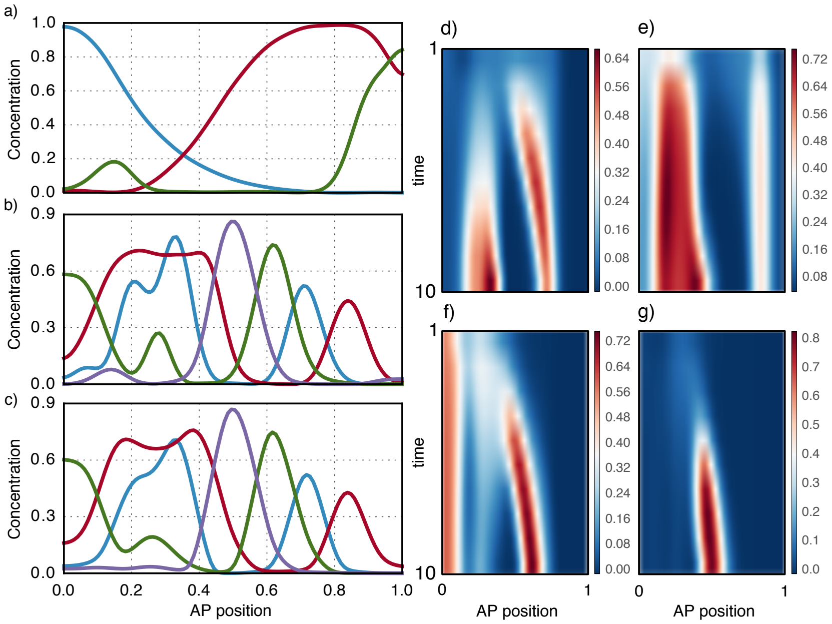


Figure 4.6: Gap gene simulation. a) Bcd (blue), *Caudal* (red) and *tailless* (green) inputs from immunostaining data. b) Hb (red), Gt (blue), Kr (purple), Kni (green) measured target profiles from immunostaining data. c) Output of optimized network with 5 hidden units, $\sigma = 5$, $\mathcal{N} = 11$ and reflecting boundary conditions. d), e), f), g) Temporal evolution of the 4 proteins for optimized result.

parameters are determinant for obtaining a good result, whereas in the previous case with positional information the result was robust to variations in diffusion constant of about one order of magnitude).

Because a relatively high diffusion constant is necessary to obtain the large feature sizes of the profiles, and there is no additional source of information providing anchoring, the smaller scale features of the profile get washed out by high diffusion and cannot be reproduced. It must be noted that the real system is 2 dimensional, and some of the small scale features here observed are not symmetrically invariant across the AP axis. Therefore we would not expect a 1D model to fully reproduce all observed features [205].

4.4 Conclusion and outlook

In this section we explored how patterns can be encoded in nonlinear dynamics of distributed systems. Taking all results into consideration, the picture that seems most consistent is that with the correct set of parameters, the network defines a flow through phase space which connects the initial (homogeneous) state to the final target state.

Since the network dynamics are described by a comparatively small set of parameters, this process can be compared to data compression of the target pattern. Naturally since

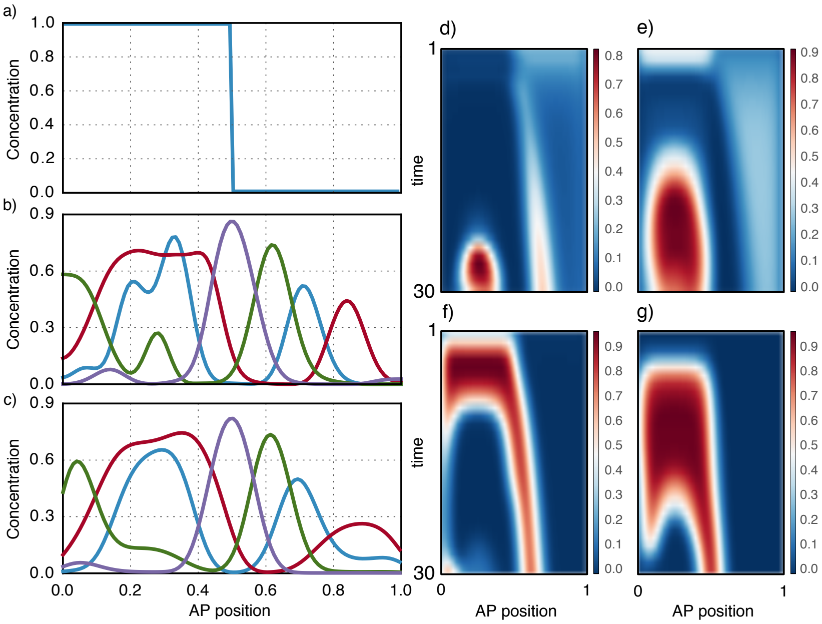


Figure 4.7: Gap gene simulation. a) Simulated input dividing embryo into two halves. b) Hb (red), Gt (blue), Kr (purple), Kni (green) measured target profiles from immunostaining data. c) Output of optimized network with 10 hidden units, $\sigma = 4$, $\mathcal{N} = 21$ and absoring boundary conditions. 30 time steps were used in this simulation. d), e), f), g) Temporal evolution of the 4 proteins for optimized result.

there are more possible patterns than combinations of parameters for simple networks, some patterns are 'simpler' than others under this framework. This idea is related to the Kolmogorov complexity of a string. Given the similarity of both concepts it is not surprising that finding the 'most compressed' representation of a pattern, or a simple network with the smallest amount of external input that can reproduce the pattern, is not a trivial problem and must be solved via a global optimization method (which is not guaranteed to converge to the global optimum).

Global optimization is necessary because the computing is parallel and distributed between the various cells in the system. Ergo, the 'program' (i. e.network dynamics) cannot perform a computation specific to each specific position. Instead, it must encode high level features of the pattern, akin to feature discovery by neural auto encoders, where each hidden unit appears to encode for one specific feature of the data [206]. In our case, we observe that network dynamics for the gap gene pattern are cross-repressive in nature, reflecting the fact that different protein spatial domains are largely disjoint in space. These dynamics, coupled to local communication between cells via diffusion result in the correct reproduction of the pattern.

Further work should focus on robustness to noise and scaling fluctuations. Intuitively emergent pattern formation models should be more robust to both sources of error since they do not rely on a fixed positional information source. On the other hand, some amount of positional information might be necessary to achieve features at different length scales. Time varying inputs would also be an interesting extension, as it has been hypothesized that networks might extract useful information from such inputs [207, 134, 135], and in addition they might allow for more compact network representations since the network does not need to fully specify the correct trajectory through phase space *a priori*.

Chapter 5

Collective decision-making in bacterial colonies

5.1 Motivation

To survive, an organism must acquire information about its surroundings and react appropriately. The choice of action to take given the currently available information is known as *decision making*. Cellular decision making underlies processes at all levels and is key to development in higher level organisms [208]. However, there are several tradeoffs to be made when committing to a decision in an uncertain environment [209, 210].

Multiple strategies can be implemented to mitigate risks in decision making [211], one of which is to rely on distributed inference and decision making for additional reliability. An important step towards understanding distributed decision making in biology is to completely characterize cell-to-cell communication protocols. The simplest model on which to do this is population-level decision making in single-celled organisms.

As described in 1.3, bacteria communicate using quorum sensing. The resulting external chemical is read by the cells and fed into a bistable gene regulatory circuits which sets cells in one of two states [212, 213] leading to a heterogeneous population. How robust to noise and fluctuations is this mechanism and what is the best strategy to communicate under such conditions?

5.2 Previous work

The model system here considered is based on a synthetic sender-receiver system implemented in *E. Coli* via plasmids [49]. The system is split into sender and receiver components, which allow for investigation of the properties of each component in isolation. The system operation is outlined in Figure 5.1.

Single cell analysis for noise quantification was popularized by Elowitz in the early 2000's [107]. This technique leverages advances in computing power and automatization in microscopy to analyze large numbers of cells directly from microscopy images, thereby

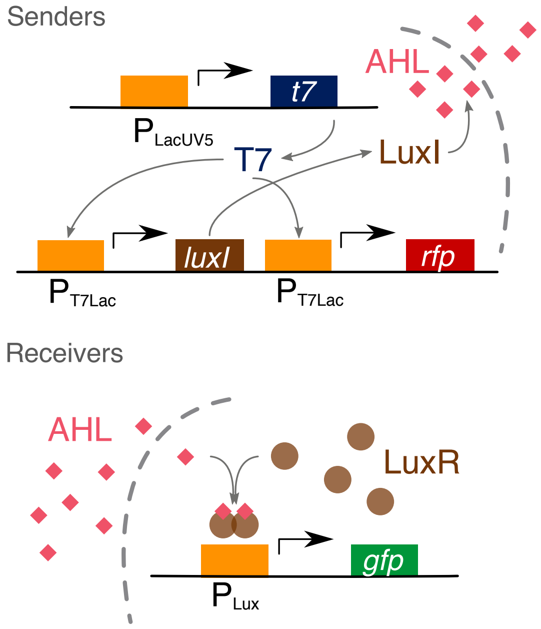


Figure 5.1: The sender-receiver system as implemented [49]. Top, sender component. T7 RNA polymerase is expressed from the LacUV5 promoter, induced by IPTG. T7 goes on to induce expression of AHL synthase LuxI. After synthesis, AHL diffuses into the extracellular medium. Bottom, receiver circuit. AHL permeates through the cell membrane and binds to LuxR. LuxR:AHL dimers bind to the Lux promoter, thereby expressing GFP.

reconstituting the probability distribution of the concentration of the induced gene as a function of time.

The general procedure is well established [214]: computer vision algorithms are used to **segment** cells from brightfield or phase contrast microscopy images; cells are followed in time via frame-by-frame **tracking**; and finally **fluorescence extraction** provides the actual values to be used in the analysis.

Several existing packages already perform this sequence of tasks, such as CellProfiler [215], CellTracer [216] or CellCognition [217]. Our goal was to strike a balance between accuracy, performance and ease of use. To do so, we developed a custom package implementing some of the best features of each of these applications.

5.3 Results

5.3.1 Image analysis and extraction of single cell data

In this section I will detail the general image analysis pipeline as implemented in the software and discuss tradeoffs in simplicity versus performance.

The software is a self contained package with a graphical user interfaced programmed in C++ using the Qt framework to provide graphical features. The presentation code is kept separate from the image analysis code which uses primarily the OpenCV library to implement the various image analysis functions. The workflow consists of a fixed image analysis pipeline, where the user imports an image stack consisting of microscopy images acquired at various time points. Then the user can step through the processing pipeline, adjusting parameters until segmentation results are adequate for all times.

It is also possible to import one or more fluorescence stacks, which are assumed to be aligned with the brightfield stack used in the segmentation workflow. Finally the analysis results can be exported for all times to a plain text document. Below I outline the major pipeline steps.

Image preprocessing The first step is image preprocessing. For efficiency 16-bit brightfield images are down converted to 8-bit[1] via a linear rescaling procedure. To avoid dynamic range loss, the user can manipulate the parameters of the rescaling transform.

The user may then define a subsection of the image stack to analyze which is then enhanced by applying well known filters: contrast enhancement, noise reduction, sharpening and resolution increase. Contrast is enhanced by calculating the brightness histogram; discarding all brightness values beneath a user-defined threshold and rescaling the brightness values such that the histogram now spans the full brightness range.

Noise reduction is accomplished by the nonlocal means filter implemented in the OpenCV library [218]. The image is sharpened by subtracting the lowpass filtered image. Finally, it is possible to double or quadruple the image resolution at this step using bicubic filtering. Since some subsequent pipeline steps are applied pixel wise, higher resolution might entail better accuracy.

Background detection We found that optimal performance was achieved by combining the output of several simple thresholding methods. First, a global brightness threshold may be set by the user. Second, a local brightness threshold is used, where a pixel is marked as background if its brightness is above the local brightness average by a pre specified amount. The local average is computed around a window of pre-specified size which should match the typical length scale of the smallest axis of a cell.

Finally, large areas with no cells which might elude the previous two methods are detached by employing a method similar to Wang *et al.* [216], where prominent edges are detected and enlarged to outline the probable location of the cells; the complement of that area is marked as background.

Cell detection Given specified maximum bounds on width and height, cell markers are created by a multi-step procedure B.2. Initially all connected regions of non-background pixels are assigned to a unique marker. Each region too big to be a single cell is then further segmented based on the brightness profile and cell geometry.

[1] This does not apply to fluorescence images, where all accuracy is preserved.

A likelihood for whether each pixel is part of the inside of a cell is assigned. This takes into account the the brightness, brightness gradient, and cell geometry (by using the distance transform). New markers are then assigned to connected regions of pixels above a certain likelihood threshold. This threshold is chosen to be the minimum value such that the regions of all markers obey the specified bound. The resulting regions are then expanded and refined using the watershed algorithm [219].

Cell classification Often, objects which are not cells are present in the microscopy image but are still detected as such due to similar brightness profiles as real cells. To overcome this problem, an optional final step of the operation pipeline entails the training of a support vector machine (SVM) to distinguish interesting cells from such outliers. This method has previously been used to distinguish cell phenotypes with success [220].

The process begins by taking each connected area of segmented pixels and calculating features summarizing its geometry and brightness, which will become a high dimensional data point d_i for the training set. Each point is selected by the user, who also marks each label as correct or incorrect directly on the user interface thereby associating each datapoint with a class $\mathcal{C} \in 0, 1$. The SVM algorithm then applies a nonlinear transformation to the feature space such that a hyperplane separating the two classes of points can be found. The implementation found in libSVM [221] has been used here. By visual inspection the user may validate the results and edit the training data to avoid over or under fitting.

Lineage tracking In order to track cells in time, a frame by frame tracking procedure was adopted. In each frame we determine a cell's parent by calculating the overlap between its assigned pixels and the pixels of detected cells in the previous frame. The cell label from the previous frame which maximizes this overlap is then set as the parent. If l_i^t is a boolean vector where pixels are marked as 1 if they belong to the ith label at frame t and 0 otherwise; the parent label for that label is defined as

$$p_{i,t} = \underset{j \in \text{labels}(t-1)}{\arg\max} \sum_{\text{pixels}} l_j^{t-1} l_i^t \ .$$

This method was compared to the minimization of the distance between the center of mass of a cell and those of its ancestors. If c_i^t is a 2d vector containing the center of mass of the ith label at frame t, then

$$p_{i,t} = \underset{j \in \text{labels}(t-1)}{\arg\min} \ c_j^{t-1} . c_i^t \ .$$

Empirically the maximum overlap method performed better than the center of mass distance for all datasets we tried. As before, the user can inspect the generated trajectories and manually correct the lineage in case of error via a graphical user interface. This allowed us to automatically extract lineage data and single cell fluorescence trajectories as a function of time.

5.3.2 Experimental results

5.3.2.1 Senders only

The experiments were designed and performed as described in [238]. Below I will present a detailed overview of the data analysis procedure.

Fig. 5.2 shows the analysis done on the data at the colony level. This coarse analysis is a zeroth-order approximation to the process which will be used as a benchmark. Fig. 5.2 A shows the total area $A(t)$ measured in pixels of all cells in the microfluidic chamber corresponding to a unique experiment. This measurement will be taken as a proxy for total cell mass M. Fig. 5.2 B shows the total integrated fluorescence $P(t)$ for all cells.

For modeling purposes, we will assume the total cell mass grows as $\dot{M} = \mu M$. A single cell will produce GFP according to $\dot{p} = \alpha - \mu p$. For simplicity let us also assume that total fluorescence is given by $P = p \times M$. In that case, using the chain rule we see that

$$\alpha = \dot{P}/M \, . \qquad (5.1)$$

In Fig. 5.2 C we see the maximum of this induction rate plotted as a function of AHL concentration. It appears to follow a Hill shape.

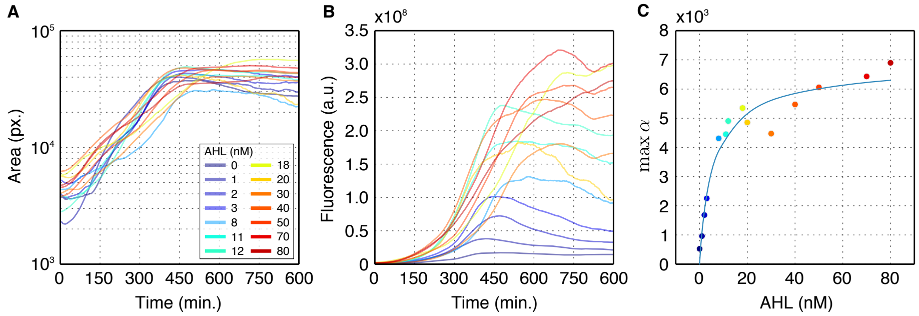


Figure 5.2: Bulk level measurements for the population in the microfluidics chamber. Different lines reflect different experiments with various levels of AHL in the medium. a) Total cell area as a function of time, a proxy for cell mass. Note exponential cell growth until ~ 450 min. b) Total GFP fluorescence as a function of time. c) Maximum of induction α as calculated according to (5.1). Hill fit with parameters $n = 0.95 \pm 0.2$ and $K = 5.3 \pm 1.4$ nM.

A probability distribution of the fluorescence can be recovered as in Fig. 5.3 B. This is a clear log-normal distribution with some outliers, which arise due to some lineages which were induced later (trajectories of which can be seen in Fig. 5.3 A). To clean up the statistical results, we want to focus only on the main induction trajectory. To filter out outliers, a gaussian mixture model[2] was fit to the log of the fluorescence data (red and green Fig. 5.3 B). The main component retrieved corresponds to the desired group of cells.

Using the data from the main induction component, we can plot the squared coefficient of variation as a function of time (variance over mean squared, Fig. 5.3 C). This data

[2] A gaussian mixture model with N components is determined by a set of gaussians $\mathcal{G}$ with means μ_i

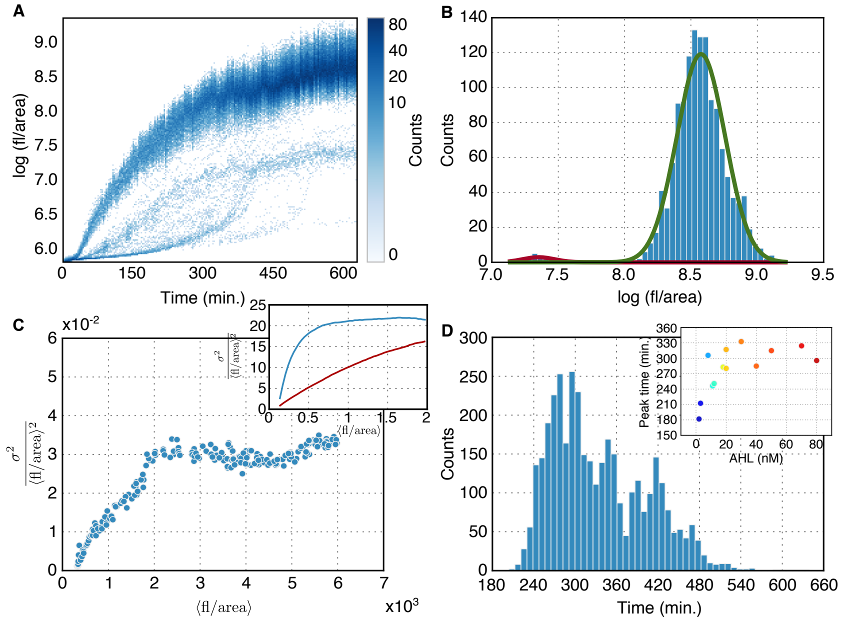


Figure 5.3: Single cell gene expression histogram and trajectories for AHL$= 50nM$. (A) Evolution of the histogram $\log p$ as a function of time. Clusters of 'late inducers' are visible. (B) Histogram of $\log p$ at time $t = 200$ (frames). Solid lines represent the two gaussian distributions resulting from the gaussian mixture fitting procedure used to single out the main lineage. (C) $\sigma^2/\langle p \rangle^2$ as a function of $\langle p \rangle$. Inset: blue, simulations as in (5.2) with $h(t) = \frac{t}{r}\Theta(t-r)$; red with $h(t) = 1$. (D) Time at which cells reach half induction ($\log p = 7.94$) Inset: Peak mean induction as a function of AHL.

hints that even the main induction trajectory might be displaying the effects of induction heterogeneity, as the fluctuations rise more quickly than expected in the beginning, only to level off. To investigate this behavior, I propose a simple model for GFP induction given by a hill type production rate and degradation:

$$\frac{dc}{dt} = h(t)\, a\, \frac{[\text{AHL}]^n}{[\text{AHL}]^n + K^n} - \lambda c + c\eta \tag{5.2}$$

where a is the overall production rate; λ is the GFP degradation/dilution rate and $h(t)$ characterizes heterogeneity in induction. A simple model for a heterogeneous system is

and covariance matrices Σ_i weighed by w_i:

$$P(m) = \sum_i^N w_i \mathcal{G}(\mu_i, \Sigma_i)$$

$h(t) = \frac{t}{r}\Theta(t-r)$, where r is a uniformly distributed random variable $\in [0,1]$ and

$$\Theta(x) = \begin{cases} 0 & \text{if } x > 0\,, \\ 1 & \text{otherwise.} \end{cases} \tag{5.3}$$

Stochastic differential equations were solved with the stochastic fourth order Runge Kutta method (B.1, [89]). η denotes a gaussian random variable with zero mean and standard deviation σ. It is possible to qualitatively recover the noise behavior as in Fig. 5.3 C using the following parameters: $n = 0.95$, $k = 5$, $\lambda = 0.7$, $a = 3$, $\sigma = 0.1$.

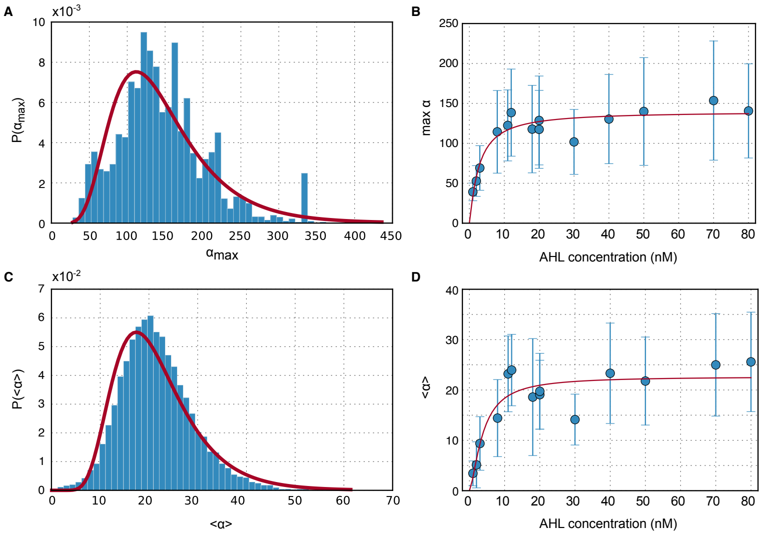


Figure 5.4: Distribution of gene expression rates α. a) histogram of $\log \alpha' \equiv \log \max \alpha(t)$ for each single cell trajectory (AHL= $50nM$); lognormal MLE fit to $P(\alpha')$; b) plot of average and standard deviation of the distribution $P(\alpha')$, hill regression c) histogram of $\log\langle\alpha(t)\rangle$ for each single cell trajectory (AHL= $50nM$); lognormal MLE fit to $P(\langle\alpha(t)\rangle)$; d) plot of average and standard deviation of the distribution $P(\langle\alpha(t)\rangle)$, hill regression;

Single cell analysis was performed by importing the automatically generated lineages from the segmentation and applying a basic heuristic to fix any tracking errors: if a cell changes fluorescence by more than 10% between two frames, it is assumed to have been missegmented (this margin was determined empirically by observation of typical fluctuation scales). In that case, cells nearby are queried and their fluorescence is compared to the parent's prior fluorescence. If any is found with closer fluorescence, it is reassigned as daughter. If none is found, the original daughter is preserved.

In the single cell trajectories we can calculate α as a function of time using (5.1). We again determine maximal induction $\max_{t\in[1,T]} \alpha$ and observe it is log-normally distributed (Fig. 5.4 A). Again it is possible to fit a Hill curve to the points (now the points correspond to the average value of the distribution) as a function of AHL concentration. The same analysis was repeated for the average value of α, with similar results.

5.3.2.2 Senders and receivers

The same experiment was repeated, this time with both senders and receivers. Different ratios of senders and receivers were pipetted into the chamber to ensure different induction levels. Can we use a theoretical model to determine how much AHL is being deposited in the microchamber by the receivers?

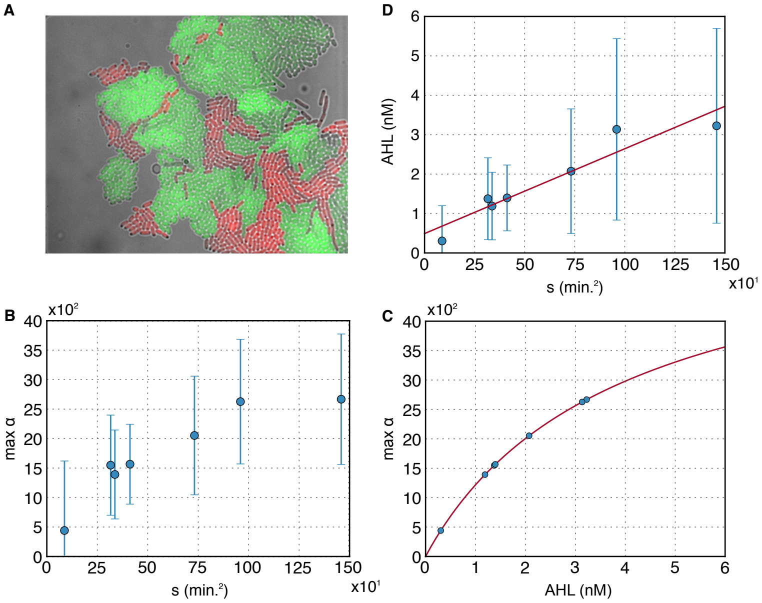


Figure 5.5: a) microfluidic chamber showing senders and receivers; b) max GFP induction for 3 select experiments as a function of the effective AHL production constant $s = \langle r \rangle t_{\max}^2$; c) determination of effective AHL concentration for the 3 experiments using previously acquired calibration curve; d) linear fit following equation (5.6), error bars via error propagation

Let us first propose a simplified model for AHL production in the system. The sender cells produce LuxI according to the law:

$$[\text{LuxI}]'(t) = rN(t)\alpha_l - \lambda[\text{LuxI}](t) \tag{5.4}$$

where r is the sender-receiver ratio, N is the total number of cells in the system, and α_l and λ are production and degradation rates. AHL is produced from LuxI, we neglect its decay, and we assume a constant outflow C via the chamber exit:

$$[\text{AHL}]'(t) = [\text{LuxI}](t)\alpha_a - C\,[\text{AHL}](t) \tag{5.5}$$

From these equations and assuming $N(t) = Ae^{\gamma t}$ we obtain that total AHL at time t is given to first approximation by (assuming γ, λ and C are small): $[\text{AHL}](t) \simeq 1/2\,\alpha_a\alpha_l Art^2$. Intuitively, this can be understood as follows: at each time t there will a number of sites producing AHL proportional to rt (due to the population growth). Integrating this with respect to time we get AHL production proportional to $1/2\,rt^2$. The missing proportionality constant depends on the AHL and LuxI production rates and population growth rate $\alpha_a \alpha_l A$.

The sender-receiver ratio r and the time at which AHL concentration is measured t are not constant between experiments. For each experiment we will fix the measurement time t to be the time at which senders are maximally induced (t_{max}). The sender-receiver ratio will be estimated from the experiment by averaging $r(t)$ in the interval $[0, t_{\text{max}}]$. We can bundle these experiment dependent parameters in a variable $s = \langle r \rangle t_{\text{max}}^2$. The final calibration curve we seek will then be given by:

$$[\text{AHL}]_{\text{eff.}} = \frac{1}{2}\alpha_a \alpha_l A s \tag{5.6}$$

Under the assumption that AHL diffuses (estimates range from 100 to 1000 $\mu m^2/s$ [222, 223, 224, 9]) quickly through the micro chamber such that each cell reads a concentration $\propto$ AHL, GFP induction (which can be calculated the same way as in the previous section) is given by the Hill equation:

$$\alpha \equiv [\text{GFP}]'(t) = \alpha_g \frac{[\text{AHL}]^n(t)}{[\text{AHL}]^n(t) + K_g^n} \tag{5.7}$$

On the one hand we can match α to a given y from each experiment (Fig 5.5 B). On the other α can be matched to an effective AHL via Eq. (5.7) (Fig 5.5 C). Putting those two elements together we have the points necessary to fit Eq. (5.6) (Fig 5.5 D).

5.4 Conclusion and outlook

With this work we demonstrated the power of single-cell analysis to quantify heterogeneity in stochastic systems. Using the microscopy image analysis program I developed, we were able to extract quantitative information on a sender-receiver system, quantify heterogeneity in induction times, and construct a sender-receiver calibration curve.

In the future we expect other laboratories and experiments [225] to use the open-source image analysis program here detailed to quantify heterogeneity in other experimental setups.

Chapter 6

Evolution of stochastic networks

6.1 Motivation

Given current high-throughput techniques to measure expression data, it has become possible to infer certain regulatory network topologies using advanced statistical techniques [226, 227]. Towards this end, a common scheme is to calculate the probability of the model parameters describing the reaction network given some model using Bayes' theorem:

$$P(a|d) = \frac{P(d|a)P(a)}{P(d)} \tag{6.1}$$

where d represents a time series measurement of expression data; and a corresponds to a specific model's parameters. To recover the likelihood $P(d|a)$, it is necessary to run a stochastic forward simulation of the model given parameters a. Assume the model is given in general by a Langevin equation of the type:

$$\dot{c}(t) = \frac{d}{dt}\, c(t) = f(c(t), a) + \eta(t) \;, \tag{6.2}$$

where f is an arbitrary function depending on a set of parameters denoted by the vector a and c is the log concentration of the chemical reactants. η represents uncorrelated, zero-mean white noise (but with a possibly time dependent variance). Following a maximum entropy approach [75, 76, 77], we can describe the average behavior of the population with a gaussian distribution:

$$P(c, t|a) = \mathcal{G}(c - \bar{c}(t), C(t)), \tag{6.3}$$

$$\mathcal{G}(c - \bar{c}, C) = \frac{1}{\sqrt{|2\pi\, C|}} e^{-\frac{1}{2}\,(c-\bar{c})^{\dagger} C^{-1} (c-\bar{c})}. \tag{6.4}$$

Here, $|C| = \det(C)$ and $c^{\dagger} x = \sum_{i=1}^{N} c_i\, x_i$ is a scalar product over the space of network state vectors. If we can calculate an approximation to this gaussian representation of the network state at time t, then the measured data can be described as a measurement process with measurement noise σ:

$$d_j = \int \mathcal{D}c\, e^{c_j}\ \mathcal{G}(c - \bar{c}(t), C(t)) + \sigma_j = e^{\bar{c}_j + \frac{1}{2}C_{jj}} + \sigma_j \tag{6.5}$$

where we integrate over all cells after marginalizing over all proteins except the jth. This measurement process can be described as another Gaussian distribution $\mathcal{G}(d_j(t) - e^{\bar{c}_j(t)+\frac{1}{2}C_{jj}(t)}, \sigma_j^2)$. With this, we can write the likelihood for the data $P(d|a)$:

$$P(d|a) = \prod_{j,t} \mathcal{G}(d_j(t) - e^{\bar{c}_j(t)+\frac{1}{2}C_{jj}(t)}, \sigma_j^2) \tag{6.6}$$

Following the idea in 2.4.2 we can write a Hamiltonian:

$$H(d, a) = -\log P(d|a) - \log P(a) \tag{6.7}$$

where $P(a) = \mathcal{G}(a - \bar{a}, \Lambda)$ is a prior for the parameters of the network to be inferred. Given this, the MAP approximation for the posterior can be calculated. A gradient descent method can be used with the expression:

$$\frac{dH}{da}(d, a) = \Lambda^{-1}(a - \bar{a}) + \frac{d}{da}\sum_{j,t} \frac{1}{2\sigma_j^2}(d_j(t) - e^{\bar{c}_j(t)+\frac{1}{2}C_{jj}(t)})^2 \tag{6.8}$$

with the second term given by:

$$-\sum_{j,t} \frac{1}{\sigma_j^2}(d_j(t) - e^{\bar{c}_j(t)+\frac{1}{2}C_{jj}(t)})\ e^{\bar{c}_j(t)+\frac{1}{2}C_{jj}(t)} \left(\frac{d\bar{c}_j}{da} + \frac{dC_{jj}}{2da}\right)\bigg|_t \tag{6.9}$$

Given this machinery, the missing piece is a method to quickly evaluate an approximation for Eq. (6.3) which additionally allows us to calculate the derivative of the time evolution with respect to the parameters a.

Such an approximation should result in dynamic equations (denoted by f) for the mean and covariance of Eq. (6.3):

$$\dot{\bar{c}}(t) = f_{\bar{c}}(\bar{c}(t), C(t)) \tag{6.10}$$

$$\dot{C}(t) = f_C(\bar{c}(t), C(t)) \tag{6.11}$$

In this case, derivatives can be obtained via (below for $f_{\bar{c}}$, but analogously for f_C):

$$\frac{d\bar{c}}{da}\bigg|_t = \int_{t_0}^{t} \frac{d\dot{\bar{c}}}{da} = \int_{t_0}^{t} \frac{df_{\bar{c}}}{da}\bigg|_{\bar{c},C} \tag{6.12}$$

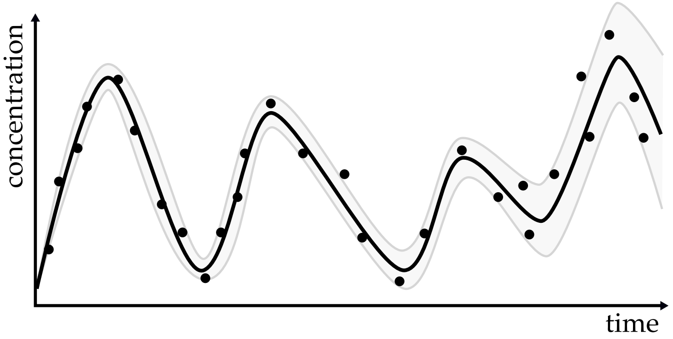


Figure 6.1: Noisy samples are extracted from a real system (dots). A gaussian model for the temporal evolution of the system is fully defined by the temporal mean (solid black) and the covariance matrix (standard deviation around the mean pictured, shaded gray). The MAP parameter estimate is such that the gaussian model is maximally consistent with the data.

6.2 Previous work

Multiple forward simulation techniques allow us [228, 208, 100, 229] to recover the probability distribution describing the population state at a given point in time. Such forward simulations can be run at different levels of detail, as described in Chapter 2 and in [99, 87, 88].

A fast forward simulation technique for chemical Master equation is the recent spectral method [230]. This method calculates the joint distribution of small uncoupled modules in the network by converting the Master equation into an easier algebraic equation. This is achieved by using the eigenfunctions of the Master equation as basis functions of a new space where each dimension evolves under an independent birth-death process. Once these algebraic equations are solved, the joint distributions can be pieced together to recover the full distribution of the system. One issue with this method is that it does not easily cope with feedback in the network, meaning only relatively simple networks can be handled.

Still in the realm of chemical Master equations, Gillespie type methods can be accelerated resorting to approximations such as Tau-leaping [90]. One can also find efficient sampling techniques for networks where certain events are rare [231, 232] in the literature. Alas, Gillespie type simulations do not allow us to calculate derivatives with respect to the parameters and are therefore unfit for our purposes.

Looking at continuous systems, from Eq. (6.2) we can derive a FPE of the form [87, 88][1]:

$$\partial_t P(c,t) = -\partial_c \left[f(c,a)\, P(c,t) \right] + \tfrac{1}{2}\partial_c^2 \left[\mathcal{X}\, P(c,t) \right] \ , \tag{6.13}$$

where $\mathcal{X}$ represents the covariance of the random process denoted by η in Eq. (6.2). Note that $\mathcal{X}$ generally depends on c.

Using this representation, a common fast forward simulation method is the linear noise approximation [87, 88]. In that case, the evolution of the covariance is determined locally

[1] Refer to Section 2.5 for more details

via the Jacobian J of the function f at the current average concentration $\bar{c}(t)$,

$$\begin{aligned}\dot{\bar{c}}(t) &= f(\bar{c}) \ , \\ \dot{C}(t) &= J(t)C(t) + C(t)J^T(t) + \mathcal{X}(t) \ . \end{aligned} \tag{6.14}$$

This method fulfills the aforementioned criteria, in the sense that we recover a Gaussian description of the system state; can take the derivative of the evolution of the system with respect to the parameters, and is fast to numerically simulate.

Recently an accurate approximation scheme for partial differential equations was developed [233] by invoking the Kullback-Leibler divergence [234] as accuracy measure. Here, it will be applied to create a simulation scheme for the SDE dynamics in the spirit of the linear noise approximation.

6.3 Results

6.3.1 Proposed method

Suppose our model evolves according to Eq. (6.2) with an initial distribution

$$P(c(0)) = \mathcal{G}(c - \bar{c}(0), C(0)) \tag{6.15}$$

Suppose $\eta = 0$; in that case the probability distribution would evolve according to:

$$P_d(c') = \mathcal{G}(c - \bar{c}(t), C(t))|_{c=c'-\delta t \dot{c}} \left| \frac{dc}{dc'} \right| \tag{6.16}$$

where $c(t + \delta t) \equiv c' = c + \delta t \dot{c}$. This is just a change of variables in the probability distribution. Connecting to Eq. (6.2), we have that $|dc'/dc|_{c=c'} = |1 + \delta t \ df/dc|_{c=c'}$.

To take into account the noise, we sum the noise to probability distribution 6.16, which corresponds to a convolution:

$$P(c(t + \delta t)) = \int d\hat{\xi} P_d(c' + \hat{\xi}) P(\hat{\xi}) \tag{6.17}$$

Now if we approximate eq 6.16 by a gaussian (next section on how to do this) we can write:

$$\begin{aligned} P(c') &= \int d\hat{\xi} \mathcal{G}(c' - \bar{c}' + \hat{\xi}, C') \mathcal{G}(\hat{\xi}, \mathcal{X}) \\ &= \mathcal{G}(c' - \bar{c}', C' + \mathcal{X}) \end{aligned} \tag{6.18}$$

Let $C'' = C' + \mathcal{X}$.

As an example of noise in deterministic example, let's work out what happens when real concentrations exhibit poissonian noise and we wish to convert that to log concentrations $\delta c = \frac{\delta n}{n} = \frac{1}{\sqrt{n}} = e^{-\frac{1}{2}(\bar{c}_j(t)+\frac{1}{2}C_{jj}(t))2}$.

$$\begin{aligned}
\mathcal{X}_{lk} &= \langle \xi_l' \xi_k' \rangle = \delta_{lk} \int_0^{\delta t} dt'\; dt''\; \langle \xi_l(t') \xi_k(t'') \rangle \\
&= \delta_{lk} \int_0^{\delta t} dt'\; dt''\; \delta(t' - t'') \sigma_0^2 e^{-\frac{c}{2}} \\
&= \delta_{lk}\; dt\; \sigma_0^2\; e^{-\frac{1}{2}(\bar{c}_j(t)+\frac{1}{2}C_{jj}(t))}
\end{aligned} \tag{6.19}$$

6.3.1.1 Kullback Leibler matching

At each time step, the distribution in Eq. 6.16 will no longer be accurately described by a gaussian. The key idea as in [233] is to find a time evolved Gaussian parametrizes by $\bar{c}'$ and C' such that it minimizes the KL distance to the time evolved distribution 6.16:

$$\begin{aligned}
D(\mathcal{G}'; P') = &- \int dc'\; \mathcal{G}(c' - \bar{c}', C') \log \frac{\mathcal{G}(c' - \bar{c}', C')}{P'(c')} \\
&- \left\langle \log \frac{\mathcal{G}(c' - \bar{c}', C')}{P'(c')} \right\rangle_{c'}
\end{aligned} \tag{6.20}$$

Defining $D(\mathcal{G}'|P') \equiv S(\bar{c}', C')$, since the Kullback-Leibler divergence is now a function only of the parameters of the Gaussian. In Section A.1.1 it is shown that this reduces to:

$$\begin{aligned}
&S(\bar{c}', C') \simeq \frac{1}{2} \log \left(\frac{|C|}{|C'|} \right) + \frac{1}{2} \mathrm{tr} \left[C'C^{-1} - \nVdash \right. \\
&\left. + \delta t (2 - C^{-1}C' - C'C^{-1}) \left\langle \frac{df}{dc} \right\rangle_{c'} \right] \\
&+ \frac{1}{2} (\bar{c}' - \bar{c})^T C^{-1} (\bar{c}' - \bar{c}) - \delta t (\bar{c}' - \bar{c})^T C^{-1} \langle f(c') \rangle_{c'} \,,
\end{aligned} \tag{6.21}$$

where $\langle \cdot \rangle_{c'}$ denotes the expectation value with respect to $\mathcal{G}'(c') = \mathcal{G}(c' - \bar{c}', C')$. The desired values of $\bar{c}'$ and C' are obtained via minimization of Eq. (6.21), i.e. by setting the derivative of S with respect to $\bar{c}'$ and C' to zero:

$$\frac{dS}{d\bar{c}'} = 0 \,,\; \frac{dS}{dC'} = 0 \tag{6.22}$$

The solution of this (refer to A.1.2), leads to

$$\begin{aligned}
&\bar{c}' \simeq \bar{c} + \langle f(c) \rangle \delta t \\
&C' \simeq C + \left\langle \frac{df}{dc} \right\rangle_c C\, \delta t + C \left\langle \frac{df}{dc} \right\rangle_c^T \delta t \,.
\end{aligned} \tag{6.23}$$

[2] Extrinsic, or log-normal noise is simpler as it corresponds to $\mathcal{X} = \mathrm{ct}$.

Taking the limit $\delta t \to 0$ in conjunction with the intrinsic noise from Eq. (6.18), the final result is obtained:

$$\begin{aligned} \dot{\bar{c}} &= \langle f(c) \rangle \\ \dot{C} &= \left\langle \frac{df}{dc} \right\rangle_c C + C \left\langle \frac{df}{dc} \right\rangle_c^T + \mathcal{X} \ . \end{aligned} \tag{6.24}$$

In general, the Gaussian expectation value of a nonlinear function is not analytically accessible in a closed form. A taylor expansion of $f(c)$ around $\bar{c}$,

$$f(c) = \sum_{n=0}^{\infty} \frac{1}{n!} \left. \frac{d^n f}{dc^n} \right|_{c=\bar{c}} (c - \bar{c})^n \ , \tag{6.25}$$

results in the expression:

$$\begin{aligned} \langle f_i \rangle_c &= f_i(\bar{c}) + \frac{1}{2} \left. \frac{d^2 f_i}{dc_k \, dc_l} \right|_{c=\bar{c}} C_{kl} + \dots \\ \left\langle \frac{df_i}{dc_j} \right\rangle_c &= \left. \frac{df_i}{dc_j} \right|_{c=\bar{c}} + \frac{1}{2} \left. \frac{d^3 f_i}{dc_j \, dc_k \, dc_l} \right|_{c=\bar{c}} C_{kl} + \dots \ , \end{aligned} \tag{6.26}$$

which reduces to (6.25), the linear noise approximation (6.14) in first order.

6.3.2 Test applications

To illustrate the benefits and limitations of our method, we applied it to several test cases. The predictions of our method are compared to stochastic simulations of the system dynamics using the Euler-Maruyama scheme [89].

6.3.2.1 Linear repressilator

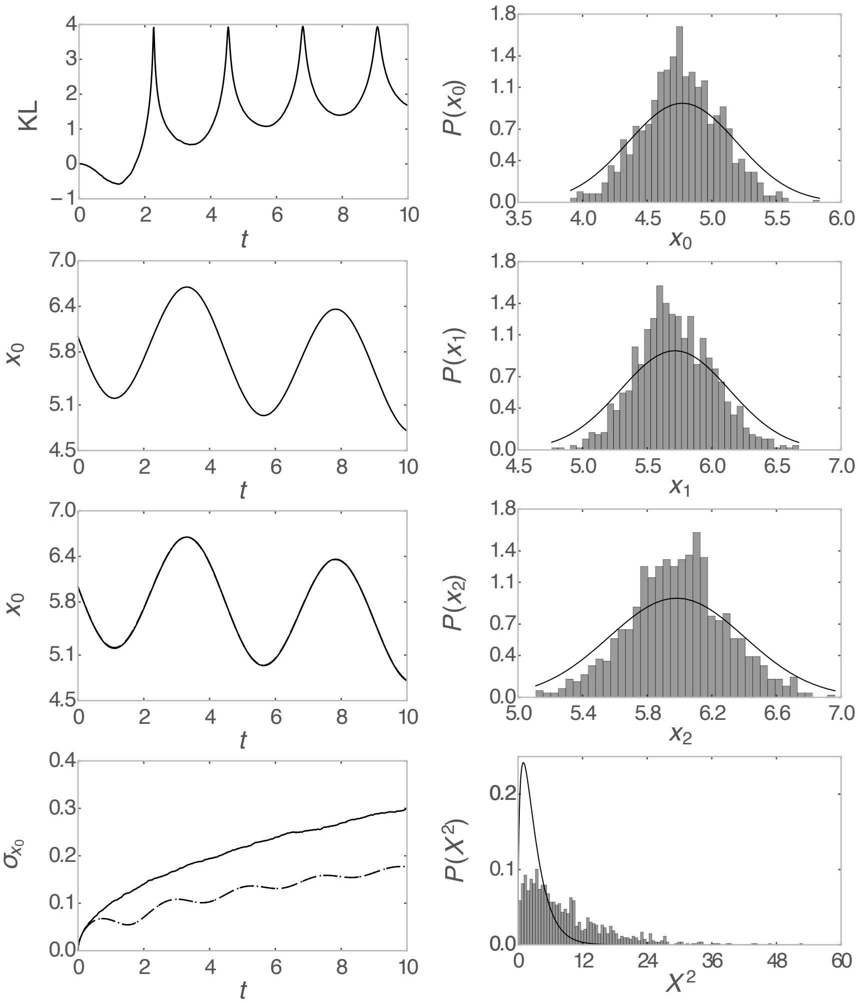
The initial experimental setup was the simulation of a three way genetic oscillator:

$$\dot{c}(t) = Ac(t) + \eta(t) \ , \tag{6.27}$$

with

$$A = \begin{pmatrix} -0.01 & 0.8 & -0.8 \\ -0.8 & -0.01 & 0.8 \\ 0.8 & -0.8 & -0.01 \end{pmatrix} \tag{6.28}$$

For an initial condition of $c(0) = \{6.0, 5.4, 6.8\}$, $N = 1000$ trajectories were calculated, and the mean and variance were calculated using the usual formulas $\bar{c}(t) = \frac{1}{N} \sum_i^N c_i(t)$ and $\sigma^2(t) = \frac{1}{N} \sum (c_i(t) - \bar{c}(t))^2$. Even in this simple case, the method is confronted with a difficulty: whenever one of the chemical species reaches a peak in its trajectory, the distribution momentarily becomes non-Gaussian. Therefore at this time the approximation becomes insufficient and therefore there is a loss of precision, as evidenced by the KL divergence oscillations in Fig. 6.2a, top.



(a) Trajectories (b) Histograms

Figure 6.2: Test application to a system of linear oscillators in the. Top panel, left: KL divergence between simulated trajectories and prediction, as per Eq. (6.20). First middle panel, left: Average of stochastic simulation with 1000 runs of system (6.27). Second middle panel, left: Comparison of stochastic average value (solid) with prediction using Eq. (6.24) (dashed) and Eq. (6.14) (dotted). Here both predictions are the same. Bottom panel: Comparison of standard deviation for the stochastic simulation (solid) with prediction using Eq. (6.24) (dashed) and Eq. (6.14) (dotted); again both predictions coincide. Right: Histogram of stochastic simulation with 1000 runs of system (6.27) and the corresponding Gaussian (solid line) predicted by our method (6.24) at the final time $t = T$ (in this case $T = 10$). The first three panels show the distributions of x_0, x_1, and x_2, respectively. Bottom panel, right: histogram of chi squared distribution for the stochastic simulation data compared to the χ^2 distribution with 3 degrees of freedom.

6.3.2.2 Van der pol oscillator

The dynamics of our stochastic van der Pol [235] system are described by the stochastic differential equations:

$$\ddot{x}_i = \mu(1 - x_i^2)\dot{x}_i - \omega_i^2 x_i + \sum_{i \neq j} \gamma_{ij}(x_j - x_i) + \xi_i \ , \tag{6.29}$$

where the parameter μ controls the nonlinearity, the matrix γ controls the coupling between the different degrees of freedom (indexed by $i, j = 1, \ldots, N$, with $N = 3$ here), and the vector ω sets the oscillation frequency of each oscillator. We assumed constant and independent noise ξ_i for each oscillator by taking the diagonal covariance matrix $\mathcal{X} = 0.1\,\delta_{i,j}$ in all calculations and simulations.

$$\ddot{x} = \mu(1 - x^2)\dot{x} - \omega^2 x + \mathrm{diag}(\gamma dx),\ \gamma = \begin{pmatrix} 0 & 6 & 0 \\ 3 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix},\ dx_{ij} = (x_j - x_i) \tag{6.30}$$

with $w = [4, 3, 4.5], \mu = 1.4, y_0 = [4, 0.4, 0.8], x_0 = [6, 5.4, 6.8]$, $\delta t = 0.001$ for stochastics and $\delta t = 0.0001$ for ODE. The very small δt are needed for the simulation because the system is nonlinear and the stochastic euler algorithm is used, which needs very small stepsize. For the ode I use the Adams-Moulton predictor-corrector [236], which is stable for much larger stepsizes.

With the van der Pol oscillator we can calculate the expectation values in (6.24) exactly, since the Taylor expansion (6.25) of its function f terminates at the third order (refer to A.1.3 for exact expressions). Thus, in this case any error can be attributed to enforcing a Gaussian shape for a possibly non-Gaussian probability distribution.

Simulations were run in two regimes, $\mu = 0.05$ and $\mu = 1.5$. As expected, as long as the probability distribution does not deviate from gaussian the method works perfectly. As soon as we see some bimodality due to the nonlinearity the method deviates from the average calculated via stochastic simulation.

6.3.2.3 Genetic circuit model

As an extension of the oscillator model, we translated it from a linear regime to a more plausible gene regulatory model, that of the "repressilator" [27]. This regulatory model is based on Hill functions (refer to Eq. (2.43)):

$$\dot{c}_i = \frac{k^n}{c_j^n + k^n} - \lambda c_i + \zeta_i \ , \tag{6.31}$$

where $j = (i+1) \mod N$ (and $\mod$ represents the modulo operator), k is the expression level of the input gene at which the target is repressed by 50 %, and n is the binding cooperativity or Hill exponent.

Here we assume only intrinsic noise is present in the system, i.e. $\sigma_c^2 \propto \langle c \rangle$, as discussed in 2.7.1.1. For mathematical ease, let's perform a change of variables by introducing a new variable $\rho = \log c$ such that

$$\dot{\rho}_i = \frac{k^n e^{-\rho_i}}{e^{\rho_j n} + k^n} - \lambda + \xi_i \tag{6.32}$$

where now the noise ξ has constant variance (0.1 in our numerical example). We again assume a Gaussian white noise process for ξ, which means that the distribution of ζ_i obtains a log-normal shape.

6.4 Conclusion and outlook

In this project we developed a novel method to approximate the evolution of SDEs using the entropic matching method first developed in [233]. This method can successfully approximate the trajectories of systems when their probability distributions remain roughly Gaussian throughout the temporal dynamics under consideration. While the use of higher order derivatives in the series expansion can be useful in strongly nonlinear case, a test for Gaussianity should be employed concurrently with the simulation to guarantee that the approximation remains relevant.

In the future we expect that this method can be used in Bayesian inference schemes, such as described in 6.1. Its advantages are fast numerical forward simulations and the ability to compute derivatives of the temporal with respect to model parameters using algorithmic differentiation methods [201].

Appendix A

Mathematical derivations

A.1 Entropic matching

Below I reproduce some details omitted from the main text related to the minimization of the entropic matching functional.

A.1.1 Derivation of the entropic matching functional

Eq. (6.20) is explicitly

$$\begin{aligned} S(\bar{c}', C') &= -\left\langle \log \frac{\mathcal{G}(c_i' - \bar{c}', C')}{\mathcal{G}(c_i - \bar{c}(t), C(t))|_{c_i = c_i' - \delta t \dot{c}_i} \left| \frac{dc_i}{dc_i'} \right|} \right\rangle_{c'} \\ &= \left\langle \left[\frac{1}{2} \log \frac{|C'|}{|C|} + \log \left| 1 - \delta t \frac{df}{dc} \right|_{c=c'} + \frac{1}{2}(c' - \bar{c}')^T C'^{-1} (c' - \bar{c}') \right.\right. \\ &\quad \left.\left. - \frac{1}{2}(c - \bar{c})^T C^{-1} (c - \bar{c})|_{c = c' - \delta t \dot{c}} \right] \right\rangle_{c'} \end{aligned} \tag{A.1}$$

We can now integrate each term of expression A.1 individually:

$$\left\langle \frac{1}{2}(c' - \bar{c}')^T C'^{-1} (c' - \bar{c}') \right\rangle_{c'} = \frac{1}{2} \mathrm{tr}(C' C'^{-1}) = \frac{1}{2} \mathrm{tr}(\nVdash) \tag{A.2}$$

The terms with a logarithm yield

$$\left\langle \frac{1}{2} \log |C| \right\rangle_{c'} = \frac{1}{2} \mathrm{tr}(\log(C)) \tag{A.3}$$

The time evolution yields

$$\left\langle \log |1 - \delta t \frac{df}{dc}| \right\rangle_{c'} = \delta t \ \mathrm{tr} \left\langle \frac{df}{dc} \right\rangle_{c'} \tag{A.4}$$

Where the following simplification has been performed

$$\log|1-\delta t\frac{df}{dc}| = \mathrm{tr}\log(1-\delta t\frac{df}{dc}) \simeq -\mathrm{tr}\left(\delta t\frac{df}{dc}\right) \tag{A.5}$$

Finally, it remains to calculate

$$\begin{aligned}&\left\langle\frac{1}{2}(c-\bar{c})^T C^{-1}(c-\bar{c})|_{c=c'-\delta t\dot{c}}\right\rangle_{c'} = \\ &\left\langle\frac{1}{2}(c'-\delta t f(c')-\bar{c}+\bar{c}'-\bar{c}')^T C^{-1}(c'-\delta t f(c')-\bar{c}+\bar{c}'-\bar{c}')\right\rangle_{c'}\end{aligned} \tag{A.6}$$

Which consists of multiple terms. Each term may be calculated in turn:

$$\left\langle\frac{1}{2}(c'-\bar{c}')^T C^{-1}(c'-\bar{c}')\right\rangle_{c'} = \frac{1}{2}\mathrm{tr}(C'C^{-1}) \tag{A.7}$$

$$\left\langle\frac{1}{2}(\bar{c}'-\bar{c})^T C^{-1}(\bar{c}'-\bar{c})\right\rangle_{c'} = \frac{1}{2}(\bar{c}'-\bar{c})^T C^{-1}(\bar{c}'-\bar{c}) \tag{A.8}$$

$$\left\langle\frac{1}{2}(c'-\bar{c}')^T C^{-1}(\bar{c}'-\bar{c})\right\rangle_{c'} = 0 \tag{A.9}$$

$$\left\langle\frac{1}{2}(\bar{c}'-\bar{c})^T C^{-1}(\delta t f(c'))\right\rangle_{c'} = \delta t\frac{1}{2}(\bar{c}'-\bar{c})^T C^{-1}\left\langle f(c')\right\rangle_{c'} \tag{A.10}$$

$$\left\langle\frac{1}{2}(\delta t f(c'))^T C^{-1}(\delta t f(c'))\right\rangle_{c'} = \delta t^2\frac{1}{2}C^{-1}\langle f^2\rangle \tag{A.11}$$

The final term is less straightforward. First we establish the identity, which can be verified by explicit derivation:

$$-C'\frac{d}{dc'}\mathcal{G}(c'-\bar{c}',C') = \mathcal{G}(c'-\bar{c}',C')(c'-\bar{c}') \tag{A.12}$$

And now we use integration by parts in the second step.

$$\begin{aligned}&\left\langle\frac{1}{2}(\delta t f(c'))^T C^{-1}(c'-\bar{c}')\right\rangle_{c'} = \\ &-\frac{1}{2}\delta t\ \mathrm{tr}\left(C^{-1}C'\left\langle\delta t f(c')\right\rangle_{c'}^T\right) = \frac{1}{2}\delta t\ \mathrm{tr}\left(C^{-1}C'\left\langle\frac{df}{dc}\right\rangle_{c'}^T\right)\end{aligned} \tag{A.13}$$

$$\begin{aligned}&\left\langle\frac{1}{2}(c'-\bar{c}')^T C^{-1}(\delta t f(c'))\right\rangle_{c'} = -\frac{1}{2}\delta t\ \mathrm{tr}\left(C^{-1}C'\left\langle\delta t f(c')\right\rangle_{c'}\right) = \\ &\frac{1}{2}\delta t\ \mathrm{tr}\left(C'^T C^{-1}\left\langle\frac{df}{dc}\right\rangle_{c'}\right) = \frac{1}{2}\delta t\ \mathrm{tr}\left(C'C^{-1}\left\langle\frac{df}{dc}\right\rangle_{c'}\right)\end{aligned} \tag{A.14}$$

Plugging it all in we get

$$\begin{aligned} S =&\frac{1}{2}\mathrm{tr}\left[\nVdash + \log(2\pi C') - \log(2\pi C) - 2\delta t \,\left\langle \frac{df}{dc} \right\rangle_{c'} - C'C^{-1} \right. \\ &\left. +\delta t C^{-1}C' \left\langle \frac{df}{dc} \right\rangle_{c'}^{T} + \delta t C' C^{-1} \left\langle \frac{df}{dc} \right\rangle_{c'} + \delta t^2 C^{-1} \langle f^2 \rangle \right] \\ &+ \delta t(\vec{c}' - \vec{c})^T C^{-1} \left\langle f(c') \right\rangle_{c'} - \frac{1}{2}(\vec{c}' - \vec{c})^T C^{-1}(\vec{c}' - \vec{c}) \end{aligned} \tag{A.15}$$

A.1.2 Minimization of the entropic matching functional

To minimize Eq. (6.21), we seek $\vec{c}'$ such that $\frac{dS}{d\vec{c}'} = 0$:

$$\begin{aligned} &\delta t \;\mathrm{tr}\left[-2\frac{d}{d\vec{c}'}\left\langle \frac{df}{dc} \right\rangle_{c'} + C^{-1}C'\frac{d}{d\vec{c}'}\left\langle \frac{df}{dc} \right\rangle_{c'}^{T} + C'C^{-1}\frac{d}{d\vec{c}'}\left\langle \frac{df}{dc} \right\rangle_{c'}\right] \\ &+ \delta t(\vec{c}' - \vec{c})^T C^{-1}\frac{d}{d\vec{c}'}\left\langle f(c') \right\rangle_{c'} + \delta t C^{-1}\left\langle f(c') \right\rangle - C^{-1}(\vec{c}' - \vec{c}) = 0 \end{aligned} \tag{A.16}$$

$$\delta t \left\langle f(c') \right\rangle = (\nVdash - \delta t C \frac{d}{d\vec{c}'}\left\langle f(c') \right\rangle_{c'}^{T} C^{-1})(\vec{c}' - \vec{c}) \tag{A.17}$$

$$\frac{\vec{c}' - \vec{c}}{\delta t} = \left(\nVdash - \delta t C \frac{d}{d\vec{c}'}\left\langle f(c') \right\rangle_{c'}^{T} C^{-1}\right)^{-1}\left\langle f(c') \right\rangle \tag{A.18}$$

$$\dot{\vec{c}}' \simeq \left\langle f(c') \right\rangle \tag{A.19}$$

Where we consider the equation only up to second order, and the first term inside the trace cancels by considering $C^{-1}C' \simeq \nVdash$. Notice how if we take only one term in the series expansion we recover the linear case.

The same procedure is repeated for the second moment $\frac{dS}{dC'} = 0$:

$$\begin{aligned} &\mathrm{tr}\left[C'^{-1}D - DC^{-1} + \delta t D C^{-1}\left\langle \frac{df}{dc} \right\rangle_{c'} + \delta t C^{-1} D \left\langle \frac{df}{dc} \right\rangle_{c'}^{T} \right. \\ &\left. - 2\delta t \frac{d}{dC'}\left\langle \frac{df}{dc} \right\rangle_{c'} + \delta t C^{-1}C'\frac{d}{dC'}\left\langle \frac{df}{dc} \right\rangle_{c'}^{T} + \delta t C' C^{-1}\frac{d}{dC'}\left\langle \frac{df}{dc} \right\rangle_{c'}\right] \\ &= \delta t(\vec{c}' - \vec{c})^T C^{-1}\frac{d}{dC'}\left\langle f(c') \right\rangle_{c'} \end{aligned} \tag{A.20}$$

$$C'^{-1} - C^{-1} + \delta t C^{-1} \left\langle \frac{df}{dc} \right\rangle_{c'} + \delta t \left\langle \frac{df}{dc} \right\rangle_{c'}^{T} C^{-1} = 0 \tag{A.21}$$

$$\dot{C}'^{-1} = -(C^{-1} \left\langle \frac{df}{dc} \right\rangle_{c'} + \left\langle \frac{df}{dc} \right\rangle_{c'}^{T} C^{-1}) \tag{A.22}$$

$$\dot{C}' = C'(C^{-1} \left\langle \frac{df}{dc} \right\rangle_{c'} + \left\langle \frac{df}{dc} \right\rangle_{c'}^{T} C^{-1})C' \tag{A.23}$$

$$\dot{C}' \simeq \left\langle \frac{df}{dc} \right\rangle_{c'} C + C \left\langle \frac{df}{dc} \right\rangle_{c'}^{T} \tag{A.24}$$

$$\tag{A.25}$$

Where the last three terms in the trace vanish for the same reason as above and we consider $\bar{c}' - \bar{c} \simeq 0$. Again for only one term in the series expansion we recover the linear case.

A.1.3 Van der Pol oscillator expressions

Below are the general expressions for the dynamics of the Van der Pol oscillator and their higher order derivatives. The $\bullet$ symbol is a placeholder for either x or y.

$$\begin{aligned}
\dot{y}_i &= \mu(1 - x_i^2)y_i - \omega^2 x_i + \sum_{j \neq i} \gamma_{ij}(x_j - x_i) \\
\dot{x}_i &= y_i \\
\frac{d\dot{y}_i}{d\bullet} &= -(2\mu x_i y_i + \omega^2 + \sum_{j \neq i} \gamma_{ij})\delta_{\bullet x_i} + \gamma_{in}\delta_{\bullet x_n} + \mu(1 - x_i^2)\delta_{\bullet y_i} \\
\frac{d\dot{x}_i}{d\bullet} &= \delta_{\bullet y_i} \\
\frac{d^2\dot{y}_i}{d\bullet^2} &= -2\mu y_i \delta_{\bullet x_i}\delta_{\bullet x_i} - 2\mu x_i \delta_{\bullet x_i}\delta_{\bullet y_i} \\
\frac{d^2\dot{x}_i}{d\bullet^2} &= 0 \\
\frac{d^3\dot{y}_i}{d\bullet^3} &= -2\mu \delta_{\bullet y_i}\delta_{\bullet x_i}\delta_{\bullet x_i} \\
\frac{d^3\dot{x}_i}{d\bullet^3} &= 0
\end{aligned} \tag{A.26}$$

Appendix B

Code samples

B.1 Stochastic differential equation simulation

The fourth order stochastic Runge-Kutta algorithm is much more accurate than the standard Euler algorithm with a minimal amount of extra computation [89].

```python
def rk4(c, n, k, l, dt):
    '''
    Adapted from
    http://people.sc.fsu.edu/~jburkardt/c_src/stochastic_rk/stochastic_rk.html
    '''
    a21 =   2.71644396264860
    a31 = - 6.95653259006152
    a32 =   0.78313689457981
    a41 =   0.0
    a42 =   0.48257353309214
    a43 =   0.26171080165848
    a51 =   0.47012396888046
    a52 =   0.36597075368373
    a53 =   0.08906615686702
    a54 =   0.07483912056879

    q1 =   2.12709852335625
    q2 =   2.73245878238737
    q3 =  11.22760917474960
    q4 =  13.36199560336697

    x1 = c
    k1 = dt * evolve(x1, n, k, l) + T.sqrt(dt) * c * rv_n

    x2 = x1 + a21 * k1
    k2 = dt * evolve(x2, n, k, l) + T.sqrt(dt) * c * rv_n

    x3 = x1 + a31 * k1 + a32 * k2
    k3 = dt * evolve(x3, n, k, l) + T.sqrt(dt) * c * rv_n

    x4 = x1 + a41 * k1 + a42 * k2
    k4 = dt * evolve(x4, n, k, l) + T.sqrt(dt) * c * rv_n

    return T.cast(x1 + a51 * k1 + a52 * k2 + a53 * k3 + a54 * k4,
       ’float32’)
```

The code below was used to generate Fig. 2.3 by simulating trajectories from Eq.(2.40) for $f = \frac{x^n}{x^n+K^n}$ with $n = 6$, $K = 0.5$ and $g = x$ using the fourth order stochastic Runge-Kutta method as detailed above.

```
import theano
import theano.tensor as T
from theano.tensor.shared_randomstreams import RandomStreams
import numpy as np
import matplotlib.pyplot as plt
import time

#define the ode function
#dc/dt  = f(c, lambda)
#c is a vector with n components
def evolve(c, n, k, l):
    return T.pow(c, n)/(T.pow(c, n)+T.pow(k,n)) - l*c

if __name__ == ’__main__’:
    #random
    srng = RandomStreams(seed=31415)

    #define symbolic variables
    dt = T.fscalar(”dt”)
    k = T.fscalar(”k”)
    l = T.fscalar(”l”)
    n = T.fscalar(”n”)
    c = T.fvector(”c”)

    #define numeric variables
    num_samples = 50000
    c0 = theano.shared(0.5*np.ones(num_samples, dtype=’float32’))
    n0 = 6
    k0 = 0.5
    l0 = 1/(1+np.power(k0, n0))
    dt0 = 0.1
    total_time = 8
    total_steps = int(total_time/dt0)
    rv_n = srng.normal(c.shape, std=0.05) #is a shared variable

    #create loop
    #first symbolic loop with everything
    (cout, updates) = theano.scan(fn=rk4,outputs_info=[c], non_sequences=[n, k, l, dt], n_steps=total_steps)
    #compile it
    sim = theano.function(inputs=[n, k, l, dt], outputs=cout, givens={c:c0}, updates=updates, allow_input_downcast=True)
    cout = sim(n0, k0, l0, dt0)
```

B.2 Iterative segmentation procedure

The full code for the segmentation software is open-source and can be obtained at the following address: https://github.com/tmramalho/bigCellBrotherGUI

```
void ImageSegmentor::createMarkersIterative(cv::Mat& origImage, cv::Mat &landscape,
int maxHeight, int maxWidth) {
    std::vector< vector<cv::Point> > ctours;
    std::vector<cv::Vec4i> hrchy;
    double hDim, wDim;
    cv::Mat targets(origImage.size(), CV_8U, cv::Scalar::all(BLACK));
    for (int th = 20; th < 250; th += 10) { //increase threshold for distance transf.
        cv::Mat threshResult = ImageProcessor::threshold(landscape, th, false);
        cv::Mat newContour; threshResult.copyTo(newContour);
        //find connected components in the thresholded picture
        cv::findContours(newContour, ctours, hrchy, CV_RETR_CCOMP, CV_CHAIN_APPROX_NONE);
        int cc = ctours.size();
        for (int j = 0; j < cc; j++) {
            cv::RotatedRect boxTemp = cv::minAreaRect(ctours[j]);
            CellCont::detectHeightWidth(boxTemp, &hDim, &wDim);
            //draw the connected components which obey the criterion
            if (hDim < maxHeight && wDim < maxWidth) {
                cv::drawContours(targets, ctours, j, cv::Scalar::all(WHITE), -1, 8, hrchy, INT_MAX);
            }
        }
    }
    cv::Mat markers(origImage.size(), CV_32S, cv::Scalar::all(0));
    cv::findContours(targets, ctours, hrchy, CV_RETR_CCOMP,
       CV_CHAIN_APPROX_NONE);
    int nc = 0;
    for( unsigned int i = 0; i< ctours.size(); i++ ) {
        if(hrchy[i][3] != -1) continue; //it's a hole!
        nc++;
        cv::drawContours(markers, ctours, i, cv::Scalar::all(nc+1)
           , -1, 8, hrchy, INT_MAX);
    }
    markersPic = markers;
}
```

Bibliography

[1] Yuri Lazebnik. Can a biologist fix a radio?-or, what i learned while studying apoptosis. *Cancer Cell*, 2(3):179–182, 2002.

[2] Uri Alon. *An introduction to systems biology: design principles of biological circuits.* CRC press, 2006.

[3] Alex Mogilner, Roy Wollman, and Wallace F. Marshall. Quantitative modeling in cell biology: What is it good for? *Developmental Cell*, 11(3):279–287, September 2006.

[4] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell.* Garland Science, New York, 5 edition edition, November 2007.

[5] Hans V. Westerhoff and Bernhard O. Palsson. The evolution of molecular biology into systems biology. *Nat Biotech*, 22(10):1249–1252, October 2004.

[6] Hidde de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *J. Comput. Biol.*, 9(1):67–103, January 2002.

[7] Leland H. Hartwell, John J. Hopfield, Stanislas Leibler, and Andrew W. Murray. From molecular to modular cell biology. *Nature*, 402:C47–C52, 1999.

[8] Jonathon Howard, Stephan W. Grill, and Justin S. Bois. Turing's next steps: the mechanochemical basis of morphogenesis. *Nat Rev Mol Cell Biol*, 12(6):392–398, June 2011.

[9] Jeffrey J. Tabor, Howard M. Salis, Zachary Booth Simpson, Aaron A. Chevalier, Anselm Levskaya, Edward M. Marcotte, Christopher A. Voigt, and Andrew D. Ellington. A synthetic genetic edge detection program. *Cell*, 137(7):1272–1281, June 2009.

[10] Charles Darwin. *On the origin of species.* John Murray, United Kingdom, 1859.

[11] Ertugrul M. Ozbudak, Mukund Thattai, Han N. Lim, Boris I. Shraiman, and Alexander van Oudenaarden. Multistability in the lactose utilization network of escherichia coli. *Nature*, 427(6976):737–740, February 2004.

[12] Erwin Frey. Evolutionary game theory: Theoretical concepts and applications to microbial communities. *Phys. Stat. Mech. Its Appl.*, 389(20):4265–4298, October 2010.

[13] Tom De Wolf and Tom Holvoet. Emergence versus self-organisation: Different concepts but promising when combined. In Sven A. Brueckner, Giovanna Di Marzo Serugendo, Anthony Karageorgos, and Radhika Nagpal, editors, *Engineering Self-Organising Systems*, number 3464 in Lecture Notes in Computer Science, pages 1–15. Springer Berlin Heidelberg, 2005.

[14] Tracey A. Lincoln and Gerald F. Joyce. Self-sustained replication of an RNA enzyme. *Science*, 323(5918):1229–1232, February 2009.

[15] Albert-László Barabási and Zoltán N. Oltvai. Network biology: understanding the cell's functional organization. *Nat Rev Genet*, 5(2):101–113, February 2004.

[16] Steven H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, March 2001.

[17] William McGinnis and Robb Krumlauf. Homeobox genes and axial patterning. *Cell*, 68(2):283–302, January 1992.

[18] Timothy S. Gardner, Diego di Bernardo, David Lorenz, and James J. Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(5629):102–105, July 2003.

[19] Ramiz Daniel, Jacob R. Rubens, Rahul Sarpeshkar, and Timothy K. Lu. Synthetic analog computation in living cells. *Nature*, advance online publication, May 2013.

[20] Piro Siuti, John Yazbek, and Timothy K. Lu. Synthetic circuits integrating logic and memory in living cells. *Nat. Biotechnol.*, 2013.

[21] Troels T. Marstrand and John D. Storey. Identifying and mapping cell-type-specific chromatin programming of gene expression. *PNAS*, 111(6):E645–E654, February 2014.

[22] Saeed Tavazoie, Jason D. Hughes, Michael J. Campbell, Raymond J. Cho, and George M. Church. Systematic determination of genetic network architecture. *Nat Genet*, 22(3):281–285, July 1999.

[23] Jesper Tegnér, M. K. Stephen Yeung, Jeff Hasty, and James J. Collins. Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *PNAS*, 100(10):5944–5949, May 2003.

[24] Andrea Rau, Florence Jaffrézic, Jean-Louis Foulley, and R. W. Doerge. Reverse engineering gene regulatory networks using approximate bayesian computation. *Stat. Comput.*, December 2011.

[25] Mukesh Bansal, Vincenzo Belcastro, Alberto Ambesi-Impiombato, and Diego di Bernardo. How to infer gene networks from expression profiles. *Mol. Syst. Biol.*, 3(1), February 2007.

[26] Marcelo Behar, Derren Barken, Shannon L. Werner, and Alexander Hoffmann. The dynamics of signaling as a pharmacological target. *Cell*, 155(2):448–461, October 2013.

[27] Michael B. Elowitz and Stanislas Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, January 2000.

[28] Stephen Payne, Bochong Li, Yangxiaolu Cao, David Schaeffer, Marc D. Ryser, and Lingchong You. Temporal control of self-organized pattern formation without morphogen gradients in bacteria. *Mol Syst Biol*, 9(1), October 2013.

[29] Lulu Qian, Erik Winfree, and Jehoshua Bruck. Neural network computation with DNA strand displacement cascades. *Nature*, 475(7356):368–372, July 2011.

[30] Amir Mitchell, Gal H. Romano, Bella Groisman, Avihu Yona, Erez Dekel, Martin Kupiec, Orna Dahan, and Yitzhak Pilpel. Adaptive prediction of environmental changes by microorganisms. *Nature*, 460(7252):220–224, June 2009.

[31] Howard C. Berg, Douglas A. Brown, and others. Chemotaxis in escherichia coli analysed by three-dimensional tracking. *Nature*, 239(5374):500–504, 1972.

[32] H Berg and E Purcell. Physics of chemoreception. *Biophysical Journal*, 20(2):193–219, November 1977.

[33] W. Bialek and S. Setayeshgar. Physical limits to biochemical signaling. *Proc. Natl. Acad. Sci. U. S. A.*, 102(29):10040, 2005.

[34] Robert G. Endres and Ned S. Wingreen. Accuracy of direct gradient sensing by single cells. *Proc Natl Acad Sci U S A*, 105(41):15749–15754, October 2008.

[35] Ruoshi Yuan and Ping Ao. Beyond ito vs. stratonovich. *arXiv:1203.6600*, March 2012.

[36] N. Barkai and S. Leibler. Robustness in simple biochemical networks. *Nature*, 387(6636):913–917, June 1997.

[37] Victor Sourjik and Ned S Wingreen. Responding to chemical gradients: bacterial chemotaxis. *Curr. Opin. Cell Biol.*, 24(2):262–268, April 2012.

[38] Jerome T. Mettetal, Dale Muzzey, Carlos Gómez-Uribe, and Alexander van Oudenaarden. The frequency dependence of osmo-adaptation in saccharomyces cerevisiae. *Science*, 319(5862):482–484, January 2008.

[39] H. El-Samad, J.P. Goff, and M. Khammash. Calcium homeostasis and parturient hypocalcemia: An integral feedback perspective. *J. Theor. Biol.*, 214(1):17–29, January 2002.

[40] Johannes Reisert and Hugh R Matthews. Response properties of isolated mouse olfactory receptor cells. *J Physiol*, 530(Pt 1):113–122, January 2001.

[41] Anton Nikolaev, Kin-Mei Leung, Benjamin Odermatt, and Leon Lagnado. Synaptic mechanisms of adaptation and sensitization in the retina. *Nat Neurosci*, 16(7):934–941, July 2013.

[42] Wenzhe Ma, Ala Trusina, Hana El-Samad, Wendell A. Lim, and Chao Tang. Defining network topologies that can achieve biochemical adaptation. *Cell*, 138(4):760–773, August 2009.

[43] Gerardo Aquino, Luke Tweedy, Doris Heinrich, and Robert G. Endres. Memory improves precision of cell sensing in fluctuating environments. *Sci. Rep.*, 4, July 2014.

[44] Sameer S. Bajikar, Christiane Fuchs, Andreas Roller, Fabian J. Theis, and Kevin A. Janes. Parameterizing cell-to-cell regulatory heterogeneities via stochastic transcriptional profiles. *PNAS*, 111(5):E626–E635, February 2014.

[45] David Dubnau and Richard Losick. Bistability in bacteria. *Mol. Microbiol.*, 61(3):564–572, August 2006.

[46] Edo Kussell and Stanislas Leibler. Phenotypic diversity, population growth, and information in fluctuating environments. *Science*, 309(5743):2075–2078, September 2005.

[47] Nathalie Q. Balaban, Jack Merrin, Remy Chait, Lukasz Kowalik, and Stanislas Leibler. Bacterial persistence as a phenotypic switch. *Science*, 305(5690):1622–1625, September 2004.

[48] Christopher M. Waters and Bonnie L. Bassler. Quorum sensing: Cell-to-cell communication in bacteria. *Annu. Rev. Cell Dev. Biol.*, 21(1):319–346, 2005.

[49] Maximilian Weitz, Andrea Mückl, Korbinian Kapsner, Ronja Berg, Andrea Meyer, and Friedrich C. Simmel. Communication and computation by bacteria compartmentalized within microemulsion droplets. *J. Am. Chem. Soc.*, 136(1):72–75, 2013.

[50] Scott F. Gilbert. *Developmental Biology*. Sinauer Associates, Sunderland, 6th edition, 2000.

[51] Conrad H. Waddington. Canalization of development and the inheritance of acquired characters. *Nature*, 150(3811):563–565, 1942.

[52] Sui Huang, Gabriel Eichler, Yaneer Bar-Yam, and Donald E. Ingber. Cell fates as high-dimensional attractor states of a complex gene regulatory network. *Phys. Rev. Lett.*, 94(12):128701, April 2005.

[53] Sui Huang and Donald E. Ingber. Shape-dependent control of cell growth, differentiation, and apoptosis: Switching between attractors in cell regulatory networks. *Experimental Cell Research*, 261(1):91–103, November 2000.

[54] James E. Ferrell. Bistability, bifurcations, and waddington's epigenetic landscape. *Curr. Biol.*, 22(11):R458–R466, 2012.

[55] Gürol M. Süel, Jordi Garcia-Ojalvo, Louisa M. Liberman, and Michael B. Elowitz. An excitable gene regulatory circuit induces transient cellular differentiation. *Nature*, 440(7083):545–550, March 2006.

[56] David Kimelman and Benjamin L. Martin. Anterior–posterior patterning in early development: three strategies. *Wiley Interdiscip. Rev. Dev. Biol.*, 1(2):253–266, 2012.

[57] Feng Liu, Alexander H. Morrison, and Thomas Gregor. Dynamic interpretation of maternal inputs by the drosophila segmentation gene network. *PNAS*, page 201220912, April 2013.

[58] P. W. Ingham. The molecular genetics of embryonic pattern formation in drosophila. *Nature*, 335(6185):25–34, 1988.

[59] Anna Kicheva, Periklis Pantazis, Tobias Bollenbach, Yannis Kalaidzidis, Thomas Bittig, Frank Jülicher, and Marcos González-Gaitán. Kinetics of morphogen gradient formation. *Science*, 315(5811):521–525, January 2007.

[60] Thomas Gregor, William Bialek, Rob R. de Ruyter van Steveninck, David W. Tank, and Eric F. Wieschaus. Diffusion and scaling during early embryonic pattern formation. *Proc Natl Acad Sci U S A*, 102(51):18403–18407, December 2005.

[61] Aude Porcher and Nathalie Dostatni. The bicoid morphogen system. *Curr. Biol*, 20(5):R249–254, March 2010.

[62] L. Wolpert. Positional information and the spatial pattern of cellular differentiation+*. *J. Theor. Biol.*, 25(1):1–47, 1969.

[63] Amanda Ochoa-Espinosa, Gozde Yucel, Leah Kaplan, Adam Pare, Noel Pura, Adam Oberstein, Dmitri Papatsenko, and Stephen Small. The role of binding site cluster strength in bicoid-dependent patterning in drosophila. *PNAS*, 102(14):4960–4965, April 2005.

[64] Bahram Houchmandzadeh, Eric Wieschaus, and Stanislas Leibler. Establishment of developmental precision and proportions in the early drosophila embryo. *Nature*, 415(6873):798–802, February 2002.

[65] Johannes Jaeger. The gap gene network. *Cell. Mol. Life Sci.*, 68(2):243–274, October 2010.

[66] Johannes Jaeger, Manu, and John Reinitz. Drosophila blastoderm patterning. *Curr. Opin. Genet. Dev.*, 22(6):533–541, December 2012.

[67] Manu, Svetlana Surkova, Alexander V. Spirov, Vitaly V. Gursky, Hilde Janssens, Ah-Ram Kim, Ovidiu Radulescu, Carlos E. Vanario-Alonso, David H. Sharp, Maria Samsonova, and John Reinitz. Canalization of gene expression and domain shifts in the drosophila blastoderm by dynamical attractors. *PLoS Comput Biol*, 5(3):e1000303, March 2009.

[68] Andrei Pisarev, Ekaterina Poustelnikova, Maria Samsonova, and John Reinitz. FlyEx, the quantitative atlas on segmentation gene expression at cellular resolution. *Nucl. Acids Res.*, 37(suppl 1):D560–D566, January 2009.

[69] Julien O. Dubuis, Gašper Tkačik, Eric F. Wieschaus, Thomas Gregor, and William Bialek. Positional information, in bits. *PNAS*, 110(41):16301–16308, October 2013.

[70] D. Stanojevic, S. Small, and M. Levine. Regulation of a segmentation stripe by overlapping activators and repressors in the drosophila embryo. *Science*, 254(5036):1385, 1991.

[71] M. D. Schroeder, C. Greer, and U. Gaul. How to make stripes: deciphering the transition from non-periodic to periodic patterns in drosophila segmentation. *Development*, 138(14):3067–3078, June 2011.

[72] Johannes Jaeger, David H. Sharp, and John Reinitz. Known maternal gradients are not sufficient for the establishment of gap domains in drosophila melanogaster. *Mechanisms of Development*, 124(2):108–128, February 2007.

[73] E. B. Lewis. A gene complex controlling segmentation in drosophila. *Nature*, 276(5688):565–570, December 1978.

[74] C. E. Shannon. Communication in the presence of noise. *Proc. IRE*, 37(1):10–21, 1949.

[75] E. T. Jaynes. *Probability Theory: The Logic of Science.* Cambridge University Press, Cambridge, UK ; New York, NY, June 2003.

[76] E. T. Jaynes. Information theory and statistical mechanics. II. *Phys. Rev.*, 108(2):171–190, October 1957.

[77] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106(4):620–630, May 1957.

[78] Ariel Caticha. Lectures on probability, entropy, and statistical physics. *0808.0012*, July 2008.

[79] G. Tkačik and A. M Walczak. Information transmission in genetic regulatory networks: a review. *J. Phys. Condens. Matter*, 23:153102, 2011.

[80] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Phys. Rev. E*, 69(6):066138, June 2004.

[81] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley-Interscience, Hoboken, N.J, 2 edition edition, July 2006.

[82] C. Gourieroux and A. Monfort. *Statistics and Econometric Models: Volume 1, General Concepts, Estimation, Prediction and Algorithms.* Cambridge University Press, 1995.

[83] E. L. Lehmann and G. Casella. *Theory of point estimation.* Springer Verlag, 1998.

[84] Torsten Enßlin and Cornelius Weig. Inference with minimal gibbs free energy in information field theory. *Phys. Rev. E*, 82, November 2010.

[85] M. E. J. Newman and G. T. Barkema. *Monte Carlo Methods in Statistical Physics.* Oxford University Press, Oxford : New York, April 1999.

[86] Gašper Tkačik, Olivier Marre, Dario Amodei, Elad Schneidman, William Bialek, and II Berry, Michael J. Searching for collective behavior in a large network of sensory neurons. *PLoS Comput Biol*, 10(1):e1003408, January 2014.

[87] N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry, Third Edition.* North Holland, Amsterdam ; Boston, 3 edition edition, May 2007.

[88] Crispin Gardiner. *Stochastic Methods: A Handbook for the Natural and Social Sciences.* Springer, softcover reprint of hardcover 4th ed. 2009 edition, November 2010.

[89] Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations.* Springer, corrected edition, August 1992.

[90] Daniel T. Gillespie, Andreas Hellander, and Linda R. Petzold. Perspective: Stochastic algorithms for chemical kinetics. *J. Chem. Phys.*, 138(17):170901, 2013.

[91] David Salomon. *A Concise Introduction to Data Compression.* Springer, London, auflage: 2008 edition, January 2008.

[92] Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications.* Springer, New York, auflage: 3rd ed. 2008 edition, 2009.

[93] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proc. Lond. Math. Soc.*, s2-42(1):230–265, January 1937.

[94] David J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge, UK ; New York, September 2003.

[95] Nir Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799 –805, February 2004.

[96] John E. Stone, David J. Hardy, Ivan S. Ufimtsev, and Klaus Schulten. GPU-accelerated molecular modeling coming of age. *J. Mol. Graph. Model.*, 29(2):116–125, 2010.

[97] Roland Schulz, Benjamin Lindner, Loukas Petridis, and Jeremy C. Smith. Scaling of multimillion-atom biological molecular dynamics simulation on a petascale supercomputer. *J. Chem. Theory Comput.*, 5(10):2798–2808, 2009.

[98] Guy Karlebach and Ron Shamir. Modelling and analysis of gene regulatory networks. *Nat. Rev. Mol. Cell Biol.*, 9(10):770–780, October 2008.

[99] Darren J. Wilkinson. Stochastic modelling for quantitative description of heterogeneous biological systems. *Nat. Rev. Genet.*, 10(2):122–133, February 2009.

[100] Jonathan M. Raser and Erin K. O'Shea. Noise in gene expression: Origins, consequences, and control. *Science*, 309(5743):2010–2013, September 2005.

[101] M. Thattai and A. Van Oudenaarden. Stochastic gene expression in fluctuating environments. *Genetics*, 167(1):523–530, 2004.

[102] Steuer Ralf. Effects of stochasticity in models of the cell cycle: from quantized cycle times to noise-induced oscillations. *J. Theor. Biol.*, 228(3):293–301, June 2004.

[103] Johan Paulsson and Måns Ehrenberg. Random signal fluctuations can reduce random fluctuations in regulated components of chemical regulatory networks. *Phys. Rev. Lett.*, 84(23):5447, June 2000.

[104] M. Thattai. Intrinsic noise in gene regulatory networks. *Proc. Natl. Acad. Sci.*, 98(15):8614–8619, July 2001.

[105] Peter S. Swain, Michael B. Elowitz, and Eric D. Siggia. Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proc Natl Acad Sci U S A*, 99(20):12795–12800, October 2002.

[106] Johan Paulsson. Summing up the noise in gene networks. *Nature*, 427(6973):415–418, January 2004.

[107] Michael B. Elowitz, Arnold J. Levine, Eric D. Siggia, and Peter S. Swain. Stochastic gene expression in a single cell. *Science*, 297(5584):1183 –1186, 2002.

[108] Andreas Hilfinger and Johan Paulsson. Separating intrinsic from extrinsic fluctuations in dynamic biological systems. *PNAS*, 108(29):12167–12172, July 2011.

[109] Y. Taniguchi, P. J. Choi, G.-W. Li, H. Chen, M. Babu, J. Hearn, A. Emili, and X. S. Xie. Quantifying e. coli proteome and transcriptome with single-molecule sensitivity in single cells. *Science*, 329(5991):533–538, July 2010.

[110] E. M Ozbudak, M. Thattai, I. Kurtser, A. D Grossman, and A. van Oudenaarden. Regulation of noise in the expression of a single gene. *Nat. Genet.*, 31(1):69–73, 2002.

[111] Vahid Shahrezaei and Peter S. Swain. Analytical distributions for stochastic gene expression. *Proc. Natl. Acad. Sci.*, 105(45):17256 –17261, November 2008.

[112] A.L. Koch. The logarithm in biology: II. distributions simulating the log-normal. *Journal of Theoretical Biology*, 23(2):251–268, May 1969.

[113] Arthur L. Koch. The logarithm in biology 1. mechanisms generating the log-normal distribution exactly. *Journal of Theoretical Biology*, 12(2):276–290, November 1966.

[114] David C. Hoyle, Magnus Rattray, Ray Jupp, and Andrew Brass. Making sense of microarray data distributions. *Bioinformatics*, 18(4):576–584, April 2002.

[115] Martin Bengtsson, Anders Ståhlberg, Patrik Rorsman, and Mikael Kubista. Gene expression profiling in single cells from the pancreatic islets of langerhans reveals lognormal distribution of mRNA levels. *Genome Res.*, 15(10):1388–1392, October 2005.

[116] N. E Buchler, U. Gerland, and T. Hwa. On schemes of combinatorial transcription logic. *Proc. Natl. Acad. Sci. U. S. A.*, 100(9):5136, 2003.

[117] Lacramioara Bintu, Nicolas E Buchler, Hernan G Garcia, Ulrich Gerland, Terence Hwa, Jané Kondev, and Rob Phillips. Transcriptional regulation by the numbers: models. *Current Opinion in Genetics & Development*, 15(2):116–124, April 2005.

[118] Lacramioara Bintu, Nicolas E Buchler, Hernan G Garcia, Ulrich Gerland, Terence Hwa, Jané Kondev, Thomas Kuhlman, and Rob Phillips. Transcriptional regulation by the numbers: applications. *Current Opinion in Genetics & Development*, 15(2):125–135, April 2005.

[119] Jean-Pierre Changeux. Allostery and the monod-wyman-changeux model after 50 years. *Annu. Rev. Biophys.*, 41(1):103–133, 2012.

[120] Harold D. Kim, Tal Shay, Erin K. O'Shea, and Aviv Regev. Transcriptional regulatory circuits: Predicting numbers from alphabets. *Science*, 325(5939):429–432, July 2009.

[121] Michael London and Michael Häusser. Dendritic computation. *Annu. Rev. Neurosci.*, 28(1):503–532, 2005.

[122] George Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.*, 2(4):303–314, 1989.

[123] Ortrud Wartlick, Anna Kicheva, and Marcos González-Gaitán. Morphogen gradient formation. *Cold Spring Harb Perspect Biol*, 1(3):a001255, September 2009.

[124] J. B. Gurdon and P.-Y. Bourillot. Morphogen gradient interpretation. *Nature*, 413(6858):797–803, October 2001.

[125] Wolfgang Driever and Christiane Nüsslein-Volhard. The bicoid protein determines position in the drosophila embryo in a concentration-dependent manner. *Cell*, 54(1):95–104, July 1988.

[126] Thomas Gregor, David W. Tank, Eric F. Wieschaus, and William Bialek. Probing the limits to positional information. *Cell*, 130(1):153–164, July 2007.

[127] O. Grimm, M. Coppey, and E. Wieschaus. Modelling the bicoid gradient. *Development*, 137(14):2253–2264, June 2010.

[128] Johannes Jaeger and Alfonso Martinez-Arias. Getting the measure of positional information. *PLoS Biol*, 7(3):e1000081, March 2009.

[129] B. Houchmandzadeh, E. Wieschaus, and S. Leibler. Precise domain specification in the developing drosophila embryo. *Phys. Rev. E*, 72(6), December 2005.

[130] Thomas Gregor, Alistair P. McGregor, and Eric F. Wieschaus. Shape and function of the bicoid morphogen gradient in dipteran species with different sized embryos. *Dev. Biol.*, 316(2):350–358, April 2008.

[131] O. Grimm and E. Wieschaus. The bicoid gradient is shaped independently of nuclei. *Development*, 137(17):2857–2862, August 2010.

[132] Shawn C. Little, Gašper Tkačik, Thomas B. Kneeland, Eric F. Wieschaus, and Thomas Gregor. The formation of the bicoid morphogen gradient requires protein movement from anteriorly localized mRNA. *PLoS Biol*, 9(3):e1000596, March 2011.

[133] F. Tostevin, P. R Ten Wolde, and M. Howard. Fundamental limits to position determination by concentration gradients. *PLoS Comput. Biol.*, 3(4):e78, 2007.

[134] Timothy Saunders and Martin Howard. When it pays to rush: interpreting morphogen gradients prior to steady-state. *Phys. Biol.*, 6(4):046020, December 2009.

[135] Sven Bergmann, Oded Sandler, Hila Sberro, Sara Shnider, Eyal Schejter, Ben-Zion Shilo, and Naama Barkai. Pre-steady-state decoding of the bicoid morphogen gradient. *PLoS Biol*, 5(2):e46, February 2007.

[136] Timothy Saunders and Martin Howard. Morphogen profiles can be optimized to buffer against noise. *Phys. Rev. E*, 80(4), October 2009.

[137] Feng He, Timothy E. Saunders, Ying Wen, David Cheung, Renjie Jiao, Pieter Rein ten Wolde, Martin Howard, and Jun Ma. Shaping a morphogen gradient for positional precision. *Biophysical Journal*, 99(3):697–707, August 2010.

[138] Thorsten Erdmann, Martin Howard, and Pieter Rein ten Wolde. Role of spatial averaging in the precision of gene expression patterns. *Phys. Rev. Lett.*, 103(25), December 2009.

[139] Eldon Emberly. Optimizing the readout of morphogen gradients. *Phys. Rev. E*, 77(4), April 2008.

[140] Gašper Tkačik, Curtis G. Callan, and William Bialek. Information capacity of genetic regulatory elements. *Phys Rev E Stat Nonlin Soft Matter Phys*, 78(1 Pt 1):011910–011910, July 2008.

[141] Gašper Tkačik, Aleksandra M. Walczak, and William Bialek. Optimizing information flow in small genetic networks. *Phys. Rev. E*, 80(3):031920, 2009.

[142] Gašper Tkačik. From statistical mechanics to information theory: understanding biophysical information-processing systems. *1006.4291*, June 2010.

[143] Gašper Tkačik, Julien O. Dubuis, Mariela D. Petkova, and Thomas Gregor. Positional information, positional error, and readout precision in morphogenesis: A mathematical framework. *Genetics*, 199(1):39–59, January 2015.

[144] E. Ziv, I. Nemenman, and C. H Wiggins. Optimal signal processing in small stochastic biochemical networks. *PLoS One*, 2(10):e1077, 2007.

[145] Yoshihiro Morishita and Yoh Iwasa. Accuracy of positional information provided by multiple morphogen gradients with correlated noise. *Phys. Rev. E*, 79(6):061905, June 2009.

[146] Yoshihiro Morishita and Yoh Iwasa. Coding design of positional information for robust morphogenesis. *Biophys. J.*, 101(10):2324–2335, November 2011.

[147] David M. Holloway, Lionel G. Harrison, David Kosman, Carlos E. Vanario-Alonso, and Alexander V. Spirov. Analysis of pattern precision shows that drosophila segmentation develops substantial independence from gradients of maternal gene products. *Dev Dyn*, 235(11):2949–2960, November 2006.

[148] David Cheung, Cecelia Miles, Martin Kreitman, and Jun Ma. Scaling of the bicoid morphogen gradient by a volume-dependent production rate. *Development*, 138(13):2741–2749, July 2011.

[149] Feng He, Ying Wen, Jingyuan Deng, Xiaodong Lin, Long Jason Lu, Renjie Jiao, and Jun Ma. Probing intrinsic properties of a robust morphogen gradient in drosophila. *Dev Cell*, 15(4):558–567, October 2008.

[150] David Cheung, Cecelia Miles, Martin Kreitman, and Jun Ma. Adaptation of the length scale and amplitude of the bicoid gradient profile to achieve robust patterning in abnormally large drosophila melanogaster embryos. *Development*, 141(1):124–135, January 2014.

[151] Inbal Hecht, Wouter-Jan Rappel, and Herbert Levine. Determining the scale of the bicoid morphogen gradient. *Proc Natl Acad Sci U S A*, 106(6):1710–1715, February 2009.

[152] Aitana Morton de Lachapelle and Sven Bergmann. Pre-steady and stable morphogen gradients: can they coexist? *Mol Syst Biol*, 6:428, November 2010.

[153] CM Miles, SE Lott, CL Luengo Hendriks, MZ Ludwig, Manu, CL Williams, and M Kreitman. Artificial selection on egg size perturbs early pattern formation in drosophila melanogaster. *Evolution*, 65(1):33–42, January 2011.

[154] Tinri Aegerter-Wilmsen, Christof M. Aegerter, and Ton Bisseling. Model for the robust establishment of precise proportions in the early drosophila embryo. *Journal of Theoretical Biology*, 234(1):13–19, May 2005.

[155] Hongtao Chen, Zhe Xu, Constance Mei, Danyang Yu, and Stephen Small. A system of repressor gradients spatially organizes the boundaries of bicoid-dependent target genes. *Cell*, 149(3):618–629, April 2012.

[156] Manu, Svetlana Surkova, Alexander V Spirov, Vitaly V Gursky, Hilde Janssens, Ah-Ram Kim, Ovidiu Radulescu, Carlos E Vanario-Alonso, David H Sharp, Maria Samsonova, and John Reinitz. Canalization of gene expression in the drosophila blastoderm by gap gene cross regulation. *PLoS Biol*, 7(3):e1000049, March 2009.

[157] Aharon Helman, Bomyi Lim, María José Andreu, Yoosik Kim, Tatyana Shestkin, Hang Lu, Gerardo Jiménez, Stanislav Y. Shvartsman, and Ze'ev Paroush. RTK signaling modulates the dorsal gradient. *Development*, 139(16):3032–3039, August 2012.

[158] I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. Dover Publications, Mineola, N.Y, October 2000.

[159] Herbert Goldstein, Charles P. Poole, and John L. Safko. *Classical Mechanics*. Addison-wesley, 3rd edition, 2001.

[160] J. R. Martin, A. Raibaud, and R. Ollo. Terminal pattern elements in drosophila embryo induced by the torso-like protein. *Nature*, 367(6465):741–745, February 1994.

[161] Travis K. Johnson, Tova Crossman, Karyn A. Foote, Michelle A. Henstridge, Melissa J. Saligari, Lauren Forbes Beadle, Anabel Herr, James C. Whisstock, and Coral G. Warr. Torso-like functions independently of torso to regulate drosophila

growth and developmental timing. *Proc Natl Acad Sci U S A*, 110(36):14688–14692, September 2013.

[162] Hermann Haken. *Synergetics: An Introduction.* Springer, 3rd ed. 1983. softcover reprint of the original 3rd ed. 1983 edition edition, March 2012.

[163] Sergi Regot, Javier Macia, Núria Conde, Kentaro Furukawa, Jimmy Kjellén, Tom Peeters, Stefan Hohmann, Eulàlia de Nadal, Francesc Posas, and Ricard Solé. Distributed biological computation with multicellular engineered networks. *Nature*, 469(7329):207–211, January 2011.

[164] Javier Macía, Francesc Posas, and Ricard V. Sole. Distributed computation: the new wave of synthetic biology devices. *Trends Biotechnol.*, 30(6):342–349, 2012.

[165] Subhayu Basu, Yoram Gerchman, Cynthia H. Collins, Frances H. Arnold, and Ron Weiss. A synthetic multicellular system for programmed pattern formation. *Nature*, 434(7037):1130–1134, 2005.

[166] Harold Abelson, Ron Weiss, Don Allen, Daniel Coore, Chris Hanson, George Homsy, Thomas F. Knight, Radhika Nagpal, Erik Rauch, and Gerald Jay Sussman. Amorphous computing. *Commun. ACM*, 43(5):74–82, May 2000.

[167] Ozalp Babaoglu, Geoffrey Canright, Andreas Deutsch, Gianni A. Di Caro, Frederick Ducatelle, Luca M. Gambardella, Niloy Ganguly, Márk Jelasity, Roberto Montemanni, Alberto Montresor, and others. Design patterns from biology for distributed computing. *ACM Trans. Auton. Adapt. Syst. TAAS*, 1(1):26–66, 2006.

[168] Radhika Nagpal. Programmable self-assembly using biologically-inspired multiagent control. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 418–425, 2002.

[169] Justin Werfel, Kirstin Petersen, and Radhika Nagpal. Designing collective behavior in a termite-inspired robot construction team. *Science*, 343(6172):754–758, February 2014.

[170] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 25–34, New York, NY, USA, 1987. ACM.

[171] Saket Navlakha and Ziv Bar-Joseph. Distributed information processing in biological and computational systems. *Commun. ACM*, 58(1):94–102, December 2014.

[172] A. M. Turing. The chemical basis of morphogenesis. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.*, 237(641):37–72, 1952.

[173] Michael Cross and Henry Greenside. *Pattern Formation and Dynamics in Nonequilibrium Systems*. Cambridge University Press, Cambridge, UK ; New York, 1 edition edition, August 2009.

[174] J. Raspopovic, L. Marcon, L. Russo, and J. Sharpe. Digit patterning is controlled by a bmp-sox9-wnt turing network modulated by morphogen gradients. *Science*, 345(6196):566–570, August 2014.

[175] Rushikesh Sheth, Luciano Marcon, M. Félix Bastida, Marisa Junco, Laura Quintana, Randall Dahn, Marie Kmita, James Sharpe, and Maria A. Ros. Hox genes regulate digit patterning by controlling the wavelength of a turing-type mechanism. *Science*, 338(6113):1476–1480, December 2012.

[176] P. W. Anderson. More is different. *Science*, 177(4047):393–396, August 1972.

[177] Michael S. Branicky. Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoretical Computer Science*, 138(1):67–100, February 1995.

[178] S. Bandini, G. Mauri, G. Pavesi, and C. Simone. Computing with a distributed reaction-diffusion model. In Maurice Margenstern, editor, *Machines, Computations, and Universality*, number 3354 in Lecture Notes in Computer Science, pages 93–103. Springer Berlin Heidelberg, January 2005.

[179] P.-M. Binder. Computation: The edge of reductionism. *Nature*, 459(7245):332–334, May 2009.

[180] Stephen Wolfram. Statistical mechanics of cellular automata. *Rev. Mod. Phys.*, 55(3):601–644, July 1983.

[181] Andreas Deutsch and Sabine Dormann. *Cellular Automaton Modeling of Biological Pattern Formation*. Modeling and Simulation in Science, Engineering and Technology. Birkhäuser Boston, Boston, MA, 1 edition, 2005.

[182] Svetlana Surkova, Elena Golubkova, Manu, Lena Panok, Lyudmila Mamon, John Reinitz, and Maria Samsonova. Quantitative dynamics and increased variability of segmentation gene expression in the drosophila krüppel and knirps mutants. *Dev. Biol.*, 376(1):99–112, April 2013.

[183] Hiroki Hamada, Masakatsu Watanabe, Hiu Eunice Lau, Tomoki Nishida, Toshiaki Hasegawa, David M. Parichy, and Shigeru Kondo. Involvement of delta/notch signaling in zebrafish adult pigment stripe patterning. *Development*, 141(2):318–324, January 2014.

[184] Yoshihiro Morishita and Yoh Iwasa. Growth based morphogenesis of vertebrate limb bud. *Bull. Math. Biol.*, 70(7):1957–1978, July 2008.

[185] Moritz Mercker, Dirk Hartmann, and Anna Marciniak-Czochra. A mechanochemical model for embryonic pattern formation: Coupling tissue mechanics and morphogen expression. *PLoS ONE*, 8(12):e82617, December 2013.

[186] Carl-Philipp Heisenberg and Yohanns Bellaïche. Forces in tissue morphogenesis and patterning. *Cell*, 153(5):948–962, May 2013.

[187] Celeste M. Nelson, Ronald P. Jean, John L. Tan, Wendy F. Liu, Nathan J. Sniadecki, Alexander A. Spector, and Christopher S. Chen. Emergent patterns of growth controlled by multicellular form and mechanics. *PNAS*, 102(33):11594–11599, August 2005.

[188] Séverine Urdy. On the evolution of morphogenetic models: mechano-chemical interactions and an integrated view of cell differentiation, growth, pattern formation and morphogenesis. *Biol. Rev.*, 87(4):786–803, 2012.

[189] Theodore J Perkins, Johannes Jaeger, John Reinitz, and Leon Glass. Reverse engineering the gap gene network of drosophila melanogaster. *PLoS Comput Biol*, 2(5):e51, May 2006.

[190] Yves Fomekong-Nanfack, Marten Postma, and Jaap Kaandorp. Inferring drosophila gap gene regulatory network: a parameter sensitivity and perturbation analysis. *BMC Syst. Biol.*, 3(1):94, 2009.

[191] Kolja Becker, Eva Balsa-Canto, Damjan Cicin-Sain, Astrid Hoermann, Hilde Janssens, Julio R. Banga, and Johannes Jaeger. Reverse-engineering post-transcriptional regulation of gap genes in drosophila melanogaster. *PLoS Comput. Biol.*, 9(10):e1003281, October 2013.

[192] Anton Crombach, Karl R. Wotton, Damjan Cicin-Sain, Maksat Ashyraliyev, and Johannes Jaeger. Efficient reverse-engineering of a developmental gene regulatory network. *PLoS Comput Biol*, 8(7):e1002589, July 2012.

[193] Hilde Janssens, Anton Crombach, Karl Richard Wotton, Damjan Cicin-Sain, Svetlana Surkova, Chea Lu Lim, Maria Samsonova, Michael Akam, and Johannes Jaeger. Lack of tailless leads to an increase in expression variability in drosophila embryos. *Dev. Biol.*, 377(1):305–317, May 2013.

[194] Maksat Ashyraliyev, Ken Siggens, Hilde Janssens, Joke Blom, Michael Akam, and Johannes Jaeger. Gene circuit analysis of the terminal gap gene huckebein. *PLoS Comput Biol*, 5(10):e1000548, October 2009.

[195] Thomas Duriez, Vladimir Parezanovic, Bernd R. Noack, Laurent Cordier, Marc Segond, and Markus Abel. Attractor control using machine learning. *ArXiv13115250 Nlin Physicsphysics*, November 2013.

[196] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-Laszlo Barabasi. Controllability of complex networks. *Nature*, 473(7346):167–173, May 2011.

[197] Sean P. Cornelius, William L. Kath, and Adilson E. Motter. Realistic control of network dynamics. *Nat Commun*, 4, June 2013.

[198] Arthur W. Burks and John Von Neumann. *Theory of self-reproducing automata.* University of Illinois Press, 1966.

[199] Navot Israeli and Nigel Goldenfeld. Coarse-graining of cellular automata, emergence, and the predictability of complex systems. *Phys. Rev. E*, 73(2):026203, February 2006.

[200] C. J. Twining and P.-M. Binder. Enumeration of limit cycles in noncylindrical cellular automata. *J Stat Phys*, 66(1-2):385–401, January 1992.

[201] A. Griewank, J. Utke, and A. Walther. Evaluating higher derivative tensors by forward propagation of univariate taylor series. *Math. Comput.*, 69(231):1117–1130, 2000.

[202] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011.

[203] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, Atlanta, Georgia, USA, 2013.

[204] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1139–1147, 2013.

[205] Jonathan Bieler, Christian Pozzorini, and Felix Naef. Whole-embryo modeling of early segmentation in drosophila identifies robust and fragile expression domains. *Biophys J*, 101(2):287–296, July 2011.

[206] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[207] Aitana Morton de Lachapelle and Sven Bergmann. Precision and scaling in morphogen gradient read-out. *Mol Syst Biol*, 6:351, March 2010.

[208] Gábor Balázsi, Alexander van Oudenaarden, and James J. Collins. Cellular decision making and biological noise: From microbes to mammals. *Cell*, 144(6):910–925, March 2011.

[209] Eric D. Siggia and Massimo Vergassola. Decisions on the fly in cellular sensory systems. *Proc. Natl. Acad. Sci.*, 110(39):E3704–E3712, 2013.

[210] Jan Drugowitsch, Gregory C. DeAngelis, Eliana M. Klier, Dora E. Angelaki, and Alexandre Pouget. Optimal multisensory decision-making in a reaction-time task. *eLife Sciences*, 3:e03005, June 2014.

[211] Theodore J. Perkins and Peter S. Swain. Strategies for cellular decision-making. *Mol Syst Biol*, 5(1), November 2009.

[212] Markus Kollmann, Linda Løvdok, Kilian Kilian Bartholomé, Jens Timmer, and Victor Sourjik. Design principles of a bacterial signalling network. *Nature*, 438(7067):504–507, November 2005.

[213] Koichi Fujimoto and Satoshi Sawai. A design principle of group-level decision making in cell populations. *PLoS Comput Biol*, 9(6):e1003110, June 2013.

[214] J. C. W. Locke and M. B. Elowitz. Using movies to analyse gene circuit dynamics in single cells. *Nat. Rev. Microbiol.*, 7(5):383–392, 2009.

[215] Anne E. Carpenter, Thouis R. Jones, Michael R. Lamprecht, Colin Clarke, In H. Kang, Ola Friman, David A. Guertin, Joo H. Chang, Robert A. Lindquist, Jason Moffat, Polina Golland, and David M. Sabatini. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol.*, 7(10):R100, October 2006.

[216] Quanli Wang, Jarad Niemi, Chee-Meng Tan, Lingchong You, and Mike West. Image segmentation and dynamic lineage analysis in single-cell fluorescence microscopy. *Cytometry A*, 77A(1):101–110, 2010.

[217] Michael Held, Michael H. A. Schmitz, Bernd Fischer, Thomas Walter, Beate Neumann, Michael H. Olma, Matthias Peter, Jan Ellenberg, and Daniel W. Gerlich. CellCognition: time-resolved phenotype annotation in high-throughput live cell imaging. *Nat. Methods*, 7(9):747–754, September 2010.

[218] G. Bradski. The OpenCV library. *Dr Dobbs J. Softw. Tools*, 2000.

[219] Gang Lin, Umesh Adiga, Kathy Olson, John F. Guzowski, Carol A. Barnes, and Badrinath Roysam. A hybrid 3d watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks. *Cytometry A*, 56A(1):23–36, 2003.

[220] Pauli Rämö, Raphael Sacher, Berend Snijder, Boris Begemann, and Lucas Pelkmans. CellClassifier: supervised learning of cellular phenotypes. *Bioinformatics*, 25(22):3028–3030, November 2009.

[221] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol. TIST*, 2(3):27, 2011.

[222] Gabriel E. Dilanji, Jessica B. Langebrake, Patrick De Leenheer, and Stephen J. Hagen. Quorum activation at a distance: spatiotemporal patterns of gene regulation from diffusion of an autoinducer signal. *J. Am. Chem. Soc.*, 134(12):5618–5626, March 2012.

[223] Woon Sun Choi, Dokyeong Ha, Seongyong Park, and Taesung Kim. Synthetic multicellular cell-to-cell communication in inkjet printed bacterial cell systems. *Biomaterials*, 32(10):2500–2507, April 2011.

[224] Burkhard A. Hense, Johannes Müller, Christina Kuttler, and Anton Hartmann. Spatial heterogeneity of autoinducer regulation systems. *Sensors (Basel)*, 12(4):4156–4171, 2012.

[225] Laure Plener, Nicola Lorenz, Matthias Reiger, Tiago Ramalho, Ulrich Gerland, and Kirsten Jung. The phosphorylation flow of the vibrio harveyi quorum sensing cascade determines levels of phenotypic heterogeneity in the population. *J. Bacteriol.*, pages JB.02544–14, March 2015.

[226] Robert J. Prill, Julio Saez-Rodriguez, Leonidas G. Alexopoulos, Peter K. Sorger, and Gustavo Stolovitzky. Crowdsourcing network inference: The DREAM predictive signaling network challenge. *Sci Signal*, 4(189):mr7, August 2011.

[227] Bernhard Steiert, Andreas Raue, Jens Timmer, and Clemens Kreutz. Experimental design for parameter estimation of gene regulatory networks. *PLoS ONE*, 7(7):e40052, July 2012.

[228] Brian Munsky, Gregor Neuert, and Alexander van Oudenaarden. Using gene expression noise to understand gene regulation. *Science*, 336(6078):183–187, April 2012.

[229] Avigdor Eldar and Michael B. Elowitz. Functional roles for noise in genetic circuits. *Nature*, 467(7312):167–173, September 2013.

[230] Aleksandra M Walczak, Andrew Mugler, and Chris H Wiggins. A stochastic spectral analysis of transcriptional regulatory cascades. *PNAS*, 106(16):6529–6534, April 2009.

[231] R. J Allen, P. B Warren, and P. R Ten Wolde. Sampling rare switching events in biochemical networks. *Phys. Rev. Lett.*, 94(1):18104, 2005.

[232] Nils B. Becker and Pieter Rein ten Wolde. Rare switching events in non-stationary systems. *J. Chem. Phys.*, 136(17):174119–174119–15, May 2012.

[233] Torsten A. Enßlin. Information field dynamics for simulation scheme construction. *Phys. Rev. E*, 87(1):013308, January 2013.

[234] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *Ann. Math. Stat.*, pages 79–86, 1951.

[235] Balth Van der Pol. On “relaxation-oscillations”. *Lond. Edinb. Dublin Philos. Mag. J. Sci.*, 2(11):978–992, 1926.

[236] Granville Sewell. *The Numerical Solution of Ordinary and Partial Differential Equations.* Wiley-Interscience, Hoboken, N.J, 2 edition edition, July 2005.

Publications

The work featured in this thesis has been submitted for publication in peer-reviewed journals.

[237] Tiago Ramalho, Marco Selig, Ulrich Gerland, and Torsten A. Enßlin. Simulation of stochastic network dynamics via entropic matching. *Phys. Rev. E*, 87:022719, Feb 2013.

[238] Tiago Ramalho, Andrea Meyer, Andrea Mückl, Korbinian Kaspner, Ulrich Gerland, and Friedrich C. Simmel. Single cell analysis of a bacterial sender-receiver system. *submitted for publication.*

[239] Tiago Ramalho, Hao Wu, and Ulrich Gerland. On the controllability of pattern formation by local interactions. *submitted for publication.*

[240] Tiago Ramalho and Ulrich Gerland. Noise-dependent optimal shapes of morphogen profiles. *in preparation.*

Acknowledgements

I am extremely lucky to have been able to always follow the path that fascinated me the most, and was only possible thanks to the constant and unconditional support of my parents, to whom I am deeply grateful. The strong encouragement provided by my dear sister has also been a positive influence over the years which I will not forget.

Thanks to Mariana who's always been on my side and helped me push through whenever I was demotivated. These past years were all the better thanks to your presence. Of course, I will not forget the good times spent with friends, both those I met in Munich and those abroad. Your names are too many to list here, but rest assured I treasure every good memory we share.

I had a great time with Patrick Hillenbrand, with whom I shared the office where we worked on our theses and who always can generate fascinating discussions. I am also very happy to have met all the other students and postdocs both in our chair and the Frey chair, with whom I always had a great time with, such as Alex, Brendan, Christina, Karl, Vladimir, Nanni, Severin and many others. I also had a great time working with master students Hao and Kathrin who not only provided great work, but also helped me see things from new perspectives.

I am also thankful for the great work performed by all the people I've had the pleasure of collaborating with, both PhD students such as Andrea Meyer, Marco Selig and Eric Smith, as well as their advisors Fritz Simmel, Thorsten Enßlin and Thomas Gregor. And finally thanks to Uli, who provided me with great freedom to explore my scientific curiosity and a very relaxed working environment.