

IMPERIAL COLLEGE LONDON

PHD THESIS

Data-driven modelling of robust Turing patterns in synthetic biofilms

by
Martina Oliver Huidobro

Supervisors: Prof. Robert G. Endres
Prof. Mark Isalan *Examiners:* Prof. Eamonn A. Gaffney
 Dr. Jose Jimenez Zarco

*A thesis submitted in fulfillment of the requirements
for the degree of PhD
in the*

Faculty of Natural Sciences
Life Sciences Department

December 20, 2023

“We can only see a short distance ahead, but we can see plenty there that needs to be done”

Alan M. Turing

IMPERIAL COLLEGE LONDON

Abstract

Faculty of Natural Sciences

Life Sciences Department

PhD

Data-driven modelling of robust Turing patterns in synthetic biofilms

by Martina Oliver Huidobro

The generation of robust spatial patterns in biological systems remains a significant yet largely unresolved question. Beyond fundamental science, answering this question would lead to ground-breaking advances in the generation of synthetic tissues, organoids, and new biomaterials. Among various hypotheses, Alan Turing proposed a model to explain pattern formation based on reaction-diffusion networks. These networks are formed of components which travel throughout space and react with each other, leading to spatial periodic patterns. However, this model is far from biological complexity and requires fine-tuning of the parameters to produce such patterns. Furthermore, natural relevant phenomena such as non-linearities, large network sizes, growth and different boundary conditions are often not addressed in Turing models. In this mathematical study, our primary objective is to investigate the characteristics of Turing patterns when these realistic phenomena are introduced. We also aim to enhance the Turing robustness of an engineered reaction-diffusion circuit to guide experimental design. Our final aim is to replicate and scrutinise experimental results where patterning occurs in bacterial tissues with synthetic reaction-diffusion gene circuits. Using high-throughput analytical and numerical methods, we examine how predictions from linear stability analysis are affected by multi-stability, absorbing boundary conditions, and growth. Subsequently, by modelling the synthetic reaction-diffusion circuit, we identify strategies to increase Turing robustness. This enhanced robustness is then experimentally tested to observe periodic patterns. Furthermore, we model these experiments by integrating a colony growth model with a PDE solver for non-linear reaction-diffusion models. To align our mathematical model more closely with experimental data, we employ machine learning techniques for parameter estimation. Our model successfully replicates experimental results, shedding light on the path forward to engineer robust patterns for downstream biotechnology applications.

Declaration of Authorship

I, Martina Oliver Huidobro, declare that this thesis titled, "Data-driven modelling of robust Turing patterns in synthetic biofilms" and the work presented in it are my own. Where information has been derived from other sources or work done in collaboration with others, I confirm that this has been indicated appropriately.

Copyright declaration

'The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work'

Acknowledgements

Although I have declared this thesis is entirely my work, it would not have been possible without the unconditional support of a group of people with whom I have shared this journey.

First off, I must express my gratitude to my supervisors, Mark and Robert, for enabling this exciting collaborative work. Robert, you've taught me since my undergraduate studies all the fascinating ways in which mathematics can be applied to biology. I will forever be thankful for your trust and support in this learning journey. Mark, your enthusiasm for synthetic biology has been contagious. Thank you for showing me how big we can dream when modifying these tiny cells. I wish you a very labyrinthian future! Both of you have played crucial roles in steering me towards an exciting future career where I get to combine two of the most current exciting fields: mathematics and biology.

I would also like to mention all my colleagues who've now become dear friends and have contributed significantly to the warm memories of this journey. Thanks to Antonio and Roozbeh from Robert's group, as well as the rest of my friends in the biomath community at Imperial for all the exciting exchange of ideas and fun conferences and socials. I would also like to thank my colleagues in Mark's group, including Jure, Emily, Tong and in particular Georg with whom I no doubt will cross paths in the future. I am also grateful to the CDT of BioDesign Engineering for their funding and support. The CDT cohort has been a source of inspiration and friendships, and I look forward to maintaining these connections.

I also want to acknowledge my friends in London who have been like a second family. Many of them are also pursuing PhDs, sharing in both the challenges and joys of this journey. A big thanks especially to Nico, who has been there for me through the best and the worst, making my days in London – and not in London – always exciting.

Finally, I would like to greatly thank my family for their invaluable support. Special thanks to my Mum for always pushing us to have the best education, while giving us all the love and support. No doubt we would not achieve these things without you. Big thanks to my Dad as well, who has given us the creativity and passion which are needed to enter this research journey. To Carmen and Jaime, my mathematician siblings, thank you for all the nerdy talks and adventurous times. Finally, big thanks to the rest of the family and in particular my grandparents who have created the tight bonds that are now the roots for me.

Contents

Abstract	ii
Declaration of Authorship	iii
Copyright Declaration	iv
Acknowledgements	v
1 Introduction	1
1.1 Background on biological patterning	1
1.1.1 Morphogenesis in nature, examples and function	1
1.1.2 Patterning theories	3
1.1.3 Wolpert's Positional information vs. Turing's Reaction-Diffusion	4
1.2 Reaction-diffusion systems	7
1.2.1 Turing patterns	7
1.2.2 Gierer-Meinhardt model	7
1.2.3 Turing patterns in nature	9
1.3 Mathematical analysis of Turing patterns	9
1.3.1 Mathematical methods to study Turing patterns	9
1.3.2 The robustness problem	10
1.3.3 Robustness of Turing patterns in realistic systems: larger networks, growth, boundaries, noise, agar thickness	11
Large networks	11
High cooperativity	12
Growth	13
Boundary conditions	15
Noise	16
Delays	17

1.4	Engineering biological reaction-diffusion patterns	17
1.4.1	Previous work on engineering spatial patterns	18
1.4.2	Engineering Turing patterns in bacterial biofilms using exogenous circuit	21
1.5	Thesis overview	24
2	Beyond classical Turing patterns	26
2.1	Synopsis	26
2.2	Linear stability analysis and the dispersion relation	27
2.2.1	Linear stability analysis for a reaction-diffusion system	27
	Stability of steady state without diffusion	27
	Stability of steady state with diffusion	29
	Linear stability analysis implementation	33
2.2.2	Model definition and database creation	33
2.2.3	Diffusion-driven instabilities in the dispersion relation	33
2.3	Correlating linear stability analysis and numerical solutions	35
2.3.1	Inferring wavelength and convergence time from dispersion relation	35
2.3.2	Dispersion to pattern shape	37
2.4	Breaking linear stability analysis predictions	38
2.4.1	Multistability in Turing	39
2.4.2	Analytical to numerical: Other types of dispersion relations, and other types of patterns	43
2.5	Introducing biological features: Absorbing boundaries and growth . .	49
2.5.1	Reflecting boundaries to absorbing boundaries	52
2.5.2	Open boundaries to growth	54
2.6	Discussion	57
3	Gene circuit design and parametrisation	62
3.1	Building model describing synthetic gene circuit	63
3.1.1	Protein equations and gene regulation	64
3.1.2	Tuning gene expression: aTc regulation of TetR and IPTG regulation of LacI	66
3.1.3	Degradation	69
3.1.4	Diffusor equations	69
3.1.5	Six-equation model	71
3.1.6	Dimensionless six-equation model	73
3.1.7	Fluorescence reporters	75
3.2	Explore parameter space and optimise robustness	75
3.2.1	Definition of parameter space based on literature parameters .	75
3.2.2	Finding Turing in six-node Turing circuit and defining robustness	77

3.2.3	Increasing Turing robustness by matching consequent dose-response curves	78
3.2.4	Parameter fine-tuning of the gene circuit to increase the likelihood of Turing patterns	80
3.3	Constrained parametrised distributions: fitting to liquid culture data of gene subcircuits	83
3.3.1	Steady-state subcircuit equations for fitting.	84
3.3.2	Fitting process and the resulting best-fit distributions.	85
3.4	Discussion	90
3.4.1	Model of synthetic gene circuit #1754	90
3.4.2	Finding Turing instabilities and optimising Turing robustness .	91
3.4.3	Model parametrisation	92
4	Patterning in Synthetic Bacterial Colonies	95
4.1	Experimental rings in small colonies with high aTc	95
4.2	Modelling framework for synthetic circuit in bacterial tissues	97
4.2.1	Alternating Direction Implicit Method with defined domains .	98
4.2.2	Static Shaped domains	98
4.2.3	Growing colony with cellular automaton	99
4.2.4	Adapting time and space in the dimensionless model	101
4.3	Colony pattern dynamics in parameter space searches	102
4.3.1	Circuit exploration in literature-based distribution	103
4.3.2	Circuit exploration in liquid-culture fits distribution	104
4.3.3	Pattern exploration around the vicinity of Turing fit solutions .	106
4.4	Elucidating mechanisms through experiment controls	109
4.4.1	Irregular growth	109
4.4.2	Boundary effects	110
4.4.3	Node deletions	111
4.4.4	Temperature variations	113
4.5	Discussion	115
4.5.1	<i>In-vitro</i> <i>E. coli</i> colonies with periodic patterns	115
4.5.2	Solving PDEs in shaped biofilms	116
4.5.3	Replicating bacterial colony patterns	117
4.5.4	Elucidating mechanisms by modelling experimental controls .	118
5	Discussion and Conclusion	120
6	Methods	124
6.1	Finding the steady states: Newton-Raphson method	124
6.2	Sampling method	125
6.3	PostgreSql database	127
6.4	Wavelength and convergence time from numerical data	128
6.5	Dispersion peak height optimisation: Adapted random walk Metropolis	128

6.6	Numerical solution by finite-difference methods	130
6.6.1	Crank Nicolson Method	131
	Introducing Dirichlet or absorbing boundary conditions	132
6.6.2	Alternating Direction Implicit Method	133
	Introducing Dirichlet or absorbing boundary conditions	134
6.6.3	Analysis of numerical solution	134
	Time for pattern convergence	134
	Wavelength prediction from numerical solution	134
6.7	Image recognition and modelling for confocal microscopy data	134
6.7.1	Tissue area recognition	134
6.7.2	Plotting superposed numerical solution as confocal microscopy results	135
6.8	Experimental methods	135
6.8.1	Transformation through electroporation	135
6.8.2	Microscopy	136
	Bibliography	139
	A Appendix: Model parameters for Chapter 2	151
	B Appendix: Model parameters for Chapter 4	153
	C Appendix: Growth regimes in cellular automata model	156
	D Appendix: Experimental conditions	159

CHAPTER 1

Introduction

1.1 Background on biological patterning

1.1.1 Morphogenesis in nature, examples and function

Periodic spatial structures in biology emerge when a tissue of cells develops heterogeneity and complexity in the spatial domain, leading to the formation of regular patterns. These spatial structures are widely observed in nature, both in two-dimensional (2D) spaces, such as the angelfish's striped skin, and in three-dimensional (3D) like the labyrinthine brain cortex. Fig. 1.1 displays a diverse array of spatially periodic patterns encountered during my PhD thesis, highlighting the prevalence of these intriguing patterns across various organisms and natural environments. Although these patterns are extremely common, a comprehensive mechanism to explain their robust assembly remains elusive to the scientific community. This topic garners interest from both fundamental and applied scientific perspectives. Understanding pattern formation in biology is crucial for comprehending how multicellular organisms develop from a single cell into structures of organized cells with complex functions. From an applied standpoint, such insights have the potential to drive advancements in the synthesis of tissues for regenerative medicine, in the development of structured organoids, or in the creation of new biomaterials.

This evolution from simplicity to complexity often plays a pivotal role in the functioning of multicellular beings. More specifically, these patterns can have strategic advantages. Fractal-shaped bacterial colonies, for instance, maximise nutrient absorption (Matsushita and Fujikawa 1990). Furthermore, distinct colourations, such as the zebra's stripes or the mesmerising eye-spots on butterfly wings (Blest. 1957),

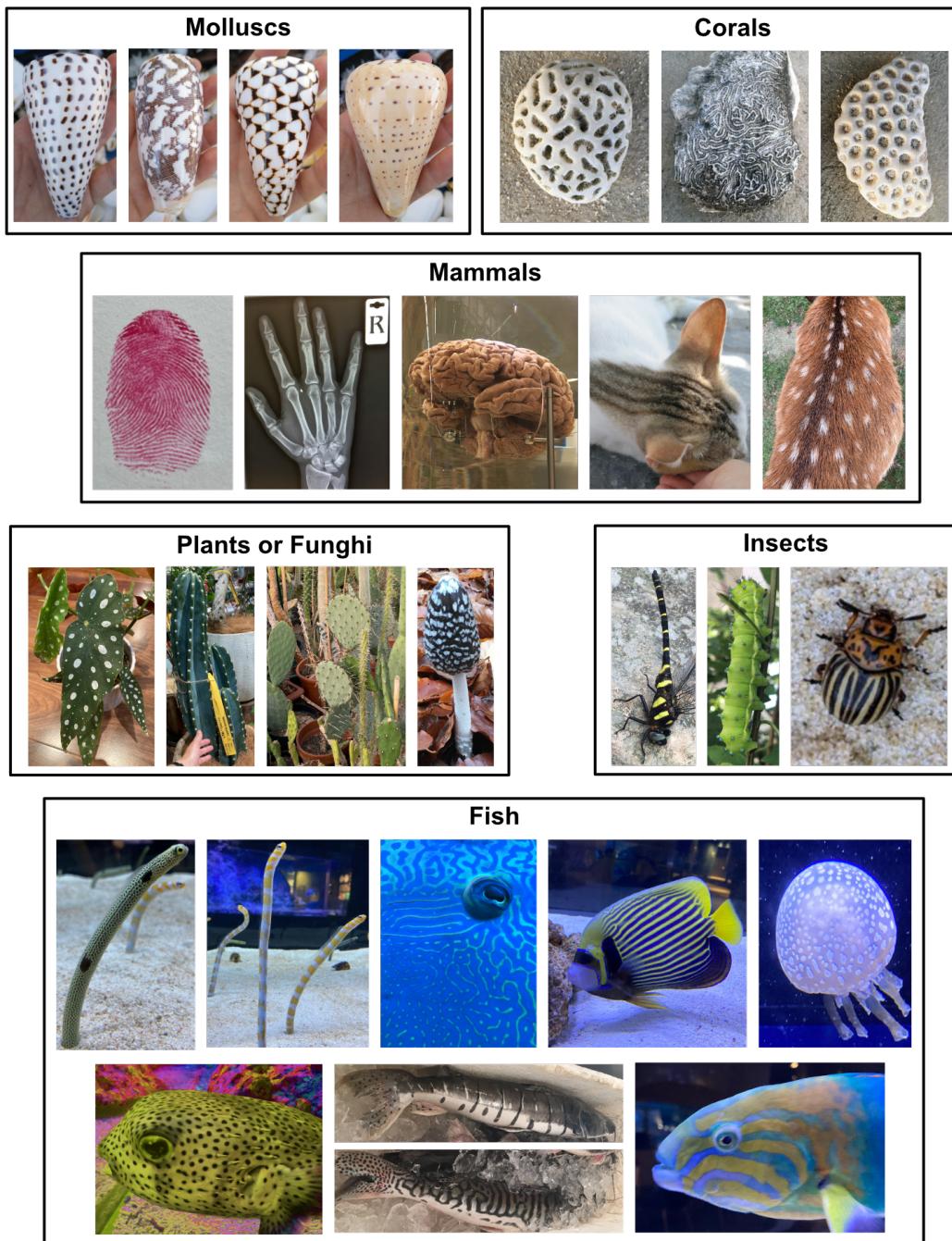


Figure 1.1: Examples of periodic patterns in biology. Patterns observed during the course of this PhD study in molluscs, corals, mammals, plants, fungi, insects and fish. A wide variety of patterns including stripes, labyrinths and spots are displayed. Some organisms even display multiple patterns (e.g. pattern surrounding the fish eye shows labyrinths, spots and stripes (Fish, top row, blue)

serve to disorient predators; either by disrupting the prey's silhouette or suggesting the prey is part of a larger entity (Stevens et al. 2006). The spirals observed in phyllotaxis offer plants a way to optimise sunlight capture, enhancing photosynthesis (Strauss et al. 2020).

The evolutionary advantage conveyed by this level of spatial organisation hints towards genetic networks being potentially responsible for such patterning mechanisms (Caro 2005). It's conceivable that evolution, through a process of random variation and natural selection, could stumble upon a pattern-forming solution. If such a pattern proves to be evolutionarily advantageous, the genetic networks driving these spatial arrangements would likely be favoured and selected over time. Comprehending these networks and the dynamics that lead to these spatial designs is essential. This area of research not only tries to understand which genetic networks and mechanisms are behind patterning. It also seeks to uncover how on a molecular scale these mechanisms can operate reliably and accurately in the inherently noisy and variable biological environment..

Additionally, the study of pattern formation extends well beyond the realms of developmental biology and morphogenesis. It lays the groundwork for innovative advancements in biotechnological sectors. This includes enabling the creation of intricately patterned tissues, efficient biofilms, and potentially even organoids with significant implications in groundbreaking applications like tissue regeneration and organ implants (Scholes and Isalan 2017). The field's promise is further evidenced by existing applications, such as polyamide membranes featuring periodic Turing structures for water purification (Tan et al. 2018), showcasing the practical potential of these developments.

1.1.2 Patterning theories

Numerous mechanisms have been proposed in the literature to explain pattern formation in biological systems. Some of the most relevant ones can be categorized into physical and diffusion-based mechanisms (Hiscock and Megason 2015; Scholes and Isalan 2017), as seen in Fig. 1.2 and Fig. 1.3. Physical-based mechanisms involve models where physical forces play an important role. Some mechanisms in this category include phase separation, which can occur through differential cell adhesion, wrinkling, which can occur through mechanical instabilities, or branching (Fig. 1.2). For example, forces and tissue mechanics are important for patterning villi in the gut, as seen in Fig. 1.2A (left) (Shyer et al. 2013), while differential cell adhesion is key for the self-assembly of pancreatic islets (Jia et al. 2007). On the other hand, diffusion-based mechanisms rely on cells in the tissue responding to a molecule that is spatially heterogeneous and therefore exhibits different phenotypes in different regions of the tissue. Among these gradient-based mechanisms, notable examples include the positional information model and the Turing model (Turing 1952; Wolpert 1969), as seen in Fig. 1.3. The focus of this thesis is on diffusion-based mechanisms, and we will therefore explore these more in detail.

Physical-based mechanisms

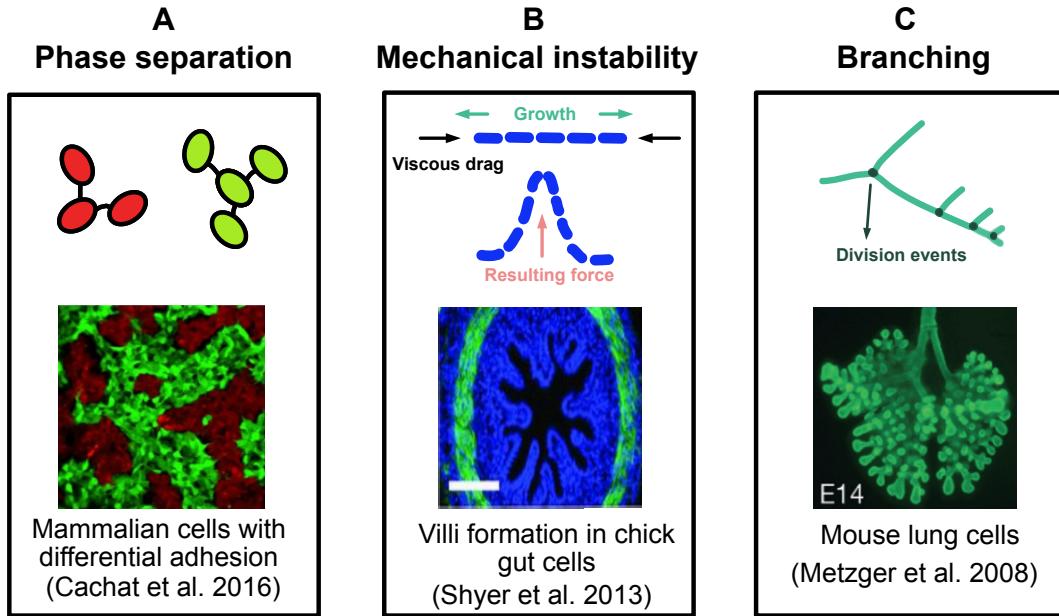


Figure 1.2: **Patterning mechanisms in biology.** **(A)** Phase separation can occur through different mechanisms including differential cell adhesion where different cell types bind to themselves. An example is the synthetically engineered patterns with differential cell adhesion. Image from Cachat et al. 2016. **(B)** Mechanical instability can occur through combined forces, as in the wrinkling of villi in the gut, where growth can generate perpendicular forces that deform the tissue. Image from Shyer et al. 2013. **(C)** Branching can occur through local cell division events which lead to tree-like structures in the lung. Image from Metzger et al. 2008.

1.1.3 Wolpert's Positional information vs. Turing's Reaction-Diffusion

The positional information, also called the French-Flag model, is underpinned by an initial gradient of morphogens, interpreted by a genetic network to induce tissue patterning (Wolpert 1969). Morphogens are signalling molecules which can act over long distances, classically by diffusion, and induce responses in the cells such as activation of gene expression (Rogers and Schier 2011). Sometimes, the origins of this morphogen gradient can be attributed to external factors such as ambient temperature, maternal impacts, or light exposure (Schier 2009). A simple genetic network interpreting this gradient can partition the tissue into three regions where the first region (green) appears for concentrations of morphogen above Threshold 1, and the third region (blue) results for concentrations below Threshold 2 (Fig. 1.3A (top)). An example of positional information occurs in the drosophila embryo, where different gene expression domains are generated by the bicoid gradient (Fig. 1.3A (bottom)). In this case, the mother acts as a source by providing bicoid mRNAs in the anterior of the fly, which diffuse towards the posterior, generating different gene expression profiles (Grimm et al. 2010).

Diffusion-based mechanisms

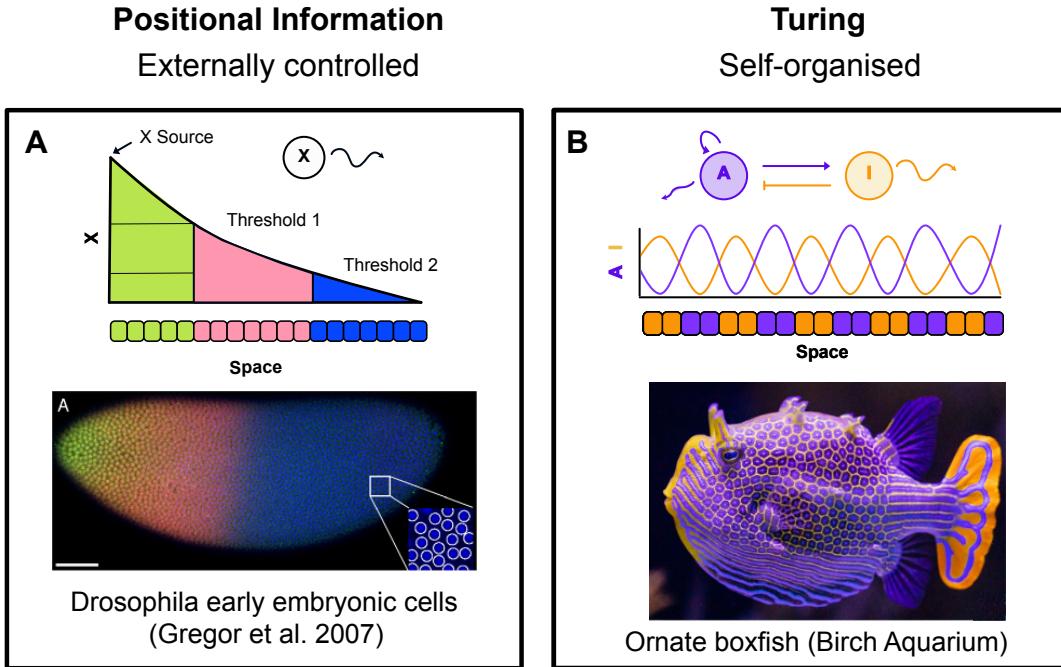


Figure 1.3: **Patterning mechanisms in biology.** (A) Positional information occurs with a gradient of morphogen X, steaming from a source. This gradient is then interpreted by cells forming three different gene expression regimes and phenotypes (green, pink, blue). The example shows an early drosophila embryo stained for DNA (blue), Hunchback protein (red), and Bicoid protein (green) with three partitioned regions. Image from Gregor et al. 2007. (B) The Turing mechanism involves a network with a slow diffusing activator (purple) and a fast diffusing inhibitor (yellow). When they interact, they form a periodic stationary pattern which can lead to spatial spots, stripes or labyrinths in biology as seen in the Ornate boxfish. Photo courtesy of the Birch Aquarium at the Scripps Institution of Oceanography.

Positional information has several caveats. The number of repeats does not scale up with tissue size, meaning bigger tissues have the same number of peaks with a larger wavelength. Therefore the periodicity observed in some biological systems cannot be replicated. Additionally, in specific instances, the presence of an initial pre-pattern might be implausible, raising questions about how patterns spontaneously emerge from a prior uniform tissue. Addressing these conundrums, the Turing model offers an alternative which provides scaling of repeats where the wavelength remains constant for larger tissues as more peaks arise. Additionally, it does not need a pre-existing gradient and is self-organising (Kondo and Miura 2010; Turing 1952).

This mechanism is based on a simple network with a slow-diffusing activator and a fast-diffusing inhibitor which interact together to generate an emergent periodic pattern. An example of this is shown in Fig. 1.3B (top) where the concentrations of the two molecules, A and I, form an out-of-phase spatial wave. In two dimensions,

the Turing model results in complex patterns such as labyrinths, spots and stripes observed in many organisms including the Ornate Boxfish (Alessio and Gupta 2023) (See Fig. 1.3B (bottom)).

Over the past 70 years, the acceptance of Turing's reaction-diffusion (RD) model and Wolpert's positional information theory in developmental biology has evolved significantly (Green and Sharpe 2015b). When Turing's reaction-diffusion model was introduced in 1952, it did not gain traction within the biological community due to various reasons. Originally, it was a theoretical concept with limited empirical evidence as the field of molecular and developmental biology was still not very advanced. Additionally, the theory was presented on a complex mathematical framework which was hard to understand and which described an overly simplistic and rigid network that did not reflect biological complexity. When Wolpert's positional information was introduced in the 70's, it immediately became popular and was established as the dominant paradigm in the field of developmental biology. It provided a conceptual framework which was easier to grasp and which could be directly applied to the observations and experiments in developmental biology such as the growth of limb buds in the chick embryo (Saunders J 1968). The concept of reaction-diffusion, previously introduced by Turing, was revived in 1972 by Gierer and Meinhardt, who proposed a refined version of Turing's model, addressing some of its limitations and making it more biologically plausible. Additionally, they introduced the general principle of "short-range activation, long-range inhibition", which made Turing's RD model more intuitive. It is important to mention that while Turing's results were calculated by hand, Gierer and Meinhardt's calculations were done using computers. These advances in computer technology along with progress in molecular biology also allowed a better understanding of reaction-diffusion models which began to attract attention.

Given its attributes, the reaction-diffusion model might provide a more complete answer to the question of patterning in biology and a more holistic perspective on the intricacies of morphogenesis. For this reason, the Turing mechanism for pattern formation will be the main focus of this work. Nevertheless, it is pivotal to recognize that these mechanisms are not strictly independent of each other (e.g. digit formation has morphogen gradients, while it is likely a Turing mechanism (Raspopovic et al. 2014)). The intricate nature of biological patterning might be best elucidated by an amalgamation of these theories, suggesting a blended framework may hold the answers (Green and Sharpe 2015a).

1.2 Reaction-diffusion systems

1.2.1 Turing patterns

Reaction-diffusion systems were originally introduced in 1952 by Alan M. Turing in his paper “The chemical basis of morphogenesis” (Turing 1952). In this article, he argues that pattern formation can be obtained when two morphogens interact with each other and diffuse at different velocities across a tissue. If these morphogens can activate the production of growth hormones or skin pigments, biological patterns leading to a phenotype can arise. The resulting patterns, called Turing patterns (TPs), can have different shapes such as stripes, labyrinths, and spots; which can be widely observed in natural systems. The concept of Turing patterns was first introduced from a mathematical perspective and was modelled using space and time-dependent partial differential equation (PDE). In the case of Turing’s seminal paper, the set of PDEs describes a two-node network (Fig. 1.4A) which has activation and inhibition terms, degradation terms and diffusion terms (Fig. 1.4B). The key feature of this system is that it can exhibit diffusion-driven instabilities: Initially, the tissue has a homogeneous concentration of morphogen under which no diffusion occurs, and under these circumstances, the system is stable and converges into an equilibrium state. When biological noise is introduced, the heterogeneity of the system leads to morphogen diffusion. Consequently, the diffusion leads to changes in the reaction rates and the system is pushed out of steady state leading to an unstable system. This unstable system converges into a stable stationary periodic pattern which is a Turing pattern (Fig. 1.4C). This whole phenomenon, called diffusion-driven instability, is the essence of Alan Turing’s paper and one of the most used mechanisms to explain morphogenesis.

1.2.2 Gierer-Meinhardt model

Although Turing’s work was of key importance in the field of developmental biology, it was not well accepted by biologists due to various reasons. Firstly, the mathematical complexity behind it made it non-intuitive for many scientists in the field and hard to explore due to the lack of computers. Furthermore, the network proposed, although it can generate patterns, is too simple to describe the complexity in biological systems and produces some unrealistic results such as the presence of negative concentrations due to the lack of non-linearities (Kondo and Miura 2010). To solve these two issues, Gierer and Meinhardt proposed a generalised version of Turing’s diffusion-driven-instabilities (Gierer and Meinhardt 1972). This alternative theory proposes that patterns can be obtained as long as short-range activation and long-range inhibition (SALI) are present. This implies that we can go away from Turing’s equations and extend the theory to networks with different numbers of nodes, different network interactions and even different signal transduction mechanisms (Murray et al. 1983; Rauch and Millonas 2004; Swindale 1980). This step was

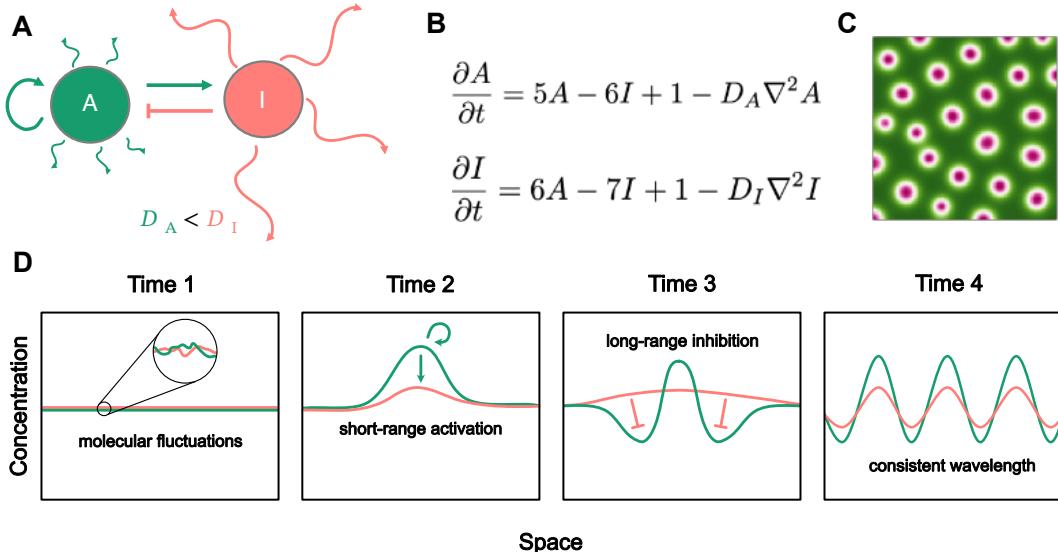


Figure 1.4: Introduction to Turing mechanism of pattern formation. (A) Two-node network based on Turing's original paper. Slow diffusing activator in green and fast diffusing inhibitor in pink. (B) Linear PDE equations for a two-node Turing system. (C) 2D simulation of equations in (B). (D) Here we display the mechanism of pattern formation with an activator-inhibitor system for an in-phase pattern. The system starts with an apparent homogeneous distribution of activator and inhibitor which displays fluctuations at the molecular scale (Time 1). Where the activator prevails over the inhibitor in these fluctuations, local activation occurs by self-activation of the activator (green) and activation of the inhibitor (pink). This is known as short-range activation (Time 2). Finally, the inhibitor diffuses further away producing long-range inhibition where the inhibitor represses activator concentration in the vicinity of the local peak (Time 3). As a result, inhibitor concentration also drops, meaning that the levels of inhibitor are not enough to repress the activator which generates another local peak. This leads to an organised system of periodic peaks and troughs of molecule concentration which dynamically settles into a stationary pattern (Time 4).

necessary to be able to attribute patterning phenomena to more complex cellular and molecular interactions described by non-linear terms and larger networks with more nodes involved. Examples of the first systems developed with non-linear terms are the typical Gierer-Meinhardt model (Gierer and Meinhardt 1972), Schnakenberg model (Schnakenberg 1979), as well as the Thomas model (Thomas and Kernevez 1976). This generalisation work sheds some light on the logic behind patterning and allows us to understand it intuitively: as an initially homogeneous system is perturbed by biological noise, local peaks of a morphogen concentration form. These transient peaks lead to a local activation effect where the concentration of all reactants increases. Due to the difference in diffusivity, the inhibitor reaches further away and long-range inhibition is achieved. Ultimately, this system settles into a stationary pattern with activation peaks and inhibition troughs (Gierer and Meinhardt 1972). This SALI mechanism proposed in this paper is shown in Fig. 1.4D.

1.2.3 Turing patterns in nature

Various links have been made between biological patterns found in nature and Turing patterning networks. As a basic example, seashell pigmentation and various types of fish skin have been replicated using simulations of Turing pattern systems. Additionally, perturbation experiments in zebrafish's skin are consistent with simulations of RD equations where the pattern regenerates in the exact same way after being physically disrupted both *in-vivo* and *in-silico* (Kondo and Miura 2010). Perturbation experiments are also carried out in the mammalian palate, providing evidence of a Turing-type reaction-diffusion mechanism (Economou et al. 2012).

From a molecular biology point of view, molecules involved in patterning and morphogenesis have been shown to be part of networks with SALI characteristics. For example, hair follicle development and fingerprint formation have been shown to rely on the same Turing reaction-diffusion network, based on signalling between WNT and BMP (Glover et al. 2023). In this case, WNT is the activator while BMP is the inhibitor. WNT directly promotes cell proliferation and indirectly hair follicle generation through synergy with the EDAR pathway. Therefore, in the presence of WNT, there will be growth and hair follicle formation. WNT distribution in space is periodic, likely due to a Turing mechanism, and therefore forms periodic hair follicle formation and ridges in the fingerprints. Other examples of Turing SALI networks are Nodal & Lefty in right left asymmetry (Nakamura et al. 2006) and finally Wnt & Dkk involved in lung branching (Langhe et al. 2005). All these examples of the relationship between biological patterns and RD systems, SALI networks and diffusion-driven instabilities strongly suggest that the Turing mechanism is linked to the self-assembling and self-regenerative patterns observed in nature.

However, the arguments mentioned above make Turing's mechanism purely a strong hypothesis of patterning and do not prove causation. This is because obtaining a 2D solution that matches the experiments does not necessarily mean the model used is the correct one. Furthermore, the involvement of a Turing-type network in patterning does not discard the possibility of an interplay of multiple mechanisms contributing to this organization. For instance, Bmp, Sox9, and Wnt constitute a Turing network involved in the development of limb digits, as proven by Raspopovic et al. 2014. However, the presence of a Shh gradient in the limb bud suggests this developmental programme could arise from a combination of Turing's reaction-diffusion and Wolpert's positional information.

1.3 Mathematical analysis of Turing patterns

1.3.1 Mathematical methods to study Turing patterns

Turing patterns are commonly studied using mathematical tools to understand their features and behaviours. The equations describing these reaction-diffusion systems

are usually too complex to solve analytically as they contain non-linearities and partial differential terms. For this reason, different methods must be used such as linear stability analysis (LSA) or numerical methods.

LSA provides information on how the stability of the system evolves as we go from a system without diffusion to a system with diffusion (Glendinning 1994). For RD systems to produce Turing patterns, they must be stable without diffusion and become unstable as diffusion is turned on (Murray 2002). This stability profile gives rise to the name diffusion-driven instability (DDI) which is an alternative name for Turing patterns. In principle, this method can tell us whether a system is a pattern generator but does not provide a solution of the pattern in space and time. Additionally, because the system is linearised around the steady state, non-linear terms or multistability might alter the final outcome. More information on how LSA is used to understand patterning and DDI's is found in Section 2.2 in the following chapter.

Alternatively, numerical methods are used to calculate the concentration of species in the RD system at every time and space point (Ramos 1983). A visual solution is obtained which provides information on the shape, wavelength, evolution and amplitude of the pattern. While numerical methods provide more information than LSA, they are more computationally expensive and can lead to numerical errors (Murray 2002). Various numerical solvers exist for this purpose, but after much research, Crank-Nicolson (CN) for one-dimensional (1D) space and Alternating Direction Implicit Method (ADI) for 2D space were considered the most adequate in terms of speed and accuracy for this project (See Section 6.6). To explore and get intuition on the numerical solutions of a system, recently developed online tools such as VisualPDE can be used before studying the problem in detail with other solvers (Walker et al. 2023). Both numerical and analytical tools have their advantages and disadvantages and must be used in combination for an optimal study of RD systems and patterning.

1.3.2 The robustness problem

Using both linear stability analysis and numerical methods, Turing patterns were widely studied to understand whether they were a plausible explanation for pattern formation in biology. Although patterns were obtained analytically and numerically using models of the SALI networks mentioned above, the robustness problem makes this mechanism questionable. The robustness problem or the fine-tuning problem refers to the small fraction of the parameter space leading to Turing patterns and high sensitivity to parameter changes. For example, original 2-node reaction-diffusion systems require large differential diffusion rates between the two morphogens as explained previously for the SALI mechanism to occur. If this constraint in parameters is not met, the robustness for pattern formation is highly reduced.

Achieving this differential diffusivity experimentally is sometimes biologically unlikely, as many of the morphogens have similar diffusion rates due to their similarities in molecular size (Oliver Huidobro et al. 2022).

The robustness issue was explored in detail in Scholes et al. 2019 using an atlas approach, where the parameter space of all 2-node and 3-node networks was studied using LSA to check for Turing patterning. The atlas study involves constructing all possible networks using adjacency matrices, where a -1 represents an inhibition edge from one node to another, 0 represents no interaction and $+1$ represents an activation. All 2×2 and 3×3 adjacency matrices are created and network symmetries are checked to remove homologous networks. By carrying out LSA on the resulting networks, it was found that although 61% of the networks can produce Turing patterns, only 0.1% of the parameter space is within Turing space. Similar results were obtained in Zheng et al. 2016 and Marcon et al. 2016.

Another issue with Turing patterns is the sensitivity to the initial conditions. Although the shape of the pattern is not strictly determined by the initial conditions, finer details like locations of spots and stripes can be influenced. Additionally, although the final pattern is the same, the transient dynamics might differ between different initial conditions (e.g. different transient patterns or convergence times before settling into the final pattern). This can be an issue as in developmental biology, processes are tightly controlled and cells must receive the exact same signals in the same location and time to robustly reproduce the pattern (Perrimon et al. 2012).

1.3.3 Robustness of Turing patterns in realistic systems: larger networks, growth, boundaries, noise, agar thickness

On one hand, Turing's mechanism seems to explain many different patterns in biology as well as their perturbation experiments. On the other hand, it does not seem like a very robust mechanism. How do these two contradicting ideas come together? How can we make patterns more robust? Typically, Turing patterns are often studied in non-realistic systems. A new direction in the field aims to study Turing patterns in more realistic systems and see how these effects affect robustness.

Large networks

Biological networks are usually far from the idealised 2-node Turing networks (Jenssen et al. 2001). Recent studies (Marcon et al. 2016; Scholes et al. 2019; Zheng et al. 2016) have demonstrated that larger networks exhibit increased robustness to variations in parameters. In particular, it has been shown that the addition of an immobile substrate to a two-node diffusible system can relax the differential diffusion constraints typically found in classic SALI topologies (Korvasová et al. 2015). Additionally, a high-throughput study explored four-node networks that evolved from adding a node to the most robust three-node topology identified in Scholes et al. 2019. Some

of these four-node networks were found to be more robust than the original three-node topology. This study, yet unpublished, was conducted by Master students supervised as part of this PhD project. It is crucial to recognize that as network models become larger, their parameter space expands exponentially. Consequently, this can lead to a decrease in robustness when sampling unknown parameters.

However, studying large networks is very computationally expensive because of various reasons, including an exponential increase in possible networks, an increase in dimensions of the parameter space, and an increase in the number of equations. This has resulted in limited studies addressing networks larger than three. Smith and Dalchau 2018 tackles this challenge by simplifying big networks into smaller ones. The reduced model is not an accurate description of the original system. However, it can predict whether a system will not show instabilities, ruling out many cases to study and reducing the computational cost of exploring larger networks. Haas and Goldstein 2021 studies large networks with six diffusors using a random matrix approach and also shows that there is a relaxation in differential diffusion constraints. This linearised random matrix approach, previously used in ecology (May 1972), could be used to explore large networks with mostly non-diffusible components. Although this random matrix method produces a very idealised model, it would resemble the scale of real biological networks and give insights in this respect.

High cooperativity

Turing models based on genetic components, are often modelled using non-linear Hill functions, which can describe the cooperative behaviour of transcription factors (TFs) when binding to DNA (Morgunova and Taipale 2017). This Hill function represents how much gene expression is repressed for a specific TF concentration

$$H(X) = \frac{1}{1 + (\frac{X}{K})^n} \quad (1.1)$$

where $H(X)$ is the fractional activation of gene expression, X is the concentration of transcription factor, n is the cooperativity factor. The higher the cooperativity of the TF, the steeper the sigmoidal Hill function is. This is due to the mechanism of binding of TFs to DNA.

Transcription factors are multimeric, meaning they form complexes of various sub-units. Cooperativity describes how the binding of a subunit to a DNA binding site can affect the binding affinity of other subunits to other binding sites. For example, cI which is a highly cooperative repressor protein used in this thesis, forms dimers to bind to a cI operator sequence in DNA. This binding increases the affinity of other cI dimers to bind other operator sites through DNA looping. Finally, up to twelve subunits can bind to the six operator sites present. This means that with small concentrations of cI, a dimer will bind the DNA and induce the binding of other dimers

resulting in a complete shutdown of gene expression. Therefore the response is relatively binary which can be modelled with a high cooperativity factor (n), resulting in a steep Hill function.

Greater cooperativity, reflected in steeper Hill curves, can expand the Turing parameter space region and reduce the constraints imposed by differential diffusivity (Diambra et al. 2015). In a way, this makes sense as very steep curves turn genetic circuits into something closer to electrical circuits which are more robust to signals if the thresholds are known.

Growth

Fixed domains are often commonly assumed in developmental biology, as pattern-forming processes (e.g. reaction-diffusion of chemical species) occur at much faster timescales than tissue growth. However, models of pattern formation in growing domains have shown growth to be extremely important in the development and morphology of the pattern. Biological examples include angelfish *Pomacanthus imperator* where uniform growth (Fig. 1.5C.i) induces the addition of new stripes in a branching point, while a constant wavelength (Kondo and Asai 1995). In synthetic systems, the importance of growth can also be observed using a chemical reaction-diffusion system that grows over time (Konow et al. 2019). Slow growth rates were likely to produce rings added on the inside of the disk as it grew (inner ring addition), while fast growth rates produced rings on the outside (outer ring addition). Intermediate growth rates formed labyrinthine structures. All these principles of pattern morphology and growth rate are verified in the model. These biological examples of growing patterns highlight the importance of including growth in models of pattern formation, and specifically Turing patterns.

To understand growth we need to look at the different components that define it. The first component is the growth rate function, which can be linear, exponential or logistic, leading to different types of tissue dynamics (Fig. 1.5A). The direction of growth is also key, which can be determined by isotropic or anisotropic growth. Isotropic growth occurs when growth rates are the same in all directions, unlike anisotropic growth where tissue grows in one direction (Fig. 1.5B). Finally, we consider where cell divisions occur. In uniform growth, the tissue expands homogeneously as cell divisions occur throughout the tissue (Fig. 1.5C.i). However, in non-uniform growth, different regions of the tissue grow at different rates. An example of this is apical growth where cell division only occurs at the boundary (Fig. 1.5C.ii). While uniform growth can be used to model growing angelfish (Kondo and Asai 1995), apical growth can be used to model the apical growth of plants or the developing limb bud (Crampin et al. 2002).

Considering the importance of growth in pattern morphology, it is key to consider how this growth might affect parametric robustness in Turing pattern formation.

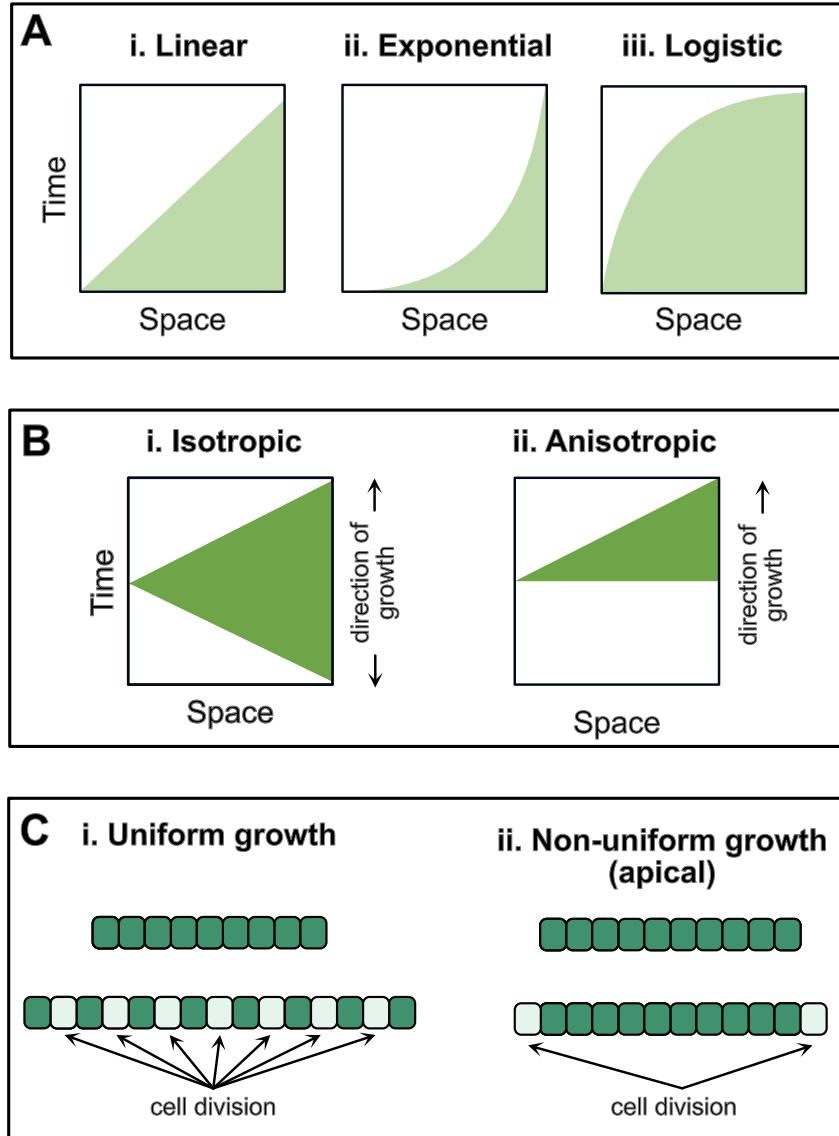


Figure 1.5: **Components of biological growth.** (A) Growth functions can be (i) linear where the growth rate is constant, (ii) exponential where growth rates increase over time or (iii) logistic where growth rates ultimately converge to zero. (B) The direction of growth determines how the tissue evolves, where isotropic growth (i) has the same growth rates in all directions, while anisotropic growth has different growth rates in different directions. (C) Uniformity of growth depends on the growth rates of different regions of the tissue. For uniform growth (i), growth rates throughout the tissue are constant, meaning cell division occurs everywhere and the tissue expands homogeneously. In non-uniform growth (ii), growth rates vary throughout the tissue such as apical growth where growth rates are only positive at the edges.

Interestingly, introducing slow isotropic growth allows certain network topologies to form Turing patterns, which would not without growth. An example of that is patterns in activator-activator networks, showing growth-induced Turing instabilities. Additionally, short-range activation and long-range inhibition can also produce Turing patterning in growing domains (Madzvamuse et al. 2010). Finally, increasing

growth rates can increase the Turing parameter space under exponential growth in Turing patterning networks. However, the analytical analysis used in Madzvamuse et al. 2010 only holds when assuming slow isotropic growth and will break under faster growth.

Under logistic and linear growth, the Turing parameter space also changes. However, the analysis used in Madzvamuse et al. 2010 only holds when assuming slow isotropic growth and will break under faster growths. This analysis is revisited in Klika and Gaffney 2017 where they manage to study faster isotropic growing domains. However, they observe that the conditions for Turing instabilities are more complex with higher growth rates and therefore more challenging to study than the ones in Madzvamuse et al. 2010. This complexity makes it difficult to carry out high-throughput studies of robustness and Turing parameter space in growing domains. Finally, other types of growth such as anisotropic growth (Krause et al. 2019) are studied, but again they do not study parametric robustness in detail.

Boundary conditions

Boundary conditions are a set of conditions that need to be satisfied at the boundary of a system where a set of differential equations is solved. In reaction-diffusion systems, boundary conditions describe what happens to the morphogens as they reach a boundary. In Neumann boundary conditions, the derivative at the boundary is defined. If the derivative is zero, reflective boundary conditions are obtained (Fig. 1.6B (top)). In Dirichlet boundaries, the value at the boundary is defined. Setting the value to zero creates an absorbing boundary where morphogens get leaked from the system (Fig. 1.6B (bottom)). Other boundaries such as periodic boundary conditions exist, where the morphogen passes through one boundary and appears in the other with the same velocity. Biological systems can exhibit this wide variety of boundary conditions depending on the context and environment. Additionally, these boundary conditions can also be replicated using experimental techniques (Krause et al. 2020; Sheth et al. 2012; Vahey and Fletcher 2014).

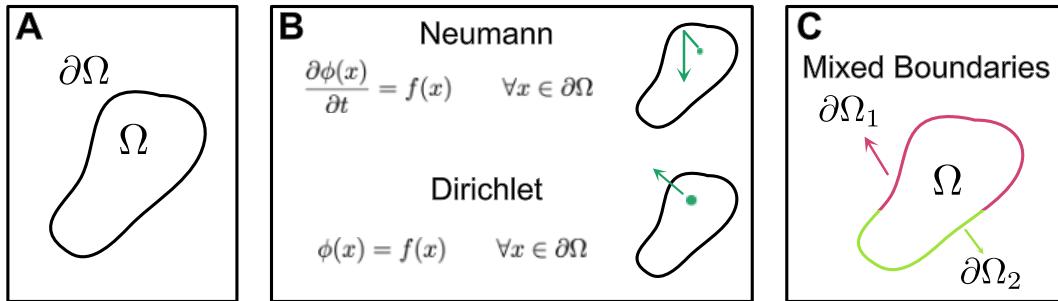


Figure 1.6: **Boundary conditions.** (A) Ω represents the region governed by the differential equations and $\partial\Omega$ represents the boundary curve where the boundary conditions apply. (B) Neumann boundary conditions (top) are used to define reflective boundaries or no-flux where the derivative is zero at the boundary. Dirichlet boundary conditions (bottom) are used to define absorbing boundary conditions where the value at the boundary is zero. (C) Mixed boundary conditions have a different boundary definition in different regions of the curve ($\partial\Omega_1$) and ($\partial\Omega_2$).

Different theoretical studies explore a wide variety of boundary conditions for Turing systems. Initially, it was shown that non-homogeneous boundary conditions where the system is not insulated (Dirichlet), are less sensitive to changes in the domain size, different initial conditions and perturbations in model parameters (Arcuri and Murray 1986). Then, mixed boundary conditions (Fig. 1.6C) where different species are subject to different boundary definitions were also shown to increase parametric robustness (Krause et al. 2021; Maini and Myerscough 1997; Maini et al. 1993). Studying the effects of these boundary conditions analytically is relatively complicated as you need to understand the bifurcation from which the Turing instability arises. For example, Woolley 2022 showed that although a Turing bifurcation is canonically a subcritical pitchfork bifurcation under Neumann boundary conditions, the Turing bifurcation is canonically a transcritical bifurcation under Dirichlet boundary conditions. This shows how much effect the boundaries have on the Turing instability, meaning they need to be considered when studying such systems.

A more applied biological example of how boundaries can affect pattern formation is seen in (Krause et al. 2020) where they study patterning synthetic biofilm in agar plates. In this case, the agar absorbs the diffusors through the boundary, disrupting the pattern formed in the biofilm. This effect can be reduced by decreasing the thickness of the agar layer. Potentially, the pattern could also be maintained by reducing the permeability of the agar through the introduction of a filter paper or reduction of pore size.

Noise

Biological systems are known for their inherent noise. Stochastic Turing patterns were studied by adding intrinsic noise in the production and degradation rates (Biancalani et al. 2010; Butler and Goldenfeld 2009, 2011). In certain non-Turing

parameter regimes, it was found noise could drive diffusion-driven instabilities. If there is a stable mode with eigenvalues slightly below zero, noise can destabilise this mode leading to a noise-induced instability and therefore the emergence of a pattern. More specifically, it was found that the requirements for large differential diffusivity between activator and inhibitor were relaxed. This showed that noise might be a source of parametric robustness, producing patterns where deterministic systems do not predict one. However, the resulting patterns are not as ordered as deterministic Turing patterns (Karig et al. 2018).

Morphological robustness was also studied using intrinsic noise, which stems from prescribing a probability that each reaction occurs, rather than an exact rate. It was shown analytically that stochastically excited wave modes correspond exactly to their deterministic analogues, leading to a similar final pattern. Additionally, by perturbing populations away from the steady state, patterns formed quicker in a stochastic system than its deterministic counterpart (Maini et al. 2012)

Delays

Gene expression delays also affect robustness for Turing pattern formation. These delays stem from the lags of transcription, where mRNAs are produced from DNA reads; and translation, where proteins are synthesised from mRNA reads. These processes occur in the scales of minutes to hours, depending on the genomic size of the organism. From a gene being activated to the protein being active, there is a delay that is often not accounted for when modelling patterning. The pattern morphology has been shown to be extremely sensitive to delays including sensitivity to the initial conditions in the final pattern, dramatic changes in the patterning lag under different delays and even pattern loss when delays are introduced (Buchler et al. 2005; Maini et al. 2012).

Overall, it is important to understand how different phenomena can affect pattern formation and introduce in models the most relevant to the natural system or experimental system being studied. Furthermore, in some cases, these biological phenomena could increase robustness to pattern formation and explain the lack of robustness found in classical Turing pattern studies (Scholes et al. 2019).

1.4 Engineering biological reaction-diffusion patterns

Engineering Turing patterns involves building systems from first principles, based on Turing's reaction-diffusion mechanism, which can replicate natural periodic patterns. To engineer biological Turing patterns specifically, cells and genetic components must be used to create the reaction-diffusion network. There are various reasons to justify the engineering of Turing patterns in biology. These justifications span from enhancing our fundamental understanding of how patterns develop in living

organisms to the potential advancements these synthetic structures could contribute to biotechnology.

To begin with, there is a wealth of biological patterning that is associated with Turing-like gene networks, as indicated by numerous studies referenced in Section 1.2.3. However, due to the tangled nature of biological systems, it is extremely hard to prove a Turing's mechanism is actually behind those biological patterns. Building a biological patterning system based on the principles of Turing's mechanism would prove this mechanism can indeed form patterns in biology. Furthermore, the integration of Turing patterns with real biological scenarios involving growth, stochasticity, or different boundary conditions could shift theoretical models towards more robust patterns that resemble more closely those found in nature.

Finally, the synthetic patterns obtained would contribute to novel nanotechnologies, such as systems with patterned biomaterial deposition (Cao et al. 2017; Din et al. 2020). For this reason, a bottom-up approach is required to build a system from first principles, which is tunable and insulated from the tangled genetic context, and which can be used to build tissues of interest.

1.4.1 Previous work on engineering spatial patterns

Engineering Turing patterns in biology is a highly challenging endeavour that has required synthetic biology to gradually develop the appropriate tools. This has involved creating simpler biological patterns, as well as chemical Turing patterns to gain the necessary insights and tools for this complex task. In synthetic biology, a wide variety of pattern generators have been engineered ranging in different complexities and pattern outcomes. To systematically categorise these synthetic patterning systems, we can group them into four levels based on design characteristics as described below and seen in Fig. 1.7 (Oliver Huidobro et al. 2022):

Level 0 circuits lack synthetic signals that diffuse through normal Fickian diffusion, where the molar flux due to diffusion is proportional to the concentration gradient. Instead, spatial structures emerge from other processes like cellular growth and gene expression freezing. Examples include synchronised oscillator circuits in bacterial colonies producing periodic concentric ring patterns without diffusing signals (Potvin-Trottier et al. 2016; Riglar et al. 2019) (See Fig. 1.7 Level 0).

Level 1 circuits do rely on diffusible components; however, these are not dynamically regulated, meaning they are not produced by the circuit as it is in the case of Turing patterns. These diffusors only act as a pre-pattern which the system will interpret (Barbier et al. 2020; Basu et al. 2005; Grant et al. 2020; Kong et al. 2017; Schaeiri et al. 2014). Although stripes and sharp boundaries can be obtained, their periodicity is limited. Hierarchical patterning circumvents the periodicity issue by increasing the number of orthogonal diffusors used, which linearly increases the number of spatial

domains (Boehm et al. 2018). While hierarchical patterns can explain some developmental periodic patterns such as the neural tube in vertebrates (Briscoe and Small 2015), it fails to capture the self-organising periodicity of Turing patterns observed in development such as digit patterning (Raspopovic et al. 2014; Sheth et al. 2012) (see Fig. 1.7 Level 1).

Level 2 circuits do incorporate dynamically regulated diffusors; however, only one diffusor is introduced. In contrast with level 1 systems, no pre-patterns are required to generate interesting behaviours. Starting from a spatially homogeneous regime, these systems can produce robust spatial oscillators such as (Danino et al. 2010) or simple single ring patterns (Cao et al. 2016; Payne et al. 2013) (see Fig. 1.7 Level 2).

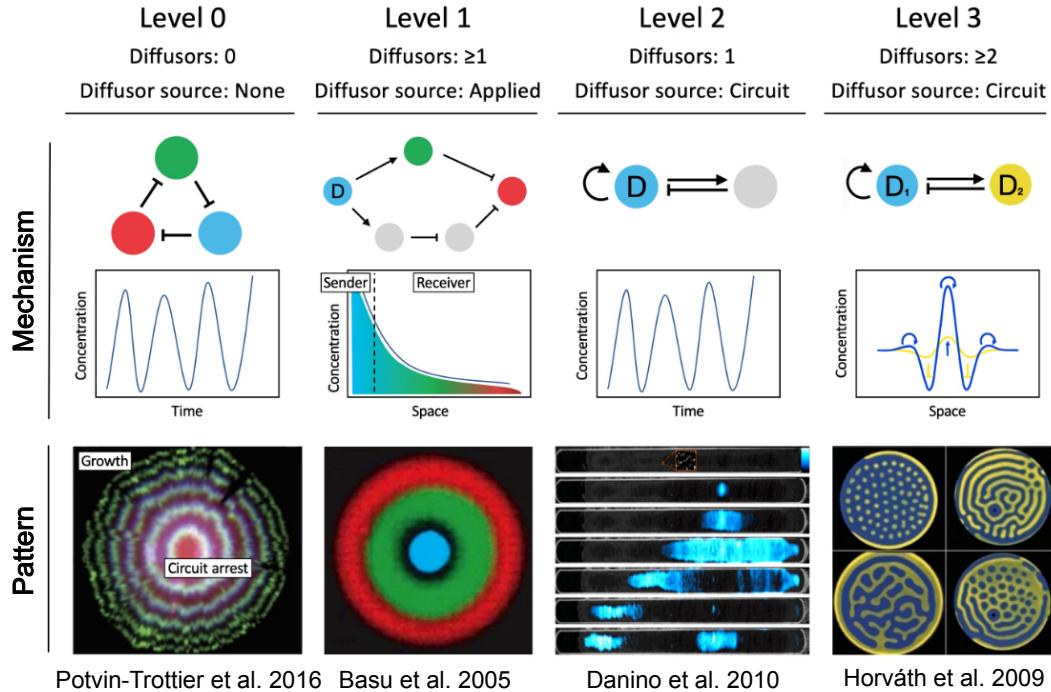


Figure 1.7: Engineered spatial patterns in synthetic biology. This figure is adapted from our published review on synthetic spatial patterning (Oliver Huidobro et al. 2022). Four levels of regulatory complexity in engineered spatial patterning systems. Each level is divided into an example circuit and the resulting pattern upon implementation. Diffusing components of the circuit are labelled with a “D”, non-diffusing nodes are unlabelled. The colour of each node corresponds to the colour of the reporter in the respective implementation. Level 0: synchronized repressilator circuit implemented in a growing bacterial colony (Potvin-Trottier et al. 2016). The plot shows the circuit oscillations in single cells or stirred liquid culture. Level 1: incoherent feedforward circuit, where the diffusor-producing sender cells (cyan) are placed in the middle of a bacterial lawn (Basu et al. 2005). The plot shows the concentration gradient of the diffusor away from the centre of the lawn. Level 2: self-activation and feedback inhibition circuit with one dynamically regulated diffusor creates spatial propagating waves and spatially synchronised oscillations (not shown) (Danino et al. 2010). The plot shows the oscillations of the circuit in single cells, or in a cell population. Level 3: self-activation and lateral-inhibition circuit with two dynamically regulated diffusors creates stationary Turing patterns in the TuIS chemical system (Horváth et al. 2009). The plot shows the localized, self-activating positive feedback of the slow-diffusing species D1 (blue curve) and the lateral inhibition of the fast-diffusing species D2 (yellow curve).

Finally, level 3 circuits use multiple dynamically regulated diffusible components (see Fig.1.7 Level 3). Turing patterns are the most prominent example of Level 3 systems as they are formed by reaction-diffusion circuits of at least two diffusors. While numerous robust systems have been engineered in Level 0,1 and 2 circuits, Level 3 or Turing pattern engineering are still in its infancy. Stochastic Turing patterns

were recently engineered in *E. coli* with a circuit implemented according to the self-activation and lateral inhibition topology, with two diffusible quorum-sensing signals (Karig et al. 2018). Additionally, solitary patterns have also been engineered in a refactored Nodal–Lefty system in HEK cells (Sekine et al. 2018). While easier to engineer due to their relaxed fine-tuning requirements, stochastic Turing patterns and solitary patterns display more irregularity in their periodic spatial structure (Butler and Goldenfeld 2011; Karig et al. 2018; Sekine et al. 2018).

While elusive in synthetic biology, regular-repeat Turing patterns have been engineered in chemical reaction systems. They were first observed in the 1990’s in the chlorite-iodide malonic acid (CIMA) reaction (Castets et al. 1990; Lengyel and Epstein 1992) and later in the thiourea-iodate-sulfite (TuIS) reaction (Horváth et al. 2009) (see Fig. 1.7 Level 3). Unlike biological systems, chemical reactions are reliably described by the simpler laws of mass action, and system parameters can often be identified (Kügler et al. 2009; Pušnik et al. 2019; Turányi 1994; Yeoh et al. 2019). Furthermore, the tuning of these systems by changing initial concentration or temperature has more predictable effects on the system (Asakura et al. 2011; Carballido-Landeira et al. 2010; Horváth et al. 2009). Lastly, chemical systems are isolated from external interacting components, unlike biological systems where burden and cross-talk between the cellular chassis and synthetic parts are inevitable (Butzin and Mather 2018; Ceroni et al. 2015; Du et al. 2020; Nielsen et al. 2016).

Although chemical systems have helped understand pattern formation and could prove key to tissue engineering applications, mechanisms in biology seem to go beyond chemical systems. Examples of patterns have been closely linked to gene expression patterns as shown in *in-situ* hybridization studies (Jing et al. 2006). For this reason, it is key to explore this avenue and build genetically based Level 3 systems that generate robust periodic patterns in biofilms.

1.4.2 Engineering Turing patterns in bacterial biofilms using exogenous circuit

Even though they have been successfully engineered in chemical systems, the complexity of biology means that Turing patterns have never been engineered in a genetic context. This can be attributed to the robustness problem including high sensitivity of parameters and a small fraction of parameter space producing Turing solutions. Additionally, designing Turing circuits with the classical 2-node topology leads to constraints in differential diffusivity that hinder the engineering of Turing patterns. Finally, this robustness issue is further amplified by the lack of orthogonal small-diffusible regulators to act as morphogens in a synthetic reaction-diffusion circuit. Orthogonality refers to the inability of two or more molecules to interact with their respective substrates, so the activator or inhibitor functions work adequately.

To address the fine-tuning problem and differential diffusivity constraints, Scholes et al. 2019 found that topology #3954 was amongst the most robust 3-node networks in terms of kinetic parameters. Additionally, this topology allowed for equal diffusivity and even faster diffusing activators. A search around neighbours of the #3954 topology was carried out to find that removing the positive feedback on node A did not decrease robustness substantially. The resulting #1754 topology seemed like a better candidate as it would be easier to build experimentally while maintaining the robustness. This topology is shown in Fig. 1.8 (grey inset)

The lack of robust biological partly was addressed in Meyer et al. 2019 and Du et al. 2020 where orthogonal morphogens were characterized, which allowed the building of the #1754 topology. These orthogonal morphogens are small quorum-sensing molecules which can activate gene expression by binding a promoter with high specificity, to activate downstream gene expression. These toolboxes are key to bridging the gap between abstract models and real-sender receiver circuits. Additionally, in Oliver Huidobro et al. 2022 we compiled a list of small molecule regulators that can diffuse and form gene circuits in *E. coli*, as well as their respective promoters and synthesis enzymes. These could be investigated in the future to use as additional morphogens to build more complex systems with more diffusible molecules.

Finally, Tica 2020 engineered the #1754 topology using the components from Meyer et al. 2019 and Du et al. 2020, to explore a Turing gene circuit for pattern formation (see Fig. 1.8 bottom). Due to the lack of specific biological parts such as an inhibitor quorum sensing molecule or a dual inhibitor/activator molecule, the biological implementation of #1754 resulted in a six-molecular species circuit instead of the original 3-node network.

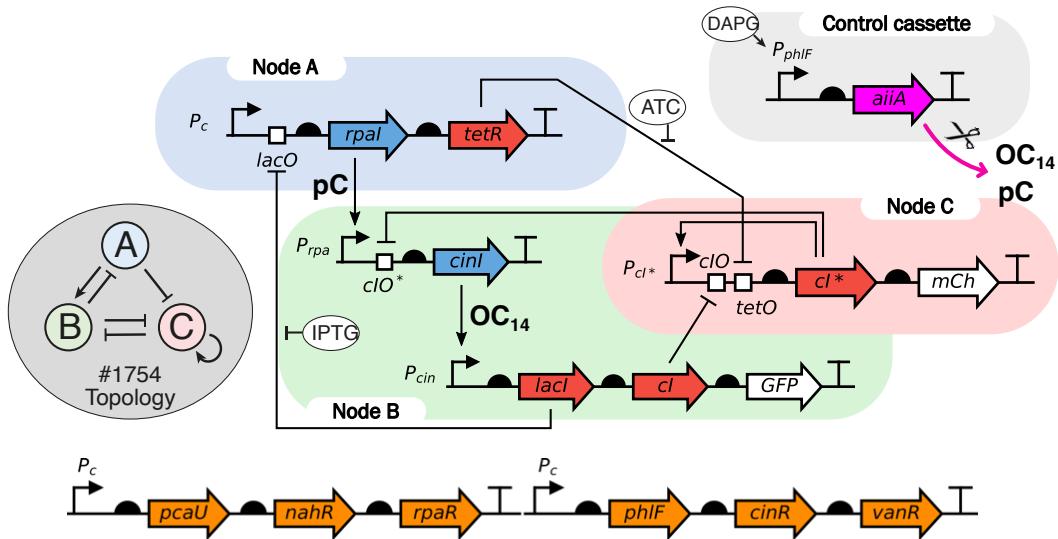


Figure 1.8: **Synthetic biology implementation of #1754 topology.** This synthetic circuit engineered by Dr. Jure Tica and Tong Zhu is a genetic abstraction of the #1754 topology in Scholes et al. 2019. This circuit is transformed so every *E. coli* cell in the biofilm has a copy inside. The original topology (left, grey inset) has only three nodes, while the synthetic circuit (right) has six genes. The six-gene circuit architecture, shown in standard notation, can be clustered into the three original nodes as seen by the blue, green and red bubbles. Diffusor synthesis enzymes, in blue, produce quorum sensing molecules *pC* and *OC₁₄*. Non-diffusible transcription factors (in red), also called repressor proteins, are *lacI*, *cl*, *cl** and *tetR*. Fluorescent proteins GFP and mCherry are in white. The circuit can be regulated by small molecules aTc, IPTG and DAPG shown in white bubbles. DAPG activates the control cassette and produces regulated degradation of small quorum sensing molecules. The bottom cassette (orange), contains the necessary regulators: *rpaR* is the *pC* receptor (Node A diffusor), *cinR* is the *OC₁₄* receptor (Node B diffusor), and *phlF* is the DAPG receptor (used to tune inducible diffusor degradation). The other three receptors *pcaU*, *nahR* and *vanR* are for the protocatechuate, salicylate and vanillin inducers, which are not used in the present circuit, but were tested in other versions of the system. These three systems can be used to introduce additional regulatory components to the system.

In this thesis, theoretical work based on this synthetic gene circuit was carried out to understand how to tune parameters experimentally and what spatial setup to use to allow pattern formation in bacterial biofilms. Additionally, once the setup is determined, the circuit is explored to predict which patterns might appear and the relevant wavelengths to look for. Finally, once patterns appear in the biofilm, the mechanisms behind these patterns are studied.

Having a genetic synthetic network that produces patterns with a predictive model would be extremely impactful on various levels. Firstly, the model would allow us to understand the patterning mechanisms behind the synthetic system and therefore shed some light on developmental biology. Currently, our understanding from top-down approaches is based on arbitrarily chosen models that match well the

experiments, but that are not directly informed on well-known and characterised gene circuits. Model selection from those experiments is practically impossible as it has been proven that different models can generate the same 1D or 2D Turing pattern (Woolley et al. 2021). In this bottom-up approach, the model used is derived from a known synthetic gene circuit that has been artificially introduced into cells. This process constrains the model to provide more insightful results. If this network can produce self-assembling periodic stationary patterns both *in-silico* and *in-vitro*, it would prove that a known reaction-diffusion system supported by genetic interactions can produce Turing patterns, therefore validating this mechanism. Secondly, having a model would allow us to inform the experimental design and how to fine-tune the system to obtain more robust desired patterns or different shapes for tissue engineering in biotechnology applications.

1.5 Thesis overview

This thesis explores Turing patterning within synthetic biology and tissue engineering, employing a modelling approach. The novelty of this thesis lies in its modelling aspects, although it also includes some experimental results.

The first chapter adopts a theoretical perspective, delving into the relationship between linear stability analysis and numerical solutions. Key aspects include estimating wavelength, convergence time, and, in some cases, predicting pattern shapes from linear stability analysis. It also examines various instabilities capable of producing stationary spatial patterns beyond classical Turing instabilities. This theoretical framework sets the stage for addressing realistic biological phenomena in pattern formation, such as varying boundary conditions and growth, crucial for understanding and engineering synthetic gene circuits for pattern formation.

The second chapter aims to develop a predictive model for the #1754 topology, informed by the synthetic gene circuit outlined in Tica 2020. This PDE model is based on the interactions between genetic components and the diffusion of quorum-sensing molecules. This model is used to explore the system's parameter space to identify regions prone to robust pattern formation and strategies for experimental tuning of the gene circuit. Additionally, the model is parameterized using liquid culture data to determine its position within the parameter space.

The final chapter focuses on the experimental and modelling aspects of synthetic biofilms and their emergent patterns. It presents initial experiments on small colonies forming periodic rings. A PDE solver, combined with a cellular automaton for modelling bacterial colony growth, is then introduced, replicating all observed patterns, including spots, wedges, and rings produced by collaborators. The chapter concludes with a detailed examination of the model's predictability against experimental perturbations, emphasizing the analysis of irregular growth, boundary effects, and node deletions.

In conclusion, this thesis significantly contributes to the fields of biology and mathematical biology, particularly in the realm of synthetic patterning. By employing a predictive model and optimizing robustness, this work demonstrates the ability to guide experimental design for pattern formation as well as replicating the complex patterns obtained. However, despite these advancements, the current level of robustness achieved remains relatively low, highlighting the need for further research. Future work should focus on deepening our understanding of the underlying mechanisms and refining techniques to enhance the robustness and reliability of synthetic patterning processes. This will be crucial in realizing the full potential of this technology in biomedical and industrial applications.

CHAPTER 2

Beyond classical Turing patterns

2.1 Synopsis

The Turing instability is a well-known mechanism for pattern formation. Turing instabilities are defined and studied using linear stability analysis (LSA), but they can also be further explored with numerical methods. LSA predicts whether a pattern will occur through the dispersion relation, where the real part of eigenvalues determines the system's stability as a function of a wavenumber parameter. Although fast, this method provides limited insight into the emerging pattern, as it relies on linearization around the steady state. On the other hand, numerical methods, while more computationally intensive, offer detailed results of the developing pattern, including the concentration of species over time and space. Numerical results also enable visualization of the resulting Turing pattern shape, which in 2D can consist of spots, stripes, or labyrinths.

To effectively conduct a high-throughput study exploring the #1754 synthetic Turing gene circuit built in Tica 2020, it's essential to leverage LSA. LSA offers a much quicker alternative compared to numerical methods for understanding the patterning capabilities of the experimental circuit. However, LSA, which relies on linearization around the steady state, sometimes fails to predict pattern formation accurately. Understanding what information can be obtained from LSA before resorting to numerical methods, and when LSA predictions break will enable us to efficiently gather extensive insights into both the patterning potential and the robustness of the system.

In this chapter, we first study the relationship between the analytical dispersion relation and the numerical results in Turing patterns. This includes understanding

how to predict wavelength, convergence time and even pattern shape from the dispersion relation. This is important to obtain information from the emergent pattern without having to solve the partial differential equation (PDE) numerically. Then we explore how LSA might not always be a good predictor of pattern formation. For example, how multistability can break LSA predictions and how other types of dispersion relation profiles or instabilities which are not classical Turing can generate stationary spatial patterns or other types of temporal-spatial patterns. Finally, we explore numerically how these patterns behave when realistic biological features present in our experimental setup are introduced, such as absorbing boundaries or growth.

2.2 Linear stability analysis and the dispersion relation

2.2.1 Linear stability analysis for a reaction-diffusion system

The following section describes how LSA is carried out for the steady states in a reaction-diffusion system. This analysis aims to find out if the steady state exhibits a Turing instability which is also called a diffusion-driven instability. When it does, the system is capable of forming spatial patterns at least transiently. As the name describes, diffusion-driven instabilities arise in these systems when a homogeneous steady state is stable to small perturbations in the absence of diffusion, and becomes unstable in the presence of diffusion (Glendinning 1994; Murray 2002). To check for Turing instabilities, the stability of the equilibrium state will be studied without and with diffusion. The method of LSA will be explained for the two morphogen reaction-diffusion system, shown below:

$$\frac{\partial A}{\partial t} = f_A(A, I) + D_A \nabla^2 A \quad (2.1a)$$

$$\frac{\partial I}{\partial t} = f_I(A, I) + D_I \nabla^2 I \quad (2.1b)$$

where $f_{A,I}$ are the non-linear production terms and $D_{A,I}$ are the diffusion constants of the two morphogens.

Stability of steady state without diffusion

To study the stability around the steady state without diffusion, only the reaction terms of Eqs.2.1 are used:

$$\frac{\partial A}{\partial t} = f_A(A, I) \quad (2.2a)$$

$$\frac{\partial I}{\partial t} = f_I(A, I) \quad (2.2b)$$

There is no space dependence here as diffusion terms have been removed. Therefore, there is only time dependence. The steady states are defined as A^* and I^* ,

which satisfy the condition:

$$f_A(A^*, I^*) = 0, \quad f_I(A^*, I^*) = 0 \quad (2.3)$$

In this section, X is a vector with the two morphogens $X = [A, I]$. LSA is carried by adding an infinitesimally small perturbation δX to the steady state X^* , and studying if the perturbation decays (stable steady state) or grows (unstable steady state) over time. The perturbation needs to be almost insignificant as Taylor expansion is carried out to linearise the system around the steady state. Therefore, the morphogen concentration can be expressed as:

$$A(t) = A^* + \delta A(t), \quad |\delta A| \ll A^* \quad (2.4a)$$

$$I(t) = I^* + \delta I(t), \quad |\delta I| \ll I^* \quad (2.4b)$$

The differential Eqs. 2.2 are evaluated at steady state, using Eqs 2.4:

$$\frac{\partial A}{\partial t} = \frac{\partial[A^* + \delta A(t)]}{\partial t} = f_A(A^* + \delta A(t), I^* + \delta I(t)) = \frac{\partial \delta A}{\partial t} \quad (2.5a)$$

$$\frac{\partial I}{\partial t} = \frac{\partial[I^* + \delta I(t)]}{\partial t} = f_I(A^* + \delta A(t), I^* + \delta I(t)) = \frac{\partial \delta I}{\partial t} \quad (2.5b)$$

As previously mentioned, the non-linear system will be linearised around the steady state using Taylor expansion. This is done to have a simpler set of equations, that represent the system around the steady state, as seen below:

$$\begin{aligned} f(A^* + \delta A, I^* + \delta I) &= f(A^*, I^*) + \frac{\partial f(A^*, I^*)}{\partial A} \delta A + \frac{\partial f(A^*, I^*)}{\partial I} \delta I + \dots \\ &\quad + \frac{1}{n!} \frac{\partial^n f(A^*, I^*)}{\partial A^n} \delta A^n + \frac{1}{n!} \frac{\partial^n f(A^*, I^*)}{\partial I^n} \delta I^n \end{aligned} \quad (2.6)$$

If δA and δI are small enough, higher order terms can be ignored as $\delta(A, I)^n$ becomes infinitesimally small. Furthermore, it can be assumed that $f(A^*, I^*) = 0$, therefore the following expression is obtained, where f corresponds to either f_A or f_I :

$$f(A^* + \delta A, I^* + \delta I) = \frac{\partial f(A^*, I^*)}{\partial A} \delta A + \frac{\partial f(A^*, I^*)}{\partial I} \delta I \quad (2.7)$$

Finally, because $\frac{\partial X}{\partial t} = \frac{\partial \delta X}{\partial t}$ at steady state (Eqs. 2.5), the change in perturbation, meaning decay or growth, can be expressed as:

$$\frac{\partial \delta A}{\partial t} = \frac{\partial f_A(A^*, I^*)}{\partial A} \delta A + \frac{\partial f_A(A^*, I^*)}{\partial I} \delta I \quad (2.8a)$$

$$\frac{\partial \delta I}{\partial t} = \frac{\partial f_I(A^*, I^*)}{\partial A} \delta A + \frac{\partial f_I(A^*, I^*)}{\partial I} \delta I \quad (2.8b)$$

The general solution can be expressed as an exponential, where σ determines the growth rate of the perturbation, and whether it grows (unstable steady state) or decays (stable steady state):

$$\delta A = A_0 e^{\sigma t} \quad (2.9a)$$

$$\delta I = I_0 e^{\sigma t} \quad (2.9b)$$

If Eq. 2.9 are introduced into Eq. 2.8, and the solution divided by $e^{\sigma t}$ on both sides, the following is obtained:

$$\sigma \delta A_0 = \frac{\partial f_A(A, I)}{\partial A} \delta A_0 + \frac{\partial f_A(A, I)}{\partial I} \delta I_0 \quad (2.10a)$$

$$\sigma \delta I_0 = \frac{\partial f_I(A, I)}{\partial A} \delta A_0 + \frac{\partial f_I(A, I)}{\partial I} \delta I_0 \quad (2.10b)$$

If the Jacobian J of this system is

$$J = \begin{bmatrix} \frac{\partial f_A}{\partial A} & \frac{\partial f_A}{\partial I} \\ \frac{\partial f_I}{\partial A} & \frac{\partial f_I}{\partial I} \end{bmatrix} \quad (2.11)$$

This can be represented as an eigenvalue-eigenvector problem using the jacobian J , the eigenvalue σ and the eigenvector $\delta X_0 = [\delta A_0, \delta I_0]$:

$$\sigma \cdot \delta X_0 = J \cdot \delta X_0 \quad (2.12)$$

Finally, σ can be obtained by solving the characteristic polynomial:

$$p(\sigma) = \det[J - \sigma I] = 0 \quad (2.13)$$

For a system with two variables, there are two eigenvalues σ_1 and σ_2 . The real part of σ corresponds to the growth rate of the perturbation. $\text{Im}(\sigma)$ corresponds to the oscillations. If all $\text{Re}(\sigma) < 0$, the steady state will be stable to perturbations, which means the system will go back to the steady state after a small perturbation is applied. If at least one $\text{Re}(\sigma) > 0$, the steady state will be unstable to perturbations, and the system will go away from the equilibrium point after being slightly perturbed. The first requirement for a Turing instability is that the system without diffusion is stable, therefore all eigenvalues must be negative.

Stability of steady state with diffusion

Once the stability of the steady state has been analysed in the absence of diffusion, the effect of diffusion on the same system will be studied. The diffusion will be introduced through the partial derivative term $D_X \nabla^2 X$.

Definition of a function in a finite system with no-flux boundary conditions: A function $X(x)$, which represents the concentration of molecule X in space x, is defined in a finite-domain $[0, L]$ with no-flux boundary conditions. No-flux boundary conditions imply that the derivative of the function at the boundaries is zero:

$$\left. \frac{\partial X}{\partial x} \right|_{x=0} = \left. \frac{\partial X}{\partial x} \right|_{x=L} = 0 \quad (2.14)$$

A general Fourier series representation of $X(x)$ under these conditions can be:

$$X(x) = \sum_{n=0}^{\infty} \left(A_n \cos \left(\frac{n\pi x}{L} \right) + B_n \sin \left(\frac{n\pi x}{L} \right) \right) \quad (2.15)$$

However, because the derivative at the boundary is zero, the cosine series is used instead. This is because the derivative of $\cos(0)$ or $\cos(n\pi)$ is zero. Therefore, the Fourier cosine series used is

$$X(x) = \sum_{n=0}^{\infty} C_n \cos \left(\frac{n\pi x}{L} \right) \quad (2.16)$$

This way, if the function $X(x)$ is expressed in terms of cosines (see Fig. 2.1), the derivatives of this function for $x=0$ and $x=L$ are zero:

$$\frac{\partial (\cos(0))}{\partial x} = 0; \quad \frac{\partial (\cos(\frac{n\pi L}{L}))}{\partial x} = 0 \quad (2.17)$$

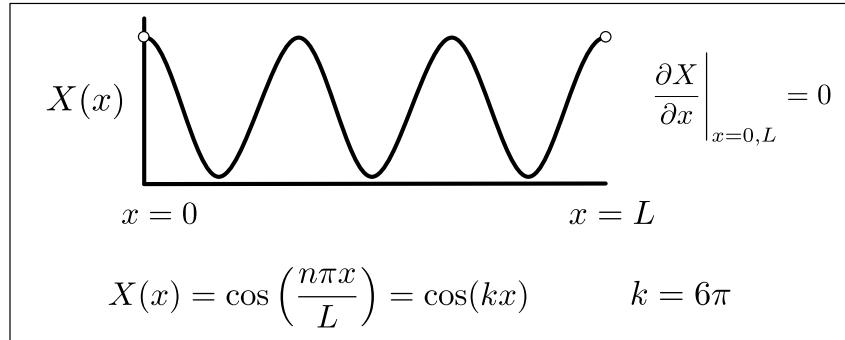


Figure 2.1: No-flux boundary conditions represented with cosine wave

Derivative of function in space: The Laplacian Operator ∇^2 is applied to $X(x)$ to represent the diffusion terms

$$\nabla^2 X(x) = \frac{\partial^2}{\partial x^2} \left(\sum_{n=0}^{\infty} C_n \cos \left(\frac{n\pi x}{L} \right) \right) = - \sum_{n=0}^{\infty} C_n \left(\frac{n\pi}{L} \right)^2 \cos \left(\frac{n\pi x}{L} \right) \quad (2.18)$$

If we replace the original cosine series expressing $X(x)$ (Eq. 2.16) into this equation, we obtain

$$\nabla^2 X(x) = -\left(\frac{n\pi}{L}\right)^2 X(x) \quad (2.19)$$

Wavenumber definition: The wavenumber k is the spatial frequency of the wave, measured in cycles or radians per unit distance. In this case, the wave $X(x)$ is expressed in cosine form. As previously explained, to satisfy the no-flux boundary conditions, the derivative of this cosine must be zero at the boundaries $x = [0, L]$. Therefore, k_n must be

$$k_n = \frac{n\pi}{L} \quad (2.20)$$

Once k_n is defined for our boundary conditions, the expression for the Laplacian becomes

$$\nabla^2 X(x) = -k_n^2 X(x) \quad (2.21)$$

Linear stability analysis of steady state with diffusion effects: Now that the PDE diffusion term has been reduced to an ODE term, and the zero-flux boundary conditions introduced, the stability of the steady state when introducing diffusion can be studied. As shown in Eqs. 2.21, the diffusion term $\nabla^2 X(x)$ can be expressed as $-k_n^2 X(x)$, therefore the linearised reaction-diffusion PDE equations can be written as

$$\frac{\partial A}{\partial t} = f_A(A^* + \delta A, I^* + \delta I) - D_A k_n^2 A \quad (2.22a)$$

$$\frac{\partial I}{\partial t} = f_I(A^* + \delta A, I^* + \delta I) - D_I k_n^2 I \quad (2.22b)$$

where $k_n = \frac{n\pi}{L} \forall n \in \{0, \mathbb{N}\}$.

In the previous section, LSA was carried around the steady state with a perturbation δX , without including diffusion. This resulted in the expression shown in Eq. 2.8, which describes the linearized reaction terms around the steady state. By adding diffusion in the Fourier domain from Eq. 2.21 we obtain the following expression for the linearized reaction-diffusion system, where A and I are now a function of k_n and t :

$$\frac{\partial \delta A}{\partial t} = \frac{\partial f_A(A^*, I^*)}{\partial A} \delta A + \frac{\partial f_A(A^*, I^*)}{\partial I} \delta I - D_A k_n^2 \delta A \quad (2.23a)$$

$$\frac{\partial \delta I}{\partial t} = \frac{\partial f_I(A^*, I^*)}{\partial A} \delta A + \frac{\partial f_I(A^*, I^*)}{\partial I} \delta I - D_I k_n^2 \delta I \quad (2.23b)$$

We wish to find a general solution of the form:

$$\delta A = A_0 e^{\sigma t} \cdot e^{ik_n x} \quad (2.24a)$$

$$\delta I = I_0 e^{\sigma t} \cdot e^{ik_n x} \quad (2.24b)$$

where $X_0 e^{\sigma t}$ represents the amplitude of the perturbations and $e^{ik_n x}$ represents the spatial oscillations (with k_n as the wavenumber). In this case, we are interested on the growth or decay of the perturbations over time. Therefore, to observe the stability of the steady state, we will study the amplitude term ($X_0 e^{\sigma t}$):

- If $\sigma > 0$: perturbation (δX) grows making $\frac{\partial \delta X}{\partial t} > 0$. Therefore, the steady state is unstable with diffusion.
- If $\sigma < 0$: perturbation (δX) decays making $\frac{\partial \delta X}{\partial t} < 0$. Therefore, the steady state is stable with diffusion.

If Eqs. 2.24 are substituted into Eqs. 2.23, the following equations are obtained:

$$\sigma \delta A_0 = \frac{\partial f_A(A^*, I^*)}{\partial A} \delta A_0 + \frac{\partial f_A(A^*, I^*)}{\partial I} \delta I_0 - D_A k_n^2 \delta A_0 \quad (2.25a)$$

$$\sigma \delta I_0 = \frac{\partial f_I(A^*, I^*)}{\partial A} \delta A_0 + \frac{\partial f_I(A^*, I^*)}{\partial I} \delta I_0 - D_I k_n^2 \delta A \quad (2.25b)$$

If the Jacobian J of this system is

$$J = \begin{bmatrix} \frac{\partial f_A}{\partial A} - D_A k_n^2 & \frac{\partial f_A}{\partial I} \\ \frac{\partial f_I}{\partial A} & \frac{\partial f_I}{\partial I} - D_I k_n^2 \end{bmatrix} \quad (2.26)$$

these pair of equations can again be treated as an eigenvalue-eigenvector problem written in the following way:

$$\sigma \cdot \delta X_0 = J \cdot \delta X_0 \quad (2.27)$$

where σ can be obtained by solving the characteristic polynomial:

$$p(\sigma) = \det[J - \sigma I] = 0 \quad (2.28)$$

In summary, to understand if perturbations around the steady state decay or grow when diffusion is introduced, Eq. 2.28 must be solved to obtain σ . This characteristic polynomial is solved for a range of k 's, being $k_n = \frac{n\pi}{L} \forall n \in \{0, \mathbb{N}\}$. The sign of σ is studied for every k_n , resulting in the dispersion relation (Fig. 2.2). If $\sigma > 0$, perturbations grow around the steady state and the system is unstable. On the contrary, if $\sigma < 0$, the system is stable and perturbations decay around the steady state. More detail on LSA can be found in Murray 2002 or Glendinning 1994.

Linear stability analysis implementation

LSA is carried out with and without diffusion for all steady states of the system found using the Newton-Raphson method (see Section 6.1). The analysis is done for all $k_n = \frac{n\pi}{L} \forall \{n \in \mathbb{N} : n \leq 5000\}$, meaning 5000 k's are sampled using linear stability analysis. L is defined as 100mm which is based on the maximum length of our experimental system, described in Chapter 4. LSA for a single steady state takes approximately approximately 0.5s.

2.2.2 Model definition and database creation

The system studied in this chapter consists of a 2-node Turing topology system with non-linear Hill terms. The activator A activates itself and I, while the inhibitor I inhibits the activator A as seen in Fig. 1.4A. However, instead of using the linear terms used in the original Turing system Fig. 1.4B (Turing 1952), we use Hill terms which represent genetic regulations. More information on how these Hill terms represent gene expression regulations can be found in Section 3.1.1.

$$\frac{\partial[A]}{\partial t} = b_A + V_A \cdot \frac{1}{1 + \left(\frac{KA}{[A]}\right)^{n_A}} \cdot \frac{1}{1 + \left(\frac{[I]}{K_I}\right)^{n_I}} - \mu_A \cdot [A] + D_A \nabla^2[A] \quad (2.29a)$$

$$\frac{\partial[I]}{\partial t} = b_I + V_I \cdot \frac{1}{1 + \left(\frac{KA}{[A]}\right)^{n_A}} - \mu_I \cdot [I] + D_I \nabla^2[I] \quad (2.29b)$$

To study this system, the parameter space is scanned using LSA and numerical methods. The parameter space is sampled using Latin hypercube sampling (LHS) (see Fig. 6.1A) drawn from loguniform distributions (see Fig. 6.1C). The LHS method is designed to cover the parameter space more efficiently with fewer samples than grid or uniform sampling (Chrisman 2014; Iman 2014). More information on sampling and distributions used can be found in the Methods Section 6.2. Ranges and values used for these distributions are shown in the Appendix Tables A.1 and A.2.

All results are stored in a PostgreSQL database created from scratch for this project. This database includes interconnected tables for model parameters, numerical parameters, analytical results, numerical solutions and pattern classification outputs. More details on the database schema can be seen in the Section 6.3. The creation of this PostgreSQL database was necessary to explore large parameter spaces and compare between analytical and numerical results in an efficient way.

2.2.3 Diffusion-driven instabilities in the dispersion relation

Classical Turing patterns, also called diffusion-driven instability (DDI) or Turing instabilities, can be detected using LSA as described above. In linear stability analysis, the dispersion relation is studied, which shows the dependency of the eigenvalues

σ on the wavenumber k_n as seen in Eq. 2.27. For a Turing instability to occur, at $k_0 = 0$, $Re(\sigma)$ must be negative, meaning the system is stable without diffusion. As diffusion is introduced and $k_n > 0$, the system becomes unstable with $Re(\sigma) > 0$. For $k_n \rightarrow \infty$, the system must be stable again and the eigenvalue negative. This is because otherwise, a Turing II instability will occur where the dominant mode has an infinitesimally large wavenumber meaning an infinitesimally small wavenumber. The relationship between wavenumber (k_n) and wavelength (λ) is

$$\lambda_n = \frac{2\pi}{k_n} \quad (2.30)$$

An example of a classical Turing instability with a positive dispersion peak is shown in Fig. 1.4B, along with the corresponding stationary spatial pattern. Alternatively, Fig. 2.2A shows a spatially homogeneous solution which results from a stable system that does not exhibit a diffusion-driven instability. These one-dimensional (1D) numerical results were produced using the Crank-Nicolson (CN) numerical scheme described in 6.6.1, programmed using *numpy* in Python from first math principles. The boundary conditions used here are Neumann boundary conditions where the derivative at the boundary is zero ($\frac{\partial U}{\partial x} = 0$) for both molecular species, A and I.

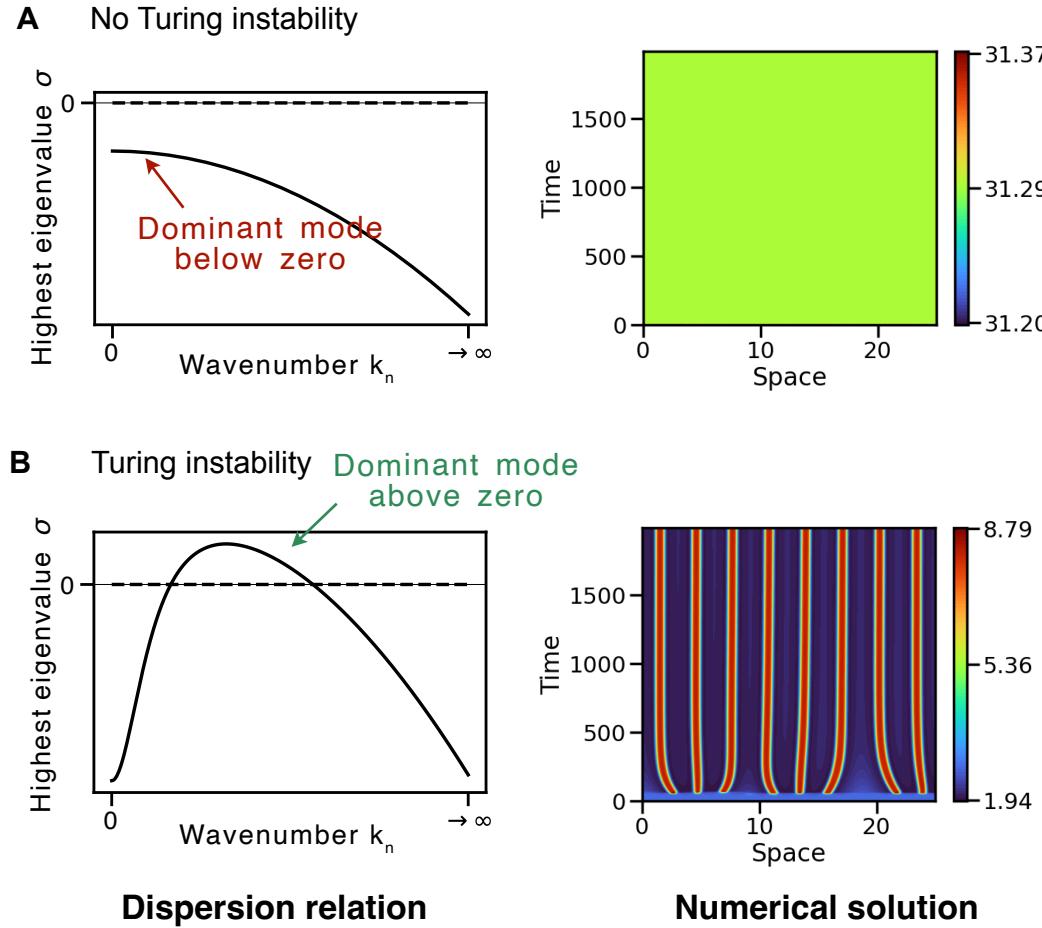


Figure 2.2: The dispersion relation of reaction-diffusion systems. **(A)** Dispersion relation of a system without Turing instability (left). Eigenvalues are computed for every wavenumber k_n . The dominant mode is stable, having an eigenvalue below zero. The numerical solution of the system for molecule A without Turing instability (right), shows homogeneity in space (x-axis) and time (y-axis). **(B)** Dispersion relation of a system with Turing instability (left). The dominant mode is unstable, having an eigenvalue above zero. The numerical solution of the system for molecule A (right), shows a periodic in space (x-axis) which is stationary in time (y-axis). The colorbar reflects molecular species concentration.

2.3 Correlating linear stability analysis and numerical solutions

2.3.1 Inferring wavelength and convergence time from dispersion relation

Although LSA cannot be used to predict the final pattern, some information can be obtained from the dispersion relation. Using the two-node circuit defined in Eq. 2.29, 500 parameter sets were simulated with Turing instabilities as the one seen in Fig. 2.2B. Subsequently, the numerical convergence time and wavelength for these patterns were measured using the algorithms described in Section 6.4. The

predicted LSA wavelength λ was obtained from the wavelength-wavenumber relationship (Eq. 2.30) using the wavenumber k_n which has the highest eigenvalue.

A linear relationship was found between the wavelength in the numerical pattern and the wavelength inferred from the dominant mode of the dispersion relation as seen in Fig. 2.3A. However, the slope of the regression fitted to the data is 1.46, meaning the numerical patterns obtained have a wavelength 1.46 times higher than the predicted LSA wavelength.

In terms of the convergence time, a non-linear correlation is observed between the highest eigenvalue (σ) and the time for convergence. Although some values do not correlate, we can overall state that systems with high eigenvalues converge faster than those with lower eigenvalues (Fig. 2.3B). However, there is a lot of noise in low eigenvalues or low convergence times.

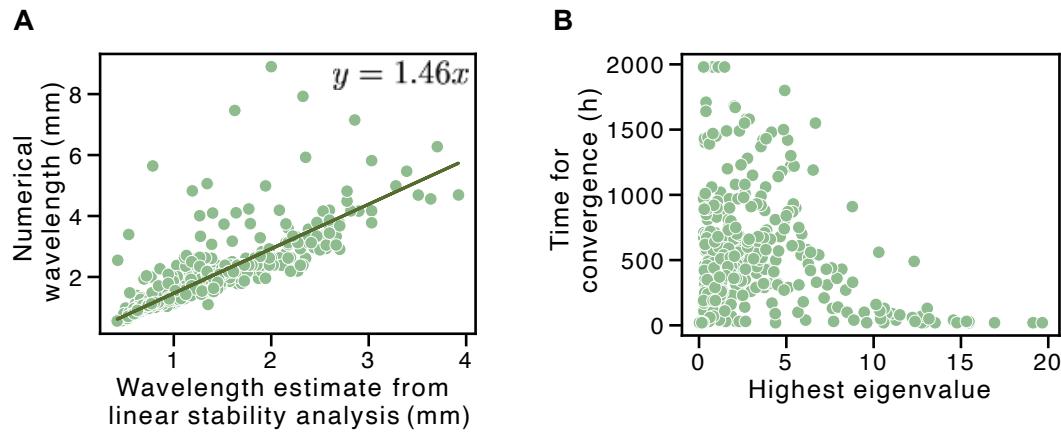


Figure 2.3: Relationship between numerical wavelength or convergence, and dispersion relation features. (A) Numerical wavelength with respect to wavelength predicted from linear stability analysis. Each sample was studied analytically and numerically (light green dots). A linear regression model was applied to produce a positive correlation with slope 1.46 (dark green line). (B) Numerical convergence time with respect to the highest eigenvalue from the dispersion relation shows a noisy decreasing function. Each sample computed numerically and analytically is a light green dot.

These relationships are extremely important if we wish to understand characteristics of our system such as wavelength and convergence time without simulating the system numerically. Especially if scanning through high dimensional spaces, such methods can be extremely informative to get an insight into patterns in this space. These relationships can be used before using numerical methods to choose relevant numerical parameters such as length of space and time of our simulation as well as dx and dt .

2.3.2 Dispersion to pattern shape

To further understand the information encoded in the dispersion relation, we focused on the dispersion peak height. The approach used consisted in optimising the dispersion peak height starting from an initial Turing pattern. This way we can understand what happens to a specific pattern in parameter space when the dispersion peak height increases. The optimisation is carried out using a Markov-Chain Monte Carlo combined with the Metropolis algorithm, where the optimised function is the height of the dispersion peak (i.e. the highest value of the highest eigenvalue). For more information on the optimisation algorithm, see Section 6.5.

For this specific optimisation work, we used a six-equation model describing the synthetic gene circuit presented in Chapter 3. The optimisation path can be observed in Fig. 2.4A where the dispersion peak was optimised from 0.22 to 4.36 after 50000 iterations. This shows how the algorithm can avoid getting stuck in local maxima to go towards a global maxima. There is no indication the algorithm is close to reaching a global maximum as the rate of improvement is still high after many iterations. The resulting numerical patterns from the optimisation are then simulated and the convergence time is measured as we did for Fig. 2.3B. In Fig. 2.4B we can see a clearer correlation of how the dispersion peak height determines the convergence time which could be modelled with an exponential decay function where x is dispersion peak height and $f(x)$ is convergence time as

$$f(x) = ae^{-bx} + c \quad (2.31)$$

This optimisation was carried out for three different initial Turing parameter sets. In Fig. 2.4C we can observe the pattern for the initial Turing pattern and the Turing pattern after optimisation (with the highest dispersion peak). While the lower dispersion peaks present labyrinthian type of patterns, the patterns post-optimisation present dot-like morphologies. This suggests that a higher dispersion peak can lead to a preference for spots over labyrinths.

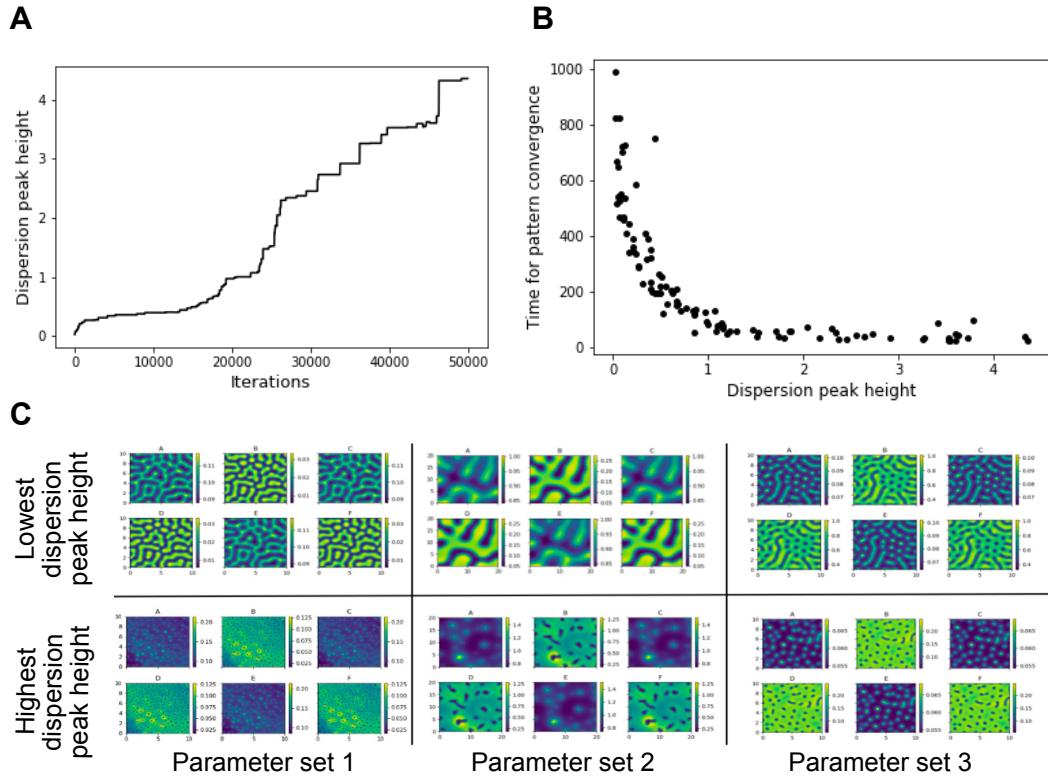


Figure 2.4: Effects of dispersion peak optimisation on convergence time and pattern shape. (A) Path of the dispersion peak optimisation using the Markov-Chain Monte Carlo with Metropolis algorithm. (B) Numerical convergence time with respect to highest eigenvalue from the dispersion relation. Each sample computed numerically and analytically is a black dot. (C) Effect of dispersion peak height on pattern shape. The top figures are non-optimised parameter sets (lowest dispersion peak height). The bottom figures represent parameter sets originating from the top parameter sets, that have been optimised for dispersion peak height (highest dispersion peak height). The six panels of every parameter set represent the 6 molecular species of the model.

2.4 Breaking linear stability analysis predictions

Although Turing instabilities are an indicator of pattern formation, these instabilities are neither sufficient nor necessary for stationary periodic patterns to occur. Current state-of-the-art literature analyses robustness for Turing pattern formation using LSA to determine the Turing patterning parameter space (Marcon et al. 2016; Scholes et al. 2019; Zheng et al. 2016). With this type of analysis, you have a binary output which determines whether a parameter leads to a classical Turing instability or not as seen in Fig. 2.5. However, the assumptions depicted in this confusion matrix might not always hold true. This is because the results from LSA give insights only into the local dynamics around a steady state. Once the system moves away from the steady state and non-linearities are introduced, the system might behave differently and the local pattern dynamics might not be maintained.

		Numerical solution	
		Pattern	No Pattern
Linear Stability Analysis	Turing I	1	0
	Else	0	1

Figure 2.5: **Binary confusion matrix linking linear stability analysis and patterning.**

2.4.1 Multistability in Turing

In this section, we demonstrate how linear stability analysis is not sufficient or necessary to predict TPs in multi-stable systems. In particular, we study in detail the dynamical behaviour of multi-stable systems during pattern formation, which will lead to the creation or breaking of the pattern. The motivation behind this arises from the high degree of multistability exhibited by biological systems, where cell-fate decisions have to be taken within this landscape (Huang and Ingber 2000; Moris et al. 2016). Multistability is especially common in systems with non-linearities and feedback loops, as the ones in biology or in our models (Leite and Wang 2009; Pham et al. 2020).

Using the two-node non-linear Turing topology (Eq. 2.29), multi-stable solutions were found and studied to understand how the patterning dynamics get affected when multiple steady-state solutions are found. First, LSA is carried out on particular parameter sets to find multiple steady states of different stability (e.g. stable, unstable, Turing I, Turing I-Hopf). Then numerical simulations are computed where the initial condition is a random uniform distribution around a particular steady state. Different pattern outcomes result depending on where in the phase diagram the initial condition is. Following the classical hypothesis used in the Turing robustness literature, we would expect stable and unstable systems to not produce patterns and Turing to produce patterns. Here, we present various examples of how this hypothesis can break under multistability conditions.

Fig. 2.6 shows a case where diffusion-driven instability conditions are not required for Turing pattern formation. The unstable state, having a dispersion relation with a peak below zero (Fig. 2.6C), managed to get into a Turing pattern regime as it is attracted by the Turing steady state. It therefore produces a stationary pattern (Fig. 2.6B), even though its dispersion relation does not predict so. This trajectory is

depicted in the phase diagram (Fig. 2.6A) which shows the steady states along with the vector field to understand the potential trajectories of the system. The phase diagram does not fully capture the dynamics as it describes the system without diffusion, while the dispersion relation and the numerical solution do consider diffusion.

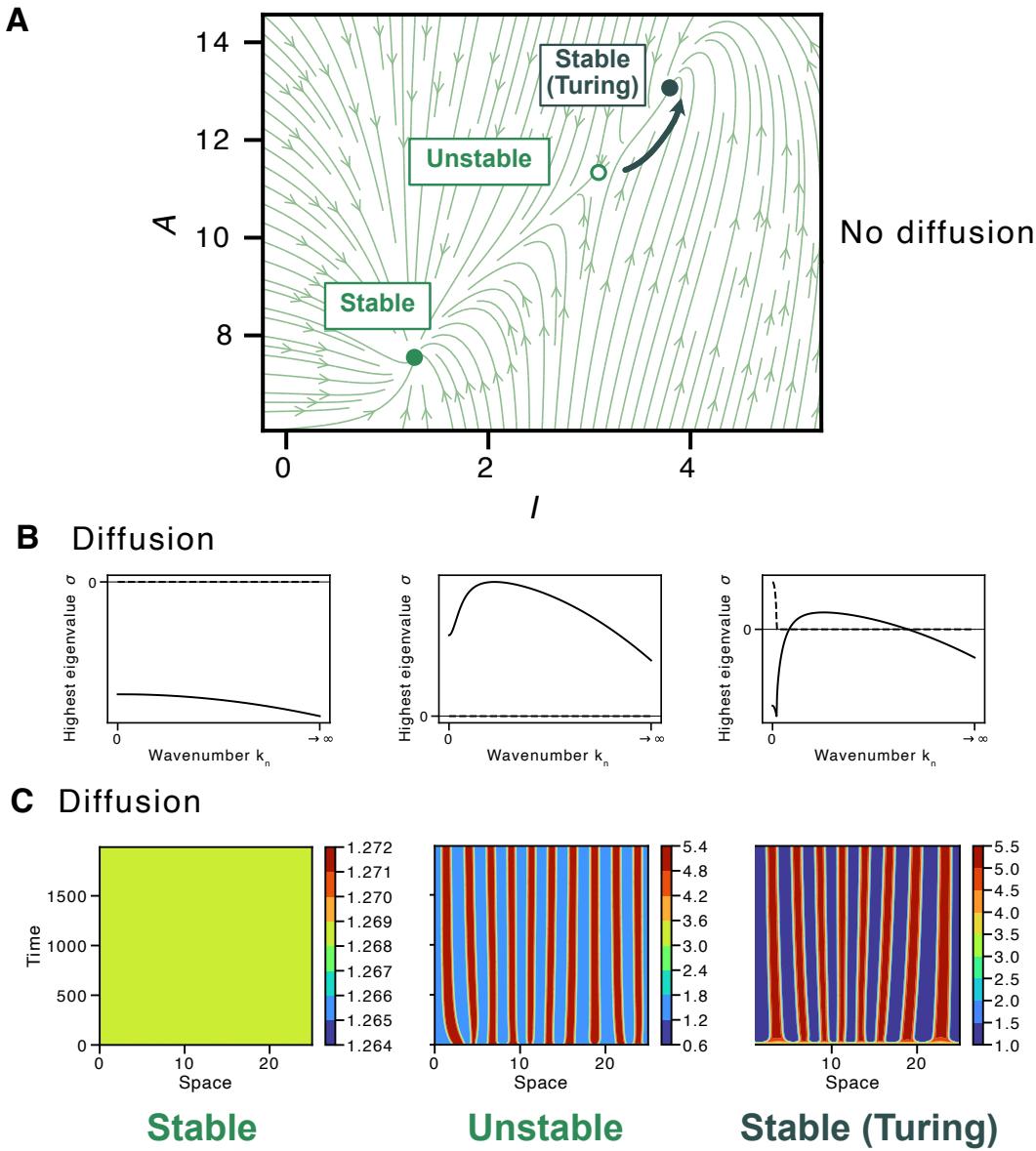


Figure 2.6: **Stationary Patterns in multistability.** (A) Phase diagram without diffusion illustrating three distinct steady states where the derivative is zero: stable, unstable, and stable (Turing). These steady states are represented within a parameter space defined by two axes: concentrations of A and I . The vector field, indicated by light green arrows, shows the direction of the derivatives of the system at various points in the parameter space. A hand-drawn trajectory is also shown (dark green arrow), demonstrating how the unstable state may evolve into the Turing state. (B) Dispersion relation showing each type of state. (C) Numerical solutions of the three steady states with diffusion, where the unstable state unexpectedly produces a Turing-like stationary pattern.

Then, we present a case where LSA incorrectly predicts stationary pattern formation. Fig. 2.7 shows an ephemeral or transient pattern that occurs in the unstable and Turing regimes. The TP initially develops in the vicinity of the Turing steady state. As the spatial heterogeneity is amplified and settles, it gets attracted by the stable steady state leading to the disruption of the pattern. This type of transient pattern behaviour has also been recently reported in Krause et al. 2023.

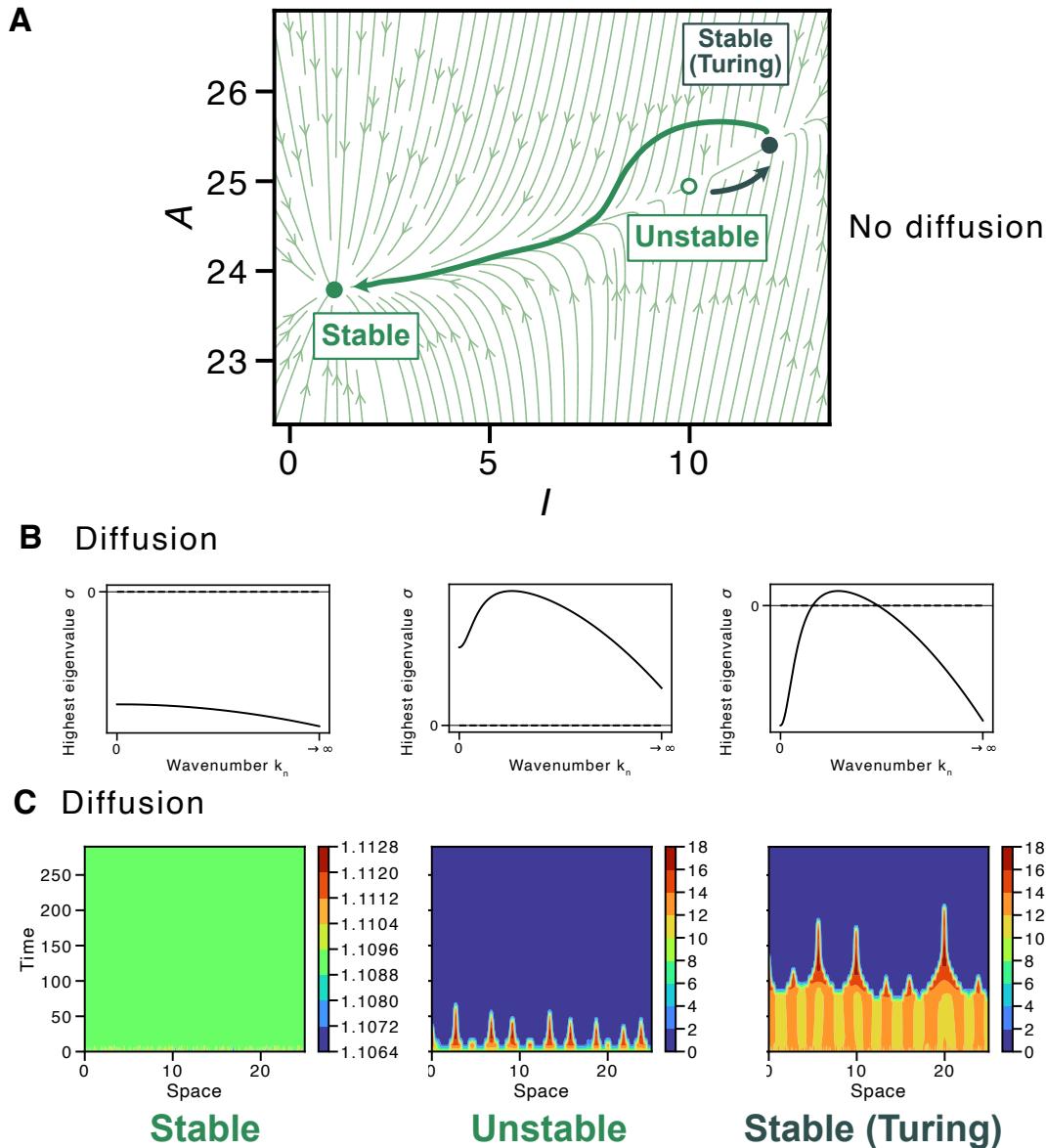


Figure 2.7: **Ephemeral patterns in multistability.** (A) Phase diagram without diffusion illustrating three distinct steady states where the derivative is zero. The hand-drawn trajectory (dark green arrow) shows an unstable state evolving into a stable (Turing), and a stable (Turing) evolving into a stable. (B) Dispersion relation showing each type of state. (C) Numerical solutions of three steady states with diffusion. Unstable and Turing produce temporary periodic stationary patterns, that then disappear and become spatially homogeneous solutions.

Other interesting examples can be found, for example, where an unstable state is surrounded by two Turing states, this unstable state will robustly lead to a Turing pattern (Fig. 2.8A). Additionally, in some cases, the unstable system settles into Turing, but the Turing system gets pulled by the stable attractor (Fig. 2.8B). Additionally, some systems even exhibit a transient pattern and the three solutions are homogeneous in time and space (Fig. 2.8C). In this case, it would be worth investigating earlier time points with more resolution, as a pattern might appear then. Interesting interactions similarly occur with multistability involving Turing I-Hopf solutions which will be mentioned in the following sections (Fig. 2.8D).

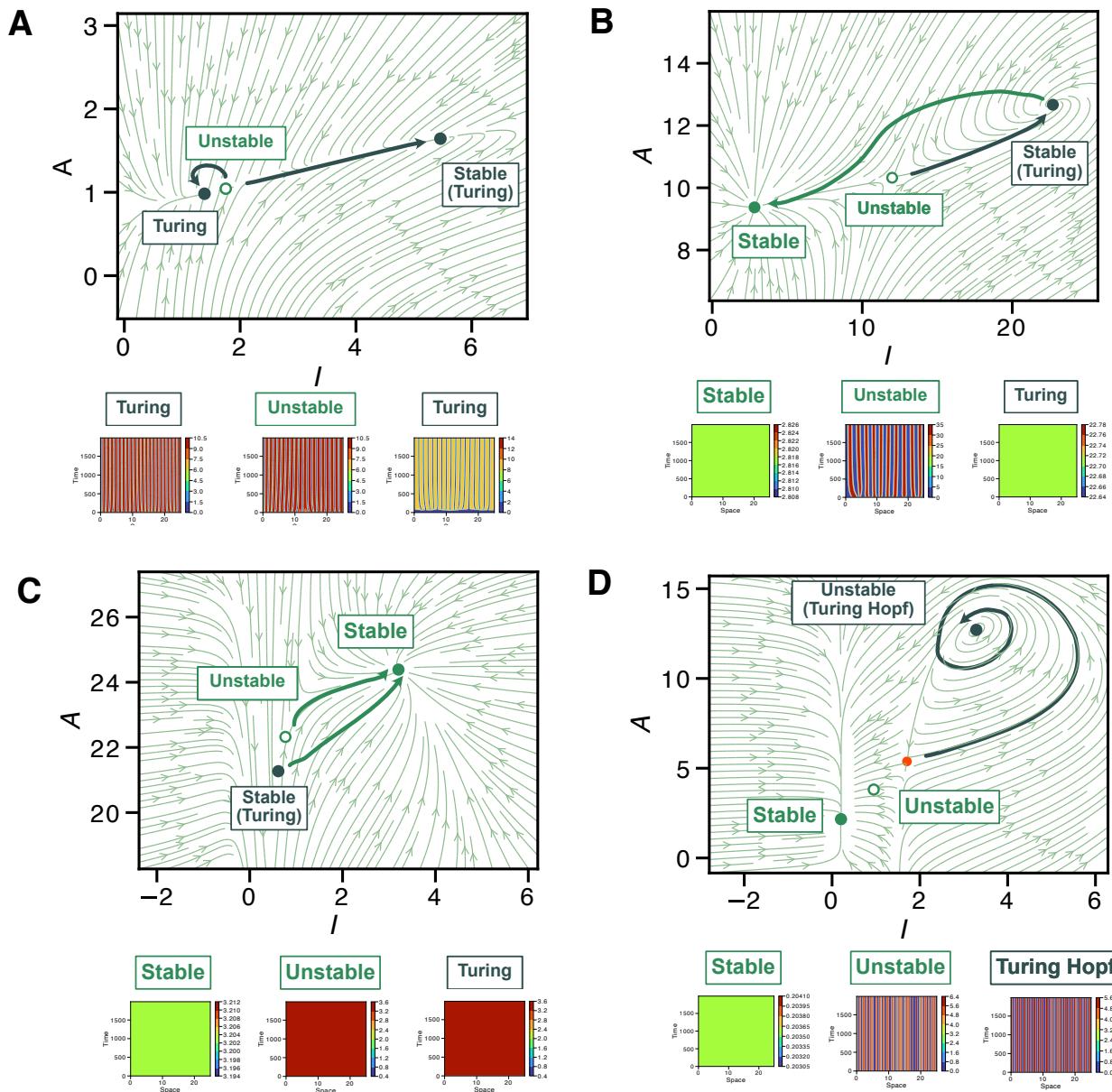


Figure 2.8: Other types of multistability dynamics. (A) Unstable surrounded by Turing robustly converges into Turing. (B) Unstable produces pattern, while Turing loses pattern. (C) Multistability disrupts all patterns. (D) Turing I-Hopf attracts unstable and generates pattern

2.4.2 Analytical to numerical: Other types of dispersion relations, and other types of patterns

Multi-stable systems are not the only case where the classical Turing instability fails to predict pattern formation. Other types of dispersion relations beyond classical Turing instabilities can produce stationary patterns and non-stationary regular patterns that might be of interest in developmental biology. This section aims to document what type of dispersion relations in mono-stable systems can be linked to what type of patterns, to gain insights into predicting pattern formation from linear stability analysis. We will first present a classification for the different types of dispersion relations. Subsequently, we will show the classification for the different types of pattern outcomes, where the system is defined in a non-growing domain with reflective boundary conditions (Neumann, where the derivative is zero at the boundary). Finally, we will reconstruct the 2x2 confusion matrix (Fig. 2.5) currently used as an assumption in the literature, to show a more complete view on how to interpret dispersion relations. The new confusion matrix will show more types of dispersion relation and more pattern outcomes leading to a 5x4 confusion matrix.

First, we classify the different dispersion relations obtained from LSA into 5 types:

- Stable dispersion relations (Fig 2.9A) have all eigenvalues σ below zero for any wavenumber k_n .
- Unstable dispersion relations (Fig 2.9B) have a positive eigenvalue at $k_0 = 0$ which eventually drops below zero as diffusion is introduced (i.e. $k_n > 0$).
- Hopf-type dispersion relations, as with any unstable dispersion relation, shows an instability without diffusion ($\sigma > 0$ for $k_0 = 0$) which eventually drops below zero for positive wavenumbers. However, in the case of the Hopf-type dispersion relation, when the eigenvalues cross the zero line, there is a pair of complex conjugate eigenvalues (Fig 2.9C). A Hopf-like dispersion relation is different to a Hopf bifurcation: a bifurcation displays a shift in stability as a model parameter changes, while the Hopf-like dispersion is a change in stability as a function of the wavenumber k_n .
- Turing I dispersion relations, as previously mentioned, are stable without diffusion, have an instability for a positive wavenumber, and finally become stable again for very large wavenumbers (Fig 2.9D).
- Turing I-Hopf dispersion relations, are a combination of Turing I and Hopf-type dispersion relations. As the Hopf-type dispersion, they are unstable without diffusion. Then, as k_n is increased, the system becomes stable with a pair of complex conjugates as the eigenvalues cross the zero line. Finally, a Turing I-type behaviour arises getting a peak above zero and decaying again for large wavenumbers (Fig 2.9E).

Other types of dispersion relation exist which are not displayed here such as Turing II, where the eigenvalues do not become stable again for very large wavenumbers. Therefore, this system displays an instability at very large wavenumbers which results in infinitesimally small wavelength patterns. These are considered to produce homogeneous solutions, except in the case of space discretization where they can produce small wavelength patterns (Wang et al. 2022). However, Turing II solutions are not possible in systems such as this one where all nodes are diffusing. This is because for $k_n \rightarrow \infty$, all eigenvalues σ must be negative (see Eq. 2.26).

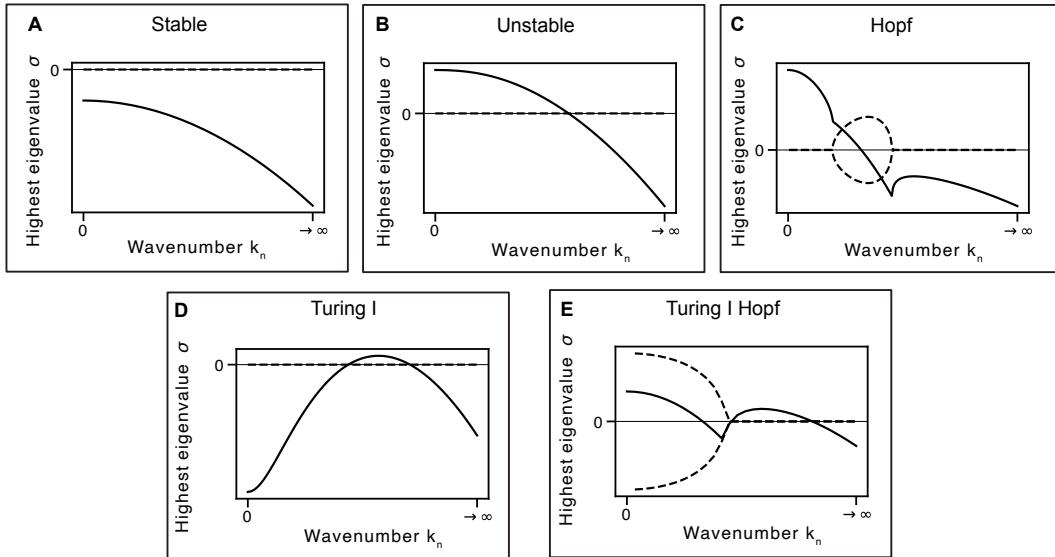


Figure 2.9: Different types of dispersion relation. Dispersion relation shows the eigenvalue as a function of wavenumber. The continuous line represents the real part, while the dotted line represents the imaginary part. **(A) Stable:** all eigenvalues are below zero. **(B) Unstable:** eigenvalues at $k_n=0$ are positive meaning the system is unstable without diffusion. **(C) Hopf:** unstable type, with hopf bifurcation for parameter wavenumber k_n . For a certain k_n , the system flips from unstable to stable, crossing the zero eigenvalue line. At the crossing, there is a pair of complex conjugate eigenvalues. **(D) Turing I:** the dispersion shows a diffusion driven instability. At $k_n=0$, the system is stable and has negative eigenvalues. As diffusion is introduced, $k_n>0$, the system becomes unstable and stable again for high k values. **(E) Turing I-Hopf:** a combination of Turing I and Hopf, meaning the system starts as a Hopf with a positive eigenvalue at $k_n=0$, then the system transitions to stable with a pair of complex conjugates. Finally, there is a dispersion peak characteristic of Turing that drops for high values of k .

As with the classification of the dispersion relations, we develop a method to classify the patterns produced numerically into homogeneous, temporal oscillator, non-stationary pattern and stationary pattern. By classifying both LSA and numerical outputs, we can generate a new confusion matrix with information about other types of dispersion relations and other types of spatio-temporal patterns.

We use a decision tree for the classification where the two layers are spatial homogeneity and convergence in time, as seen in Fig. 2.10. This decision tree leads to the 4 types of patterns mentioned. A pattern will be considered spatially homogeneous if the final snapshot U for any of the two molecular species fulfils the following condition

$$\frac{\max(U) - \min(U)}{\max(U)} \leq 0.01 \quad (2.32)$$

A pattern will be considered converged if the last 30 time points for any of the two molecular species fulfils the following condition

$$\frac{\max(U[-30 :]) - \min(U[-30 :])}{\max(U[-30 :])} \leq 0.05 \quad (2.33)$$

The thresholds chosen were fine-tuned by testing them on the numerical patterns to obtain the best classification results. Using these two characteristics, spatial homogeneity and convergence, we can obtain 4 classes of patterns as seen in Fig. 2.10:

- Homogeneous patterns are homogeneous in space and converge in time.
- Temporal oscillators, also called limit cycles, are homogeneous in space but do not converge, as they oscillate in time.
- Non-stationary patterns are not homogeneous in space and do not converge in time.
- Finally, Stationary patterns are not homogeneous in space but converge in time.

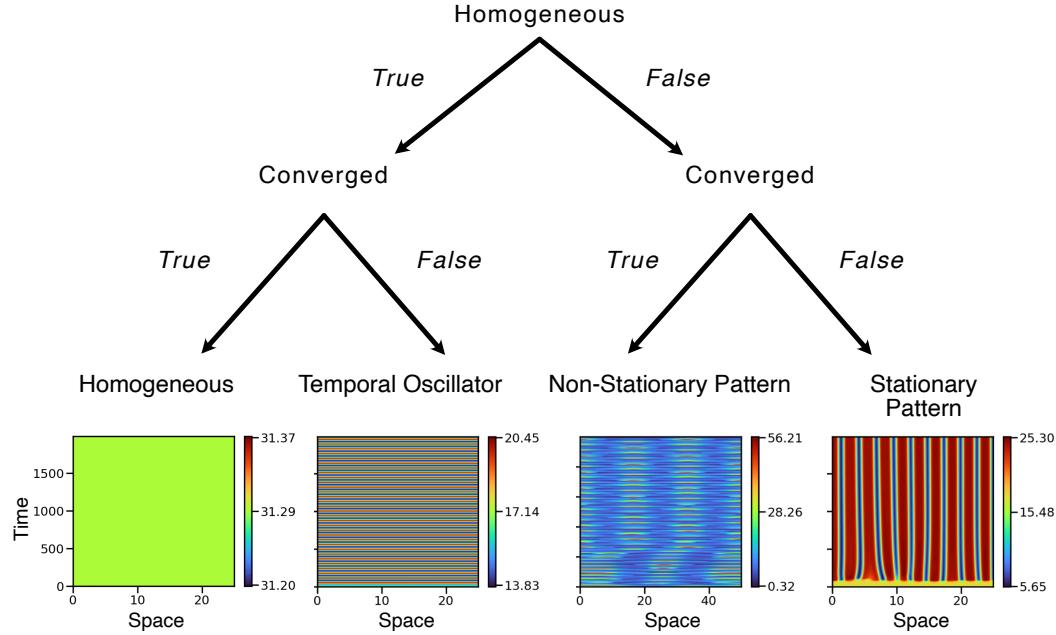


Figure 2.10: Decision tree for pattern classification in non-growing domains with reflective boundaries. A decision tree is based on two layers: spatial homogeneity and convergence. The numerical solutions for the four different pattern outcomes including a homogeneous, temporal oscillator, non-stationary pattern and stationary pattern are shown below.

High-throughput studies like Marcon et al. 2016; Scholes et al. 2019; Zheng et al. 2016 only consider Turing I as patterning and the rest is discarded. Here, we explore beyond Turing I and stationary patterns to give a more complete view of the relationship between linear stability and spatio-temporal patterns. 9,087 parameter sets are analysed from the non-linear Turing model (Eqs. 2.29) to obtain dispersion relations and numerical patterns. Because systems such as Turing and Turing I-Hopf are undersampled in the parameter space, the dataset is enriched with them to ensure we capture enough distinct dynamical behaviours. Multi-stable systems are not included in this analysis to understand the direct relationships between the dispersion relation and numerics. The two types of classifications described above are applied, and a new 5x4 confusion matrix is generated (Fig. 2.11).

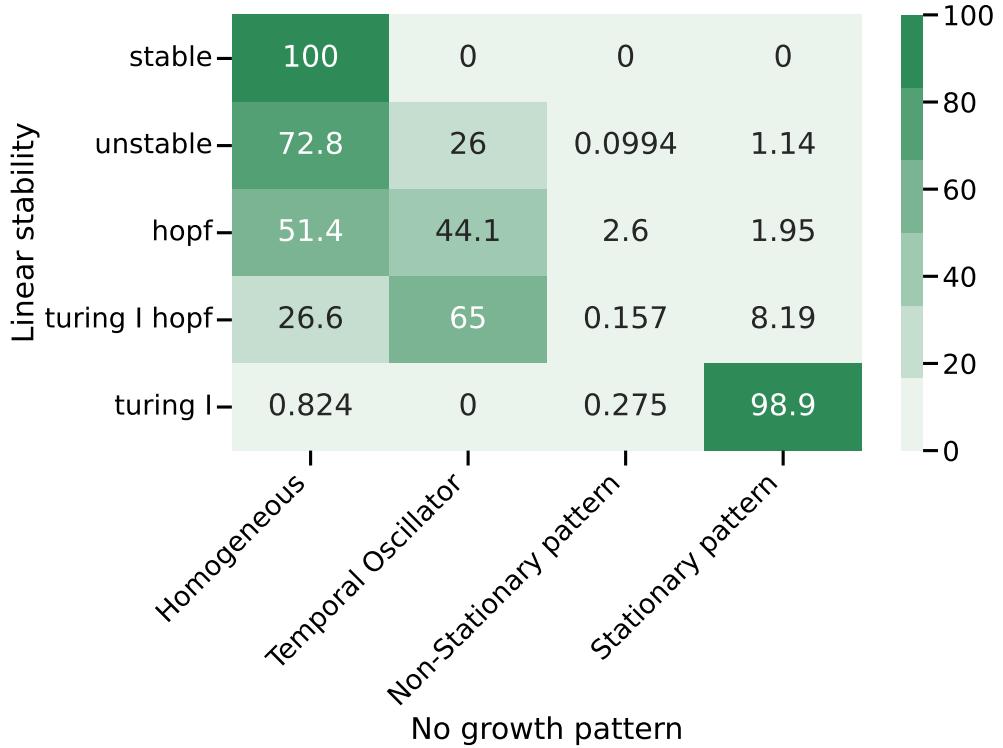


Figure 2.11: **Confusion matrix linking LSA output (rows) and numerical pattern outcome (columns).** Numbers show the percentage of solutions across the LSA output rows.

It is worth pointing out some interesting examples obtained from this confusion matrix. For example, Turing I-Hopf seems to lead to stationary patterns often (see Fig. 2.12A), meaning they would need to be included in high-throughput robustness studies. The production of periodic patterns by Turing I-Hopf instabilities has been already documented in the literature Liu et al. 2007. Some unstable dispersion relations can also lead to stationary patterns (see Fig. 2.12B). Additionally, Hopf solutions display interesting non-stationary patterns (see Fig. 2.12C). Unfortunately, some Hopf solutions are misclassified as Stationary patterns (see Fig. 2.12D) due to an inadequate threshold of the convergence classification. In this specific case, a higher window than 30 should be taken when considering convergence and a higher threshold for homogeneity. It is important to understand that this is an initial exploratory analysis where misclassification might occur. This is because of the difficulty of setting common thresholds for a wide variety of numerical solutions with different wavelengths, time scales and dynamic ranges.

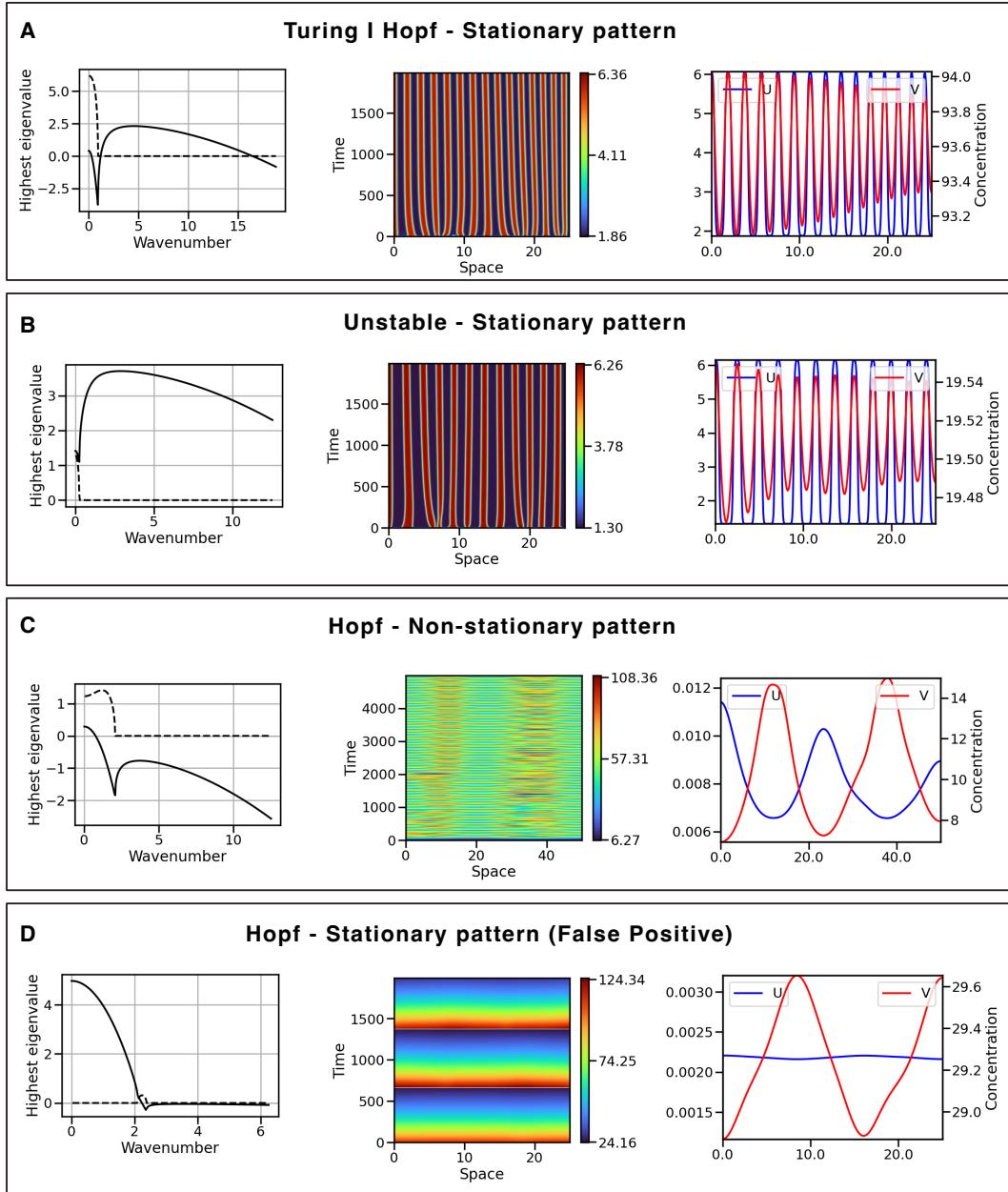


Figure 2.12: Examples of numerical solutions and their respective dispersion relations. Dispersion relation produced with LSA (left), numerical-time series (centre) and final time numerical snapshot (right) produced with Crank-Nicolson. **(A)** Turing I-Hopf solution produces stationary periodic patterns. **(B)** Unstable solution produces stationary periodic patterns. **(C)** Hopf solution produces non-stationary regular patterns. **(D)** Hopf solution produces a Temporal Oscillator which is misclassified as a Stationary pattern.

2.5 Introducing biological features: Absorbing boundaries and growth

As seen in the previous section, both multistability effects and numerical solutions can break the hypothesis that only classical Turing I systems can produce stationary periodic patterns. Here, we look deeper into how other aspects linking the theory closer to the biological reality can also break this hypothesis. In particular, we will look at how adding an absorbing boundary condition and growth to a reaction-diffusion system might induce or break patterning. This particular direction was inspired by experiments described in the next sections where growing bacterial colonies in agar are used as a platform to engineer Turing patterns using synthetic gene circuits.

The absorbing boundaries are introduced by using a Dirichlet boundary condition where the concentration at the boundary is zero, as opposed to the previously used Neumann boundaries where the derivative at the boundary is zero. This particular boundary is chosen as the boundary of the bacterial colonies modelled in Chapter 4 act as an absorbing boundary due to the lack of morphogen production in the agar. More details on the encoding of boundaries in Crank Nicolson can be found in Fig. 1.6 or Section 6.6.1.

Growth is introduced as apical isotropic linear growth, where cells are added to both boundaries with a linear growth rate. Linear growth is chosen as this is the growth observed in our experimental colonies (see Fig. C.1). More information on the different types of growth can be found in Fig. 1.5. This type of growth is inspired in linearly growing bacterial colonies where division occurs mainly at the edges. However, in this case, solutions are in 1D to speed up computation and simplify results. Growth of the tissue is encoded in a 1D binary vector, where cells are denoted as 1 and empty space as 0. The number of 1's grows linearly, which represents the tissue expanding. This vector is used as a mask, where 1's determine the computation of reaction-diffusion terms and 0's determine only the computation of diffusion. Therefore, while reaction-diffusion occurs in the tissue; only diffusion occurs in the empty regions. Fig. 2.13 shows a Turing pattern being simulated in this type of growing domain with isotropic edge growth and absorbing boundary conditions.

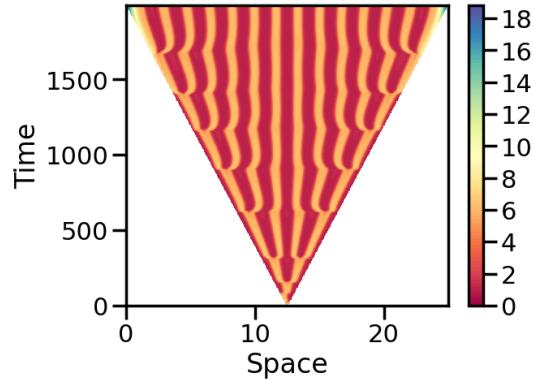


Figure 2.13: Turing pattern in linearly growing domain with absorbing boundary conditions. Time-series where the X axis denotes space while the Y axis denotes time. The colour indicates the concentration of species A. Peak doubling occurs at the edges of the growing system.

As with the non-growing reflective boundary condition patterns in the previous section (see Fig. 2.10), we want to classify the numerical output to quantify the different types of patterns obtained, and the transitions as absorbing boundaries and growth are introduced. However, due to the absorbing boundary conditions and growth, patterns are rarely spatially homogeneous or converged, which makes the previous classification method unsuitable. The new classification system is developed based on the number of peaks. Peaks are detected using the Python *find_peaks* algorithm with parameter *prominence* = 0.05. Again, the thresholds for the peak finding algorithm need to be fine-tuned and depending on the numerical outcome, misclassification might occur.

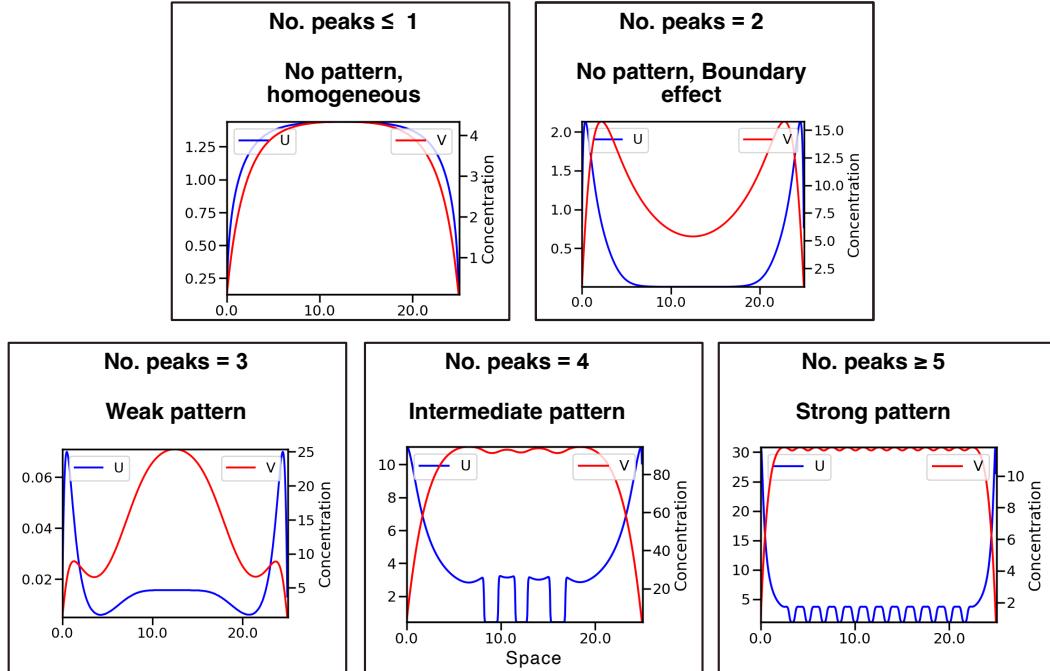


Figure 2.14: Classification based on peaks for solutions with absorbing boundary conditions and growth. Different numerical solutions in non-growing domains with absorbing boundary conditions are classified based on number of peaks. The solutions are a snapshot in time showing the concentration of A and I (shown as U and V here) in the left and right Y axis respectively. The X-axis designates space in mm.

The peak classification shown in Fig. 2.14, retrieves information on whether there is a pattern at all and whether this pattern is only a pattern at the boundary or is a periodic pattern that would scale up with tissue length as Turing patterns do. The 5 different types of patterns are:

- Patterns with one peak will be considered homogeneous as they result from the morphogens being reduced at the boundary due to absorption (Fig. 2.14 no pattern, homogeneous).
- Patterns with two peaks are also considered not to be patterned states as the two peaks might arise at the boundary for one of the diffusors due to the depletion of the other (Fig. 2.14 no pattern, boundary effect).
- Patterns with three peaks start displaying a pattern more similar to Turing repeats, although we cannot prove the number of peaks would scale with tissue length as Turing patterns do (Fig. 2.14 weak pattern).
- Patterns with four peaks could still be purely a boundary effect, but is less likely as the number of repeats points towards tissue scaling being possible (Fig. 2.14 intermediate pattern).

- Finally, patterns with five peaks and above can be considered strong patterns and would be most similar to classical Turing patterns in non-growing no-flux boundary domains (Fig. 2.14 strong pattern).

Using this classification method, 9,087 parameter sets were simulated and classified with absorbing boundary conditions and growth. Again, all these parameter sets have only a single steady state to ensure patterning effects are due to boundaries and growth and not due to multistability.

2.5.1 Reflecting boundaries to absorbing boundaries

When absorbing boundaries are added, periodic patterns might get created, disrupted or remain the same. The confusion matrix in Fig 2.16 shows this transition by comparing the classification output of reflecting boundaries versus absorbing boundaries. As previously explained, the classification in Section 2.4.2 could not be used for absorbing boundary conditions as these prevent the pattern from being completely homogeneous or stationary. Therefore, two types of classifications have to be compared for each type of boundary. Although the classification outputs cannot be directly compared, this confusion matrix is useful to identify interesting cases: If absorbing boundary conditions had no effect, we would expect homogeneous and temporal oscillator categories in the y-axis to become no pattern, homogeneous or boundary effect. Additionally, we would expect stationary patterns to become strong patterns. This is commonly the case as seen by the dark-green squares of Fig. 2.16, which shows boundaries do not often affect the pattern. However, some exceptions are present which are further studied in Fig. 2.17.

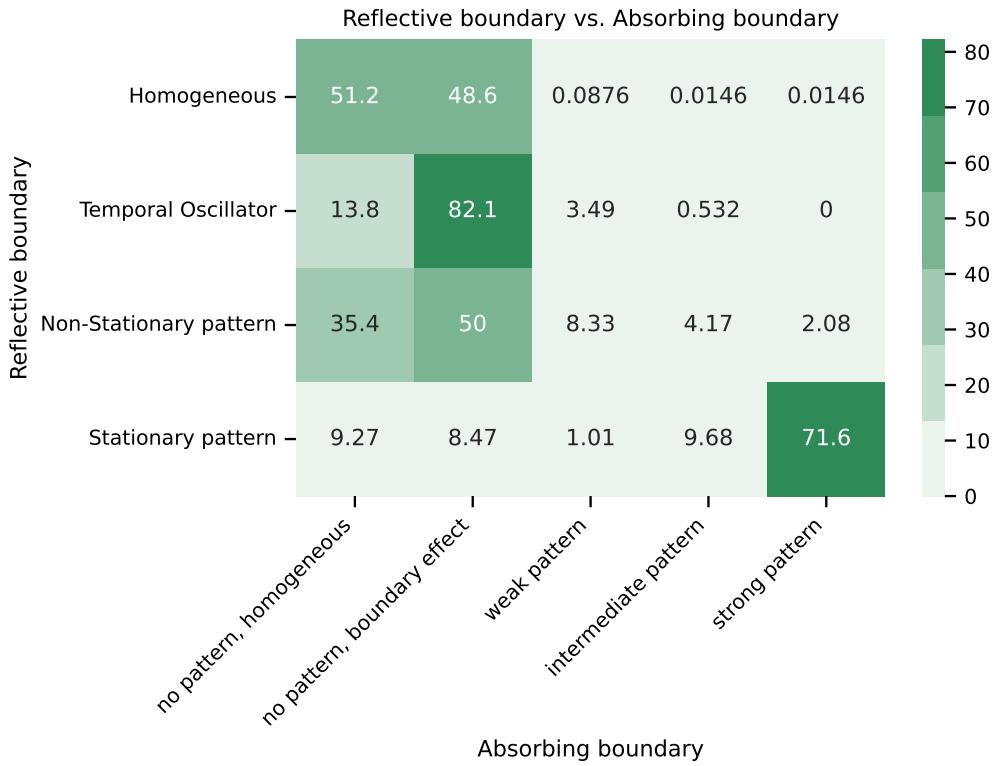


Figure 2.15: Confusion matrix linking system outcome with reflective boundary and absorbing boundary. Numbers show the percentage of solutions across the rows with reflective boundary output.

Figure 2.16: Effects of absorbing boundary conditions on patterning

By studying the exceptions in this confusion matrix, we selected four interesting cases to display in this thesis, which helps us understand how absorbing boundary conditions might affect pattern formation. Firstly, a homogenous oscillator under reflective boundaries might become a travelling wave when an absorbing boundary condition is introduced (Fig. 2.17A). Absorbing boundary conditions can also help speed up the dynamics of pattern formation as seen in Fig. 2.17B where the pattern arises 20-fold faster than with reflective boundaries. These types of boundaries can also help increase the dynamic range of the pattern as seen in Fig. 2.17C. However, this might be the same effect seen in Fig. 2.17B where the reflective boundaries form a pattern with longer time scales, therefore having a small dynamic range in earlier time points. Finally, absorbing boundaries might also disrupt the pattern formation by reducing the pattern wavelength, as seen in Fig. 2.17D. These results only correspond to a specific set of equations under a specific parameter distribution sampling. However, other effects might occur when using other systems including model parameters and numerical parameters.

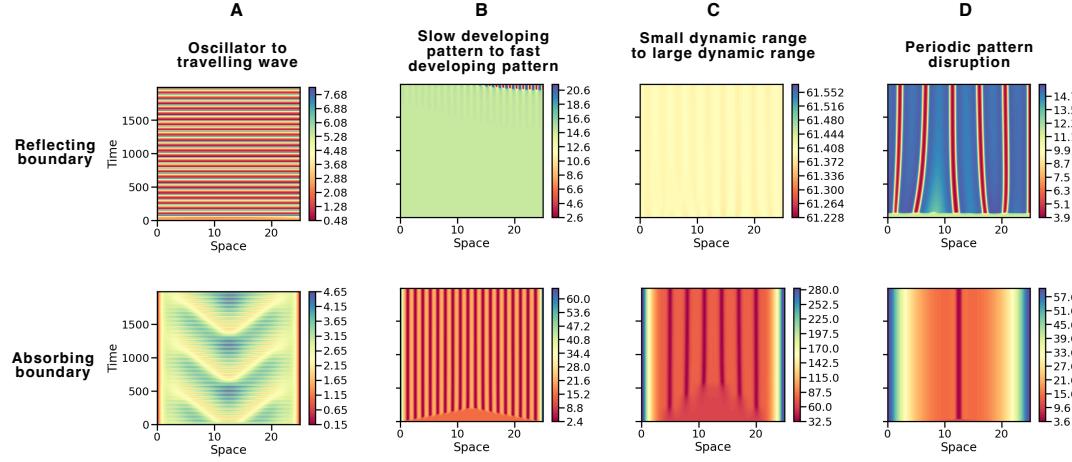


Figure 2.17: Examples of pattern transitions from reflecting boundaries to absorbing boundaries. The system is simulated numerically in a non-growing domain using reflecting boundaries (top) and absorbing boundaries (bottom). Time-series numerical solutions of A are shown for each case.

2.5.2 Open boundaries to growth

The patterning of the system is also studied when growth is added. In this case, both non-growing and growing domains have absorbing boundary conditions so we can pinpoint the effects of growth. The confusion matrix in Fig. 2.18 shows the correlation between non-growing and growing systems for 9,087 parameter sets. Overall, we can see a strong diagonal, more populated at the bottom of the diagonal. This suggests that growth often does not affect pattern formation, and when it does it is more likely to destroy patterning. Again, this result might be affected by the system of equations and the parameters used such as model parameters or numerical parameters (e.g. length of the system or growth rate).

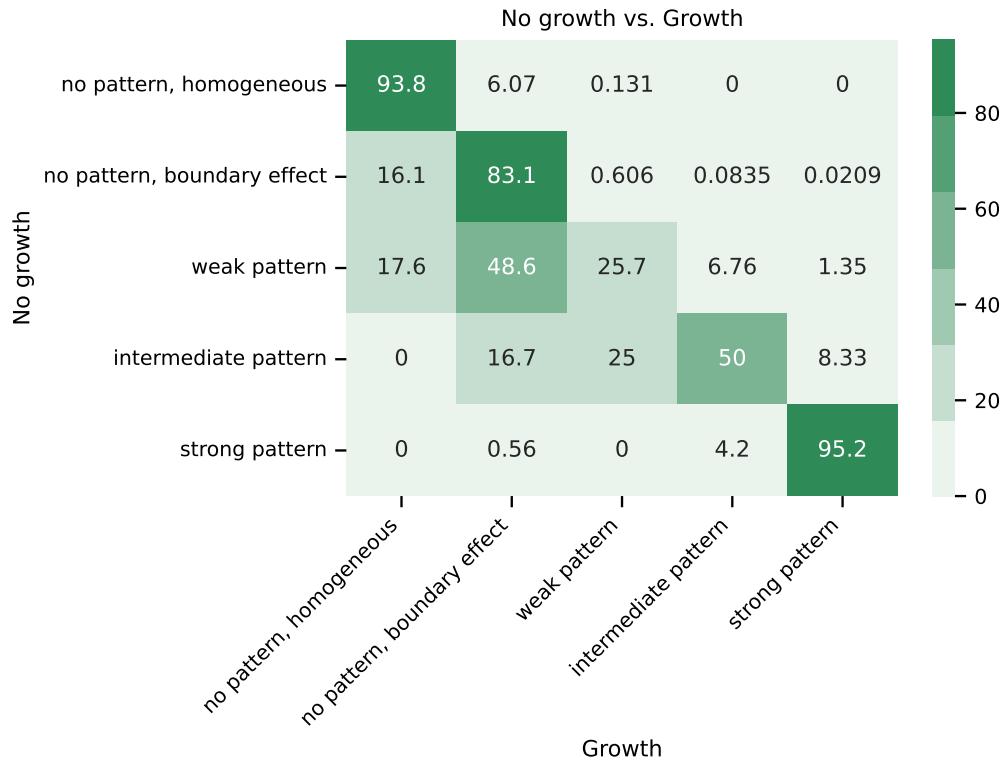


Figure 2.18: Confusion matrix linking system outcome with no growth and linear growth. Numbers show the percentage of solutions across the rows with no growth. A strong diagonal of dark green squares is depicted, which shows little effect when adding growth.

From the confusion matrix, we investigated further some of the non-diagonal examples where growth added robustness, which is shown in Fig. 2.19. However, in this case, the classification did not give good results. This is mainly because non-stationary patterns could be classified into different categories depending on the time point chosen, as the number of peaks varied over time. Therefore, the comparison between growing and non-growing did not work as it ended up being a comparison between two time points.(Fig. 2.19A-B). In some cases, we could identify patterns where the number of peaks increased just by having growth (Fig. 2.19C). Interestingly, we observe that in growing domains, we generally observe outer ring addition in for this system and growth rates (Fig. 2.13).

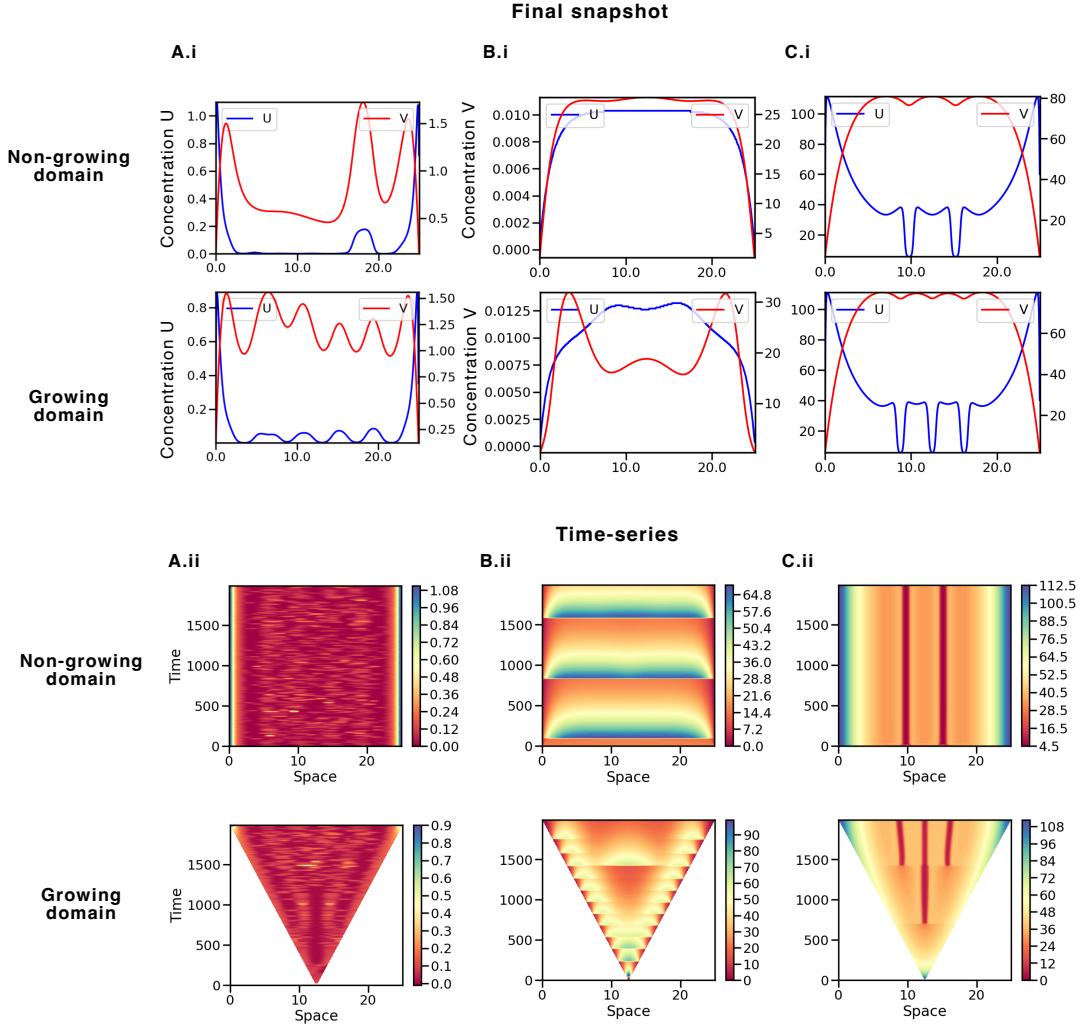


Figure 2.19: Examples of pattern transitions from non-growing domains to linearly growing domains. Three cases are shown (A, B, C). For each case, we show both growing and non-growing solutions, as well as a final snapshot (i) and time-series (ii). Peaks can be well observed in the final snapshot (i), while the pattern trajectory can be observed in the time series (ii). Simulations were carried out with CN solver in 1D.

Finally, in Fig. 2.20, we can see a Sankey diagram which represents the flows from one type of pattern to another as absorbing boundaries and growth are added. Pattern classes are ranked top-to-bottom from less patterned to more. Overall, we can see a flow upward towards less patterned outcomes as these effects are introduced. However, some flows go downwards, meaning absorbing boundaries and growth might convey robustness in some cases.

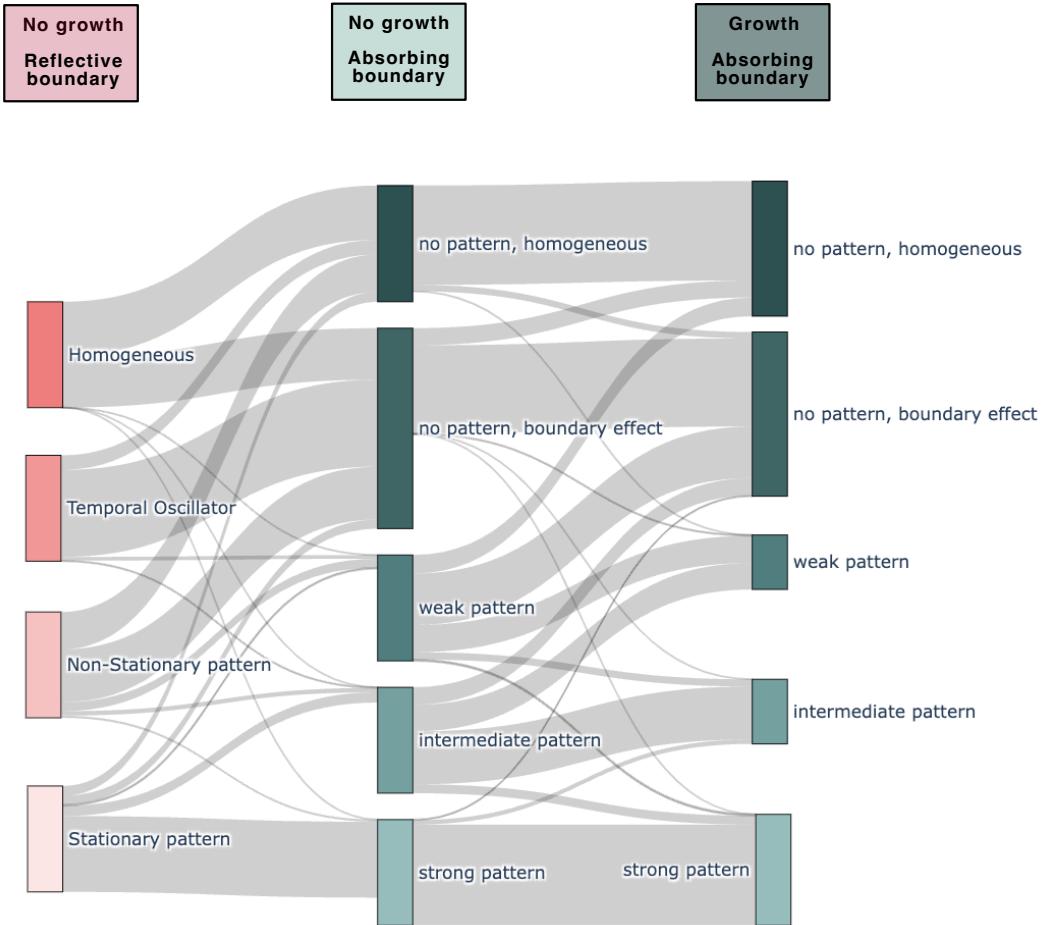


Figure 2.20: **Sankey Diagram showing flow of pattern outcomes.** The 3 layer Sankey shows the flows of patterns when going from reflective boundary conditions (left) to absorbing boundary conditions (centre) to growth. Two colour codes, pinks and blues, are used for the two types of classifications (Homogeneity-convergence decision tree and number of peaks)

2.6 Discussion

This chapter reflects all the fundamental work needed before investigating Turing patterns in the experimentally built genetic circuit which is introduced in bacterial colonies. Theoretical insights were obtained, that will later be applied in the following chapters. For this reason, the basic Turing activator-inhibitor two-node topology was chosen, which is easier to interpret and faster to compute results. However, for results to be translatable to the experimental gene circuits, non-linear Hill terms were used to describe real genetic interactions between the two nodes.

Information in the dispersion relation

We first started by understanding what information we could obtain by LSA about the patterns formed. The current state of the art merely uses the dispersion relation to see if there is a diffusion-driven instability or not. We went beyond this to

derive relationships between the dispersion relation and the numerical wavelength, convergence time and pattern shape. In the wavelength case, it seems like numerical patterns have a slightly bigger wavelength than the predicted wavelength from the dispersion relation. Unexpectedly, this relationship is not $y = x$, but $y = 1.46x$. Non-linearities might drive the system to choose slightly lower wavenumber modes which yield higher wavelengths numerically. However, a linear correlation holds, meaning we can still use LSA to predict pattern wavelength. More cases with higher wavelengths would need to be investigated to understand if this correlation is generally maintained.

A negative correlation between dispersion height and convergence time to stationary patterns was found. This can be explained, as a high eigenvalue leads to a faster exponential increase of the amplitude of the perturbations as seen in Eqs 2.24, and therefore faster time-scales. However, this correlation was less clear than in the wavelength study and only works well when studying related neighbouring patterns where a decaying exponential function is seen (Fig 2.4B). In non-neighbouring patterns (Fig 2.3B), this relationship did not capture the behaviour of patterns with low eigenvalues or low convergence time. Overall, higher dispersion peaks point towards faster forming patterns, but this should be double-checked with numerics if comparing patterns from different regions of the parameter space.

Finally, dispersion peak height also seemed to be an indicator of pattern shape in neighbouring regions of the parameter space. Higher dispersion peaks led to spot-like patterns, while lower dispersion peaks led to labyrinthian patterns. A potential explanation of this is that numerical patterns with lower dispersion peaks have not fully converged as their time scales are slower, although patterns seemed to have qualitatively converged when the time series were observed. Future work should involve testing those simulations for larger times to ensure the hypothesis is correct. Experimental work to engineer Turing patterns could benefit from faster-forming patterns as this would result in less stressed cellular tissues by avoiding nutrient depletion, accumulation of toxic molecules or cell death. By looking at which parameters changed during the optimisation, we could tune the experimental circuit to modify those parameters and therefore tune our pattern time scales. Additionally, tuning pattern shape or wavelength could be useful if different biotechnology applications require different morphologies or scales. Although we have only shown optimisation for the case of dispersion peak height, the same optimisation could be carried out to choose a particular pattern wavelength.

Breaking linear stability analysis predictions

As seen above, LSA can produce information on whether a system produces periodic stationary patterns and about the features of these patterns. However, we have

shown as well that these predictions are not as reliable as we might think, and numerical methods are therefore needed. In this chapter, we focused on how multistability might break these predictions, and how systems with other types of dispersion relations profiles can generate other stationary and non-stationary patterns.

Current literature does not explore multistability in Turing in detail. High-throughput studies either do not mention multistability or mention it but do not address it by discarding such cases (Marcon et al. 2016; Scholes et al. 2019; Zheng et al. 2016). Here we explored different cases of multistability and displayed the different potential outcomes of how the various states interact. We showed the mechanism of how unstable systems can gain Turing pattern dynamics through multistability, as well as Turing states lose their patterning. Additionally, we documented ephemeral patterns which are transient patterns occurring as the system jumps from Turing to stable states. These ephemeral patterns could still be interesting for developmental biology as the pattern can serve to temporarily activate the necessary genes to produce a non-transient phenotype (e.g. digit formation only needs periodic pattern at a specific time when the growth hormone is being produced to generate fingers (Raspopovic et al. 2014). Understanding how the interaction between different steady states affects pattern formation is key as multistability is extremely present and plays an important role in biological systems (Laurent and Kellershohn 1999).

In addition to multistability, current robustness studies do not explore alternative dispersion relations. Here we showed how certain systems with unstable or Turing I-Hopf dispersion relations can generate Turing-like stationary periodic patterns (Fig. 2.12). Even though the robustness of these systems for periodic pattern formation is limited (Fig. 2.11), in certain regions of the parameter space Hopf and unstable systems robustly led to patterns in 100% of the cases. Therefore, systems with such dispersion relations should not be completely ruled out when studying pattern formation. Considering them might increase the robustness slightly, however not enough to explain the difference between the robust mechanisms of nature and the non-robust Turing patterns. Additionally, some systems with a Hopf dispersion relation produced interesting non-stationary periodic patterns. These non-stationary patterns could be of great interest in the field of developmental and synthetic biology as gene expression arrest could turn them into periodic stationary patterns. Additionally, these non-stationary patterns could serve as a periodic pre-pattern for the initial symmetry breaking. Overall, when considering symmetric breaking events and periodicity in developmental biology, we should think beyond classical Turing dispersion relations and consider Hopf, Turing I-Hopf and simple unstable systems that might also explain periodic patterning in biology.

Introducing realistic biological phenomena: Absorbing boundaries and growth

Following this theoretical route, we studied the effects of absorbing boundaries and growth in spatio-temporal patterns. The motivation behind this was our experimental setup which involves Turing gene circuits in growing bacterial biofilms with absorbing boundary conditions. Many studies have focused on understanding how growth or boundaries might affect pattern formation as seen in Section 1.3.3. However, these studies are extremely theoretical and hard to derive any assumptions for our experimental system. Additionally, they are often based on complex analytical methods, yielding results which are hard to interpret. For this reason, we performed a high-throughput numerical study to understand the effects that the type of growth and boundaries of our system have on patterning. Our bacterial colonies grow in a linear manner, where growth occurs mainly at the edges. For this reason, we generated this type of linear edge growth. In terms of boundaries, empty agar absorbs the diffusors at the edge so we introduced absorbing boundary conditions. However, the numerical parameters (length, growth rate, time) were not specifically those of our system as they had to be fine-tuned to attain convergence, display enough peaks, and be computationally feasible. Future work should focus on matching the numerical parameters to be more similar to our experimental system.

Overall, adding absorbing boundary conditions improved patterning more than adding growth. Boundaries could induce robustness by generating non-stationary regular patterns or increasing the speed of pattern formation. However, patterns could also be disrupted by adding absorbing boundaries (Fig. 2.17). As previously mentioned, non-stationary regular patterns might be extremely interesting for developmental biology as well as biotechnology applications. Additionally, increasing the speed of pattern formation could be useful to observe the patterns within the time duration of the experiment.

On the other hand, the growth used did not seem to improve robustness for pattern formation as more cases ended up at the bottom of the diagonal in the confusion matrix (Fig. 2.18). This goes against the literature which describes growth-induced Turing patterns (Madzvamuse et al. 2010)). This might be because growth rates and types of growth (exponential or logistic) produce different results. Using different growth rates might lead to an increase in robustness or to a different type of pattern such as interior stripe growth rather than outer stripe addition shown in Fig. 2.13. The growth rate used here was slower than the experimental growth rate as long times were initially simulated to reach convergence in the non-growing reflective boundaries case. Future work should focus on testing robustness for the specific growth rates of our system. Alternatively, insights into which growth rates promote pattern formation the most could be useful to tune experiments accordingly.

The main unresolved challenge of this chapter was optimising the classification

methods to correctly classify patterns. Firstly, two different classification methods were used which made comparing results harder. Additionally, many patterns were misclassified, which meant the results were not necessarily statistically significant. However, the method still allowed the finding of interesting cases where growth or boundaries affected patterning. Better fine-tuning of threshold parameters or new classification methods is needed to avoid misclassification. Fine-tuning the thresholds is an extremely difficult task due to the wide variety of solutions with different dynamic ranges, time scales and wavelengths. This can be solved by simulating larger fields, however this is computationally expensive. Additionally, the peak number classification used for the solutions with absorbing boundary conditions and growing domains relied only on final snapshots. This was mainly a problem when classifying non-stationary patterns as the number of peaks in these solutions increases and decreases over time, which might lead to misclassification. A potential way forward could be to analyse image datasets with deep learning methods to classify resulting patterns without relying on manually set rules or thresholds.

CHAPTER 3

Gene circuit design and parametrisation

Up to now, Turing circuits that give rise to regular patterns, have only been engineered in chemical systems. Using synthetic biology, we aimed to engineer Turing patterns in biological systems using gene circuits that exhibit reaction-diffusion (RD). An RD synthetic gene circuit was built in Tica 2020 using gene regulatory functions and quorum sensing components. The reaction term comes from activation or inhibition by proteins binding to promoter or operator regions upstream of a gene and regulating its expression. The diffusion term describes the quorum-sensing molecules acting as morphogens and diffusing across the tissue, as well as forming complexes with receptors to regulate gene expression. Transitioning from a synthetic RD circuit to the generation of Turing patterns is not a trivial task due to the lack of parametric robustness of Turing systems. To address this complex task, a model is needed which can help us understand the dynamic behaviour of the RD gene circuit and whether it can produce patterns. Additionally, because of the lack of parametric robustness, it is important to learn how to tune parameters to increase the probability of obtaining patterns.

In this chapter, we present a model which describes this gene circuit. Additionally, we explore the parameter space based on literature-informed ranges to understand what fraction of it leads to Turing and how to tune the circuit to obtain a higher probability of patterning. Finally, to obtain a model which is more closely linked to our experimental system and yields more accurate predictions, we parametrise it using liquid culture data.

Work in this chapter and the following Chapter 4 has been published as a preprint in Tica et al. 2023a.

3.1 Building model describing synthetic gene circuit

In this section, I present a model which describes the synthetic gene circuit engineered in Tica 2020. The circuit can be seen in Fig. 3.1, which was already presented in Chapter 1 but is repeated here to facilitate access to the reader.

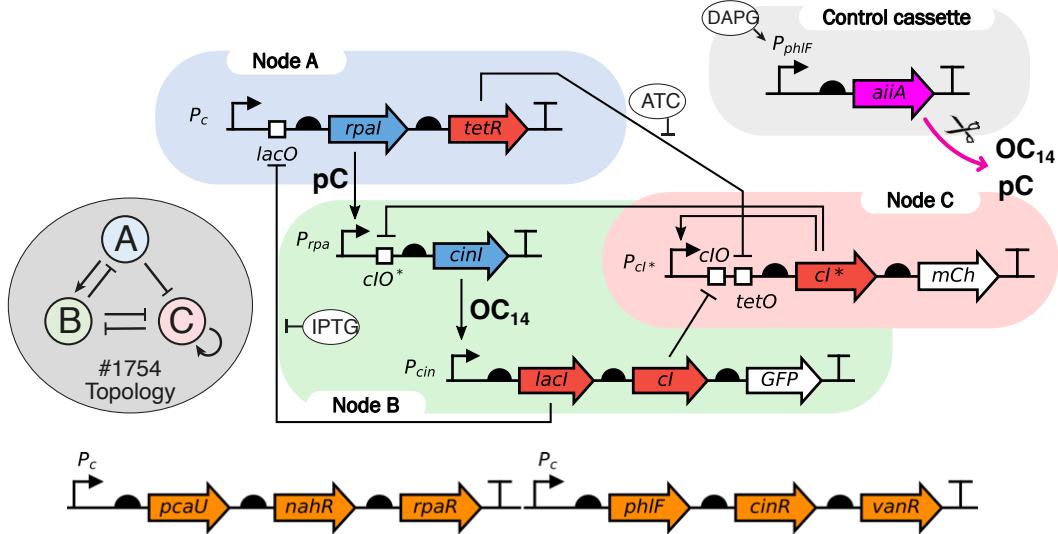


Figure 3.1: **Synthetic biology implementation of #1754 topology.** This synthetic circuit engineered by Dr. Jure Tica and Tong Zhu is a genetic abstraction of the #1754 topology in Scholes et al. 2019. This circuit is transformed so every *E. coli* cell in the biofilm has a copy inside. The original topology (left, grey inset) has only three nodes, while the synthetic circuit (right) has six nodes. The six-gene circuit architecture, shown in standard notation, can be clustered into the three original nodes as seen by the blue, green and red bubbles. Diffusor synthesis enzymes, in blue, produce quorum sensing molecules *pC* and OC_{14} . Non-diffusible transcription factors (in red), also called repressor proteins, are *lacI*, *cl*, *cl** and *tetR*. Fluorescent proteins GFP and mCherry are in white. The circuit can be regulated by small molecules *aTc*, *IPTG* and *DAPG* shown in white bubbles. *DAPG* activates the control cassette and produces regulated degradation of small quorum sensing molecules. The bottom cassette (orange), contains the necessary regulators: *rpaR* is the *pC* receptor (Node A diffusor), *cinR* is the OC_{14} receptor (Node B diffusor), and *phlF* is the *DAPG* receptor (used to tune inducible diffusor degradation). The other three receptors *pcaU*, *nahR* and *vanR* are for the protocatechuate, salicylate and vanillin inducers, which are not used in the present circuit, but were tested in other versions of the system. These three systems can be used to introduce additional regulatory components to the system.

A partial differential equation (PDE) system is used to describe the reaction and diffusion terms of this synthetic gene circuit (Fig. 3.1). This system of equations describes the concentration of molecules which are the dependent variables, in time and space which are the independent variables. The molecules modelled are the six regulator genes and two diffusor molecules. Diffusor molecules act as morphogens in this RD circuit. This model includes:

- Constant background production due to promoter leakiness
- Activator-repressor regulated gene expression using non-linear sigmoidal Hill terms
- Linear degradation of proteins and diffusor molecules or morphogens
- Diffusion of morphogens in a two-dimensional (2D) space
- Effect of tuning molecules aTc, IPTG and DAPG on the circuit and diffusion

The dynamics of a generic protein (X) and generic morphogen (U) of the gene circuit are modelled in 2D as

$$\frac{\partial[X]}{\partial t} = b_X + V_X \cdot \frac{1}{1 + \left(\frac{KA}{[A]}\right)^{n_A}} \cdot \frac{1}{1 + \left(\frac{[I]}{K_I}\right)^{n_I}} - \mu_X \cdot [X] \quad (3.1a)$$

$$\frac{\partial[U]}{\partial t} = k_1 \cdot [A] - \mu_U \cdot [U] + D_U \nabla^2[U] \quad (3.1b)$$

This results in a PDE model with eight equations that correspond to the two diffusors (pC and OC14) and six protein species of the circuit: RpaI, CinI, TetR, LacI, cI* and cI. This PDE model is reduced to a six-equation system by assuming a quasi-steady state for the diffusors, with production and degradation kinetics much faster than those of proteins. Finally, the model is non-dimensionalised to enable its parametrisation using liquid culture dose-response data. In the subsections below, the different terms of Eq. 3.1 are discussed. Furthermore, the process of model reduction, non-dimensionalisation and fitting is also described.

3.1.1 Protein equations and gene regulation

The rate of protein production can be defined as

$$V = V_{max} \cdot \theta([X]) \quad (3.2)$$

where θ is a function of a generic protein X concentration which represents the fractional activation of the system. Full activation of protein production is denoted as $\theta = 1$ while full inhibition is represented by $\theta = 0$. V_{max} is the maximal rate of expression. θ can be represented by a Hill function and an inverse-Hill function to describe cooperative binding, derived using the law of mass action with all-or-none binding to multiple binding sites (Weiss 1997). We further apply the quasi-steady state assumptions for activator and inhibitor binding to the promoter, as well as for the mRNA dynamics, as these timescales are much faster than protein production

(Andersen et al. 1998; Bremer and Dennis 2008). This leads to the following expression of θ for non-competitive activation (A) and inhibition (I)

$$\theta([A], [I]) = \frac{1}{1 + \left(\frac{K_A}{[A]}\right)^{n_A}} \cdot \frac{1}{1 + \left(\frac{[I]}{K_I}\right)^{n_I}} \quad (3.3)$$

This Hill function is used in the Eq. 3.1a and describes promoter activity as a function of the two inputs $[A]$ and $[I]$, where K_A and K_I are half-activation/inhibition concentrations, n_A and n_I are the Hill coefficients. Additionally, most promoters are leaky, which we account for by introducing a small rate of background production b_X . corresponds to the first term in Eq. 3.1a

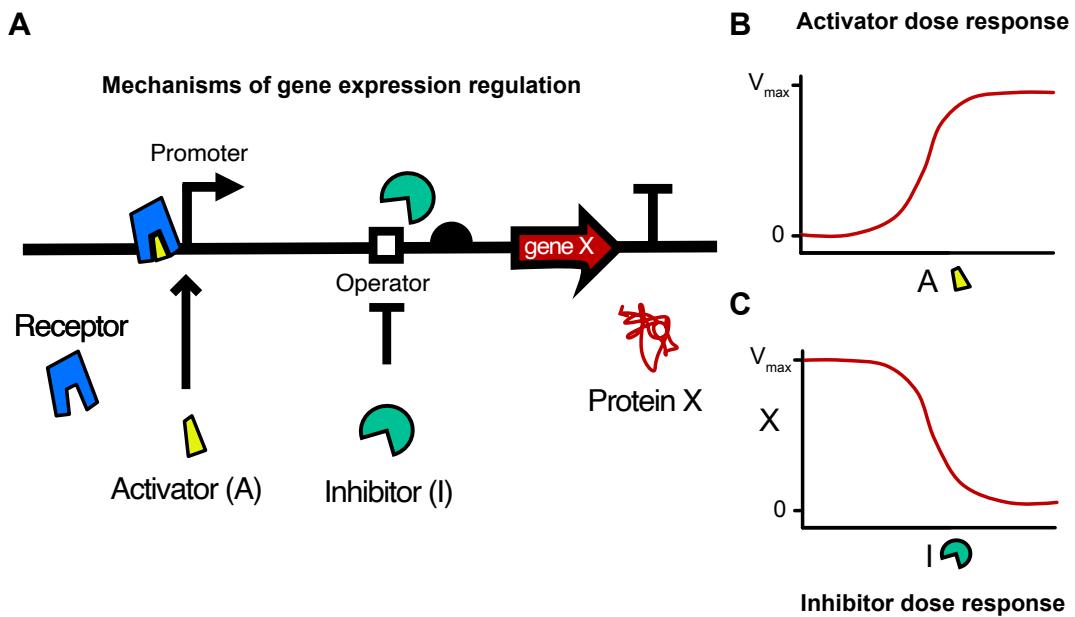


Figure 3.2: Activation and Inhibition of gene expression. (A) Mechanisms of gene expression regulation: The activator molecule (yellow) binds a receptor (blue), and this complex binds the promoter to activate gene expression. The inhibitor (green) binds the operator and inhibits gene expression. In the case of this model, activators are diffusors (pC , OC_{14}) with their respective receptors ($RpaR$, $CinR$) and inhibitors are proteins ($RpaI$, $CinI$, $TetR$, $LacI$, cI , cI^*). (B) Expression of X dependent on activation by A modelled with a Hill function. (C) Expression of X dependent on inhibition by I modelled with an inverse-Hill function.

A visual representation of the mechanisms of activation and inhibition modelled in Eq. 3.3 can be observed in Fig. 3.2. Using Eq. 3.2 and Eq. 3.3, activator and inhibitor dose curves are can be generated (Fig. 3.2B-C). These dose-response curves show the relationship between the activator or inhibitor and the rate of production of protein X. Steeper curves are generated with higher cooperativity constants, which are typical of cooperative systems where multiple transcription factors (TFs) can bind simultaneously. These steep curves resemble electrical engineering circuits, which

use step functions instead of dose-response curves, meaning a specific inducer concentration will drive the system from 0 to V_{max} . Finally, the K parameter in Eq. 3.3 corresponds to the concentration of inducer required to reach production of X at $1/2V_{max}$.

3.1.2 Tuning gene expression: aTc regulation of TetR and IPTG regulation of LacI

Experimentally, the circuit was designed to be tuned in a variety of ways using aTc, IPTG and DAPG. This tuning was used to achieve parameter combinations that are more favourable for patterning. The following section introduces aTc and IPTG tuning into the model.

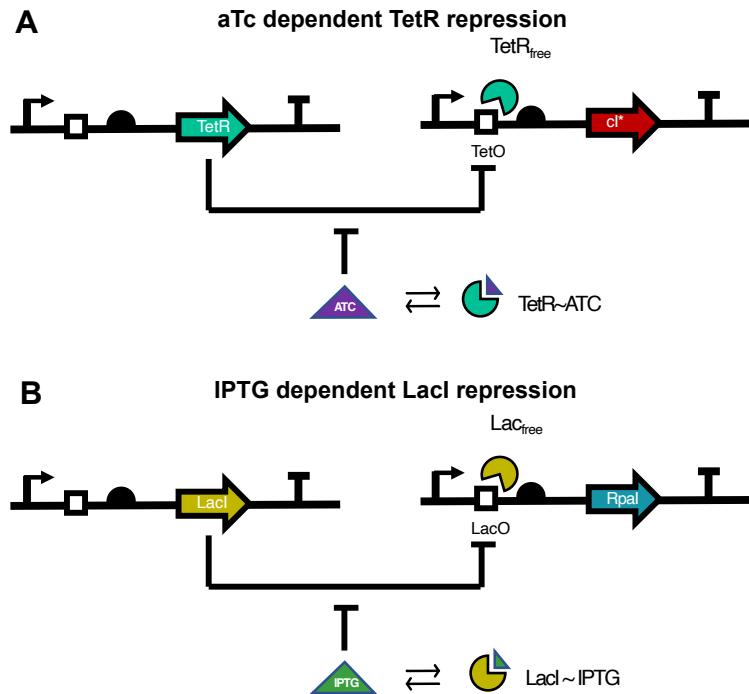


Figure 3.3: **aTc and IPTG dependent repressions.** **(A)** TetR represses cI^* by binding the TetO operator sequence. This repression can be inhibited by competitive binding of aTc to TetR. **(B)** LacI repression of $Rpal^*$ is inhibited by IPTG through competitive binding to LacI.

As shown in Fig. 3.3A, aTc binds to TetR and inactivates it. Only free, unbound TetR can bind TetO and inactivate the expression of cI^* . The binding of aTc to TetR is modelled by a reversible equilibrium. The affinity of binding is given by the k_{on} and k_{off} rate constants.



This equilibrium happens at much faster rates than the protein production and degradation reactions of the model. For this reason, quasi-steady state is assumed

$$\frac{\partial \text{TetR-aTc}}{\partial t} = k_{on}[\text{TetR}_{free}][aTc]^n - k_{off}[\text{TetR-aTc}] \approx 0 \quad (3.5)$$

By rearranging terms and using $K_d = k_{off}/k_{on}$, a simplified expression for $[\text{TetR}_{free}]$ is obtained, which is then used in the model

$$[\text{TetR}_{free}] = K_D \cdot \frac{[\text{TetR-naTc}]}{[aTc]^n} \quad (3.6)$$

However, because $[\text{TetR} - naTc]$ is unknown, the $[\text{TetR}_{total}]$ and $[\text{TetR}_{free}]$ are used instead

$$[\text{TetR-naTc}] = [\text{TetR}_{total}] - [\text{TetR}_{free}] \quad (3.7)$$

to obtain the following expression

$$[\text{TetR}_{free}] = K_D \cdot \frac{[\text{TetR}_{total}] - [\text{TetR}_{free}]}{[aTc]^n} \implies \quad (3.8a)$$

$$[\text{TetR}_{free}](1 + \frac{K_D}{[aTc]^n}) = \frac{K_D \cdot [\text{TetR}_{total}]}{[aTc]^n} \implies \quad (3.8b)$$

$$[\text{TetR}_{free}] = \frac{K_D \cdot [\text{TetR}_{total}]}{[aTc]^n + K_D} = \frac{[\text{TetR}_{total}]}{1 + \frac{[aTc]^n}{K_D}} \quad (3.8c)$$

which says that the concentration of unbound TetR is given by $[aTc]$, $[\text{TetR}_{total}]$ and the equilibrium constant K_D .

For non-cooperative systems where $n = 1$, $K_D = [L_{half}]$. However, for cooperative systems where $n > 1$, $[L_{half}] = \sqrt[n]{K_D}$. We define a new variable $K_A = \sqrt[n]{K_D} \leftrightarrow K_D = K_A^n$ that we use to replace K_D in the following expression, where $K_A^n = K_{\text{TetR-naTc}}[aTc]^n$. This was also done for other K_D parameters of the model.

$$[\text{TetR}_{free}] = \frac{[\text{TetR}_{total}]}{1 + (\frac{[aTc]}{K_{\text{TetR-aTc}}})^{n_{aTc}}} \quad (3.9)$$

Parameter $K_{\text{TetR-aTc}}$ is the dissociation constant of aTc to TetR, whereas n_{aTc} is the cooperativity of TetR and aTc binding. This expression for unbound TetR ($[\text{TetR}_{free}]$), can then be introduced into the production rate of cI*. cI* has an operator sequence downstream of the promoter, TetO. This TetO is bound by TetR to inhibit the expression of the mRNA. We will model the production of cI* dependent on TetR binding, using an inverse-Hill term that describes repression by this binding as described by Eq. 3.2 and Eq. 3.3.

$$\frac{\partial[cI^*]}{\partial t} = V_{max} \cdot \left(1 + \left(\frac{[TetR_{free}]}{K_{TetR-TetO}} \right)^{n_{TetO}} \right)^{-1} = V_{max} \cdot \left(1 + \left(\frac{TetR_{total}}{K_{TetR-TetO} \cdot (1 + (\frac{aTc}{K_{TetR-aTc}})^{n_{aTc}})} \right)^{n_{TetO}} \right)^{-1} \quad (3.10)$$

We can simplify the denominator so

$$\frac{\partial[cI^*]}{\partial t} = V_{max} \cdot \left(1 + \left(\frac{[TetR_{free}]}{K_{TetR-TetO-aTc}} \right)^{n_{TetO}} \right)^{-1} \quad (3.11)$$

where

$$K_{TetR-TetO-aTc} = K_{TetR-TetO} \cdot (1 + (\frac{aTc}{K_{TetR-aTc}})^{n_{aTc}}) \quad (3.12)$$

The parameter $K_{TetR-TetO-aTc}$ is the dissociation constant of TetR to TetO, $K_{TetR-aTc}$ is the affinity of aTc to TetR, n_{aTc} is the Hill coefficient of aTc and TetR, and n_{TetO} is the Hill coefficient of TetR to TetO. Therefore, the higher the aTc, the higher the production of cI*, which is the transcription factor repressing node B.

The same logic can be applied for the IPTG regulating LacI inhibition (see Fig. 3.3B), where unbound LacI can be or $[LacI_{free}]$ can be expressed as

$$[LacI_{free}] = \frac{[LacI_{total}]}{1 + (\frac{[IPTG]}{K_{LacI-IPTG}})^{n_{IPTG}}} \quad (3.13)$$

LacI-free will be then able to bind to LacO and inhibit RpaI and TetR production in the following manner

$$\frac{\partial[RpaI]}{\partial t} = V_{max} \cdot \left(1 + \left(\frac{[LacI_{free}]}{K_{LacI-LacO}} \right)^{n_{LacO}} \right)^{-1} = V_{max} \cdot \left(1 + \left(\frac{LacI_{total}}{K_{LacI-LacO} \cdot (1 + (\frac{IPTG}{K_{LacI-IPTG}})^{n_{IPTG}})} \right)^{n_{LacO}} \right)^{-1} \quad (3.14)$$

We can again simplify the denominator so

$$\frac{\partial[RpaI]}{\partial t} = V_{max} \cdot \left(1 + \left(\frac{[LacI_{free}]}{K_{LacI-LacO-IPTG}} \right)^{n_{LacO}} \right)^{-1} \quad (3.15)$$

where,

$$K_{LacI-LacO-IPTG} = K_{LacI-LacO} \cdot (1 + (\frac{IPTG}{K_{LacI-IPTG}})^{n_{LacO}}) \quad (3.16)$$

The parameter $K_{LacI-LacO-IPTG}$ is the dissociation constant of TetR to TetO, $K_{LacI-IPTG}$ is the affinity of IPTG to LacI, n_{IPTG} is the Hill coefficient of IPTG and LacI, and n_{LacO} is the Hill coefficient of LacI to LacO. Therefore, the higher the IPTG, the higher the production of RpaI, which is the synthesis enzyme producing pC to activate node B.

3.1.3 Degradation

All the species of the circuit, including proteins and diffusors, were modelled to undergo linear, or first order, degradation

$$-\mu_{P_1}[X] \quad (3.17)$$

as seen in Eq. 3.1. The degradation rate parameters for the protein species and diffusors are readily available in the literature (Andersen et al. 1998; Kaufmann et al. 2005) and can be found in Table 3.1. They are scale-free parameters that only depend on time, which makes them more translatable between different experimental contexts and units of measurement.

Degradation tag	Molecule	Rate [h^{-1}]
LVA 1	CinI, LacI, cI, cI*, TetR 1	1.14
ASV	RpaI, GFP, mCherry 3	0.30
diffusors	pC, OC ₁₄	0.0225

Table 3.1: **Degradation Parameters.** Degradation coefficients of protein species with degradation tags LVA and ASV, and basal degradation of diffusors.

The proteins of the circuit had short peptide sequences added at their 3' ends, known as degradation tags, that promote their breakdown by the cell's proteolytic enzymes. The diffusor's degradation could be further induced by adding exogenous DAPG to the system which induces degradation by AiiA lactonase.

3.1.4 Diffusor equations

Diffusors (pC and OC_{14}) are synthesised by enzymes (rpaI and cinI), and then bound to receptors (rpaR and cinR) to activate gene expression by binding to the promoters (Prpa and Pcin) as seen Fig. 3.2. The circuit receptors are expressed constructively from a low-copy pCC1 plasmid, The circuit receptors are expressed constructively from a low-copy pCC1 plasmid, and are therefore modelled with a constant concentration. The quasi-steady state was assumed for the very fast equilibrium that forms between the receptor and the diffusors. The receptor-inducer and receptor-promoter binding equilibria were modelled with mass action kinetics.

The enzymatic production of the diffusors was modelled with a simple linear production term dependent on synthesis enzyme concentration ([A], [B]), and a rate

constant (K_1 and K_2). It is assumed that the precursor substrate concentration is in excess and does not influence the reaction rate.

The diffusors are also modelled to undergo linear degradation, parameterised by μ_U and μ_V as seen in Eq. 3.17. This degradation is a combination of spontaneous hydrolysis in water, enzymatic AiiA-dependent degradation, and other cellular metabolic processes (Kaufmann et al. 2005; Momb et al. 2008; Wang et al. 2004). Finally, the diffusor movement through space is modelled with simple diffusion terms and parameterised with diffusion coefficients D_U and D_V .

$$\frac{\partial[U]}{\partial t} = k_1 \cdot [A] - \mu_U \cdot [U] + D_U \nabla^2[U] \quad (3.18a)$$

$$\frac{\partial[V]}{\partial t} = k_2 \cdot [B] - \mu_V \cdot [V] + D_V \nabla^2[V] \quad (3.18b)$$

The time dynamics of diffusor synthesis are much faster than that of protein production. Therefore, the rate of change of diffusors is expected to be much higher than that of proteins. These rapid fluctuations of diffusors can be approximated to equilibrium by using the quasi-steady state approximation, meaning the rate is set to zero. This leads to an expression of diffusor concentration that is linearly correlated with their synthesis enzyme and rate constant, and inversely correlated with their degradation rate.

$$\frac{\partial[U]}{\partial t} = k_1 \cdot [A] - \mu_U \cdot [U] = 0 \longrightarrow U = \frac{k_1}{\mu_U} [A] \quad (3.19a)$$

$$\frac{\partial[V]}{\partial t} = k_2 \cdot [B] - \mu_V \cdot [V] = 0 \longrightarrow V = \frac{k_2}{\mu_V} [B] \quad (3.19b)$$

In this process, the diffusion of U and V is artificially assigned to A and B, since

$$U = \frac{k_1}{\mu_U} [A] \longrightarrow D_U \nabla^2[U] = D_U \frac{k_1}{\mu_U} \nabla^2[A] \quad (3.20a)$$

$$V = \frac{k_2}{\mu_V} [B] \longrightarrow D_V \nabla^2[V] = D_V \frac{k_2}{\mu_V} \nabla^2[B] \quad (3.20b)$$

These quasi-steady state expressions are used to model the diffusors implicitly within the gene equations. Therefore, instead of using eight equations for the six genes and two diffusors; we have six equations for the six genes with the two diffusors modelled implicitly. U and V will be modelled within A and B in the Hill activation and diffusion terms

$$\frac{\partial[A]}{\partial t} = b_A + V_A \cdot \frac{1}{\left(1 + \left(\frac{[I]}{K_I}\right)^{n_I}\right)} - \mu_A \cdot [A] + \frac{k_1 D_U}{\mu_U} \cdot \nabla^2[A] \quad (3.21)$$

$$\frac{\partial[B]}{\partial t} = b_B + V_B \cdot \frac{1}{1 + \left(\frac{\mu_u K_{ub}}{k_1 [A]}\right)^{n_{ab}}} \cdot \frac{1}{1 + \left(\frac{[I]}{K_I}\right)^{n_I}} - \mu_B \cdot [B] + \frac{k_2 D_V}{\mu_V} \cdot \nabla^2[B] \quad (3.22)$$

Although these genes are not explicitly modelled, a linear relationship between downstream genes in the same operon.

3.1.5 Six-equation model

We combined all the terms described above including basal production, activator or inhibitor-regulated production, tuning molecules, linear degradation and implicit diffusors. All these terms are applied to the circuit shown in Fig 3.1. This results in a six-equation model with information on the six proteins in space and time

$$\frac{\partial[A]}{\partial t} = b_A + V_A \cdot \frac{1}{\left(1 + \left(\frac{[D]}{K_{da}}\right)^{n_{da}}\right)} - \mu_A \cdot [A] + \frac{k_1 D_U}{\mu_U} \cdot \nabla^2[A] \quad (3.23a)$$

$$\frac{\partial[B]}{\partial t} = b_B + V_B \cdot \frac{1}{1 + \left(\frac{\mu_u K_{ub}}{k_1 [A]}\right)^{n_{ab}}} \cdot \frac{1}{1 + \left(\frac{[E]}{K_{eb}}\right)^{n_{eb}}} - \mu_B \cdot [B] + \frac{k_2 D_V}{\mu_V} \cdot \nabla^2[B] \quad (3.23b)$$

$$\frac{\partial[C]}{\partial t} = b_C + V_C \cdot \frac{1}{\left(1 + \left(\frac{[D]}{K_{da}}\right)^{n_{da}}\right)} - \mu_C \cdot [C] \quad (3.23c)$$

$$\frac{\partial[D]}{\partial t} = b_D + V_D \cdot \frac{1}{1 + \left(\frac{\mu_v K_{vd}}{k_2 [B]}\right)^{n_{bd}}} - \mu_D \cdot [D] \quad (3.23d)$$

$$\frac{\partial[E]}{\partial t} = b_E + V_E \cdot \frac{1}{\left(1 + \left(\frac{[C]}{K_{ce}}\right)^{n_{ce}}\right)} \cdot \frac{1}{1 + \left(\frac{[F]}{K_{fe}}\right)^{n_{fe}}} \cdot \frac{1}{1 + \left(\frac{K_{ee}}{[E]}\right)^{n_{ee}}} - \mu_E \cdot [E] \quad (3.23e)$$

$$\frac{\partial[F]}{\partial t} = b_F + V_F \cdot \frac{1}{1 + \left(\frac{\mu_v K_{bd}}{k_2 [B]}\right)^{n_{bd}}} - \mu_F \cdot [F] \quad (3.23f)$$

where K_{da} and K_{ce} are expressed in terms of the tuning molecules such as

$$K_{da} = K_{lacI-lacO} \left(1 + \frac{[IPTG]}{K_{lacI-IPTG}}\right)^{n_{IPTG}} \quad (3.24a)$$

$$K_{ce} = K_{tetR-tetO} \left(1 + \frac{[aTc]}{K_{tetR-aTc}} \right)^{n_{aTc}} \quad (3.24b)$$

The six dependent variables can be found in Table 3.2. They can be thought of as protein concentrations, although they are also directly linked to the gene expressed to produce that protein. The units are nM.

Table 3.2: Model dependent variables. The model-dependent variables are the concentrations of the six regulatory proteins. The names of the molecule and the original node they belong to in the original #1754 topology are also shown.

Variable	Biological molecule	Node in original 3-node circuit
[A]	RpaI	A
[B]	CinI	B
[C]	TetR	A
[D]	LacI	B
[E]	cI*	C
[F]	cI	B

A description of the parameters used in the model can be found in Table 3.3.

Table 3.3: Model parameters, description and units

Parameter	Description	Units
t	Time	h
x	Space	mm
b_X	Background production rate of X	nM/h
V_X	Induced max production rate of X	nM/h
K_{XY}	Dissociation constant of X binding to Y regulator DNA	nM
n_{XY}	Cooperativity constant of X binding to Y regulator DNA	1
μ_X	Degradation rate of X	1/h
D_X	Diffusion rate of diffusor X	mm^2/h

For K_{XY} and n_{XY} , activators X will bind to a receptor molecule that binds the promoter sequence upstream of gene Y; while inhibitor X will bind to the operator sequence upstream of gene Y.

3.1.6 Dimensionless six-equation model

A dimensionless model (or non-dimensional model) is derived to better understand the nature of the parameters, and to simplify the fitting process, as explained below. The work on transforming the six-equation model into a non-dimensional form was done in collaboration with Dr. Roozbeh Pazuki. The non-dimensionalisation involves the following transformations of dependent and independent variables of concentration (X), time (t) and space (x and y).

$$X = \frac{b_x}{\mu_x} X^*, t = \frac{t^*}{\mu_a}, x = \sqrt{\frac{k_1 D_u}{\mu_a \mu_u}} x^*, y = \sqrt{\frac{k_1 D_u}{\mu_a \mu_u}} y^* \quad (3.25)$$

and the following transformations of the system parameters V, μ , K and D:

$$V_x^* = \frac{V_x}{b_x}, \mu_x^* = \frac{\mu_x}{\mu_a}, K_{yx}^* = \frac{\mu_x}{b_x} K, D_r = \frac{k_2 D_v \mu_u}{k_1 D_u \mu_v} \quad (3.26)$$

It is important to note that time and degradation of all species is now relative to the degradation rates of A (RpAI) which is an estimated parameter from the literature (see Table 3.1). If this parameter changes throughout the experiment, the effects should be absorbed by the dimensionless model. If those transformations are applied to the original six-equation model shown in Eqs. 3.23, the following dimensionless model is obtained

$$\frac{\partial[A^*]}{\partial t^*} = 1 + V_a^* \left(\frac{1}{1 + \left(\frac{[D^*]}{K_{da}^*} \right)^{n_{da}}} \right) - [A^*] + \nabla^2[A^*] \quad (3.27a)$$

$$\frac{\partial[B^*]}{\partial t} = \mu_b^* \left(1 + V_B^* \left(\frac{1}{1 + \left(\frac{\mu_u K u b^*}{k_1 [A^*]} \right)^{n_{ab}}} \right) \cdot \left(\frac{1}{1 + \left(\frac{[E^*]}{K_{eb}^*} \right)^{n_{eb}}} \right) - [B^*] \right) + D_r \nabla^2[B^*] \quad (3.27b)$$

$$\frac{\partial[C^*]}{\partial t^*} = \mu_c^* \left(1 + V_c^* \left(\frac{1}{1 + \left(\frac{[D^*]}{K_{da}^*} \right)^{n_{da}}} \right) - [C^*] \right) \quad (3.27c)$$

$$\frac{\partial[D^*]}{\partial t^*} = \mu_d^* \left(1 + V_d^* \left(\frac{1}{1 + \left(\frac{\mu_v K_{vd}^*}{k_2 [B^*]} \right)^{n_{vd}}} \right) - [D^*] \right) \quad (3.27d)$$

$$\frac{\partial [E^*]}{\partial t^*} = \mu_e^* \left(1 + V_e^* \left(\frac{1}{1 + \left(\frac{[C^*]}{K_{ce}^*} \right)^{n_{ce}}} \right) \left(\frac{1}{1 + \left(\frac{[F^*]}{K_{fe}^*} \right)^{n_{fe}}} \right) \left(\frac{1}{1 + \left(\frac{K_{ee}^*}{[E^*]} \right)^{n_{ee}}} \right) - [E^*] \right) \quad (3.27e)$$

$$\frac{\partial [F^*]}{\partial t^*} = \mu_f^* \left(1 + V_f^* \left(\frac{1}{1 + \left(\frac{\mu_v K_{vd}^*}{k_2 [B^*]} \right)^{n_{vd}}} \right) - [F^*] \right) \quad (3.27f)$$

A summary of the model parameters including notation, original units and dimensionless units can be found in Table 3.4.

Table 3.4: Dimensionless model parameters and unit transformations

Parameter	Description	Original Units	Dimensionless Units
X^*	Molecular species	nM	$\frac{\mu_x}{b_x} X \rightarrow \frac{nM/h}{nM/h} = 1$
OC_{14}	NodeB inducer	nM	$\frac{k_2 b_B}{\mu_v \mu_b} B^* \rightarrow 1$
b_x^*	Background production rate	nM/h	no b_x
V_x^*	Induced max production rate	nM/h	$\frac{V_x}{b_x} \rightarrow \frac{nM/h}{nM/h} = 1$
K_{yx}^*	Dissociation constant	nM	$\frac{\mu_x}{b_x} K_{yx} \rightarrow \frac{nM/h}{nM/h} = 1$
n_{yx}	Cooperativity constant	1	1
μ_x^*	Degradation rate	$1/h$	$\frac{\mu_x}{\mu_a} \rightarrow \frac{1/h}{1/h} = 1$
D_x	Diffusion rate	mm/h	$\frac{k_2 D_v \mu_u}{k_1 D_u \mu_v} \rightarrow 1$
t	Time	h	$\mu_a \cdot t \rightarrow h \cdot h^{-1} = 1$
x	Space	mm	$\sqrt{\frac{k_1 D_1}{\mu_a \mu_v}} / x \rightarrow \sqrt{\frac{h^{-1} mm^2 h^{-1}}{h^{-1} h^{-1}}} / mm = 1$

3.1.7 Fluorescence reporters

In addition to the six genes involved in circuit regulation, there are two fluorescent reporter genes: GFP and mCherry. GFP is in the same operon as lacI and cI, meaning a single mRNA is transcribed with the three genes. The difference is that each has its own Ribosome Binding Site (RBS), so they are translated into proteins independently. Therefore, we can assume GFP is linearly correlated with lacI and cI. The same thing can be inferred for mCherry and cI*. Instead of modelling GFP and mCherry as two extra equations, lacI and cI* concentrations will be used instead when studying how the fluorescent distributions look spatially

$$[GFP] = \alpha_{GFP} * [LacI] \quad (3.28a)$$

$$[mCherry] = \alpha_{mCherry} * [cI^*] \quad (3.28b)$$

where α is the fractional translation between the two proteins in the same operon.

Both the dimensional and the non-dimensional six-equation models (Eqs. 3.23 and Eqs. 3.27) are a close representation of the synthetic gene circuit engineered by Tica 2020 shown in Fig. 3.1. The non-dimensional model (Eqs. 3.27) will be used to explore the potential of this gene circuit for pattern formation by studying GFP and mCherry distributions in an *in-silico* model tissue.

3.2 Explore parameter space and optimise robustness

Before performing any experiments on the gene circuit, the parameter space of our system was explored to understand which possible spatio-temporal dynamics could be generated by this circuit. In particular, it was studied whether this six-node synthetic biology implementation of the original three-node network from Scholes et al. 2019 (Fig. 3.1) could produce Turing patterns. Finally, the parameter space was explored to understand how to tune the genetic circuit experimentally to maximise the likelihood of Turing pattern formation. Two approaches were taken which include investigating the effects of matching dose-response curves to increase Turing robustness and understanding the effects of individual parameters on Turing robustness.

3.2.1 Definition of parameter space based on literature parameters

The model is first studied by initialising the model parameters from distributions derived from the literature. This multi-dimensional distribution enables a full understanding of the gene circuit behaviour in all possible parameter regimes. A distribution for each parameter was generated, and the multi-dimensional parameter space was sampled using Latin Hypercube Sampling (see 6.2) (Bergstra et al. 2012;

Iman 2014), which maximises the sampling efficiency with fewer samples. Depending on the uncertainty of the parameter, different distributions were used such as Loguniform, Gaussian and Fixed. For each parameter of the model, literature searches were carried out to define biologically realistic lower and upper bounds. Diffusion rates (D) were estimated in unpublished data (Tica et al. 2023b). Diffusor production rate constants (k) in Pai and You 2009; Schaefer et al. 1996. Protein degradation rates in Andersen et al. 1998, while diffusor degradation in Kaufmann et al. 2005. Cooperativity values (n) in Babic and Little 2007. Other parameters that had not been well parametrised in the literature were defined with bounds such as the maximum production rate (V_X), background production rate (b_X) and Dissociation constant (K_{XY}). Although we have less confidence in these last parameter estimations, bounds were still obtained from the literature (Pušnik et al. 2019; Scholes et al. 2019). The types of distributions used for sampling and the values used for each parameter are listed in Table 3.5. To obtain dimensionless parameters from these, the bounds of the distributions are recalculated, and the distributions sampled between these new bounds as specified in the ‘Distribution’ column of Table 3.5. The transformations required to get the dimensionless bounds or values can be found in Eq. 3.25 and Eq. 3.26 or Table 3.4. For example, for Dr^* which is $(k_2 D_v \mu_u) / (k_1 D_u \mu_v)$, a loguniform distribution was sampled between new bounds [0.01, 100].

Table 3.5: Literature-derived parameter distributions. The distributions used include Loguniform and Gaussian. Some parameters are fixed to single values and not sampled. The parameters shown here are dimensional, so these distributions are then used in the dimensionless form of parameters shown in Eq. 3.25 and Eq. 3.26.

Parameter	Description	Distribution	Value
V_X	Induced max production rate of X	Loguniform	10-1000
b_X	Background production rate of X	Loguniform	0.1-1
D_U, D_V	Diffusion rate of pC and OC_{14}	Loguniform	0.1-10
K_{XY}	Dissociation constant of X binding to Y regulator DNA	Loguniform	0.1-250
K_{ee}	Dissociation constant of clI* binding its own promoter (P_{clI*})	Fixed	0.01
k_1, k_2	pC and OC_{14} production rate constants	Fixed	0.0183
μ_u, μ_v	pC and OC_{14} degradation rates	Fixed	0.0225
μ_{LVA}	Degradation rates of proteins with LVA tag (CinI, LacI, clI, clI*, TetR $\rightarrow \mu_B, \mu_C, \mu_D, \mu_E, \mu_F$)	Gaussian	mean=1.143, std=mean*0.1
μ_{ASV}	Degradation rates of proteins with LVA tag (RpaI $\rightarrow \mu_A$)	Fixed	0.3
n_{ub}, n_{vd}	Cooperativity of pC and OC_{14} with receptors binding to P_{rpa} and P_{cin} promoters respectively	Fixed	[1,2]
$n_{da}, n_{fe}, n_{ee}, n_{eb}, n_{ce}$	Cooperativity of LacI, clI, clI*, clI*, TetR binding to lacO, cIO, P_{clI*} , cIO*, tetO operators and promoter respectively	Fixed	[2,5,4,4,3]

3.2.2 Finding Turing in six-node Turing circuit and defining robustness

Linear stability analysis was used initially to understand whether the circuit built synthetically could produce Turing patterns. By searching the parameter space defined above using linear stability analysis, several diffusion-driven instabilities were found. When simulated these gave rise to Turing patterns varying from labyrinths to spots as seen in Fig. 3.4. Although the robustness is limited (as described later), the synthetic gene circuit engineered by Tica 2020 can indeed produce stationary periodic patterns resulting from a Turing mechanism.

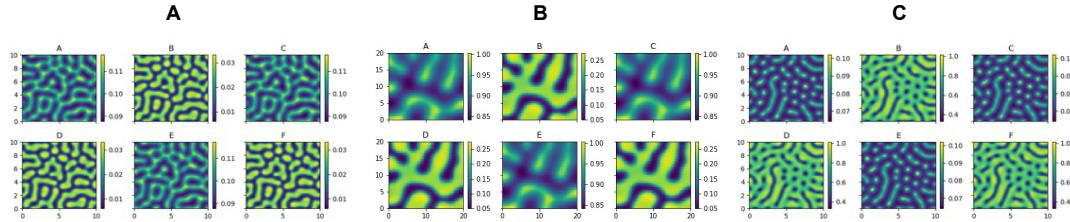


Figure 3.4: Examples of Turing patterns in the six-node circuit. Three simulations of the six-equation model with Turing I parameter sets. In each solution (A, B, C) there are six images which represent every dependent variable of the model. The solver used is the Alternating Direction Implicit Method (ADI) method with non-growing square domains and reflecting boundary conditions. Periodic patterns are observed.

3.2.3 Increasing Turing robustness by matching consequent dose-response curves

One of the aims of the model is to understand how to increase the robustness for Turing pattern formation *in-silico* before the experimental work. This way, we can then use the insights to maximise the chances of finding Turing patterns *in-vitro*. In this thesis, we describe Turing robustness as the volume of the parameter space leading to Turing I and Turing I Hopf patterns. Turing I Hopf patterns are included in this definition as in the context of the six-equation model and the parameters used, they seem to always produce periodic stationary patterns. This can be seen in the following chapter where we explore these solutions numerically (see Fig. 4.8).

The first approach consisted in investigating how a well-balanced circuit improves the likelihood of patterning. Once the components of the circuit are put together inside the cell, these components might not necessarily match well together. Gene circuits, as opposed to digital circuits, do not have binary inputs and outputs. Instead, gene expression is regulated continuously, where a continuous increase in inducer leads to a continuous change in protein concentration as seen in the dose-response curves of Fig. 3.5A (left). In this case, inducer $[A0^*]$ determines how much $[A1^*]$ is produced with a sigmoidal relationship. The dose-response function is modelled with Hill terms as seen in Eq.3.3. A well-matched transfer function occurs when the protein expression levels stemming from dose-response 1 (Fig. 3.5A (left)) are compatible with the sensitivity of the regulatory components they act on on dose-response 2 (Fig. 3.5A (right)). For example, a transfer function is not ‘matched’ when a very sensitive promoter is completely turned on even with background levels of activator. This can occur experimentally if the background levels of the activator are sufficiently high because of a leaky promoter or strong RBS. In this case, an induction of the activator above the background would not result in further activation, leading to a loss of dynamic range (Fig. 3.5B (red)).

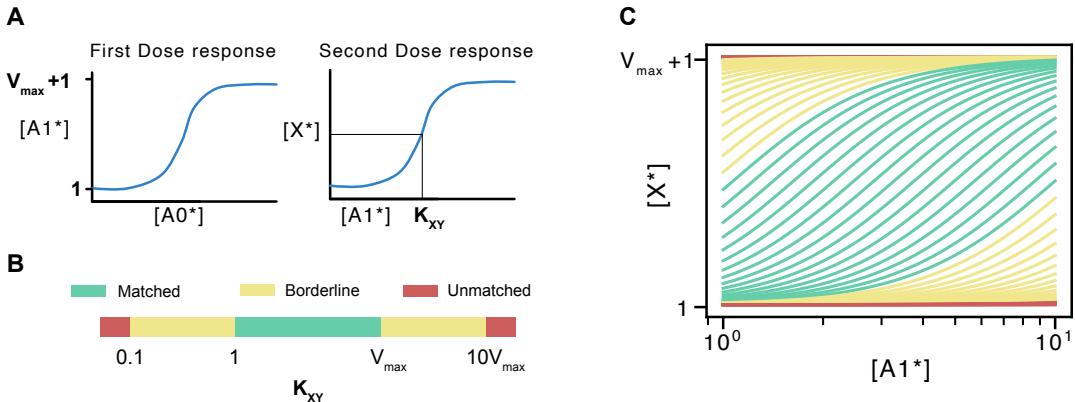


Figure 3.5: Balancing of dose-response curves for an inducer-dependent activation. (A) The first dose-response curve (left) describes the $[A0^*]$ dependent production of $[A1^*]$ which can vary from 1 to $V_{\max} + 1$. $[A1^*]$, which is the output of the first dose-response, serves as an input for the second dose-response (right). The second dose-response curve describes the $[A1^*]$ dependent production of $[X^*]$. The transfer function of the two dose-response curves is said to be balanced or matched if K_{XY} is within the bounds $[A1^*]$. (B) Diagram showing the values of parameter K_{XY} from the second dose-response which generates unmatched (red), borderline matched (yellow), and matched (green) transfer functions. These parameters are with respect to V_{\max} from the first dose-response. (C) Dose-response curves are generated with parameters from the three regions in (B). Matched dose-response curves (green) are the most sensitive to inducer, showing the biggest dynamic range.

Mathematically, to ensure a transfer function is matched, the K_{XY}^* of the second dose-response producing X needs to be within the dynamic range of the dose-response 1 producing $A1$ as seen in Fig. 3.5A. The lower bound is the steady state $[X^*]_{ss0}$ when the gene is completely turned off, $\theta = 0$. The upper bound is the steady state $[X^*]_{ss1}$ when the gene is completely turned on, $\theta = 1$. The two bounds can be obtained by replacing θ by 0 or 1 and finding the steady state where the derivative is zero so

$$[X^*]_{ss0} = 1; \quad [X^*]_{ss1} = 1 + V_{\max}^* \quad (3.29)$$

In the dimensionless model, both $[X^*]$ and K_{XY}^* are unitless (Table 3.4) and can therefore be compared. The transfer function of $A1$ to X is said to be matched with a downstream component if K_{XY}^* is between the upper and lower bounds as

$$1 \leq K_{XY}^* \leq (1 + V_X^*) \quad (3.30)$$

In Fig. 3.5B, the green region corresponds to those parameter ranges. Additionally, in Fig. 3.5C, the green curves correspond to dose-response curves where the transfer function is matched. The closer K_{XY}^* is to these upper and lower bounds, the less balanced the upstream component is with the downstream component. For example, with K_{XY}^* smaller than 1, given that $[X^*]$ can never approach a value smaller

than its basal level of 1 at steady state, $[X^*]$ would always be high enough to cause near-maximal regulation of the downstream component. On the other hand, with K_{XY}^* larger than $(1 + V_X^*)$, $[X^*]$ could never become high enough to cause near-maximal regulation of the downstream component. Other two ranges are defined which are borderline matched and unmatched (Fig. 3.5B). As seen in Fig. 3.5C, borderline curves are almost not affected by the inducer and unmatched curves are not affected at all.

The parameter space defined in the previous chapter is divided into these three categories (matched, borderline and unmatched). Linear stability analysis was carried out on the three categories to understand if matching transfer functions would increase the volume of Turing parameter space. Just by matching the transfer functions, the robustness goes up by 20-fold which is a significant increase (see Fig. 3.6) compared to non-balanced circuits.

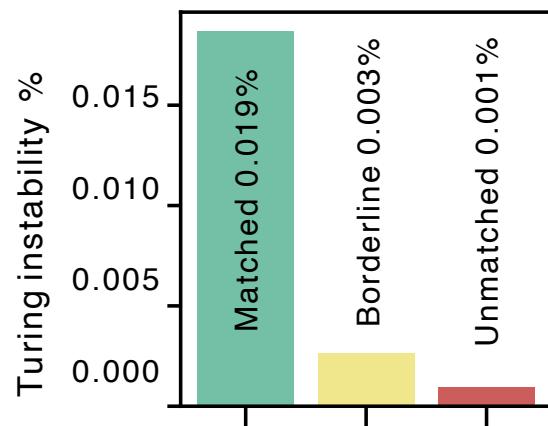


Figure 3.6: Turing robustness as a percentage of parameter space with respect to matching of dose-response curves. Turing instabilities include Turing I and Turing I-Hopf. Robustness from matched to unmatched increases 20-fold.

Following these insights, the transfer functions of the circuit components were matched by tuning the plasmid copy number, the strength of ribosome binding sites (RBSs), start codons and degradation tags (Andersen et al. 1998; Hecht et al. 2017; Wang et al. 2011) by Dr. Jure Tica and Tong Zhu. Transfer function matching yielded a well-functioning circuit with a better capacity to produce spatial patterns as discovered in this modelling study. The matched dose-response curves can be seen in the next section in Fig. 3.9.

3.2.4 Parameter fine-tuning of the gene circuit to increase the likelihood of Turing patterns

Another way of increasing the Turing parameter space is to tune parameters independently. In this section, we explore the distribution of obtained Turing parameter sets to understand how to tune each one to increase the robustness of our system.

Using the literature distribution defined above with only matched parameter sets, we perform linear stability analysis to find Turing parameter sets. In Fig. 3.7, we can observe how the distribution of Turing samples differs from the general sampling for each dimensionless parameter of the model.

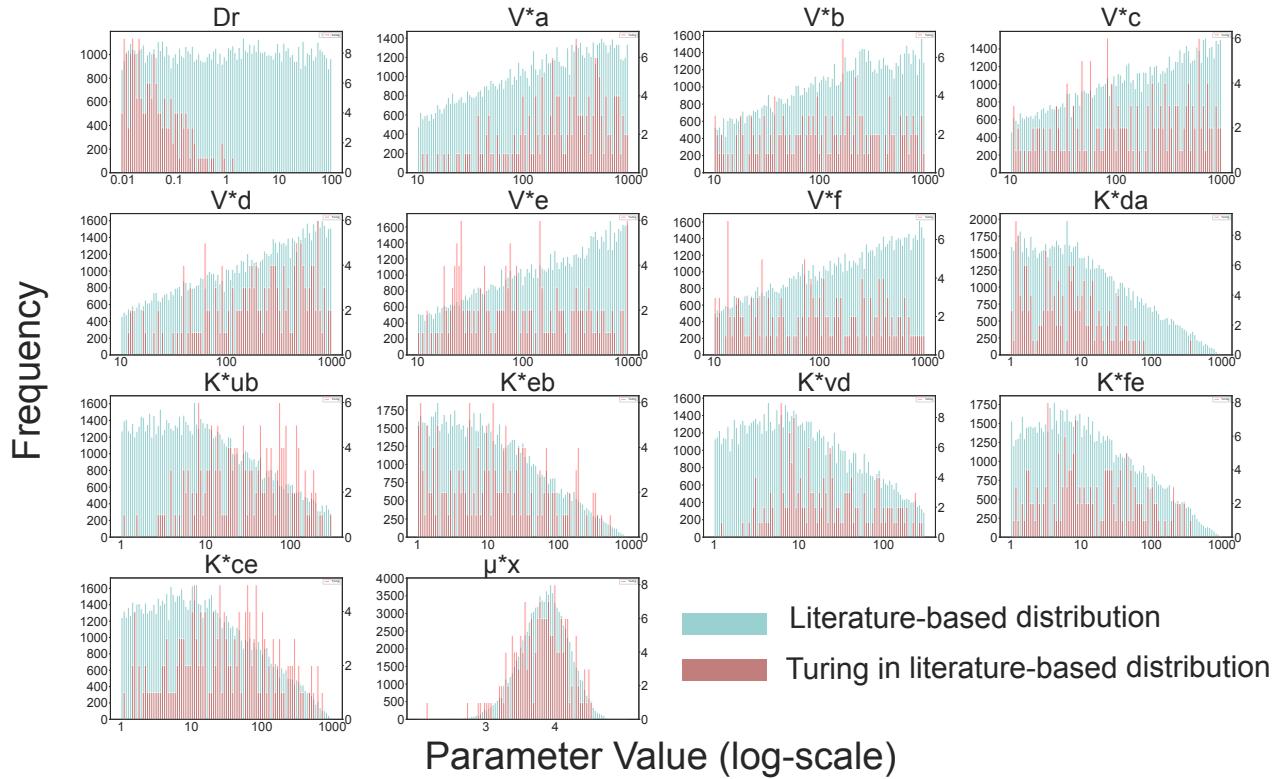


Figure 3.7: Broad literature-based distributions for model parameters Distributions for dimensionless parameters of the model based on literature ranges. Blue distributions correspond to the sampled literature-informed distributions, where we made sure to only consider parameters where all the circuit transfer functions are matched. Red distributions correspond to Turing instabilities found in the blue distributions using linear stability analysis. Turing instabilities include Turing I and Turing I-Hopf. n's are ignored because they are fixed.

Turing pattern systems appear to have a bias for certain regions of parameters (Fig. 3.7 red). This is very clear for diffusion, where Turing patterns usually appear in regions with $D_r < 1$. D_r is (D_V/D_U) meaning OC_{14} should diffuse slower than pC. Experimentally, the D_{OC14}/D_{pC} ratio has been measured to show it is 0.25 (unpublished work, Tica et al. 2023b). This means our experimental setup has a diffusion ratio which is favourable for Turing pattern formation.

Additionally, used the model to study how to fine-tune the circuit using the exogenous tuning molecules IPTG, aTc and DAPG to increase Turing pattern likelihood. K^*_{da} and K^*_{ce} are dependent on IPTG and aTc respectively, through an increasing function, as seen in Eq. 3.24a and Eq. 3.24b. Therefore, to understand the effects of these two exogenous tuning molecules we can look at their respective K parameters.

K_{da}^* Turing parameters have a similar distribution to the sampled distribution, implying that IPTG does not affect the robustness of the gene circuit to Turing pattern formation. On the other hand, the K_{ce}^* Turing parameters are more skewed to higher values than in the sampled distribution, suggesting that aTc increases robustness for Turing pattern formation. This is further shown by a more extensive sampling of the same distribution with five different K_{ce}^* values and measuring the Turing robustness in each (Fig. 3.8)

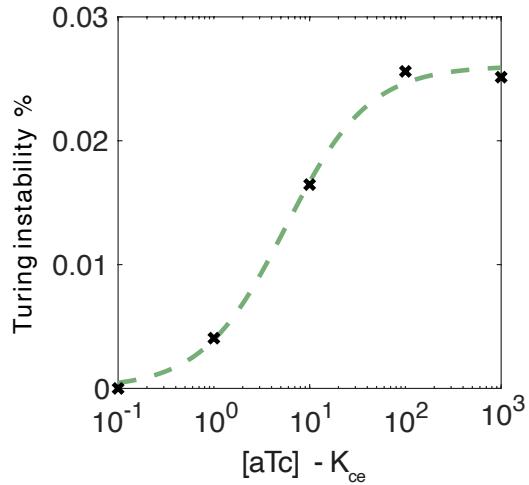


Figure 3.8: **Turing I robustness as a function of aTc or K_{ce} .** The highest Turing robustness is 0.025 with a high K_{ce} value of 10^2 . After this, robustness plateaus or decays slightly.

Finally, DAPG increases diffusor degradation by increasing μ_u and μ_v , however, these parameters are hidden in the dimensionless model, where the diffusor equations were integrated into the protein equations. As seen in Eqs 3.27b,d,f; μ_u and μ_v have a positively linear relationship with K_{ub}^* and K_{vd}^* , respectively. Both K_{ub}^* and K_{vd}^* have skewed distributions towards higher values compared to the sampled distribution (Fig. 3.7). This means that adding exogenous DAPG could also increase the robustness of the circuit for pattern formation.

In this section, we have obtained insights into parameter tuning to understand how to optimise robustness for Turing pattern formation. Following guidance from the model, experimentalists went on to match the transfer functions of the circuit components and add high levels of aTc. Before testing the circuit for patterning, experimentalists tested the behaviour of the circuit in liquid culture to make sure it behaved according to the design, and that the transfer functions were matching. Because the full circuit contains many closed-feedback loops, full-circuit liquid culture results are hard to interpret. Hence, subcircuit controls with no closed feedback loops were investigated instead to test whether the circuit was well-matched and behaved as expected. The conditions tested were under high aTc. Some relevant subcircuits tested can be seen in Fig. 3.9 left. The three subcircuits show how the

dose-response curves are within the responsive ranges, as the green and red curves act according to the other (Fig. 3.9 right). This means meaning the K and V parameters are within the matched region. Subcircuit #2 shows how the circuit is responsive to aTc which means aTc can be used for optimising Turing pattern robustness.

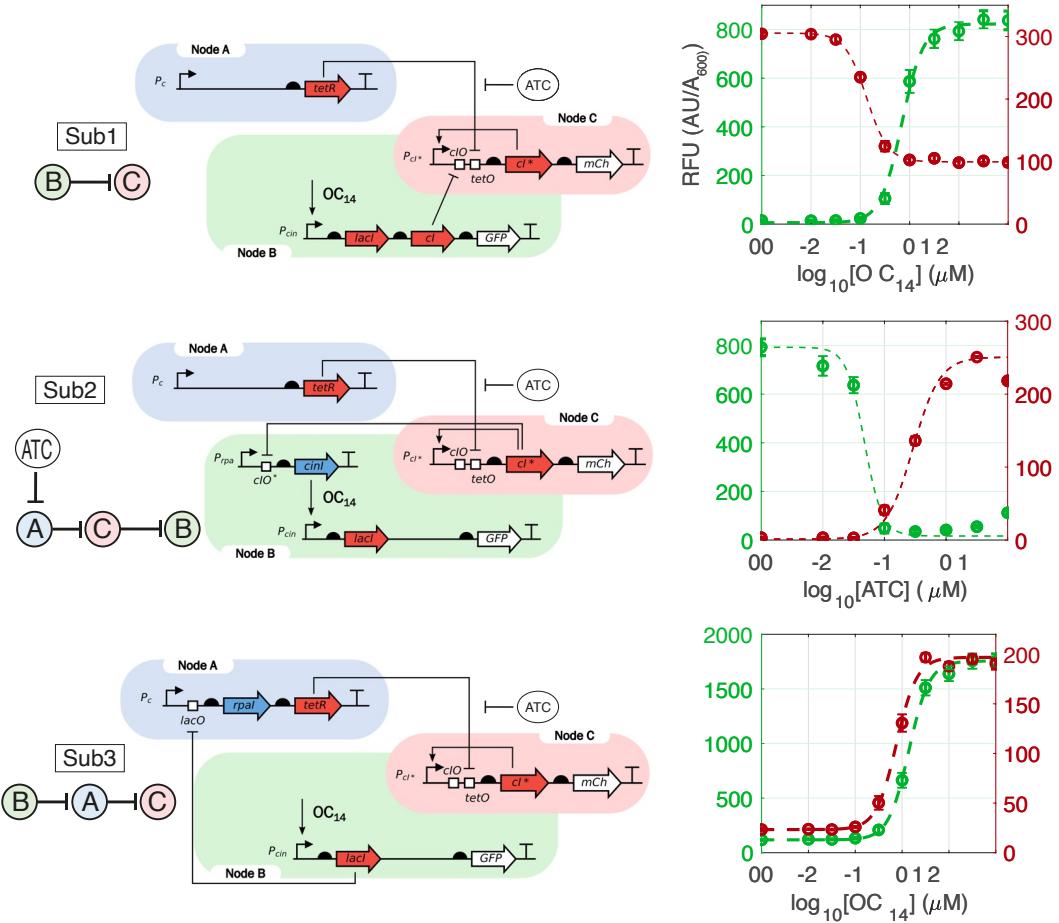


Figure 3.9: dose-response curves of subcircuits. On the left, the three subcircuits Sub1, Sub2 and Sub3 are shown, which test interactions of the circuit. On the right, the dose-response curves correspond to each subcircuit produced by Dr. Jure Tica and Dr. Georg Wachter. Each curve shows liquid culture fluorescence in RFU units (y-axis) 18 hours after induction with different levels of inducer (x-axis). GFP on the left and mCherry on the right axis (unit AU / A₆₀₀, mean \pm SEM, n = 3). We can see all dose-response curves are matched as in Sub1 and Sub3, red responds to the green and in Sub2 green responds to the red.

3.3 Constrained parametrised distributions: fitting to liquid culture data of gene subcircuits

The dose-response curves obtained (Fig. 3.9 left) are not only useful to characterise the circuit, but also for the parametrisation of the model. Using the dimensionless model, the fluorescence dose-response curves #1 and #3 can be fitted to obtain values for K_{XY} and V_X using a multivariate analysis approach. This is possible because

the non-dimensionalisation facilitates the comparison between the model and experimental dose-response curves with simple and intuitive methods.

3.3.1 Steady-state subcircuit equations for fitting.

As previously seen in Eq.3.29, the non-dimensionalisation transforms the dose-response range so it goes from 1 to $V_X + 1$. This transformation can be seen in Fig. 3.10, right. For the experimental data to match the model, it is divided by the smallest fluorescence value within each experiment and is expressed in relative fold-change units (Fig. 3.10, left). Fold-change units can take values from 1 to F_{\max}/F_{\min} (maximal and minimal fluorescence levels in the original, untransformed dataset, respectively). Because the dose-response curves of subcircuit #1 and subcircuit #3 are OC14-dependent, the experimental OC14 units (μM) are also non-dimensionalised using the transform

$$[B^*] = [OC_{14}] \cdot \frac{\mu_b \mu_v}{k_2 b_B} \quad (3.31)$$

Now the dimensionless model dose-response curves and experimental dose-response curves are both expressed on relative scales and are compatible for fitting (Fig. 3.10, bottom).

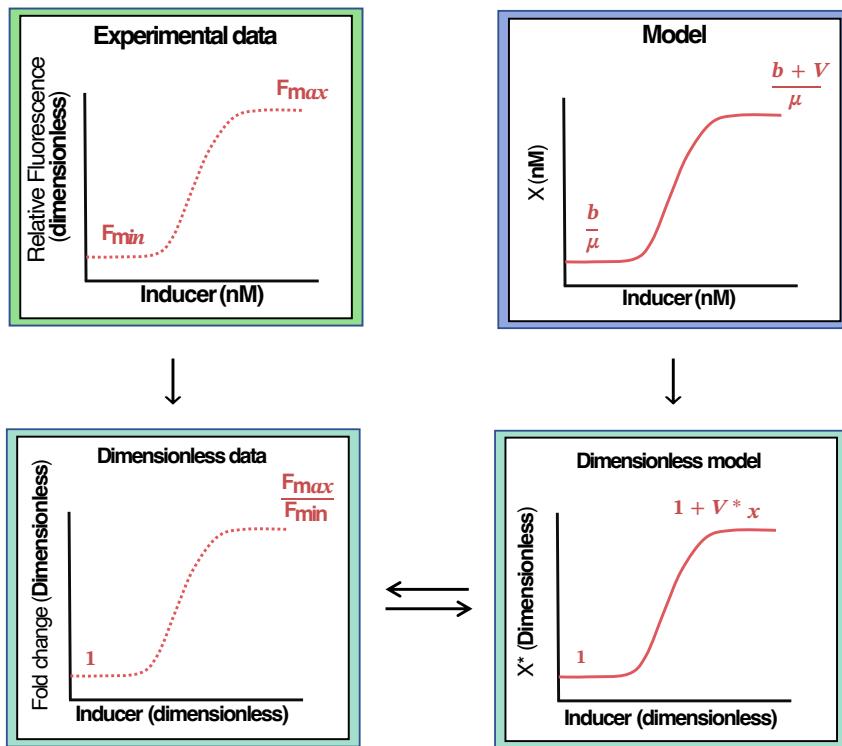


Figure 3.10: Experimental data and model transformation for parametrisation. Experimental data is scaled by the smallest fluorescence value, so the minimum value is 1 (left). The model is nondimensionalised as explained in the sections above so the smallest value is 1 and biggest is $V_X + 1$ (right). In both cases, units are dimensionless, and the basal level is 1 (bottom).

The models for the subcircuits are derived from the main PDE system (Eqs. 3.27). Subcircuit #1 only involves species [F] and [E] (cI and cI*), whereas subcircuit #3 involves species [C], [D] and [E] (TetR, LacI and cI*). Detailed diagrams of these two subcircuits can be found in Fig. 3.9 left. All other species are set to zero. For the model parametrisation, based on the subcircuit designs, we derive steady-state expressions for the dynamically regulated species so that

$$\frac{\partial X}{\partial t} = 0; \quad X = X_{eq} \quad (3.32)$$

For Subcircuit #1 we obtain the following steady-state expressions:

$$F_1 = 1 + V_f \left(\frac{1}{1 + \left(\frac{\mu_v K_{vd}}{k_v [O_{C14}]} \right)^{n_{vd}}} \right) \quad (3.33a)$$

$$E_1 = 1 + V_e \left(\frac{1}{1 + \left(\frac{F_1}{K_{fe}} \right)^{n_{fe}}} \right) \quad (3.33b)$$

For Subcircuit #3 we obtain the following steady-state expressions:

$$D_3 = 1 + V_d \left(\frac{1}{1 + \left(\frac{\mu_v K_{vd}}{k_v [O_{C14}]} \right)^{n_{vd}}} \right) \quad (3.34a)$$

$$C_3 = 1 + V_c \left(\frac{1}{1 + \left(\frac{D_3}{K_{da}} \right)^{n_{da}}} \right) \quad (3.34b)$$

$$E_3 = 1 + V_e \left(\frac{1}{1 + \left(\frac{C_3}{K_{ce}} \right)^{n_{ce}}} \right) \quad (3.34c)$$

3.3.2 Fitting process and the resulting best-fit distributions.

In addition to scaling and nondimensionalising, the lowest GFP data points were excluded, because fluorescence readings were insufficiently sensitive to measure concentration at these points (Fig. 3.9). This improved the quality of the fit and allowed us to identify a broader range of suitable solutions.

The two-equation systems Eqs. 3.33,3.34 are fitted independently to the experimental dataset with the python `scipy.optimize.curve_fit` package, which uses the Levenberg-Marquardt to minimise the sum of squared errors (SSE), given by

$$SSE = \sum_{i=1}^i (y_i - f(x_i))^2 \quad (3.35)$$

where y_i is the experimental data and $f(x_i)$ are the two systems of equations parameterised by V_C^* , V_D^* , V_{E1}^* , V_{E3}^* , V_F^* , K_{vd}^* , V_{fe}^* , V_{da}^* and V_{ce}^* . The minimisation algorithm generates a vector of best-fit parameters k

V_C^*	V_D^*	V_{E1}^*	V_{E3}^*	V_F^*	K_{vd}^*	V_{fe}^*	V_{da}^*	V_{ce}^*
9.95	6.50	1.99	7.8	3.64	18.94	2.26	67.92	3.47

Table 3.6: Vector of best fit parameters k

Two best-fit parameters are obtained for V_E^* as this parameter is present in both subcircuit models. However, only V_{E1}^* from subcircuit #1 is considered. These parameters are used to generate the following dose-response curves (Fig. 3.11).

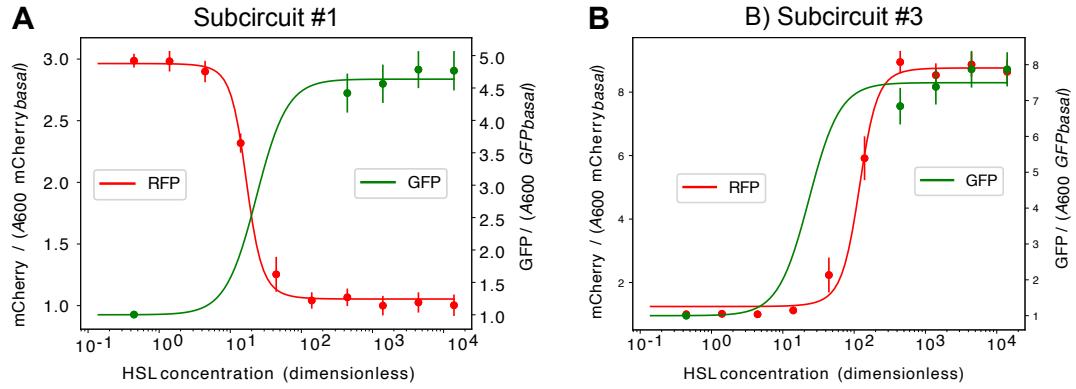


Figure 3.11: Best fit for processed data set. Processed dataset is fitted using (A) Eq. 3.33 for subcircuit #1 and (B) Eq. 3.34 for subcircuit #3.

The minimisation algorithm produces a covariance matrix C_k , which is the inverse of the Hessian matrix and represents the derivative of the loss function in the different parameter dimensions. In other words, this Hessian matrix represents how the loss increases or decreases when parameters are varied together

$$H_{L_k} = \begin{bmatrix} \frac{\partial^2 L}{\partial k_1^2} & \frac{\partial^2 L}{\partial k_1 \partial k_2} & \dots & \frac{\partial^2 L}{\partial k_1 \partial k_n} \\ \frac{\partial^2 L}{\partial k_2 \partial k_1} & \frac{\partial^2 L}{\partial k_2^2} & \dots & \frac{\partial^2 L}{\partial k_2 \partial k_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 L}{\partial k_n \partial k_1} & \frac{\partial^2 L}{\partial k_n \partial k_2} & \dots & \frac{\partial^2 L}{\partial k_n^2} \end{bmatrix} \quad (3.36)$$

A multivariate Gaussian distribution is generated using k and C_k

$$X \approx N(k, C_k) \quad (3.37)$$

with a probability density function $p(x; k, C_k)$

$$p(x; k, C_k) = \frac{\exp\left(-\frac{1}{2}(x - k)^\top C_k^{-1}(x - k)\right)}{\sqrt{(2\pi)^k C_k}} \quad (3.38)$$

The multivariate Gaussian distribution is the generalization of a normal distribution to higher dimensions. For example, for 2-dimensional Gaussian distributions, when the covariance of two parameters X and Y is positive, the parameters are positively correlated (Fig. 3.12, right). This means that if the parameters are increased together, the behaviour of the system should change minimally (and the error to the data should not increase). On the other hand, a negative covariance leads to an inverse correlation of the parameters, meaning when one increases the other should decrease to ensure the error does not increase (Fig. 3.12, left). Finally, a covariance of zero means the X and Y parameters are completely independent, producing a circular distribution (Fig. 3.12, center).

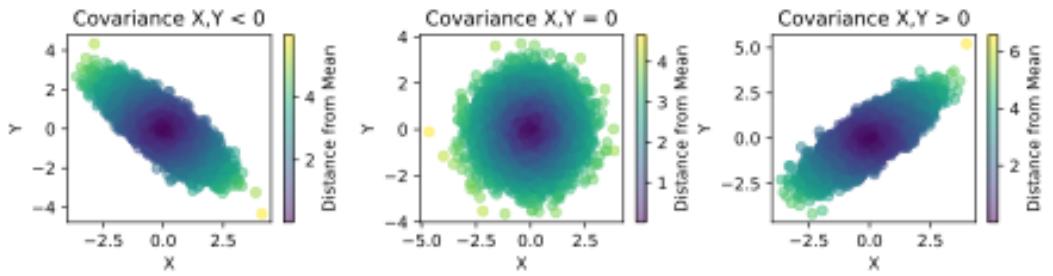


Figure 3.12: Multivariate Gaussian distributions. Pairplots of X and Y parameter distributions with (left) negative covariance, (centre) zero covariance and (right) positive covariance. The colour represents the distance to the mean, or in other words to the best-fit parameter (k). Blue are the parameter (k) with minimal loss and yellow are neighbouring parameters with bigger loss.

The multivariate Gaussian distribution generated from the vector of best-fit parameters k and the covariance matrix C_k can be seen as a pairplot in Fig. 3.13. Positive, negative and no correlations were observed in these distributions. This figure shows how certain parameters have positive covariance such as V_c^* and K_{ce}^* . A high V_c^* indicates a strong production of TetR, while a high K_{ce}^* indicates a higher concentration of TetR to have the same inhibition effects. Therefore, if both are increased together, the behaviour of the system should remain the same and the dose-response fit should be unaltered. Some parameters have a negative covariance such as K_{vd}^* and K_{fe}^* . While a high K_{vd}^* increases the production of cI, a low K_{fe}^* decreases the amount of cI needed to have the same inhibition effects. Therefore if varied in opposite directions, the circuit behaviour remains unaltered. Others are independent like V_d^* and K_f^* which determine the production rates of the molecules which affect the circuit independently.

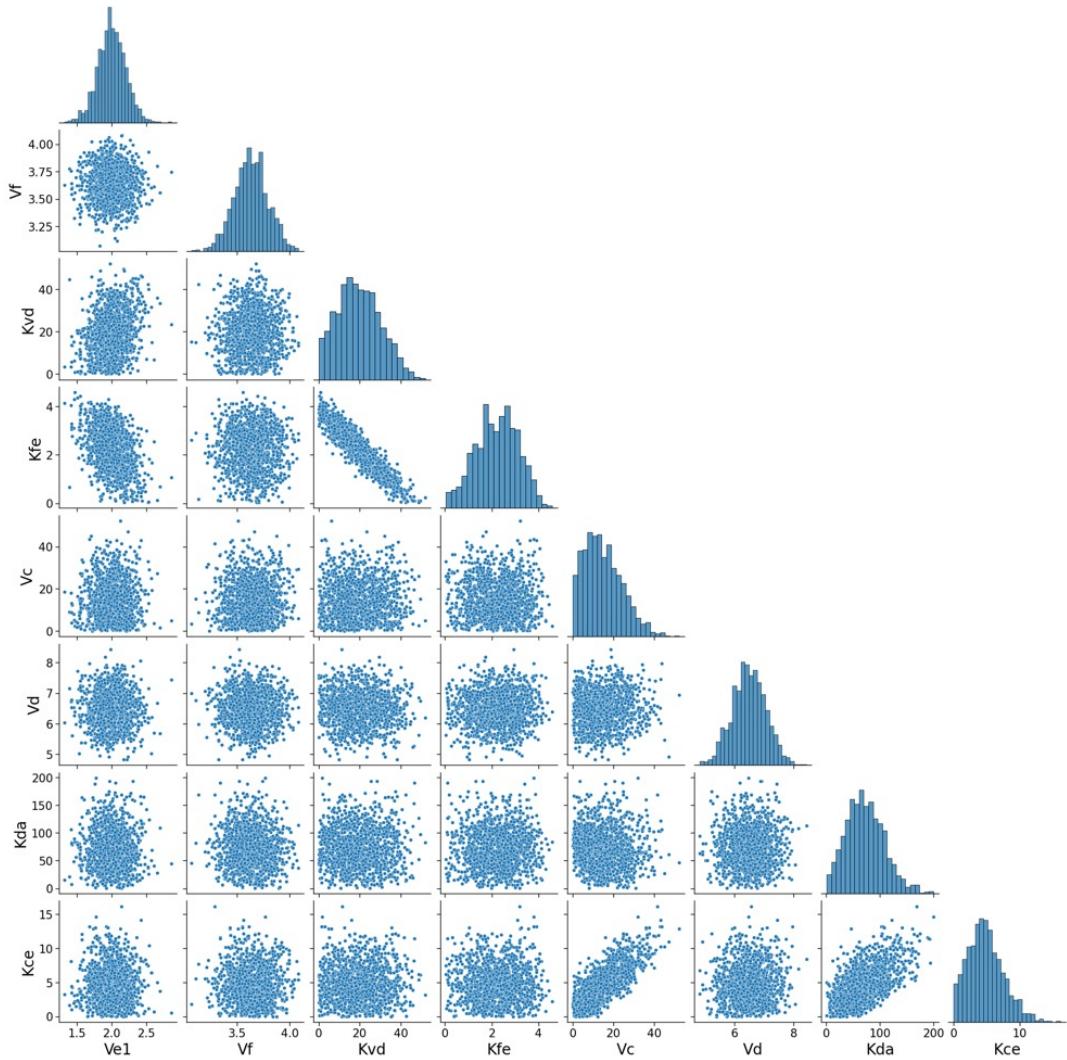


Figure 3.13: Multivariate Gaussian distributions for fitted parameters. Pairplot of distribution resulting from fitting subcircuit #1 and #3, using $q = 10$. This distribution has a probability density function $p(x; k, 10 \cdot C_k)$ where k is the best fit parameter vector and C_k is the covariance matrix. Diagonals represent the univariate distributions. The non-diagonals represent the multivariate distributions of parameter pairs, where we can observe positive and negative correlations as well as independent pairs of parameters. Negative parameters are removed from the distribution as these are biologically not possible.

The width of the multivariate Gaussian distributions can be increased by multiplying C_k by a scalar factor q where $q > 1$. This process unconstrains the fits and allows more error with respect to the experimental data. To search for Turing patterns close to the fitted parameter combination k , the value of q is progressively increased until a Turing parameter set is found through linear stability analysis. Through this process, the first 3 Turing parameter combinations are found with $q = 10$. These are the closest Turing solutions to the best fit parameter combinations. The dose-response curves generated by the $q = 10$ distribution are shown in Fig. 3.14. Within all the

curves shown, 3 of them are generated by Turing parameter sets. The three Turing parameter sets are ensured to be ‘balanced’, where the input/output relationships of the components are matched (see Section 3.2.3).

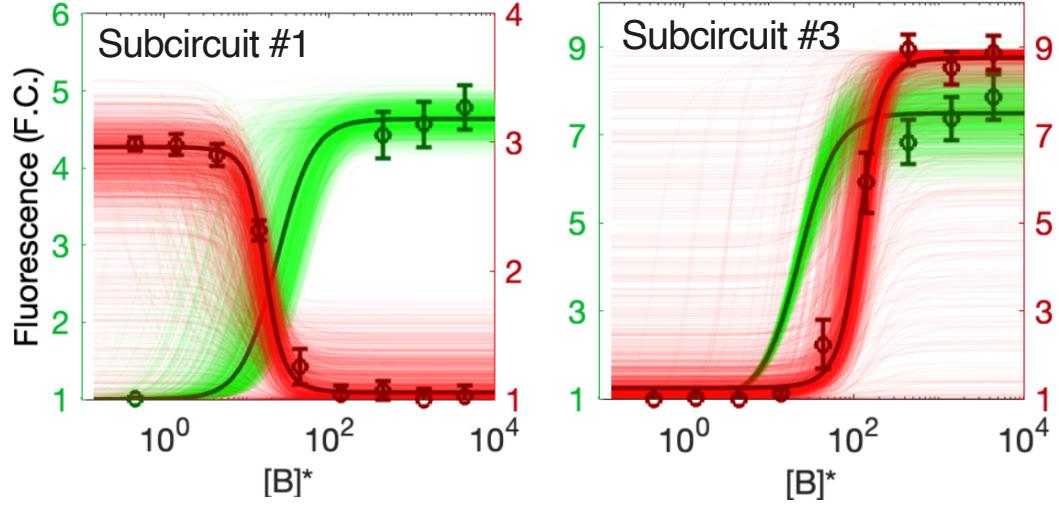


Figure 3.14: Fitted OC14 dose-response curves produced using the multivariate Gaussian distributions. Dots show experimental data, the thick line is generated from best fit parameters and the thin lines are derived from multivariate Gaussian distributions centred around the best fit with $q = 10$. The parameters used to generate these curves come from the multivariate analysis optimisation, for subcircuit 1 (left) and subcircuit 3 (right). The parameter distributions are shown in Fig. 3.13

The corresponding $q = 10$ multivariate Gaussian parameter distributions are shown in Fig. 3.13. The distributions of the multivariate Gaussian in 1-parameter dimension and the Turing parameter sets are shown in Fig. 3.15. This figure shows where the parameters of Turing solutions lie in comparison with the fitted distributions. Overall, only parameter K_{da}^* is far off from the Turing solutions. Lower values of K_{da}^* would be required to increase the likelihood of finding Turing solutions. This K_{da}^* parameter, which is captured in Subcircuit #3 (see Eq. 3.34b), has the highest uncertainty be

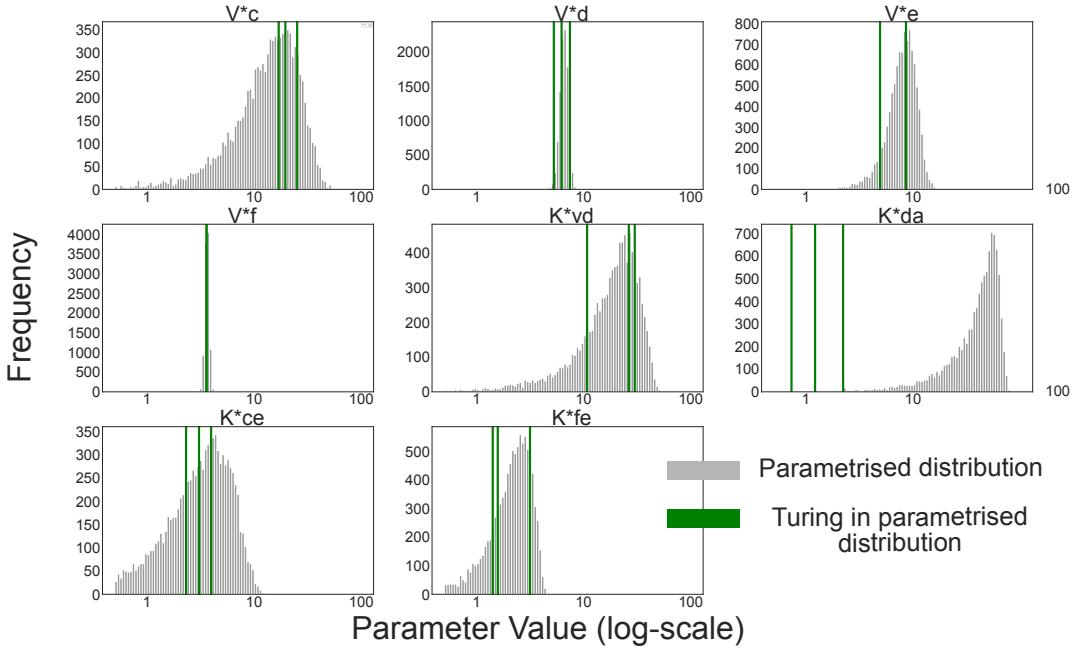


Figure 3.15: Constrained parametrised distributions for V_X^* and K_{XY}^* parameters in 1 dimension. Distributions for dimensionless parameters based on fitting to liquid-culture experimental data. Grey distribution corresponds to parameters from fitting with $q = 10$. Green vertical lines correspond to the 3 Turing parameter sets found in the grey distribution using linear stability analysis. Parameters which generate non-matched dose-response curves are removed.

3.4 Discussion

A synthetic gene circuit was built with the aim of engineering periodic pattern formation in synthetic *E. coli* biofilms (Fig. 3.1). This chapter described the model built to understand the spatio-temporal dynamics of this synthetic gene circuit. The parameter space was explored to show the circuit can produce Turing patterns, and in certain regions, the likelihood of finding these patterns is higher. Finally, the model was parametrised using liquid culture data to constrain the model parameters to experimentally realistic values.

3.4.1 Model of synthetic gene circuit #1754

The model presented was a PDE system which describes the concentrations of the six proteins in space and time. This model gives us insights into how protein concentrations change over time and in space and whether any spatio-temporal patterns form. Because the experimental circuit is the largest reaction-diffusion (RD) gene circuit ever built, the model is highly complex with many variables and parameters. This results in computationally expensive numerical simulations and a high-dimensional parameter space with over 20 dimensions, which is complicated to sample.

To simplify the model, many assumptions were made. For example, quasi-steady states were assumed for mRNA production, protein-promoter binding, receptor-diffusor binding and diffusor synthesis. Although these processes exist in a quasi-steady state equilibrium, they might produce delays which are not taken into account in this model. Investigating these delays would be necessary in the future to understand whether they disrupt pattern formation as shown in Maini et al. 2012. Additionally, degradation was assumed to be linear, which might not be accurate in some cases. High levels of proteins, burden or age-dependent effects might collapse the degradation machinery (Kintaka et al. 2020) leading to non-linear dynamics which are not currently captured. Additionally, diffusor precursors as well as diffusor receptors were assumed to be in excess to avoid extra equations and parameters. The parameters of the model were also assumed to be constant over time, which might not be realistic due to cell burden, nutrient depletion or other effects in the biofilm. Future work in this direction would involve exploring experimentally the relationship between these effects and the model parameters. An interesting example of this would be to define a time-dependent nutrient equation on which kinetic parameters are dependent. Finally, this model assumed fluorescent protein concentration is linearly correlated with fluorescence. Although this assumption is often correct (Csibra and Stan 2022; Soboleski et al. 2005), several factors can disrupt the linearity. For example, low concentrations of fluorescent protein might not be detected by the instruments, leading to a non-linear relationship. pH and temperature can also affect fluorescent values (Ward et al. 1982). When using LacI and cI* to model GFP and RFP fluorescence respectively, we need to take into account these assumptions on linearity and understand when they might break.

3.4.2 Finding Turing instabilities and optimising Turing robustness

Regardless of all the assumptions, the model was extremely useful in giving us insights into the system. Firstly, the model parameter space was searched using linear stability analysis to find Turing I instabilities. Turing I-Hopf instabilities were also searched for, as they have been shown to produce stationary periodic patterns in many cases as seen in Section 2.4.2. The model did produce Turing I and Turing I-Hopf instabilities, meaning periodic stationary patterns as the ones observed in Fig. 3.4 could occur experimentally. When the gene circuit was built inspired by the original #1754 topology, more nodes and connections had to be implemented to replicate the original functions. Therefore, it was unclear whether these changes would remove the capabilities of patterning. Proving the genetic implementation of this topology could produce patterning was necessary before searching for Turing patterns experimentally.

The model also helped understand parameter regimes in which the gene circuit was more likely to produce Turing instabilities, including both Turing and Turing I-Hopf.

By matching circuit components to have responsive dose-response curves, the robustness could be increased 20-fold from 0.001% to 0.19% as seen in Fig. 3.6. This robustness could be further increased by tuning specific parameters such as decreasing the D_{OC14}/D_{pC} ratio, increasing aTc or increasing DAPG. Different levels of aTc were explored in a system with matched dose-response curves. It was seen that robustness could be increased from 0 to 0.025 by increasing K_{ce} , which is equivalent to adding exogenous aTc to the experiment.

These results guided the experimental fine-tuning of the circuit by matching dose-response curves and adding exogenous aTc and DAPG. Decreasing D_r is harder, as there is no current good method for controlling the diffusion of quorum sensing molecules. A potential route of decreasing the diffusion rate of a quorum-sensing molecule is through receptor sequestering: When the number of receptors is much larger than the number of quorum sensing ligands, there is an irreversible uptake of some molecules which has been linked to shorter range communication (i.e. slower diffusion constants (Gestel et al. 2021)). Therefore, to decrease D_r , we can slow down the diffusion of OC_{14} by increasing the expression of the CinR receptor which will bind to OC_{14} . This can be done by tuning the RBS for higher expression.

Overall, this model enabled us to design experimental strategies for tuning the robustness of the system so Turing patterns could be obtained experimentally. However, it is important to note that although they have been optimised, the robustness levels obtained are still extremely low. Further investigation to understand how to further increase the percentage of Turing solutions in parameter space is needed. In this thesis, robustness optimisation was carried out by tuning model parameters. However, it is worth investigating whether growth, boundary conditions or even a pre-pattern could increase the probability of obtaining Turing patterns with this circuit.

3.4.3 Model parametrisation

Working with large parameter spaces based on literature parameters is useful for understanding the overall potential of the circuit architecture. However, once the system is fine-tuned by matching dose-response curves and adding exogenous aTc, it is important to constrain the parameter space to that of the fine-tuned circuit, so more accurate results are produced. Additionally, because of the high dimensionality of the parameter space, it is useful to parametrise the model to obtain a smaller parameter space which is easier to sample. Dose-response curves from the fine-tuned circuit in liquid culture were produced which could be used for parametrisation. Parametrisation was possible because of the dimensionless model. This dimensionless model could produce dose-response curves such as the experimental ones by taking the steady state expressions under different inducer concentrations (Eqs 3.33, 3.34). Relative fluorescent units from the experimental dose-response

curves could be directly compared to dimensionless units of our dimensionless model as seen in Fig. 3.10.

It is important to note that the experimental data used for parametrisation represents the behaviour of cells in liquid culture. Parameters might vary slightly for cells growing in colonies on agar surfaces, especially as they become old. Dose-response curves in agar could be produced in the future. This would allow us not only to understand the parameters of cells growing in agar dishes, but also have different fits for old cells in the middle and new cells at the edge of the colony.

By fitting the dose-response models to the experimental liquid culture data, we aimed to obtain a distribution of model parameters that replicated the experimental data with a certain level of uncertainty. A distribution is preferred over a single fit as biological systems exhibit noise and there is a certain level of uncertainty in the fixed parameters of the model. This distribution could be obtained by using a Bayesian approach or a multivariate Gaussian optimisation approach. With Approximate Bayesian Computation, a prior is given which is updated by sampling the parameter space to obtain a posterior distribution that fits the data. This Bayesian approach works well if prior knowledge needs to be incorporated into the model and if non-linear relationships between parameters are present. However, it is more computationally expensive. On the other hand, multivariate Gaussian optimisation is a much more computationally efficient algorithm based on a simple least squares minimisation to produce a mean value and a covariance matrix. This algorithm is less flexible in terms of the prior given and the relationships between parameters (i.e. non-linearities are not captured). Additionally, the distribution is only centred around the best fit mean value, meaning the posterior might be limited.

Using the multivariate Gaussian approach, a distribution of parameters was obtained which fitted the experimental dose-response with a certain level of uncertainty. This allowed for a much more constrained parameter space which is easier to sample and gives more accurate results. Additionally, the fitting process allowed us to understand and verify correlations between parameters. All these correlations can be explained using the logic of the circuit architecture, which means the distributions are correct and the model accurately describes the dynamical behaviour of the circuit. Interestingly, Turing solutions were found within these distributions. Theoretically, these Turing solutions are the closest solutions in parameter space that fit the experimental data. Therefore, we should expect their dynamic behaviour to match the experiments more closely than any other Turing solution found in the literature-based parameter space.

However, a high level of uncertainty ($q = 10$) was needed to find such Turing solutions in the distribution. This lack of robustness in the fitted distribution can be attributed mainly to the K_{da}^* parameter, which has a higher distribution than Turing solutions. This K_{da}^* , which is the $[LacI]$ needed for half repression of node A,

is an uncertain parameter as no fluorescent reporters are linked to node A. Currently, the only data used is on node B, with a GFP reporter, and node C, with an mCherry reporter. Having dose-response curves which report on every gene of the circuit would get rid of issues on node A parametrisation, and potentially increase the Turing robustness in the fitted distribution. A potential route to improve this parametrisation technique would be to use RT-qPCR, which is a technique used for mRNA quantification. mRNA counts of every one of the six genes would be measured for different inducer concentrations, hence obtaining information on the currently 'hidden' nodes. Additionally, because fluorescence levels are sometimes not an accurate predictor of protein levels as previously discussed, using mRNA counts would improve issues such as the low fluorescent values not being detected. Overall, this parametrisation method goes beyond studying pattern formation and could be useful to parametrise any genetic circuit model in synthetic biology.

CHAPTER 4

Patterning in Synthetic Bacterial Colonies

Using the insights obtained in the previous chapter, a wide range of microscopy experiments were carried out in optimal Turing conditions to obtain patterning in *E. coli* biofilms. The aim is to observe fluorescent periodic patterns when these biofilms are observed under the microscope. The model is key to elucidate which mechanisms are behind the patterns formed, and specifically to investigate if periodic patterns obtained are a result of Turing's mechanism.

In this section, experimental results produced in this thesis are shown. Further microscopy results done by researchers at the Isalan Lab are also presented. A framework to numerically solve reaction-diffusion systems in shaped and growing biofilms is developed to compare model outputs and experimental results. Finally, perturbation experiments are carried out computationally, to predict experimental perturbations and elucidate the mechanisms behind pattern formation.

4.1 Experimental rings in small colonies with high aTc

In the previous chapter, certain conditions which could affect robustness for Turing pattern formation were found. As already shown, the circuit components were matched by Dr. Jure Tica and Tong Zhu (see Fig 3.9) to improve robustness as seen in Fig. 3.6. In this section, I investigate the circuit in a growing colony biofilm using confocal microscopy under the optimal Turing conditions obtained in Chapter 3. More specifically, I look at how the biofilm fluorescent patterns change with different levels of aTc in small colonies

MK01 *E. coli* cells were transformed using electroporation (Section 6.8.1) with the full circuit. This involved the introduction of 4 plasmids with the three nodes and

the regulator cassette as seen in Fig. 3.1 and Table 6.1 into our *E. coli* cells. For this, the cells were then plated in 6 well MatTek plates and grew as individual colonies which are radially growing biofilms. These growing colonies are then imaged using confocal microscopy. Confocal microscopy is a type of fluorescence microscopy used to image thicker objects, where the beam of light focuses on one depth level, meaning you can get a single z-plane of fluorescence. This way we can obtain a two-dimensional (2D) fluorescence pattern stemming from a single focal plane of the colony (Semwogerere and Weeks 2005). Detailed methods on colony plating and microscopy can be found in Section 6.8.2. Red and green channels were imaged to detect mCherry and GFP as seen in Fig. 4.1A,B. These channels can then be superposed to get a GFP-mCherry combined reading (Fig. 4.1C).

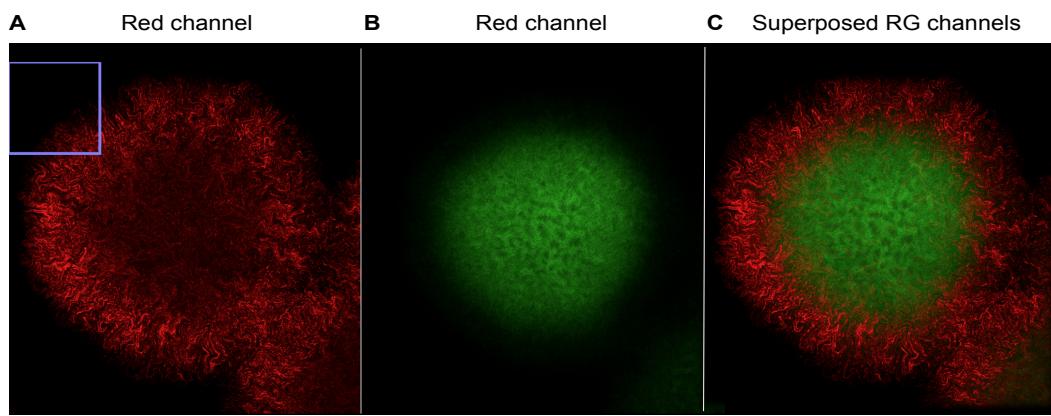


Figure 4.1: 2D plane of *E. coli* bacterial colony transformed with #1754 synthetic circuit. Image obtained with confocal microscopy. (A),(B) Red and green fluorescence channels with relative fluorescent units show the normalised concentrations of GFP and mCherry. (C) The superposed channel contains both red and green channels showing red at the edge and green at the centre.

To test the impact of aTc on the patterning of the circuit, we prepared the agar on the MaTek plates with different levels of aTc, ranging from 0 to $10^1 \mu M$. Different colony patterns arise from this aTc ‘walk’ as seen in Fig. 4.2A. The colonies exhibiting more spatially heterogeneous behaviour are those with high aTc ($10^1 \mu M$). This high aTc condition is further explored by imaging every day. In Fig. 4.2B, we see how over time, the center of the colony oscillates from black to red to green. As this happens, rings get added from the centre as in Konow et al. 2019’s interior ring growth mechanism. The final snapshot (64h) shows green, red, green, red progression starting from the centre. High DAPG conditions were also tested, as the model predicted this could also increase the likelihood of Turing patterns. However, unlike with aTc, no patterns were observed with a high DAPG concentration.

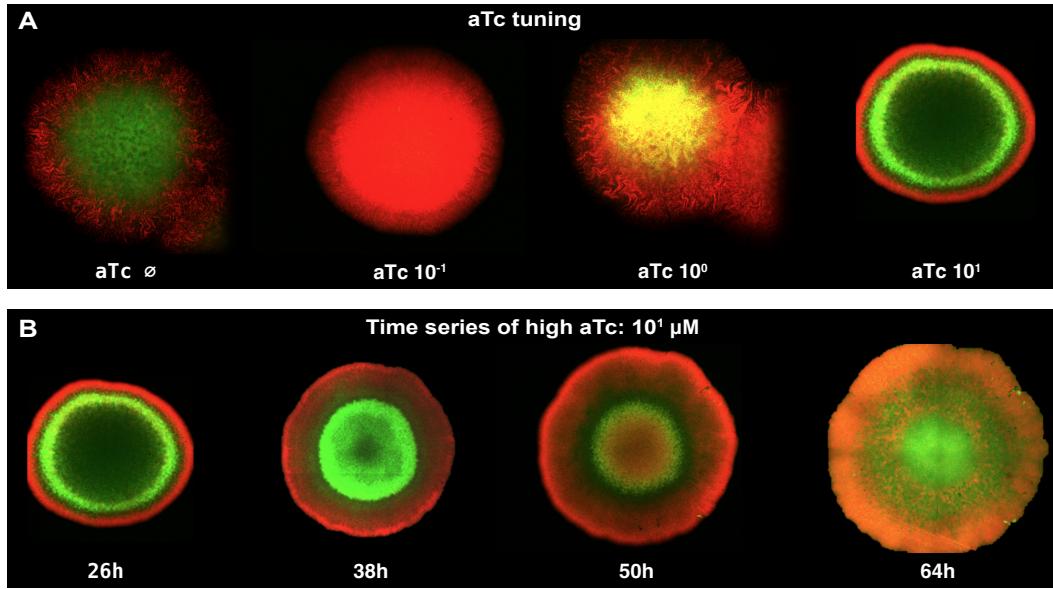


Figure 4.2: Confocal images of small colonies with gene circuit #1754. (A) Colonies with different aTc conditions at time 26h: no aTc $10^{-1} \mu M$, $10^0 \mu M$, $10^1 \mu M$. (B) Time series of single colony with high aTc condition ($10^1 \mu M$). Interior ring growth dynamics can be observed. The centre oscillates from black (26-38h) to red (50h) to green (64h). The black centre and the red centre become rings at the next time point.

Following this work, we continued studying this synthetic patterning system more in-depth. Two routes were followed. The first one involved specific shaped domains achieved by imprinting the agar with bacteria using shaped objects (Fig. 4.4). The aim was to understand how the pattern adapts under different shaped domains. The second route, and most explored, involved larger colonies to determine whether more repeats would form in a Turing-like behaviour. This was achieved by carefully diluting the sample and plating a single colony without any neighbours.

4.2 Modelling framework for synthetic circuit in bacterial tissues

Most theoretical studies which involve Turing patterns, numerically simulate their system using square domains with no-flux or periodic boundary conditions. However, these numerical domains are often not biologically realistic. For example, the system we have developed experimentally involves more specific conditions including shaped domains, stochastic growth and absorbing boundary conditions. To have a predictable model of our experimental system, the numerical solver had to be adapted to include such domain characteristics.

4.2.1 Alternating Direction Implicit Method with defined domains

All simulations in this chapter are performed in a 2D space to match the 2D focal plane captured by the confocal microscopy. For this purpose, the numerical solver schema Alternating Direction Implicit Method (ADI) is used. This numerical solver produces a 2D space solution in time for the n number of species of the model. This method is chosen over Crank-Nicolson (CN) used in the previous chapter, as it is more efficient to solve 2D problems due to the matrix diagonalisation (see Section 6.6). More specific details of ADI can be found in Section 6.6.2.

ADI is originally defined to solve square domains. To integrate our specific domains with the solver, a masking method is used where a "shape matrix" is passed containing the shape of the domain. The "shape matrix" is a Boolean matrix of $I \times J$ size which contains information on the location of the cells. 1's determine cells, while 0's determine agar. When passing this matrix to the solver, the algorithm computes reaction and diffusion terms in 1 positions while it only computes diffusion in 0's. Fig. 4.3 right shows the "shape matrix" where 1's are black and 0's are white. Additionally, this figure shows what functions are computed in which regions. How the "shape matrix" is defined depends on the experimental setup. Using this masking method, we will obtain a solution for the n number of species of our model, in time and space, within the biofilm.

$$\frac{\partial U}{\partial t} = f(U, \dots) + D_U \frac{\partial^2 U}{\partial x^2}$$

$$f(U, \dots)$$

- Basal production (b)
- Regulated production (V, K)
- Degradation (μ)

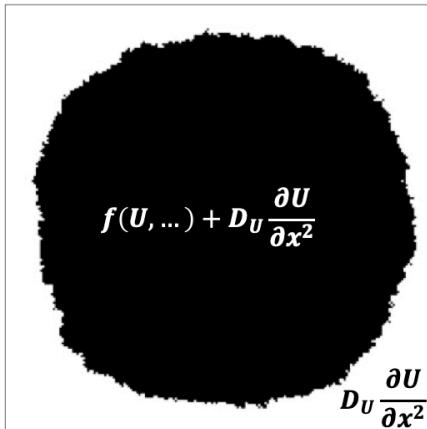


Figure 4.3: **Masking of PDEs with tissue shape.** Our PDEs are composed of reaction and diffusion terms (left). The masking method (right) consists of computing the different terms of the PDE in different regions of the system. A shape matrix (square with a black circle) determines where the reaction and diffusion terms are computed (black) and where only the diffusion term is computed (white).

4.2.2 Static Shaped domains

Following the small rings produced in this thesis, Tong Zhu from the Isalan Lab started experimenting with non-circular domains. Specific shapes were obtained by imprinting the agar with bacteria using shaped objects. Fig. 4.4A shows the resulting biofilm after impregnating the agar with bacteria using the edge of a glass coverslip.

In this thesis, the biofilm shape is replicated by using a "shape matrix" derived from the microscopy image Fig. 4.4B. This is done through image recognition on the microscopy snapshots to detect areas with cells (see Section 6.7.1).

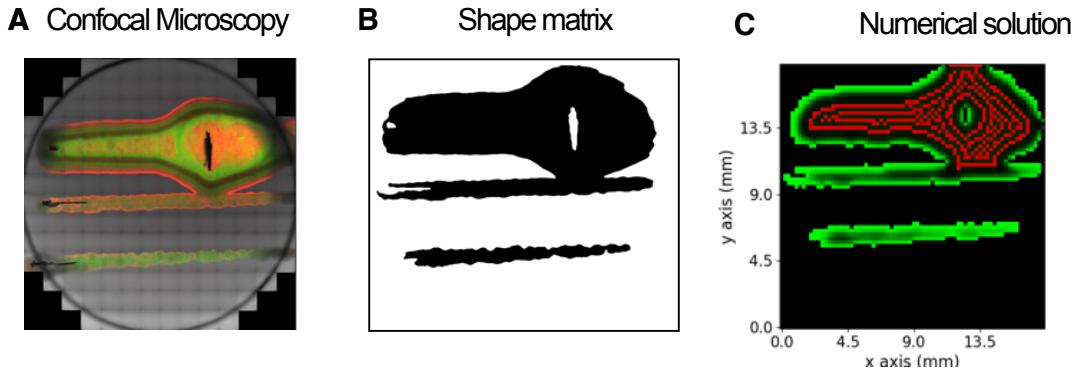


Figure 4.4: Image recognition of confocal microscopy image for PDE masking. (A) Confocal microscopy of three *E. coli* biofilms grown by seeding with the edge of a glass coverslip. This experiment was done by Tong Zhu. (B) Shape matrix obtained using image recognition on biofilm in (A) to detect cells. Black shows cells while white shows agar. (C) Numerical solution of six-equation model with a Turing parameter set, using the masking method with the shape matrix in (B).

Using the masking method with ADI explained in 4.2.1, a numerical solution was computed within the defined biofilm. The PDE system used is the six-equation model (Eq. 3.23), which describes synthetic circuit #1754, using a Turing parameter set found through linear stability analysis. The numerical solution is shown in Fig. 4.4C as a superposition of the red and green channels. Details on how to plot the numerical solution of the six-equation as a red-green image like confocal microscopy can be found in Section 6.7.2. The stripes obtained with the Turing model adapt to the shape of the tissue. Some stripes with the same adaptation to the tissue shape can be observed in the confocal microscopy experiment (Fig. 4.4A). This shows the importance of modelling tissue shape in Turing patterns.

4.2.3 Growing colony with cellular automaton

Most of the work we carried out experimentally to explore the system was done in bacterial colonies. These colonies are radially growing biofilms with stochastic cell division. Therefore, the static image recognition method used above is not suitable as we do not have time-series confocal data for most samples to create a dynamic "shape matrix". To recreate the dynamic behaviour of the colony growing, a stochastic growth model was developed using a cellular automaton.

A cellular automaton is a discrete model of computation constructed with a few basic rules (Gardner 1970), which can accurately describe how the shape of the bacterial colony evolves over time. As the "shape matrix", the cellular automaton consists of a Boolean 2-dimensional matrix, where grid points can be in a cell (1) or agar

(0) state. This Boolean matrix evolves over time when the following three rules are applied: If a cell (1) grid-point has any agar (0) neighbours, it will divide into the neighbouring agar (0) gridpoint with a probability p_d . No cell death (1 to 0 transition) is permitted. Newborn cells inherit the full concentration of their mother cells. This last assumption is taken as mRNA transcript homeostasis ensures the concentration of mRNAs is maintained at cell division and as cell size increases by scaling between transcription rates and cell size (Berry and Pelkmans 2022; Volteras et al. 2023). Because we are modelling protein concentration, we can then assume that mRNA concentration is linearly correlated with protein concentration for synthetic genes like ours. If some dilution occurs, this can be accounted for in the degradation terms of the PDE model.

To model a single colony, a 0's matrix is initialised with a 1 in the middle, describing the first cell as it occurs in single-cell colonies. When the cellular automaton rules are applied to this initial matrix, a circular cell domain starts growing stochastically, resembling a bacterial colony (see Fig.4.5A-B). The division process consists of a probabilistic process where division occurs or not based on a probability of division (p_d). The division process is iteratively applied to the matrix until the final time (T) is reached. The computation is applied every 'm' hours, so

$$T = \sum_{n=1}^{T/m} m \cdot n \quad (4.1)$$

e.g. if $m = 0.2$ at $T = 0.2, 0.4, 0.6 \dots$ etc . The growth rate can be tuned by increasing the probability of division (p_d) or decreasing m , so it matches the experimental growth rates. Linear growth curves are obtained for the simulations with the cellular automata, similar to the experimental ones (see Appendix C). Furthermore, different p_d 's can be applied to different regions of the matrix to achieve faster-growing subregions within the colony (see Fig. 4.5C). Finally, to simulate two colonies, two 1's are placed at a distance (see Fig. 4.5D).

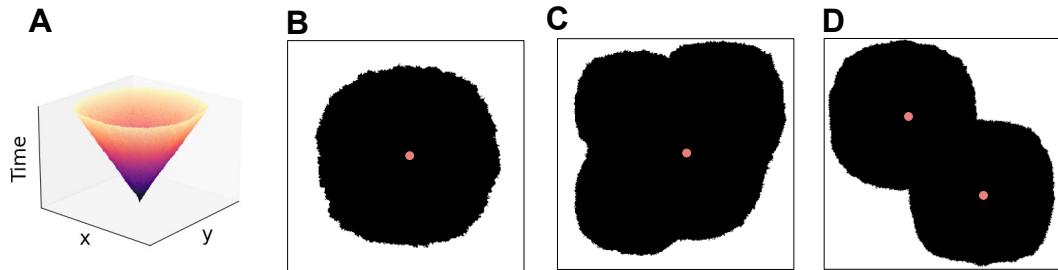


Figure 4.5: Stochastic model of colony growth with cellular automaton. (A) Time series of colony growth in 2D. Purple depicts older cells, while yellow shows newer ones. (B) Final snapshot of a colony growing with growth rates homogeneous throughout the colony. (C) Final snapshot of a colony growing with different rates throughout the colony. (D) Two colonies growing and merging with homogeneous growth rates. Black shows cells, white shows agar and the pink dot shows the initial cell from which the colony grows.

The masks obtained can be used to compute the solution of the PDE in the biofilm. Unless otherwise stated, a reflective boundary condition (Neumann) is used at the edge of the square where the agar finishes. Taking an arbitrary parameter set, this method produces a numerical result similar to that of the colonies obtained in Section 4.1. The comparison can be seen in Fig. 4.6 where the red edge with green centre is reproduced. This pattern, which is commonly seen throughout, most likely stems from the boundary effects at the edge of the colony which is captured by the masking method.

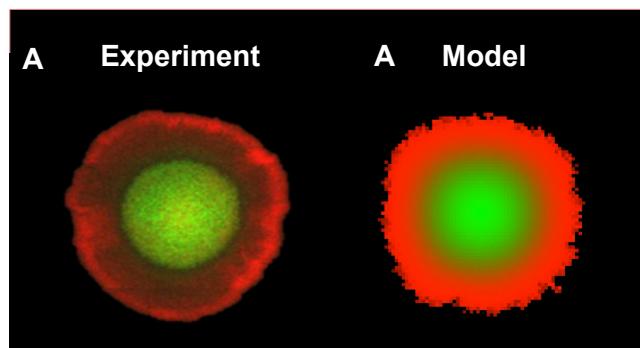


Figure 4.6: PDE model with colony growth replicates experimental colonies (A) Red-green superposition image of small E.coli colony with circuit #1754. (B) Numerical solution of masked PDE using a cellular automaton colony growth model. Both images show a colony with a red edge and a green center.

4.2.4 Adapting time and space in the dimensionless model

When solving our system numerically, time and space are key parameters that define the bounds of our simulation. As seen in the previous chapter, time and space are

dimensionless and dependent on the model's parameters

$$t = \frac{t^*}{\mu_a}, \quad x = \sqrt{\frac{k_1 D_u}{\mu_a \mu_u}} x^* \quad (4.2)$$

When sampling the parameter space, the parameters related to time and space transformations are fixed, so $\mu_a = 0.3 h^{-1}$, $k_1 = 0.0183 h^{-1}$, $\mu_u = 0.0225 h^{-1}$. The diffusion rate D_u is the only one not fixed and it ranges from $0.1 - 10 mm^2 h^{-1}$. Using Eq. 4.2, time is transformed so that $t^* = 3.3 \cdot t$. Space is also transformed but is dependent on D_u , which is sampled from a range. Therefore, for $D_u = [0.1, 10] mm^2 h^{-1}$, $x^* = [1.92 - 0.192] \cdot x$. Using realistic experiment time and space, the following dimensionless values are obtained: For time $t = 166 h$, dimensionless time $t^* = 50$. Dimensionless space $x^* = 16$ is taken so $x = [8.32, 83.2] mm$, with $D_u = [0.1, 10] mm^2 h^{-1}$ respectively. These values of t and x lie within realistic physical parameters for our system. These transformations are dependent on parameters that may vary experimentally, and therefore some uncertainty must be allowed. The experimental values used here are an example of how to carry out the transformations, however all simulation parameters for space and time can be found on Tables B.1, B.2, C.1.

The space resolution taken is $dx^* = 0.1$. If transformed to non-dimensional units, this would be $dx = [0.052 - 0.520] mm = [52 - 520] \mu m$. This means that the biological size of every pixel of the simulation varies from $52 \mu m$ to $5200 \mu m$ depending on the D_u chosen. Because the model is dimensionless, the spatial resolution is lower if a higher diffusion constant for U is used. If we assume an *E. coli* cell has a size of around $1 \mu m$ (Shiomi et al. 2009), in every pixel there are around 52 to 5200 *E. coli* cells.

4.3 Colony pattern dynamics in parameter space searches

The experimental work in this thesis produced the first rings observed in the system, where small colonies were used (Section 4.1). Following this, experiments were carried out in larger colonies by Dr. Jure Tica, Tong Zhu, Dr. Georg Wachter and Dr. Dario Bazzoli from the Isalan Lab. From here onwards, all experimental results were produced by them unless otherwise stated. The only experimental work produced in this thesis is the one shown in Fig. 4.2. A wide variety of patterns were produced by growing larger colonies. In this section we replicate these experimental patterns with the growing colony model described above.

4.3.1 Circuit exploration in literature-based distribution

In parallel to microscopy work, the model predicted a wide variety of patterns could be produced in different dynamical regimes, which matched the experimental variability. This was shown by exploring the parameter space of our circuit within literature-based ranges. The distribution used is described in Section 3.2.1 and more specifically Table 3.5. In this exploration, linear stability analysis was carried out and outputs were classified following linear stability classification in Fig. 2.9. The frequencies in parameter space of the linear stability solutions are shown in Fig. 4.7.

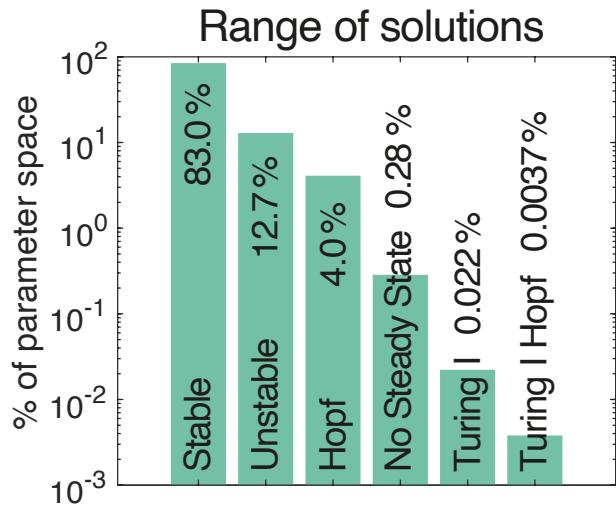


Figure 4.7: Types of dispersion relation solutions in literature-based distribution. The frequency in parameter space for every type of dispersion is shown in terms of the percentage of sampled parameter sets.

Examples of different linear stability analysis outputs were then solved numerically, masked by a growing colony obtained with the cellular automata algorithm. The different final pattern snapshots and time series can be seen in Fig 4.8. A wide variety of patterns and dynamics is observed including stationary rings, travelling waves, stationary and non-stationary spots, labyrinths and bistability wedges. This global model analysis, with biologically relevant parameters obtained from the literature, showed that our circuit can produce a broad range of spatial patterns, together with spatially homogenous solutions. Overall, Turing I-Hopf and Turing I patterns are the most interesting heterogeneities due to their periodicity and stationarity. However, they are not very robust (0.022% and 0.0037% respectively). On the other hand, Hopf solutions also produce interesting heterogeneities which often sometimes periodic and seem to occur more robustly (4%). It is important to add that Turing I Hopf solutions seem to produce patterns more robustly in this chapter than in Chapter 2. All of the patterns shown are single steady state systems so we can understand the individual and isolated behaviour of that dynamical system. However, some interesting multistable systems are present such as the Turing-Hopf-Unstable system

shown in Fig. 4.9 Rings #1.

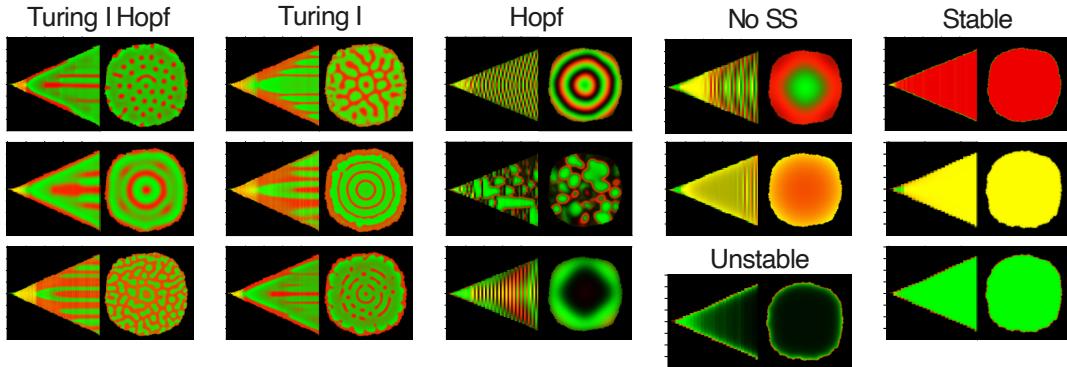


Figure 4.8: Simulations of the hybrid PDE-bacterial colony solver for different types of dispersion relations. The kymographs on the left show the time series of the colony cross-section. The plot on the right shows the final snapshot of the simulation. In both, the green and red channels are superposed. Turing I Hopf and Turing I solutions show stationary periodic patterns including outer ring addition. Hopf solutions show oscillatory solutions including interior ring growth.

This wide variety of patterns was explored, and corresponding experimental solutions were found (see Fig. 4.9). As the model predicted, the system can experimentally produce rings, spots, and wedges. However, although labyrinths commonly appear in the model, they have not been found *in-vitro*.

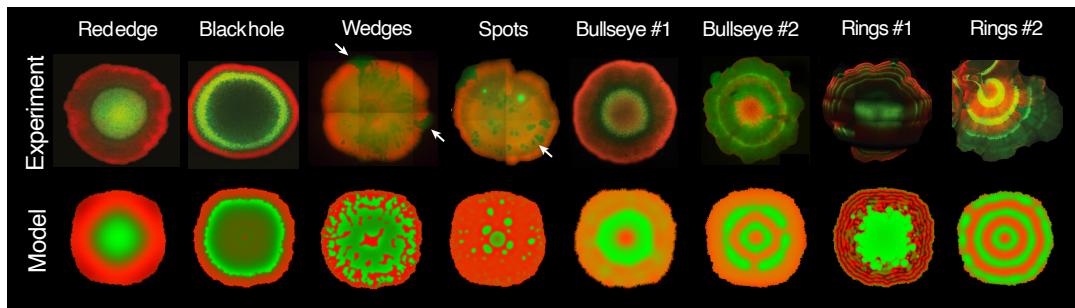


Figure 4.9: Variety of spatial patterns observed experimentally as well as in the model. Various spatial patterns are observed when the full circuit is tested in growing colonies in different experimental conditions (upper row). The white arrows show two wedges and a region of spot formation. Colony size and tuning conditions are listed in Table S1. Patterns are reproduced with the circuit model (bottom row); for parameters see Suppl. Info. 4. All images except Red edge, Black hole and Bullseye #1 can be attributed to Dr. Jure Tica, Tong Zhu and Dr. Georg Wachter. All parameters for these simulations can be found on Tables B.1, B.2, C.1. The experimental conditions and colony sizes are shown in Table D.1.

4.3.2 Circuit exploration in liquid-culture fits distribution

In the previous section, the model was explored using large literature-based distributions and was compared to experiments in a different range of tuning conditions.

In this section, we focus on a more constrained region of the parameter space obtained by fitting the model to liquid culture data (Section 3.3). This parametrisation aims to explore a model which is better linked to the experimental system in terms of parameters and behaviour. More specifically, this constraining is carried out to prove that the obtained experimental patterns exist in this fitted parameter space, in particular the more Turing-like concentric rings (Fig.4.9 Rings #2).

In Section 3.3.2, the model was fitted to dose-response curves of subcircuits under high aTc and matched transfer functions conditions. These are the same conditions where the Fig.4.9 Rings #2 appear. As previously explained and shown in Fig. 3.15, linear stability analysis was carried out on the fitted multivariate Gaussian distribution with $q = 10$, and three Turing parameter sets were found. These are the closest three Turing parameter sets to the best-fit solution. Those three parameter sets were simulated using a colony growth mask (see Fig. 4.10). Their dominant features include a central circular green (Case #1) or red (Case #2) spot, surrounded by a ring of green fluorescence, and a wedge or labyrinthian-like pattern. These are all observed within the experimental data (see Fig 4.9). Although Case #1 seems to have similar rings to the ones we are looking to replicate (seen in Fig.4.9 Rings #2), more rings would be needed to prove that the periodic ring-like behaviour exists in the fit distribution.

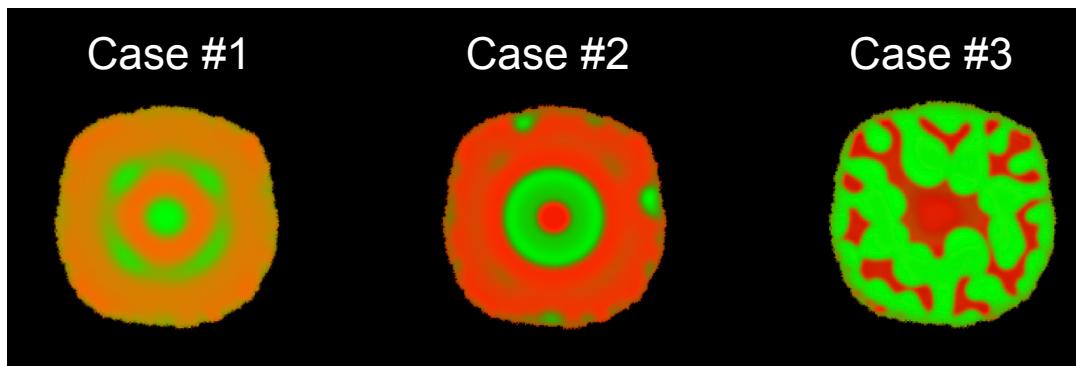


Figure 4.10: Growing colony simulations of Turing solutions with fitted parameters. Parameters for these simulations can be found in Tables B.1, B.2, C.1.

Three routes can be taken to further explore the patterns present in the fit distribution. The first one would be to explore different numerical parameters (e.g. space, time and growth rate) to obtain a different pattern. However, numerical parameters are harder to explore from a computational parallelisation point of view. The second one would be to sample more from the fit distribution to find more Turing parameters and simulate those. However, Turing parameters are not commonly found and therefore it is also a computationally expensive task to obtain few results. Finally, a third route exists which involves searching very closely around the vicinity of the already obtained parameter sets. This route is chosen as it is the most efficient way of obtaining many Turing parameter sets that still belong to the fit distribution.

4.3.3 Pattern exploration around the vicinity of Turing fit solutions

The parameter space found around the fitted Turing parameter sets is explored. To do this, a small noise deviation is applied to the parameters with different levels of relative uncertainty. For each parameter p , a normal distribution is generated with mean $\mu = p$ and standard deviation $\sigma = p \cdot u$. u is the level of uncertainty which ranges from 0.01 to 0.2 (or 1% to 20% relative uncertainty). This small noise perturbation to all parameters, which generates similar steady-state dose-response behaviour, allows us to further explore the Turing parameter space near the best fit to the liquid culture.

For each value of uncertainty, $2 \cdot 10^3$ parameter combinations were analysed. The Turing patterning robustness with different amounts of noise is shown in Fig. 4.11A, showing how robustness decreases as more noise in the parameters is added. This figure shows how the local parameter space around Turing conditions is highly enriched with patterns (e.g adding a relative uncertainty of 1% around a Turing I solution produces 33% Turing I solutions, whereas an uncertainty of 5% produces 5% Turing I solutions). The average relative uncertainty in the V_m and K_m parameters between biological repeats in liquid culture data of Fig. 1b was 4.8%. This indicates that if a patterning region was found, Turing patterns could be sufficiently common to be reproduced. The different analytical solutions for a relative uncertainty of 1% are shown in Fig. 4.11B, where we observe not only Turing I solutions, but also Turing I Hopf and Hopf solutions. For this relative uncertainty (1%), numerical solutions are computed and shown in Fig. 4.11C. Within the small vicinity of a ring-like Turing parameter set, we can find mainly rings and spots. In particular, we can find solutions with multiple concentric rings such as the one marked with an arrow in Fig. 4.11C Case#4.

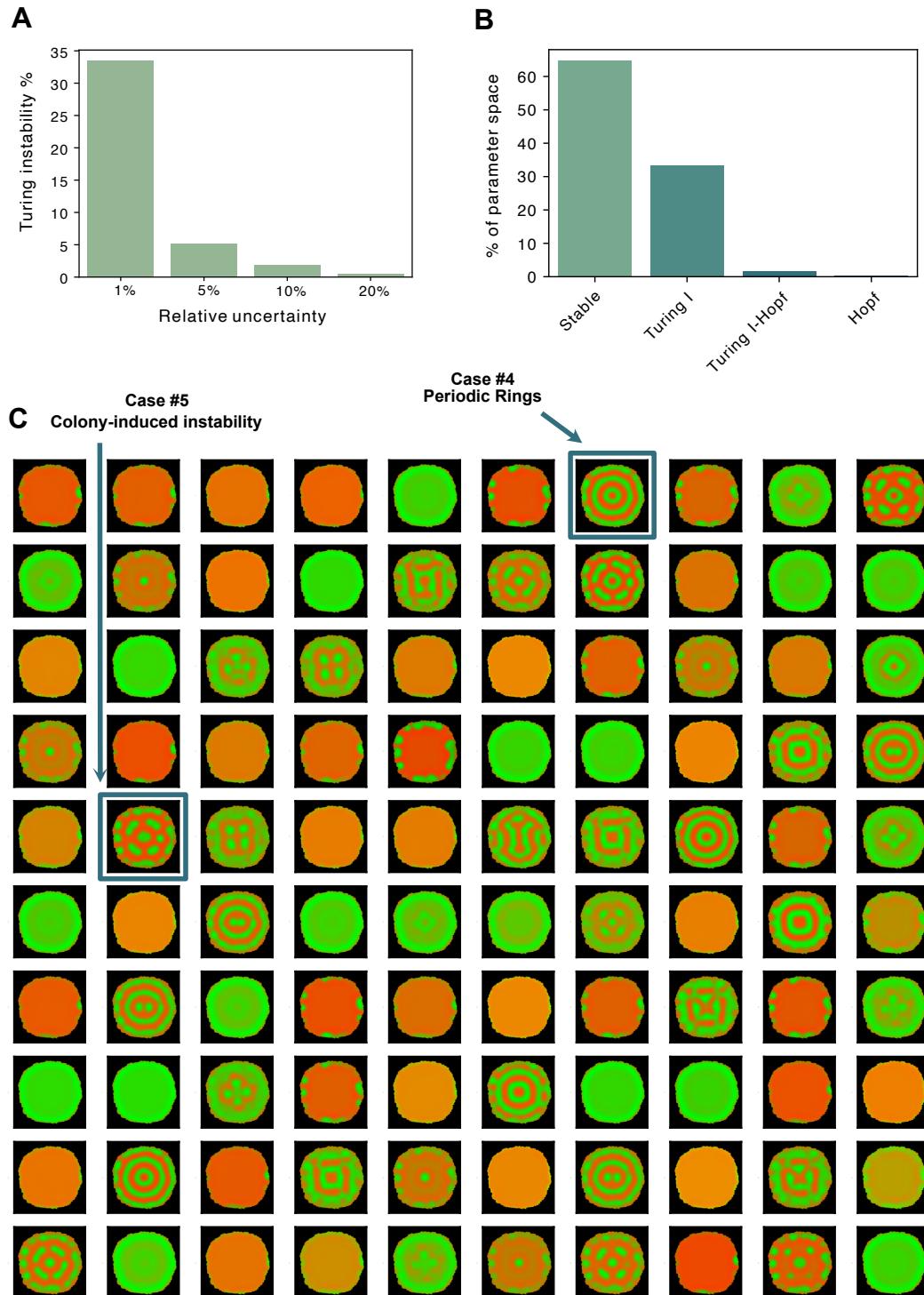


Figure 4.11: **Search around fitted Turing parameter set.** (A) Turing robustness with different levels of relative uncertainty. (B) Using 1% relative uncertainty (mean $\mu = p$ and standard deviation $\sigma = p \cdot 0.01$), frequency of different analytical solutions: Simple stable 64.8%, Turing I 33.5%, Turing I-Hopf 1.5%, Hopf 0.25%. (C) Numerical simulations in growing colonies of 1% noise distribution. Arrow points to Case #4 which is a solution with multiple periodic rings and Case #5 where a colony-induced instability is present.

The time series of this particular Case #4 Turing solution is explored and compared to different snapshots of the bacterial colony patterns. Outer ring addition dynamics of both the model colony and the experimental colony are shown in Fig. 4.12, where rings get added to the edge of the colony. One aspect to highlight is the diffusion ratio Dr of this solution. Assuming that the enzymatic production and degradation rates of the two diffusers pC and $OHC14$ are equal ($k_1 = k_2$, $\mu U = \mu V$), a Dr of 0.15 is very close to an experimentally measured Dr of 0.25 Tica et al. 2023b.

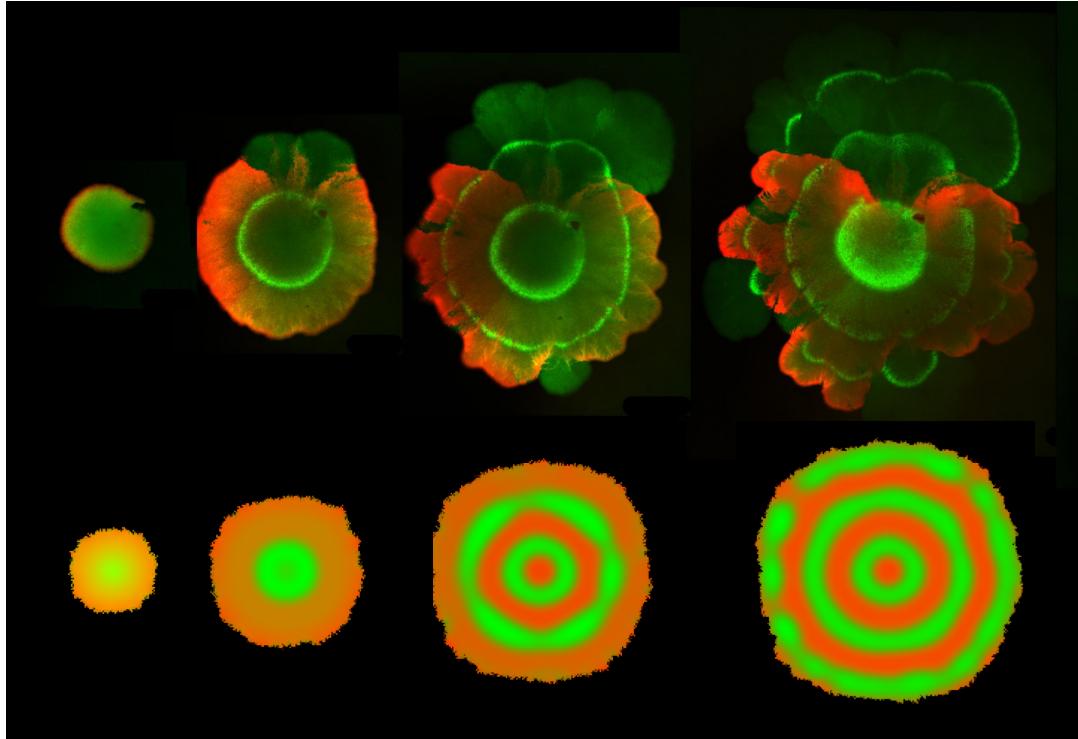


Figure 4.12: **Time series of growing colony in experiment and model.** The top row shows confocal microscopy snapshots of the experiment at different time points. Image taken by Dr. Jure Tica. The bottom row shows the time series of the growing colony model. The model parameters used correspond to a Turing parameter combination found in the vicinity of the fitted distribution as seen in Fig. 4.11 Case #4. In both experiment and model, a similar dynamic behaviour occurs where rings get added to the outside of the colony.

Other interesting solutions are found such as the colony-induced Turing pattern (Fig 4.11C Case #5). In Fig. 4.13 we see an example of an instability induced by stochasticity in cell division or growth: the dispersion relation doesn't show any unstable modes; however, the simulation shows a clear periodic heterogeneity. The dominant mode of this dispersion relation (Fig. 4.13A) has a wavenumber of 1.8, which corresponds to a wavelength of $2\pi/1.8 = 3.49$. This approximately corresponds to the wavelength of the produced pattern (Fig. 4.13B), meaning this stable mode has been excited to unstable and resulted in a Turing pattern.

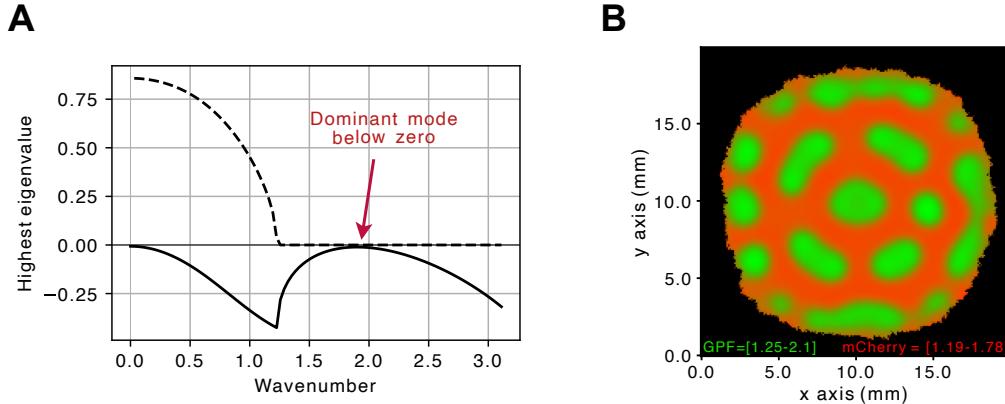


Figure 4.13: **Colony-induced instability.** (A) Dispersion relation showing a stable system. The most dominant mode (wavenumber=1.8, wavelength=3.49). (B) Regular pattern is observed in the numerical solution of a stable system with dominant mode just below zero. The model parameters used correspond to a Turing parameter combination found in the vicinity of the fitted distribution as seen in Fig. 4.11 Case #5.

4.4 Elucidating mechanisms through experiment controls

Although the patterns obtained experimentally might seem qualitatively similar to Turing solutions in bacterial colonies, further controls are needed to strengthen the hypothesis that these are Turing patterns. Such controls involve introducing a perturbation in the model, which leads to a change in pattern features; and obtaining the same change in pattern when producing such perturbation experimentally.

Most of these perturbations except the deletions section will be applied to the Case #4 Turing condition shown in Fig 4.12, where multiple rings appear when using parameters from the fit distribution.

4.4.1 Irregular growth

The first control involves studying the pattern under different sizes of biofilm. An interesting way to look at this problem is by understanding how the pattern changes within the same colony in smaller or larger areas. By tuning the growth rates of the cellular automata differently in different regions of the colony, we can obtain faster-growing domains which will be larger than others (see Fig. 4.5C). This resembles the bacterial colonies produced which have faster growing edges. The model shows that larger domains will produce more rings than shorter domains (see Fig. 4.14 right). This prediction is correct, as experimental data produces the same behaviour (see Fig. 4.14 left). Additionally, in both model and experiment, this irregular growth leads to discontinuity in the concentric rings.

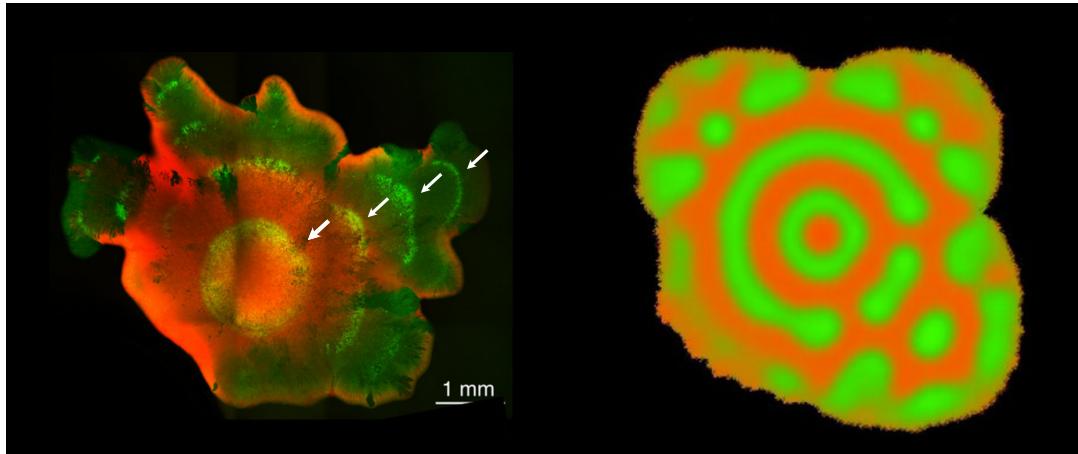


Figure 4.14: Effects of irregular growth in growing colony pattern. Both in experiment (left) and model (right), more stripes are found in faster-growing fields which form a larger domain

4.4.2 Boundary effects

The second control involves studying how boundary conditions might affect the resulting pattern. The boundary conditions at the edge of the colony are always assumed to be absorbing boundary conditions because the diffusors produced in the biofilm get absorbed by the empty agar because of diffusion. This boundary is indirectly encoded with the masking process (Fig. 4.3). However, the boundary at the edge of the simulation square or in other words, where the agar would finish, has to be defined in the ADI numerical solver. As in Chapter 2, the absorbing boundaries are introduced by using a Dirichlet boundary condition where the concentration at the boundary is zero $u = 0$ as opposed to the previously used Neumann boundaries where the derivative at the boundary is zero. More details on the encoding of boundaries can be found in Section 6.6.

Up to now, all simulations in this Chapter were computed using reflective boundary conditions at the edge of the square. Here, we introduce absorbing boundary conditions and see that this perturbation leads to fewer rings being produced. The hypothesis that absorbing boundary conditions make weaker patterns is then tested experimentally. Absorbing and Reflecting experimental boundaries can be introduced using different sizes of agar plates where the colony grows as seen in Fig. 4.15. A larger plate will resemble an absorbing boundary condition as we assume diffusors build-up will be less prominent and there will always be a diffusor flux towards the edges of the plate. On the other hand, a small plate will lead to the quick accumulation of diffusors and therefore these will quickly be reflected as they reach the boundary. Experimental colonies behave according to model predictions, displaying fewer rings when grown on bigger plates.

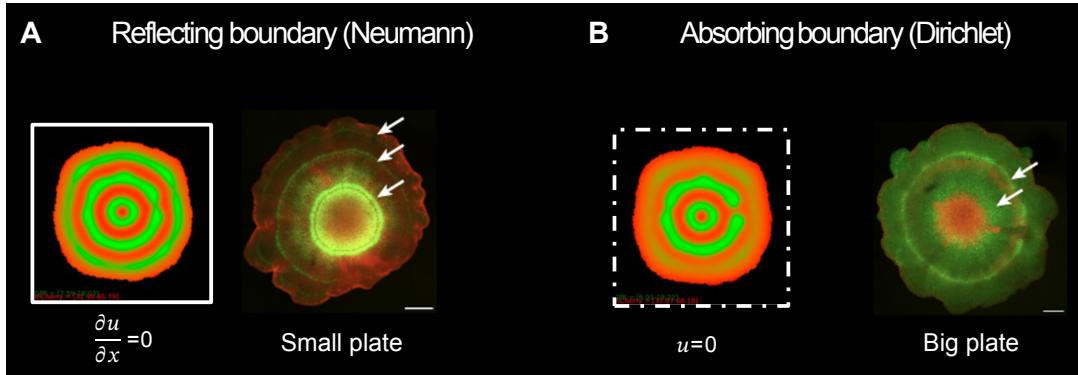


Figure 4.15: **Effects of boundaries on pattern formation.** The boundary condition affects the patterns. **(A)** Multiple rings form when cells are grown in smaller wells. Smaller wells are simulated using a reflecting boundary introduced with Neumann boundary conditions. **(B)** Fewer rings form when grown in larger dishes. Smaller wells are simulated using an absorbing boundary introduced with Dirichlet boundary conditions.

4.4.3 Node deletions

Another important perturbation is deleting nodes of the circuit to study how patterning is affected. Deletions of each node described in Fig. 4.16 were studied using analytical and numerical methods, by re-sampling the parameter space with the literature-based distribution. The models were defined by taking the original six equation model and removing the following species: RpaI and TetR for Node A deletion, cinI for Node B deletion and cI* for Node C deletion (see Fig. 4.16).

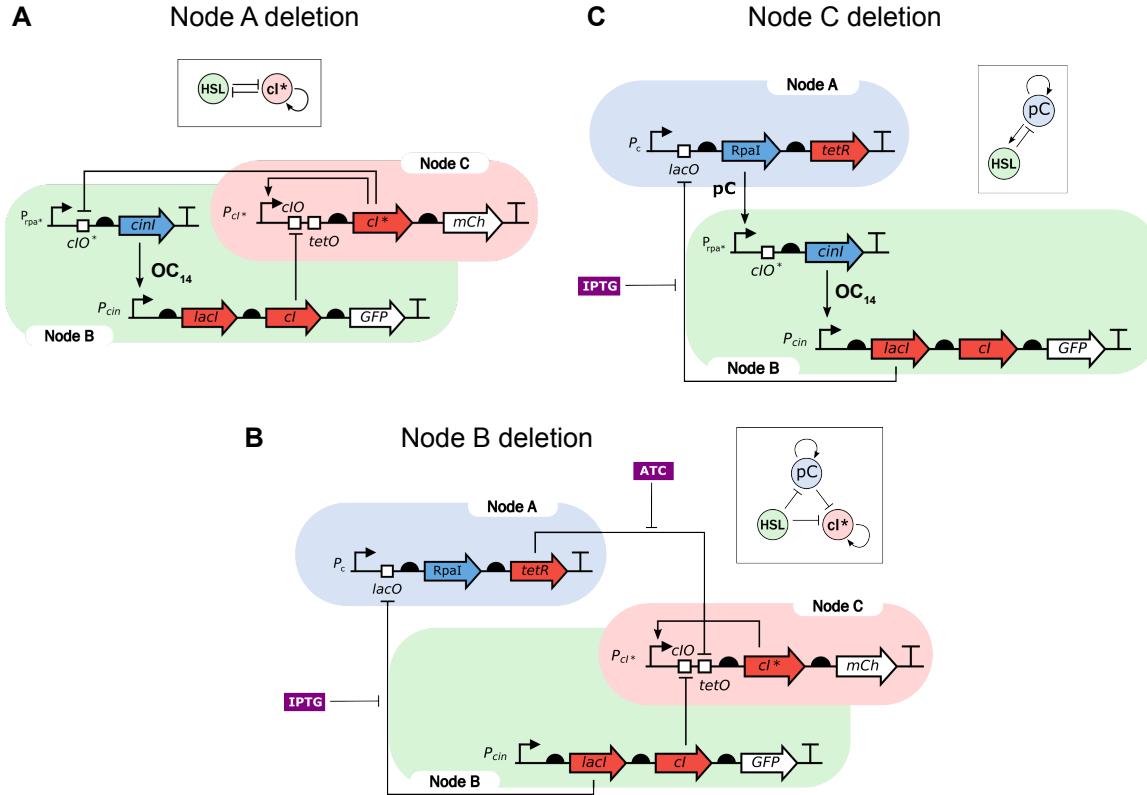


Figure 4.16: Diagrams of circuit deletions. (A) Node A deletion involves getting rid of $RpaI$ and $tetR$ genes. This results in a 2 node circuit with a single diffusor. (B) Node B deletion involves removing the $cinI$ gene as well as the inhibition from node C to $cinI$. This leaves a 3-node circuit with no diffusors regulating gene expression. (C) Node C deletion involves removing cl^* and GFP genes, which leads to a 2-node circuit with two diffusors with the original Turing topology.

No Turing I or Turing I Hopf instabilities were found when sampling these three circuits using linear stability analysis. Some samples were simulated using numerical methods and no periodic patterns were observed either. Therefore, the model predicts no patterns should arise from the node deletion variants shown in Fig. 4.16.

These variants were then built experimentally and tested by the Isalan group. Deletions for node A and node B involve removing the plasmid encoding for that node as in the model. Deletion for node B involved deleting $cinI$ which is the enzyme producing OC_{14} , therefore disconnecting node B from the circuit (Fig. 4.16). In this variant, node C is deleted too as its only feedback on the circuit is on $cinI$. Thin green stripes in the GFP channel, similar to those of the full circuit in Fig. 4.12, were observed with the three controls after imaging the circuit daily, which causes a temperature drop each time (see Fig. 4.17).

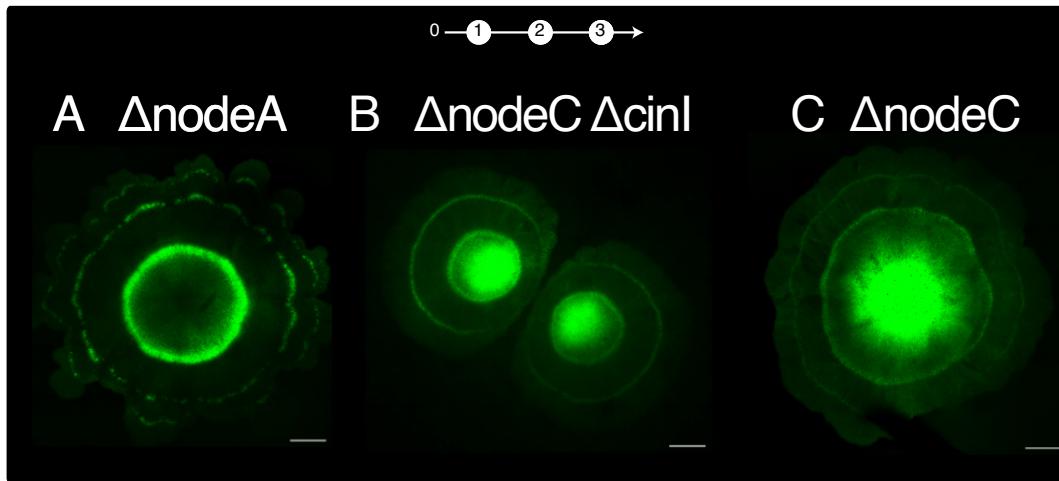


Figure 4.17: Control experiments testing circuit deletions. All images are taken 4 days after plating, and only the GFP channel is shown. The schematic on top shows on which days a cold-shock has been applied by imaging (white circles). Controls are (A) node A deletion, (B) node B deletion and (C) node C deletion. Images taken by Dr. Jure Tica

In this control, the model and experiments disagree as the model does not predict the ring patterns appearing experimentally. Therefore, these specific rings in the controls cannot be explained with the Turing mechanism and other mechanisms need to be explored.

4.4.4 Temperature variations

A potential explanation for the rings is that stripe formation was induced by the imaging procedure, possibly because of a drop in temperature during the imaging. This hypothesis was not tested with the model because of lack of time. However, it was explored experimentally by Dr. Jure Tica and Dr. Dario Bazzoli and will be shown here for story completeness. The results in this section have not been produced as part of this thesis.

Instead of imaging periodically, the colonies with the full circuit were subjected to cold shocks by putting the plate at room temperature for an hour from the 37° degree incubator. First, they cold-shocked the colony every day for four days and four rings were observed on the 5th (see Fig. 4.18A). Then, they only cold-shocked on day one and imaged on day four after plating and this showed three rings (see Fig. 4.18B). Finally, no cold shocks were applied until imaging on day four, where no rings were observed as seen in Fig. 4.18C. This proved that periodic cold shocking helps, but it is not required for periodic patterning. However, a single cold shock on day one is required to get periodic stripes.

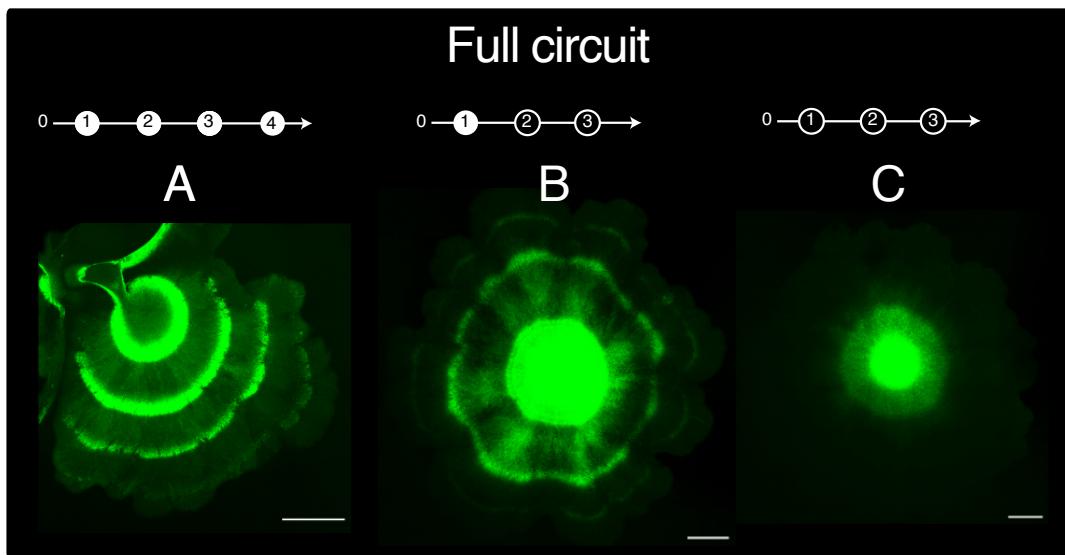


Figure 4.18: Full circuit experiments under variation of cold shocks. Green channel of circuit #1754 exposed to different cold shocks. Cold shocks involve taking the plate out of the incubator for 1h at room temperature. This simulates the cold shock applied to the colony when imaging it in the confocal microscope. Top diagrams show when the cold shocks were applied (white circles), and when they were not (black circles). **(A)** A cold shock is applied every day for four days, and the image is taken on the 5th day, showing four rings. **(B)** A cold shock is applied only on day one and the image is taken in day four, showing three stripes. **(C)** Cold shock is applied only on the 4th day when it is imaged: no rings appear. Images taken by Dr. Jure Tica

To further prove that no periodic cold shocks were needed for periodic patterning, we sought to obtain a time series of the colony while keeping the cells at a constant temperature. Colonies were grown for a day and then transferred to the microscope, where the first and only cold shock occurred. The cells were kept at 37°C with a stage-top incubator, and images were taken every hour. Consistent with the cold shock hypothesis, the first stripe formed soon after the start of the imaging. Surprisingly, additional stripes formed on the edge of the colony (Fig. 4.19). Both the timelapse (Fig. 4.19) and day1-day4 imaging (Fig. 4.18B) show stripes without everyday imaging. This suggests that after the first stripe is seeded additional stripes can form in a circuit-dependent, reaction-diffusion process.

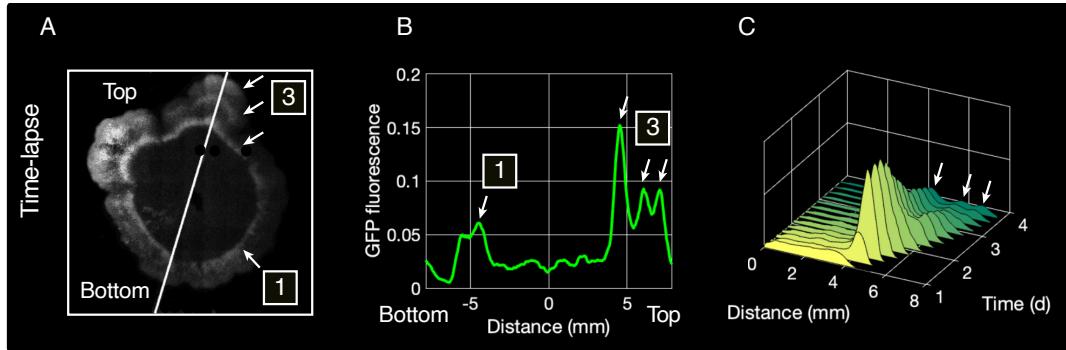


Figure 4.19: GFP fluorescence was imaged over 60 hours at a constant temperature of 37 °C. Imaging began after the colony had grown for 24 hours, meaning a single cold shock was applied at 24 hours. (A) The image of the final timepoint shows three stripes in the top right corner of the colony where growth is most prominent (3), and only a single stripe forms at the bottom where there is less growth (1). (B) Plot of the fluorescence along the white slanted line in the micrograph with moving average smoothing. (C) The spatiotemporal profile of GFP evolution shows that the stripes form at the edge of the colony. After an initial burst of fluorescence, the signal decreases over time. Images taken by Dr. Jure Tica

4.5 Discussion

The focus of this chapter was the engineering of patterns in biofilms using the #1754 topology. Both experiments and models of synthetic biofilms with spatial patterning were presented, including a descriptive model to get insights into the patterning mechanisms present.

4.5.1 *In-vitro E. coli* colonies with periodic patterns

Even though this thesis focused mainly on modelling work, some experimental work was produced to get familiarised with the genetic circuits and the microscopy techniques. The insights obtained during this work were key to building useful models with accurate assumptions.

During these experiments, the first periodic rings obtained with the #1754 synthetic circuit were obtained. These were produced in small colonies with high aTc conditions. In Chapter 3, the model predicted that the likelihood of Turing instabilities was higher with high aTc concentrations. Although it is unclear if the colonies obtained display Turing patterns, aTc did indeed lead to spatial heterogeneities, therefore making the model's predictions useful in guiding the experimental fine-tuning.

The spatial heterogeneities displayed interior ring growth dynamics, where rings were deposited from the inside of the colony and then stretched out (see Fig. 4.2 B). This behaviour looked similar to the interior ring growth obtained in Hopf instabilities in Fig. 4.8 Hopf, top row. Unlike Turing solutions (Turing I-Hopf and Turing

I) where stationary rings were added to the outside, Hopf solutions showed oscillating centres that form into rings in the next step, as in the colonies obtained. The most likely mechanism behind these specific colonies is therefore probably a Hopf oscillator in a growing colony. This is also supported by the high probability (4%) for Hopf solutions in the fine-tuned parameter space (see Fig. 4.7).

4.5.2 Solving PDEs in shaped biofilms

Following the experimental work produced in this thesis, experimental collaborators explored the system further using larger colonies or biofilms with particular shapes. A PDE-masking algorithm was built to understand the patterning dynamics of the reaction-diffusion systems in these different biofilm shapes as well as growing domains. This algorithm is flexible enough that the PDE system can be solved in any mask provided. Additionally, for complex shapes, image recognition was used to produce this mask as seen in Fig. 4.4. However, static domains are not the best assumption for growing colonies, as growth may play an important part in tissue dynamics as seen in Chapter 2. To encode domain growth into the algorithm, a stochastic cellular automaton was used to produce a growing mask which replicates bacterial colony growth.

One of the caveats of this algorithm is that growth only occurs in the edges and there is no cell death. In bacterial colonies, although growth is more prominent in the edge, some growth and death also occur in the centre. Dilution effects were also not taken into account in this algorithm, however, they can be accounted for in the linear degradation terms of the PDE model. Another issue with the model is that a 2D space was assumed. Although confocal microscopy only shows a 2D space domain, *E. coli* colonies have a 3D dome shape (Wimpenny 1979). Therefore, a 2D snapshot of a 3D model would be more accurate to capture the RD dynamics of the gene circuit. However, a 2D PDE model was used as it was less computationally expensive.

The PDE model used for these simulations was the six-equation non-dimensional model. This non-dimensional form of the model makes time and space less intuitive, as these variables have been transformed into non-dimensional. However, by understanding the parameters of our system we can retransform space and time to their original units: hour and mm. This transformation is more complicated for space as it is dependent on D_U , which is sampled from a range.

In terms of spatial resolution, the system was pushed to the smallest dx possible in terms of computational resources. A pixel of the simulation grid contains around 52-5200 *E. coli* cells, meaning the system is not discrete at the cell level. However, we assumed that neighbouring cells have similar dynamics, especially for Turing patterns with larger wavelengths. Although it is unusual to have a spatial resolution which varies with diffusion rates, the pattern resolution is scaled in the same

manner. A low spatial resolution (e.i. 5200 *E. coli* per pixel) occurs for large diffusion constants, as well as the pattern wavelength which will be larger with higher diffusion constants.

Overall, this masking method enabled the investigation of Turing patterns in shaped and growing domains. It was observed that the shape highly affects the resulting Turing patterns obtained when masking the PDE solver with a specific shape. Throughout this exploration, we saw that Turing patterns tend to form concentric lines following the shape of the biofilm. This can be an example of how shape conveys morphological robustness to patterning in biology.

4.5.3 Replicating bacterial colony patterns

Using the PDE algorithm masked with the cellular automaton, all patterns obtained experimentally were replicated with the model. Having documented which types of dispersion relation led to which types of patterns in growing colonies (see Fig. 4.8), we used linear stability analysis to understand the potential mechanisms behind each pattern obtained. The common red-edge and green-centre pattern (Fig. 4.6) could result from any type of dispersion relation, meaning this was a simple boundary effect. This pattern occurs when diffusers at the edge of the colony get depleted as they leak out onto the agar. By looking at the circuit (Fig. 3.1), we can observe that diffusors are direct GFP activators and mCherry inhibitors. Therefore, as they get depleted, red is produced and green decays.

Other more complex patterns such as the rings and spots, only appeared in Turing I, Turing I-Hopf and Hopf instabilities (see Fig. 4.8). More specifically, interior ring growth could be explained with Hopf instabilities while outer ring addition could be explained with Turing I and Turing I-Hopf instabilities. This coincides with experimental work on Turing patterns in radially growing domains (Konow et al. 2019). In this study, they observed that outer ring addition is the most common Turing pattern growth mode, while interior ring growth in Turing patterns only occurs for extremely slow growth rates. Labyrinths, which were found in numerical solutions, were not found experimentally. It is interesting to note that periodic patterns in Turing I-Hopf instabilities occurred more commonly in this model than in Chapter 2. This shows how Turing I-Hopf instabilities generate patterns more robustly in certain models or parameter regions.

Another feature of the system which was replicated by the model was the pattern variability or lack of reproducibility. By searching around the vicinity of a parameter set, many Turing patterns with different shapes could be found. This could explain why in *in-vitro*, the same experimental conditions led to different pattern outcomes.

Although the outer ring addition and spots obtained can be explained with a Turing mechanism, the robustness of such is extremely low. Other effects could be increasing the Turing robustness, making this outcome more likely. For example,

colony-induced patterns were observed (see Fig. 4.13), where the dispersion relation showed no instability but the numerical solution showed periodic spots. This could be due to the effects of growth, the absorbing boundary conditions or the stochasticity in cell division. Additionally, it was found that the robustness around a Turing parameter set was extremely high (33.5% with 1% uncertainty) compared to the general robustness (0.0025%) as seen in Fig. 4.11A. Therefore, if by chance the system is explored in experimental conditions that lead to observing a Turing pattern, it is likely to observe another Turing pattern when repeating the experiment under the same conditions.

4.5.4 Elucidating mechanisms by modelling experimental controls

Looking at how different aspects of the pattern compare in the model and in the experiment, can help us elucidate the mechanisms of patterning. In particular, we aimed to elucidate the mechanism behind the outer ring addition patterns observed in the time-series of Fig. 4.12 top. The first mechanism that comes to mind is a Turing mechanism, as this produces outer ring addition in our model. More specifically, a Turing parameter set obtained from the fitted distributions closely replicates the time-series data of the outer ring addition patterns (See Fig. 4.12 bottom). This means that parameters which are obtained from fitting to liquid-culture experimental data can produce a Turing pattern which resembles our periodic ring patterns. It is key to mention that the diffusion ratio of this parameter set was the same as the measured diffusion constants in Tica et al. 2023b. The Turing hypothesis is further strengthened if a Turing parameter set with an experimental diffusion ratio can produce the same experimental rings.

Additionally, several model perturbations applied to this Turing parameter set coincided with the experimental perturbations. In experiments, faster-growing regions of the colony produced more rings, which was replicated with the irregular growth in the cellular automata algorithm. Furthermore, more rings were produced in small plates which was replicated by the model using reflecting boundaries as opposed to absorbing boundaries. This result is reminiscent of Krause et al. 2020, where a thick agar layer that absorbed diffusors disrupted Turing patterning. If perturbations to a Turing model generate the same behaviour in the model as in the experiment, a Turing mechanism is likely behind such a system.

Another perturbation produced was the deletion of nodes, which generated similar concentric as seen in Fig. 4.17. Linear stability analysis and numerical simulations determined that such circuits could not produce Turing instabilities. This suggested rings could be produced by another mechanism such as cold shock effects. The cold shock hypothesis was tested and indeed it was found that by imaging the colony every day, periodic rings could form as a result of the periodic cold shocks. The cold shock hypothesis was further proved by the fact that the stripes consistently coincided with the outline of the colony at the previous timepoint (Fig. 4.12). If the edges

of the colony are more transcriptionally active and the node B promoter or degradation machinery is temperature dependent, a cold shock will induce a temporal overexpression of GFP in the edge of the colony. If the cells are shocked periodically, a periodic pattern will appear independently of the reaction-diffusion system.

However, time-series experiments and cold shocking at different frequencies of the full circuit showed that periodic rings could still appear without periodic shocking. Preliminary experiments of the same setup with the circuit deletions did not yield any periodic patterns, suggesting a Turing circuit must be present for those periodic patterns to appear independent of periodic cold shocks. This suggests that the reaction-diffusion architecture of the gene circuit #1754 is responsible for the periodic patterning. However, a pre-pattern seeded by the first cold shock is required for the periodic pattern to occur. Interestingly, this points to a hybrid Turing and Positional Information mechanism where a pre-pattern provides Turing with an increased robustness to produce periodic patterns. Future directions involve modelling the #1754 gene circuit with a pre-pattern to understand whether the robustness for periodic pattern formation increases as this positional information is introduced.

Definitive evidence of Turing patterns would be the emergence of non-symmetrical patterns like regular spots or labyrinths. However, rings are typically the most prevalent Turing pattern in radially growing domains (Konow et al. 2019). The spots observed in Fig. 4.9 suggest a patterning mechanism beyond cold shocking due to their asymmetry. While labyrinths would provide further proof, so far, only rings and spots have been observed in the sampled fitted parameter distributions and in the experiments.

CHAPTER 5

Discussion and Conclusion

Engineering biological Turing patterns, a mechanism that can explain many patterns in biology, has long been one of the holy grails in synthetic biology. This thesis is dedicated to developing a modelling approach that will assist in the experimental design of robust patterning experiments using Turing synthetic gene networks. Initially, it adopts a theoretical perspective, investigating analytical and numerical methods for studying such patterns. Furthermore, it aims to theoretically understand how biological factors, such as multistability, boundaries, growth, and domain shape, influence patterning. Once this foundational knowledge is acquired, a model for the synthetic experiment is constructed, enabling the identification of robust parameter spaces conducive to Turing patterning. Additionally, by employing numerical solvers tailored to our experimental setup, we seek to determine whether the observed patterns are exclusively Turing patterns or a result of a combination of various mechanisms.

In the first results chapter, our findings using numerical methods revealed instances where the assumption that only Turing instabilities can produce periodic patterns is not accurate. For instance, we observed that multistability can lead to pattern formation in non-Turing steady states and can also result in transient patterns that disrupt Turing steady states. Additionally, periodic patterns were noted in both Turing-Hopf and unstable steady states. These discoveries suggest that high-throughput methods testing Turing pattern robustness may need to revise their underlying assumptions in future research. Furthermore, these numerical methods allowed for an in-depth exploration of realistic biological effects, such as the influence of boundaries and growth on pattern formation and disruption. Future research should focus on developing more refined classification methods and investigating various types

of growth, similar to those observed in our colonies. Lastly, this chapter also highlights how pattern wavelength, temporal scales, and shape can be tuned using insights from dispersion relation analysis and optimization methods. Future research could focus on applying such optimizations to our experimental system. This would enhance our understanding of how to adjust parameters to achieve varying wavelengths and time scales, as well as to transition from spot patterns to labyrinths.

In the second chapter, a reaction-diffusion model based on a synthetic gene circuit is introduced. This synthetic RD circuit, engineered in Tica 2020, is based on the architecture of the Turing topology #1754 from Scholes et al. 2019. The six-equation PDE model accounts for inducer-dependent gene expression, degradation, and diffusion. Utilizing insights from the first chapter, an extensive high-throughput sampling of the parameter space for this model is conducted. It is observed that the gene circuit can form Turing patterns, including Turing I and Turing I-Hopf instabilities. Additionally, regions within the parameter space with an increased likelihood of patterning are identified. These regions can be experimentally attained through several methods, including matching dose-response curves and adding aTc or DAPG. However, the robustness of the system in these optimised regions is still relatively low. Extended work is required to enhance further the likelihood of Turing pattern formation and achieve robust patterning experiments. This may involve more than parameter tuning, as other effects such as growth, boundaries or noise could be key in explaining the robustness of patterns observed in nature.

Additionally, the large parameter space of the model is constrained by parameterising the system using liquid-culture data. A new method for model parameterisation is introduced, involving the non-dimensionalisation of the PDE system to allow comparisons of protein concentrations with relative fluorescent units. This method goes beyond pattern formation studies, as it could be useful to parametrise other ODE and PDE models in synthetic biology involving gene circuits. Once the parameters are constrained, we look for Turing instabilities in the fits and find some. Once the parameters are constrained, Turing instabilities in the fitted distributions are identified. These Turing instabilities are theoretically the closest Turing parameter sets in parameter space to the experimental system. Future work should focus on producing more accurate fits using RT-qPCR data, where dose-response curves for every molecular species in the circuit are developed. Additionally, alternative methods such as Approximate Bayesian Computation could be explored, which allows for the incorporation of prior knowledge and the exploration of non-linear relationships between parameters.

The engineered genetic circuit was examined in both the second and third chapters. The second chapter delved into high-throughput searches in the parameter space using linear stability analysis, while the third chapter compares patterns obtained *in-vitro* and *in-silico* through numerical methods. Building on insights about optimising robustness from the second chapter, in this thesis periodic rings were produced in

growing colonies under high aTc conditions with matched dose-response curves. A specialized PDE solver was developed to simulate the shape and growth of these biofilms and the dynamics of the RD gene circuit. Image recognition was applied to microscopy images to model static domains, and a cellular automaton was used for colony growth. Future work could merge these techniques, using domains from image recognition as initial conditions for the cellular-automaton algorithm.

Further experiments were carried out by collaborators, yielding a diverse array of patterns in the colonies. All patterns could be replicated with the masked PDE solver, including rings, spots and even labyrinths which were not found experimentally. Stationary regular patterns could be modelled with Turing I instability and Turing I-Hopf instability models, in particular outer ring addition dynamics. Non-stationary patterns could be modelled with Hopf instability models, which generated mostly interior ring growth dynamics. Linear stability analysis searches revealed high robustness for Hopf instabilities but extremely low for Turing instabilities, making experimental observation of the latter theoretically impossible. However, in the vicinity of a Turing parameter set, the robustness was much higher, indicating the possibility of repeatable pattern observation. Future work to develop more sophisticated classification methods for numerical patterns could clarify if robustness in growing colonies exceeds predictions from linear stability analysis. This would aid in understanding how various factors associated with colony growth contribute to the system's robustness in patterning. Notably, certain colony-induced patterns, not found in the first chapter, were documented, suggesting that the growth type used in our experiments might facilitate pattern formation. Further theoretical work, similar to that conducted in the first chapter, should be undertaken for this specific experimental system.

The outer ring addition dynamics generated by the Turing parameter set from the fitted distribution were studied in detail to further understand the circuit dynamics in colonies. This parameter set was chosen to be explored more in detail as its Turing behaviour should be the closest to a Turing mechanism in our experimental circuit. In particular, it was explored to determine if the underlying mechanism for outer ring addition in our colonies was indeed a Turing mechanism. While the time-series data, irregular growth controls and plate size controls aligned with a Turing model, the inability to replicate rings in node deletion controls suggested a Turing model might not fully explain this phenomenon. Periodic cold shocks appeared to induce outer ring addition patterns. However, a single initial cold shock followed by constant temperature also resulted in periodic patterns. This suggests the explanation is related to a hybrid of theories: a positional information mechanism where a prepattern is seeded by the cold shock, combined with a Turing pattern in which periodic rings self-assemble. Spots, as non-symmetrical patterns independent of radial shape, challenged the necessity of a cold shock pre-pattern for periodic patterns to occur. The observation of labyrinths, which are also asymmetrical, would further

disprove this hypothesis. In any case, further research should investigate this hybrid prepattern-Turing mechanism by integrating the effects of cold shock into the reaction-diffusion dynamics. The cold shock could be modelled in two ways: either by modifying the model parameters at a specific time point t or by starting the simulation with the initial condition being the first stripe generated by the cold shock. This pre-pattern could be generated using image recognition from microscopy images. Implementing such a model could reveal whether pattern formation is more robust with the inclusion of this pre-pattern, thus indicating that robust morphogenesis is likely the result of a combination of multiple mechanisms. Interestingly, although Alan Turing's work was based on a homogeneous starting condition, he recognized the biological unreality of this assumption as he stated: "Most of an organism, most of the time is developing from one pattern to another, rather than from homogeneity into a pattern" (Turing 1952).

This work has significantly contributed to the engineering of reaction-diffusion patterns in synthetic biofilms through the use of predictive models. Engineering these biofilms is essential for deepening our understanding of developmental biology, especially regarding the mechanisms responsible for robust spatial organisation. This research indicates that a plausible explanation could involve a combination of prepatterns, Hopf oscillators, and Turing instabilities. Looking ahead, there is a vast array of potential applications for synthetic patterns. Future endeavours with this circuit will involve substituting fluorescent genes with those necessary for the synthesis of novel materials, thereby creating patterns through biomaterial deposition. Other long-term applications could include the synthesis of tissues for regenerative medicine or the development of organoids for use in organ models or transplants.

CHAPTER 6

Methods

6.1 Finding the steady states: Newton-Raphson method

To obtain the roots of the system, the Newton-Raphson algorithm was implemented. The Newton-Raphson algorithm, also known as Newton's method, is a numerical technique for finding approximate roots of a real-valued function. It starts with an initial guess, which is then refined through iterations. In each iteration, the method uses the current approximation to find a better one, applying the formula

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (6.1)$$

where $f(x)$ is the function and $f'(x)$ its derivative.

This algorithm was implemented in Python using a tolerance value of 10^{-6} and a maximum of 15 iterations after which the algorithm stopped searching for a root. Because the system had the potential for multi-stability, several initial conditions need to be searched to obtain all steady states. In total 100 initial conditions were analysed by the Newton-Raphson to obtain all the roots of the system. The 100 conditions were sampled with Latin-hypercube sampling of a n dimensional space (n being the number of species), from a uniform distribution with a range from 10^{-3} and 10^3 .

6.2 Sampling method

If no parameters from a system are known, sampling of the whole parameter space must be done to understand the system's behaviour. Different methods exist for sampling parameter spaces. Several studies have shown that Latin-hypercube sampling (LHS) has a higher efficiency than grid sampling or random sampling when searching through high-dimensional spaces. The efficiency of LHS over grid sampling might be explained because not all dimensions of the model are important, meaning some parameters might be sloppy. Therefore, not all parameters have to be explored thoroughly as done in grid sampling (Bergstra et al. 2012; Iman 2014). The three sampling regimes are shown in Fig. 6.1A.

In LHS, a distribution and a number of desired samples are given as an input. The algorithm then divides the space sections according to the distribution given (e.g. in a normal distribution, more sections will appear near the mean value). Then, one sample is positioned randomly somewhere in each section. For a 2-dimensional parameter space, no samples can be in the same column or row. This can be scaled up to high multidimensional spaces. The LHS scheme is shown in Fig. 6.1B.

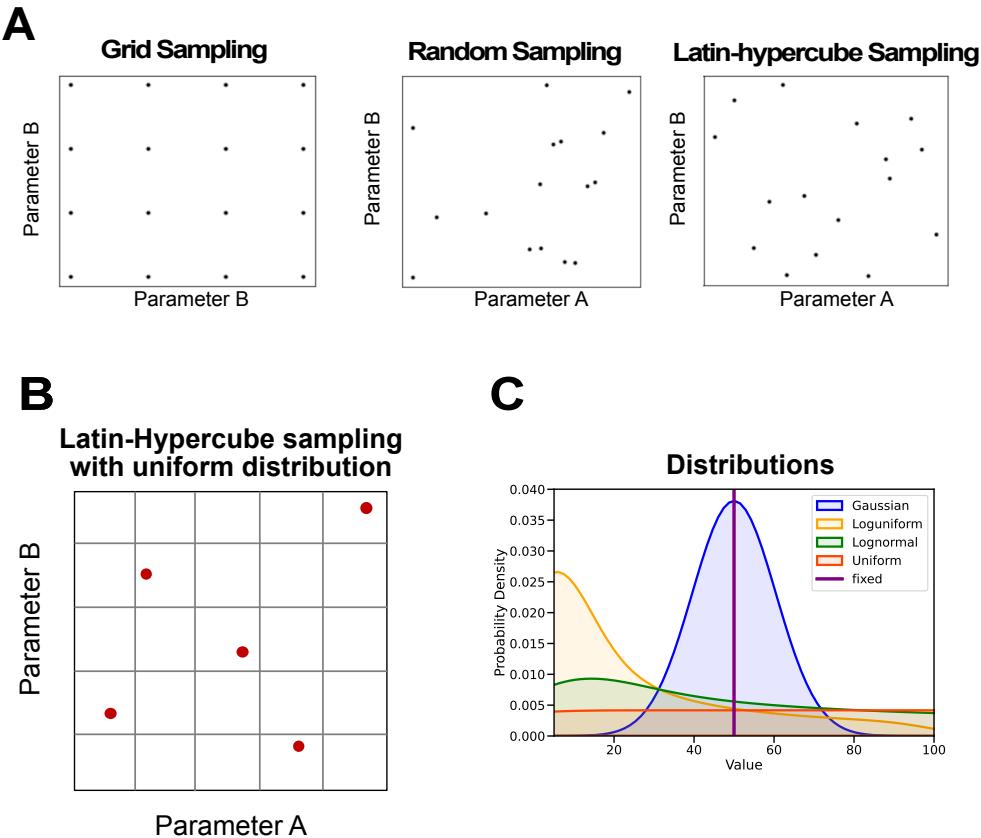


Figure 6.1: Sampling method for high dimensional spaces. (A) Types of potential sampling methods. (B) Latin-Hypercube sampling with uniform distribution for a 2-dimensional parameter space. Space is separated into 5 sections for each parameter, leading to 5 samples (red dot). No sample is present in the same row or column. (C) Parameter distributions used for Latin-hypercube sampling. The four different types of parameters have different distributions depending on the ranges defined. All of them are uniform distributions on a log scale.

The distribution given as an input to the LHS algorithm will be the resulting distribution of your samples. For this search, several distributions are chosen (Fig. 6.1C). Amongst those, we commonly use the uniform distributions in a logarithmic scale (log-uniform distribution). The uniform distribution, although it does not describe many phenomena in biology, can be useful when no prior knowledge is known about the parameters (Frank 2009). The logarithmic component is used to make sure parameters from all scales are represented equally, instead of having a higher frequency of values from larger scales. Log-normal distributions are commonly used for modelling in biology, however, due to the nonexistent prior knowledge on our parameter values, the log-uniform is used instead. When we have some level of certainty about a parameter, a Gaussian distribution is used. Finally, if we are completely sure, a fixed distribution is used.

6.3 PostgreSql database

A PostgreSQL was created from scratch to store and organise all the linear stability analysis and numerical results. This schema is composed of 5 tables as seen in Fig. 6.2. Model parameters and Simulation parameters have information on the kinetic parameters of the model and the simulation parameters of the numerical solver. A set of model parameters or simulation parameters is encoded with a primary key for each table: model_param_id and sim_param_id. Linear stability analysis output contains information on the linear stability analysis output for a specific model_param_id. This information involves the type of dispersion relation, the highest eigenvalue, the LSA estimated wavelength...etc. Simulation output contains information on the one-dimensional (1D) or two-dimensional (2D) time-series data obtained from the simulation. Each set of data has a model_param_id and a sim_param_id associated to it. Finally, the Pattern type table has information on the type of pattern of a specific numerical output. This is based on the defined classifications (e.g. convergence, spatial homogeneity and number of peaks). As with the Simulation output table, each data point has a model_param_id and a sim_param_id associated to it.

This method enables the understanding of relationships between numerical and analytical outputs and between different types of solutions. Additionally, it creates a safe and documented way to store all results in an organised manner.

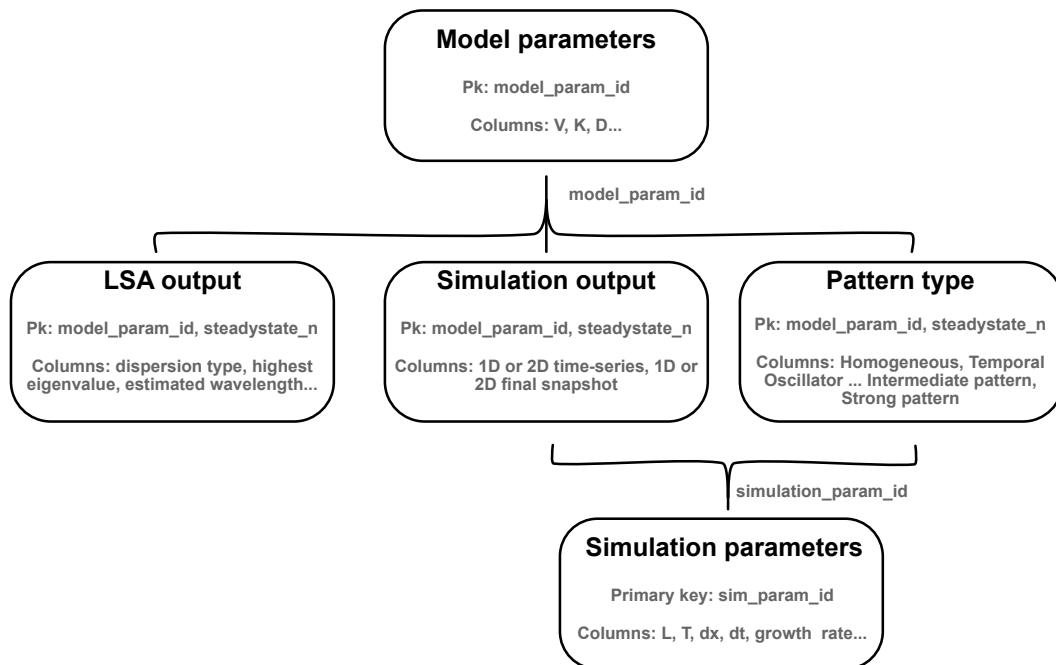


Figure 6.2: PostgreSQL database schema

6.4 Wavelength and convergence time from numerical data

The wavelength was obtained by computing the average distance between peaks in the solution. A wavelength is found for every species of the model, which in the case of Chapter 2 is 2 species. Then the average between the various wavelengths (which are expected to be similar) is computed. The average is found using the *findpeaks* package in Python with all the default parameters and a prominence of 0.5.

The convergence time is obtained by testing when in time the pattern stops being converged starting from the end. A pattern will be considered converged if the last 30 time points for any of the two molecular species fulfils the following condition

$$\frac{\max(U[-30 :]) - \min(U[-30 :])}{\max(U[-30 :])} \leq 0.05 \quad (6.2)$$

6.5 Dispersion peak height optimisation: Adapted random walk Metropolis

In this section, the optimisation of the dispersion peak height using an adapted random walk-metropolis (RWM) algorithm will be explained. The RWM algorithm is a common type of Markov Chain Monte Carlo (MCMC) method that uses a Metropolis Algorithm. The RWM algorithm sampled from the posterior distribution, without getting stuck in local maxima. This algorithm is used in a Bayesian context when trying to fit a model with parameters θ to data D . A probability distribution is obtained, which suggests what parameters θ are better to represent the data D . However, in this case, the aim is not to fit a model to any data D but to maximise the dispersion peak height. Therefore, a variant of the RWM algorithm will be developed as shown in Fig. 6.3.

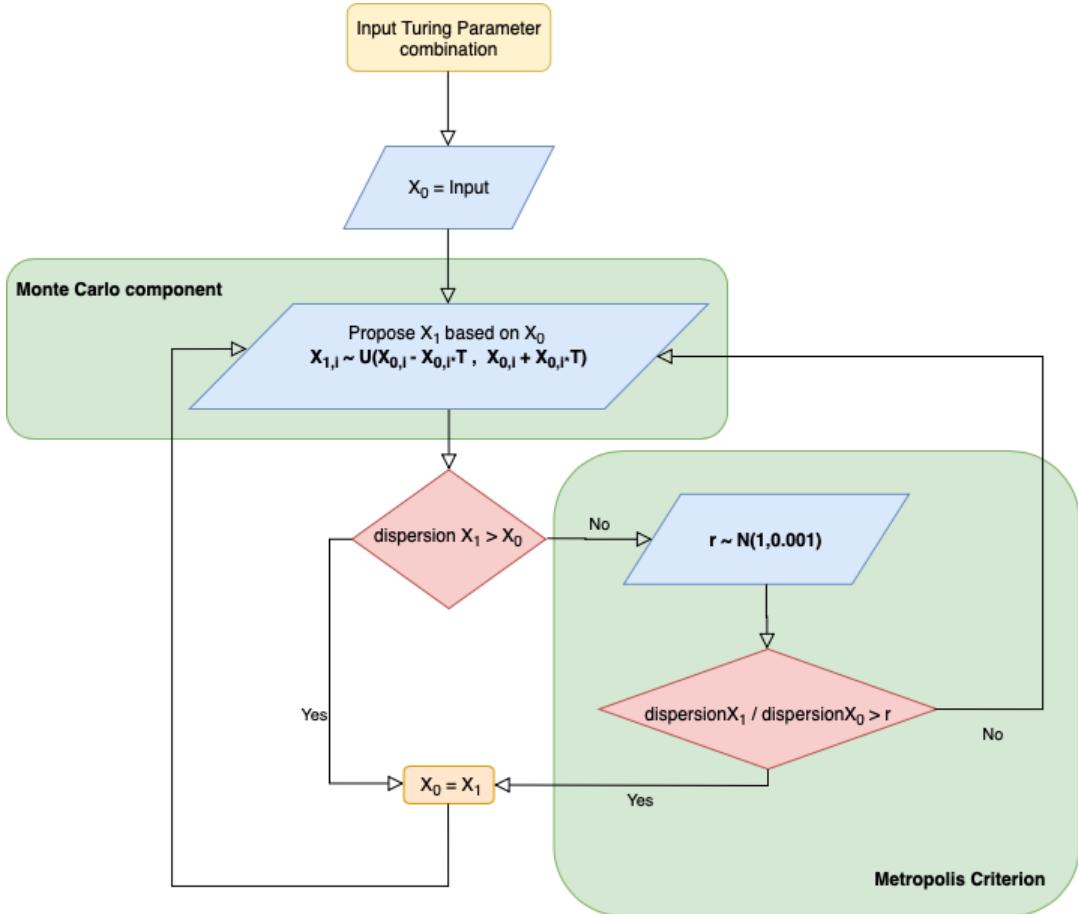


Figure 6.3: Adapted random walk-Metropolis algorithm workflow.

The starting point of the algorithm is to propose a Turing parameter set which will be the starting parameter set to be optimized in the process. From that parameter set, a new parameter set is proposed where all parameters are varied slightly. The variation is chosen randomly from a uniform distribution around the parameter value. The uniform distribution is defined as $U(X_0 - X_0 T, X_0 + X_0 T)$, where X_0 is the initial parameter to be varied and T is a temperature constant that will define the amount of variation to be applied. In this case, $T = 0.1$. In this case, $T = 0.1$. So if $X_0 = 100$, the uniform distribution is $U(90,110)$. This step is done for all parameters of the parameter set at every iteration, producing a new parameter set X_1 . The Markov Chain component is present because the step taken is only dependent on the current state, and not on information prior to that. The Monte Carlo is due to the randomness involved in choosing the new parameter set. In the normal RWM algorithm, the posterior of X_0 and X_1 are compared to see which parameter set is a better fit to the data D . However, for the purpose of optimising dispersion, the posterior is neither available nor relevant. Instead, the dispersion peak height value is used. Once the new step X_1 is taken, the dispersion peak height (d_{X_1}) is calculated and compared to the dispersion peak height of X_0 (d_{X_0}). If the dispersion peak height has improved, $d_{X_1} > d_{X_0}$, the move is accepted and X_1 becomes X_0 . If no improvement has been

made, $d_{X_1} < d_{X_0}$, the Metropolis algorithm comes into place: The ratio of the dispersions is calculated, $r = \frac{d_{X_1}}{d_{X_0}}$ and compared to a normal distribution $N(1, 0.001)$. If the ratio is higher than a random number from the distribution $N(1, 0.001)$, the move is accepted and X_1 becomes X_0 . Otherwise, the move is rejected. This ensures that big decreases in the dispersion peak height ($r \ll 1$) are less likely to be accepted than small decreases ($r \approx 1$). Usually, the RWM uses a distribution $U(0, 1)$ for this step. An optimization with this distribution was attempted, resulting in no significant improvement of the dispersion peak height. Therefore, the variant of the $N(1, 0.001)$ was introduced to ensure a more strict regime is in place, hence reducing the number of accepted negative steps.

6.6 Numerical solution by finite-difference methods

Obtaining a solution for a system of equations can become a complex problem if working with a large system of non-linear PDEs. Because an analytical expression for the solution is almost impossible to obtain, finite-difference methods are used for cases like this one. Finite-difference methods consist of discretising space and time to approximate the PDE system to a system of algebraic equations that can be easily solved by matrix algebra techniques (Morton and Mayers 1994). By discretising time and space, the two independent variables can be expressed as:

$$t_n = n \cdot \Delta t, \quad n = 0, \dots, N - 1 \quad (6.3a)$$

$$x_j = j \cdot \Delta x, \quad j = 0, \dots, J - 1 \quad (6.3b)$$

While Δt and Δx are the time steps and the space steps respectively, N and J are the number of discrete time and space points in our grid. Δt and Δx can be defined as $\Delta t = \frac{T}{N}$ and $\Delta x = \frac{L}{J}$ respectively where T and L are the final time and space values in the grid. The aim is to derive a numerical solution that approximates the unknown analytical solution so $U(j\Delta x, n\Delta t) \approx u(j\Delta x, n\Delta t)$, where U is the analytical solution and u is the numerical solution.

When working with a numerical solver, the solver can perturb the system behaviour due to the effects of the time-step, the integration method or the computer arithmetic. When choosing a scheme to numerically solve a PDE, three different characteristics of the scheme need to be considered: Consistency, stability and convergence. Firstly, for a scheme to be consistent, the truncation error must be reduced as $\Delta t \rightarrow 0$ or/and if $\Delta x \rightarrow 0$. The truncation error results from using a simple approximation to represent an exact mathematical formula. Secondly, the numerical method is said to be stable if the error (truncation or round-off) is not magnified as the number of time steps tends to infinity. Finally, as the Lax equivalence theorem states, the scheme is said to be convergent if both consistency and stability are fulfilled. This means that at any fixed point, if time and space discretisations tend to zero, the numerical

solution will tend towards the exact solution (Smith 1985). The methods chosen to solve this system of equations are Crank-Nicolson (CN) for 1 dimension in space and Alternating Direction Implicit Method (ADI) for 2 dimensions. These methods are chosen because they are both unconditionally stable as shown by von Neumann stability analysis (Strikwerda 2004). The unconditional stability is important to allow for larger Δt and Δx , without getting an amplification of the error. Larger Δt and Δx will result in reduced computational power. Although CN is less computationally expensive than ADI, it becomes extremely complex when scaled up to multiple dimensions. On the other hand, ADI has a simpler structure in 2 dimensions that can be solved easily using the tridiagonal matrix algorithm. Hence, CN is used to obtain 1D space solutions while ADI is used for 2D.

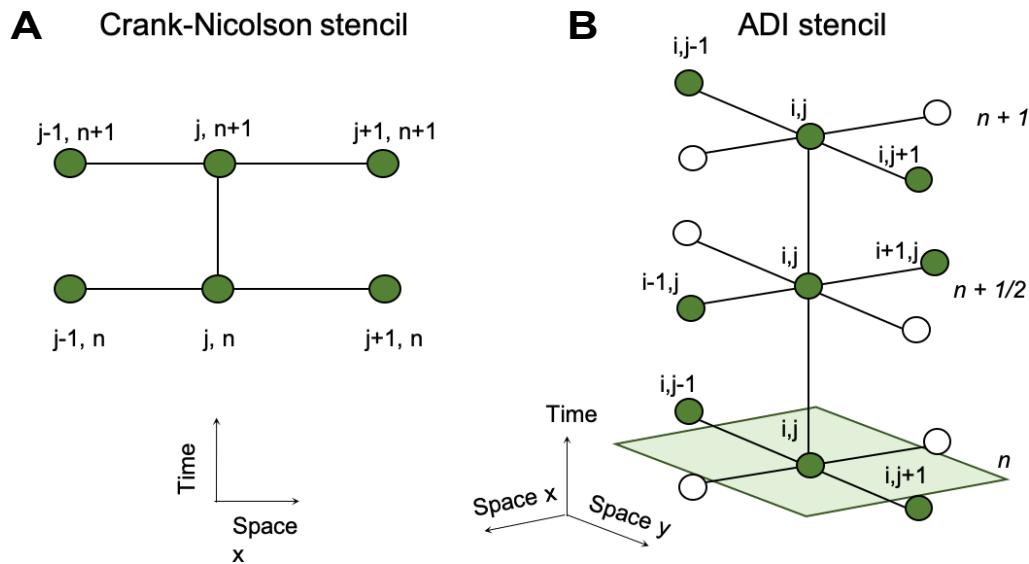


Figure 6.4: **Stencils for numerical solution.** A stencil is a geometric representation with nodes and edges, that represents the points of interest for the numerical approximation. The points of interest, which are the ones present in the equations, are shown in green. j and n are the current space and time points. **(A)** Crank-Nicolson stencil used in 1D numerical simulations. The axes are time and space (x). **(B)** ADI stencil used in 2D numerical simulations. The axes are time and 2D space (x,y).

6.6.1 Crank Nicolson Method

Consider a reaction-diffusion system with one space dimension

$$\frac{\partial u}{\partial t} = f(u) + D \frac{\partial^2 u}{\partial x^2} \quad (6.4)$$

The spatial part of the equation can be approximated to

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_{x=j\Delta x, t=n\Delta t} \approx \frac{1}{2\Delta x^2} \left(U_{j+1}^n - 2U_j^n + U_{j-1}^n + U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1} \right), \quad (6.5)$$

while the production function can be approximated to $f(U_j^n)$. The approximations can be better visualised using the CN stencil (See Fig.6.4a). Applying CN stencil to the grid point (i,j), the reaction-diffusion system can be expressed as

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{D}{2\Delta x^2} (U_{j+1}^n - 2U_j^n + U_{j-1}^n + U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}) + f(U_j^n) \quad (6.6)$$

By reordering this approximation into a linear equation, the resulting problem is defined by a simple linear equation containing matrices A and B. Where $\mathbf{U}^{n+1} = [U_0^n, \dots, U_{J-1}^n]$, the simplified system can be expressed as:

$$\mathbf{U}^{n+1} = A^{-1}(B\mathbf{U}^n + f^n) \quad (6.7)$$

This method simplifies the complex system into a linear system that can be solved numerically. The solution given will be a 1D space solution of the reaction-diffusion system. Although the method is unconditionally stable, the solution can contain oscillations if $\frac{\Delta t}{\Delta x^2} > \frac{1}{2}$ (Trefethen 1996). Therefore, the ratio will be kept below $\frac{1}{2}$ to avoid errors.

Introducing Dirichlet or absorbing boundary conditions

CN method can be implemented with Neumann no-flux boundary conditions or Dirichlet absorbing boundary conditions. For Neumann no-flux boundary conditions:

$$\frac{\delta u}{\delta t} = f(u) + D \frac{\partial^2 u}{\partial x^2}, \quad \left. \frac{\partial u}{\partial x} \right|_{x=0,L} = 0 \quad (6.8)$$

the values of U_j at $j = 0$ and $j = J - 1$ are

$$U_{j=0} = U_{j=-1} \quad \& \quad U_{j=J-1} = U_{j=J} \quad (6.9)$$

These value of $U_{j=0}$ and $U_{j=J-1}$ is replaced into the CN stencil shown in Eq. 6.6. These values are chosen to ensure the derivative at the boundary is zero.

Similarly, Dirichlet absorbing boundary conditions are represented by the following system

$$\frac{\delta u}{\delta t} = f(u) + D \frac{\partial^2 u}{\partial x^2}, \quad U \Big|_{x=0,L} = 0 \quad (6.10)$$

and have values of U_j at $j = -1$ and $j = J$ such as

$$U_{j=-1} = 0 \quad \& \quad U_{j=J} = 0 \quad (6.11)$$

These values of $U_{j=0}$ and $U_{j=J-1}$ is replaced into the CN stencil shown in Eq. 6.6. These values are chosen to ensure the value at the boundary is zero.

6.6.2 Alternating Direction Implicit Method

As done in the CN scheme, a reaction-diffusion system and its boundary conditions will be considered. However, in this case, two spatial dimensions will be introduced.

$$\frac{\delta u}{\delta t} = f(u) + D \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \quad \frac{\partial u}{\partial x} \Big|_{x=0,L} = 0 \quad \frac{\partial u}{\partial y} \Big|_{y=0,L} = 0 \quad (6.12)$$

If the CN stencil is applied to this 2D spatial problem, the system would contain banded matrices on the right and left-hand sides, which would be very expensive to invert. ADI offers an alternative in which tridiagonal matrices are inverted instead of banded matrices (less computational power required). The characteristic of ADI is the time step Δt is split into two, and each half-time step is computed. This means, that to compute the change at each time step, first we compute $U_{i,j}^{n+1/2}$ and from there, $U_{i,j}^{n+1}$ is calculated. This results in two different equations:

$$\begin{aligned} \frac{U_{i,j}^{n+1/2} - U_{i,j}^n}{\Delta t/2} &= \frac{D}{2\Delta x^2} \left(U_{i+1,j}^{n+1/2} - 2U_{i,j}^{n+1/2} + U_{i-1,j}^{n+1/2} \right) \\ &\quad + \frac{D}{2\Delta y^2} \left(U_{i,j+1}^n - 2U_{i,j}^n + U_{i,j-1}^n \right) + \Delta t f(U_{i,j}^n) \end{aligned} \quad (6.13a)$$

$$\begin{aligned} \frac{U_{i,j}^{n+1} - U_{i,j}^{n+1/2}}{\Delta t/2} &= \frac{D}{2\Delta x^2} \left(U_{i+1,j}^{n+1/2} - 2U_{i,j}^{n+1/2} + U_{i-1,j}^{n+1/2} \right) \\ &\quad + \frac{D}{2\Delta y^2} \left(U_{i,j+1}^{n+1} - 2U_{i,j}^{n+1} + U_{i,j-1}^{n+1} \right) + \Delta t f(U_{i,j}^{n+1/2}) \end{aligned} \quad (6.13b)$$

In the first half-time step (Equation 46a), the x derivative is taken implicitly, and in the second half-time step (Equation 46b), the y derivative is taken implicitly. As done in CN, the approximation is reordered into a linear system. Two families of linear systems appear:

$$A\mathbf{U}_{x,i}^{n+1/2} = \mathbf{b}_i + \mathbf{f}(\Delta t \mathbf{U}_{x,i}^n), \quad i = 0, \dots, I-1 \quad (6.14a)$$

$$C\mathbf{U}_{y,j}^{n+1} = \mathbf{d}_j + \mathbf{f}(\Delta t \mathbf{U}_{y,j}^{n+1/2}), \quad j = 0, \dots, J-1 \quad (6.14b)$$

Again, this method also simplifies a complex system into a linear system that can be solved numerically, as in CN. However, this method allows for the introduction of a new spatial dimension and therefore produces a 2D spatial solution. The workings of this method can be better understood with the ADI stencil (See Fig. 6.4b). ADI will be used to visualise patterns in 2D.

Introducing Dirichlet or absorbing boundary conditions

As with the CN method, ADI can also be implemented with Neumann no-flux boundary conditions or Dirichlet absorbing boundary conditions. The implementation is the same as in CN (See Section 6.6.1). However, in this case, boundaries are considered for both i and j , both in steps U^n and $U^{n+1/2}$.

6.6.3 Analysis of numerical solution

The speed of pattern formation and pattern wavelength are identified by performing additional analysis on the 1D numerical data.

Time for pattern convergence

The development of the pattern follows a certain behaviour: The molecule concentrations are initially homogeneous; then a pattern gets formed progressively; and finally, the pattern is in its final state and the solution remains constant. The time for pattern convergence is measured by comparing the solution (one point in space) at every time point to the solution at the final time point. If the difference is smaller than a tolerance value of 10^{-4} that time point is taken as the convergence time point where the pattern has finished to develop.

Wavelength prediction from numerical solution

The findpeaks package is used from the *scipy.signal* Python library. All peaks in the final time point of the 1D simulation are found through the findpeaks package. The average distance between peaks is taken and that distance is averaged throughout the 6 species.

6.7 Image recognition and modelling for confocal microscopy data

6.7.1 Tissue area recognition

From a confocal microscopy image, we want to obtain the shape of the biofilm. The image is opened and processed using the PIL package in Python. After splitting the image into RGB, the blue channel is selected. Because the cells show red and green fluorescence, the blue channel is minimal in the tissue compared to the agar. A Gaussian filter is used to smooth the image and get a continuous biofilm. Blue channel pixels with values above a threshold (20) are selected and fixed to 0 in the

"shape matrix". The rest of the pixels are set to 1 which determine the area of the biofilm.

6.7.2 Plotting superposed numerical solution as confocal microscopy results

To compare the model and experiments, the six variable solutions stemming from the six-equation model have to be translated into the red-green superposed images obtained from the confocal microscopy. The red and green channels are obtained from the solutions of species D (LacI) and E (cI^*), which are assumed to have a linear relationship with GFP and mCherry respectively (Eq 3.28) through parameter α . These two channels are transformed using min-max scaling to range from 0-255:

$$X_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \cdot 255 \quad (6.15)$$

This way, we end up with a red and green channel ranging from 0-255 (Fig. 6.5A,B). This can be plotted as an RGB image where the blue channel is zero (Fig. 6.5C). This normalisation also occurs when adjusting the gain for each fluorophore during the confocal microscopy. Therefore, as long as the linear relationship between LacI and GFP fluorescence holds true (same for mCherry), the normalised superposition we use for plotting should also be correct. In some instances, this linear relationship is not correct. Therefore, future studies to characterise fluorescence output and protein concentrations are needed.

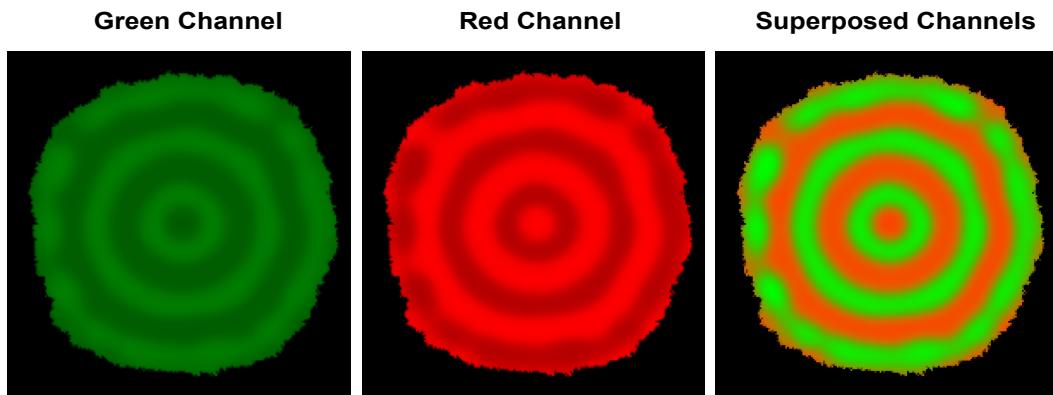


Figure 6.5: Fluorescent channels from obtained from the simulation

6.8 Experimental methods

6.8.1 Transformation through electroporation

1 μL of DNA containing the 4 plasmids was added to 25 μL of electrocompetent MK01 *E. coli* cells. These four plasmids make up the full gene circuit seen in Fig. 3.1. Details of these plasmids are in 6.1. This mixture was placed on ice for a minute

before being transferred to an electroporation cuvette. The cells were then electroporated at $100\ \Omega$, $25\ \mu F$, and $1.8\ kV$. Post-electroporation, the cells were immediately suspended in $500\ \mu L$ of SOC medium inside the cuvette. This suspension was then moved to an aeration tube and allowed to recover for 1h shaking at $37\ ^\circ C$, 220rpm. Finally, $100\mu l$ of the culture was plated and grown in LB agar plates with the necessary antibiotics and grown overnight at $37\ ^\circ C$.

Plasmid	Resistance	Contains ...	Copy number
pCOLA	Kan 50	Node A	Medium (20 – 40) 21
pCDF	Spec 50	Node B	Medium (20 – 40) 21
pET	Amp 100	Node C	Medium (20 – 40) 21
pCC1	CA 10	Regulator cassette	Single copy 22

Table 6.1: Circuit plasmids built by Dr. Jure Tica and Tong Zhu. Each circuit node is encoded in a different cassette.

6.8.2 Microscopy

After electroporation, transformed cells were grown in 2xYT medium (Sigma-Aldrich Y1003) with the required antibiotics to an OD600 of 1.5. They were then diluted in fresh 2xYT by a factor of 10^4 . 6 well MatTek plates with glass coverslips on the bottom were used for imaging. 1.4% (w/v) agar (Sigma-Aldrich A5306) was plated in the 5 wells containing the desired antibiotics and inducers on top of the coverslip. The diluted cells were then added on the agar and spread with beads (Novagen 71013). The plates were then sealed in parafilm to avoid drying out of the agar and covered in aluminium foil to prevent photobleaching of the fluorescent proteins (in case the signal of the emergent pattern is very weak). The plates were then incubated at $37\ ^\circ C$ for 4 days and imaged daily. Colonies grew over time in the wells of the MatTek plates as shown in Fig. 6.6.



Figure 6.6: Bacterial colonies with synthetic gene circuit grown on 6-well MatTek plates.

These colonies were then imaged in the confocal Leica SP8 microscope with a 10x objective. Confocal microscopy was carried out in the Facility for Imaging by Light Microscopy (FILM) at Imperial College London on a Leica SP8. FILM is part-supported by funding from the Wellcome Trust and BBSRC. More details of confocal imaging protocol including imaging parameters can be found in (Tica 2020).

The opening and superposing of the confocal microscopy images is performed in Fiji (ImageJ v2.1.0). None of the images are subjected to post-processing by linear or non-linear colour map transformations.

Bibliography

- Alessio, Benjamin M. and Ankur Gupta (Nov. 2023). "Diffusiophoresis-enhanced Turing patterns". In: *Science Advances* 9 (45). ISSN: 2375-2548. DOI: 10.1126/sciadv.adj2457.
- Andersen, Jens Bo, Claus Sternberg, Lars Kongsbak Poulsen, Sara Petersen Bjørn, Michael Givskov, and Søren Molin (June 1998). "New Unstable Variants of Green Fluorescent Protein for Studies of Transient Gene Expression in Bacteria". In: *Applied and Environmental Microbiology* 64 (6), pp. 2240–2246. ISSN: 0099-2240. DOI: 10.1128/AEM.64.6.2240-2246.1998.
- Arcuri, P. and J.D. Murray (July 1986). "Pattern sensitivity to boundary and initial conditions in reaction-diffusion models". In: *Journal of Mathematical Biology* 24 (2). ISSN: 0303-6812. DOI: 10.1007/BF00275996.
- Asakura, Kouichi, Ryo Konishi, Tomomi Nakatani, Takaya Nakano, and Masazumi Kamata (2011). "Turing pattern formation by the CIMA reaction in a chemical system consisting of quaternary alkyl ammonium cationic groups". In: *Journal of Physical Chemistry B* 115 (14), pp. 3959–3963. ISSN: 15205207. DOI: 10.1021/jp111584u.
- Babic, Andrea C. and John W. Little (Nov. 2007). "Cooperative DNA binding by CI repressor is dispensable in a phage lambda variant". In: *Proceedings of the National Academy of Sciences* 104 (45), pp. 17741–17746. ISSN: 0027-8424. DOI: 10.1073/pnas.0602223104.
- Barbier, Içvara, Rubén Perez-Carrasco, and Yolanda Schaeferli (2020). "Controlling spatiotemporal pattern formation in a concentration gradient with a synthetic toggle switch". In: *Molecular Systems Biology* 16 (6), pp. 1–15. ISSN: 1744-4292. DOI: 10.15252/msb.20199361.
- Basu, Subhayu, Yoram Gerchman, Cynthia H. Collins, Frances H. Arnold, and Ron Weiss (Apr. 2005). "A synthetic multicellular system for programmed pattern formation". In: *Nature* 434 (7037), pp. 1130–1134. ISSN: 00280836. DOI: 10.1038/nature03461.

- Bergstra, James, James Bergstra@umontreal Ca, and Yoshua Bengio@umontreal Ca (2012). *Random Search for Hyper-Parameter Optimization Yoshua Bengio*.
- Berry, Scott and Lucas Pelkmans (2022). "Mechanisms of cellular mRNA transcript homeostasis". In: *Trends in Cell Biology*.
- Biancalani, Tommaso, Duccio Fanelli, and Francesca Di Patti (Apr. 2010). "Stochastic Turing patterns in the Brusselator model". In: *Physical Review E* 81 (4), p. 046215. ISSN: 1539-3755. DOI: 10.1103/PhysRevE.81.046215.
- Blest., A David (1957). "The function of eyespot patterns in the Lepidoptera". In: 209–258.
- Boehm, Christian R., Paul K. Grant, and Jim Haseloff (2018). "Programmed hierarchical patterning of bacterial populations". In: *Nature Communications* 9 (1), pp. 1–10. ISSN: 20411723. DOI: 10.1038/s41467-018-03069-3.
- Bremer, Hans and Patrick P. Dennis (Feb. 2008). "Modulation of Chemical Composition and Other Parameters of the Cell at Different Exponential Growth Rates". In: *EcoSal Plus* 3 (1). ISSN: 2324-6200. DOI: 10.1128/ecosal.5.2.3.
- Briscoe, James and Stephen Small (2015). *Morphogen rules: Design principles of gradient-mediated embryo patterning*. DOI: 10.1242/dev.129452.
- Buchler, Nicolas E, Ulrich Gerland, and Terence Hwa (2005). "Nonlinear protein degradation and the function of genetic circuits". In: *Proceedings of the National Academy of Sciences* 102.27, pp. 9559–9564.
- Butler, Thomas and Nigel Goldenfeld (Sept. 2009). "Robust ecological pattern formation induced by demographic noise". In: *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* 80 (3). ISSN: 15393755. DOI: 10.1103/PhysRevE.80.030902.
- (July 2011). "Fluctuation-driven Turing patterns". In: *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* 84 (1), p. 011112. ISSN: 15393755. DOI: 10.1103/PhysRevE.84.011112.
- Butzin, Nicholas C. and William H. Mather (2018). "Crosstalk between Diverse Synthetic Protein Degradation Tags in Escherichia coli". In: *ACS Synthetic Biology* 7 (1), pp. 54–62. ISSN: 21615063. DOI: 10.1021/acssynbio.7b00122.
- Cachat, Elise, Weijia Liu, Kim C Martin, Xiaofei Yuan, Huabing Yin, Peter Hohenstein, and Jamie A Davies (2016). "2-and 3-dimensional synthetic large-scale de novo patterning by mammalian cells through phase separation". In: *Scientific reports* 6.1, p. 20664.
- Cao, Yangxiaolu, Yaying Feng, Marc D Ryser, Kui Zhu, Gregory Herschlag, Changyong Cao, Katherine Marusak, Stefan Zauscher, and Lingchong You (Nov. 2017). "Programmable assembly of pressure sensors using pattern-forming bacteria". In: *Nature Biotechnology* 35 (11), pp. 1087–1093. ISSN: 1087-0156. DOI: 10.1038/nbt.3978.
- Cao, Yangxiaolu, Marc D. Ryser, Stephen Payne, Bochong Li, Christopher V. Rao, and Lingchong You (2016). "Collective Space-Sensing Coordinates Pattern Scaling in Engineered Bacteria". In: *Cell* 165 (3), pp. 620–630. ISSN: 10974172. DOI: 10.1016/j.cell.2016.03.006.

- Carballido-Landeira, Jorge, Vladimir K. Vanag, and Irving R. Epstein (2010). "Patterns in the Belousov-Zhabotinsky reaction in water-in-oil microemulsion induced by a temperature gradient". In: *Physical Chemistry Chemical Physics* 12 (15). ISSN: 14639076. DOI: 10.1039/b919278f.
- Caro, TIM (2005). "The adaptive significance of coloration in mammals". In: *BioScience* 55.2, pp. 125–136.
- Castets, V, E Dulos, and P De Kepper (1990). "Experimental Evidence of a Sustained Standing Turing-Type Nonequilibrium Chemical Pattern". In: *Physical Review Letters* 64 (24).
- Ceroni, Francesca, Rhys Algar, Guy Bart Stan, and Tom Ellis (May 2015). "Quantifying cellular capacity identifies gene expression designs with reduced burden". In: *Nature Methods* 12 (5), pp. 415–418. ISSN: 15487105. DOI: 10.1038/nmeth.3339.
- Chrisman, Lonnie (2014). *Latin Hypercube vs. Monte Carlo Sampling 3 Comments / Analytica tips, Analytics Methods, Risk and uncertainty*.
- Crampin, Edmund J, William W Hackborn, and Philip K Maini (2002). "Pattern formation in reaction-diffusion models with nonuniform domain growth". In: *Bulletin of mathematical biology* 64, pp. 747–769.
- Csibra, Eszter and Guy-Bart Stan (2022). "Absolute protein quantification using fluorescence measurements with FPCountR". In: *Nature Communications* 13.1, p. 6600.
- Danino, Tal, Octavio Mondragón-Palomino, Lev Tsimring, and Jeff Hasty (2010). "A synchronized quorum of genetic clocks". In: *Nature* 463 (7279), pp. 326–330. ISSN: 00280836. DOI: 10.1038/nature08753.
- Diambra, Luis, Vivek Raj Senthivel, Diego Barcena Menendez, and Mark Isalan (Feb. 2015). "Cooperativity to increase turing pattern space for synthetic biology". In: *ACS Synthetic Biology* 4 (2), pp. 177–186. ISSN: 21615063. DOI: 10.1021/sb500233u.
- Din, M. Omar, Aida Martin, Ivan Razinkov, Nicholas Csicsery, and Jeff Hasty (May 2020). "Interfacing gene circuits with microelectronics through engineered population dynamics". In: *Science Advances* 6 (21). ISSN: 2375-2548. DOI: 10.1126/sciadv.aaz8344.
- Du, Pei et al. (2020). "De novo design of an intercellular signaling toolbox for multi-channel cell-cell communication and biological computation". In: *Nature Communications* 11 (1), pp. 1–11. ISSN: 20411723. DOI: 10.1038/s41467-020-17993-w.
- Economou, Andrew D, Atsushi Ohazama, Thantrira Porntaveetus, Paul T Sharpe, Shigeru Kondo, M Albert Basson, Amel Gritli-Linde, Martyn T Cobourne, and Jeremy B A Green (Mar. 2012). "Periodic stripe formation by a Turing mechanism operating at growth zones in the mammalian palate". In: *Nature Genetics* 44 (3), pp. 348–351. ISSN: 1061-4036. DOI: 10.1038/ng.1090.
- Frank, S. A. (Aug. 2009). "The common patterns of nature". In: *Journal of Evolutionary Biology* 22 (8), pp. 1563–1585. ISSN: 1010-061X. DOI: 10.1111/j.1420-9101.2009.01775.x.
- Gardner, Martin (1970). "Mathematical games". In: *Scientific american* 222.6, pp. 132–140.

- Gestel, Jordi van et al. (Apr. 2021). "Short-range quorum sensing controls horizontal gene transfer at micron scale in bacterial communities". In: *Nature Communications* 12 (1), p. 2324. ISSN: 2041-1723. DOI: 10.1038/s41467-021-22649-4.
- Gierer, A and H Meinhardt (1972). *A Theory of Biological Pattern Formation*.
- Glendinning, Paul. (1994). *Stability, instability, and chaos : an introduction to the theory of nonlinear differential equations*. Cambridge University Press, p. 388. ISBN: 0521415535.
- Glover, James D. et al. (Mar. 2023). "The developmental basis of fingerprint pattern formation and variation". In: *Cell* 186 (5), 940–956.e20. ISSN: 00928674. DOI: 10.1016/j.cell.2023.01.015.
- Grant, Paul K. et al. (Dec. 2020). "Interpretation of morphogen gradients by a synthetic bistable circuit". In: *Nature Communications* 11 (1), p. 5545. ISSN: 2041-1723. DOI: 10.1038/s41467-020-19098-w.
- Green, Jeremy B. A. and James Sharpe (Apr. 2015a). "Positional information and reaction-diffusion: two big ideas in developmental biology combine". In: *Development* 142 (7), pp. 1203–1211. ISSN: 1477-9129. DOI: 10.1242/dev.114991.
- Green, Jeremy BA and James Sharpe (2015b). "Positional information and reaction-diffusion: two big ideas in developmental biology combine". In: *Development* 142.7, pp. 1203–1211.
- Gregor, Thomas, David W Tank, Eric F Wieschaus, and William Bialek (2007). "Probing the limits to positional information". In: *Cell* 130.1, pp. 153–164.
- Grimm, Oliver, Mathieu Coppey, and Eric Wieschaus (2010). "Modelling the Bicoid gradient". In: *Development* 137.14, pp. 2253–2264.
- Haas, Pierre A. and Raymond E. Goldstein (2021). "Turing's Diffusive Threshold in Random Reaction-Diffusion Systems". In: *Physical Review Letters* 126 (23). ISSN: 10797114. DOI: 10.1103/PhysRevLett.126.238101.
- Hecht, Ariel, Jeff Glasgow, Paul R. Jaschke, Lukmaan A. Bawazer, Matthew S. Munson, Jennifer R. Cochran, Drew Endy, and Marc Salit (Apr. 2017). "Measurements of translation initiation from all 64 codons in *E. coli*". In: *Nucleic Acids Research* 45 (7), pp. 3615–3626. ISSN: 0305-1048. DOI: 10.1093/nar/gkx070.
- Hiscock, Tom W and Sean G Megason (2015). "Mathematically guided approaches to distinguish models of periodic patterning". In: *Development* 142.3, pp. 409–419.
- Horváth, Judit, István Szalai, and Patrick De Kepper (2009). "An experimental design method leading to chemical turing patterns". In: *Science* 324 (5928), pp. 772–775. ISSN: 00368075. DOI: 10.1126/science.1169973.
- Huang, Sui and Donald E Ingber (2000). "Shape-dependent control of cell growth, differentiation, and apoptosis: switching between attractors in cell regulatory networks". In: *Experimental cell research* 261.1, pp. 91–103.
- Iman, Ronald L. (Sept. 2014). "Latin Hypercube Sampling". In: Wiley. DOI: 10.1002/9781118445112.stat03803.

- Jenssen, Tor-Kristian, Astrid Laegreid, Jan Komorowski, and Eivind Hovig (2001). "A literature network of human genes for high-throughput analysis of gene expression". In: *Nature genetics* 28.1, pp. 21–28.
- Jia, Dongxuan, Daniel Dajusta, and Ramsey A Foty (2007). "Tissue surface tensions guide in vitro self-assembly of rodent pancreatic islet cells". In: *Developmental Dynamics: An Official Publication of the American Association of Anatomists* 236.8, pp. 2039–2049.
- Jing, Xiao hong, Sheng mei Zhou, Wei qing Wang, and Yu Chen (May 2006). "Mechanisms underlying long- and short-range nodal signaling in Zebrafish". In: *Mechanisms of Development* 123 (5), pp. 388–394. ISSN: 09254773. DOI: 10.1016/j.mod.2006.03.006.
- Karig, David, K. Michael Martini, Ting Lu, Nicholas A. DeLateur, Nigel Goldenfeld, and Ron Weiss (June 2018). "Stochastic Turing patterns in a synthetic bacterial population". In: *Proceedings of the National Academy of Sciences of the United States of America* 115 (26), pp. 6572–6577. ISSN: 10916490. DOI: 10.1073/pnas.1720770115.
- Kaufmann, Gunnar F et al. (2005). "Revisiting quorum sensing. Discovery of additional chemical and biological functions for 3-oxo-N-acylhomoserine lactones". In: *Proceedings of the National Academy of Sciences* 102.2, pp. 309–314.
- Kintaka, Reiko et al. (2020). "Genetic profiling of protein burden and nuclear export overload". In: *ELife* 9, e54080.
- Klika, Václav and Eamonn A. Gaffney (2017). "History dependence and the continuum approximation breakdown: The impact of domain growth on Turing's instability". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473 (2199). ISSN: 14712946. DOI: 10.1098/rspa.2016.0744.
- Kondo, Shigeru and Rihito Asai (Aug. 1995). "A reaction-diffusion wave on the skin of the marine angelfish Pomacanthus". In: *Nature* 376 (6543), pp. 765–768. ISSN: 0028-0836. DOI: 10.1038/376765a0.
- Kondo, Shigeru and Takashi Miura (Sept. 2010). *Reaction-diffusion model as a framework for understanding biological pattern formation*. DOI: 10.1126/science.1179047.
- Kong, Wentao, Andrew E. Blanchard, Chen Liao, and Ting Lu (2017). "Engineering robust and tunable spatial structures with synthetic gene circuits". In: *Nucleic Acids Research* 45 (2), pp. 1005–1014. ISSN: 13624962. DOI: 10.1093/nar/gkw1045.
- Konow, Christopher, Noah H. Somberg, Jocelyne Chavez, Irving R. Epstein, and Milos Dolnik (2019). "Turing patterns on radially growing domains: experiments and simulations". In: *Physical Chemistry Chemical Physics* 21 (12), pp. 6718–6724. ISSN: 14639076. DOI: X.
- Korvasová, K., E.A. Gaffney, P.K. Maini, M.A. Ferreira, and V. Klika (Feb. 2015). "Investigating the Turing conditions for diffusion-driven instability in the presence of a binding immobile substrate". In: *Journal of Theoretical Biology* 367, pp. 286–295. ISSN: 002225193. DOI: 10.1016/j.jtbi.2014.11.024.
- Krause, Andrew L., Meredith A. Ellis, and Robert A. Van Gorder (2019). "Influence of Curvature, Growth, and Anisotropy on the Evolution of Turing Patterns on

- Growing Manifolds". In: *Bulletin of Mathematical Biology* 81 (3), pp. 759–799. ISSN: 15229602. DOI: 10.1007/s11538-018-0535-y.
- Krause, Andrew L., Eamonn A. Gaffney, Thomas Jun Jewell, Václav Klika, and Benjamin J. Walker (Aug. 2023). "Turing instabilities are not enough to ensure pattern formation". In.
- Krause, Andrew L., Václav Klika, Jacob Halatek, Paul K. Grant, Thomas E. Woolley, Neil Dalchau, and Eamonn A. Gaffney (Oct. 2020). "Turing Patterning in Stratified Domains". In: *Bulletin of Mathematical Biology* 82 (10), p. 136. ISSN: 0092-8240. DOI: 10.1007/s11538-020-00809-9.
- Krause, Andrew L., Václav Klika, Philip K. Maini, Denis Headon, and Eamonn A. Gaffney (July 2021). "Isolating Patterns in Open Reaction–Diffusion Systems". In: *Bulletin of Mathematical Biology* 83 (7), p. 82. ISSN: 0092-8240. DOI: 10.1007/s11538-021-00913-4.
- Kügler, Philipp, Erwin Gaubitzer, and Stefan Müller (2009). "Parameter identification for chemical reaction systems using sparsity enforcing regularization: A case study for the chlorite-iodide reaction". In: *Journal of Physical Chemistry A* 113 (12). ISSN: 10895639. DOI: 10.1021/jp808792u.
- Langhe, Stijn P. De, Frédéric G. Sala, Pierre-Marie Del Moral, Timothy J. Fairbanks, Kenneth M. Yamada, David Warburton, Robert C. Burns, and Saverio Bellusci (Jan. 2005). "Dickkopf-1 (DKK1) reveals that fibronectin is a major target of Wnt signalling in branching morphogenesis of the mouse embryonic lung". In: *Developmental Biology* 277 (2), pp. 316–331. ISSN: 00121606. DOI: 10.1016/j.ydbio.2004.09.023.
- Laurent, Michel and Nicolas Kellershohn (1999). "Multistability: a major means of differentiation and evolution in biological systems". In: *Trends in biochemical sciences* 24.11, pp. 418–422.
- Leite, Maria Conceição A and Yunjiao Wang (2009). "Multistability, oscillations and bifurcations in feedback loops". In: *Mathematical Biosciences & Engineering* 7.1, pp. 83–97.
- Lengyel, I. and I. R. Epstein (1992). "A chemical approach to designing Turing patterns in reaction-diffusion systems". In: *Proceedings of the National Academy of Sciences of the United States of America* 89 (9), pp. 3977–3979. ISSN: 00278424. DOI: 10.1073/pnas.89.9.3977.
- Liu, Ruey Tarn, Sy Sang Liaw, and Philip K. Maini (2007). "Oscillatory turing patterns in a simple reaction-diffusion system". In: *Journal of the Korean Physical Society* 50 (1 I), pp. 234–238. ISSN: 03744884.
- Madzvamuse, Anotida, Eamonn A. Gaffney, and Philip K. Maini (2010). "Stability analysis of non-autonomous reaction-diffusion systems: The effects of growing domains". In: *Journal of Mathematical Biology* 61 (1), pp. 133–164. ISSN: 03036812. DOI: 10.1007/s00285-009-0293-4.
- Maini, P. K. and M. R. Myerscough (1997). "Boundary-driven instability". In: *Applied Mathematics Letters* 10 (1), pp. 1–4. ISSN: 08939659. DOI: 10.1016/S0893-9659(96)00101-2.

- Maini, Philip K., Robert Dillon, and Hans G. Othmer (1993). "Steady-State Patterns in a Reaction Diffusion System with Mixed Boundary Conditions". In: Springer US, pp. 231–235. DOI: 10.1007/978-1-4615-2433-5_21.
- Maini, Philip K., Thomas E. Woolley, Ruth E. Baker, Eamonn A. Gaffney, and S. Seirin Lee (Aug. 2012). "Turing's model for biological pattern formation and the robustness problem". In: *Interface Focus* 2 (4), pp. 487–496. ISSN: 2042-8898. DOI: 10.1098/rsfs.2011.0113.
- Marcon, Luciano, Xavier Diego, James Sharpe, and Patrick Müller (2016). "High-throughput mathematical analysis identifies turing networks for patterning with equally diffusing signals". In: *eLife* 5 (APRIL2016). ISSN: 2050084X. DOI: 10.7554/eLife.14022.
- Matsushita, Mitsugu and Hiroshi Fujikawa (1990). *Diffusion-Limited Growth in bacterial colony formation*.
- May, Robert M. (Aug. 1972). "Will a Large Complex System be Stable?" In: *Nature* 238 (5364), pp. 413–414. ISSN: 0028-0836. DOI: 10.1038/238413a0.
- Metzger, Ross J, Ophir D Klein, Gail R Martin, and Mark A Krasnow (2008). "The branching programme of mouse lung development". In: *Nature* 453.7196, pp. 745–750.
- Meyer, Adam J., Thomas H. Segall-Shapiro, Emerson Glassey, Jing Zhang, and Christopher A. Voigt (Feb. 2019). "Escherichia coli "Marionette" strains with 12 highly optimized small-molecule sensors". In: *Nature Chemical Biology* 15 (2), pp. 196–204. ISSN: 15524469. DOI: 10.1038/s41589-018-0168-3.
- Momb, Jessica, Canhui Wang, Dali Liu, Pei W. Thomas, Gregory A. Petsko, Hua Guo, Dagmar Ringe, and Walter Fast (July 2008). "Mechanism of the Quorum-Quenching Lactonase *AiiA* from *Bacillus thuringiensis*. Substrate Modeling and Active Site Mutations". In: *Biochemistry* 47 (29), pp. 7715–7725. ISSN: 0006-2960. DOI: 10.1021/bi8003704.
- Morgunova, Ekaterina and Jussi Taipale (Dec. 2017). "Structural perspective of cooperative transcription factor binding". In: *Current Opinion in Structural Biology* 47, pp. 1–8. ISSN: 0959440X. DOI: 10.1016/j.sbi.2017.03.006.
- Moris, Naomi, Cristina Pina, and Alfonso Martinez Arias (2016). "Transition states and cell fate decisions in epigenetic landscapes". In: *Nature Reviews Genetics* 17.11, pp. 693–703.
- Morton, Keith William and David Francis Mayers (1994). *Numerical solution of partial differential equations : an introduction*. Cambridge : Cambridge university press, 1994.
- Murray, J. D (2002). *Mathematical Biology II: Spatial Models and Biomedical Applications*. Third Edit. Springer.
- Murray, J. D., G. F. Oster, and A. K. Harris (1983). "A mechanical model for mesenchymal morphogenesis". In: *Journal of Mathematical Biology*. ISSN: 03036812. DOI: 10.1007/BF00276117.

- Nakamura, Tetsuya, Naoki Mine, Etsushi Nakaguchi, Atsushi Mochizuki, Masamichi Yamamoto, Kenta Yashiro, Chikara Meno, and Hiroshi Hamada (Oct. 2006). "Generation of Robust Left-Right Asymmetry in the Mouse Embryo Requires a Self-Enhancement and Lateral-Inhibition System". In: *Developmental Cell* 11 (4), pp. 495–504. ISSN: 15345807. DOI: 10.1016/j.devcel.2006.08.002.
- Nielsen, Alec A.K., Bryan S. Der, Jonghyeon Shin, Prashant Vaidyanathan, Vanya Paralanov, Elizabeth A. Strychalski, David Ross, Douglas Densmore, and Christopher A. Voigt (Apr. 2016). "Genetic circuit design automation". In: *Science* 352 (6281), aac7341–aac7341. ISSN: 10959203. DOI: 10.1126/science.aac7341.
- Oliver Huidobro, Martina, Jure Tica, Georg K. A. Wachter, and Mark Isalan (June 2022). "Synthetic spatial patterning in bacteria: advances based on novel diffusible signals". In: *Microbial Biotechnology* 15 (6), pp. 1685–1694. ISSN: 1751-7915. DOI: 10.1111/1751-7915.13979.
- Pai, Anand and Lingchong You (Jan. 2009). "Optimal tuning of bacterial sensing potential". In: *Molecular Systems Biology* 5 (1). ISSN: 1744-4292. DOI: 10.1038/msb.2009.43.
- Payne, Stephen, Bochong Li, Yangxiaolu Cao, David Schaeffer, Marc D. Ryser, and Lingchong You (2013). "Temporal control of self-organized pattern formation without morphogen gradients in bacteria". In: *Molecular Systems Biology* 9 (1), pp. 1–10. ISSN: 17444292. DOI: 10.1038/msb.2013.55.
- Perrimon, Norbert, Chrysoula Pitsouli, and Ben-Zion Shilo (2012). "Signaling mechanisms controlling cell fate and embryonic patterning". In: *Cold Spring Harbor perspectives in biology* 4.8, a005975.
- Pham, Viet-Thanh, Sundarapandian Vaidyanathan, and Tomasz Kapitaniak (2020). "Complexity, dynamics, control, and applications of nonlinear systems with multistability". In: *Complexity* 2020, pp. 1–7.
- Potvin-Trottier, Laurent, Nathan D. Lord, Glenn Vinnicombe, and Johan Paulsson (2016). "Synchronous long-term oscillations in a synthetic gene circuit". In: *Nature* 538 (7626), pp. 514–517. ISSN: 14764687. DOI: 10.1038/nature19841.
- Pušnik, Žiga, Miha Mraz, Nikolaj Zimic, and Miha Moškon (Sept. 2019). "Computational analysis of viable parameter regions in models of synthetic biological systems". In: *Journal of Biological Engineering* 13 (1). ISSN: 17541611. DOI: 10.1186/s13036-019-0205-0.
- Ramos, J.I (Dec. 1983). "A review of some numerical methods for reaction-diffusion equations". In: *Mathematics and Computers in Simulation* 25 (6), pp. 538–548. ISSN: 03784754. DOI: 10.1016/0378-4754(83)90127-1.
- Raspopovic, J., L. Marcon, L. Russo, and J. Sharpe (2014). "Digit patterning is controlled by a Bmp-Sox9-Wnt Turing network modulated by morphogen gradients". In: *Science* 345 (6196), pp. 566–570. ISSN: 10959203. DOI: 10.1126/science.1252960.
- Rauch, Erik M. and Mark M. Millonas (2004). "The role of trans-membrane signal transduction in turing-type cellular pattern formation". In: *Journal of Theoretical Biology*. ISSN: 00225193. DOI: 10.1016/j.jtbi.2003.09.018.

- Riglar, David T. et al. (2019). "Bacterial variability in the mammalian gut captured by a single-cell synthetic oscillator". In: *Nature Communications* 10 (1), pp. 1–12. ISSN: 20411723. DOI: 10.1038/s41467-019-12638-z.
- Rogers, Katherine W and Alexander F Schier (2011). "Morphogen gradients: from generation to interpretation". In: *Annual review of cell and developmental biology* 27, pp. 377–407.
- Saunders J, John W (1968). "Ectodermal-mesenchymal interactions in the origin of limb symmetry". In: *Epithelial-mesenchymal interactions*.
- Schaefer, A L, D L Val, B L Hanzelka, J E Cronan, and E P Greenberg (Sept. 1996). "Generation of cell-to-cell signals in quorum sensing: acyl homoserine lactone synthase activity of a purified *Vibrio fischeri* LuxI protein." In: *Proceedings of the National Academy of Sciences* 93 (18), pp. 9505–9509. ISSN: 0027-8424. DOI: 10.1073/pnas.93.18.9505.
- Schaerli, Yolanda, Andreea Munteanu, Magüi Gili, James Cotterell, James Sharpe, and Mark Isalan (Sept. 2014). "A unified design space of synthetic stripe-forming networks". In: *Nature Communications* 5. ISSN: 20411723. DOI: 10.1038/ncomms5905.
- Schier, Alexander F. (2009). *Nodal morphogens*. DOI: 10.1101/cshperspect.a003459.
- Schnakenberg, J. (1979). "Simple chemical reaction systems with limit cycle behaviour". In: *Journal of Theoretical Biology*. ISSN: 10958541. DOI: 10.1016/0022-5193(79)90042-0.
- Scholes, Natalie and Mark Isalan (Oct. 2017). *A three-step framework for programming pattern formation*. DOI: 10.1016/j.cbs.2017.04.008.
- Scholes, Natalie, David Schnoerr, Mark Isalan, and Michael Stumpf (Sept. 2019). "A Comprehensive Network Atlas Reveals That Turing Patterns Are Common but Not Robust". In: *Cell Systems* 9 (3), 243–257.e4. DOI: 10.1016/j.cels.2019.07.007.
- Sekine, Ryoji, Tatsuo Shibata, and Miki Ebisuya (Dec. 2018). "Synthetic mammalian pattern formation driven by differential diffusivity of Nodal and Lefty". In: *Nature Communications* 9 (1). ISSN: 20411723. DOI: 10.1038/s41467-018-07847-x.
- Semwogerere, Denis and Eric R Weeks (2005). "Confocal microscopy". In: *Encyclopedia of biomaterials and biomedical engineering* 23, pp. 1–10.
- Sheth, Rushikesh, Luciano Marcon, M. Félix Bastida, Marisa Junco, Laura Quintana, Randall Dahn, Marie Kmita, James Sharpe, and Maria A. Ros (2012). "Hox genes regulate digit patterning by controlling the wavelength of a turing-type mechanism". In: *Science*. ISSN: 10959203. DOI: 10.1126/science.1226804.
- Shiomi, Daisuke, Hideo Mori, and Hironori Niki (2009). "Genetic mechanism regulating bacterial cell shape and metabolism". In: *Communicative and integrative biology* 2.3, pp. 219–220.
- Shyer, Amy E, Tuomas Tallinen, Nandan L Nerurkar, Zhiyan Wei, Eun Seok Gil, David L Kaplan, Clifford J Tabin, and L Mahadevan (2013). "Villification, how the gut gets its villi". In: *Science* 342.6155, pp. 212–218.
- Smith, Gordon D (1985). *Numerical solution of partial differential equations: finite difference methods*. Oxford university press.

- Smith, Stephen and Neil Dalchau (2018). "Model reduction enables Turing instability analysis of large reaction–diffusion models". In: *Journal of the Royal Society Interface* 15 (140). ISSN: 17425662. DOI: 10.1098/rsif.2017.0805.
- Soboleski, Mark R, Jason Oaks, and William P Halford (2005). "Green fluorescent protein is a quantitative reporter of gene expression in individual eukaryotic cells". In: *The FASEB journal: official publication of the Federation of American Societies for Experimental Biology* 19.3, p. 440.
- Stevens, Martin, Innes C. Cuthill, Amy M.M. Windsor, and Hannah J. Walker (Oct. 2006). "Disruptive contrast in animal camouflage". In: *Proceedings of the Royal Society B: Biological Sciences* 273 (1600), pp. 2433–2438. ISSN: 14712970. DOI: 10.1098/rspb.2006.3614.
- Strauss, Soren, Janne Lempe, Przemyslaw Prusinkiewicz, Miltos Tsiantis, and Richard S. Smith (Jan. 2020). "Phyllotaxis: is the golden angle optimal for light capture?" In: *New Phytologist* 225 (1), pp. 499–510. ISSN: 14698137. DOI: 10.1111/nph.16040.
- Strikwerda, John C (2004). *Finite difference schemes and partial differential equations*. SIAM.
- Swindale, N. V. (1980). "A model for the formation of ocular dominance stripes". In: *Proceedings of the Royal Society of London - Biological Sciences*. ISSN: 09628452. DOI: 10.1098/rspb.1980.0051.
- Tan, Zhe, Shengfu Chen, Xinsheng Peng, Lin Zhang, and Congjie Gao (2018). "Polyamide membranes with nanoscale Turing structures for water purification". In: *Science*. ISSN: 10959203. DOI: 10.1126/science.aar6308.
- Thomas, Daniel and Jean Pierre Kernevez (1976). "Analysis and control of immobilized enzyme systems". In: *International Symposium on Analysis and Control of Immobilized Enzyme Systems (1975: Compiegne, France)*. North-Holland.
- Tica, J et al. (Jan. 2023a). "A three-node Turing gene circuit forms periodic spatial patterns in bacteria". In: *bioRxiv*, p. 2023.10.19.563112. DOI: 10.1101/2023.10.19.563112.
- Tica, Jure (2020). "Engineering Turing Patterns in *E. coli* with a bottom-up Synthetic Biology approach." In.
- Tica, Jure, Manman Chen, Shulei Luo, and Mark Isalan (2023b). "A novel approach to measuring diffusion of signalling molecules with the help of *E. coli* reporter strains".
- Trefethen, Lloyd Nicholas (1996). "Finite difference and spectral methods for ordinary and partial differential equations". In.
- Turing, A M (1952). "The Chemical Basis of Morphogenesis". In: *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 237 (641), pp. 37–72. ISSN: 0962-8436. DOI: 10.1098/rstb.1952.0012.
- Turányi, Tamás (1994). "Parameterization of reaction mechanisms using orthonormal polynomials". In: *Computers and Chemistry* 18 (1). ISSN: 00978485. DOI: 10.1016/0097-8485(94)80022-7.

- Vahey, Michael D and Daniel A Fletcher (Feb. 2014). "The biology of boundary conditions: cellular reconstitution in one, two, and three dimensions". In: *Current Opinion in Cell Biology* 26, pp. 60–68. ISSN: 09550674. DOI: 10.1016/j.ceb.2013.10.001.
- Volteras, Dimitris, Vahid Shahrezaei, and Philipp Thomas (2023). "Global transcription regulation revealed from dynamical correlations in time-resolved single-cell RNA-sequencing". In: *bioRxiv*, pp. 2023–10.
- Walker, Benjamin J., Adam K. Townsend, Alexander K. Chudasama, and Andrew L. Krause (Aug. 2023). "VisualPDE: rapid interactive simulations of partial differential equations". In.
- Wang, Baojun, Richard I. Kitney, Nicolas Joly, and Martin Buck (2011). "Engineering modular and orthogonal genetic logic gates for robust digital-like synthetic biology". In: *Nature Communications* 2 (1). ISSN: 20411723. DOI: 10.1038/ncomms1516.
- Wang, LianHui, LiXing Weng, YiHu Dong, and LianHui Zhang (Apr. 2004). "Specificity and Enzyme Kinetics of the Quorum-quenching NAcyl Homoserine Lactone Lactonase *AHLLactonase*". In: *Journal of Biological Chemistry* 279 (14), pp. 13645–13651. ISSN: 00219258. DOI: 10.1074/jbc.M311194200.
- Wang, Sheng, Jordi Garcia-Ojalvo, and Michael B. Elowitz (Dec. 2022). "Periodic spatial patterning with a single morphogen". In: *Cell Systems* 13 (12), 1033–1047.e7. ISSN: 2405-4712. DOI: 10.1016/J.CELLS.2022.11.001.
- Ward, William W, Hugh J Prentice, Amy F Roth, Chris W Cody, and Sue C Reeves (1982). "Spectral perturbations of the Aequorea green-fluorescent protein". In: *Photochemistry and Photobiology* 35.6, pp. 803–808.
- Weiss, James N. (1997). "The Hill equation revisited: uses and misuses". In: *The FASEB Journal* 11 (11), pp. 835–841. ISSN: 0892-6638. DOI: 10.1096/fasebj.11.11.9285481.
- Wimpenny, Julian (1979). "The growth and form of bacterial colonies". In: *Microbiology* 114.2, pp. 483–486.
- Wolpert, L (1969). *Positional Information and the Spatial Pattern of Cellular Differentiation*.
- Woolley, Thomas E. (Sept. 2022). "Boundary Conditions Cause Different Generic Bifurcation Structures in Turing Systems". In: *Bulletin of Mathematical Biology* 84 (9), p. 101. ISSN: 0092-8240. DOI: 10.1007/s11538-022-01055-x.
- Woolley, Thomas E., Andrew L. Krause, and Eamonn A. Gaffney (May 2021). "Be-spoke Turing Systems". In: *Bulletin of Mathematical Biology* 83 (5). ISSN: 15229602. DOI: 10.1007/s11538-021-00870-y.
- Yeoh, Jing Wui, Kai Boon Ivan Ng, Ai Ying Teh, Jing Yun Zhang, Wai Kit David Chee, and Chueh Loo Poh (July 2019). "An Automated Biomodel Selection System (BMSS) for Gene Circuit Designs". In: *ACS Synthetic Biology* 8 (7), pp. 1484–1497. ISSN: 21615063. DOI: 10.1021/acssynbio.8b00523.

Zheng, M. Mocarlo, Bin Shao, and Qi Ouyang (Nov. 2016). "Identifying network topologies that can generate turing pattern". In: *Journal of Theoretical Biology* 408, pp. 88–96. ISSN: 10958541. DOI: 10.1016/j.jtbi.2016.08.005.

APPENDIX A

Appendix: Model parameters for Chapter 2

Table A.1: **Variant 1**

Parameter	Distribution	Value
V_m	Loguniform	(10,1000)
K_m	Loguniform	(0.1,250)
μ_m	Loguniform	(0.001,50)
D_B	Loguniform	(0.001, 10)
D_A	Fixed	1
b	Fixed	0.01
n	Fixed	2

Table A.2: Variant 1 or 2

Parameter	Distribution	Value
V_m	Loguniform	(10,10000)
K_m	Loguniform	(0.1,100)
μ_m	Loguniform	(1,100)
D_B	Fixed	(0.01 or 10)
D_A	Fixed	(10 or 0.01)
b	Loguniform	(10,10000)
n	Loguniform	(2,4)

APPENDIX B

Appendix: Model parameters for Chapter 4

Simulation ID	Analytical solution	Growth regime	D_r^*
Fig. 4.9.1	No steady state	T50	9.09
Fig. 4.9.2	Turing I	T25	0.01
Fig. 4.9.3	Turing I Unstable Stable	T25	0.02
Fig. 4.9.4	Turing I	T50	0.02
Fig. 4.9.5	Turing I hopf	T25	0.32
Fig. 4.9.6	Turing I	T100 Open boundary	0.15
Fig. 4.9.7	Turning I Hopf unstable	T25	0.02
Fig. 4.9.8	Turing I	T100	0.15
Fig. 4.8.1	Turing I Hopf	T100	0.10
Fig. 4.8.2	Turing I Hopf	T100	0.32
Fig. 4.8.3	Turing I Hopf	T100	0.03
Fig. 4.8.4	Turing I	T100	0.03
Fig. 4.8.5	Turing I	T100	0.01
Fig. 4.8.6	Turing I	T100	0.03
Fig. 4.8.7	Hopf	T50	0.16
Fig. 4.8.8	Hopf	T50	0.58
Fig. 4.8.9	Hopf	T50	3.69
Fig. 4.8.10	No steady state	T50	9.09
Fig. 4.8.11	No steady state	T50	64.34
Fig. 4.8.12	Stable Unstable Stable	T25	1.25
Fig. 4.8.13	Stable Turing Unstable	T50	0.01
Fig. 4.8.14	Stable	T25	25.66
Fig. 4.8.15	Stable	T25	97.13
Fig. 4.10.1	Turing I	T50	0.15
Fig. 4.10.2	Turing I Hopf	T50	0.10
Fig. 4.11.4	Turing I	T100	0.15
Fig. 4.11.5	Stable Peak near zero	T100	0.15
Fig. 4.15.B	Turing I	T100 Open boundary	0.15
Fig. 4.14	Turing I	T110 Different growth rates	0.15

Table B.1: **First table of model params.** The analytical solution is derived using linear stability analysis and describes the properties of the steady-states of the system, including Turing I, Hopf, Simple stable, Simple unstable and no steady-states. The growth regimes that determine the PDE and cellular automata parameters are further specified in Table C.1. The diffusion rate is also included.

Simulation ID	Dr*	Va*	Vb*	Vc*	Vd*	Ve*	Vf*	Kub*	Kvd*	Kda*	Kce*	Kfe*	Keb*	Kee*	μ_{ASV}	μ_{LVA}	nfe
Fig. 4.9.1	9.09	74.05	169.38	669.89	41.12	39.58	76.52	15.15	120.03	1.70	100	10.73	5.64	0.01	3.37	1	5
Fig. 4.9.2	0.01	150.29	54.23	28.08	266.88	45.87	10.43	29.36	42.53	2.85	100.00	1.19	17.84	0.01	3.86	1	5
Fig. 4.9.3	0.02	795.88	29.76	14.26	21.69	19.04	48.66	150.98	6.42	1.46	100.00	15.15	11.34	0.01	4.46	1	5
Fig. 4.9.4	0.02	45.20	603.17	234.59	158.98	70.72	13.73	13.93	239.29	1.97	100.00	2.35	60.56	0.01	3.99	1	5
Fig. 4.9.5	0.32	48.36	786.32	31.16	264.07	30.32	164.65	14.38	251.50	7.89	100	28.85	28.75	0.01	3.83	1	5
Fig. 4.9.6	0.15	21.67	321.54	16.88	5.29	8.60	3.57	43.93	26.33	0.73	2.34	1.41	1.53	0.001	3.57	1	8
Fig. 4.9.7	0.02	113.58	725.58	322.38	240.91	21.66	13.29	80.20	10.45	24.96	100.00	7.56	4.46	0.01	3.53	1	5
Fig. 4.9.8	0.15	21.67	321.54	16.88	5.29	8.60	3.57	43.93	26.33	0.73	2.34	1.41	1.53	0.001	3.57	1	8
Fig. 4.8.1	0.10	495.68	872.38	697.87	372.08	109.70	14.56	53.14	219.76	4.99	100.00	2.75	77.39	0.01	3.70	1	5
Fig. 4.8.2	0.32	48.36	786.32	31.16	264.07	30.32	164.65	14.38	251.50	7.89	100.00	28.85	28.75	0.01	3.83	1	5
Fig. 4.8.3	0.03	301.23	83.90	491.96	868.92	131.53	63.43	133.37	68.45	9.85	100.00	3.09	73.83	0.01	3.70	1	5
Fig. 4.8.4	0.03	193.50	22.54	626.85	106.80	402.97	39.13	13.91	4.82	6.61	100.00	23.70	336.49	0.01	3.18	1	5
Fig. 4.8.5	0.01	19.53	12.34	123.40	51.73	40.95	91.98	7.07	2.98	6.18	100.00	50.56	32.69	0.01	4.56	1	5
Fig. 4.8.6	0.03	816.37	77.89	58.00	63.95	59.97	12.78	110.34	49.06	3.22	100.00	3.38	50.31	0.01	3.49	1	5
Fig. 4.8.7	0.16	11.99	82.80	517.54	118.80	12.02	500.10	4.90	16.60	22.21	100.00	372.67	2.19	0.01	3.59	1	5
Fig. 4.8.8	0.58	311.01	452.00	609.80	140.64	511.01	264.43	267.90	23.40	17.44	100.00	80.26	10.13	0.01	3.86	1	5
Fig. 4.8.9	3.69	138.98	718.76	731.67	574.33	893.73	274.09	45.83	231.33	3.65	100.00	127.68	5.88	0.01	3.82	1	5
Fig. 4.8.10	9.09	74.05	169.38	669.89	41.12	39.58	76.52	15.15	120.03	1.70	100.00	10.73	5.64	0.01	3.37	1	5
Fig. 4.8.11	64.34	229.84	199.15	822.04	58.37	137.50	18.27	9.68	70.64	2.17	100.00	8.96	23.36	0.01	3.26	1	5
Fig. 4.8.12	1.25	11.76	134.85	286.24	168.35	306.73	398.68	3.15	15.85	160.35	100.00	84.35	1.32	0.01	3.13	1	5
Fig. 4.8.13	0.01	165.48	100.21	52.73	377.51	192.76	603.27	70.15	12.06	127.57	100	131.70	1.66	0.01	4.45	1	5
Fig. 4.8.14	25.66	825.22	30.51	130.99	30.67	166.25	630.74	238.35	4.73	13.41	100.00	1.85	147.94	0.01	3.16	1	5
Fig. 4.8.15	97.13	259.08	179.44	127.45	674.32	19.68	14.12	76.32	3.39	392.99	100.00	1.28	14.09	0.01	2.81	1	5
Fig. 4.10.1	0.15	21.74	321.61	16.95	5.27	8.54	3.57	0.15	26.39	16.95	5.27	1.42	1.52	0.001	3.60	1	8
Fig. 4.10.2	0.10	102.69	431.77	25.23	7.44	4.90	3.61	0.10	29.99	25.23	7.44	1.58	0.79	0.001	3.69	1	8
Fig. 4.11.4	0.15	21.67	321.54	16.88	5.29	8.60	3.57	43.93	26.33	0.73	2.34	1.41	1.53	0.001	3.57	1	8
Fig. 4.11.5	0.15	21.94	327.41	16.71	5.30	8.55	3.58	44.06	26.51	0.72	2.31	1.41	1.53	0.001	3.62	1	8
Fig. 4.15.B	0.15	21.67	321.54	16.88	5.29	8.60	3.57	43.93	26.33	0.73	2.34	1.41	1.53	0.001	3.57	1	8
Fig. 4.14	0.15	21.67	321.54	16.88	5.29	8.60	3.57	43.93	26.33	0.73	2.34	1.41	1.53	0.001	3.57	1	8

Table B.2: **Second table of model params.** The kinetic parameters for the dimensionless model are listed here. μ_{ASV} corresponds to μ_A while μ_{LVA} corresponds to μ_B , μ_C , μ_D , μ_E , μ_F . Cooperativity rates are not shown in this table but fixed to $n_{ub} = 1$, $n_{ee} = 4$, $n_{eb} = 4$, $n_{vd} = 2$, $n_{da} = 2$ and $n_{ce} = 3$.

APPENDIX C

Appendix: Growth regimes in cellular automata model

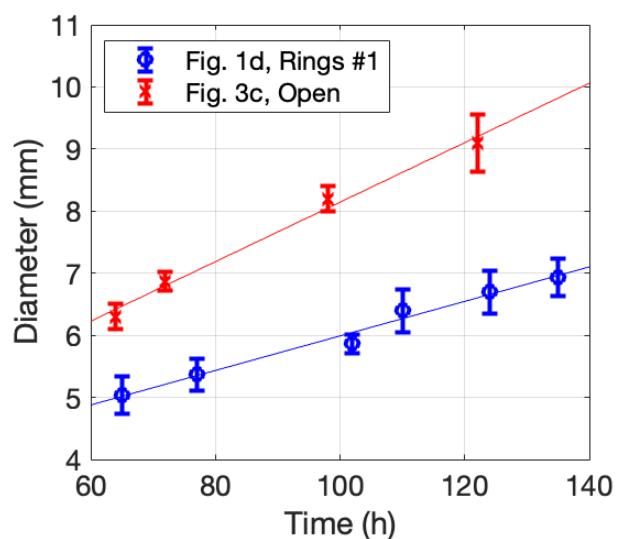


Figure C.1: **Growth rate of two experimental colonies.** Figure produced by Dr. Jure Tica. Image processing software was used to measure the size of the colony diameter across two different axes, this variability is captured in the error bars (\pm SEM, $n = 2$). Two colonies that grow equally in all directions (isotropic growth) were used as examples.

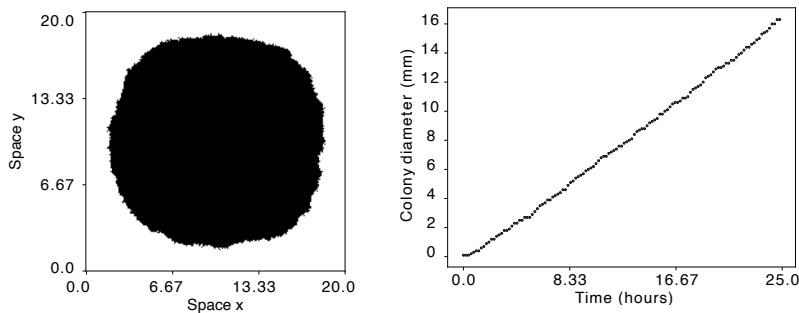
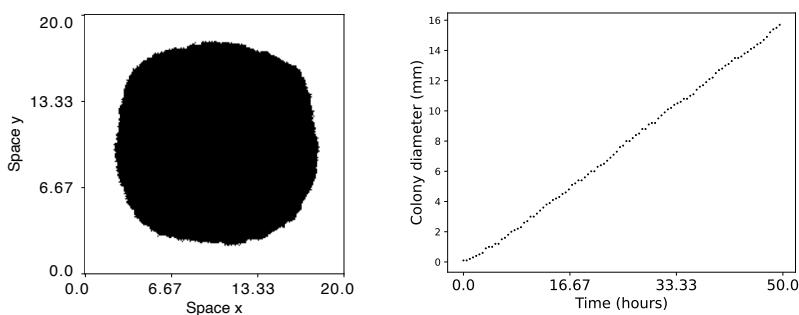
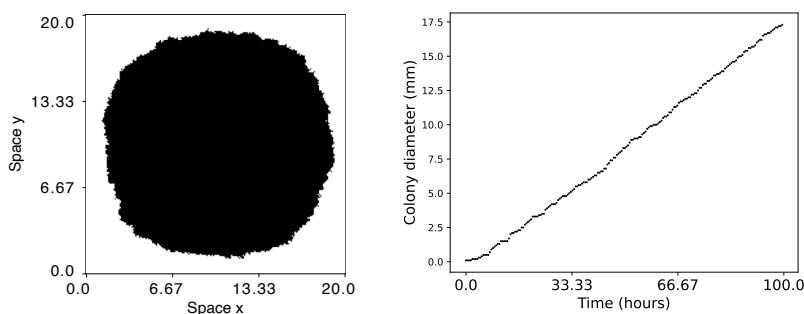
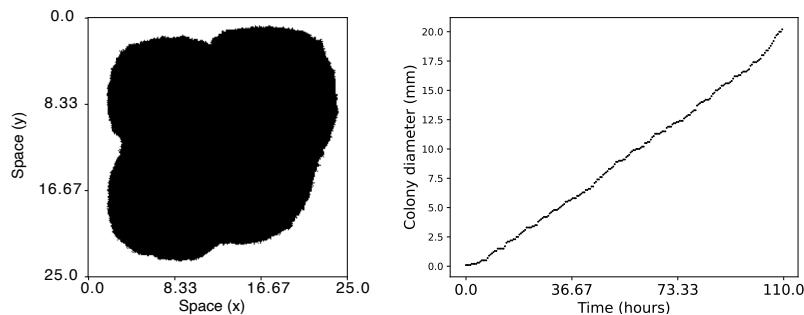
Growth regime T25**Growth regime T50****Growth regime T100****Growth regime T110 - Different growth rates**

Figure C.2: Bacterial colony and growth curves from cellular automata simulations. Using different growth regimes (left) final snapshot of bacterial colony and (right) growth curve of bacterial colony. Dimensionless time and space is shown here.

Growth regime	L	T	J	N	dx	dt	Boundary	Random seed	m	p_d
T10	20	10	200	500	0.1	0.02		1 1	0.2	0.7
T25	20	25	200	1250	0.1	0.02		1 1	0.2	0.7
T50	20	50	200	2500	0.1	0.02		1 1	0.5	1
T50 Open boundary	20	50	200	2500	0.1	0.02		2 1	0.5	1
T100	20	100	200	5000	0.1	0.02		1 1	0.5	0.38
T110 Different growth rates	25	110	250	5500	0.1	0.02		1 1	0.5 0.38 and 0.91	

Table C.1: Cellular automata and PDE solver parameters for different growth regimes.

APPENDIX D

Appendix: Experimental conditions

Figure	Inducers	Timepoint (hours)	Colony Diameter (mm)
Fig. 4.9, Red Edge	10 µM ATC	38	3.5
Fig. 4.9, Black hole	10 µM ATC	26	3.3
Fig. 4.9, Wedges*	15 µM ATC	100	10.6
Fig. 4.9, Spots*	30 µM ATC	100	6.4
Fig. 4.9, Bullseye #1	10 µM ATC	50	4.6
Fig. 4.9, Bullseye #2	10 µM ATC	120	8.6
Fig. 4.9, Rings #1	20 µM ATC	124	6.5
Fig. 4.9, Rings #2 Fig. 3e	10 µM ATC	120	4.0

Table D.1: Concentrations of inducers and colony sizes for the patterning conditions shown in Fig. 4.9. These images were collected by Jure Tica, Georg Wachter, Tong Zhu and me. Spots and wedges are shown with an asterisk (*) because they were obtained with widefield microscopy rather than confocal microscopy, in 4xYT media rather than 2xYT. Agar is 1.4 % (w/V) in all conditions.