2019

Mathematical modeling for pattern design in networks of mammalian cells

BOSTON UNIVERSITY

GRADUATE SCHOOL OF ARTS AND SCIENCES

AND

COLLEGE OF ENGINEERING

Dissertation

MATHEMATICAL MODELING FOR PATTERN DESIGN IN NETWORKS OF MAMMALIAN CELLS

by

DEMARCUS I. BRIERS

B.S., Prairie View A&M University, 2014

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2019

Approved by

First Reader

Calin Belta, Ph.D.
Tegan Family Distinguished Faculty Fellow
Professor of Mechanical Engineering
Professor of Systems Engineering
Professor of Electrical and Computer Engineering

Second Reader

Ahmad S. Khalil, Ph.D.
Assistant Professor of Biomedical Engineering
Associate Director of the Biological Design Center

DEDICATION

This dissertation is dedicated to free thinkers of the past, present, and future who dare to change the world.

ACKNOWLEDGMENTS

I would like to thanks all of my family, friends, and colleagues who have inspired me during my PhD. Specifically. I would like to thank my family Deborah Briers, Jasmine Briers, James M Briers, James N. Briers, Sheryl Rosaya, Geraldine Rosaya, Frances Briers, James C. Briers, and Greg Rosaya for supporting me with memoirs and foods from home. I would also like to thank my friends Karlton Tatum, Leila Pirhaji, David Liu, Olexander Semenov, Rachael Ivison, Brandon Hynson, and Rudy. I would like to thank my colleagues Iman Haghighi, Ashley Libby, Noushin Mehdipour, Evan Appleton, David Joy, Tristian Daifuku, and Debra Murray.

M'ole mesi mo famiye, zami-té, e collèg-té ki ede-yé e gidé-yé mo dan mo PhD. M'ole mesi mo famiye Deborah Briers, Jasmine Briers, James M Briers, James N. Briers, Sheryl Rosaya, Geraldine Rosaya, Frances Briers, James C. Briers, and Greg Rosaya. M'ole mesi mo zami-té Karlton Tatum, Leila Pirhaji, David Liu, Olexander Semenov, Rachael Ivison, Brandon Hynson, and Rudy. M'ole mesi mo collèg-té Iman Haghighi, Ashley Libby, Noushin Mehdipour, Evan Appleton, David Joy, Tristian Daifuku, e Debra Murray.

MATHEMATICAL MODELING FOR PATTERN DESIGN IN NETWORKS OF MAMMALIAN CELLS

DEMARCUS I. BRIERS

Boston University, Graduate School of Arts and Sciences and College of Engineering, 2019

Major Professor: Calin Belta, PhD

ABSTRACT

During the early stages of embryonic development, mammalian cells communicate, undergo morphological changes, and self-assemble into highly organized tissues, and eventually organ systems. Recently, there have been several efforts to engineer the multicellular patterning in mammalian cells to better understand early development and create organoid systems to better understand human disease and drug interactions. However, existing approaches to engineer large scale multicellular patterning in mammalian cells are limited to reproducing well known behaviors or trail-and-error based design. In this thesis, I developed mathematical models to predictively design and quantitatively validate de novo multicellular patterning in mammalian cells. First, I have developed a computational to automate self-organized multicellular organization in human pluripotent stem cells quantitatively matches the in vitro velocity distribution, temporal dynamics of CRISPR induced perturbations to protein expression, and the resulting changes in spatial organization in human pluripotent stem cell colonies. I have also developed a mathematical model to predict the programmable self-assembly from a single cell into 3D shapes. Overall, this work offers insights into how mathematical

modeling can be integrated with pattern recognition and optimization algorithms to efficiently discover and direct self-organized multicellular patterning in cell aggregates and tissues.

CONTENTS

Dedication **iv**

Acknowledgements **v**

Abstract **vi**

List of Tables **xii**

List of Figures **xiii**

List of Symbols and Abbreviations **xv**

1 Background and Motivation **1**

1.1 Multicellular Networks of Mammalian Cells 1

1.2 Agent-Based Models for Multicellular Pattern Design 4

1.3 Multicellular Pattern Recognition 7

1.4 Formal Methods for Biological Pattern Optimization 8

1.5 Overview of Chapters 9

2 Pattern Design Automation in Human Pluripotent Stem Cells **11**

2.1 Pattern Synthesis: in silico prediction and automated discovery of spatial behaviors 13

2.2 Computational Model of Human PSC Self-Organization 16

2.2.1 Formulating Parameters for Design Automation 23

2.3 Quantitative Pattern Classification 24

2.4 Automated Discovery of Pattern Producing Conditions 25

2.5 In Silico Model Accurately Predicts In Vitro Experimental Validation 29
2.6 Colony Organization Impacts Patterns of iPSC Differentiation 31
2.7 Discussion 33
2.8 Materials and Methods 35
2.9 Agent-Based Model of Cell Sorting 35
2.9.1 Cellular Potts Model Environment 36
2.9.2 Cellular Potts Model Dynamics 37
2.9.3 Physical Units and Other Cellular Phenomena 41
2.10 Model Fitting to Empirical data 41
2.10.1 Morphology Characterization 41
2.10.2 Velocity Characterization (Space vs. Time) 42
2.10.3 Temporal Knockdown Characterization (Protein Expression vs Time) 43
2.10.4 Spatial Pattern Characterization (Protein Expression vs Space) 45
2.11 *In Vitro* Materials and Methods 46
2.11.1 Human iPSC culture: 46
2.11.2 Mixed Colony Generation: 47
2.11.3 Immunofluorescence Staining and Imaging: 47
2.11.4 Protein Quantification: 48
2.11.5 mRNA quantification: 48
2.11.6 Time Lapse Imaging: 49
2.11.7 BMP4 differentiations 50
2.12 Comparison of *in vitro* and *in silico* Spatial Patterns 50
2.12.1 Software 51
2.12.2 Supplementary Figures 52

3 Pattern Synthesis in 3D Stem Cell Aggregates 61

3.1 Introduction 61

3.2 Stochastic Agent-Based Model 63

3.2.1 Description of Stem Cell Agents 64

3.2.2 Structural Constraints in 3D 67

3.2.3 Stochastic Rules for State Changes 68

3.2.4 Visualization 70

3.3 Optimizing 3D Spatio-temporal Patterns 71

3.3.1 Quad-tree Representation of Spatial Behaviors 72

3.3.2 Formal Specification of Global Patterns 74

3.3.3 Quantification 76

3.3.4 Parameter Synthesis 78

3.4 Future Work 79

4 Agent-based Model for Self-Assembled 3D Multicellular Shapes 81

4.1 Introduction 81

4.2 Computational Model of Autonomous 3D Self-Assembly 81

4.2.1 Cell Environment 82

4.2.2 Cell Logic 82

4.2.3 The Developmental Tree 83

4.3 Case Studies: Programmatic Self-Assembly 86

4.4 Conclusions and Future Work 86

Conclusion and Directions 90

A Apendix of Computational Methods 92

A.1 Pattern Optimization 92

A.1.1 Quad-Tree Representation of an Image 92

A.1.2 Tree Spatial Superposition Logic 94

A.1.3 Quantitative Robustness 97

A.1.4 Particle Swarm Optimization 98

Bibliography **102**

Curriculum Vitae **111**

LIST OF TABLES

2.1 List of design parameters that map to experimental perturbations. . 23

3.1 Number of samples generated to learn and test TSSL classifiers . . . 75

LIST OF FIGURES

1.1 Multicelllular Networks vs Biological Networks 2

1.2 Actuators for Controlled Cell Patterning 6

2.1 Pattern Design Automation 15

2.2 Characterizing Single and Collective Cell Interactions 22

2.3 Quantitative Pattern Classification with TSSL 25

2.4 Computational synthesis of de novo spatial patterns and in vitro validation 27

2.5 Differentiation of Patterned colonies 33

2.6 Single Cell Morphology Measurements 52

2.7 CRISPRi CDH1 knockdown immunofluorescence 53

2.8 CRISPRi cell line mRNA knockdowns 54

2.9 Quadtree of a Matrix 54

2.10 Quadtree of a Checkerboard 55

2.11 Time Series Bullseye and Multi-island Simulation 55

2.12 Temoral Evolution of TSSL Scores 56

2.13 In silico vs. in vitro spatial metrics 57

2.14 Karyotype of CRISPRi Cell Line 58

2.15 Image Segmentation Pipelines 59

2.16 Modeling Fitting Parameters for stem cell multicellular patterning . 60

3.1 Transient Multicellular Differentiation Patterns 65

3.2 Sample Simulation for a parameter set 71

3.3 Quad-tree representation of a matrix 72

3.4 Sample images of simulations derived from the optimized parameter values 79

4.1 Actuators for 3D Self-Assembly 84

4.2 Assembly of two independent tetrahedrons 87

4.3 In Silico Verification 88

LIST OF ABBREVIATIONS

CPM .	Cellular Potts Model
iPSC .	induced Pluripotent Stem Cell
MN .	Multicellular Network
SpaTeL	Spatial Temporal Logic
TSSL .	Tree Spatial Superposition Logic

CHAPTER 1
Background and Motivation

1.1 MULTICELLULAR NETWORKS OF MAMMALIAN CELLS

The adult human body is a complex multicellular network (MN) containing approximately 37.2 trillion cells (excluding bacteria) (Bianconi et al., 2013). Similar to the infrastructure networks of a city in the 21st century (social, traffic, telecommunications, power, etc.), mammalian cells form complex communication networks that allow cells to sense and respond to their environment. Each cell is a node in the network, and diverse cell-cell communication mechanisms are the links that connect (Fig 1A) cells to their environment. MNs are not to be confused with the phrase "biological network" where biomolecules (such as genes, proteins, and metabolites) and their molecular interactions are represented as a network (Fig 1C). The key differentiator of multicellular networks is that whole cells are nodes in the network rather than biomolecules.

Engineering MNs to form pre-defined cell patterning and cell structures provides a powerful tool to better understand and regenerate organ-like tissues, commonly called organoids, and create more accurate experimental models of human disease and the response of human tissues to drug therapies. The development of more robust organoids will facilitate research studies that have previously been hindered by the technical and ethical difficulties of conducting research studies in living people. While several labs have developed high functioning organoids derived from pluripotent and adult stem cells (Huch & Koo, 2015), many organoids have low rates of successful assembly and do not fully replicate their *in vitro* counterparts. This is partially due to a lack of control of the spatiotemporal organiza-

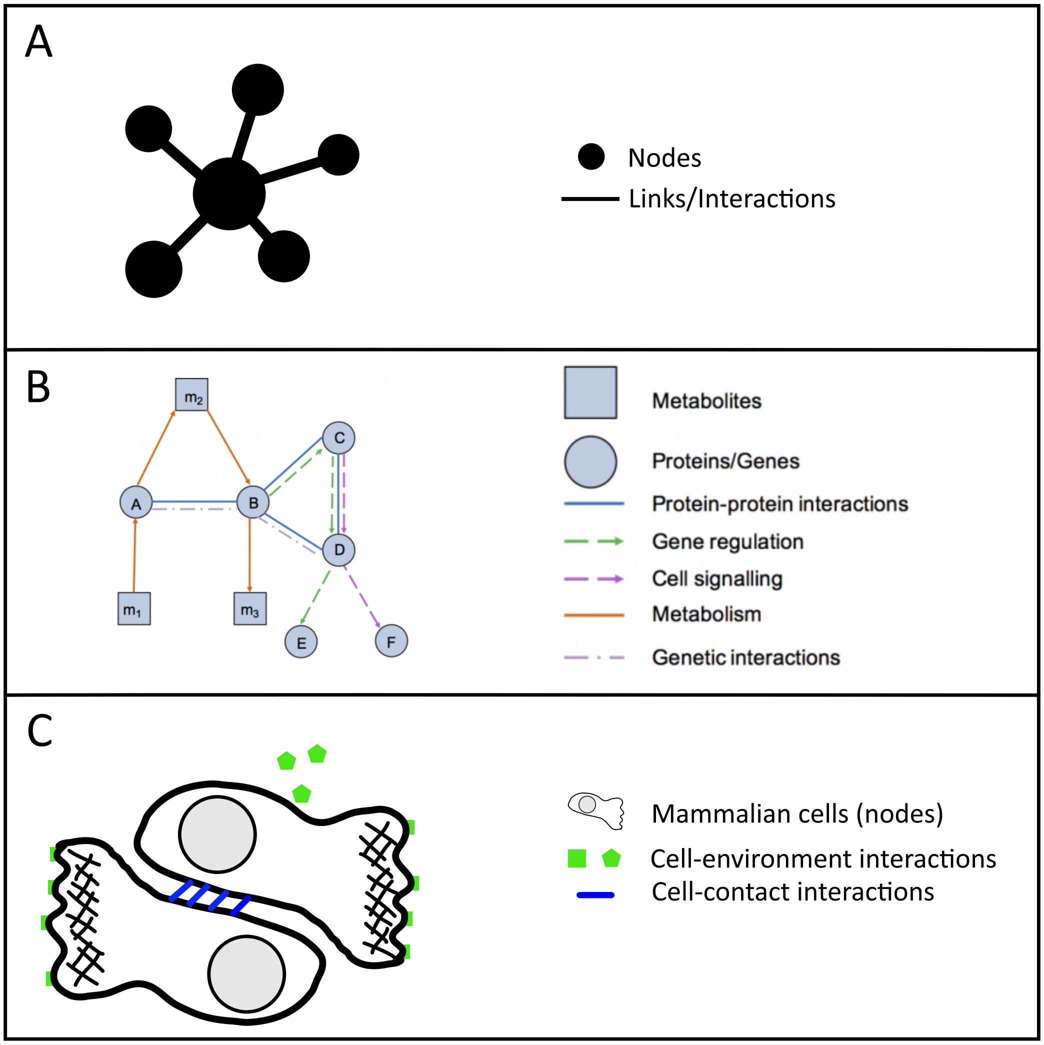


Figure 1.1: Multicellular Networks vs Biological Networks.
A) A Generic Network containing nodes and links. B) A multicellular network replaces nodes with cells and links with cell-contact and cell-environment interactions. C) A biological network where biomolecules replace nodes and physical interactions=represent links. Network in section A by Three Six Five from the Noun Project. Figure adapted from Davies (2008) and www.ebi.ac.uk course on Network analysis of protein data

tion while producing organoids and living structures.

Current methods to design multicellular structures have produced multi-cellular patterning such as concentric rings (Toda et al., 2018) in fibroblast cells using synthetic engineering, concentric rings of stem cells that differentiated into the germ layers observed during early embryonic development using microcontact patterning (Warmflash et al., 2014), or 3D tubes that allow fluid flow using 3D bioprinting (Homan et al., 2016). These approaches have introduced novel tools that will be used in future tissue engineering endeavors. However, these approaches using synthetic engineering rely on trail-and-error to design patterning. These unguided approaches will be difficult to produce precise patterns as the variety and timing of cell actuations are combined to form more complex patterning. 3D bioprinting and microcontact scaffolding suffer from a complimentary challenge. These scaffold-based approach are guided, but produce static patterns. Many developmental organization events proceed in temporal sequences (eg germ layer differentiation) that are difficult to engineer with static, scaffold-based approaches.

In this thesis, I collaboratively developed mathematical models to both predict and direct self-organized multi-cellular patterns and self-assembly of 3D shapes using mammalian cells. Designing complex spatial behaviors in mammalian systems requires the ability to intelligently direct single-cell and population-scale behaviors which is a common theme in developmental biology and tissue engineering. Traditionally, designing multicellular behaviors has been a manual task that required human expertise which also introduces human bias. In the following sections of this chapter, I briefly review how computational modeling, pattern recognition, and mathematical optimization can automate the design of multicellular systems.

1.2 AGENT-BASED MODELS FOR MULTICELLULAR PATTERN DESIGN

Agent-based models (ABM) are a popular collection of computational modeling frameworks that can describe multicellular pattern formation from the perspective of individual cells and their interactions. This is in contrast to continuum models that represent large scale tissue dynamics and omit the behavior of individual cells. The factors that are typically relevant when modeling multicellular and tissue organization include: 1) representing cell behaviors at difference spatial scales (eg from single-cell protein expression to collective cell migration), 2) representing cell behaviors at different time scales, 3) representing cell mechanics, 4) representing biochemical/electrical signaling, and 5) being computationally efficient.

While no single agent-based framework accurately recapitulates all aspects driving patterning in mammalian cells, it has been proposed that a relatively small set of morphological actuators can be combined in different permutations to generate a wide range of synthetically engineered structures (Davies, 2008; Cachat et al., 2014). These morphological actuations are (Fig 1.2): cell proliferation, apoptosis, cell fusion, cell adhesion, membrane stiffness, cell migration (eg collective cell migration, chemotaxis, haptotaxis), cell sorting (Cachat et al., 2014), and folding/contraction (Ricotti et al., 2017). In addition to these morphological actuations, contact-mediated signaling (Toda et al., 2018), biochemical signaling (White et al., 2013, 2015), and bioelectric signaling (McNamara et al., 2018; Molitoris et al., 2016) play an important role in tissue organization and communication. Given these diverse cell actuators and communication mechanisms, spatial organization can be controlled in a similar manner to simple electronic circuits using endogenous (Libby et al., 2018a) and synthetic control (Toda et al., 2018) of these actuators and communication channels.

Agent-based models are well suited for predictively modeling multi-cellular patterning since many population-scale behaviors, such as cell sorting, *emerge* from changes in single-cell behaviors; in the case of cell sorting this can be differences in cell-cell adhesion (Cachat et al., 2014; Libby et al., 2018a). These morphological operators can be combined with complex cell logic by introducing gene knockdowns, gene activations, or synthetic gene circuits so that cells can self-assemble into multicellular structures and tissue-like mimics.

Agent-based models can have very different mathematical formalisms and thus must be evaluated carefully to accurately represent the relevant cell actuations driving the multicellular patterning being investigated. When modeling the role of tissue mechanics (eg cell-cell adhesion, cell migration, cell shape, haptotaxis), the most common ABMs frameworks are Cellular Automaton (Vaca-González et al., 2017), Cellular Potts framework (Voss-Bohme, 2012), Vertex models, Cell Center-based modelsi (this includes particle models and rigid body models), and hybrid models that combine aspects of multiple ABM frameworks (Taylor et al., 2017). Van Liedekerke et al. (2015) gives a comprehensive review of the strengths and weaknesses of each agent-based modeling framework.

In combination with agent-based models, differential calculus is a common approach to model dynamics within cells, using Ordinary Differential Equations(ODEs), and model the dynamics of communication between cells using Partial Differential Equations (PDEs). The Reaction-Diffusion framework (Kondo & Miura, 2010) is a popular approaches that unifies ODE and PDE frameworks to model single-cell gene/protein dynamics (or reaction) with short and long range cell communication (via diffusion). By combining differential calculus with agent-based models,

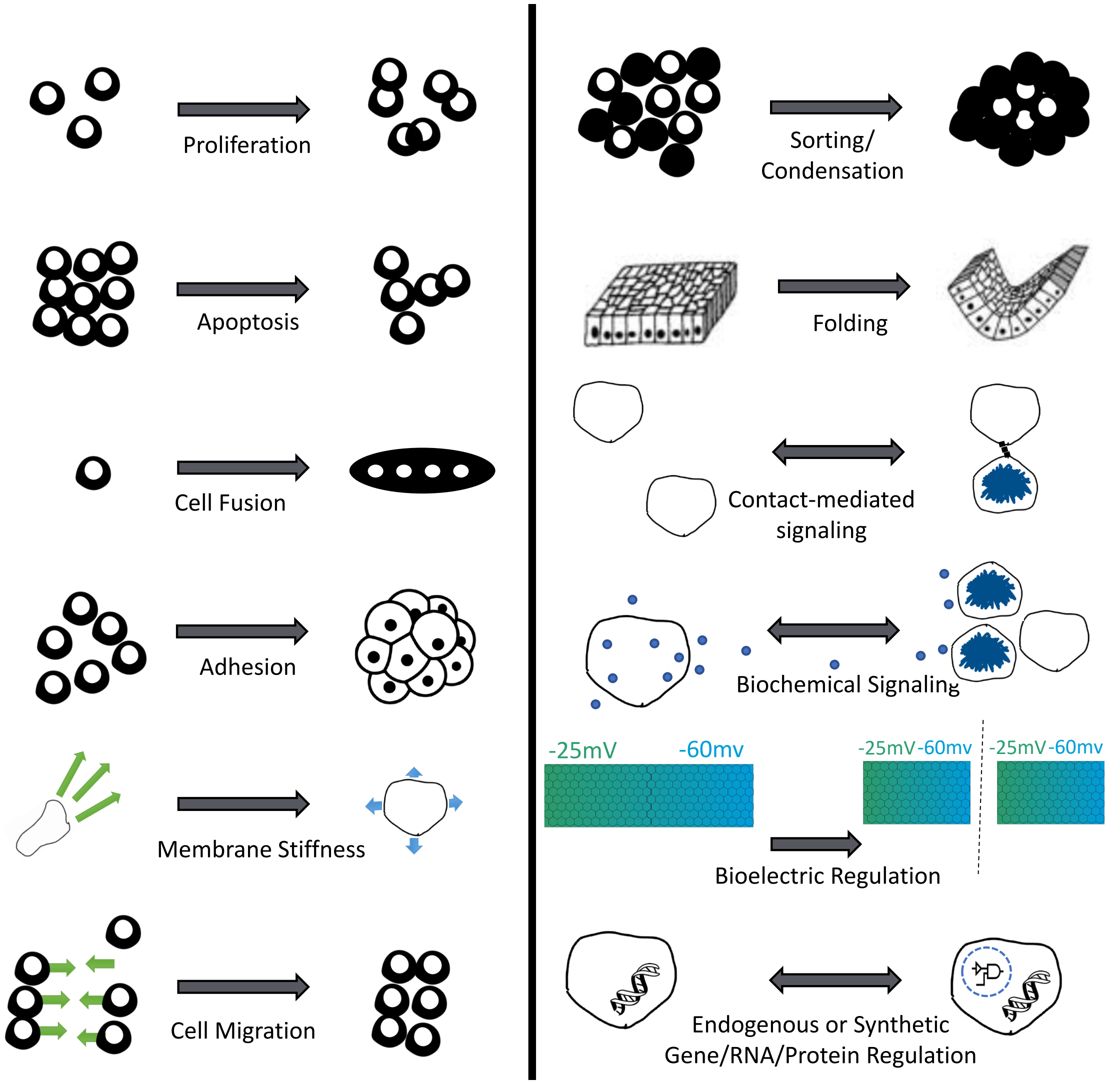


Figure 1.2: Actuators for Controlled Cell Patterning. Figure adapted from Davies (2008). It has been proposed that a relatively small set of morphological actuators can be combined in different permutations to generate a wide range of synthetically engineered structures.

researchers are able to explain patterning at multiple temporal and spatial scales that can emerge from mechanical, biochemical, or bioelectric phenomena. In the next section I describe computational approaches to identify and quantify images of multicellular patterning simulations and experiments.

1.3 MULTICELLULAR PATTERN RECOGNITION

As computational modeling plays a more direct role in guiding *in vitro* multicellular patterning, the amount of data, typically images, becomes a bottleneck in gaining insights. Several types of pattern recognition metrics and algorithms exist to differentiate 2D and 3D patterns and object. To design multicellular patterning, pattern recognitions metrics rapidly quantify if simulated or experimental images/video matches a previously observed or desirable behavior.

With respect to recognizing patterns in multicellular systems there are three common approaches: statistical image descriptors, unsupervised machine learning, and supervised machine learning machine learning. In Computer Vision, image descriptors can describe elementary shapes, colors, textures (Nanni et al., 2010), or motion in images or videos. Image descriptors typically produce a histogram or distribution of values that can be considered a 'fingerprint' of the pattern. Given this fingerprint several distance measures such as the Euclidean distance, Earth Movers Distance, Kullback Leibler Divergence, Jensen Shannon Divergence, or other distance measures can be used to measure the distance between an observed pattern and the desired pattern.

Machine learning algorithms allows computers to learn from data without explicitly being given the rules. Unsupervised machine learning approaches are utilized when there are no labels describing how images should be grouped or classified. Unsupervised learning can be used to discover groups of similar images (clustering), project the image from a high to low dimensional space (dimensionality reduction) for visualization or simpler interpretation. Unsupervised learning can also be used to estimate the distribution of pixel values in an image (kernel density estimation). Supervised Machine Learning algorithms are utilized when label are assigned to input data describing how samples should be grouped or labeled (Bishop, 2006). For example, labels can indicate if a image contains healthy or cancerous cells. While Support Vector Machine(SVM) based classifiers have been widely use to predict the phenotype of biological images (Bauer et al., 2011; Campanini et al., 2004; Maglogiannis & Zafiropoulos, 2004), deep learning approaches are becoming increasing common for predicting phenotypes and recognizing patterns in biomedical images (Long et al., 2014; Ciçek et al., 2016; Esteva et al., 2017)

1.4 FORMAL METHODS FOR BIOLOGICAL PATTERN OPTIMIZATION

While traditional machine learning approaches are well suited for segmenting biomedical images and recognizing patterns, *in silico* driven design of biological patterning can also be addressed using formal methods approaches. Formal methods are mathematically based approaches to specify and verifying the behavior of software and hardware systems. Formal methods have been used in diverse applications such as specifying and verifying spatiotemporal behaviors of power grids (Haghighi et al., 2015), or planning the motion of ground vehicles (Vasile, 2016). Formal methods utilize specification languages such as Linear Temporal Logic (LTL) or Signal Temporal Logic (STL) to describe how behaviors of systems change over time or space (Gol et al., 2014; Haghighi et al., 2015). These specification languages can be combined with measures that describe the distance an observation is from the specification.

This combination of specification and verification of the desired behaviors has recently been applied to biological patterning (Gol et al., 2014; Briers et al., 2016). In order to control biological pattern formation, the machine learning or statistical measure of the pattern must produce a quantitative metric representing a notion of distance between the observed pattern and the desired pattern. Pattern metrics that produce a binary classification ('is a pattern' or 'not a pattern') do not allow iterative refinement of design. With quantitative evaluations of complex specifications we can use previously developed tools from the formal methods community to optimize biological patterns and shapes *in silico*.

1.5 OVERVIEW OF CHAPTERS

In the following chapter I develop computational models of multicellular mammalian systems and formulate pattern design as an optimization problem. My research involves the application of formal methods rather than the development of these approaches. In the chapters 2 and 3, I describe how agent-based models of multicellular patterning can be combined with formal methods to direct self-organized multicellular pattern formation in human and murine stem cells. I chapter 2, I model how CRISPR interference induced perturbations of cell-cell adhesion, cell membrane stiffness, and cell locomotion produce emergent patterning in large stem cell populations. I also demonstrate how a formal methods based framework (Pattern Synthesis) of model driven design allows us to control the location and early lineage commitment of pluripotent cells. This framework can provides precise control of sub-populations during the development of stem cell-derived tissues and organoids. In chapter 3, I demonstrate how the Pattern Synthesis framework can discover controls that push transient patterns observed in murine stem cells into a steady state patterns. In Chapter 4, I developed a computational model enabling the programmatic assembly of cells into 3D user-defined structures. While chapters 2 and 3 focus on modeling dynamic patterning of sub-populations, chapter 4 focuses on modeling the creating of geometric *building blocks* from cells, which in-turn self-assemble into a larger 3D structure.

CHAPTER 2

Pattern Design Automation in Human Pluripotent Stem Cells

During the early stages of embryonic development, patterned self-assembly of cells is essential for the organization of primitive germ layers, multicellular tissues, and eventually complex organ systems (Montero & Heisenberg, 2004). Similarly, human pluripotent stem cells (hPSCs) maintain the ability to self-organize, differentiate to all three germ layers, and generate 3D organoids that replicate primitive tissue structure and function (Bredenoord et al., 2017; Warmflash et al., 2014; Sasai, 2013) hence, hPSCs provide a robust system to simultaneously observe, quantify, predict, and ultimately control collective cellular behaviors (Pir & Le Novère, 2016). The ability to direct how heterotypic cell populations self-organize and coordinately specify cell fate impacts many aspects of tissue engineering, including the possibility of directing organogenesis via novel cell-intrinsic routes.

Several in vitro and in silico frameworks for multicellular patterning have been developed, however the ability to predict and direct de novo multicellular organization has yet to be demonstrated. Previously, several groups (Molitoris et al., 2016; Tewary et al., 2017; Warmflash et al., 2014) have observed radial organization of differentiated germ layers induced by restricting hPSC colonies to micropatterned islands, or have used molecular engineering of cell surface and/or substrate properties to extrinsically control cell location and subsequent multicellular patterning in vitro (Chandra Ravi A. et al., 2005; Hsiao et al., 2009; Molitoris et al., 2016; Toda et al., 2018). However, the resulting patterns either arise spontaneously with limited control over organization, or are extrinsically driven and override the mechanisms that regulate cell-intrinsic morphogenic assembly. Theoretical in silico frameworks exist to computationally model multicellular organization (Bartocci et al., 2016a; Briers et al., 2016; Sharpe, 2017) and automate the design of non-spatial cellular logic (Nielsen et al., 2016). However, although computational techniques can test general principles of biology in silico, it is typically difficult to directly map these models to specific in vitro mechanisms and perturbations, making it challenging to systematically synthesize experimentally tractable perturbations in silico that can be accurately reproduced in vitro.

In this study, we paired quantitative experimental perturbations of human induced pluripotent stem cells (hiPSCs) with computational modeling, machine learning, and optimization to facilitate a closed loop cycle of in silico hypothesis generation that could be experimentally validated in vitro. To predict multicellular pattern formation, we combined a multi-scale Cellular Potts model (Graner & Glazier, 1992; Krieg et al., 2008; Magno et al., 2015; Marée et al., 2007; Ouchi et al., 2003; Pir & Le Novère, 2016) of mechanically driven cell sorting with an automated machine learning and optimization procedure, referred to as Multicellular Pattern Synthesis (Bartocci et al., 2016a; Briers et al., 2016), that consisted of four steps (Fig. 1). First we created a computational model of observed hiPSC self-organization that quantified collective stem cell dynamics and captured how targeted changes in sub-populations of cells affected stem cell colony patterning. Second, a supervised machine learning classifier was then trained to quantify pattern similarity using images from our computational model. Third, we employed mathematical optimization, using Particle Swam Optimization (PSO), to simulate thousands of potential designs and discover experimental conditions that produce specified patterns in in silico simulations. Finally, we found that treating hiPSCs with the in silico predicted conditions resulted in the formation of the desired pattern formations in vitro, validating the system. Furthermore, directed multicellular patterning led to predictable spatial differentiation of hiPSCs to different germ lineages.

2.1 PATTERN SYNTHESIS: IN SILICO PREDICTION AND AUTOMATED DISCOVERY OF SPATIAL BEHAVIORS

To observe multicellular pattern formation, we used a previously established hiPSC line with a DOX inducible CRISPRi system, allowing for temporal gene knockdown (KD) wherein mixed populations establish KD in only a portion of the colony, creating a symmetry breaking event and subsequent pattern formation (Libby et al., 2018a; Mandegar et al., 2016). However, the generation of new patterns in a predictable manner requires the ability to test large numbers of experimental conditions that would require a massive amount of time and manual effort to comprehensively test the vast number of experimental parameters possible. For example, considering only a single gene KD where KD timing, length of experiment, degree of KD, and proportion of the population that displays the KD are varied, over 5000 experimental conditions are necessary to combinatorially test the effect of that KD. Alternatively, a machine learning and optimization algorithm, such as Pattern Synthesis (Bartocci et al., 2016a; Briers et al., 2016), can automatically and efficiently discover experimental conditions and robustly predict the de novo self-organization of hiPSCs into desired target patterns.

Pattern Synthesis required two inputs: a model of hiPSC behavior, and images of the desired pattern (i.e. goal) outcomes. First, we developed a computational model of hiPSC colony organization as a result of a single gene KD (Fig. 2.1A). Next, we generated images of desired and undesired spatial patterns to train a machine learning algorithm that establishes a pattern classifier with a quantitative metric of pattern similarity (Fig. 2.1B) (Bartocci et al., 2016a; Haghighi et al., 2015). Given these inputs, we formalized pattern discovery as an optimization problem where the objective was to maximize the similarity score of images from our computational model to our desired spatial pattern (Fig. 2.1C). The variation between different simulations was based upon five categories of in vitro perturbations that could be readily created in hiPSC colonies (Fig. 2.1D).

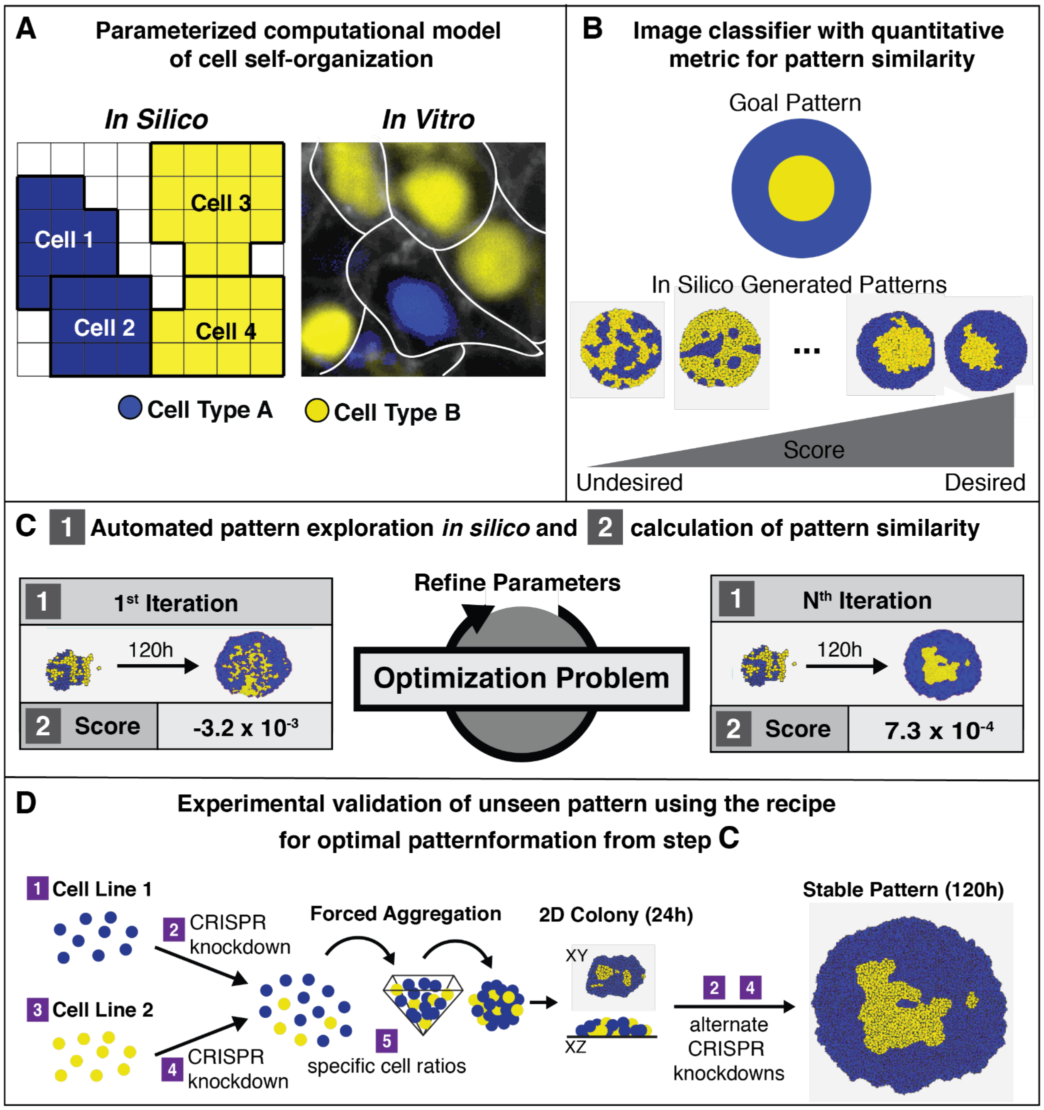

A
Parameterized computational model of cell self-organization
In Silico
In Vitro
Cell 1
Cell 2
Cell 3
Cell 4
Cell Type A
Cell Type B
B
Image classifier with quantitative metric for pattern similarity
Goal Pattern
In Silico Generated Patterns
...
Score
Undesired
Desired
C
1 Automated pattern exploration in silico and 2 calculation of pattern similarity
1 1st Iteration
120h
2 Score
-3.2 x 10-3
Refine Parameters
Optimization Problem
1 Nth Iteration
120h
2 Score
7.3 x 10-4
D
Experimental validation of unseen pattern using the recipe for optimal patternformation from step C
1 Cell Line 1
2 CRISPR knockdown
3 Cell Line 2
4 CRISPR knockdown
Forced Aggregation
5 specific cell ratios
2D Colony (24h)
XY
XZ
2 4
alternate CRISPR knockdowns
Stable Pattern (120h)

Figure 2.1: Pattern Design Automation.
Pattern Design Automation is an computational framework to predict and control multicellular spatial patterning. A) The first input to pattern synthesis is a computational model of mechanically driven spatial patterning in iPSC colonies. Five parameters of the computational model map directly to perturbations that can be made by experimentalists, and the output of the model was a series of images. B) The second input to pattern synthesis in a trained image classifier that produces a numerical score indicating the similarity of an image to the desired pattern class. In this scenario our desired pattern was a Bullseye pattern. C) Given the parameterized model and pattern classifier, Particle Swarm Optimization was used to explore parameter combinations, which map directly to in vitro perturbations, in order to identify the optimal conditions to produce the desired pattern in silico. D) Given the recipe of perturbations suggested by parameter optimization, we validate the control strategy is able to produce the desired pattern in vitro.

2.2 COMPUTATIONAL MODEL OF HUMAN PSC SELF-ORGANIZATION

Several different experimental and computational studies support the vital role of local cell-cell mechanical interactions in spatial patterning. Most notably, differences in cell-cell adhesion (Foty & Steinberg, 2005; Jia et al., 2007; Steinberg, 1975), cell-cell repulsion (Taylor et al., 2017), and cortical tension (Brodland, 2002; Krieg et al., 2008) have been shown to orchestrate spatial organization in diverse organisms such as Dictyostelium discoideum (slime mold) (Kay & Thompson, 2009; Palsson, 2008), Danio rerio (zebrafish) (Merks et al., 2008), and in human cells (Bentley et al., 2014; Toda et al., 2018). However, it is challenging to both predict and control spatial patterning in human iPSCs since the design of multicellular systems rapidly increases in complexity when considering the dynamics of single cell mechanics and cell-cell interactions. These dynamics include, but are not limited to, temporal changes in cell-cell adhesion and cortical tension associated proteins, cell type abundance, cell division, and cell migration velocities.

To capture the complex dynamic interactions involved in multicellular patterning, we developed a data-driven Cellular Potts Model (CPM) to predict spatial patterning in hiPSCs due to the time-dependent modulation of cell-cell adhesion and cortical tension (Supplementary Text). The Cellular Potts Model (CPM) represents the spatial environment of stem cells grown in a monolayer using a 2D square lattice grid (Fig. 2.1A). Each square region in the grid (i.e. a lattice site) is equal to 1 square micrometer, hence each lattice site represents a partial region of a cells membrane or the medium surrounding a cell. A cell ID is assigned to each lattice site to identify the region of a cell that occupies a lattice site. For example, 100 lattice sites each having a cell ID equal to 1 represent a single stem cell with an area of 100 square micrometers. Complex behaviors such as preferential cell-

cell adhesions, cortical tension, and cell migration, are achieved by copying lattice sites to adjacent regions, which in the CPM is known as a copy attempt. Each copy attempt is accepted with a probability related to a Hamiltonian function (SI Equations 3-5). The Hamiltonian function is the sum of four competing forces influencing intracellular behaviors and cell interactions with the environment: 1) conservation of cell area, 2) locally polarized cell migration, 3) cell-cell adhesion, 4) and cortical tension (SI Equations 5-10). Every competing force is represented by a score and a weight, where the score represents a reward or penalty depending on the divergence of a cell from its target behavior, while the weight represents the relative importance of the respective cell behavior.

Briefly, using the CPM, we modeled an in vitro system consisting of two populations of iPSCs co-cultured for up to 120 hours. The model incorporates numerous biological parameters, such as cell area conservation, cell proliferation, cell-cell adhesion, intracellular cortical tension, cell migration, and time-dependent modulation of properties implicated in cellular organization as a result of mosaic KD. To connect the in silico model to potential genetic targets for in vitro experimental manipulation, we focused on modeling changes in cell-cell adhesion via E-cadherin (CDH1) and changes in cortical tension via Rho associated coiled-coil containing protein kinase (ROCK1) levels. CDH1 is a classical cadherin cell-cell adhesion molecule, whose modulation allows for changes in the adhesive interactions between neighboring cells, and ROCK1 is a protein kinase that regulates non-muscle myosin activity and indirectly modulates the actinomyosin cytoskeletal tension within and between cells.

To fit the in silico model to an in vitro experimental training set, pairwise in vitro characterization experiments were performed to determine the relationship between space, time, and protein expression (Fig. 2.2A) in wildtype (WT), CDH1 KD and ROCK1 KD hiPSCs. These relations were established by in vitro measurements of single cell morphological changes (Fig. 2.S1), migration velocities (Fig. 2.2B-D), protein expression changes (Fig. 2.2E-G), and colony organization (Fig. 2.2H-I) before and after mosaic KD of CDH1 and ROCK1 in hiPSC colonies. The purpose of these characterization experiments was two-fold: 1) to reduce the complex interactions into quantifiable relationships, 2) and to create a closed-loop mapping between in vitro perturbations and in silico simulation parameters.

To characterize cell morphology, brightfield images of wild-type (WT), CDH1(-), and ROCK1(-) cells were collected 144 hours after gene knockdown. Single cell in vitro cell area and membrane length measurements (Fig. 2.S1) were acquired to set the target cell area and target cortical tension in the simulations, respectively. In the CPM, the weight associated with the cortical tension constraint regulates how readily a cell can change its cell membrane length and relates to cell membrane stiffness. Due to differences in cell crowding in the center versus the edge of colonies, cell morphology measurements were fixed given a cells mechanical modulation and its radial position in the colony (Table 2.S1, Fig. 2.S1). Cell division was assumed to be asynchronous amongst the population, and cell division times specific to each type of knockdown were incorporated into the model to provide an accurate depiction of population growth kinetics.

The relationship between cells in space with respect to time was characterized by measuring the in vitro distribution of individual cell velocities, resulting in an empirical median velocity of 0.29 +/- 0.17 m/minute (Figs. 2.2B, 2.2D). The distribution of velocity values was then used to model collective cell migration as locally polarized motility where the direction of cell migration is influenced by the relative cell adhesion strength of neighboring cells (Czirók et al., 2013; Szabó & Merks, 2013). Cell-cell adhesion and cortical tension parameters for WT cells were selected to mimic the in vitro velocity measurements, producing a comparable distribution with a median in silico velocity of 0.31 +/- 0.12 m/minute (Figs. 2.2C, 2.2D). Importantly, the in silico generated velocity distributions were not significantly different from the in vitro measured velocities (Mann-Whitney U test, p = 0.05). An additional requirement that individual cells remained part of dense epithelial colonies without migrating from the exterior borders was also imposed to match the hiPSC phenotype observed in vitro. After fitting the model to empirical data of cell morphology and velocity, single cell morphology and collective cell migration of human iPSC colonies were accurately recapitulated.

To examine the relationship between protein expression time, CDH1 and ROCK1 were knocked down using CRISPRi, and the relative mRNA and protein expression was assessed for 6 consecutive days via qPCR, fluorescence microscopy, (Fig. 2.2E, Fig 2.S2) and western blot analyses (Libby et al., 2018a). Due to our previous observation of the phenotypic robustness of CDH1 knockdown in promoting cell self-organization (Libby et al., 2018a), we designed several CRISPRi guide RNAs to target CDH1 producing different levels of transcriptional knockdown at 10%, 25%, 30%, and 98% compared to WT expression. A single guide RNA for ROCK1 knockdown was used to achieve 80% knockdown of WT expression levels (Fig. 2.7). The data was normalized (min-max [0,1]), and using least squares optimization, Hill Functions were fit to the normalized median expression (per day) to create a time-dependent response function for CDH1 knocked down to 90%, 75%,70%, and 2% of the original mRNA expression (Fig. 2.2F). This range of knockdown efficiency allowed us to computationally model how differing levels of CDH1 expression could impact spatial patterning. Using the same approach as the CDH1 knockdown, we created a Hill response function for ROCK1 knocked down to 20% mRNA expression. Because of a delay in protein knockdown compared to mRNA levels, the Hill functions were shifted by 24 hours to account for the delay in protein loss (Fig. 2.2G), allowing us to model the average change in ROCK1 protein expression for individual cells over time.

Given the previous characterization experiments, we were able to model collective cell migration and temporal changes in cell mechanics. To model the spatial patterning due to the temporal modulation of cell-cell adhesion via CDH1 or cortical tension via ROCK1, either inducible ROCK1 knockdown or inducible CDH1 knockdown iPSCs were co-cultured with WT iPSCs, where knockdown of gene expression was induced upon mixing the two cell types. Then, images of the mixed populations were collected 96 hours after gene knockdown and co-culture. As previously reported (Libby et al., 2018a), mixed colonies with a subpopulation of cells that had reduced CDH1 or ROCK1 expression produced distinct mosaic patterns due to reduced cell-cell adhesion and increased membrane stiffness properties respectively (Figs. 2.2H-I (left)). In silico, parameter sweeps were run over a range of adhesion strength and membrane length values to explore the phenotypic space resulting from decreases in cell-cell adhesion and increases in membrane stiffness. Computationally varying the adhesion strength produced a variety of spatiotemporal patterns due to progressively weaker cell-cell adhesion or progressively stiffer cell membrane parameter values. Double-blind analysis of in silico and in vitro generated data sets was then conducted to identify parameters that yielded closely matching multicellular patterns (Figs. 2.2H-I (right)). Given the characterization experiments of cell morphology, cell migration velocity, time-dependent modulation of cell mechanics, and the resulting spatial organization, the computational model was able to recapitulate the spatial patterning due to the CDH1 and ROCK1 knockdowns.

Overall, after incorporating in vitro measurements into our computational model, we accurately recapitulated hiPSC spatial patterns with the initial experimentally derived parameters in mixed colonies of WT and CDH1 KD cells or WT and ROCK1 KD cells (Libby et al., 2018a). The full detailed description of the characterization experiments is outlined in the methods subsection.

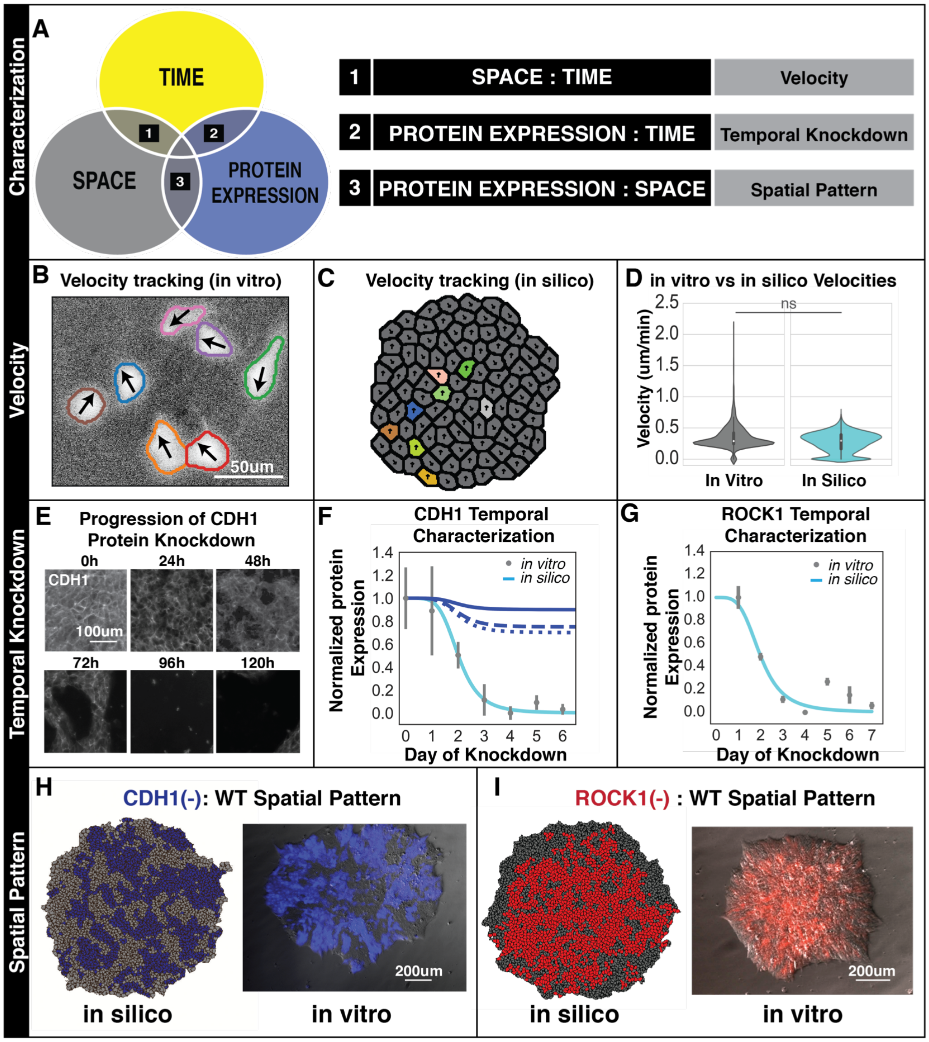


Figure 2.2: Pairwise experiments to characterize dynamic changes in spatiotemporal behaviors.

A) We characterize cell behaviors in a pairwise manner to reduce the complexity of possible interactions. B) We performed paired in vitro and C) in silico experiments to match the velocity distributions of iPSCs. D) The grey violin plot represents the distribution of in vitro velocities, while the cyan violin plot represents the distribution of in silico velocities. Using the Mann-Whitney U test, there was no statically significant difference in cell velocity (p value < 0.05). E) Representative images of DOX inducible modulation of protein expression. F) We used Hill Functions to mathematically model CDH1 knockdown over time from quantification of mRNA by qPCR and then adding a 24h delay to account for protein production. Grey circles represent the normalized median expression 0-6 days after CDH1 knockdown. Error bars represent 1 standard deviation from the median. G) A Hill Function to model ROCK1 knocked down. H) Paired in vitro and silico images of spatial patterning 96 hours after CDH1 knockdown in a subpopulation of cells (blue). I) 96 hours after ROCK1 knockdown in a subpopulation of cells (red).

Design Parameter	Parameter Domain
Cell Line 1 (CL1)	CDH1-0, CDH1-70, CDH1-75, CDH1-90, ROCK1-20, WT
Cell Line 2 (CL2)	CDH1-0, CDH1-70, CDH1-75, CDH1-90, ROCK1-20, WT
Knockdown Time of CL1	[-120 hours, 120 hours]
Knockdown Time of CL2	[-120 hours, 120 hours]
Abundance of CL2	[.05, 0.95]

Table 2.1: List of design parameters that map to experimental perturbations. CL1 and CL2 are chosen from our library of mechanically tunable cell lines: CDH1-0, CDH1-70, CDH1-75, CDH1-90, ROCK1-20,Wildtype. The number following the cell line indicates the relative expression of the gene in comparison to WT. The knockdown times of CL1 and CL2 range from 120 hours before co-culture to 120 hours after co-culture in 24 hour increments. Finally, the abundance of CL2 cells in relation to CL1 can be vary from 5 percent to 95 percent of the colony in increments of 5 percent.

2.2.1 Formulating Parameters for Design Automation

Given the success of the computational model to accurately recapitulate hiPSC spatial patterns with the initial experimentally derived parameters, five new design parameters were introduced to simulate in vitro experimental perturbations. While the previous sections describe the typical model parameters to match the behavior of a computational model to experimental data, design parameters allowed us to mimic changes in experimental design by human experimenters. The five design parameters are: the gene knockdown target of the first cell type, the knockdown time for cell type 1, the gene knockdown target of cell type 2, the knockdown time for cell type 2, and the ratio of the distinct cell types (Table 2.1). These additional design parameters allowed us to advance past simply recapitulating known patterns to quantitatively predict novel spatiotemporal self-organization behaviors of hiPSCs.

2.3 QUANTITATIVE PATTERN CLASSIFICATION

The second input to the Pattern Synthesis procedure was a supervised image classifier known as Tree Spatial Superposition Logic (TSSL) (Bartocci et al., 2016a). TSSL uses a quadtree data structure to represent spatial relationships in an image at multiple levels of detail, where the highest level captures global aspects of an image, while the lower levels capture local spatial relationships (Fig. 3A-B) (Bartocci et al., 2016a; Finkel & Bentley, 1974; Jackins & Tanimoto, 1983). A rule-based machine learning algorithm (RIPPER) (Cohen, 1995) was employed to automatically learn a set of rules over the values of quadtree vertices specific to an in silico training set of 3,000 positive and 13,000 negative manually rendered images of cells precisely organized into target patterns, such that a quantitative score of pattern similarity could be assigned to any image from the associated quadtrees (See Appendix A). The magnitude of the similarity score, which can range from -1 to +1, indicates how strongly a simulation image matches (positive scores) or violates (negative scores) the target spatial behavior. Use of a TSSL robustness score replaces qualitative manual observation of simulation images with a quantitative score of pattern similarity (Fig. 2.3C). By quantifying how well images from an in silico multicellular arrangement matched images of our target organization, we enabled the optimization algorithm (described in the next section) to incrementally improve and learn a unique combination of design parameters that could give rise to a desired goal pattern.

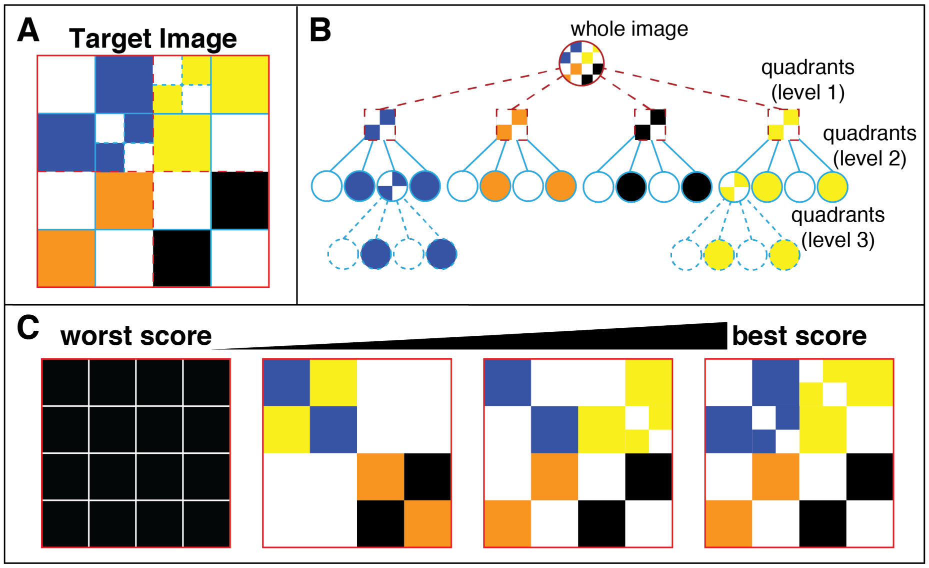


Figure 2.3: Quantitative Pattern Classification with TSSL.
A quadtree is used to represent an image at multiple levels of detail. A-B) Shows a representative quad tree for an example image. An image A) is subdivided into sequential quadrants until the color value of the quadrant is one singular color. Both the values and branches of the tree are specific to each image. C) Given a quadtree representation of a target image, TSSL produces a numerical score corresponding to the similarity of an image to the desired target image.

2.4 AUTOMATED DISCOVERY OF PATTERN PRODUCING CONDITIONS

The mathematical model presented in Section 1 allows us to simulate more than 40,000 distinct parametric conditions and study the emerging behaviors of hiPSCs much faster than in vitro experiments. Distributing the computation over 12 processors at 2.1 GHz on a server cluster, it only took approximately 5 minutes to simulate the evolution of one cell population over 120 hours, thereby making it possible to perform over 1300 in silico experiments in the same time required for a single in vitro experiment. The speed of simulation allowed us to examine a wide range of different experimental conditions in a rapid and inexpensive manner. However, due to the tens of thousands of experimental conditions to consider and the resulting months of computation for such a large number of simulations, it is quickly becomes impractical to simulate every possible set of conditions to identify parameter combinations that yield the highest robustness score(s). Thus, to more efficiently explore distinct combinations of design parameters, we employed Particle Swarm Optimization (PSO) . Given the TSSL pattern classifier and the parameterized model of cell sorting, Particle Swarm Optimization (PSO) was used to identify the optimal set of conditions to produce the desired spatial patterning.

Early development is dependent on a range of different cell patterning events for proper germ layer specification, neural tube formation, limb bud generation etc. that would be beneficial to mimic in an in vitro setting. We decided to first attempt a concentric ring (i.e. Bullseye) pattern, defined as one population of 50 or more connected cells completely surrounded by a second population (Fig. 2.4A). The annular Bullseye pattern was chosen because similar asymmetric cell organization occurs multiple times in human development, such as during early embryo compaction leading to the formation of the inner cell mass in the human blastocyst. We also set a secondary target of Multi-Island goal patterns, consisting of at least three distinct clusters of 25 or more cells completely surrounded by a separate larger population (Fig. 2.4B), which is reminiscent of embryonic hematopoiesis during blood islet generation in the developing liver. We explored five independent experimental conditions that directly influenced the appearance of spatial patterns in 100 cell heterotypic colonies of human iPSCs after 120 hours of co-culture. These 5 experimentally tunable conditions, previously described as design parameters, are listed in Table 2.1.

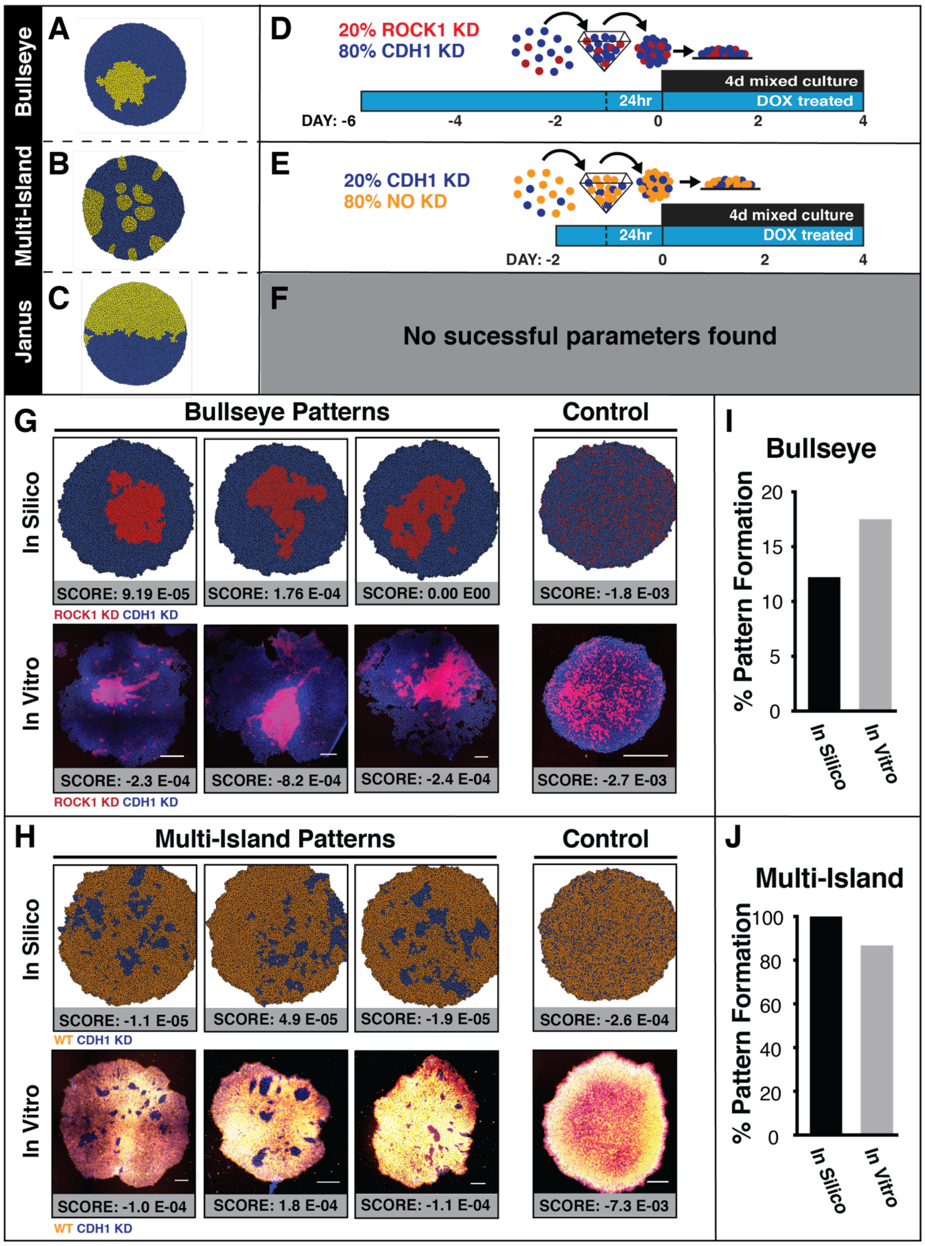


Figure 2.4: Computational synthesis of de novo spatial patterns and in vitro validation A-C) Schematics of example target patterns given as classifiers in the machine learning pattern synthesis process. D-F) Parameters produced by Pattern Synthesis that predict the creation of the desired patterns: Bullseye, Island, Janus. G-H) Comparisons of patterns predicted in silico and the resulting patterns seen in vitro. Pluripotent colonies labeled for DAPI (blue) and CDH1 (red/orange) to distinguish populations. TSSL robustness scores show how well a simulation matches or in vitro experiments match the specification. I-J) Successful pattern creation rates, comparing in silico to in vitro results (Bullseye n = 286 colonies, Multi-Island n = 168 colonies).

To find the optimal conditions that yielded target spatial patterns, we automated the design of our experimental conditions. TSSL classifiers were trained from an initial in silico training set consisting of 3,000 positive and 13,000 negative examples for either target pattern (Bullseye or Multi-Islands) individually. The classifiers were tested on an independent in silico set of 1,000 positive and 5,000 negative examples with a 98.2 percent classification accuracy for the Bullseye and 96.9 percent classification accuracy for the Multi-Islands pattern, demonstrating that the automated classifiers could reliably detect and distinguish between desired and undesired spatial patterns. Given the classifier for each pattern and the parameterized model of cell sorting, PSO was used to determine the parameterization of the model for the highest robustness score with respect to each pattern. In order to accommodate the stochastic nature of the simulations, the average robustness score for three distinct simulations was computed for any fixed set of parameter valuations. The PSO algorithm was able to determine optimal parameters for each type of pattern within 20 iterations (equivalent to 2,160 simulations), requiring approximately 8 consecutive days of parallel computation on 12 processors at 2.1 GHz for each pattern. For any in silico simulation that generated a successful pattern (Fig 2.S6), a time course of the simulation was evaluated to demonstrate the TSSL score improved by at least one order of magnitude and reached a steady state (Fig 2.S7). Analyzing the temporal dynamics of robustness scores (pattern similarity scores), provided insight into the exact time a pattern emerged in silico, and optimized design parameters for target patterns that did not perfectly match our specification, but still resembled the desired spatial behavior.

In addition to automating the design of de novo spatial patterns, we could also determine the feasibility of any spatial pattern given the tunable conditions in our system. It is noteworthy that the resulting robustness score can be negative (violating the pattern specification), indicating that the cell population is unable to perfectly recapitulate the desired spatial behavior. For example, the PSO algorithm was able to determine that a perfectly symmetrical Janus pattern (left-right) was not achievable with the primary experimental variables (i.e. timing of CDH1/ROCK1 knockdowns and the ratio of cell types co-cultured in an approximately 2D monolayer). (Fig. 2.4C).

2.5 IN SILICO MODEL ACCURATELY PREDICTS IN VITRO EXPERIMENTAL VALIDATION

The Patterning Synthesis algorithm yielded different sets of instructions to produce either a Bullseye pattern or a Multi-Island pattern of hiPSCs (Fig. 2.4D-F). The Pattern Synthesis predicted that a mixture of 1:4 ROCK1 KD iPSCs to CDH1 KD iPSCs that were independently pretreated with DOX for 6 days prior to mixing and cultured together for 4 days was needed in order to achieve a Bullseye pattern (Fig. 2.4D). In contrast, to generate Multi-Island patterns, the Pattern Synthesis predicted that a mixture of WT cells with CDH1 KD at a ratio of 1:4 with DOX pretreatment of iPSCs for 48h prior to mixing (Fig. 2.4E).

Based on these predictions, in vitro experiments were performed using the exact specified conditions and the incidence of pattern formation was independently analyzed for in silico and in vitro results. We characterized the morphology of in silico and in vitro generated patterns by interrogating subpopulation cluster circularity, number of clusters, and cells per cluster within the colony (Fig. 2.S8). The in silico model predicted that under these conditions a Bullseye Pattern would be obtained about 12 percent of the time and a Multi-Island Pattern achieved 100 percent of the time (Figs. 2.4I, 2.4J). We attained comparable results from our in vitro experiments where we obtained a bullseye in about 16 percent of the in vitro experiments run (Figs. 2.4G, 2.4I). Whereas Multi-Island Patterns appeared about 87 percent of the time in the in vitro experiments (Figs. 2.4H, 2.4J).

In addition, TSSL robustness scores were generated for both the in silico and in vitro experiments. The TSSL robustness scores demonstrate that the optimal bullseye and islands patterns have large robustness scores in comparison to their respective control images (at least an order of magnitude difference). It is important to note that the TSSL classifier scores are highly comparable only when they are calculated in the same setting; thus, a simulation vs a simulation control is quite comparable whereas an in silico simulation vs an in vitro experimental image will inherently differ to some extent. Image classification in different environments is a well-known limitation of machine learning algorithms , yet we could still use the previously discussed spatial metrics (i.e. cell cluster number, size, circularity) to directly compare simulation and experimental images and quantitatively establish improvements in pattern formation from the control experiments in an unbiased manner.

Notably, the in vitro experimental results were more variable and yielded a wider range of experimental results. The difference between in silico and in vitro could be due to the biological variability involved in wet lab experimentation. Additionally, the model only accounted for genetic manipulation of CDH1 and ROCK1 and did not take into account possible stochastic differences in cellular gene regulation, which may cause subtle variations in cellular behavior that the model would be unable to predict in silico. However, overall, these results demonstrate that in silico modeling could accurately classify and predict novel pattern formation achieved by hiPSC self-organization in vitro.

2.6 COLONY ORGANIZATION IMPACTS PATTERNS OF IPSC DIFFERENTIATION

During human development, cell autonomous pattern formation is intimately coupled with cell fate decisions that altogether lead to complex tissue structures. However, current methods offer limited control over multi-lineage differentiation of PSCs and multicellular organization of the differentiating sub-populations. Therefore, we interrogated how the experimentally generated multicellular patterns predicted by the computational modeling affected subsequent human iPSC differentiation. We examined the initial cell fate commitment after treatment with BMP4 (Fig. 2.5A), where cells remain either OCT4(+) or SOX2(+) before differentiating to mesendoderm or ectoderm, respectively. In this case, an OCT4(+)SOX2(-) cell indicating a cell that has begun to differentiate toward the mesendodermal lineage and a OCT4(-)SOX2(+) cell indicating a cell that have begun to differentiate toward the ectodermal lineage (39). WT control cell colonies displayed a radial differentiation pattern with OCT4(-)SOX2(+) cells occupying the center region while cells at the edges of the colony were OCT4(+)SOX2(-), similar to previously published reports (3,7). A similar radial pattern was maintained in pure ROCK1 KD cell colonies, although the area of OCT4(-)SOX2(+) cells was expanded compared to the WT and the observed colonies were dome shaped with greater vertical height than the WT colonies. Interestingly, pure CDH1 KD iPSCs lost the radial differentiation pattern in fate specification exhibited by the WT control and ROCK1 KD cells. In contrast to the ROCK1 KD cells, pure CDH1 KD cells had a reduced amount of OCT4(-)SOX2(+) and maintained similar z-dimension heights to the

WT colonies. The loss of the radial differentiation phenotype indicated that CDH1 potentially regulates position sensing of individual hiPSCs within the larger context of cohesive colonies. In the differentiating Bullseye colonies, the mixed populations behaved similar to their respective pure KD populations. The ROCK1 KD population at the center of the Bullseye had a larger OCT4(-)SOX2(+) population and displayed equivalent heights in the z-plane as seen in the pure ROCK1 KD populations. Similarly, the surrounding CDH1 KD population in the bullseye patterns lost radial symmetry and displayed uneven differentiation patterns similar to the pure CDH1 KD colonies (Fig. 2.5B). Thus, the genetic manipulations used to control multicellular organization were also indicative of the resultant differentiation phenotype of the patterned colony.

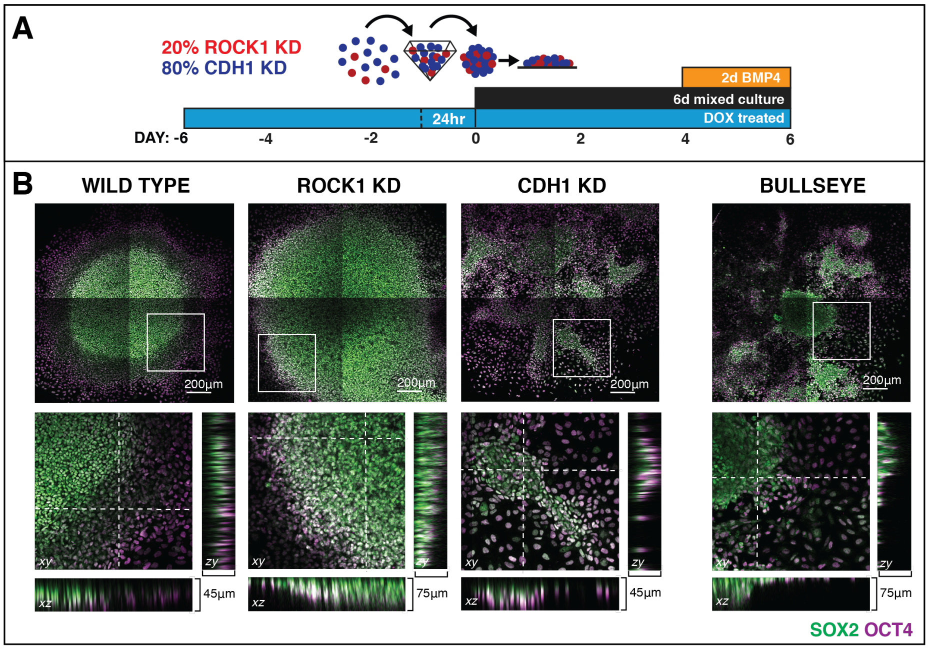


Figure 2.5: Differentiation of Patterned colonies
A) Schematic of two day differentiation induced by addition of BMP4 to cell culture media. B) Tiled confocal images of pure population control differentiations followed by Bullseye pattern differentiation stained for OCT4 (purple) and SOX2 (green) expression. Inserts show z-dimension where the height of the colony varies between the Wildtype and CDH1 knockdown (KD) colonies and the ROCK1 knockdown (KD) and Bullseye patterned colonies (45um vs 75um respectively).

2.7 DISCUSSION

Cell-intrinsic patterning of multicellular stem and progenitor populations is a critical feature of morphogenic events that occur throughout early development (Deglincerti et al., 2016; Ducibella & Anderson, 1975; Montero & Heisenberg, 2004; Sasai, 2013). Thus, systems in which multicellular organization can be robustly controlled and perturbed will help to elucidate key mechanisms in development and symmetry breaking events. Currently, the study of symmetry breaking events often involves the manipulation of cell extrinsic factors, for example, varying morphogen gradients (Demers et al., 2016; GeunChung et al., 2005), changes in substrate patterning (Hsiao et al., 2009; Théry et al., 2006) and/or the creation of restrictive boundary conditions (Tewary et al., 2017; Théry, 2010; Warmflash et al., 2014). In contrast, attempts to influence patterning events using synthetic biology approaches often rely on implementation of an artificial circuit that uses reaction diffusion gradients to establish multicellular patterns (Greber & Fussenegger, 2010; Sohka et al., 2009).

In this study, we demonstrate the induction of active multicellular organization through controlled perturbation of intrinsic cell mechanisms without imposing exogenous boundary conditions. We developed a computational model capable of predicting empirically testable molecular perturbations to generate desired multicellular spatial patterns in hiPSC colonies. Using agent-based model predictions of spatiotemporal pattern formation, we were able to predict and achieve new patterns in silico and in vitro without using extrinsic patterning methods (i.e. hydrogels, micropatterning). Optimized design parameters achieved desired organization of cells within a colony, and promoted the expected initial mesendoderm or ectoderm lineage fates. Ultimately, these results demonstrate that machine learning and mathematical optimization enable predictive and controlled spatial self-organization in heterogeneous populations of pluripotent cells, which is a critical initial step for hiPSC self-assembly prior to lineage commitment and subsequent organoid and tissue formation.

Previous attempts to pair computational models with experimental morphogenic systems have been largely observational and rarely demonstrate the ability to design phenotypes in silico that can be recapitulated in vitro. In this study, both the in silico and in vitro aspects can be adapted to additional parameters, truly taking advantage of machine learning and optimization to generate desired multicellular patterns. With respect to extending in vitro perturbations, CRISPR technology can be adapted to repress or activate any accessible genes related to cell patterning and organogenesis. As additional biological parameters are considered, we can quantitatively characterize the effect on cell patterning, and the in silico model can be refined to take those factors into account (Briers et al., 2016; White et al., 2015), enabling interrogation beyond cell mechanics and into other realms of cell-cell communication such as paracrine signaling gradients and gap junction connectivity (Glen et al., 2018; White et al., 2013). Ultimately, the combination of agent-based modeling, machine learning, and directed symmetry breaking provides a novel route to engineer complex multicellular tissue structures that go far beyond simple observation of pattern formation, and facilitate targeted mechanistic studies that address fundamental principles of development and morphogenesis.

2.8 MATERIALS AND METHODS

2.9 AGENT-BASED MODEL OF CELL SORTING

We modeled the mechanical properties of interacting human induced pluripotent stem cells (hiPSCs) with an extended Cellular Potts Model (CPM). In the model of mechanically driven self-organization in hiPSCs, cell-cell interaction mechanics were explained by four physical properties of cells. 1) cell-cell adhesion, 2) cortical tension, 3) conservation of volume, 4) and directionally persistent cell migration (Magno et al., 2015). Below, we describe how the extended CPM was used to recapitulate spatiotemporal patterns and predictively design de novo spatiotemporal behaviors.

2.9.1 Cellular Potts Model Environment

We defined the environment of a CPM simulation S on a 2D square lattice domain $S \in {\mathbb{Z}_+}^2$. Each lattice site, $x = (m, n) \in S$, represented a coordinate location where $m \in \mathbb{Z}_+$ and $n \in \mathbb{Z}_+$ were the horizontal and vertical coordinates of each lattice site respectively. The spatial resolution of each lattice site was $1\mu\mathrm{m}^2$.

To represent the location of hiPSCs, each lattice site x was assigned a value σ_x, conventionally called the *spin* or cell index of a site, from the set of cell indices $k \in K$ given $K = \{1, ..., N(t)\}$ where $N(t)$ was the number of cells in the simulation at time t. Lattice sites that represent empty space where there is no hiPSC covering the lattice site were assigned a cell index of 0. In the CPM, a stem cell C_k was composed of multiple lattice sites. A stem cell C_k was defined as the set of lattice sites with the same cell index $C_k = \{x \in S : \sigma_x = k\}$. Since a single cell was composed of multiple lattice sites, the CPM was able to capture fluctuations in a cell's shape with a granularity that is not possible with Type A cellular automaton or center-based models (Van Liedekerke et al., 2015) Each cell was also assigned a cell type τ that determined its intracellular and extracellular behaviors.

Next, we summarize two common metrics to describe cell morphology in a CPM simulation; cell area and cell membrane length. These metrics are important since their values in the model were directly measured from microscopy images. For a discussion of these metrics see (Magno et al., 2015; Voss-Bohme, 2012).

Given that each lattice site had an area of 1um, the area of a cell at time t in the simulation was defined as the number of lattice sites encompassed by a cell:

$$a_{k,t} = |x \in S : \sigma_x = k|, \tag{2.1}$$

where $|\cdot|$ denoted the cardinality of a set. The time varying membrane length of a cell $p_{k,t}$, synonymously called the perimeter or surface length in other studies (Magno et al., 2015; Voss-Bohme, 2012), was defined as the number of lattice interfaces bordering other cells or empty space:

$$l_{k,t} = 0.5 \times \sum_{interfaces\{x,x'\}} \delta(k, \sigma_x), \tag{2.2}$$

where x' represented any of the lattice sites adjacent to x, $(m\pm1, n) \vee (m, n\pm1)$ in 2D. The Kronecker symbol δ was defined by $\delta(u,v) = 1$ if $u = v$ and $\delta(u,v) = 0$ if $u \neq v$. An interface (x, x') was a shared border between lattice sites. To avoid counting adjacent lattice sites inside a cell, the CPM only summed interfaces between lattice sites with different cell ID's; when $\delta(\sigma_x, \sigma_{x'}) = 0$. Put simply, we were measuring the perimeter of each stem cell.

2.9.2 Cellular Potts Model Dynamics

The CPM used a function called the **Hamiltonian** H to describe the energy for any *configuration* of cells. Cell motility evolved by choosing a random lattice site x, a region of a cell-cell interface or a cell-media interface, and attempted to copy it to a random neighboring lattice site $x^{'}$. The Hamiltonian was defined as the sum of four constraints that represent four physical properties of simulated stem cells: 1) conservation of cell area, 2) locally polarized cell migration, 3) cell-cell adhesion, 4) and cell membrane length which commonly represents cortical tension. In the CPM, the goal was to minimize the Hamiltonian or minimize violations of the desired cellular behaviors. Therefore, each constraint calculated a decrease (reward) or increase (penalty) in the configuration energy due the collective properties of cells in the simulation. When a change in a lattice site was proposed, this affects

H. If the proposed change was accepted, the change in H was defined as $\triangle H$. A proposed change for a cell's lattice site was accepted with the following probability:

$$if \triangle H < -Y, P(\sigma \rightarrow \sigma') = 1 \tag{2.3}$$

$$otherwise, P(\sigma \rightarrow \sigma') = e^{-(\triangle H + Y)/T)}, \tag{2.4}$$

where the yield $Y = 0.1$ and the temperature $T = 10$. Simply, if the proposed change in local cell position resulted in less energy, then the change was accepted. If the proposed update would have resulted in greater energy (ΔH), then the change was only accepted with a very low probability.

CPM Configuration Energy: The free energy for a configuration of cells was defined as the sum of four constraints: local cell-cell/cell-ECM adhesion, cell area conservation, cell membrane length, and locally polarized cell migration:

$$H = H_{adhesion} + H_{area} + H_{membraneLength} + H_{migration} \tag{2.5}$$

For a configuration of cells, the free energy due to cell adhesion was

$$H_{adhesion} = \sum_{k \in K} J_{\tau(\sigma_x),\tau(\sigma_{x'})}(1 - \delta(\sigma_x, \sigma_{x'})), \tag{2.6}$$

where $J_{\tau(\sigma_x),\tau(\sigma_{x'})}$ represented the cell adhesion strength between lattice sites σ_x and $\sigma_{x'}$ that was defined by their cell type $\tau(\sigma_{x'})$. $(1-\delta_{\sigma_x,\sigma_{x'}})$ restricted these calculations to interfaces between cells instead of all lattice sites, and improved the efficiency of the simulation (Starruß et al., 2014). Although not explicit in our notation, the cell adhesion strength was a time-dependent function controlled by protein expression. The energy due to cells resisting changes from their resting area was defined as

$$H_{area} = \sum_{k \in K} \lambda_a (a_{k,t} - A_{k,t})^2, \tag{2.7}$$

where $A_{k,t}$ represented the target area of a cell, $a_{k,t}$ represented the current area of a cell, and λ_a was the relative strength of area conservation term.

The cortical tension constraint was defined as:

$$H_{membraneLength} = \sum_{k \in K} \lambda_l (l_{k,t} - L_{k,t})^2, \tag{2.8}$$

where $l_{k,t}$ represented the current membrane length of a cell at time t, $L_{k,t}$ was the target membrane length, and λ_l was the strength of the cortical tension constraint. As a proxy for increasing or decreasing the cell membrane length, the Equivalent Circular Perimeter (ECP) was used to set the membrane length for a cell given its current area. The ECP of a non-circular 2D object was defined as the perimeter of a circle with equivalent surface area as the non-circular object:

$$ECP(k) = 2\sqrt{a_{k,t}\pi}.$$

The target membrane length was calculated using a membrane length proportionality constant:

$$L_{k,t} = r_k ECP(k), \tag{2.9}$$

where r_k was the membrane proportionality constant. To find r_k the membrane length and area of cells were measured and divided by the ECP of the cell. This ratio of membrane length to ECP was equal to the membrane proportionality constant r_k. The ECP allowed us to calculate the membrane length of a cell of any area that would have a comparable shape to empirical measurements.

To capture directionally persistent cell migration, we modeled "polarized cell migration" as the tendency of cells to bias their movement in the same direction as their previous direction of movement as described in (Czirók et al., 2013; Szabó et al., 2010). Cells had a target direction $\vec{t}$ based on previous movement where CPM updates in this direction were preferred (they decreased the energy in H). For each copy attempt $x \rightarrow x'$, the cell center was displaced in direction $\vec{s'}$. The change in energy due to migration in this direction was defined as:

$$H_{migration} = -\mu a_k(\vec{t} \cdot \vec{s'}), \tag{2.10}$$

where μ was the strength of cell migration, $a_{k,t}$ was the cell area at time t, $\vec{t}$ was a unit vector giving the target direction, and $\vec{s}$ was a unit vector giving the current direction of a stem cell if the CPM update $(x \rightarrow x')$ was to be accepted. The function was multiplied by -1 since updates in the direction of $\vec{t}$ have a dot product that approached +1 as the angle between $\vec{t}$ and $\vec{s}$ approached zero. Multiplying by -1 resulted in decreased configuration energy for cells moving in the same direction as the target direction vector.

For every MCS, the target direction at any time $(\vec{t_t})$ was updated continuously given the displacement of a cell's centroid $\Delta O = \mathbf{O_t} - \mathbf{O_{t-1}}$. This target direction included how 'decay-time' D of the previous direction and the current cell displacements contributed to the current polarity of the cell (Starruß et al., 2014; Szabó et al., 2010):

$$\vec{t_t} = (1 - D)\vec{t_{t-1}} + D(\Delta O/|\Delta O|). \tag{2.11}$$

2.9.3 Physical Units and Other Cellular Phenomena

Cell division was symmetrical (the parent cell divided into 2 equally sized daughter cells), and the timing of cell division was asynchronous. This was achieved by assigning a uniformly distributed "division counter" d_c for each cell at $t = 0$ between 0 and the division time d_t. This counter was incremented at each time step of the simulation, and a cell would divide when $d_c = d_t$. d_c was then reset to 0 for both daughter cells. Cell division times were calculated from in vitro doubling rates and modeled to be 18 hours for CDH1(-) cells, and 20 hours for all other cells.

2.10 MODEL FITTING TO EMPIRICAL DATA

In the main text we provide a brief explanation of the characterization experiments to fit our computational model. Here we describe the mathematical transformations, mapping functions, and model parameters associated with these characterization experiments. The parameters fit during this process are summarized in Table S1.

2.10.1 Morphology Characterization

To characterize cell morphology, microscopy images of Wildtytpe (WT), CDH1 knockdown, and ROCK1 knockdown cells were collected 5 days (120 hours) after co-culture and gene knockdown. The three types of colonies were characterized; purely wildtype, wildtype and CDH1 knockdown in a 1:1 ratio, and wildtype:ROCK1 knockdown in a 1:1 ratio. We measured the cell area, perimeter, and ECP at the center and periphery of colonies (Fig. 2.6). The median cell area was used to set the target cell area ($A_{k,t}$) in our simulations (Fig. 2.6, Table 1). The median membrane proportionality constant (r_k = perimeter / ECP) was used to set the target membrane length $L_{k,t}$ in our simulations (Fig. 2.6, Fig. 2.16).

2.10.2 Velocity Characterization (Space vs. Time)

We then performed pairwise characterizations of space, time, and protein expression (Fig. 3). We characterized space vs. time by measuring the velocity (change in distance over time) of wildtype cells in dense colonies. Mixed aggregates of 90% WT and 10% CRISPRi cells without a targeting guide were generated. With the addition of doxycycline (DOX) to the cell culture media, the CRISPRi no guide population expressed a cytoplasmic mCherry marker which allowed individual cells to be distinguished from the untagged WT background (Fig. 2B). 24 colonies were imaged for 6 hours at 5 minutes/image creating a time series of 73 frames. Each frame was individually normalized and thresholded using non-local means (Otsu, 1979). Cell migration tracks were generated by following matching contours between frames where a matching contours share at least ten pixels overlap. We used watershed segmentation to separate adjacent cells. Instantaneous frame to frame velocity was calculated as

$$v_{inst} = [(x_{cm,2} - x_{cm,1})/\Delta t, (y_{cm,2} - y_{cm,1})/\Delta t], \tag{2.12}$$

where $x_{cm,2}$ was the center of mass of each segmented cell body at the currently observed frame and $x_{cm,1}$ was the center of mass of each segmented cell body at the previous frame, and Δt was 5 minutes. Taking the average magnitude of the per-cell instantaneous velocity over 24 colonies gave an median velocity of 0.29 um/minute.

We then ran parameter sweeps to fit model parameters that affect cell migration

(Table 1):

- MCS - copy attempts per simulation hour.
- $J_{WT,WT}$ - adhesion energy or reward per micrometer of cell border between wildtype cells.
- μ - strength of self-propulsion
- λ_l - strength of cortical tension.

We chose the parameter combination where the simulation velocity distribution matched the empirical velocity distribution, and remained a dense colony (0.34 um/minute). It is important to note that we chose optimal model parameters using the distribution of cell velocities and not the median cell velocity. We ran 24 simulations to mimic the the experimental setup of the *in vitro* characterization. Using the Mann Whitney U test, there was no significant difference in the distribution of cell velocities; p-value threshold of 0.05 and p-value for 24 *in silico* colonies was 0.051. After fitting the model to empirical data of cell morphology and velocity, we could recapitulate the cell morphology and collective cell migration of wildtype stem cell colonies without genetic modulation (Movies S1-S2).

2.10.3 Temporal Knockdown Characterization (Protein Expression vs Time)

We then characterized the time dependent knockdown of CDH1 and ROCK1 expression which were responsible for changes in cell-cell adhesion(J) and the strength of cortical tension (λ_l). CDH1 was knocked down using CRISPRi, and the relative expression was measured for 6 consecutive days. The relative mRNA expression of CDH1 was quantified by quantitative PCR and protein expression of CDH1 was measured by immuno-fluorescence microscopy, which displayed a 24hour delay from the mRNA knockdown. The data was min-max normalized to a domain of [0,1] using the median expression for each day:

$$y(t_k)' = (y(t_k) - y_{min})/(y_{max} - y_{min}), \tag{2.13}$$

where t was the time since the knockdown, y_t was the expression at t days after knockdown, y_{max} was the max expression from all days, and y_{min} was the minimum expression over all days. The K_m (repression coefficient) and n (Hill coefficient) of the Hill Function for repression were fit to the normalized median expression to create a response function using least squares optimization:

$$F(t) = 1/(1 + (K_m/t_k)^n), \tag{2.14}$$

where K_m was the time half expression occurs, and n was the hill coefficient. Given value for the parameters k_m and n (Table S1) we now had a continuous response function for the expression of CDH1 given a knockdown time that we could modulate.

To characterize ROCK1 expression, we characterized a 6 day CRISPRi knockdown of the gene expression using quantitative PCR. We assumed mRNA expression changed 24 hours ahead of protein expression, so we shifted the time axis forward by one day to account for the delay. The median expression for each day was min-max normalized to a domain of [0,1] (Equation S13). The K_m(repression coefficient) and n (Hill coefficient) of the hill function for repression were fit to the normalized median expression to create a response function using least squares optimization (Equation S14, Fig. 2,Table S1).

2.10.4 Spatial Pattern Characterization (Protein Expression vs Space)

Given the previous characterization experiments, we were able to model wildtype cell division, cell area, cell migration, and cortical tension. However, the time-dependent modulation of protein express had to be mapped to changes in cell behavior. To model the the time-dependent modulation of cell-cell adhesion via CDH1 and cortical tension via ROCK1, fluorescent microscopy images were collected 96 hours after mixing either ROCK1 KD or CDH1 KD cells with wildtype hiPSCs. To observe a robust spatial phenotype (Libby et al., 2018b), the ROCK1 KD cells were pretreated with DOX to induce KD for 4 days prior to mixing with wildtype, and the CDH1 KD cells were induced with DOX upon mixing. Then *in silico*parameter sweeps were run overlaying a range of adhesion energy/membrane length values that rescaled the normalized Hill Function, and produced a range of spatial patterns due to progressively weaker cell-cell adhesion or progressively stiffer cell membrane parameter values. We then conducted blind experiments to fix adhesion strength and membrane stiffness parameters which most closely matched *in vitro* spatial patterning for CDH1 and ROCK1 knockdowns respectively (Fig. 2H-I).

$$F'(t) = y_{knockdown} + (y_{wildtype} - y_{knockdown})/(1 + (K_m/t)^n), \tag{2.15}$$

where $y_{knockdown}$ was the adhesion strength (J) or target membrane length (L_k, t) of knockdown cell lines in the model, and $y_{wildtype}$ was the adhesion strength (J) or target membrane length of wildtype cells in the model. In Equation 15, we scaled the normalized response function from [0,1] to the range of model parameters $[y_{knockdown}, y_{wildtype}]$.

2.11 *IN VITRO* MATERIALS AND METHODS

2.11.1 Human iPSC culture:

All human induced pluripotent stem cells (hiPSCs) were seeded at a density of 12,000 cells per cm^2 in feeder-free media conditions on growth factor-reduced matrigel (BD Biosciences) and daily fed MTeSR^{TM} medium (STEMCELL Technologies)(Ludwig et al., 2006). When hiPSC confluency reached 75 percent, cells were dissociated and singularized using Accutase (STEMCELL Technologies). Single cells were counted using a Invitrogen Countess Automated Cell Counter (Thermofisher Scientific), re-plated at previously described density, and in the first 24hrs after passaging, fed with MTeSR medium supplemented with the small molecule Rho-associated coiled-coil kinase (ROCK) inhibitor Y-276932 (10M; Selleckchem) to promote survival (Park et al., 2015; Watanabe et al., 2007).

CRISPRi knockdown lines were previously generated as described in (Mandegar et al., 2016), where 20 base pair guides were designed using the Broad Institute sgRNA design website (Doench et al., 2016). 20 base pair sequences were cloned into the gRNA-CNKB vector using restriction enzyme BsmBI digestions, followed by ligation with T4 DNA ligase as described in (Mandegar et al., 2016). 200,000 cells of the CRISPRi-Gen1C or CRISPRi-Gen2 hiPSC lines from the Conklin Lab were nucleofected with individual gRNA vectors using the Human Stem Cell Nucleofector Kit 1 solution with the Amaxa nucleofector 2b device (Lonza). Cells were then plated at increasing dilutions into 3 wells of a 6-well plate coated with growth factor-reduced matrigel (BD Biosciences) in MTeSR^{TM} supplemented with Y-276932 (10M) for 2 days. Then the nucleofected hiPSCs were treated with blasticidin (10g/ml) for a selection period of 7 days. Surviving colonies for each gRNA were pooled and passaged in MTeSR$^{\text{TM}}$ with blasticidin (10g/ml) and Y-27632 (10M) for a single day then transitioned to MTeSR$^{\text{TM}}$ media only. After stable polyclonal populations of hiPSCs were established for each gRNA, cells were karyotyped by Cell Line Genetics (Libby et al., 2018b)(Fig. 2.14). Finally, knockdown efficiency was tested by the addition of doxycycline (2M) to the culture media for 6 days and subsequent qPCR of mRNA levels of respective genes compared to time matched controls of the same line without CRISPRi induction.

2.11.2 Mixed Colony Generation:

Mixed population hiPSC colonies were generated using forced aggregation via PDMS microwells in a 24-well tissue culture plate (975 400X400m wells per well) (Hookway et al., 2016; Libby et al., 2018b). hiPSCs were dissociated and singularized using Accutase (STEMCELL Technologies) and subsequently counted using a Invitrogen Countess Automated Cell Counter (Thermofisher Scientific). The proper ratios of cells to create 100 cell aggregates were then seeded into PDMS wells in MTeSR$^{\text{TM}}$ with Y-27632 (10M), centrifuged at 200g for 5 minutes, and allowed to compact overnight (18h). Aggregates were then washed out of the PDMS wells with fresh MTeSR$^{\text{TM}}$ and re-plated into a growth factor reduced matrigel (BD Biosciences) coated PerkinElmer CellCarrier$^{\text{TM}}$-96 plates at 10/aggregates/cm^2 and fed daily with MTeSR$^{\text{TM}}$.

2.11.3 Immunofluorescence Staining and Imaging:

Human iPSCs were fixed for 25 minutes with 4% paraformaldehyde (VWR) and subsequently washed 3 times with PBS. Fixed colonies were simultaneously blocked and permeablized with a 1X PBS solution with 0.3% Triton X-100 (Sigma Aldrich) and 5%Normal Donkey Serum (Jackson Immunoresearch) for 1 hour at room temperature. Samples were then incubated with primary antibodies overnight at 4°C in a 1X PBS solution with 1% Bovine serum albumin (Sigma Aldrich) and 0.3% Triton-X. Samples were washed 3 times and then incubated for 1 hour at room temperature with secondary antibodies. Primary antibodies used were: anti-OCT4 (SantaCruz 1:400), anti-SOX2 (AbCAM 1:400), and anti-Ecadherin (AbCAM 1:200). All secondary antibodies were used at 1:1000 and purchased from Life Technologies.

Mixed colonies were imaged using a Ziess Observer.Z1 (Ziess) and an InCell Analyzer2000 (GE Healthcare), and confocal images were obtained using a Zeiss LSM880 Confocal w/ Airyscan (Ziess) microscope. Images were analyzed in ImageJ and in python using the skimage package (Walt et al., 2014).

2.11.4 Protein Quantification:

Protein quantification for CDH1 KD was first quantified by immunofluorescence imaging of mixed colonies of WT-GFP hiPSCs and CDH1 KD colonies (Libby et al., 2018b). Total fluorescence of CDH1 was measured by a python script that compared fluorescence of the CDH1 channel normalized to the amount of WT cells vs KD cells (determined by GFP fluorescence)(Fig. 2.7). This data was supplemented by Western blot data from the previously published KD of CDH1 and ROCK1 in (Libby et al., 2018b).

2.11.5 mRNA quantification:

The relative gene expression following CRISPRi knockdown was previously reported in (Libby et al., 2018b) and used as a reference to establish knockdown timing curves used in our *in silico* simulations. As previously reported (Libby et al., 2018b), total mRNA isolation from dissociated hiPSCs was performed using an RNeasy Mini Kit (QIAGEN) according to manufacturer's instructions and quantified with a Nanodrop 2000c Spectrometer (ThermoFisher). Obtained mRNA was then used to synthesize cDNA using an iScript cDNA Synthesis kit (BIORAD). A StepOnePlus Real-Time PCR system (Applied Biosciences) was used to quantify and detect gene expression by Fast SYBR Green Master Mix (ThermoFisher Scientific). Relative gene expression was determined by normalizing comparative threshold(C_t) values to the house keeping gene 18S rRNA. Gene expression was then displayed as a fold change comparison to the day 0 control before the start of gene knockdown. The NCBI Primer-BLAST website was used to design the primers. Statistical analysis was conducted using a two-tailed unpaired t-test between any two groups ($p<0.05$, $n=3$).

2.11.6 Time Lapse Imaging:

Mixed hiPSC colonies were imaged on optically clear PerkinElmer CellCarrier^{TM}-96 plates on an inverted AxioObserver Z1 (Ziess) with an ORCA-Flash4.0 digital CMOS camera (Hamamatsu). Using ZenPro software, colony locations were mapped and a single colony was imaged every 30 minutes over the course of 12 hours. Time lapse imaging occurred from hours 24-36 and from hours 96-208 after mixed colony plate down. The 12 hour series of images were then used to compare *in silico* to *in vitro* pattern formation and organization of cells. Additionally, mixed colonies of wildtype and CRISPRi-Gen1C without knockdown guides were imaged for 6 hours every 5 minutes from hours 60-66 after plate down. These 6 hour image series were used to generate velocity values as previously described in section 2.2 (Velocity Characterization).

2.11.7 BMP4 differentiations

Successfully patterned hiPSC colonies were differentiated for 48h in RPMI 1640 (Life Technologies) cell culture media supplemented with BMP4 (R&D Systems) at a 50 uM/ml concentration. The colonies were then fixed with 4% PFA for 25 min and subsequently analyzed.

2.12 COMPARISON OF *IN VITRO* AND *IN SILICO* SPATIAL PATTERNS

We used *in vitro* and *in silico* images to calculate the total number of cells in an image, the number of clusters, and the circularity of each cluster (Fig. 2.13). Our work-flow for comparing patterns (Fig. 2.15) involves splitting the images into single color RGB channels using the python module scikit-image (Walt et al., 2014; Jones et al., 2001–).

For *in silico* images each channel represents a different cell type. After splitting the image into color channels we detect the number of islands in a colony. For *in silico* images cell are separated by a black border so we dilate the image, remove small objects, then remove small holes in the image with scikit-image. Contiguous regions are considered clusters. We the overlay a mask of individual cells onto each cluster using a logical AND comparison of the image masks to determine if the cell cluster meets our criteria to be considered an island. Using only the cell clusters we consider islands, we then calculate the number of cells per island and the circularity of the islands using the formula circularity = 4πArea/Perimeter^2

In contrast to the work-flow for simulation images, for *in vitro* images one channel represents all cell nuclei and the other channel represents cells stained for the protein CDH1, which delineates the CDH1 knockdown cells from the WT or ROCK1 knockdown cells. For the *in vitro* images, the CDH1 channel was thresholded and then dilated to create a CDH1+ cell mask followed by removal of small objects and holes to create a smooth segmentation. To generate the island masks, isolated CDH1 negative clusters were identified using the label function on the inverse of the CDH1+ mask. Individual cells were localized by detecting local maximum intensity in the DAPI channel images then the number of DAPI peaks per island were calculated using the logical AND of the island and CDH1 negative masks. Finally, we used the function "regionprops" to calculate the cluster area and perimeter for each island, which were then employed to calculate the island circularity with the above formula.

2.12.1 Software

The mathematical model was implemented in Morpheus, v1.9.1, RRID:SCR_014975 (Starruß et al., 2014). Model fitting of single-cell morphology, cell velocities, temporal knockdown characterizations, and spatial pattern characterizations were performed with custom Python code (modules: scipy, numpy, matplotlib, pandas, seaborn, scikit-image)(Jones et al., 2001–). Image preprocessing, segmentation, and quantification of cell and colony morphology was performed with custom Python code (modules: numpy, scipy, scikit-image). Pattern classifications and specification mining was performed using the data mining software WEKA. Quantification of pattern similarity and pattern optimization were performed with TSSL (MATLAB + Java) which can be accessed from:

http://sites.bu.edu/hyness/software/

2.12.2 Supplementary Figures

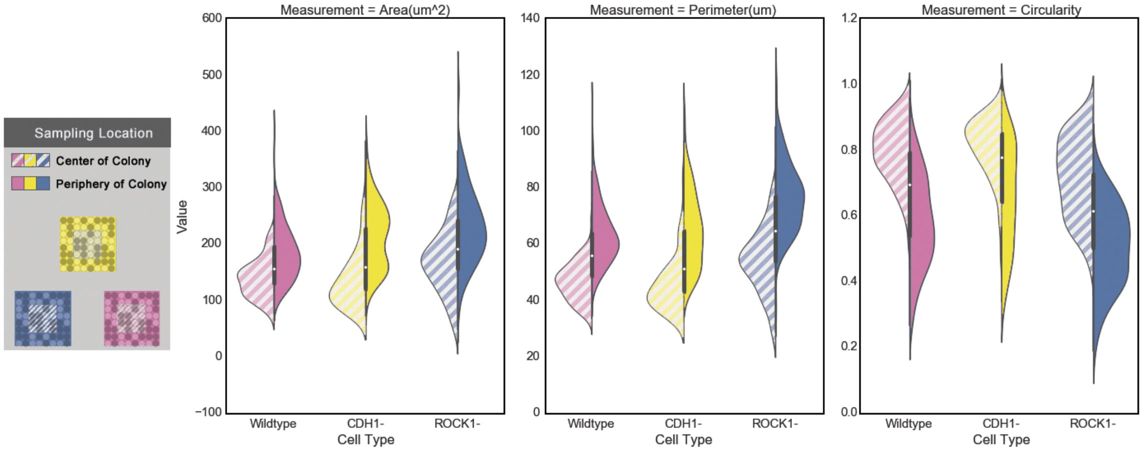


Figure 2.6: Violin Plots of single cell morphology (cell area, cell perimeter, cell circularity) for wildtype hPSC, hPSC with CDH1 knocked down, and hPSC with ROCK1 knocked down in relation to the cells position in the colony. Cells have distinct distributions for both morphological measurements. The area and perimeter measurements with respect to each cell line were used to fit our computational model.

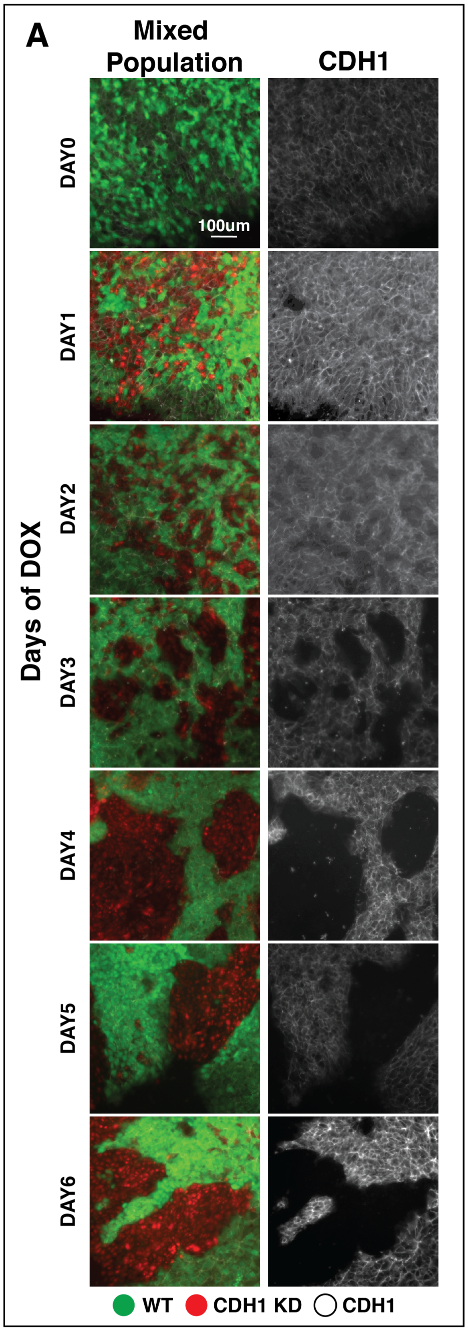


Figure 2.7: Knockdown of CDH1 quantified by immunofluorescence imaging (A) Mixed populations of wildtype (GFP+) and CDH1 KD cells (GFP-) were imaged daily and fluorescence of CDH1 levels was normalized to GFP fluorescence.

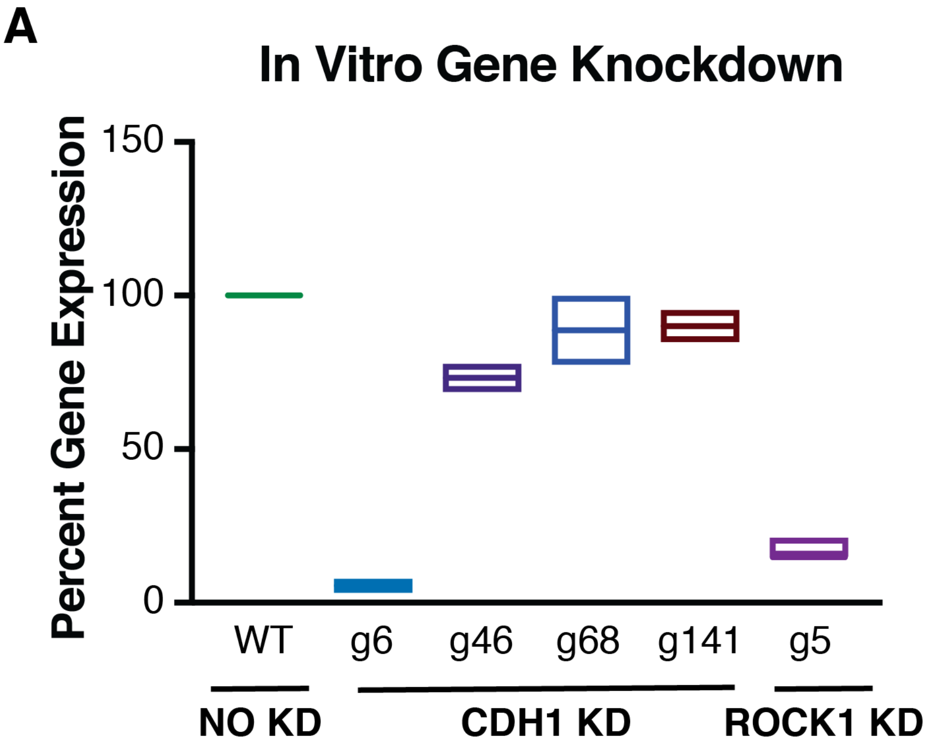


Figure 2.8: CRISPRi cell line mRNA knockdowns. (A) Percent knockdown of either CDH1 or ROCK1 when compared to wildtype control populations on day six of DOX treatment (n=3)

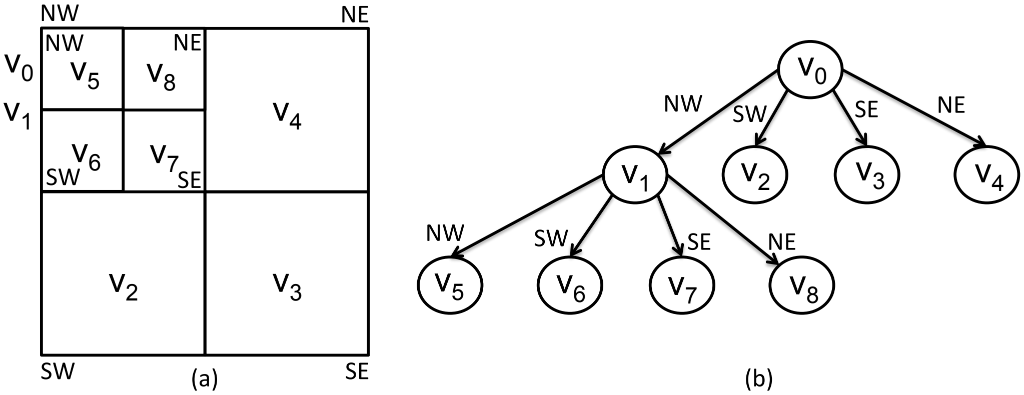


Figure 2.9: Quad-tree representation (b) of a matrix (a).

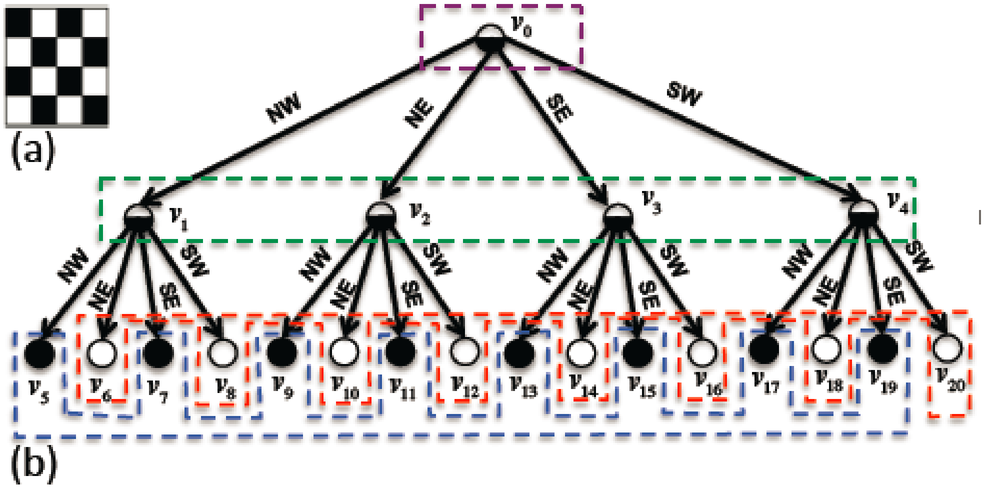


Figure 2.10: Quad-tree representation (b) of a 4 by 4 checkerboard pattern (a).

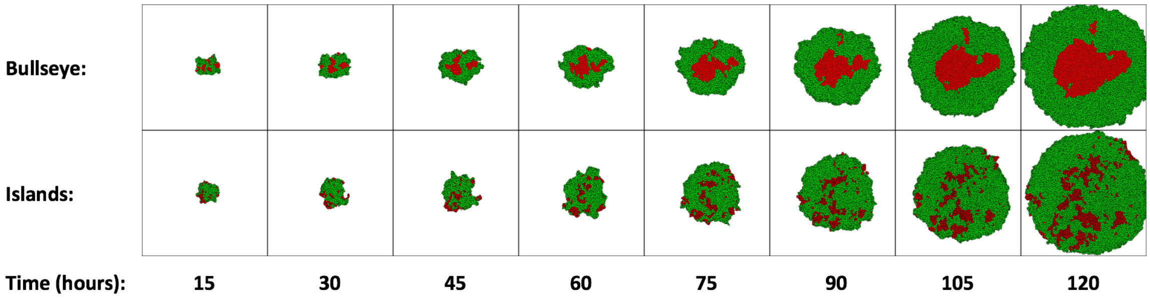


Figure 2.11: Sample simulations derived from the parameters presented in Fig. 2.4 to produce Bullseye and Multi-island patterns.

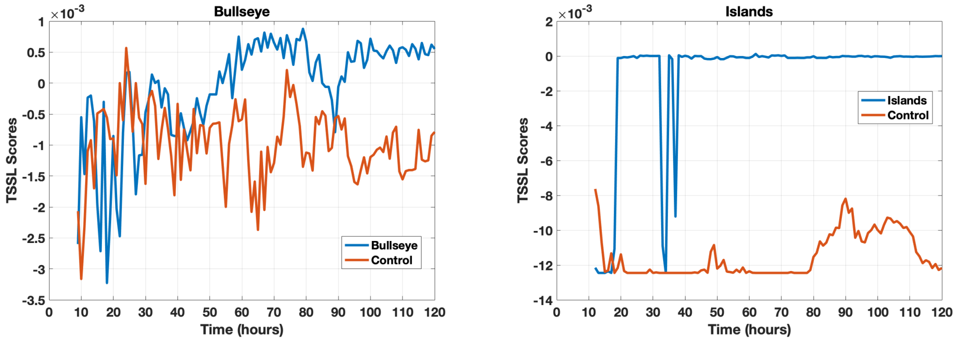


Figure 2.12: Evolution of the TSSL robustness score over time for the sample simulations illustrated in Fig. 2.11; producing the Bullseye pattern (left) and Multi-islands pattern (right).

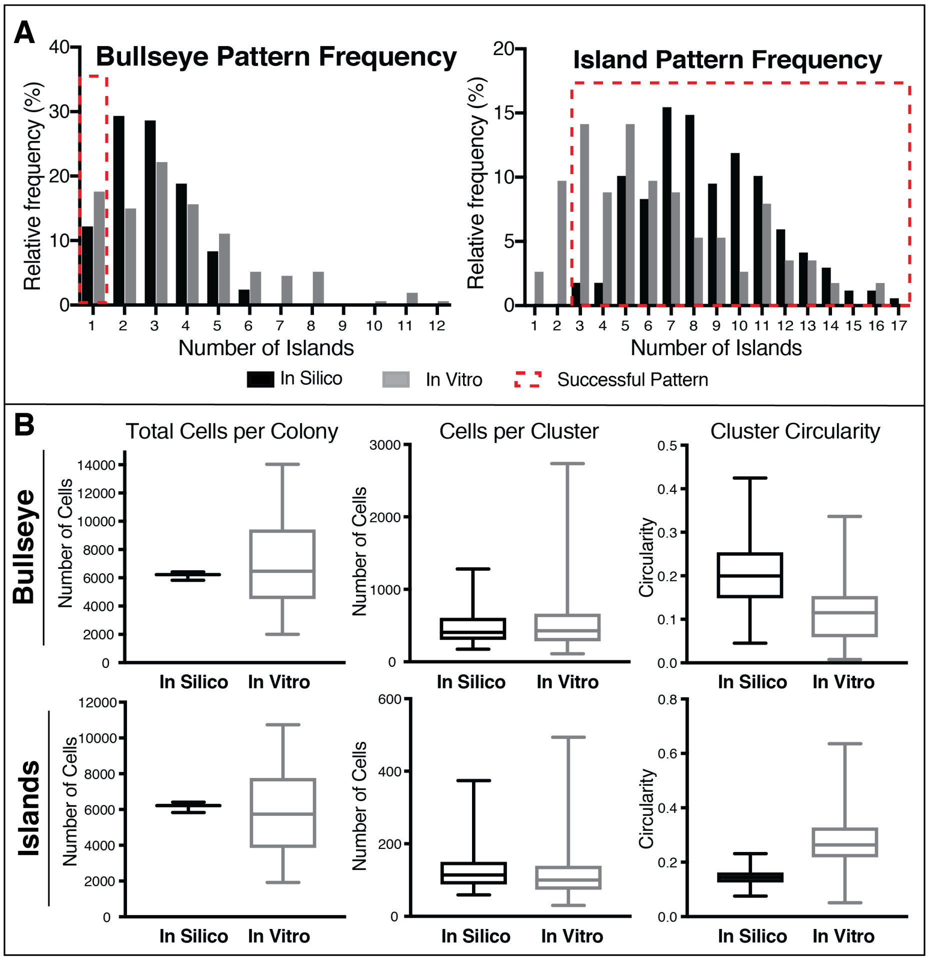
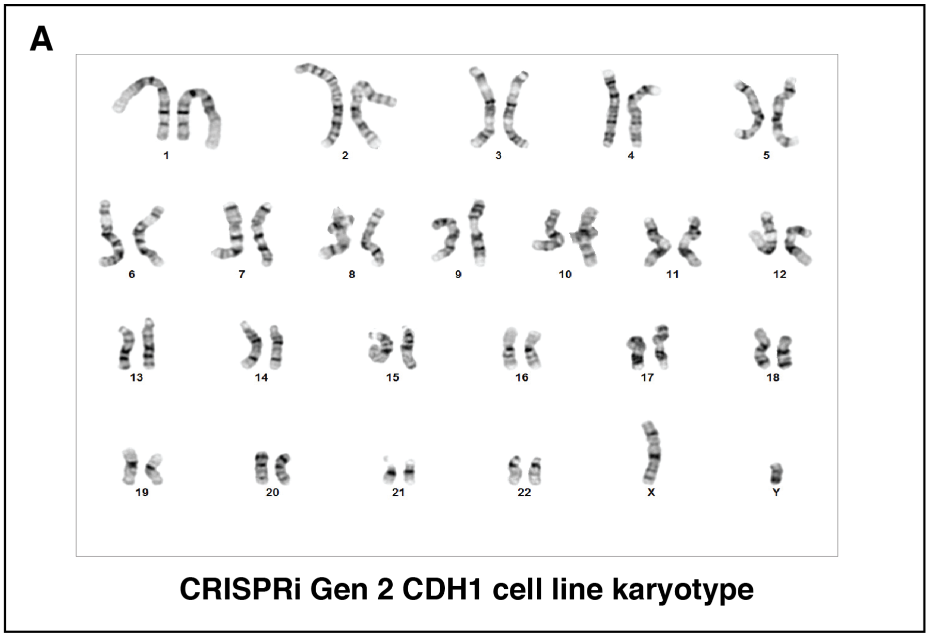


Figure 2.13: In silico vs. in vitro metrics (A) Distribution of islands, where for the Multi-island patterns, an island is defined as a homotypic cluster of 25 or more cells, and for the Bullseye patterns, an island is defined as 50 or more cells. Successful Bullseye patterns will display one island, successful Multi-island patterns will display 3 or more islands. (Bullseyes n=148 colonies; Islands n=79) (B) Distribution of in silico and in vitro patterned colonies with regard to total cells per colony, number of cells per cluster, and cluster circularity. In Bullseye patterns clusters consist of ROCK1 knockdown cells. In Multi-island patterns clusters consist of CDH1 knockdown cells.



CRISPRi Gen 2 CDH1 cell line karyotype

Figure 2.14: Karyotype of CRISPRi Gen2 CDH1 hiPS Cell line (A) the Gen2 CDH1 hiPSC line was karyotypically normal.

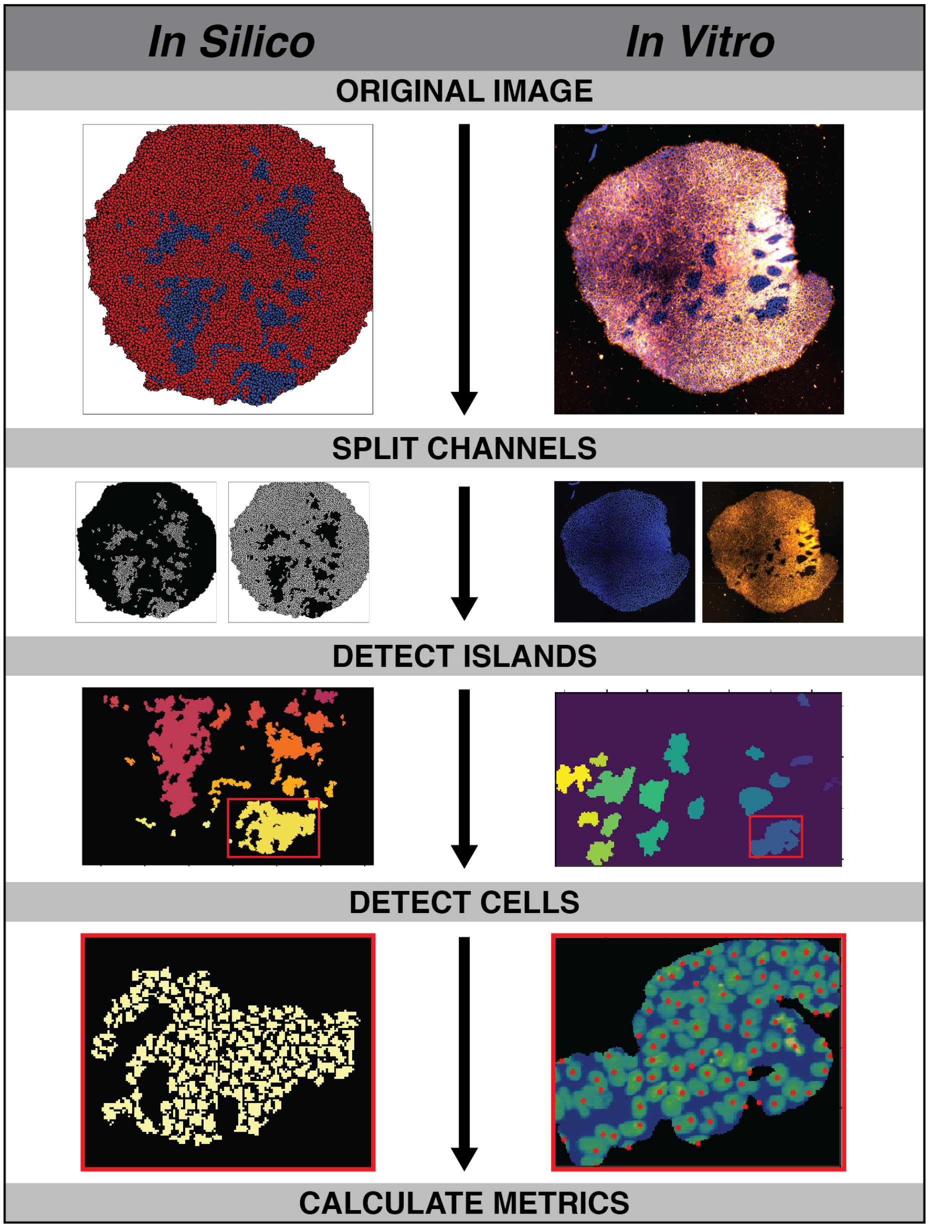
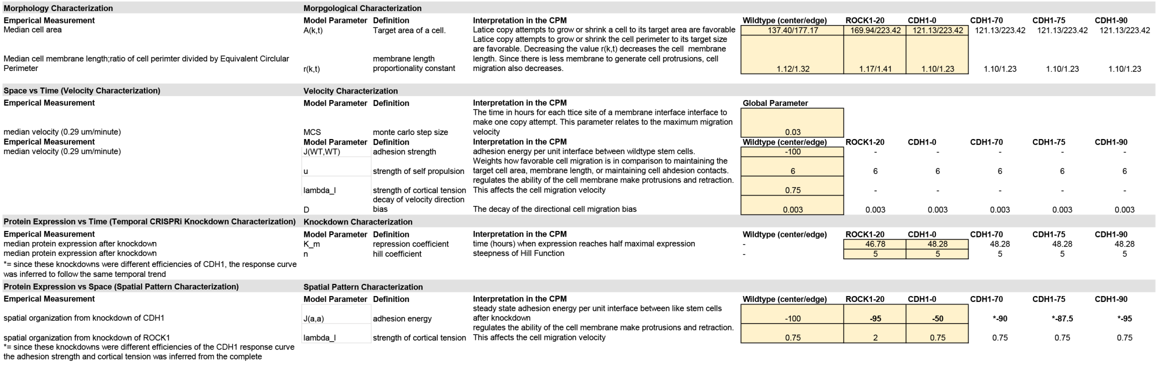


Figure 2.15: In silico vs. in vitro image segmentation work flow to quantify and compare spatial patterns



Morphology Characterization	**Morpgological Characterization**								
Emperical Measurement	**Model Parameter**	**Definition**	**Interpretation in the CPM**	**Wildtype (center/edge)**	**ROCK1-20**	**CDH1-0**	**CDH1-70**	**CDH1-75**	**CDH1-90**
Median cell area	A(k,t)	Target area of a cell.	Latice copy attempts to grow or shrink a cell to its target area are favorable	137.40/177.17	169.94/223.42	121.13/223.42	121.13/223.42	121.13/223.42	121.13/223.42
Median cell membrane length;ratio of cell perimter divided by Equivalent Circlular Perimeter	r(k,t)	membrane length proportionality constant	Latice copy attempts to grow or shrink the cell perimeter to its target size are favorable. Decreasing the value r(k,t) decreases the cell membrane length. Since there is less membrane to generate cell protrusions, cell migration also decreases.	1.12/1.32	1.17/1.41	1.10/1.23	1.10/1.23	1.10/1.23	1.10/1.23
Space vs Time (Velocity Characterization)	**Velocity Characterization**								
Emperical Measurement	**Model Parameter**	**Definition**	**Interpretation in the CPM**	**Global Parameter**					
median velocity (0.29 um/minute)	MCS	monte carlo step size	The time in hours for each ttice site of a membrane interface interface to make one copy attempt. This parameter relates to the maximum migration velocity	0.03					
Emperical Measurement	**Model Parameter**	**Definition**	**Interpretation in the CPM**	**Wildtype (center/edge)**	**ROCK1-20**	**CDH1-0**	**CDH1-70**	**CDH1-75**	**CDH1-90**
median velocity (0.29 um/minute)	J(WT,WT)	adhesion strength	adhesion energy per unit interface between wildtype stem cells.	-100	-	-	-	-	-
	u	strength of self propulsion	Weights how favorable cell migration is in comparison to maintaining the target cell area, membrane length, or maintaining cell ahdesion contacts.	6	6	6	6	6	6
	lambda_l	strength of cortical tension	regulates the ability of the cell membrane make protrusions and retraction. This affects the cell migration velocity	0.75	-	-	-	-	-
	D	decay of velocity direction bias	The decay of the directional cell migration bias	0.003	0.003	0.003	0.003	0.003	0.003
Protein Expression vs Time (Temporal CRISPRi Knockdown Characterization)	**Knockdown Characterization**								
Emperical Measurement	**Model Parameter**	**Definition**	**Interpretation in the CPM**	**Wildtype (center/edge)**	**ROCK1-20**	**CDH1-0**	**CDH1-70**	**CDH1-75**	**CDH1-90**
median protein expression after knockdown	K_m	repression coefficient	time (hours) when expression reaches half maximal expression	-	46.78	48.28	48.28	48.28	48.28
median protein expression after knockdown	n	hill coefficient	steepness of Hill Function	-	5	5	5	5	5
*= since these knockdowns were different efficiencies of CDH1, the response curve was inferred to follow the same temporal trend									
Protein Expression vs Space (Spatial Pattern Characterization)	**Spatial Pattern Characterization**								
Emperical Measurement	**Model Parameter**	**Definition**	**Interpretation in the CPM**	**Wildtype (center/edge)**	**ROCK1-20**	**CDH1-0**	**CDH1-70**	**CDH1-75**	**CDH1-90**
spatial organization from knockdown of CDH1	J(a,a)	adhesion energy	steady state adhesion energy per unit interface between like stem cells after knockdown	-100	**-95**	**-50**	***-90**	***-87.5**	***-95**
spatial organization from knockdown of ROCK1	lambda_l	strength of cortical tension	regulates the ability of the cell membrane make protrusions and retraction. This affects the cell migration velocity	0.75	2	0.75	0.75	0.75	0.75
*= since these knockdowns were different efficiencies of the CDH1 response curve the adhesion strength and cortical tension was inferred from the complete									

Figure 2.16: Modeling Fitting Parameters for stem cell multicellular patterning

CHAPTER 3
Pattern Synthesis in 3D Stem Cell Aggregates

3.1 INTRODUCTION

There has been an increasing effort in recent years to reproduce the microenvironment of self-renewing biological processes such as embryogenesis (White et al., 2015; Kinney et al., 2014), tissue development (Darabi et al., 2008), and organoid formation (Shkumatov et al., 2014) by forcing stem cells to aggregate into three dimensional spheroids known as embryoid bodies (EBs). Similar to the self-renewing stem cells residing in its *in vivo* niche, EBs undergo patterned differentiation controlled by local cues (Morrison & Spradling, 2008; Scadden, 2006).

Instead of modeling this patterned differentiation at a global level, the authors of (White et al., 2013, 2015) developed an agent-based model that captures proliferation and spatio-temporal patterning of EB differentiation using local interaction rules. They also developed quantitative network metrics to verify their agent-based model captures a distribution of patterns observed experimentally. These pattern classes describe differentiation occurring from the outside-in, the inside-out, in globular (spheroid) subclusters, or in snake-like subclusters.

Even though White et al. (2015) was able to accurately recapitulate the distribution of pattern trajectories in EBs, their exhaustive search of parameter combinations becomes computationally expensive as the number of parameters increases. Therefore, we employ an alternate approach that quantifies how strongly the spatial behavior of a dynamical system resembles a global pattern. Then, we use this quantifier as the basis for an optimization procedure to synthesize model parameters.

There has been an increasing effort in recent years to use formal logics as descriptors of spatial properties. Linear spatial superposition logic (LSSL) was successfully used in Grosu et al. (2009) to identify self-similar texture. Richer spatial and spatio-temporal logics were introduced in Bartocci et al. (2016b); Haghighi et al. (2015) with the capability of describing complex patterns in networked or distributed dynamical systems.

This paper is closely related to Bartocci et al. (2016b); Haghighi et al. (2015). In Bartocci et al. (2016b), tree spatial superposition logic (TSSL) is defined and used to enforce the emergence of steady state Turing patterns in a biochemical reaction-diffusion system. In Haghighi et al. (2015), spatial temporal logic (SpaTeL) was introduced to formally express time-varying spatial patterns. Both these logics are equipped with quantitative semantics, which can be used as a robustness score for how a dynamical system satisfies desired specifications. Using these formal quantification tools, in this paper, we develop a heuristic optimization procedure in which we run simulations with interactions at the local level, make quantitative spatio-temporal observations at the global level, and select parameter values with the highest robustness score. We show that simulations produced by parameter values resulting from this framework closely resemble the predefined patterns observed experimentally.

In Sec. 3.2, we describe an agent-based model characterizing state changes in embryoid bodies. This model is a modified version of the model introduced in White et al. (2013). Although our model uses the same rules to enforce stem cell differentiation, we utilize more efficient algorithms for growing cell populations, identifying neighbors, and visualizing global behaviors. These modifications decrease the simulation run time and increase the pattern classification accuracy. In

Sec. 3.3, the formal methods approach used to synthesize model parameters is presented. Sec. 3.4 presents our final conclusions and a few directions for future work.

3.2 STOCHASTIC AGENT-BASED MODEL

The purpose of performing parameter optimization using an agent-based model of 3D stem cell differentiation is to explore the range of patterns that emerge from local interaction rules. This includes synthesizing experimentally observed patterns or even *de novo* patterns that were not observed experimentally. This enables biologists to better understand the local mechanisms governing morphogenesis (e.g. paracrine signaling) and efficiently test assumptions about these mechanisms.

In this paper we assume local signaling regulates stem cell differentiation in EBs. We also assume differentiation can be described as the binary classification that captures if a cell has lost the ability to specialize into any cell type (*loss of pluripotency*)Livigni & Brickman (2013). Differentiation can describe any transition down the hierarchy of stem cell specialization Morrison & Spradling (2008), but this binary classification, Fig. **??** and Fig. **??**, allows us to investigate early cell fate decisions that likely influence tissue and organ development. Using the *loss of pluripotency* indicator, the protein Oct4, White et al. (2013) has shown that there is a time-dependent distribution of intermediate patterns (Fig. **??**) before complete differentiation (Fig. **??**) in EBs.

The proposed stochastic agent-based model is a modified version of the approach described in White et al. (2013, 2015), where it is assumed that stem cell differentiation is dictated by basal stochasticity, local negative feed forward signaling, and positive feedback signaling. Our modeling approach uses the same set of local differentiation rules. However, two main modifications were made. First, we use a more efficient approach (KD-trees) to identify a cell's neighborhood. Second, EBs are analyzed as transparent cross-sectional images. This approach is more in line with images that are produced experimentally, such as Fig. 3.1, where cells within a certain distance from cross-sections of EBs are observable. This visualization approach allows for more accurate pattern classification in images, as demonstrated in Sec. 3.3.

The agent-based model of stem cell differentiation used in this paper is described in detail in the rest of this section.

3.2.1 Description of Stem Cell Agents

Consider a network of $N(t)$ locally interacting stem cells in which each cell is labeled by an integer $i \in \{1, \cdots, N(t)\}$, where $N(t)$ is the number of stem cells at time $t \in \{0, \cdots, T\}$ and T is the earliest time at which all cells have differentiated. We represent this network of stem cells as a graph $G(t) = (V(t), E(t))$, where the vertex $i \in V(t)$ represents the ith cell at time t and $(i, j) \in E(t)$ if the corresponding cells i and j are interacting neighbors at time t.

Asynchronous Cell Division: A *primary* cell k

$\in \{1, \cdots, N(0)\}$, which is present at the start of a simulation, initially divides at a random time $t_1^k \in [0, \delta]$, where δ is the length of the cell cycle for a single cell. The set of time points that a primary cell $k \in \{1, \cdots, N(0)\}$ or its daughters divide is denoted by $D_k = \{t_1^k, t_2^k, \cdots, t_M^k\}$, where

$$\begin{aligned} 0 \leq t_m^k \leq T \quad \forall m \in \{1, \cdots, M\} \\ t_{m+1}^k - t_m^k = \delta \quad \forall m \in \{1, \cdots, M-1\}, \end{aligned} \tag{3.1}$$

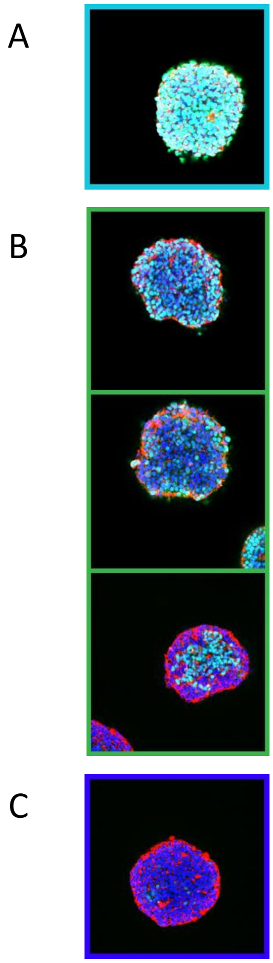


Figure 3.1: Confocal microscopy multichannel images of EBs stained with DAPI (nuclear stain, blue), phalloidin (red), and Oct4 (cyan) shown at a depth of 25 um for EBs. (a) Early stem cell populations are primarily pluripotent and uniformly express Oct4 (cyan). (b) Over multiple days, a distribution of differentiation patterns are observed before every cell is differentiated into an Oct4-negative status. These patterns are labeled as inside-out (left),globular (center),and outside-in(right). (c) After 7 days most stem cells have differentiated into specialized cell lineages (no Oct4, loss of cyan).

where m is the mth division, and $M$is the final division of a cell during a simulation. D_k is interpreted as the set of asynchronous division times for the initial population of stem cells. Since experiments have shown that undifferentiated stem cells divide in approximately 19 hours White et al. (2013), we assume that $\delta = 19$ in embryonic stem cells.

The set of new cells introduced to the system at time t as a result of cell division is denoted by

$$NEW(t) = \{N(t-1)+1, \cdots, N(t)\}. \tag{3.2}$$

After each division, one daughter cell replaces the parent cell with the same label and the other daughter cell is placed adjacently in a random direction. We define a mapping $\mathcal{P} : NEW(t) \to V(t-1)$ where $\mathcal{P}(i)$ is the parent cell from which $i \in NEW(t)$ divided.

Cell Neighborhood: Each cell is assumed to be a sphere, with the center located at $l_i(t) \in \mathbb{R}^3$ for $i \in \{1, \cdots, N(t)\}$ and constant radius r. Cell radius is assumed to be constant since changes in the size of a cell over time are negligible White et al. (2013). We also define the set of all cell locations at t as

$$L(t) = \{l_1(t), l_2(t), \cdots, l_{N(t)}(t)\}. \tag{3.3}$$

The Euclidean distance between two cells is denoted by $d(l_i(t), l_j(t))$. Two cells i,j are considered neighbors if they are in contact with one another. Therefore, we construct the set of graph edges $E(t)$ such that

$$(i,j) \in E(t) \subseteq V(t) \times V(t) \iff d(l_i(t), l_j(t)) \leq 2r.$$

The neighborhood of cell i is denoted by $e_i(t)$.

$$e_i(t) = \{j \in \{1, \cdots, N(t)\} \mid (i,j) \in E(t)\}. \tag{3.4}$$

Looping through every pair of vertices in $V(t)$ to construct $E(t)$ is exponential in $|V(t)|$. Instead we use a space partitioning tree, KD-tree, to find neighboring cells. A KD-Tree is a special case of the binary space partitioning tree, which is used to partition points in a k-dimensional space, find nearest neighbors, and perform range queries Bentley (1975).

3.2.2 Structural Constraints in 3D

EB's are an *in vitro* system used to study stem cell differentiation in 3D Cha et al. (2015); Kinney et al. (2014). To accurately capture the mechanical forces of an EB, a mass-spring model is employed. Like the model in White et al. (2013), 1000 cells is the size of the initial population. The radius, circularity, average connection length, average connection number, and density of the simulated EB's show these simulations resemble *in vitro* EBs White et al. (2013). This resulting structure represents the set of primary cells at the initial time ($t = 0$).

Collisions and Interactions: Cells are represented as rigid spheres connected by springs. The rigid spheres model the incompressibility of the cell nucleus, and the springs represent the malleability of the cytoplasm. We enforce rigid sphere behavior for every pair of interacting cells.

$$d(l_i(t), l_j(t)) \geq 2r, \quad \forall (i,j) \in E(t). \tag{3.5}$$

Equation (3.5) can be violated as a result of population growth. When cell i divides, a new cell j is randomly placed in the network such that $d(l_i(t), l_j(t)) = 2r$. This can cause cell j to overlap with some of the cells surrounding i. If the constraint in Equation (3.5) is violated, then distance corrections F_{ij} for the pair of cells (i, j) are determined using the equation:

$$F_{ij} = -\mathcal{X} k_c, \tag{3.6}$$

where k_c is a constant that represents the strength of the interaction, and $\mathcal{X}$ represents the amount of displacement between the cytoplasm of two cells ($\mathcal{X} = d(l_i(t), l_j(t)) - 2r$). This interaction system is solved iteratively until all pairs of cells satisfy Equation 3.5.

3.2.3 Stochastic Rules for State Changes

We chose to model stem cell differentiation as a 3-state process with differentiation controlled by cell division Johnston et al. (2007). The state of cell i at time $t \in \{0, \cdots, T\}$ is denoted by $x_i(t) \in \{\mathcal{U}, \mathcal{T}, \mathcal{D}\}$, where $x_i(t) = \mathcal{U}$ if cell i is undifferentiated, $x_i(t) = \mathcal{T}$ if cell i is transitioning, and $x_i(t) = \mathcal{D}$ if a cell i is differentiated at time t.

At $t = 0$, every cell is initialized in state $\mathcal{U}$. At each time step of the simulation, three *stochastic rules* regulate the probability that a cell will differentiate. Biologically, these stochastic rules represent basal stochasticity (random rule), the influence of undifferentiated neighbors to cause a cell to change state (negative feed forward), and the influence of differentiated neighbors to cause a cell to change state (positive feedback). These rules are derived in White et al. (2013, 2015) and were motivated by Sun & Komarova (2012).

Rule 1 *(Basal Differentiation) The probability that a cell i changes its state from $\mathcal{U}$ to*

$\mathcal{T}$, regardless of the state of its neighbors, is given by:

$$p_1^i = \alpha, \tag{3.7}$$

where $\alpha \in (0,1)$ is a constant.

Rule 2 *(Local Negative Feedforward) The probability that undifferentiated neighboring cells cause a cell i to change its state from $\mathcal{U}$ to $\mathcal{T}$ is given by:*

$$p_2^i = \frac{1}{1 + (U_{norm,i}/k_1)^{n_1}}, \tag{3.8}$$

where k_1 and n_1 are tuning parameters and $U_{norm,i}$ is defined as the percentage of neighboring cells of cell i that are undifferentiated or transitioning.

Rule 3 *(Local Positive Feedback) The probability that differentiated neighboring cells cause a cell i to change its state from $\mathcal{U}$ to $\mathcal{T}$ is given by:*

$$p_3^i = \frac{D_{norm,i}^{n_2}}{k_2^{n_2} + D_{norm,i}^{n_2}}, \tag{3.9}$$

where k_2 and n_2 are tuning parameters and $D_{norm,i}$ is defined as the percentage of neighboring cells of cell i that are differentiated.

Three independent random variables $\{r_1^i, r_2^i, r_3^i\}$ uniformly distributed between 0 and 1 are assigned to each cell i, $i \in \{1, \cdots, N(t)\}$. For a cell i in state $x_i(t) = \mathcal{U}$, the state of the cell is set to $\mathcal{T}$ until the next cell division if

$$r_1^i < p_1^i \vee r_2^i < p_2^i \vee r_3^i < p_3^i \tag{3.10}$$

evaluates to true. A cell at state $x_i(t) = \mathcal{T}$ will change its state to $\mathcal{D}$ when the next cell division occurs. This mimics the notion that cell division cycles control stem cell differentiation Johnston et al. (2007). It important to realize that Equations 3.8 and 3.9 resemble Hill equations, and their respective tuning parameters influence how many neighboring cells in a specific state will cause a cell to differentiate. For all cells that are differentiated ($x_i = \mathcal{D}$), the division time δ is approximately 51 hours White et al. (2013).

3.2.4 Visualization

The model described above was implemented in Python as a simulation tool that receives a list of parameters

$$\Pi = (\alpha, k_1, n_1, k_2, n_2) \in \Omega, \tag{3.11}$$

where

$$\Omega = \omega_\alpha \times \omega_{k_1} \times \omega_{n_1} \times \omega_{k_2} \times \omega_{n_2}, \tag{3.12}$$

and $\omega_j \subseteq \mathbb{R}$ is an allowed interval for parameter j. For a given set of parameter values Π, the simulator produces a sequence of images $Y_t(\Pi)$, where $t \in \{0, 1, \cdots, T\}$ denotes the time point in hours. Fig. 3.2 illustrates the output produced by a simulation. The simulator produces red spheres for cells in undifferentiated or transitioning state ($x_i(t) \neq \mathcal{D}$) and black spheres for differentiated cells ($x_i(t) = \mathcal{D}$).

Although cells were located in a 3 dimensional space, we visualized differentiation at each time point t as a transparent cross-sectional image. Transparent 2D images allowed us to accurately classify patterns in images without the computational complexity required to represent 3D images. In cross-sectional experimental images (Fig. 3.1), undifferentiated cells appear cyan and differentiated cells

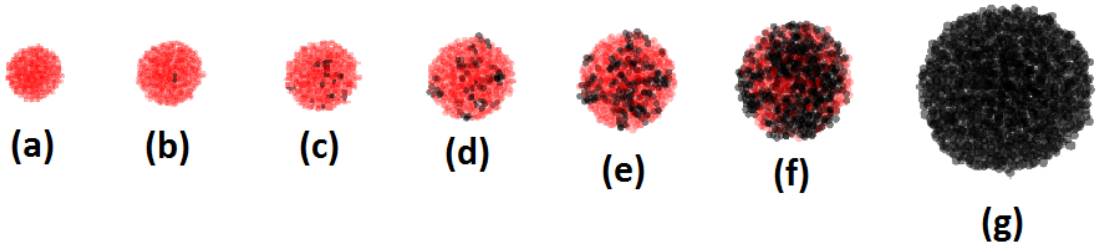


Figure 3.2: Sample images from a simulation for the parameter set $\Pi = (0.01, 0.1, 25, 0.3, 25)$ at time (a) $t = 0$ (b) $t = 5$ (c) $t = 10$ (d) $t = 15$ (e) $t = 20$ (f) $t = 30$ (g) $t = 60$. Cells are colored red when in the undifferentiated state, and black upon differentiation.

appear blue. The choice of red and black instead of cyan and blue was made in order to create a more significant contrast between the RGB concentrations of differentiated and undifferentiated cells, making it easier to learn pattern classifiers.

3.3 OPTIMIZING 3D SPATIO-TEMPORAL PATTERNS

In this section, we explore the parameter set $\Pi \in \Omega$, Eq. (3.11), in the agent-based model of Sec. 3.2 with the purpose of determining valuations such that the simulations of the model produce images that best resemble specific predetermined patterns. Specifically, we are interested in inside-out, outside-in, and globular patterns similar to Fig. **??**.

Since *in silico* simulations of the model are very time-consuming, (approximately one hour on average on a machine with a 2.4GHz core i7 CPU and 8 GB RAM), it is impractical to explore the parameter space with a brute-force approach. Instead, we implement the formal methods approach presented in Bartocci et al. (2016b) and Haghighi et al. (2015) to solve this problem. In this approach, desirable patterns are formally specified by logical formulas. These formulas are then used to analyze the behavior of a dynamical system and synthesize desired spatio-

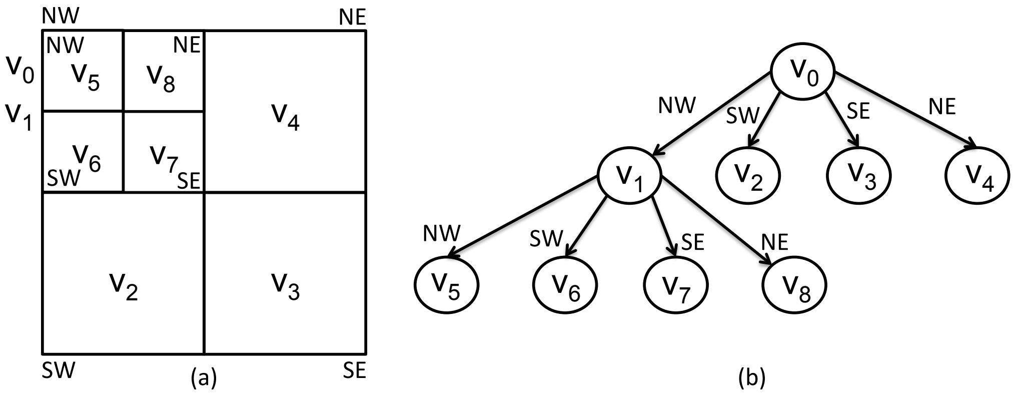


Figure 3.3: Quad-tree representation (b) of a matrix (a)

temporal behaviors.

3.3.1 Quad-tree Representation of Spatial Behaviors

Consider an RGB representation of an $m \times n$ image as the matrix A where the element $a_{ij} = \langle a_{ij}^{(r)}, a_{ij}^{(g)}, a_{ij}^{(b)} \rangle$ is the normalized RGB values for the pixel located on the ith row and jth column of the image. Thus,

$$0 \leq a_{ij}^{(c)} \leq 1 \text{ for } c \in \{r, g, b\}.$$

Given a matrix A, $A[i_s, i_e; j_s, j_e]$ is used to denote the submatrix created by selecting rows with indices from i_s to i_e and columns from j_s to j_e. Following Bartocci et al. (2016b), we represent the matrix A as a quad-tree defined as follows, A quad-tree $Q = (\mathcal{V}, R)$ is a quaternary tree Finkel & Bentley (1974) representation of matrix A where each vertex $v \in \mathcal{V}$ represents a submatrix of A and the relation $R \subset \mathcal{V} \times \mathcal{V}$ defines four children for each vertex that is not a leaf.

Fig. 3.3 demonstrates how a quad-tree is built from a matrix. In this figure, we label each edge in the quad-tree with the direction of the sub-matrix represented by the child: north west ($NW$), north east ($NE$), south west ($SW$), and south east (SE). In Fig. 3.3(b), v_0 represents the complete matrix A. v_1 represents $A[1,\lfloor m/2\rfloor;1,\lfloor n/2\rfloor]$, where m is the total number of rows and n is the total number of columns in A. v_5 represents $A[1,\lfloor m/4\rfloor;1,\lfloor n/4\rfloor]$, etc. The construction of a quad-tree in this paper slightly differs from Bartocci et al. (2016b). In Bartocci et al. (2016b), the assumption is made that A has a size of $2^k \times 2^k$ so that each submatrix can be divided into four equal-sized partitions. Here, we have relaxed this requirement by allowing non-equal submatrices to be children of a node. Furthermore, Bartocci et al. (2016b) defines a leaf as a vertex of the quad-tree for which all the elements of a submatrix have the same values. While this approach works perfectly for the 32×32 network that is studied in that paper, it can be problematic for larger images since the number of vertices in a quad-tree grows exponentially as more levels are added to it. In this paper, we construct quad-trees with a fixed depth d, regardless of the size and other characteristics of A. A *representation function* $\mu^{(c)}(v) : \mathcal{V} \to [0,b] \times [0,b]$ for sub-matrix $A[i_s,i_e;j_s,j_e]$ represented by vertex $v \in \mathcal{V}$ of the quad-tree $Q = (\mathcal{V},R)$ is a function that provides the mean value and variance for the concentration of RGB colors in a particular region of the space represented by the vertex v. where $c \in \{r,g,b\}$ is an RGB color. The function $\mu^{(c)}$ provides the mean value and variance for the concentration of RGB colors in a particular region of the space represented by the vertex v.

We use quad-trees to analyze spatial patterning in images. In the rest of this section, we assume that any given image derived from the simulator of Sec. 3.2 ($Y_t(\Pi)$) is translated into a corresponding quad-tree ($Q_t(\Pi)$).

3.3.2 Formal Specification of Global Patterns

In Bartocci et al. (2016b), the logic, called tree spatial superposition logic (TSSL), introduced that is capable of formally specifying patterns in a network of locally interacting agents. They show that this logic is sophisticated enough to describe complicated patterns such as Turing patterns in biochemical reaction-diffusion systems. We use this logic to express inside-out, outside-in, and globular patterns in the model of Sec. 3.2. First, we present a brief introduction to TSSL. Refer to Bartocci et al. (2016b) for a thorough explanation of this logic, definitions of syntax and semantics, and its properties. A TSSL formula is recursively constructed using the following:

- Linear predicates over valuations for the representation function. e.g., $\mu_1^{(r)} > \lambda_1$, $\mu_1^{(b)} < \lambda_2$.
- Boolean operators: e.g., $\neg\phi$, $\phi_1 \wedge \phi_2$, and $\phi_1 \vee \phi_2$.
- Spatial operators. e.g., $\exists_B \bigcirc \phi$, $\forall_B \bigcirc \phi$, where B is a nonempty subset of the set of directions $\{NW, NE, SW, SE\}$.

The spatial operators $\exists_B \bigcirc$ and $\forall_B \bigcirc$ are read as *there exists in directions B next* and *for all directions B next*, respectively. $\exists_B \bigcirc \phi$ is interpreted as follows: For at least one of the nodes located in the next level of the quad-tree labeled with one of the directions in B, ϕ must be satisfied. $\forall_B \bigcirc \phi$ specifies that for all such nodes ϕ must be satisfied.

TSSL formulas can be viewed as pattern classifiers. Although TSSL is capable of describing complicated spatial behaviors, it is difficult in general to write a formula that describes a complex pattern. In Bartocci et al. (2016b), the authors

Pattern	Positive learning images	Negative learning images	Positive testing images	Negative testing images
Outside-in	6000	22000	2000	6000
Inside-out	6000	22000	2000	6000
Globular	8000	20000	2000	6000

Table 3.1: Number of samples generated to learn and test TSSL classifiers

propose to use machine learning techniques in order to find such a formula from a given set of positive and negative examples.

Assume a set of positive images ($\mathcal{Y}_+$), illustrating a desirable pattern, and a set of negative images ($\mathcal{Y}_-$) are available. We can create set $\mathcal{L}$ from these images as

$$\mathcal{L} = \{(Q_y, +) \mid y \in \mathcal{Y}_+\} \cup \{(Q_y, -) \mid y \in \mathcal{Y}_-\},$$

where Q_y is the quad-tree generated from image y. The set $\mathcal{L}$ is separated into a learning set $\mathcal{L}_L$ (used to train a classifier) and a testing set $\mathcal{L}_T$ (used to test the classifier obtained from $\mathcal{L}_L$) such that $\mathcal{L} = \mathcal{L}_L \cup \mathcal{L}_T$. A rules-based learner called RIPPER Cohen (1995) is used learn a set of classification rules from $\mathcal{L}_L$ and these rules are transformed into a TSSL formula Bartocci et al. (2016b).

We applied this framework to obtain classifiers for three classes of patterns: inside-out, outside-in, and globular. First, a tool was developed that creates learning and testing sets for these patterns by generating random images corresponding to different patterns as well as images without any particular behavior. The sizes of the learning and testing sets for each of the patterns are presented in Table 3.1.

Three learning sets $\mathcal{L}_L^{\text{inside-out}}$, $\mathcal{L}_L^{\text{outside-in}}$, and $\mathcal{L}_L^{\text{globular}}$ were created by constructing quad-trees with a depth $d = 5$ from these images. Next, RIPPER was employed to learn a TSSL classifier for each pattern: $\Phi_{\text{inside-out}}$, $\Phi_{\text{outside-in}}$, and Φ_{globular}. The algorithm terminated in 42 minutes for $\Phi_{\text{outside-in}}$, 74 minutes for $\Phi_{\text{inside-out}}$, and 161 minutes for Φ_{globular}, using an iMac with 2.8 GHz Intel core i5 CPU and 32 GB RAM.

The classification rate for the testing sets in Table 3.1 are 99% for outside-in. 98% for inside-out, and 96% for the globular pattern. $\Phi_{\text{inside-out}}$, $\Phi_{\text{outside-in}}$, and Φ_{globular} will be used later to automatically check whether an image generated by the simulator of Sec. 3.2 demonstrates one of the three specified patterns.

3.3.3 Quantification

A formal recursive definition for the qualitative semantics of TSSL is presented in Bartocci et al. (2016b). These semantics can be used to assign a true (satisfied) or false (violated) label to a TSSL specification with respect to an unlabeled quad-tree. To provide information about how strongly an image satisfies or violates the given property, TSSL is also equipped with a recursive quantitative semantics definition which assigns a real value to a TSSL formula ϕ with respect to vertex $v \in \mathcal{V}$ of quad-tree $Q = (\mathcal{V}, R)$; denoted by $\rho(\phi, v)$. It is proven in Bartocci et al. (2016b) that TSSL quantitative semantics are sound. In other words, a quad-tree Q satisfies a formula ϕ ($Q \models \phi$) if $\rho(\phi, v_0) > 0$ where v_0 is the root of Q, and Q violates ϕ ($Q \not\models \phi$) if $\rho(\phi, v_0) < 0$. Therefore, the problem of checking whether an image contains a pattern expressed as a TSSL formula reduces to computing its quantitative valuation. Moreover, the absolute value of $\rho(\phi, v_0)$ can be viewed as a measure of how strongly ϕ is satisfied (or violated) by Q. Hence, the quantitative valuation of a formula with respect to a quad-tree is called its *robustness*.

TSSL quantitative valuation is an effective tool to quantify emergence of a pattern in an individual image. However, notice that the simulator of Sec. 3.2 produces a time sequence of images, and patterns can emerge at any time point in general. Consequently, we need to quantify the emergence of global behaviors in time-varying quad-trees. We use a temporal extension of TSSL, called spatial temporal logic (SpaTeL) for this purpose. SpaTeL was first introduced in Haghighi et al. (2015) to study time-varying spatial patterns in networked systems. Refer to Haghighi et al. (2015) for a complete explanation and formal definitions for SpaTeL. In this paper, we focus on SpaTeL formulas that are used to synthesize patterns at an unknown time. A SpaTeL formula is formed by nesting TSSL formulas inside temporal operators. Consider a time sequence of quad-trees Q_t, where $t \in \{0, 1, \cdots, T\}$, with the following specification: the TSSL formula Φ_{pattern} must eventually be satisfied within the interval $t \in \{t_1, \cdots, t_2\}$. This specification can be formalized by the following SpaTeL formula:

$$\Psi_{\text{pattern}} = F_{[t_1, t_2)} \phi_{\text{pattern}}. \tag{3.13}$$

SpaTeL is also equipped with quantitative semantics. The quantitative valuation for Ψ_{pattern} with respect to the sequence of quad-trees Q_t is denoted by $\rho_t(\Psi_{\text{pattern}}, v_0(t))$.

$$\rho_t(\Psi_{\text{pattern}}, v_0(t)) = \max_{t \in \{t_1, \cdots, t_2\}} \rho(\phi_{\text{pattern}}, v_0(t)), \tag{3.14}$$

where $v_0(t)$ is the root of Q_t.

It is proven that SpaTeL quantitative semantics are sound (has a positive valuation for satisfying sequences of quad-trees and a negative valuation for violating ones) Haghighi et al. (2015). Similar to TSSL, SpaTeL's quantitative valuation (robustness) is shown to be a good measure of how strongly a time varying quad-tree (a sequence of images) satisfies or violates a spatio-temporal specification Haghighi et al. (2015). Note that in Equation 3.13 $\rho_t(\Psi_{\text{pattern}}, v_0(t)) > 0$ if the TSSL formula ϕ_{pattern} is satisfied by at least one of the quad-trees in $\{Q_t \mid t_1 \leq t < t_2\}$. Therefore, we can use SpaTeL's quantitative valuation for Ψ_{pattern} to quantify the emergence of a pattern (e.g., outside-in, inside-out, or globular) in a sequence of images simulated from the model of Sec. 3.2.

3.3.4 Parameter Synthesis

At this point, we are able to quantify how strongly a sample trace $Q_t(\Pi)$ from the simulator of Sec. 3.2 satisfies or violates emergence of a pattern specified by the SpaTeL formula (3.13). This metric can serve as the fitness function in an optimization process over the parameter space from Equation (3.12). The goal is to determine the parameterization Π_{pattern} that maximizes this metric:

$$\Pi_{\text{pattern}} = \arg\max_{\Pi \in \Omega} \rho_t(\Psi_{\text{pattern}}, v_0(t)) \tag{3.15}$$

Many optimization methods can be used to solve (3.15). Inspired by Bartocci et al. (2016b) and Haghighi et al. (2015), we employed particle swarm optimization (PSO) Kennedy (2010). PSO is a heuristic solution to unconstrained optimization problems that is capable of solving problems with irregular search spaces and does not require the fitness function to be differentiable.

This procedure was performed for the three patterns inside-out, outside-in, and globular. Inspired by earlier analysis of similar models in White et al. (2013) and White et al. (2015), we chose to fix the values for α, n_1, and n_2 at 0.005, 25,

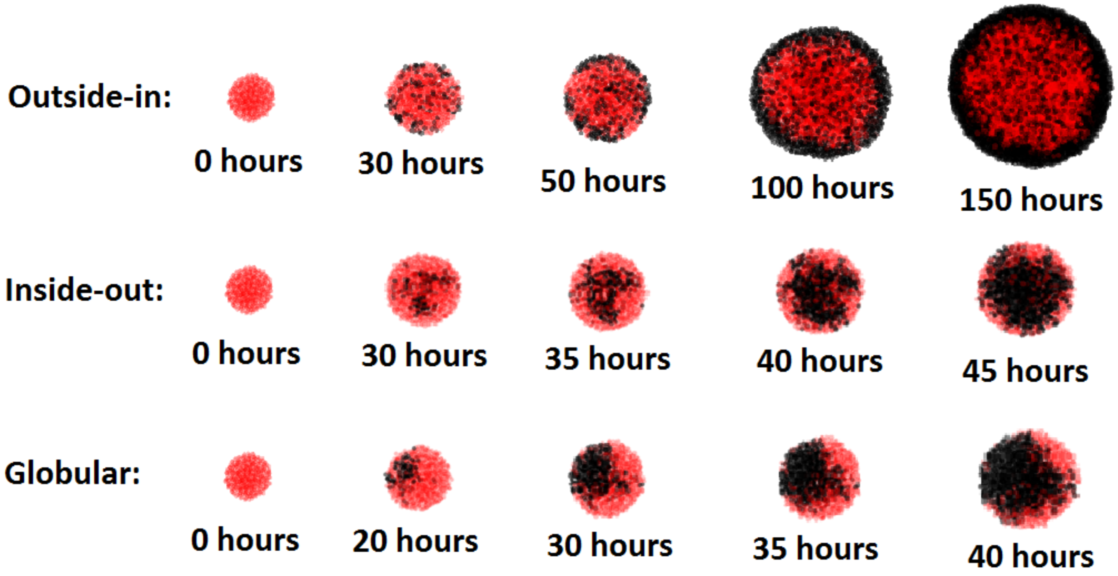


Figure 3.4: Sample images of simulations derived from the optimized parameter values: $(k_1, k_2) = (0.08, 0.92)$ for outside-in, $(k_1, k_2) = (0.33, 0.09)$ for inside-out, and $(k_1, k_2) = (0.50, 0.11)$ for globular.

and 25, respectively. The PSO search algorithm was performed on parameters $(k_1, k_2) \in [0, 1] \times [0, 1]$. The optimized parameter values are presented in Fig. 3.4. The computation was distributed on a cluster with 8 processors at 2.1GHz and the running time was about 30 hours for the outside-in pattern, 11 hours for the inside-out pattern, and 16 hours for globular. Sample simulations derived from these parameters are demonstrated in Fig. 3.4. As illustrated in this figure, the simulated traces resulting from these parameters qualitatively resemble the patterns of Fig. **??** at the transitioning stage.

3.4 FUTURE WORK

Directions for future work include exploring how context-independent parameters, such as cell division rates, affect the emergence of patterns. We also plan to study stem cell aggregate shapes other than spheres (e.g., cones, rods). More complex and biologically interesting patterns will be studied by incorporating diffusion among locally interacting cells in the model. Also, a more effective fitness function for the optimization procedure may be developed by combining TSSL/SpaTeL robustness with other metrics such as White et al. (2013, 2015). Finally, we will define oct-trees as 3D extensions of quad-trees and use them to learn formal classifiers for patterns that are only observable in a 3D space.

CHAPTER 4
Agent-based Model for Self-Assembled 3D Multicellular Shapes

4.1 INTRODUCTION

Multicellular pattern formation is an emergent behavior that drives complex behaviors in mammalian cell systems (such as embryonic development). Currently, there are several exploratory approaches to induce multicellular patterning in mammalian cells using differential adhesion (Toda et al., 2018; Fot, 2013), micro-patterned surfaces (Warmflash et al., 2014; Molitoris et al., 2016), biochemical sender-receiver systems, or 3D bioprinting. However, inducing these behaviors requires significant manual intervention from the experimenter or requires the assistance of artificial scaffolds. Approaches that require manual design or manual intervention are difficult to scale up and produce more complex patterning. Scaffold-based and micro-patterned systems fix the pattern into a substrate an do not allow pattern to change over time as is often seen during embryonic development. Further, most of these approaches are driven by trial-and-errors. To overcome limitations of trail-and-error driven design I developed a computational model to predict and quantify the self-assembly into a desired 3D structures with out manual intervention.

4.2 COMPUTATIONAL MODEL OF AUTONOMOUS 3D SELF-ASSEMBLY

Note: all mathematical formulas and symbols in this chapter have no relation to mathematical symbols in previous chapters unless explicitly stated.

With the objective of enabling the programmatic self-assembly of cells into user-defined 3D structures, an agent-based model of the self-assembly of geomet-

ric *building blocks* from cells, which in-turn self-assemble into larger 3D structures. While this work is theoretical and has yet to be experimentally validated, I provide references to mechanisms of programmatic self-assembly in this model. This model was implemented using a Python 3 wrapper to the Open Dynamics Engine (physics) implemented by the Panda3D game engine. The code and documentation for running the simulator can be accessed at:
https://github.com/dmarcbriers/

4.2.1 Cell Environment

The environment E of the simulation is three dimensional euclidean space $E \in R^3$ where R is the set of real numbers. As described in Chapter 1, this model belongs to the *center-based* class of agent-based models. Each cell C_i, where i is the cell index, is represented as a rigid spherical body having a position, radius, orientation, and rotation. All cells are in a 3D aqueous solution with external propulsion generated by an Orbital Shaker. Therefore, we assume the collision of cells with water molecules can be approximated by Brownian Motion. There are no scaffolds in the environment to aid in cell migration and organization besides other cells.

4.2.2 Cell Logic

Each cell has genetic circuit logic controlling several cell actuations as listed in Fig 4.1. Given the gene circuits and logic actuators I demonstrate the potential to design and model cells that autonomously controls aspects of proliferating and developing into an organized system without manual intervention or external scaffolding during the development of the structure.

In this system, cells have six actuations that are controlled by two synthetic gene circuits. One circuit allows cells to count the number of cell division cycles it has progressed through and is referred to as a *counter*. The *counter* behaves as an indicator of time and allows cells to optionally change their behavior after each cell cycle. During mammalian development, hematopoietic stem cells remember the number of self-renewal divisions and demonstrate phenotypic changes due to this division history (Bernitz et al., 2016).

The second genetic circuit directly controls the type of actuation is that activate and is known as the *register*. The register is able to read the *counter*, and the register is also able express proteins given the state of the cell and the genes encoded on the circuit in the current cell. It is important to note that when a cell divides, it can copy and re-distribute its gene circuit asymmetrically to its daughter cells which allows rapid symmetry breaking in this system (Chiu & Jiang, 2017).The *register* circuit can activate five classes of morphological actuations (Fig 4.1). These actuations are allowing proliferation, inducing cell cycle arrest (preventing proliferation), inducing apoptosis (causing the cell to be degraded), expressing generic/homotypic cell-cell adhesion proteins (such as cadherins (Toda et al., 2018; Maître & Heisenberg, 2013)), and expressing paired/heterotypic cell-cell adhesion proteins.

4.2.3 The Developmental Tree

The process to programatically control the cell proliferation and the expression the correct proteins to produce a desired 3D object is refereed to as the *developmental tree*. The developmental tree can be represented as a nondeterministic Finite State Machine (FSM). The states of the FSM describe the currently expressed proteins under the control of the synthetic gene circuit. The states of the FSM S_j, where

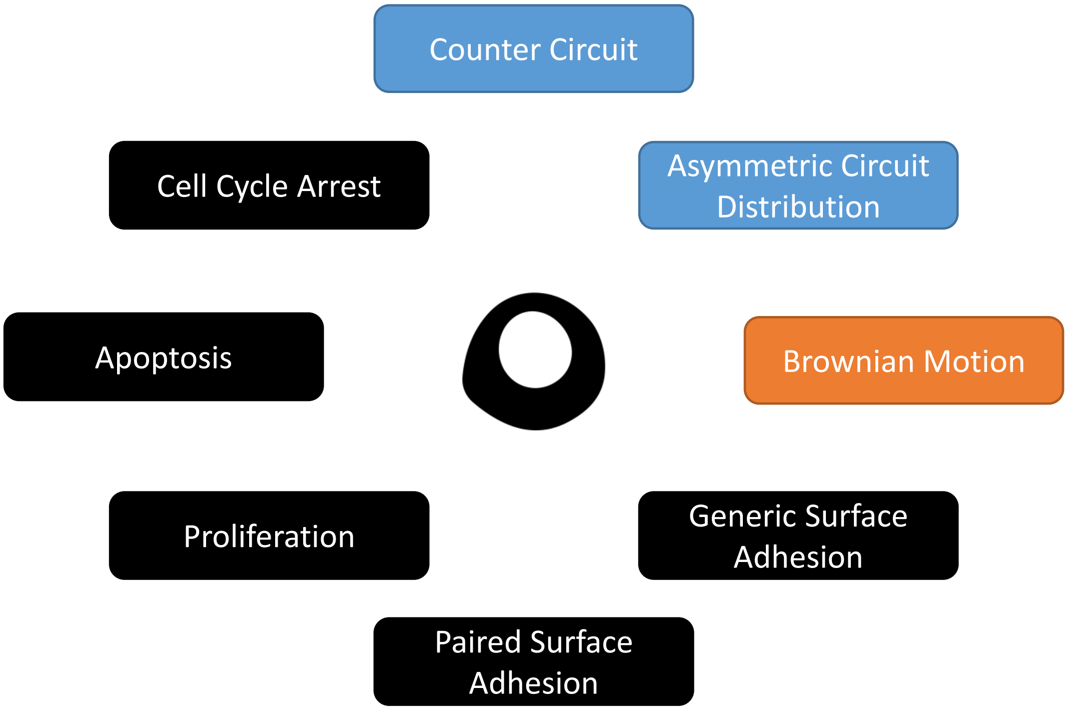


Figure 4.1: Actuators for 3D Self-Assembly. Out of the eight actuations, 6 actuations are morphological actuations, 1 actuator is logical (counter circuit), and 1 actuation (Brownian Motion) is induced by an orbital shaker mixing fluid environment around cells.

$j \in N$ (N is the set of nonnegative integers) map to a set of proteins expressed by the synthetic circuit register of the cell. S_j also represent what cell state its daughter cells can transition to. Therefore, a cell state can be considered a unique node in the developmental tree and not simply a unique set of expressed proteins. This will be clarified in the case studies.

The transitions of the FSM represent how a parent cell produces two daughter cells with respective states S_j induced by recombinases during cell division. Each transition is represented by a mapping from a single state to a 3-tuple represented by the states of the daughter cells and the probability that this transition will occur. For example, if a cell with state 0 produced two daughters cells in states 1 and 2 with a 100 percent probability, this transition will be represented as $S_0 : (S_1, S_2, 1.0)$.

4.2.3.1 Cell-Cell Interactions

Mammalian cells have traditionally been engineered using homotypic cell surface binding mechanisms to induce spatial patterning. In addition to homotypic cell-cell adhesion, we model the ability of cells to express paired surface binding proteins which mimic a lock-and-key mechanism. This allows a more selective cell adhesion behaviors. While paired proteins A1-A2 binds, a cell expressing protein A1 does not adhere to a cell expressing that same protein A1. The format for specifying a homotypic adhesion protein is a letter sequence followed by the number zero (eg A0, B0, Z0). For paired surface binders there is a letter sequence followed by a the number 1 or 2 indicating which type of protein pair is being expressed (A1-A2, B1-B2, Z1-Z2).

In our simulations, we represent cell-cell adhesion by a ball-socket joint. Cell are free to change their rotation and orientation as long as they maintain a fixed distance from cells it shares a join with. If cells do not adhere they will repel each other following the equation for elastic collisions.

4.3 CASE STUDIES: PROGRAMMATIC SELF-ASSEMBLY

We create two demonstrations of self-assembled structures. The smallest building blocks in these case studies is considered to be a tetrahedron made of 4 cells. In the first case study 2 tetrahedrons that do not adhere to to other tetrahedron was self-assembled starting fro a single cell. In Fig 4.2 we show the FSM and the development plan that created 2 independent tetrahedrons. This demonstrate the ability to achieve both genetic and spatial symmetry breaking. The results of this simulation are shown in Fig 4.3.

The second case case study was to design a synthetic circuit that would induce a single cell to self-assemble into a lowercase letter "d". An alphabetical letter was chose because if of asymmetry in the geometric shape which is not observed in tetrahedron. Also, the lowercase letter could misalign in the 3D plane, introducing a degree of failure that was not frequently seen in the tetrahedron examples. In five simulations, the letter d was qualitatively verified in 2 out of hose simulations.

4.4 CONCLUSIONS AND FUTURE WORK

In this chapter, a mathematical model was developed to simulate the major cell logic and cell mechanical interactions necessary to predict the self-assembly of 3D shapes. Our computational model integrates the Brownian motion of cells in a rotating fluid, elastic collisions, selective cell adhesion using single or paired surface binding proteins, and gene circuit-driven logic to direct the self-assembly of

Cell States	Syntetic Proteins Expressed
0	
1,3,4	A
2,5,6	B
7,8,9,10	A, CellCycleArrest
11,12,13,14	B, CellCycleArrest

Cell State	Transition
0	(1,2,1.0)
1	(3,4,1.0)
2	(5,6,1.0)
3	(7,8,1.0)
4	(9,10,1.0)
5	(11,12,1.0)
6	(13,14,1.0)

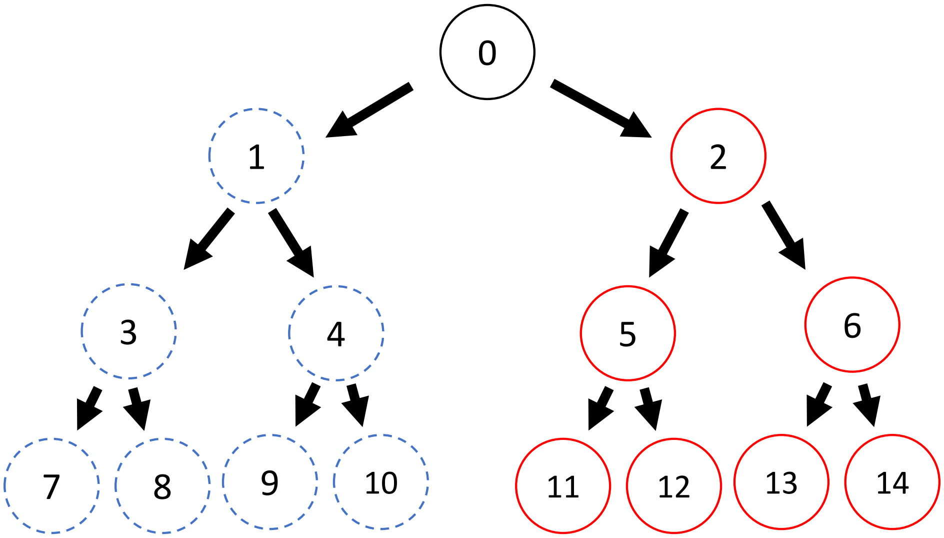


Figure 4.2: Assembly of two independent tetrahedrons. The FSM and the development plan that created 2 independent tetrahedrons. This demonstrate the ability to achieve both genetic and spatial symmetry breaking.

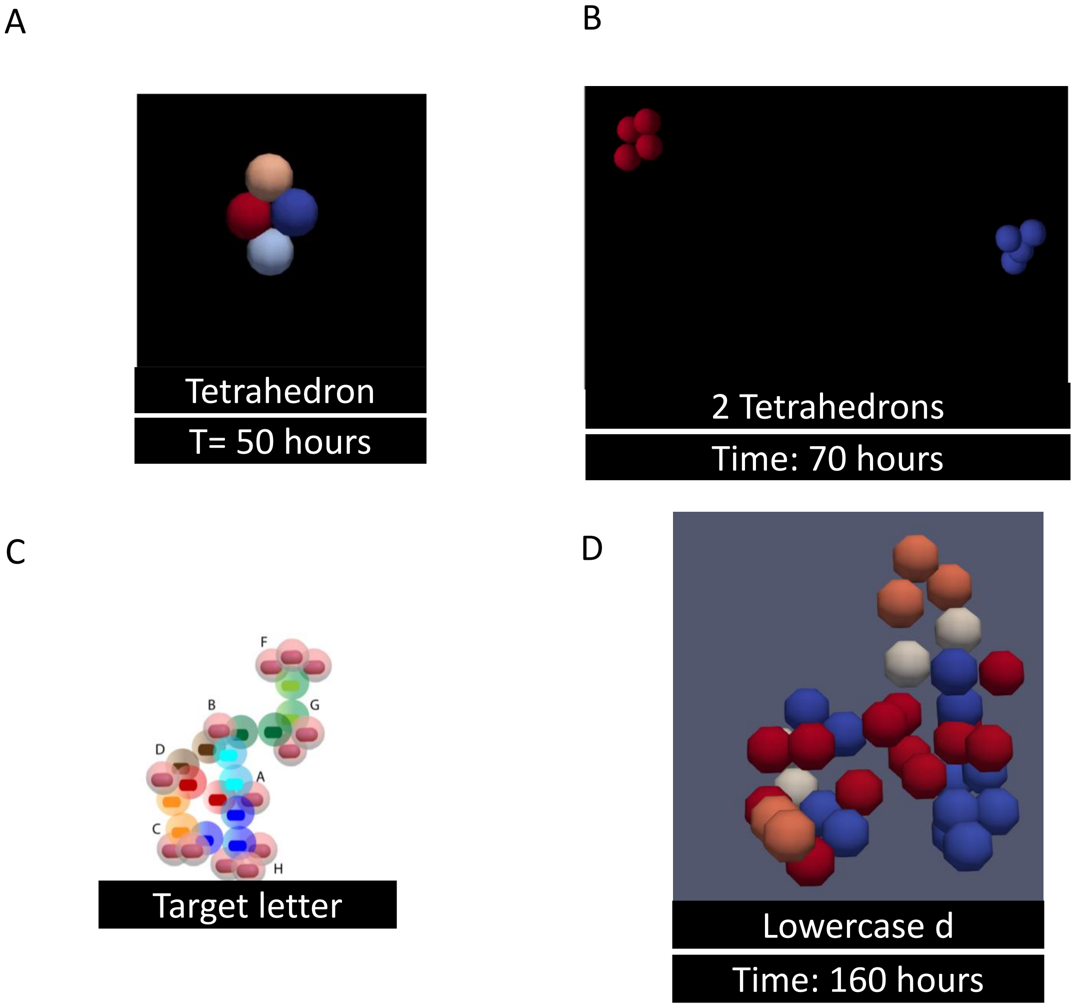


Figure 4.3: In Silico Verification. A) Verification of the self-assembly into a tetrahedron. B) Verification of the self-assembly into 2 independent tetrahedron. C) Proposed design of cell-cell adhesion to construct a lowercase d. D) Verification of the self-assembly into a lowercase letter d.

individual cells into multicellular geometric shapes. It was demonstrated that a single cell can be programmed to proliferate, assemble into building blocks (tetrahedrons), arrest cell division, and ultimately assemble into larger 3D structures. In a coordinated effort we simulated the self-assembly of a tetrahedron, two independent tetrahedron and a lowercase letter d.

Although these shapes are relatively simple, this work necessity of computational modeling to rapidly design and and verify the design of novel living system. In Fig 4.3 these shapes took 50 hours to 160 hours to build. However, the simulations all ran in under 10 minutes. Future work would include quantitative approaches to verify successful shapes and adding the ability of cell to actively migrate given they can adhere to neighboring cells. This would allow the model to predict the rich set of cell sorting patterns that arise from differential adhesion and collective cell migration.

CONCLUSION AND DIRECTIONS

There are several challenges that remain for mathematical and computational modeling to reach its potential in designing pattern formation in networks of mammalian cells. The current limitations include the availability of computationally efficient modeling platforms, simple user-interfaces to specify 2D and 3D patterning, and integrative modeling framework that consider both cellular control and external controls (eg optogenetics, robotic micromanipulation).

While many computational models are available after publication, many of them are not extensible or understandable to researchers and developers outside of the group that published the work. In communities such as metabolic modelings community, there have been several iterations shared between independent researchers due to the development of standardized tools and representations of models. While many journal require the publication of models, this should require the publication and documentation of the code much like the documentation of experimental protocols.

when developing models of patterning in mammalian systems, there are numerous agent-based frameworks that are not easily integrated. While some models focus on cell mechanics, other frameworks are better suited to predict cell migration, diffusion, or the formation of bioelectrical networks. It would be a great aid to the community to see integrative modeling frameworks capturing cell mechanics, migration, gap-junction communication, bioelectrical signaling, and diffusion in a unified yet computationally efficient modeling framework.

Finally, there is a need to consider external control of multicellular patterning in modeling frameworks. In this thesis, I demonstrated computational modelings that are designed around the notion of controlling tunable parameters to predict and design desired spatiotemporal behaviors. However, I did not consider dynamic perturbations to mammalian systems such as micro-manipulation from robotic platforms or optogenetics that can make changes in spatial signaling in the order of second to minutes.

Mathematical and computational modeling is a powerful tool that can be combined with machine learning to greatly improve the development or more complex tissues, and quantify the reliability of engineered mammalian systems.

APPENDIX A

Apendix of Computational Methods

A.1 PATTERN OPTIMIZATION

In order to automatically compare patterns produced by the models described in Chapters 2 and 3 from different parameterizations and determine optimal parameter values, we needed a measure capable of quantifying how close any given pattern was to the desired one. A very effective algorithm was proposed in (Bartocci et al., 2016a) for this purpose.

A.1.1 Quad-Tree Representation of an Image

Consider an RGB representation of an $m \times n$ image as the matrix A where the element $a_{ij} = \langle a_{ij}^{(r)}, a_{ij}^{(g)}, a_{ij}^{(b)} \rangle$ is the normalized RGB values for the pixel located on the ith row and jth column of the image. Thus,

$$0 \leq a_{ij}^{(c)} \leq 1 \text{ for } c \in \{r, g, b\}.$$

Given a matrix A, $A[i_s, i_e; j_s, j_e]$ was used to denote the submatrix created by selecting rows with indices from i_s to i_e and columns from j_s to j_e. Following (Bartocci et al., 2016a; Briers et al., 2016), we represented the matrix A as a quad-tree. A quad-tree $Q = (\mathcal{V}, R)$ is a quaternary tree (Finkel & Bentley, 1974) representation of matrix A where each vertex $v \in \mathcal{V}$ represents a submatrix of A and the relation $R \subset \mathcal{V} \times \mathcal{V}$ defines four children for each vertex that is not a leaf.

Fig. 2.9 demonstrates how a quad-tree is built from a matrix. In this figure, we label each edge in the quad-tree with the direction of the sub-matrix represented by the child: north west (NW), north east (NE), south west (SW), and south east

(SE). In Fig. 2.9(b):

- v_0 represents the complete matrix A.
- v_1 represents $A[1, \lfloor m/2 \rfloor; 1, \lfloor n/2 \rfloor]$, where m is the total number of rows and n is the total number of columns in A.
- v_2 represents $A[\lfloor m/2 \rfloor + 1, m; 1, \lfloor n/2 \rfloor]$.
- v_3 represents $A[\lfloor m/2 \rfloor + 1, m; \lfloor n/2 \rfloor + 1, n]$.
- v_4 represents $A[1, \lfloor m/2 \rfloor; \lfloor n/2 \rfloor + 1, n]$.
- v_5 represents $A[1, \lfloor m/4 \rfloor; 1, \lfloor n/4 \rfloor]$,
- etc.

We used the procedure described in (Briers et al., 2016) to construct quad-trees, which is slightly different from (Bartocci et al., 2016a). In (Bartocci et al., 2016a), the assumption was made that A has a size of $2^k \times 2^k$ so that each submatrix could be divided into four equal-sized partitions. Here, we relaxed this requirement by allowing non-equal submatrices to be children of a node. Furthermore, (Bartocci et al., 2016a) defined a leaf as a vertex of the quad-tree for which all the elements of a submatrix had the same values. While this approach works perfectly for the 32×32 network that is studied in that paper, it can be problematic for larger images since the number of vertices in a quad-tree grows exponentially as more levels are added to it. In this paper, we constructed quad-trees with a fixed depth of 5, regardless of the size and other characteristics of A.

The representation function $\mu^{(c)}(v) : \mathcal{V} \to [0,b] \times [0,b]$ was defined for submatrix $A[i_s, i_e; j_s, j_e]$ represented by vertex $v \in \mathcal{V}$ of the quad-tree $Q = (\mathcal{V}, R)$ as follows (Briers et al., 2016):

$$\mu^{(c)}(v) = (\mu_1^{(c)}, \mu_2^{(c)})$$

$$\mu_1^{(c)} = \tfrac{1}{(i_e - i_s + 1)(j_e - j_s + 1)} \sum_{i,j \in \{i_s, \cdots, i_e\} \times \{j_s, \cdots, j_e\}} a_{ij}^{(c)}, \tag{A.1}$$

$$\mu_2^{(c)} = \tfrac{1}{(i_e - i_s + 1)(j_e - j_s + 1)} \sum_{i,j \in \{i_s, \cdots, i_e\} \times \{j_s, \cdots, j_e\}} (a_{ij}^{(c)} - \mu_1^{(c)})^2,$$

where $c \in \{r, g, b\}$ was an RGB color. The function $\mu^{(c)}$ provided the mean value and variance for the concentration of RGB colors in a particular region of the space represented by the vertex v.

Quad-trees can be interpreted as multi resolution representation of images, as the nodes that appear in deeper levels provide statistical information for higher resolutions and nodes that appear on higher levels correspond to more global characteristics of an image.

A.1.2 Tree Spatial Superposition Logic

In (Bartocci et al., 2016a), a formal logic, called tree spatial superposition logic (TSSL), was introduced. TSSL is capable of formally specifying global patterns in a network of locally interacting agents. The authors showed that this logic is sophisticated enough to describe complicated patterns such as Turing patterns in biochemical reaction-diffusion systems. In this paper, we used this logic to express various patterns that are studied here (Fig. 3.4). First, we present a brief introduction to TSSL. The reader can refer to (Bartocci et al., 2016a) for a thorough explanation of this logic, definitions of syntax and semantics, and its properties.

A TSSL formula is recursively constructed using the following:

- Linear predicates over valuations for the representation function. For example: $\mu_1^{(r)} > 0.8$ or $\mu_1^{(b)} < 0.5$.
- Boolean operators, such as $\neg\phi$, $\phi_1 \wedge \phi_2$, and $\phi_1 \vee \phi_2$.
- Spatial operators: $\exists_B \bigcirc \phi$, $\forall_B \bigcirc \phi$, where B is a nonempty subset of the set of directions $\{NW, NE, SW, SE\}$.

The spatial operators $\exists_B\bigcirc$ and $\forall_B\bigcirc$ are read as *there exists in directions B next* and *for all directions B next*, respectively. $\exists_B \bigcirc \phi$ is interpreted as follows: For at least one of the nodes located in the next level of the quad-tree labeled with one of the directions in B, ϕ must be satisfied. $\forall_B\bigcirc\phi$ specifies that for all such nodes ϕ must be satisfied. We demonstrate how TSSL can be used to express spatial patterns through an example.

Consider a 4×4 checkerboard as illustrated in Fig. 2.10(a). This pattern can be expressed as the following TSSL formula φ. The quad-tree of Fig. 2,10(b) satisfies φ.

$$\begin{aligned}
&\varphi = \forall_{\{NW,NE,SW,SE\}} \bigcirc [\\
&\forall_{\{NE,SW\}} \bigcirc (\mu_1^{(r)} \geq 1 \wedge \mu_1^{(g)} \geq 1 \wedge \mu_1^{(b)} \geq 1) \wedge\\
&\forall_{\{NW,SE\}} \bigcirc (\mu_1^{(r)} \leq 0 \wedge \mu_1^{(g)} \leq 0 \wedge \mu_1^{(b)} \leq 0)]
\end{aligned} \tag{A.2}$$

TSSL formulas can be viewed as formal pattern descriptors or pattern classifiers. Although TSSL is capable of describing complicated spatial behaviors in an image, it is difficult in general to write a formula that describes a complex pattern. In (Bartocci et al., 2016a), the authors proposed to use machine learning techniques in order to find such a formula from a given set of positive and negative examples.

Assume a set of positive images ($\mathcal{Y}_+$), illustrating a desirable pattern, and a set of negative images ($\mathcal{Y}_-$), in which the desirable pattern was not present, were available. We created a set $\mathcal{L}$ from these images as:

$$\mathcal{L} = \{(Q_y, +) \mid y \in \mathcal{Y}_+\} \cup \{(Q_y, -) \mid y \in \mathcal{Y}_-\},$$

where Q_y was the quad-tree generated from image y. The set $\mathcal{L}$ was separated into a learning set $\mathcal{L}_L$ (used to train a classifier) and a testing set $\mathcal{L}_T$ (used to test the classifier obtained from $\mathcal{L}_L$) such that $\mathcal{L} = \mathcal{L}_L \cup \mathcal{L}_T$. A rules-based learner called RIPPER (Cohen, 1995) was used to learn a set of classification rules from $\mathcal{L}_L$. Each of these rules was in the form:

$$R_i : C_i \Rightarrow Label_i,$$

where C_i was a Boolean formula over linear predicates over the representation values of the nodes of a quad-tree and $Label_i \in \{+, -\}$. We used the Weka workbench (Frank et al., 2016) for implementing RIPPER. Each C_i was then translated into an equivalent TSSL formula Φ_i. Since the classification rules were interpreted as nested if-else statements, the TSSL formula equivalent to the entire set of classification rules corresponding to the positive class was written as:

$$\Phi_+ = \bigvee_{j \in R_+} \left(\Phi_j \wedge \bigwedge_{i=1,\cdots,j-1} \neg \Phi_i \right), \tag{A.3}$$

where R_+ was the set of indices of rules labeled positive.

A.1.3 Quantitative Robustness

A TSSL formula can be created for any desired spatial pattern by following the procedure described in the previous section. If this formula is evaluated as true for a given image, it means that the image contains the required pattern. On the other hand, a false evaluation of the formula means that the pattern does not exist. However, this qualitative evaluation of TSSL descriptors does not provide any information about how strongly an image demonstrates the required pattern.

In order to provide information information about how strongly an image satisfies or violates the given property, TSSL was also equipped with a recursive quantitative semantics definition which assigned a real value to a TSSL formula ϕ with respect to vertex $v \in \mathcal{V}$ of quad-tree $Q = (\mathcal{V}, R)$; denoted by $\rho(\phi, v)$. The TSSL quantitative valuation was derived recursively as follows:

- $\rho(\mu_i^{(c)} \geq d, v) = \mu_i^{(c)}(v) - d.$
- $\rho(\mu_i^{(c)} \leq d, v) = d - \mu_i^{(c)}(v).$
- $\rho(\neg\phi, v) = -\rho(\phi, v).$
- $\rho(\phi_1 \wedge \phi_2, v) = \min(\rho(\phi_1, v), \rho(\phi_2, v)).$
- $\rho(\phi_1 \vee \phi_2, v) = \max(\rho(\phi_1, v), \rho(\phi_2, v)).$
- $\rho(\exists_B \bigcirc \phi, v) = 0.25 \max_{b \in B}(\rho(\phi, v_b))$ where v_b was the child vertex of v with label b.
- $\rho(\forall_B \bigcirc \phi, v) = 0.25 \min_{b \in B}(\rho(\phi, v_b))$ where v_b was the child vertex of v with label b.

It was proven in (Bartocci et al., 2016a) that TSSL quantitative semantics are sound. In other words, a quad-tree Q satisfied a formula ϕ ($Q \models \phi$) if $\rho(\phi, v_0) > 0$ where v_0 was the root of Q, and Q violated ϕ ($Q \not\models \phi$) if $\rho(\phi, v_0) < 0$. Therefore, the problem of checking whether an image contains a pattern expressed as a TSSL formula was reduced to computing its quantitative valuation $\rho(\phi, v_0)$. Moreover, the absolute value of $\rho(\phi, v_0)$ was viewed as a measure of how strongly ϕ was satisfied (or violated) by Q. Hence, the quantitative valuation of a formula with respect to a quad-tree was called its *robustness*. This property is demonstrated in Fig. 3.3.

A.1.4 Particle Swarm Optimization

Consider an agent-based model with a set of parameters $p \in \Omega \subset \mathbb{R}^{N_p}$, where Ω was the possible set of parameter ranges and N_p was the total number of parameters. For instance, in the model described in Chapter 3, we had $N_p = 5$ parameters.

The output of the model was a sequence of T images where $A[t]$ was the image corresponding to time step $t \in \{0, 1, \ldots, T\}$ and T was the total duration of simulation. Our goal was to determine parameter values that result in emergence of a required pattern in the sequence of images derived from the model. Recall that we could specify the pattern using a TSSL formula Φ_{Pattern}. Moreover, each image $A[t]$ could be translated into a corresponding quad-tree $Q[t]$ with root $v_0[t]$. Therefore, for a fixed parameterization p, we could quantify the resulting sequence of images with $S(p)$ using the following equation:

$$S(p) = \max_{0 \le t \le T} \rho(\Phi_{\text{Pattern}}, v_0[t]), \tag{A.4}$$

where ρ was the TSSL robustness as described in the previous section. Note that since the model (Chapter 3) was stochastic in nature, $S(p)$ was a random variable and would have a different value every time a sample simulation was produced using the model with the parameters p. If $S(p) > 0$, there exists at least one image in that particular sequence for which the TSSL robustness was positive and the pattern was present. On the other hand, the pattern had not emerged in the sample simulation if $S(p) < 0$. We called $S(P)$ the *robustness* degree for parametrization p. Now, the problem became finding the parameterization p^* that maximized the score $S(p)$. Since $S(p)$ was a random variable, we choose to maximize its expected value:

$$p^* = \arg\max_{p \in \Omega} \mathbb{E}(S(p)), \tag{A.5}$$

which means that we were looking for the parameterization p^* that *on average* produceed patterns with highest possible robustness score. If we simulated the model n times from parameters p, the expected value could be approximated with the sample mean:

$$\mathbb{E}(S(p)) \approx \tilde{S}(p) = \frac{1}{n} \sum_{i=1}^{n} S_i(p), \tag{A.6}$$

where $S_i(p)$ was the robustness score for parameters p in the ith simulation. In general, a large sample is needed to achieve an accurate approximation. however, it was shown in (Haghighi et al., 2015) that in practice, a relatively small n suffices for the purpose of optimization. In this paper, we computed the average robustness for three sample simulations in every case ($n = 3$).

Many optimization methods can be used to solve this optimization problem. Inspired by (Bartocci et al., 2016a), we employed particle swarm optimization (PSO) (Kennedy, 2011) to solve this problem. PSO is a heuristic solution to unconstrained optimization problems that is capable of solving problems with irregular search spaces,is easily distributable, and does not require the objective function to be differentiable.

The PSO algorithm worked as follows: The procedure began by randomly initializing a set of M particles with positions $z_i \in \Omega$ and velocities z_i'. The position of a particle was a candidate solution, and the velocity was a search direction from the current solution. Next, n simulations were produced and n sequences of quadtrees $Q[t](z_i)$ were created for each particle and the average robustness degree $\tilde{S}(z_i)$ was evaluated for each set of simulations represented by particle z_i. The position of the ith particle that had performed best so far was stored in the variable z_i^*, and the optimal value of z_i^* was denoted by z^*. After all particles had been evaluated, the positions and velocities were updated according to the following relations:

$$\begin{aligned} z_i' &\leftarrow W z_i + \eta(r_p)(z_i^* - z_i) + \eta(r_g)(z^* - z_i) \\ z_i &\leftarrow z_i + z_i', \end{aligned} \tag{A.7}$$

where $\eta(r_i)$ was a random number uniformly distributed over $[0, r_i]$ and the parameters $W \in \mathbb{R}, r_p, r_g$ are tuned by the user (Shi & Eberhart, 1998). This iterative process continued until a termination criterion was met.

If $\tilde{S}(p^*)$ was positive or negative but sufficiently close to zero, we had found the optimal parameterization of the model for the required pattern. This occurred for the Bullseye and Multi-Islands patterns in Fig. 3.4D,E. The optimal parameterization is shown in Fig. 3.4. On the other hand, $S(p^*) \ll 0$ indicated that even for the best possible parameterization of the model, the required pattern did not emerge, meaning that the model was not capable of producing that pattern at all. This occurred for the Janus (Left-Right) pattern (Fig. 4F).

Fig. 2.11 demonstrates two sample simulations, one for the Bullseye pattern and one for the Multi-Islands pattern. Fig. 2.12 shows how the corresponding TSSL scores evolve over time for each simulation. It is seen in this figure that the scores gradually improve until at some points the desired patterns are formed

BIBLIOGRAPHY

(2013). Differential adhesion in model systems. *WIREs Dev Biol*, *2*(October), 631–645.

Bartocci, E., Gol, E. A., Haghighi, I., & Belta, C. (2016a). A formal methods approach to pattern recognition and synthesis in reaction diffusion networks. *IEEE Transactions on Control of Network Systems*.

Bartocci, E., Gol, E. A., Haghighi, I., & Belta, C. (2016b). A formal methods approach to pattern recognition and synthesis in reaction diffusion networks. *IEEE Transactions on Control of Network Systems*, *PP*(99), 1–1.

Bauer, S., Nolte, L.-P., & Reyes, M. (2011). Fully automatic segmentation of brain tumor images using support vector machine classification in combination with hierarchical conditional random field regularization. In G. Fichtinger, A. Martel, & T. Peters (Eds.) *Medical Image Computing and Computer-Assisted Intervention MICCAI 2011*, Lecture Notes in Computer Science, (p. 354361). Springer Berlin Heidelberg.

Bentley, J. L. (1975). Multidimensional Binary Search Trees Used for Associative Searching. *ACM*, *18*(9), 509–517.

Bentley et al., K. (2014). The role of differential VE-cadherin dynamics in cell rearrangement during angiogenesis. *Nature cell biology*, *16*(4), 309–321.

Bernitz, J. M., Kim, H. S., MacArthur, B., Sieburg, H., & Moore, K. (2016). Hematopoietic stem cells count and remember self-renewal divisions. *Cell*, *167*(5), 1296–1309.e10.

Bianconi, E., Piovesan, A., Facchin, F., Beraudi, A., Casadei, R., Frabetti, F., Vitale, L., Pelleri, M. C., Tassani, S., Piva, F., Perez-Amodio, S., Strippoli, P., & Canaider, S. (2013). An estimation of the number of cells in the human body. *Annals of Human Biology*, *40*(6), 463–471. PMID: 23829164.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag.

Bredenoord, A. L., Clevers, H., & Knoblich, J. A. (2017). Human tissues in a dish: The research and ethical implications of organoid technology. *Science*, *355*(6322), eaaf9414.

Briers, D., Haghighi, I., White, D., Kemp, M. L., & Belta, C. (2016). Pattern synthesis in a 3d agent-based model of stem cell differentiation. In *IEEE 55th Conference on Decision and Control (CDC)*, (pp. 4202–4207).

Brodland, G. W. (2002). The Differential Interfacial Tension Hypothesis (DITH): a comprehensive theory for the self-rearrangement of embryonic cells and tissues. *Journal of biomechanical engineering*, *124*(2), 188–197.

Cachat, E., Liu, W., Hohenstein, P., & Davies, J. A. (2014). A library of mammalian effector modules for synthetic morphology. *Journal of Biological Engineering*, *8*(1), 26.

Campanini, R., Dongiovanni, D., Iampieri, E., Lanconelli, N., Masotti, M., Palermo, G., Riccardi, A., & Roffilli, M. (2004). A novel featureless approach to mass detection in digital mammograms based on support vector machines. *Physics in Medicine and Biology*, *49*(6), 961975.

Cha, J., Bae, H., Sadr, N., Manoucheri, S., Edalat, F., Kim, K., Kim, S., Kwon, I., Hwang, Y.-S., & Khademhosseini, A. (2015). Embryoid body size-mediated differential endodermal and mesodermal differentiation using polyethylene glycol (peg) microwell array. *Macromolecular Research*, *23*(3), 245–255.

Chandra Ravi A., Douglas Erik S., Mathies Richard A., Bertozzi Carolyn R., & Francis Matthew B. (2005). Programmable Cell Adhesion Encoded by DNA Hybridization. *Angewandte Chemie*, *118*(6), 910–915.

Chiu, T.-Y., & Jiang, J.-H. R. (2017). Logic synthesis of recombinase-based genetic circuits. *Scientific Reports*, *7*(1), 12873.

Ciçek, ., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3d u-net: Learning dense volumetric segmentation from sparse annotation. In S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, & W. Wells (Eds.) *Medical Image Computing and Computer-Assisted Intervention MICCAI 2016*, Lecture Notes in Computer Science, (p. 424432). Springer International Publishing.

Cohen, W. W. (1995). Fast effective rule induction. *Machine Learning Proceedings 1995*, (pp. 115–123).

Czirók, A., Varga, K., Méhes, E., & Szabó, A. (2013). Collective cell streams in epithelial monolayers depend on cell adhesion. *New Journal of Physics*, *15*.

Darabi, R., Gehlbach, K., Bachoo, R. M., Kamath, S., Osawa, M., Kamm, K. E., Kyba, M., & Perlingeiro, R. C. R. (2008). Functional skeletal muscle regeneration from differentiating embryonic stem cells. *Nature Methods*, *14*(2), 134–143.

Davies, J. A. (2008). Synthetic morphology: prospects for engineered, self-constructing anatomies. *Journal of Anatomy*, *212*(6), 707–719.

Deglincerti, A., Croft, G. F., Pietila, L. N., Zernicka-Goetz, M., Siggia, E. D., & Brivanlou, A. H. (2016). Self-organization of the *in vitro* attached human embryo. *Nature*, *533*(7602), 251–254.

Demers, C. J., Soundararajan, P., Chennampally, P., Cox, G. A., Briscoe, J., Collins, S. D., & Smith, R. L. (2016). Development-on-chip: in vitro neural tube patterning with a microfluidic device. *Development*, *143*(11), 1884–1892.

Doench, J. G., Fusi, N., Sullender, M., Hegde, M., Vaimberg, E. W., Donovan, K. F., Smith, I., Tothova, Z., Wilen, C., Orchard, R., Virgin, H. W., Listgarten, J., & Root, D. E. (2016). Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. *Nature Biotechnology*, *34*(2), 184–191.

Ducibella, T., & Anderson, E. (1975). Cell shape and membrane changes in the eight-cell mouse embryo: Prerequisites for morphogenesis of the blastocyst. *Developmental Biology*, *47*(1), 45–58.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, *542*(7639), 115118.

Finkel, R. A., & Bentley, J. L. (1974). Quad trees a data structure for retrieval on composite keys. *Acta informatica*, *4*(1), 1–9.

Foty, R. A., & Steinberg, M. S. (2005). The differential adhesion hypothesis: A direct evaluation. *Developmental Biology*, *278*(1), 255–263.

Frank, E., Hall, M., & Witten, I. (2016). The weka workbench. *Data mining: Practical machine learning tools and techniques. Burlington: Morgan Kaufmann*.

GeunChung, B., A.Flanagan, L., WooRhee, S., H.Schwartz, P., P.Lee, A., S.Monuki, E., & LiJeon, N. (2005). Human neural stem cell growth and differentiation in a gradient-generating microfluidic device. *Lab on a Chip*, *5*(4), 401–406.

Glen, C. M., McDevitt, T. C., & Kemp, M. L. (2018). Dynamic intercellular transport modulates the spatial patterning of differentiation during early neural commitment. *Nature Communications*, *9*(1), 4111.

Gol, E. A., Bartocci, E., & Belta, C. (2014). A formal methods approach to pattern synthesis in reaction diffusion systems. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, (pp. 108–113). IEEE.

Graner, F., & Glazier, J. A. (1992). Simulation of biological cell sorting using a two-dimensional extended Potts model. *Physical Review Letters*, *69*(13), 2013–2016.

Greber, D., & Fussenegger, M. (2010). An engineered mammalian band-pass network. *Nucleic Acids Research*, *38*(18), e174–e174.

Grosu, R., Smolka, S. A., Corradini, F., Wasilewska, A., Entcheva, E., & Bartocci, E. (2009). Learning and detecting emergent behavior in networks of cardiac myocytes. *Communications of the ACM*, *52*(3), 97–105.

Haghighi, I., Jones, A., Kong, Z., Bartocci, E., Grosu, R., & Belta, C. (2015). Spatel: a novel spatial-temporal logic and its applications to networked systems. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, (pp. 189–198). ACM.

Homan, K. A., Kolesky, D. B., Skylar-Scott, M. A., Herrmann, J., Obuobi, H., Moisan, A., & Lewis, J. A. (2016). Bioprinting of 3D Convoluted Renal Proximal Tubules on Perfusable Chips. *Scientific Reports*, *6*, 34845.

Hookway, T. A., Butts, J. C., Lee, E., Tang, H., & McDevitt, T. C. (2016). Aggregate formation and suspension culture of human pluripotent stem cells and differentiated progeny. *Methods*.

Hsiao, S. C., Shum, B. J., Onoe, H., Douglas, E. S., Gartner, Z. J., Mathies, R. A., Bertozzi, C. R., & Francis, M. B. (2009). Direct Cell Surface Modification with DNA for the Capture of Primary Cells and the Investigation of Myotube Formation on Defined Patterns. *Langmuir*, *25*(12), 6985–6991.

Huch, M., & Koo, B.-K. (2015). Modeling mouse and human development using organoid cultures. *Development*, *142*(18), 3113–3125.

Jackins, C. L., & Tanimoto, S. L. (1983). Quad-Trees, Oct-Trees, and K-Trees: A Generalized Approach to Recursive Decomposition of Euclidean Space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-5*(5), 533–539.

Jia, D., Dajusta, D., & Foty, R. A. (2007). Tissue surface tensions guide in vitro self-assembly of rodent pancreatic islet cells. *Developmental Dynamics*, *236*(8), 2039–2049.

Johnston, M. D., Edwards, C. M., Bodmer, W. F., Maini, P. K., & Chapman, S. J. (2007). Mathematical modeling of cell population dynamics in the colonic crypt and in colorectal cancer. *PNAS*, (pp. 24–29).

Jones, E., Oliphant, T., Peterson, P., et al. (2001–). SciPy: Open source scientific tools for Python. http://www.scipy.org/.

Kay, R. R., & Thompson, C. R. L. (2009). Forming Patterns in Development without Morphogen Gradients: Scattered Differentiation and Sorting Out. *Cold Spring Harbor Perspectives in Biology*, *1*(6), a001503.

Kennedy, J. (2010). Particle swarm optimization. In *Encyclopedia of Machine Learning*, (pp. 760–766). Springer.

Kennedy, J. (2011). Particle swarm optimization. *Encyclopedia of machine learning*, (pp. 760–766).

Kinney, M., Hookway, T., Wang, Y., & McDevitt, T. (2014). Engineering three-dimensional stem cell morphogenesis for the development of tissue models and scalable regenerative therapeutics. *Annals of Biomedical Engineering*, *42*(2), 352–367.

Kondo, S., & Miura, T. (2010). Reaction-diffusion model as a framework for understanding biological pattern formation. *Science*, *329*(5999), 1616–1620.

Krieg, M., Arboleda-Estudillo, Y., Puech, P.-H., Käfer, J., Graner, F., Müller, D. J., & Heisenberg, C.-P. C.-P. (2008). Tensile forces govern germ-layer organization in zebrafish. *Nature cell biology*, *10*(4), 429–36.

Libby, A., Joy, D., So, P.-L., Mandegar, M., Muncie, J., Mendoza-Camacho, F., Weaver, V., Conklin, B., & McDevitt, T. C. (2018a). Spatiotemporal mosaic self patterning of pluripotent stem cells using crispr interference. *eLife*, (p. 23).

Libby, A. R. G., Joy, D. A., So, P.-L., Mandegar, M. A., Muncie, J. M., Weaver, V. M., Conklin, B. R., & McDevitt, T. C. (2018b). Spatiotemporal mosaic patterning of pluripotent stem cells using CRISPR interference. *bioRxiv*, (p. 252189).

Livigni, A., & Brickman, J. M. (2013). Previews Oct4 : The Final Frontier , Differentiation Defining Pluripotency. *Developmental Cell*, *25*(6), 547–548.

Long, J., Shelhamer, E., & Darrell, T. (2014). Fully convolutional networks for semantic segmentation. *arXiv:1411.4038 [cs]*. ArXiv: 1411.4038.

Ludwig, T. E., Bergendahl, V., Levenstein, M. E., Yu, J., Probasco, M. D., & Thomson, J. A. (2006). Feeder-independent culture of human embryonic stem cells. *Nature Methods*, *3*(8), 637–646.

Maglogiannis, I. G., & Zafiropoulos, E. P. (2004). Characterization of digital medical images utilizing support vector machines. *BMC Medical Informatics and Decision Making*, *4*(1), 4.

Magno, R., Grieneisen, V. A., & Marée, A. F. (2015). The biophysical nature of cells: potential cell behaviours revealed by analytical and computational studies of cell surface mechanics. *BMC Biophysics*, *8*(1), 8.

Mandegar, M. A., Huebsch, N., Frolov, E. B., Shin, E., Truong, A., Olvera, M. P., Chan, A. H., Miyaoka, Y., Holmes, K., Spencer, C. I., Judge, L. M., Gordon, D. E., Eskildsen, T. V., Villalta, J. E., Horlbeck, M. A., Gilbert, L. A., Krogan, N. J., Sheikh, S. P., Weissman, J. S., Qi, L. S., So, P.-L., & Conklin, B. R. (2016). CRISPR Interference Efficiently Induces Specific and Reversible Gene Silencing in Human iPSCs. *Cell Stem Cell*, *18*(4), 541–553.

Marée, A. F. M., Grieneisen, V. A., & Hogeweg, P. (2007). The Cellular Potts Model and Biophysical Properties of Cells, Tissues and Morphogenesis. In A. R. A. Anderson, M. A. J. Chaplain, & K. A. Rejniak (Eds.) *Single-Cell-Based Models in Biology and Medicine*, (pp. 107–136). Basel: Birkhäuser Basel. https://doi.org/10.1007/978-3-7643-8123-3_5.

Maître, J.-L., & Heisenberg, C.-P. (2013). Three functions of cadherins in cell adhesion. *Current Biology*, *23*(14), R626R633.

McNamara, H. M., Dodson, S., Huang, Y.-L., Miller, E. W., Sandstede, B., & Cohen, A. E. (2018). Geometry-Dependent Arrhythmias in Electrically Excitable Tissues. *Cell Systems*, (pp. 1–12).

Merks, R. M. H., Perryn, E. D., Shirinifard, A., & Glazier, J. A. (2008). Contact-inhibited chemotaxis in de novo and sprouting blood-vessel growth. *PLoS Computational Biology*, *4*(9).

Molitoris, J. M., Paliwal, S., Sekar, R. B., Blake, R., Park, J., Trayanova, N. A., Tung, L., & Levchenko, A. (2016). Precisely parameterized experimental and computational models of tissue organization. *Integr. Biol.*, *8*, 230–242.

Montero, J.-A., & Heisenberg, C.-P. (2004). Gastrulation dynamics: cells move into focus. *Trends in Cell Biology*, *14*(11), 620627.

Morrison, S. J., & Spradling, A. C. (2008). Stem cells and niches: Mechanisms that promote stem cell maintenance throughout life. *Cell*, *132*(4), 598 – 611.

Nanni, L., Lumini, A., & Brahnam, S. (2010). Local binary patterns variants as texture descriptors for medical image analysis. *Artificial Intelligence in Medicine*, *49*(2), 117125.

Nielsen, A. A. K., Der, B. S., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalski, E. A., Ross, D., Densmore, D., & Voigt, C. A. (2016). Genetic circuit design automation. *Science*, *352*(6281), aac7341.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, *9*(1), 62–66.

Ouchi, N. B., Glazier, J. A., Rieu, J. P., Upadhyaya, A., & Sawada, Y. (2003). Improving the realism of the cellular Potts model in simulations of biological cells. *Physica A: Statistical Mechanics and its Applications*, *329*(3-4), 451–458.

Palsson, E. (2008). A 3-D model used to explore how cell adhesion and stiffness affect cell sorting and movement in multicellular systems. *Journal of Theoretical Biology*, *254*(1), 1–13.

Park, S., Kim, D., Jung, Y.-G., & Roh, S. (2015). Thiazovivin, a Rho kinase inhibitor, improves stemness maintenance of embryo-derived stem-like cells under chemically defined culture conditions in cattle. *Animal Reproduction Science*, *161*, 47–57.

Pir, P., & Le Novère, N. (2016). Mathematical Models of Pluripotent Stem Cells: At the Dawn of Predictive Regenerative Medicine. In U. Schmitz, & O. Wolkenhauer (Eds.) *Systems Medicine*, (pp. 331–350). New York, NY: Springer New York. https://doi.org/10.1007/978-1-4939-3283-2_15.

Ricotti, L., Trimmer, B., Feinberg, A. W., Raman, R., Parker, K. K., Bashir, R., Sitti, M., Martel, S., Dario, P., & Menciassi, A. (2017). Biohybrid actuators for robotics: A review of devices actuated by living cells. *Science Robotics*, *2*(12), eaaq0495.

Sasai, Y. (2013). Next-generation regenerative medicine: Organogenesis from stem cells in 3d culture. *Cell Stem Cell*, *12*(5), 520530.

Scadden, D. T. (2006). The stem-cell niche as an entity of action. *441*(June), 1075–1079.

Sharpe, J. (2017). Computer modeling in developmental biology: growing today, essential tomorrow. *Development*, *144*(23), 4214–4225.

Shi, Y., & Eberhart, R. C. (1998). Parameter selection in particle swarm optimization. In *International conference on evolutionary programming*, (pp. 591–600). Springer.

Shkumatov, A., Baek, K., & Kong, H. (2014). Matrix rigidity-modulated cardiovascular organoid formation from embryoid bodies. *PLoS ONE*, *9*(4), e94764.

Sohka, T., Heins, R. A., Phelan, R. M., Greisler, J. M., Townsend, C. A., & Ostermeier, M. (2009). An externally tunable bacterial band-pass filter. *Proceedings of the National Academy of Sciences*, *106*(25), 10135–10140.

Starruß, J., De Back, W., Brusch, L., & Deutsch, A. (2014). Morpheus: A user-friendly modeling environment for multiscale and multicellular systems biology. *Bioinformatics*, *30*(9), 1331–1332.

Steinberg, M. S. (1975). Adhesion-guided multicellular assembly: a commentary upon the postulates, real and imagined, of the differential adhesion hypothesis, with special attention to computer simulations of cell sorting. *Journal of Theoretical Biology*, *55*(2), 431 – 443.

Sun, Z., & Komarova, N. L. (2012). Stochastic modeling of stem-cell dynamics with control. *Mathematical Biosciences*, *240*(2), 231–240.

Szabó, A., & Merks, R. (2013). Cellular potts modeling of tumor growth, tumor invasion, and tumor evolution. *Frontiers in oncology*, *3*(April), 87.

Szabó, A., Unnep, R., Méhes, E., Twal, W. O., Argraves, W. S., Cao, Y., & Czirók, A. (2010). Collective cell motion in endothelial monolayers. *Physical Biology*, *7*(4), 046007.

Taylor, H. B., Khuong, A., Wu, Z., Xu, Q., Morley, R., Gregory, L., Poliakov, A., Taylor, W. R., & Wilkinson, D. G. (2017). Cell segregation and border sharpening by eph receptor–ephrin-mediated heterotypic repulsion. *Journal of The Royal Society Interface*, *14*(132).

Tewary, M., Ostblom, J., Prochazka, L., Zulueta-Coarasa, T., Shakiba, N., Fernandez-Gonzalez, R., & Zandstra, P. W. (2017). A stepwise model of Reaction-Diffusion and Positional-Information governs self-organized human peri-gastrulation-like patterning. *Development*, (p. dev.149658).

Théry, M. (2010). Micropatterning as a tool to decipher cell morphogenesis and functions. *J Cell Sci*, *123*(24), 4201–4213.

Théry, M., Racine, V., Piel, M., Pépin, A., Dimitrov, A., Chen, Y., Sibarita, J.-B., & Bornens, M. (2006). Anisotropy of cell adhesive microenvironment governs cell internal organization and orientation of polarity. *Proceedings of the National Academy of Sciences*, *103*(52), 19771–19776.

Toda, S., Blauch, L. R., Tang, S. K., Morsut, L., & Lim, W. A. (2018). Programming self-organizing multicellular structures with synthetic cell-cell signaling. *Science*, *361*(6398), 156–162.

Vaca-González, J. J., Gutiérrez, M. L., Guevara, J. M., & Garzón-Alvarado, D. A. (2017). Cellular automata model for human articular chondrocytes migration, proliferation and cell death: An in vitro validation. *In Silico Biology*, *12*(3-4), 83–93.

Van Liedekerke, P., Palm, M. M., Jagiella, N., & Drasdo, D. (2015). Simulating tissue mechanics with agent-based models: concepts, perspectives and some novel results. *Computational Particle Mechanics*, *2*(4), 401–444.

Vasile, C.-I. (2016). *Motion planning and control: a formal methods approach*. Ph.D. thesis, Boston University. https://open.bu.edu/handle/2144/17081.

Voss-Bohme, A. (2012). Multi-scale modeling in morphogenesis: a critical analysis of the cellular Potts model. *PLoS ONE*, *7*(9), e42852.

Walt, S. v. d., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., & Yu, T. (2014). scikit-image: image processing in Python. *PeerJ*, *2*, e453.

Warmflash, A., Sorre, B., Etoc, F., Siggia, E. D., & Brivanlou, A. H. (2014). A method to recapitulate early embryonic spatial patterning in human embryonic stem cells. *Nature Methods*, *11*, 847.

Watanabe, K., Ueno, M., Kamiya, D., Nishiyama, A., Matsumura, M., Wataya, T., Takahashi, J. B., Nishikawa, S., Nishikawa, S.-i., Muguruma, K., & Sasai, Y. (2007). A ROCK inhibitor permits survival of dissociated human embryonic stem cells. *Nature Biotechnology*, *25*(6), 681–686.

White, D. E., Kinney, M. a., McDevitt, T. C., & Kemp, M. L. (2013). Spatial pattern dynamics of 3D stem cell loss of pluripotency via rules-based computational modeling. *PLoS computational biology*, *9*(3), e1002952.

White, D. E., Sylvester, J. B., Levario, T. J., Lu, H., Todd Streelman, J., McDevitt, T. C., & Kemp, M. L. (2015). Quantitative multivariate analysis of dynamic multicellular morphogenic trajectories. *Integr. Biol.*.

CURRICULUM VITAE

[redacted]

[redacted]

[redacted]

[redacted]