

QUANTIFYING ROBUSTNESS OF THE GAP GENE NETWORK

by

Elizabeth Anne Andreas

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

Doctor of Philosophy

in

Mathematics

MONTANA STATE UNIVERSITY
Bozeman, Montana

May 2024

© COPYRIGHT

by

Elizabeth Anne Andreas

2024

All Rights Reserved

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to several people and organizations without whom this journey would not have been possible.

First and foremost, I genuinely appreciate and thank my advisors, Tomáš and Bree, for their guidance, expertise, and continuous support throughout my PhD journey. Their dedication has been crucial to my academic and personal growth, and I deeply value their mentorship.

I'd also like to thank the rest of my committee, Jack, Scott and Lisa for their support, encouragement and time. The courses I have taken from Scott and Jack have reinforced my love of applied mathematics. My research would not have been possible without the generous funding support from the MT PEAKS program, granted to me by Lisa, and the SMART scholarship. They made it possible for me to focus on my work without financial stress, and for that, I am truly grateful.

To my fiancé, Joshua, for enduring the highs and lows of this process with me, I'm grateful. Your perseverance has been a silent form of support and your belief in me has been a source of strength and motivation. Thank you for being there, every step of the way.

To each person who has been a part of this journey, your support has been vital and appreciated. Thank you for the roles you've played in helping me achieve this goal.

TABLE OF CONTENTS

1. INTRODUCTION	1
2. BACKGROUND	5
2.1 Graph Theory	6
2.2 Drosophila Melanogaster.....	9
2.2.1 Anterior-Posterior (A-P) Patterning in Drosophila Melanogaster.....	9
2.2.2 The Gap Gene Regulatory Network [48].....	12
2.2.3 ACDC Dynamic Modules of the Gap Gene Network	13
2.3 Statistical Background	15
2.3.1 Absorbing Markov Chain	15
2.3.2 Multiple Linear Regression	16
2.4 DSGRN	23
2.4.1 Switching Systems	23
2.4.2 Factor Graph and DSGRN Parameter Graph	24
2.4.3 State Transition Graph (STG).....	30
2.4.4 Morse Graphs.....	33
3. SPATIAL MODELING USING DSGRN.....	35
3.1 Factor Graph Layers	35
3.2 Factor Graph Layers for DSGRN Realizable Parameters	41
3.3 Interpreting External Variables as Parameter Changes	46
4. EXPRESSING EXPERIMENTAL DATA AS MORSE GRAPHS.....	48
4.1 Descriptive Pattern and Phenotype Pattern Graph	48
4.2 Drosophila MelanogasterExample	50
5. MODELING SPATIAL GRADIENTS AND MATCHING OBSERVA- TIONS WITH DSGRN	55
5.1 Chemical Gradient Graph	56
5.2 Developmental Paths	57
5.3 Condensed Chemical Gradient Graph	59
5.4 Path Graph.....	66
6. ROBUSTNESS SCORES	71
6.1 Bottlenecks Scored by Optimized Weighted Cut(QWCut) of the Path Graph	73

TABLE OF CONTENTS { CONTINUED

6.2 Using Absorbing Markov Chains (AMC) to Score Leak (P2) and Skip (P3)	78
6.3 Size of Lifted Path Graph in Chemical Gradient Graph (P4)	79
6.4 Scoring	80
7. RESULTS	81
8. DISCUSSION	90
REFERENCES CITED	94
APPENDICES	99
APPENDIX A : Model Results and MLR Validity	100
APPENDIX B : Network Measure Results Before Normalization	103

LIST OF TABLES

Table	Page
2.1 Logic parameters.....	27
4.1 Data discretization.....	52
7.1 Sampling Results	84
A.01 Basic analysis results.....	101
A.02 MLR model results	101
A.03 VIF results for MLR model	102

LIST OF FIGURES

Figure	Page
2.1 Trunk gap gene expression.....	11
2.2 Gap gene network	13
2.3 ACDC networks and model depiction	14
2.4 Multiple linear regression example.....	19
2.5 Example of a DSGRN computation.....	25
2.6 Depiction of a factor graph	29
2.7 Overview of DSGRN	30
3.1 Factor graph layers and modeling external variables	40
4.1 A-P axis data and regions.....	51
4.2 StrongEdges network and phenotype pattern	53
4.3 StrongEdges phenotype pattern graph	54
5.1 Standard condensation graph verse condensation via Morse graph equivalence.....	61
5.2 Condensed chemical gradient graphs for FullConn and StrongEdges	63
5.3 Diagonal sets $S_{i,j}^d$ and $S_{i,j}^a$ example.....	68
5.4 Path graph for FullConn and StrongEdges.....	70
6.1 Bottlenecks, attraction, region skipping and path graph size	72
7.1 Network features	83
7.2 Violin plots of normalized robustness scores	86
7.3 Network topology for best and worst scoring networks.....	87
7.4 Regression coefficients for MLR model explanatory variables.....	88
A.01 MLR diagnostic plots	102
B.01 Violin plots of robustness scores before normalization.....	104

ABSTRACT

Early development of *Drosophila melanogaster* (fruit fly) facilitated by the gap gene network has been shown to be incredibly robust, and the same patterns emerge even when the process is seriously disrupted. We investigate this robustness using a previously developed computational framework called DSGRN (Dynamic Signatures Generated by Regulatory Networks). Our mathematical innovations include the conceptual extension of this established modeling technique to enable modeling of spatially monotone environmental effects, as well as the development of a collection of graph theoretic robustness scores for network models. This allows us to rank order the robustness of network models of cellular systems where each cell contains the same genetic network topology but operates under a parameter regime that changes continuously from cell to cell. We demonstrate the power of this method by comparing the robustness of two previously introduced network models of gap gene expression along the anterior-posterior axis of the fruit fly embryo, both to each other and to a random sample of networks with same number of nodes and edges. We observe that there is a substantial difference in robustness scores between the two models. Our biological insight is that random network topologies are in general capable of reproducing complex patterns of expression, but that using measures of robustness to rank order networks permits a large reduction in hypothesis space for highly conserved systems such as developmental networks.

CHAPTER ONE

INTRODUCTION

Molecular processes in cells are subject to substantial levels of noise caused by variability in the number of enzymes and in other cellular machinery, as well as thermal noise that may affect enzymatic rates. In spite of facing this high inherent level of uncertainty, certain macroscopic phenotypes of the cell are very predictable and robust. This is particularly true for developmental programs, where the final phenotype is very robust to even severe perturbations. Understanding the principles of genetic network structure and a set of controls that are responsible for this robustness have been at the center of interest for many years.

One of the best studied systems is the segmentation of the *Drosophila melanogaster* (fruit fly) body plan during development. The segmentation is determined through gap, pair-rule and segment-polarity genes. In this study, we focus on the regulation of the gap genes *hunchback* (*hb*), *giant* (*gt*), *Krüppel* (*Kr*) and *knirps* (*kni*) which comprise the gap gene network and are responsible for establishing segmentation along the anterior-posterior (A-P) axis of the embryo. Initial conditions for gap gene expression are given by maternal gradients of the proteins *Bicoid* (*Bcd*) and *Caudal* (*Cad*), which are inherited by the embryo from the mother and present in decreasing and increasing amounts along the anterior-posterior (A-P) axis, respectively [19, 18].

This system has been modeled by several research groups [48, 25, 34, 19]. To explain the experimental data, Verdet al. [48] assume that there are different subnetworks, called ACDC dynamic modules, active in different regions along the A-P axis. They showed that each module could reproduce the data observed in each particular region at the end of the late stages of gap gene expression.

In this thesis, we propose that a single network functioning at different parameter values across spatial locations can explain the observed data at the end of late-stage gap gene expression, in contrast to a sequence of distinct networks. In particular, we hypothesize that the levels of maternal gradients Bcd and Cad provide different parameterizations for the gap gene network, and that such a parameterized collection of copies of the same network is responsible for the formation of the segmentation pattern. Apart from establishing if such a model is capable of reproducing experimental data, we are also interested in the question of robustness. How robust is such a t ?

To answer these questions we developed a model to evaluate if a parameterized network fits spatial experimental data, and we developed several network model robustness scores that we use to quantify the robustness of that t . This model is computationally efficient enough to be evaluated over hundreds of very large graphs representing network model dynamical behavior. To accomplish this, we use the DSGRN (Dynamic Signatures Generated by Regulatory Networks) [7, 11] approach previously used for assessing network model t with time series data and use it to model spatial data. For a regulatory network RN , DSGRN constructs a parameter graph $PGrN$ which represents a finite decomposition of the parameter space for an ODE model dRN , where paths in $PGrN$ represent a continuous change of parameters in this ODE system. For each parameter node $p \in PGrN$, DSGRN computes the summary of network dynamics from which one can extract a qualitative description of the stable equilibria of the system. To match a network model RN to spatial data, we seek paths $ir \in PGrN$ along which the qualitative description of stable equilibria matches experimentally observed expression levels of gene products. If such a path exists, we say the network RN is capable of reproducing the data.

To address the robustness question, for each network RN we study the shape of the subgraph P of all such matching paths. We evaluate to what extent this subgraph has bottlenecks (indicating the fragility of development at some spatial position), we score how

many paths can leave the subgraph \mathbb{P} without completing the developmental program, we score how many paths can skip a segment, and we evaluate the overall size of a subgraph of the graph of all paths. We compute these scores for nearly 1000 network models that have the same number of nodes (4) and edges (8) as two "canonical" network models. One of them is the network that is the union of the three ACDC submodules proposed by Verlet al. [48]. The second network is a subnetwork consisting of stronger regulatory interactions from the gap gene network derived by Verlet al. [47] using work by Ashyraliyev et al. [2].

The DSGRN approach to modeling network dynamics is an essential tool without which evaluating the complete set of global dynamics of hundreds of networks with 4 nodes and 8 edges would not be possible. However, even with this approach, there are more paths in \mathbb{P}_{GRN} that have to be examined than can be reasonably computed. We develop graph constructions based on condensation graphs that allow the computation and handling of these large sets.

Our analysis produces evidence suggesting that previously explored network models and motifs tend to have higher robustness scores when compared to randomly generated networks, indicating consistency of our results with previous work [48]. On the other hand, more local features such as the number of positive loops, number of negative loops, or number of negative edges does not seem to have a significant effect on robustness scores. Importantly, our work implies that particular features of network structure are capable of imparting robustness independent of the specific genes involved, which suggests that network structure itself may be subject to evolutionary pressure.

The organization of the thesis is as follows. In Chapter 2, we provide enough background on graph theory, *D. melanogaster*, Markov chains, and multiple linear regression, for the reader to obtain a solid understanding of modeling choices. This chapter also covers all DSGRN background necessary to understand the DSGRN parameter graph \mathbb{P}_{G} and Morse graphs, which are the "dynamic signatures" of DSGRN describing network behavior.

These are used in Chapter 4 to provide a mechanism for matching DSGRN predictions to experimental spatial data, such as that seen in [D. melanogasterdevelopment](#).

The interpretation of certain paths in PG as spatial expression patterns is presented in Chapter 3. In Chapter 5, we introduce carefully constructed subgraphs $\mathcal{P}G$ that incorporate information about spatial gradients, such as the maternal gradients important for proper segmentation of the [D. melanogasterembryo](#). Particularly important is a subgraph called the path graph. In Chapter 6, we quantify features of the path graph that permit us to assess the robustness of [D. melanogasterdevelopment](#) in terms of the breadth and quality of the match between DSGRN predictions and experimental observations. In Chapter 7, we apply these scores to nearly 1000 networks to compare robustness across network topology. We conclude with a discussion in Chapter 8.

Work based on this thesis has been published, see [1]. All scripts and processed data used to produce the figures and results in this manuscript can be found https://github.com/Eandreas1857/2023_GGN_Robustness

CHAPTER TWO

BACKGROUND

This chapter introduces the necessary background material for understanding the techniques developed in the presented work. Section 2.1 covers fundamental definitions in graph theory, progressing towards concepts such as strongly connected (path) components, condensation graphs, Hasse diagrams, and regulatory networks. The importance of strongly connected components lies in their role in graph theoretic methods developed in Chapter 5. Similarly, understanding strongly connected path components and Hasse diagrams is crucial for grasping the Morse graph of DSGRN, as detailed in Section 2.4. Regulatory networks serve as the input for DSGRN and are vital in biological studies. The section concludes by defining feedback loops, which are an important feature of complex regulatory networks.

Section 2.2 provides enough background on *Drosophila melanogaster* and the ACDC submodules for the reader to obtain a solid understanding of the modeling choices made throughout Chapters 3, 4, 5 and 6, as well as the difference in modeling choices from earlier work.

Section 2.3 contains an overview of absorbing Markov chains and multiple linear regression (MLR). Absorbing Markov chains contribute to our quantification of robustness in Section 6.2, while multiple linear regression is applied in Chapter 7 to analyze the data generated by our methods. The chapter concludes with an introduction to DSGRN, laying a foundational understanding of the DSGRN parameter graph $\mathbb{P}G$, which plays a crucial role throughout Chapter 3. Throughout this document, bold text corresponds to terms actively being defined.

2.1 Graph Theory

Let $G = (V, E)$ be an ordered pair where V is a finite set of vertices, called nodes, and E is a finite set of edges between the nodes. Specifically, if $E = \{ (u, v) \mid u, v \in V, u \neq v \}$, where each $(u, v) \in E$ is unordered, then G is called an undirected graph. When $E = \{ (u, v) \mid u, v \in V, u \neq v \}$, where each $(u, v) \in E$ is ordered, then G is called a directed graph. Note in this case, we call u the source and v the target of the edge (u, v) . Additionally, $|V|$ and $|E|$ denote the number of nodes and edges of a graph respectively.

Definition 2.1. A weighted directed graph $G = (V, E; W)$ is a directed graph equipped with a non-negative weight w_{ij} assigned to each directed edge $(u, v) \in E$. We organize weights in a $|V| \times |V|$ weight matrix $W = (w_{ij})$. If $(u, v) \in E$, then $w_{ij} \geq 0$.

A path in a graph G from $u \in V$ to $v \in V$ is a sequence of edges

$$(u, u_1), (u_1, u_2), \dots, (u_{n-1}, u_n), (u_n, v) \in E$$

from u to v . We denote a path from u to v by $u \rightsquigarrow v$. A path is called a cycle when $u = v$, and a simple cycle is a cycle with no repeated nodes or edges. A directed graph is acyclic if it contains no cycles, including self-loops. An undirected graph is acyclic if it doesn't contain a simple cycle.

Definition 2.2. A directed graph $G = (V, E)$ is strongly connected if every pair of nodes $u, v \in V$ has a directed path from u to v and from v to u . A strongly connected subgraph H of G is said to be maximal if there is no strongly connected subgraph $H^1 \subsetneq G$ with $H \subsetneq H^1 \subsetneq G$. A maximal strongly connected subgraph is referred to as a strongly connected component. Singleton vertices that are not strongly connected to any other node are strongly connected components that consist of that node and no edges. A strongly connected path component is a strongly connected component that has at least one edge [29].

Definition 2.3. The condensation graph of a directed graph G is an acyclic graph where each vertex represents a strongly connected component of the graph. An edge exists between two distinct nodes in the condensation graph, say u, v , whenever there is a path from one node in the strongly connected component represented by u to a node in the strongly connected component represented by v .

Hasse diagrams and regulatory networks are special types of graphs particularly important to our work. A Hasse diagram is a graphical representation of a finite partially ordered set (poset) that can be strict or non-strict. A non-strict poset is a set P , along with the operation \preceq , denoted by $p \preceq q$, with following properties

- $\hat{\ } p \preceq p$ for all $p \in P$ (reflexivity)
- $\hat{\ }$ if $p_0 \preceq p_1$ and $p_1 \preceq p_0$, then $p_0 = p_1$ (antisymmetry)
- $\hat{\ }$ if $p_0 \preceq p_1$ and $p_1 \preceq p_2$ then $p_0 \preceq p_2$ for all $p_0, p_1, p_2 \in P$ (transitivity).

An object (P, \preceq) is called a strict poset if

- $\hat{\ } p \bullet p$ for all $p \in P$ (irreflexivity)
- $\hat{\ }$ $p_0 \preceq p_1$ implies $p_1 \bullet p_0$ for all $p_0 \preceq p_1 \in P$ (asymmetry)
- $\hat{\ }$ if $p_0 \preceq p_1$ and $p_1 \preceq p_2$ then $p_0 \preceq p_2$ for $p_0 \preceq p_1 \preceq p_2 \in P$ (transitivity).

The transitive reduction of an acyclic graph $G = (V, E)$ is the unique subgraph $H = (V, E^1)$ with the smallest subset of edges $E^1 \subseteq E$ that satisfies the condition that a path $u \rightsquigarrow v$ exists in G if and only if a (possibly distinct) path $u \rightsquigarrow v$ exists in H . Then a Hasse diagram of a poset P is formally defined as the transitive reduction H of the acyclic graph $G = (P, E)$ with a directed edge $(u, v) \in E$ for $u \preceq v$ if and only if $u \bullet v$.

A regulatory network is the main conceptual model of gene cell regulation in systems biology. This structure expresses the molecular species that are controlled by other molecules

and the type of the directed interactions between them. The molecular species are usually proteins or mRNA molecules. Formally, a regulatory network is a directed graph, denoted $RN = (V; E)$, where V is the set of nodes and the edges $E \subseteq V \times V \times \{-1, 1\}$ denote interactions between the network nodes: the edge $(v_i; v_j; 1) \in E$ indicates that v_i is an activator of v_j (denoted by $v_i \tilde{N} v_j$), while the edge $(v_i; v_j; -1) \in E$ (denoted $v_i \% v_j$), indicates that v_i is an inhibitor of v_j . An ordered pair $(v_i; v_j) \in E$ represents either $v_i \tilde{N} v_j$ or $v_i \% v_j$. Informally, given $v_i \tilde{N} v_j$, we would expect that when v_i has high (low) expression, then v_j has high (low) expression. Additionally, given $v_i \% v_j$, we would expect that when v_i has high (low) expression, then v_j has low (high) expressions.

Definition 2.4. Given a regulatory network $RN = (V; E)$, a source of a node v_j is a node v_i such that $(v_i; v_j) \in E$. A target of v_j is a node v_k such that $(v_j; v_k) \in E$. The set of sources and targets of a node v_j are given by

$$S_{v_j} = \{v_i \mid (v_i; v_j) \in E\} \quad \text{and} \quad T_{v_j} = \{v_k \mid (v_j; v_k) \in E\}.$$

Feedback within a regulatory network refers to a process where a subgraph of the regulatory network has an output that also serves as an input to the same subgraph [30], or when two (or more) subgraphs of the network are connected in such a way that each subgraph influences the other [51]. This feedback can either amplify its own production, called positive feedback, or inhibit it, referred to as negative feedback [30, 51]. A feedback loop is a specific type of feedback where the subgraph forms a cycle. When a feedback loop has an odd number of repressing edges, it forms a negative feedback loop (NFL) and when it has an even number of repressing edges (or none) it forms a positive feedback loop (PFL) [30]. Positive feedback loops are generally more susceptible to disturbances because perturbations to the network state are magnified. Conversely, negative feedback has been shown to promote stability in the network by diminishing the impact of perturbations [4, 23, 51].

2.2 Drosophila Melanogaster

In this section, we first introduce gap genes, maternal gradients, and anterior-posterior patterning determined by these genes and gradients. Next, we describe the gap gene regulatory network. The exact topology of this regulatory network is still under debate. A major goal of this thesis is to compare different models of the gap gene network and evaluate their robustness. We note one element common to all models is that gap gene activity is partially determined by the presence of spatial protein gradients extending along the developmental axis.

2.2.1 Anterior-Posterior (A-P) Patterning in Drosophila Melanogaster

Early development (i.e., development of the embryo) of *D. melanogaster* has three main stages, classified by the number of nuclear divisions called cycles: syncytial cleavage stage, syncytial blastoderm stage, and gastrulation (see Figure 2.1(right)). During the syncytial cleavage stage, the embryo undergoes nine nuclei divisions. By cycle 10, the nuclei have transitioned into the syncytial blastoderm, where they layer the surface of the embryo. Cell walls form between nuclei immediately at the end of the syncytial blastoderm stage (cycle 14A), creating a single layer of cells around the core of the embryo called the cellular blastoderm [12]. Cycle 14B marks the beginning of gastrulation; during this stage the cell layer at the surface of the embryo moves and forms three germ layers that shape the exoskeleton and nervous system (ectoderm), the intestinal organs (endoderm) and the remaining organs (mesoderm) [18, 12].

During the formation of the germ layers, the ectoderm and the mesoderm migrate to the lower side (ventral) of the embryo, forming what is called the germ band. The germ band then undergoes germ band extension, where it extends to the back (posterior) region of the embryo and then wraps around to the top (dorsal) region of the embryo [12].

During germ band extension, distinct segments begin to emerge from the head (anterior) to the posterior of the embryo, where each segment is responsible for distinct portions of the adult fly. Segment determination occurs before gastrulation, the genes responsible for this segmentation were found experimentally by inducing genetic mutations and describing the resulting phenotypes [32, 13, 18]. These experiments resulted in the discovery of a class of so-called gap genes whose knockouts cause entire segments along the anterior-posterior (A-P) axis to be deleted. We focus on the trunk gap genes [18] *hunchback* (*hb*), *giant* (*gt*), *Krüppel* (*Kr*) and *Knirps* (*kni*), which play a central role in the formation of the middle part of the A-P axis, namely between 35% and 75% egg length [25].

The trunk gap genes are, in part, regulated by maternal protein gradients, *Bicoid* (*Bcd*) and *Caudal* (*Cad*) in addition to *Nanos* (*Nos*) and existing maternal *Hb*, the gene product of *hb*. However, *Nos* has only been found to regulate maternal *Hb*, and while maternal *Hb* is important in earlier stages of development, is mostly gone by cellularization [49, 17, 18, 13]. Therefore we will limit our discussion of maternal gradients to *Bcd* and *Cad*.

The interaction of *Bcd* and *Cad* creates opposing gradients from anterior to posterior. During egg development, *Bcd* proteins are concentrated at the anterior of the egg, while *Cad* is uniformly spread throughout the egg. After fertilization, *Bcd* begins to diffuse, creating a gradient of concentration decreasing from the anterior of the embryo to the posterior. A repression of *Bcd* on *Cad* creates a smooth gradient of *Cad* that increases from anterior to posterior [31, 40, 13]. These gradients impact the protein expression patterns of the trunk gap gene proteins, which are regions along the A-P axis where each protein has high or low concentration [31, 32, 13]. Domain boundaries for a particular protein are where the protein expression pattern is transitioning from high concentration to low or vice versa. Domain boundaries sharpen during late-stage development (cycle 14A) of the embryo (see Figure 2.1(right)), a process controlled by trunk gap genes rather than maternal gradients [18]. Trunk gap gene regulation associated with the late-stage segmentation process

Figure 2.1: (left) Early development of the *D. melanogaster* embryo. The numbers indicate cleavage cycle. Figure from [18], reproduced/adapted with permission from jcs.biologists.org. (right) Trunk gap gene expressions during each time class T1 through T8 during cycle 14A, which shows gap gene domain boundaries sharpening. The light areas show the high protein expression for each of the trunk gap genes along the A-P axis. The plots shown in the bottom row shows the protein expression data for time classes T1 through T8. Figure from [18].

can be described by four main regulatory mechanisms, as articulated by [18]:

1. Activation by maternal gradients Bcd and Cad maintain gap gene expression [31] as domain boundaries sharpen.
2. Auto-activation: Many early models of the gap gene network showed that auto-activation of each gene was essential [28], though more recently it has been shown that auto-activation is not strictly essential as models have been able to reproduce the data without auto-regulation [19, 34]. Experimentally, *hb* has the strongest evidence for auto-activation [38, 19, 34].

3. Strong repressive feedback between complementary genes The strongest experimental evidence for late-stage trunk gap gene regulation is between the pair *pbx* and *kni*, and the pair *kr* and *gt* [19]. Both pairs exhibit mutual strong repression with each other, called repressive feedback .
4. Regulation between non-complementary genes There is also experimental evidence that there are interactions between the other genes that are not complementary [19], though the exact type, strength, and potential effect of these interactions have only been examined by mathematical models [18].

2.2.2 The Gap Gene Regulatory Network [48]

Though many models have been shown to faithfully replicate the protein expression of the trunk gap genes [19, 48, 25, 34], we will focus our study on the gap gene network as described in [48] and shown in Figure 2.2. Edges are color-coded according to their source protein. Dotted lines indicate weak regulatory interaction while bold lines indicate stronger regulatory interaction [48, 47]. We call these weak edges and strong edges , respectively. We will make the reasonable assumption that strong edges are more likely to be the dominating regulatory factors in the protein expression levels. The left panel in Figure 2.2 is a regulatory network representation of the gap gene network while on the right is a spatial representation, which shows the extent of protein expression along the A-P axis.

We hypothesize that a sequence of parameter changes representing the impact of Bcd and Cad within a single network is capable of recapitulating the protein expression level data (see Section 3.3 and Chapter 5). There is a solid biological argument for choosing to model maternal gradients as a change in network parameters. In late-stage gap gene regulation, at any point along the A-P axis, the maternal gradients are relatively constant. That is, within a single nucleus there is not a significant change in the level of Bcd and Cad. Therefore Bcd and Cad can be viewed as part of the environmental conditions of the nucleus

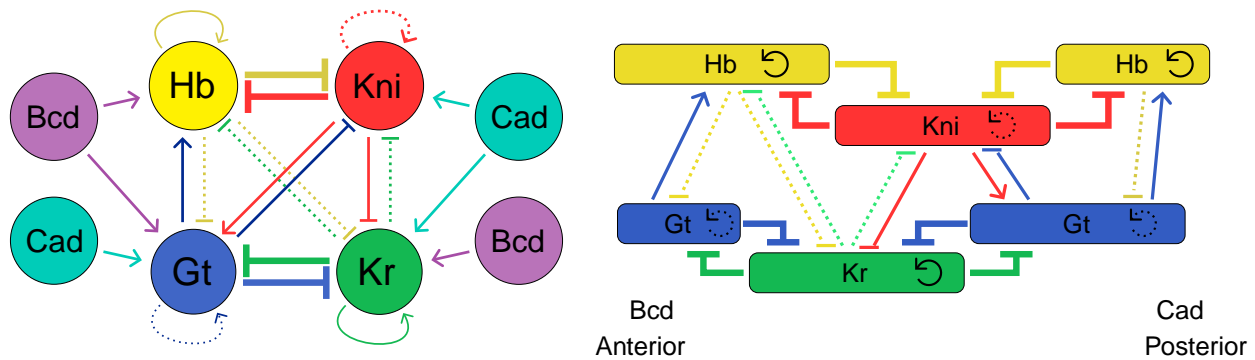


Figure 2.2: Gap gene network. (left) The gap gene network used in [48]. The edge widths depict the strength of the interaction; dotted edges are the weakest interactions and the bold edges are the strongest. (right) Simplified spatial representation of gene regulation from anterior-posterior (A-P) position 35% to 75% for the gap gene network. The violet gradient indicates the concentration of Bcd, and the cyan gradient indicates the concentration of Cad. The horizontal extent of the boxes represents spatial positions with high late-stage protein expression levels. Figure adapted from [48].

that help determine network parameters and not as active participants of the network, see Figure 2.3(C).

2.2.3 ACDC Dynamic Modules of the Gap Gene Network

During their study of the gap gene regulatory network in Figure 2.2(left), Verlet et al. [48] partitioned a slightly reduced version of the spatial representation of the gap gene network shown in Figure 2.2(right) into three subnetworks they described as dynamic modules. According to their definition, a dynamic module of the gap gene regulatory network is a subgroup of the genes that control protein expression in a region of the A-P axis. They postulate that the A-P axis can be split into three regions, each of which has a single gene that does not participate in network dynamics (i.e. is inactive) in that region. They assume that between A-P positions 35-47% (region 1), Kni is inactive, between 49-59% (region 2), Gt is inactive and between 61-75% (region 3), Hb is inactive, shown in Figure 2.3(A). Thus, they create three dynamic modules (Figure 2.3(B)), all isomorphic to the ACDC signaling motif [33], that are active in regions 1, 2 and 3, respectively. Verlet et al. [48] showed that these

loops. We call this network the fully connected network (FullConn), see Figure 2.3. We will evaluate this network using our methods to see if it can faithfully capture the protein expression data, which we describe in Section 4.2.

2.3 Statistical Background

2.3.1 Absorbing Markov Chain

A Markov chain is a discrete stochastic process describing a possible sequence of states where the probability of each state occurring depends only on the current state in the process [35]. Formally, a Markov chain has a finite number of states s_0, s_1, \dots, s_{n-1} for $n \in \mathbb{N}$ where the probability of transitioning from state s_k to state s_{k+1} depends only on the current state s_k and not on previous states s_{k-1}, \dots, s_0 , hence $P(s_{k+1} | s_k; s_{k-1}, \dots, s_0) = P(s_{k+1} | s_k)$. As a consequence, a Markov chain can be represented by a transition matrix W where W_{ij} is given by $W_{ij} = P(s_j | s_i)$ and

$$\sum_{j=0}^{n-1} W_{ij} = 1 \quad (2.1)$$

for each $i = 0, \dots, n-1$ [45]. An absorbing state is a state s_i with $P(s_i | s_i) = 1$ and a transient state is any state that isn't absorbing. An absorbing Markov chain (AMC) is a Markov chain where each state can reach an absorbing state in a finite number of steps [9, 45]. The probabilities of transitioning from transition state s_i to absorbing states s_j in an AMC can be calculated from its transition matrix W [45]. The transition matrix W of an absorbing Markov chain can be transformed to the canonical form

$$W = \begin{pmatrix} Q_{t \times t} & R_{t \times r} \\ 0_{r \times t} & I_{r \times r} \end{pmatrix}$$

where t is the number of transient states, r is the number of absorbing states, $0_{r \times t}$ is the zero matrix and $I_{r \times r}$ is the identity matrix [9, 45]. The matrix $Q_{t \times t}$ describes the transitions between transient states and $R_{t \times r}$ describes the transition from transient states to absorbing

states. The probability of transition from transient states s_i to another transient states s_j in k steps is given by the $p_{ij}^{(k)}$ th entry of Q^k [45]. Additionally, the expected number of times a transient state s_j is visited from s_i is the $p_{ij}^{(k)}$ th entry of

$$N = \sum_{k=0}^{\infty} Q^k = (I - Q)^{-1}; \tag{2.2}$$

where N is called the fundamental matrix of the absorbing Markov chain [45]. Lastly, the $p_{ij}^{(k)}$ th entry of the matrix

$$B = NR \tag{2.3}$$

gives the probability of transitioning from transition state s_i to absorbing states s_j [45].

2.3.2 Multiple Linear Regression

The statistical technique Multiple Linear Regression (MLR) is used to estimate the linear relationship between a quantitative response variable (dependent variable) and two or more predictor variables (independent variables), which can be a combination of quantitative and categorical [16, 20, 21].

When a categorical predictor variable is used in a regression model, it is called a factor, and the individual categories are called levels of the factor [36]. Assume there are p quantitative predictor variables and q categorical predictor variables related to a response variable, with a random sample of n observations (data points). For each observation $i = 1; 2; \dots; n$ we let y_i denote the response variable for, $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})^T \in \mathbb{R}^p$ denote the quantitative predictor values, and C_1, C_2, \dots, C_q denote the categorical predictors, with $c_j = (c_{j1}, c_{j2}, \dots, c_{jq})$ denoting the level of each factor observation belongs to. Note that each c_j can be descriptive rather than numerical.

Definition 2.5. Consider a model with categorical predictor C_k consisting of A_1, \dots, A_{m_k} levels and suppose there are n observations. Then we need $m_k - 1$ indicator variables indicating what level observation $i = 1; 2; \dots; n$ belongs to. Specifically, suppose A_{m_k} is the

baseline level, then for level $l = 1; \dots; m_k - 1$ the indicator variable for observation i is given by

$$I_{kj} = \begin{cases} 0 & \text{if } i \in RA_j \\ 1 & \text{if } i \in PA_j \end{cases}$$

If $I_{kj} = 0$ for all $j = 1; \dots; m_k - 1$ then observation i belongs to the baseline level l_{m_k} [36, 21].

Given the model training set $\{(x_i; c_i; y_i) \mid i = 1; 2; \dots; n\}$, the MLR model assumes the response y_i can be approximated by a hypothesis function $h; (x_i; c_i)$ [36, 21] given by

$$h; (x_i; c_i) = \theta_0 + \sum_{j=1}^p x_{ij} \theta_j + \sum_{k=1}^{m_k-1} I_{kj} \theta_k \tag{2.4}$$

where

1. θ_j and θ_k are the model parameters (usually called coefficients),
2. I_{kj} denotes the indicator variable for level l of the categorical variable C_k , and
3. m_k denotes the number of levels of C_k .

Notice that we assume that the level m_k is the baseline level for all C_k . By assumption, $y_i = h; (x_i; c_i) + \epsilon_i$, where ϵ_i is an error term accounting for the influence on y_i unexplained by the predictor variables x_i and c_i .

Let $I_{k1} = [I_{k11}, I_{k12}, \dots, I_{k1m_k-1}]^T$ and $\theta_k = [\theta_{k1}, \theta_{k2}, \dots, \theta_{k(m_k-1)}]^T$ for $k = 1; \dots; q$. Then the hypothesis function can be written in matrix notation as

$$h = X \theta \tag{2.5}$$

where

$$h = \begin{bmatrix} h(x_1; c_1) \\ h(x_2; c_2) \\ \vdots \\ h(x_n; c_n) \end{bmatrix}; \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_q \end{bmatrix}; \quad \text{and} \quad X = \begin{bmatrix} 1 & x_1^T & I_{11} & I_{21} & \dots & I_{q1} \\ 1 & x_2^T & I_{12} & I_{22} & \dots & I_{q2} \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_n^T & I_{1n} & I_{2n} & \dots & I_{qn} \end{bmatrix}$$

Note that X is called the design matrix. In general, the β that solves equation (2.5) is not unique. Instead, the goal is to find β that minimizes the ordinary least squares (OLS) equation

$$J(\beta) = \sum_{i=1}^n e_i^2 = (y - X\beta)^T (y - X\beta) \quad (2.6)$$

where $e_i = y_i - \hat{y}_i = y_i - \sum_{j=1}^p x_{ij}\beta_j$ and $y = [y_1, y_2, \dots, y_n]^T$ [21, 20]. Note that e_i is called the residual of observation i . The solution of equation (2.6) is given by solving the normal equations

$$X^T X \beta = X^T y$$

which can be derived from $\nabla J(\beta) = 0$. This minimization problem has a unique solution, provided that the columns of the design matrix X are linearly independent [21].

Once the parameters are determined, the hypothesis function can be used for prediction. However, the parameters can also be used to understand the relationship between the predictor variables and the response. The parameters can be interpreted as follows [36]; β_0 is the estimated mean of the response when $x_j = 0$ and i belong to all baseline levels, giving

$$\beta_0 = \sum_{j=1}^p x_{ij} \beta_j = 0 \quad \text{and} \quad \beta_k = \sum_{k=1}^{m_k-1} I_k \beta_k = 0:$$

Each β_j for $j = 1, \dots, p$ is the unit change in the estimated mean of the response for each one unit increase in x_{ij} . Lastly, each β_k is the difference in the estimated mean of the response when an observation belongs to the k th level of C_k for $k = 1, \dots, q$, after accounting for other variables in the model.

For a simple example, suppose we have the training data as seen in Figure 2.4(left) composed of a single quantitative variable x_1 and a single categorical variable C_1 which has three levels (shown by color). Looking at the data, we have the following observations. First, it appears that there is a linear relation between the response and the predictor variable. Second, it appears that the level of C_1 that an observation belongs to has an impact on the

value of the response. Using a MLR model on this data can help determine if the second observation is correct, assuming the first observation is true. Assume level 3 (shown in red) is the baseline of C_1 , then the model is given by

$$h(\mathbf{x}_i; \boldsymbol{\beta}) = \beta_0 + \beta_1 x_{i1} + \beta_{11} I_{11}(\mathbf{x}_i) + \beta_{12} I_{12}(\mathbf{x}_i)$$

with parameter vector $\boldsymbol{\beta} = [\beta_0, \beta_1, \beta_{11}, \beta_{12}]^T$. Figure 2.4(right) shows a depiction of the hypothesized linear relationship between the predictors and the response. Note how the baseline level's linear fit has an intercept of β_0 . Recall that β_0 is the mean of the response when $x_1 = 0$ and i is in the baseline level. The remaining lines depict the change in the mean of the response when an observation belongs to either level 1 or 2. Furthermore, notice how these linear fits are parallel linear lines with a slope of β_1 , this slope shows how the response changes as x_1 increases a single unit, regardless of what group an observation belongs to. This model is sometimes called the parallel lines model.

Figure 2.4: Multiple linear regression example on generated sample data (left), with one quantitative variable x_1 and one categorical variable with 3 levels shown in red (baseline), blue and green (right).

Remark 2.1. There are other forms of the MLR model hypothesis equation, such as equations that include interaction terms between the groups that allow for a change in the slope for different groups [36, 21]. However, this is out of the scope of this thesis.

Before getting into model assumptions, a few terms need to be introduced. The mean squared error is given by

$$s^2 = \frac{J_{p,q}}{n - p}$$

where n is the number of observations in the training set and p is the length of β . Let $y = (y_1, y_2, \dots, y_n)^T$ be the response variable and let $\mu = \frac{1}{n} \sum_{i=1}^n y_i$ denote the true response mean.

The coefficient of determination R^2 [21] is given by

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \mu)^2}{\sum_{i=1}^n (y_i - \mu)^2}$$

Note that R^2 is a measure of how well y is explained by the predictor variables in the model.

An issue that can arise while training an MLR model is when an observation is overly influential. Consider $H = X(X^T X)^{-1} X^T$, the diagonal elements h_{ii} of H for $i = 1, 2, \dots, n$ is called the leverage of observation i [21]. Leverage is a measure of how far away an observation is from the rest of the observations. Specifically, when a point that has high leverage is removed from the dataset there is a potential that it will cause large changes to the parameter. Cook's distance measures the effect on the parameter when an observation is deleted. The Cook's distance of observations is given by

$$D_i = \frac{r_i^2}{s^2} \frac{h_{ii}}{p(1 - h_{ii})^2}$$

where $r_i = \frac{y_i - \hat{y}_i}{s \sqrt{1 - h_{ii}}}$ are the standardized residuals [21]. Observations with both high leverage and a high Cook's distance are considered to be influential. An observation with leverage over $\frac{2p}{n}$ is generally considered a high-leverage observation [21, 16]. An observation with a Cook's distance between 0.5 and 1 is considered to have moderate influence, and anything greater than 1 is considered high influence [21, 16].

Recall that to guarantee a unique solution that minimizes the OLS equation, the columns of the design matrix X must be linearly independent. However, having columns that are close to being dependent is also a problem because it can lead to large coefficients in the model. The term multicollinearity is used to describe a situation where the columns

of X are either linearly dependent or close to being linearly dependent [21]. This can be tested using a measure termed the variance inflation factor (VIF), which measures how close to linearly dependent the columns of X are. Let $X_1, \dots, X_{|X|}$ denote the columns of X , the VIF of column X_j is given by

$$f_j = \frac{1}{1 - R_j^2};$$

for $j = 1, \dots, |X|$, where R_j^2 is the coefficient of determination using X_j as the dependent variable and $X_1, \dots, X_{j-1}, X_{j+1}, \dots, X_{|X|}$ as the independent variables [21]. X_j is orthogonal to the other columns of X , then $R_j^2 = 0$ implying $f_j = 1$. When X_j has a strong linear relationship with at least one of the remaining columns of X then $R_j^2 = 1$ showing that the stronger the linear relationship, the larger f_j becomes [21, 20]. It is generally accepted that $f_j = 5$ indicates no issue with multicollinearity between X_j and the remaining columns of X [20].

In a MLR model, it is assumed that there are no influential observations in the training set and that multicollinearity is not an issue among the predictor variables. In addition to these, the following are also assumptions on the training set [21, 16, 20, 36]:

- ^ Linearity: the relationship between the predictor variables and the response variable is linear.
- ^ Equal Variance: the size of the error in the prediction doesn't change across the values of the independent variable.
- ^ Normality: the residuals follow a normal distribution.
- ^ Independence of observations: the values for the predictor variables for each observation are obtained independently of each other.

In practical applications, the MLR model assumptions only need to be approximately true, we now discuss verification tools and guidelines for assessing if the training set

approximately satisfies the assumptions. Linearity can be graphically verified by looking at each predictor variable plotted against the response variable on a scatter plot and observing if the relationship is approximately linear. Another way to test this assumption is to plot the fitted values \hat{y}_i against the residuals $\hat{\epsilon}_i$. If the residuals appear to be randomly spread out about 0, this suggests that the assumption that the relationship is linear is reasonable [21, 16, 20, 36].

Equal variance can be verified using a plot of the fitted values \hat{y}_i and the square root of the standardized residuals $\hat{\epsilon}_i$ (called the scale-location plot). In this plot, an evenly spaced band following some identified curve indicates that the equal variance assumption is approximately satisfied [21, 16, 20, 36]. Furthermore, a normal quantile-quantile (QQ) plot serves as a method to evaluate if the residuals are normally distributed. This is achieved by plotting the quantiles of the residuals from the training set against the theoretical quantiles expected for normally distributed residuals. If the plotted points generally follow the diagonal line, it is inferred that the training set residuals follow a normal distribution [21, 16, 36].

Lastly, a MLR model requires that the observations are independent from each other [36]. As an example of data that could have an independence issue, suppose data are collected on temperature at different weather stations. Further, suppose two of those stations, say A and B, are located in the same city. Then the readings at station A are likely to be similar to those of station B, indicating these observations are dependent because readings at one station can be predicted from the other. Independence of observations can be verified by considering the sources of the data, or by inspecting a plot of the residuals against some variable that might be related to the dependency of observations, such as time, location, or even the order in which observations were collected. Any patterns or trends in the plot can suggest a violation of the independence assumption.

2.4 DSGRN

In this section, we discuss a modeling approach called DSGRN (Dynamic Signatures Generated by Regulatory Networks) [7] that captures networks dynamics across global parameter space. The structures that will be especially important are the parameter graph constructed from factor graphs (Section 2.4.2) and the Morse graph capturing the dynamics at each DSGRN parameter (Section 2.4.4).

2.4.1 Switching Systems

We associate to a regulatory network $RN = (V, E, \sigma)$ with $|V| = M$ a system of M ordinary differential equations (ODEs) with piecewise constant nonlinearities called a switching system [14, 15, 43, 44, 42, 8, 39]. With a slight abuse of notation in the interest of clarity, we use v_j to denote either a node in V or the corresponding variable in a dynamical system that evolves according to

$$\dot{v}_j = -\gamma_j v_j + \sum_{i \in V} \sigma_{j,i} v_i \quad j = 1, \dots, M \quad (2.7)$$

where $\gamma_j > 0$ is the decay rate of v_j and $\sum_{i \in V} \sigma_{j,i} v_i$ is a product of sums of step functions $\sigma_{j,i} v_i$ for each $v_i \in V$ given by

$$\sigma_{j,i} v_i = \begin{cases} l_{j,i} & \text{if } v_i \geq \theta_{j,i} \\ u_{j,i} & \text{if } v_i < \theta_{j,i} \end{cases} \quad (2.8)$$

if $v_i \geq \theta_{j,i}$ and

$$\sigma_{j,i} v_i = \begin{cases} l_{j,i} & \text{if } v_i < \theta_{j,i} \\ u_{j,i} & \text{if } v_i \geq \theta_{j,i} \end{cases} \quad (2.9)$$

if $v_i < \theta_{j,i}$. Here $l_{j,i}$ and $u_{j,i}$ are called the lower (low) and upper (high) level of effect of node v_i on node v_j , where $0 \leq l_{j,i} \leq u_{j,i}$. The threshold $\theta_{j,i}$ for node v_i is where the effect on target v_j of the regulator node v_i switches. We assume that the values of $\theta_{j,i}$ for

any node v_i are distinct. Suppose there are K sources of v_j , $|S_{p_j q}| = K$, where the nodes $v_{i_1}; \dots; v_{i_k}$ are activators of v_j and the nodes $v_{i_{k+1}}; \dots; v_{i_K}$ are inhibitors of v_j . Then for the computations in this thesis we choose the expression

$$p_j v_j q = p + p_{v_{i_1} q} \dots + p_{v_{i_k} q} - p_{v_{i_{k+1}} q} - \dots - p_{v_{i_K} q} \quad (2.10)$$

This form, often used in switching systems [14, 43, 42], was motivated by the fact that transcriptional activators often act additively and that the transcriptional repressors physically block transcription initiation. This choice is not conceptually necessary but is currently implemented in the DSGRN software [11]. See Figure 2.5(a,b) for an example of an RN and its associated switching system.

2.4.2 Factor Graph and DSGRN Parameter Graph

The values $l_{j,i}; u_{j,i}; u_{j,i}$ are non-negative parameters of system (2.7), where we assume decay rates of 1 for simplicity. Traditionally, to characterize the behavior of the ODE system over parameter space, a (necessarily sparse) parameter sampling would be performed. DSGRN takes a different approach and divides parameter space into a finite number of regions defined by inequalities, and evaluates coarse but informative signatures of dynamic behaviors of the network that are invariant within each region [7]. Since the number of regions is finite, it is in principle possible to compute these coarse signatures over all of parameter space for a switching system associated RN, although the number of regions grows combinatorially and exhaustive computations become rapidly intractable. In this section, we introduce the inequalities that define DSGRN parameter regions and arrange them into a parameter graph that reflects region adjacency in the parameter space.

To do so, we define order parameters and logic parameters. For a node v with $|T_{p v q}|$ target nodes, and therefore $|T_{p v q}|$ thresholds, one for each $v_k \in T_{p v q}$, an order parameter defines an ordering of these thresholds. A logic parameter defines how the finite collection of possible inputs to node v is related to the $|T_{p v q}|$ thresholds of v .

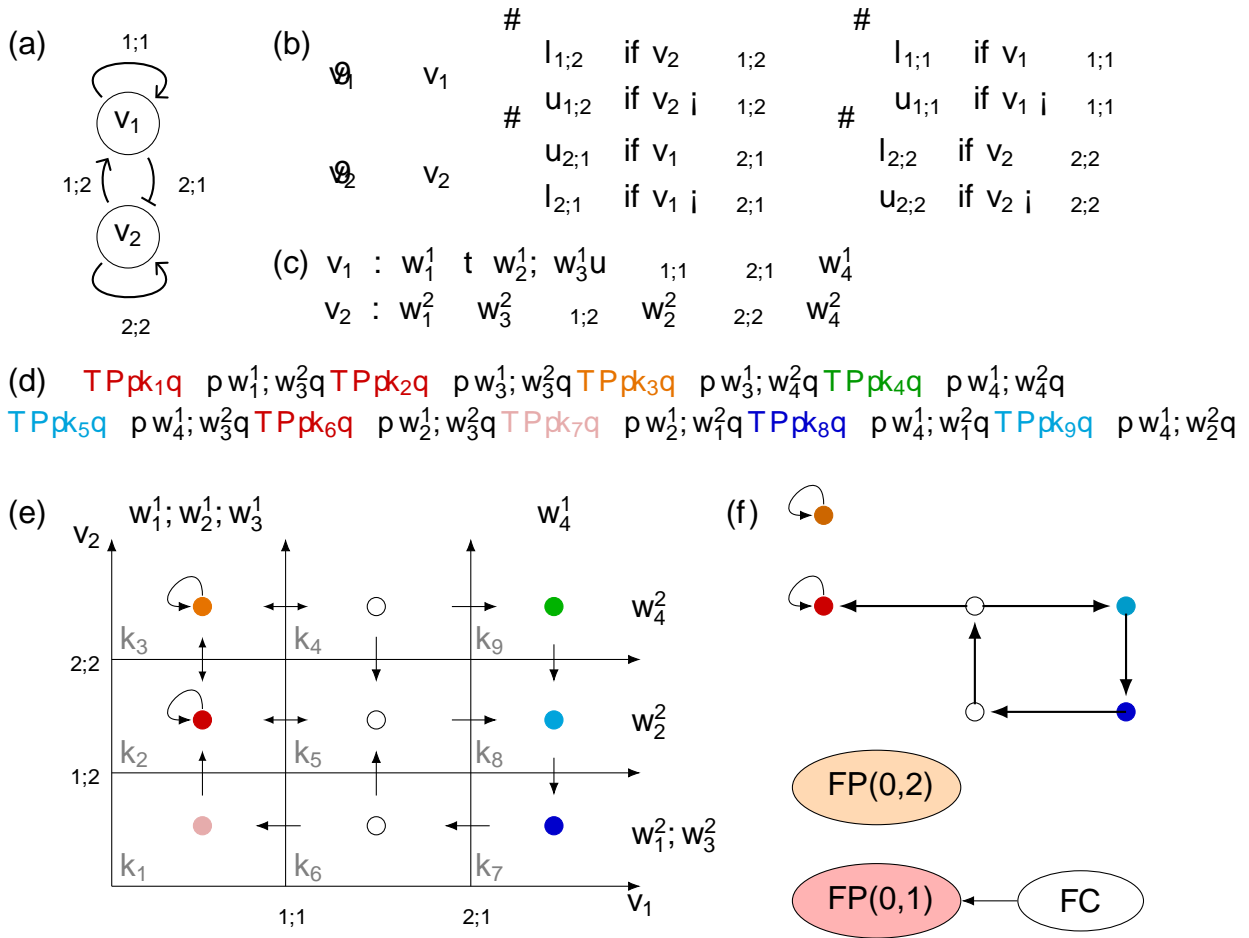


Figure 2.5: (a) The RN . (b) ODEs with decay rates λ_1 λ_2 λ_1 associated with (a). (c) A choice of DSGRN parameter where w_1^i p $l_{i;1}; l_{i;2} q$ w_2^i p $l_{i;1}; u_{i;2} q$ w_3^i p $u_{i;1}; l_{i;2} q$ and w_4^i p $u_{i;1}; u_{i;2} q$ for $i = 1; 2$, see Section 2.4.2 for more details. (d) List of target points for each domain k_i of the phase space in (e). Note the colors of the TR() match the color of the vertex of the domain where that target point falls.(e) Phase space decomposition into the nine domains by thresholds $1;1$; $1;2$; $2;2$ and $2;1$. Each domain is represented by a circular vertex inside the domain. Arrows are the depiction of the direction of trajectories of a switching ODE system model in (b). The choice of DSGRN parameter is depicted above and to the right of phase space. For example, we have w_4^1 $i = 2;1$, since domains k_7 , k_8 and k_9 are all above $2;1$ we write w_4^1 above these. The vertices and arrows form the state transition graph. See Section 2.4.3 for details. (f) The Morse graph (below) is associated with the state transition graph in (e) and the strongly path connected components, or Morse nodes, are associated with each node of the Morse graph (above), where the strongly connected path components are associated to domains k_2 ; k_3 ; k_5 ; k_6 ; k_7 and k_8 in phase space and the edges are reachability conditions.

Definition 2.6. Let $v_j \in PV$ be a node in RN with source nodes $S_{pv_j} = \{v_{s_1}, \dots, v_{s_K}\}$ and target nodes $T_{pv_j} = \{v_{t_1}, \dots, v_{t_L}\}$. The thresholds associated with v_j are $\theta_j = \{\theta_{i_1,j}, \dots, \theta_{i_T,j}\}$. An order parameter for v_j is a bijective map $\sigma_j : \theta_j \rightarrow \tilde{N} = \{0, 1, \dots, |T_{pv_j}| - 1\}$ that induces a total ordering of the thresholds associated with v_j . We call O_j the set of all order parameters for vertex v_j .

Let

$$R_j = \{ \langle l_{j;s_1}; u_{j;s_1} \rangle, \dots, \langle l_{j;s_K}; u_{j;s_K} \rangle \}$$

be a lattice of inputs to the node v_j under the product order induced by

$$\langle l_{j;s_k}; u_{j;s_k} \rangle \leq \langle l_{j;s_{k'}}; u_{j;s_{k'}} \rangle \text{ for all } s_k, s_{k'} \in S_{pv_j} \quad (2.11)$$

That is, we will write $w \leq w' \in R_j$ whenever, for $w = \langle p_{a_1}; \dots; p_{a_K} \rangle$ and $w' = \langle p_{a'_1}; \dots; p_{a'_K} \rangle$ we have $a_k \leq a'_k$ for all $k = 1, \dots, K$ and at least one of the inequalities is strict. Let $X_j = \{0, 1, \dots, |T_{pv_j}| - 1\}$ be the set of $|T_{pv_j}| - 1$ integers that enumerates the intervals between the thresholds. A logic parameter λ_j for v_j is a map $\lambda_j : R_j \rightarrow X_j$, which satisfies

$$w \leq w' \implies \lambda_j(w) \leq \lambda_j(w')$$

i.e. it is monotone. We call L_j the set of all logic parameters for vertex v_j . A factor parameter for a node $v_j \in PV$ is a pair $p_j : p_j \in R_j \times L_j \rightarrow O_j$.

The set R_j contains all the possible input values into a node v_j and the map λ_j inserts the inputs between the thresholds. If $\lambda_j(w) = m$ then we say w is above m thresholds. We chose to reuse the relation symbol \leq on R_j to facilitate the following simplification of notation: we will write $w \leq_{ij}$ when $\lambda_j(w) \leq i$ although the spaces R_j and X_j are not strictly comparable.

See Table 2.1 for example parameters for node v_1 in Figure 2.5, which has two in-edges and two out-edges. The lattice of inputs

$$R_1 = \{ \langle l_{1;1}; u_{1;1} \rangle, \langle l_{1;2}; u_{1;2} \rangle, \langle l_{1;1}; l_{1;2}; u_{1;1}; u_{1;2} \rangle, \langle l_{1;1}; l_{1;2}; u_{1;1}; u_{1;2} \rangle, \langle l_{1;1}; l_{1;2}; u_{1;1}; u_{1;2} \rangle, \langle l_{1;1}; l_{1;2}; u_{1;1}; u_{1;2} \rangle \}$$

logic parameter					logic parameter inequality description								
$1pw_1^1q$	$0;$	$1pw_2^1q$	$0;$	$1pw_3^1q$	$0;$	$1pw_4^1q$	0	w_1^1	t	$w_2^1; w_3^1u$	w_4^1	$1;1$	$1;2$
$1pw_1^1q$	$0;$	$1pw_2^1q$	$0;$	$1pw_3^1q$	$1;$	$1pw_4^1q$	2	w_1^1	w_2^1	$1;1$	w_3^1	$1;2$	w_4^1
$1pw_1^1q$	$0;$	$1pw_2^1q$	$2;$	$1pw_3^1q$	$1;$	$1pw_4^1q$	2	w_1^1	$1;1$	w_3^1	$1;2$	w_2^1	w_4^1

Table 2.1: Logic parameter examples (left) and corresponding inequality descriptions (right) for a network node with two in-edges and two out-edges and order parameter $p_{1;1}q$ $0;$ $1p_{1;2}q$ 1 .

is partially ordered $p_{1;1}; l_{1;2}q$ $tp_{1;1}; u_{1;2}q$ $pu_{1;1}; l_{1;2}qu$ $pu_{1;1}; u_{1;2}q$ with respect to the product order. The out-edges have thresholds $s_{1;1}$ and $s_{1;2}$ with the set of order parameters O_1 consisting of two functions

$$O_1 = \{tp_{1;1}q, 1p_{1;2}q\}$$

while the set of logic parameters is the set of functions $s_i : R_1 \times \tilde{N} \rightarrow [0; 1; 2]u$. Select without loss one of the two order parameters $s_1 : \{1\}$ which we interpret as $s_{1;1}$ $s_{1;2}$. Using the notation $w_1^1 = p_{1;1}; l_{1;2}q$ $w_2^1 = p_{1;1}; u_{1;2}q$ $w_3^1 = p_{u_{1;1}; l_{1;2}q}$ and $w_4^1 = p_{u_{1;1}; u_{1;2}q}$ we list in Table 2.1 three logic parameters with the corresponding description in terms of inequalities. While Table 2.1 only shows 3 logic parameters for our example, there are 20 in total. Hence, $|L_1 \cup O_1| = 40$, showing v_1 has 40 factor parameters.

Remark 2.2. In the special case where $|\Gamma(v_j)| = 0$, i.e. node v_j has no out-edges, DSGRN assumes that v_j still can attain high and low levels of expression. To implement this, a "ghost" threshold is assigned, and the parameters for v_j are taken to be the same as if $|\Gamma(v_j)| = 1$.

The set of factor parameters for a network node can be represented as a graph.

Definition 2.7. Given a RN $(p; V; E; q)$ the factor graph $(F_j; p; V_j; E_j; q)$ for a regulatory node $v_j \in V$ is an undirected graph with a node $v_j \in V_j$ for each factor parameter of v_j and edges between nodes whenever there is a single inequality change between two factor

parameters; i.e., there is an edge between $p_j^i; q$ and $p_j^k; q$ if exactly one of the following is satisfied:

1. (Logical adjacency) $\sum_j p_j^i; q \neq \sum_j p_j^k; q$ for $\forall P_j$ except for exactly one P_j , in which case

$$\sum_j p_j^i; q - \sum_j p_j^k; q = 1;$$

and $\sum_j p_{i_m;j}^i; q = \sum_j p_{i_m;j}^k; q$ for all $i_m; j \in P_j$, or

2. (Order adjacency) $\sum_j p_j^i; q \neq \sum_j p_j^k; q$ and there exists exactly one set of integers $s; m; n$ such that

$$\sum_j p_j^i; q = s - 1 \text{ for all } \forall P_j, \sum_j p_{i_m;j}^i; q = \sum_j p_{i_m;j}^k; q \text{ for } \forall m; n \text{ and}$$

$$\sum_j p_{i_m;j}^i; q = s; \sum_j p_{i_n;j}^i; q = s - 1$$

$$\sum_j p_{i_m;j}^k; q = s - 1; \sum_j p_{i_n;j}^k; q = s;$$

The factor graph of node v_1 from Figure 2.5 can be seen in Figure 2.6. Note that this graph has all 40 factor parameters, where the order parameter is depicted above the nodes and the logic parameter is encoded in the labels, where a label abc corresponds to the logic parameter with $p_{w_1}^a; q$, $p_{w_2}^b; q$, $p_{w_3}^c; q$, and $p_{w_4}^d; q$. Logical adjacencies are shown as black edges and order adjacencies are shown as red edges.

Order adjacencies exist only between subfactor graphs, or isomorphic subgraphs of a factor graph that contain only logical adjacencies. For a regulatory node P_j , the order parameters $\sum_j p_{O_j}^i; q$ are related by a group of permutations $|T_{P_j; q}|$, that permute threshold labels. As a consequence, for each factor parameter $p_{P_j}^i; q$ with a threshold order \sum_j there are $|T_{P_j; q}|$ parameters $p_{P_j}^i; q$, where threshold labels are permuted by. Therefore, each factor graph contains a collection of $|T_{P_j; q}|$ subfactor graphs. An example of this can be seen in the Figure 2.6 factor graph, which has two subfactor graphs.

A DSGRN parameter is the choice of one factor parameter for each P_j . Figure 2.5(c) shows an example of a DSGRN parameter for the RN shown in (a). Figure 2.7

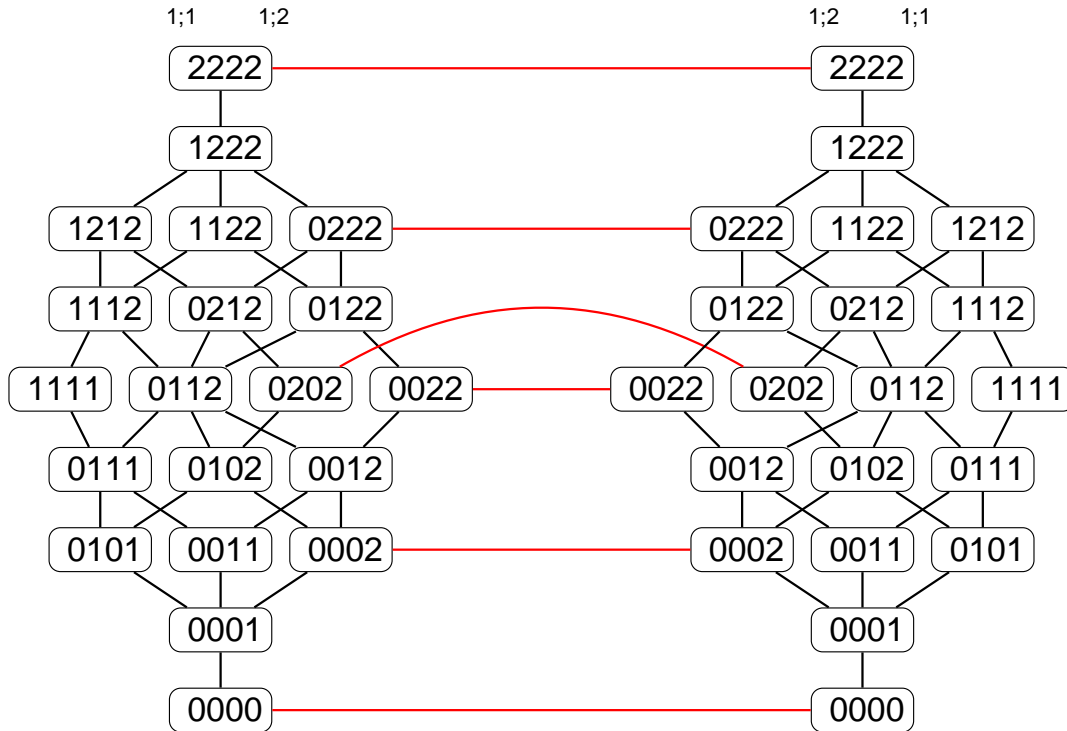


Figure 2.6: Factor graph for a node v_1 from Figure 2.5. Each node labeled $abcd$ represents a logic parameter, with $a = p_1, b = p_2, c = p_3$ and $d = p_4$. The left-hand side of the graph has factor parameters associated with the order parameters $(1;1, 1;2)$ while the right-hand side is associated with $(1;2, 1;1)$. Black edges show logical adjacencies and red edges show order adjacencies.

shows all nine DSGRN parameters for a two node and two edge RN as labels on graph.

Definition 2.8. Let $RN = (V; E; q)$ be a regulatory network with $|V| = M$ and let $F_j = (V_j; E_j; q)$ denote the factor graphs of each $v_j \in V$. The DSGRN parameter graph is an undirected graph $PG = (P; C; q)$ constructed from DSGRN parameters as follows. The vertex set P consists of a node for each DSGRN parameter, i.e.,

$$P = \{ p_j : j = 1, \dots, M \}$$

Additionally, there exists an edge $(p_i, p_j) \in C$ if and only if there exists exactly one $v_j \in V$ such that $p_i, q \in E_j$ and $p_j \in q$ otherwise. In other words, there exists an adjacent change in inequalities in exactly one factor parameter graph.

Consider the RN from Figure 2.5(a) and the factor graph in Figure 2.6 for node v_1 . Since v_2 also has two in-edges and two-edges, then it has a factor graph that is isomorphic to the factor graph for v_1 . The parameter graph for this RN is constructed by taking the product of the factor graph for v_1 with the factor graph of v_2 . Then this parameter graph has a total of 1600 nodes. While 1600 appears to be a large number of parameter nodes, it represents a finite decomposition of $R^{3|E|} = R^{12}$, which then permits exhaustive exploration of parameter space. A fully constructed DSGRN parameter graph for a two-node and two-edge network can be seen in Figure 2.7.

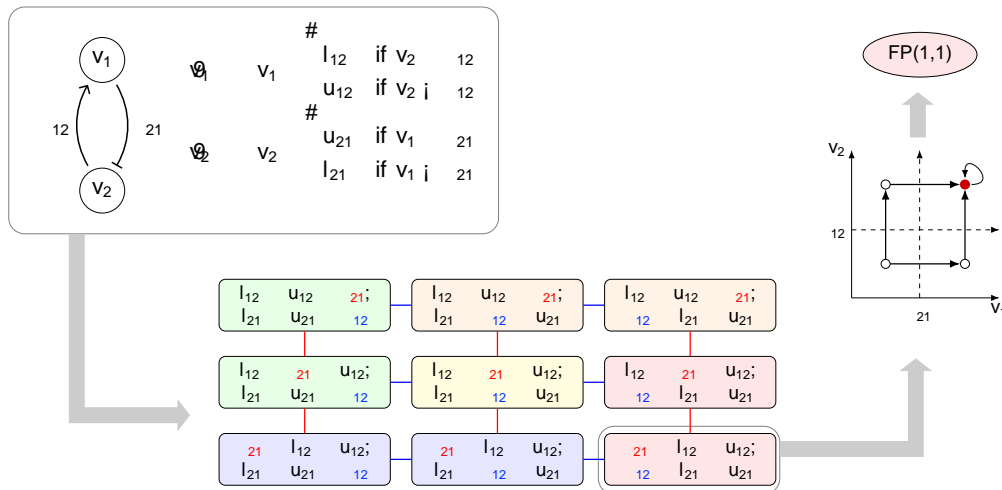


Figure 2.7: Basic overview of DSGRN structure for a two node and two edge regulatory network as shown in the upper left box, along with the system of ODEs used to model network dynamics. The undirected graph is the full DSGRN parameter graph for the example regulatory network, the node labels are the factor parameters for node v_1 (top line) and v_2 (bottom line). Node colors indicate DSGRN parameters that resulted in the same Morse graph. Edges colored red (vertical) are indicating that the factor parameter for v_1 has an adjacent inequality change, while the blue (horizontal) edges are indicating the factor parameter for v_2 has an adjacent inequality change. On the right is a depiction of the state transition graph and Morse graph for the bottom right DSGRN parameter graph node.

2.4.3 State Transition Graph (STG)

Given a regulatory network $RN = (V, E, \theta)$ the set of thresholds θ_j for $v_j \in V$ divide the interval $[0, 8q]$ into $|T_{\theta_j}| + 1$ intervals, namely $[0, \theta_{1,j}]; [\theta_{1,j}, \theta_{2,j}]; \dots; [\theta_{|T_{\theta_j}|}, 8q]$,

described by the set $K_j = \{x \in \mathbb{R}^M \mid v_j(x) \geq \theta_j\}$. The collection of thresholds $\theta = (\theta_1, \dots, \theta_M)$ divides \mathbb{R}^M into a finite number of M -dimensional rectangles called domains. Let \mathcal{K} denote the collection of all such domains. This collection is represented by the M -tuples of integers $X = (x_1, \dots, x_M)$ via a bijection $\gamma: \mathcal{K} \rightarrow \mathbb{Z}^M$. For example, domain k_1 in the lower left of Figure 2.5(e) has label $\gamma(k_1) = (0, 0)$ since both v_1 and v_2 exceed 0 thresholds each. The arrows between the domains represent the direction of trajectories of a switching ODE system model that is consistent with network structure.

The set X is the set of nodes of the state transition graph (STG). The directed edges between the nodes in STG indicate the direction of flow between neighboring (non-diagonal) domains. As we will now show, the edge directions are uniquely determined by the choice of DSGRN parameter. We first define the domain target points.

Definition 2.9. Given a regulatory network $RN = (V, E, \sigma)$ with $|V| = N$, let \mathcal{P} be the associated set of DSGRN parameters. Fix a parameter $p \in \mathcal{P}$. For each domain $k \in \mathcal{K}$, the switching function $v_j(x)$ for node $v_j \in V$ is constant for all $x \in k$. Let $\gamma(k) = (p_1, \dots, p_N)$ denote the vector of these values. Note that the flow in each domain k converges to a point determined by

$$Q_k = -\Lambda^{-1} \gamma(k) \quad (2.12)$$

Here Λ is a diagonal matrix, with decay rates λ_j as its diagonal entries. Then a target point for k is

$$T(p, k) = -\Lambda^{-1} \gamma(k) \quad (2.13)$$

When $T(p, k) \in k$, we call k an attracting domain.

We now translate the map $\gamma: \mathcal{K} \rightarrow \mathbb{Z}^M$ which is the map $\mathcal{K} \rightarrow \mathbb{Z}^M$ to a map on the space X . For $p \in \mathcal{P}$, the map $F^0: X \rightarrow \mathbb{Z}^M$ defined by

$$F^0(x; p) = y \quad \text{where } y = T(p, \gamma(x))$$

That is, $y \in P(X)$ is an integer signature of the domain where the target point of domain $1(p, q)$ lies. We are ready to define a multi-valued map F on X that gives rise to the STG.

Definition 2.10. The multi-valued map $F : X \rightarrow P(\tilde{N}(X))$ is generated by F^0 and defined by

^ If $F^0(p, q) = x$ then $F(p, q) = \{x\}$.

^ For any component $j \in \{1, \dots, N\}$ and $P = \{1, 1\}$ satisfying $F_j^0(p, q) = x_j$ the state

$$x_j = x_j; \quad x_i = x_i \text{ for } i \neq j$$

satisfies $x \in F(p, q)$.

Note that $x \in P(X)$ is a fixed point of F if and only if x is a fixed point of F^0 . The multivalued map F can be represented as a STG, see Figure 2.5(e). As an example, we construct a STG for the regulatory network in Figure 2.5(a) at a particular DSGRN parameter. Suppose $\tau_1 = \tau_2 = 1$ and consider the DSGRN parameter $\rho = (\rho_1, \rho_2)$ with $\rho_1 = \rho_{1,1}, \rho_2 = \rho_{2,2}$. Assume that order parameters are

$$\begin{aligned} \rho_{1,1} &= 0; & \rho_{2,1} &= 1 \\ \rho_{1,2} &= 0; & \rho_{2,2} &= 1; \end{aligned}$$

and logic parameters are

$$\begin{aligned} \tau_1 : & l_{1,1} = l_{1,2} = \tau, \quad u_{1,1} = l_{1,2}; \quad l_{1,1} = u_{1,2}, \quad \tau_{1,1} = \tau_{2,1} = u_{1,1} = u_{1,2} \\ \tau_2 : & l_{2,2} = l_{2,1} = u_{2,2} = l_{2,1} = \tau_{1,2} = l_{2,2} = u_{2,1}, \quad \tau_{2,2} = u_{2,2} = u_{2,1}. \end{aligned}$$

This choice of parameter ρ determines the STG in Figure 2.5(e), which is superimposed on phase space. For example, consider the domain k_1 , which is the bottom left domain. All the regulatory nodes in domain k_1 are below their thresholds, thus the ordinary differential equations in this domain are

$$\dot{v}_1 = \rho_{1,2} - u_{1,1} \dot{v}_2 = \rho_{2,2} - u_{2,1}$$

with $TP_{k_1,q} = p_{1,2} \cup \{l_{1,1}; l_{2,2}u_{2,1}\}$. Notice the choice of DSGRN parameter ϵ implies that the value $l_{1,2} = l_{1,1} = \epsilon_{1,1}$, while $l_{1,2} = l_{2,2}u_{2,1} = \epsilon_{2,2}$. Therefore the target point $TP_{k_1,q}$ is in domain k_2 . Repeating this for every domain $k_j; j = 1; \dots; 9$ we construct the STG in 2.5(e).

2.4.4 Morse Graphs

A Morse graph is a compact description of the global dynamics of a regulatory network RN at a specific parameter node in PG that is derived from the STG. The most important features to notice of a STG are that (1) there can be single cells that are attracting, corresponding to the presence of a stable equilibrium, and (2) stable or unstable cyclic behavior can be identified. A Morse graph is a summary of this recurrent behavior in the STG, as described by the arrangement of strongly connected path components, see Figure 2.5(f).

We summarize the following definition from [7], using graph theory concepts defined in Section 2.1. The Morse decomposition $MD(p)$ of a STG for $p \in P$ is the set of all strongly connected path components of the STG. Consider any two strongly connected path components $s_1; s_2 \in MD(p)$. If there is a path in the STG from s_2 to s_1 then we say $s_1 \succ s_2$, defining a partial order \succ on $MD(p)$. The Morse graph of the STG, denoted $MG(p)$, is the Hasse diagram of $(MD(p), \succ)$, and the vertices of $MG(p)$ are called Morse nodes.

In order for the Morse graph to provide interpretable information, we label each Morse node in a way that suggests the dynamics associated with the underlying strongly connected path component of the STG. The notation $FP(w)$ is used to label a Morse node where the corresponding strongly connected path component consists of a single attracting domain $k \in K$ with label $w = p_{k,q} \in PX$. For example, in Figure 2.5(e) we see that the domain k_3 is an attracting region with label $p_{0;2,q}$ and thus the corresponding Morse node will be labeled $FP(p_{0;2,q})$.

The full cycle label (FC) annotates Morse nodes where there is a closed path

$t \in \{k^0, k^2, \dots, k^m, k^0\}$ in K where each edge $k^i \tilde{N} k^{i-1} \pmod{m-1}$ follows the directed edge in STG and crosses a threshold for each node. For example, the STG in Figure 2.5(e) gives rise to a Morse graph with two Morse nodes labeled FP, as well as a full cycle FC that has a path to one of the fixed points (Figure 2.5(f)). The full cycle represents the path in phase space $k_6 \tilde{N} k_5 \tilde{N} k_8 \tilde{N} k_7 \tilde{N} k_6$.

Definition 2.11. The leaves of the Morse graph, i.e. the Morse nodes with no out-edges, are called stable Morse nodes. All others are unstable Morse nodes. A monostable Morse graph is a Morse graph containing a single stable Morse node. A monostable fixed point is the unique stable Morse node in a monostable Morse graph that has an FP annotation.

CHAPTER THREE

SPATIAL MODELING USING DSGRN

DSGRN is inherently suited to systems of ordinary differential equations and not to partial differential equations. However, we can approximate the effect of temporally constant yet spatially varying external variables on a dynamical system via a directed sequence of parameter changes in the system. The goal of this section is to introduce the necessary rigor for this modeling framework. This procedure necessitates a re-imagining of a factor graph as a graded poset (see Theorem 2). The ranks of the graded poset are used to define factor graph layers that impose an unambiguous direction of flow through the factor graph, allowing for external variables to the dynamical system to be modeled as monotone changes in the factor graph.

3.1 Factor Graph Layers

In this section, we define a partial order on the factor graph by first defining it on every subfactor graph. Recall from Section 2.4 that $\tau = \{i_1; i_2; \dots; i_{|T_{PV_j}|}\} \cup \{u\}$ is the collection of thresholds of nodes $v_j \in PV$, and O_j is the set of all order parameters for v_j . Given a factor graph $F_j = (V_j; E_j)$, let $G_j^i = (V_j^i; E_j^i)$ be a subfactor graph of F_j that is associated with a particular order parameter i_j .

Note F_j has a set of lowest parameter nodes Lp_j

$$Lp_j : \{i_j \in O_j \mid \exists v_j \in PV_j \text{ such that } i_j \leq \tau(v_j) \text{ for all } w \in PR_j \cup u\}$$

and a set of highest parameter nodes Hp_j

$$Hp_j : \{i_j \in O_j \mid \exists v_j \in PV_j \text{ such that } i_j \geq \tau(v_j) \text{ for all } w \in PR_j \cup u\}$$

with $p_j \in O_j$ and R_j as defined in Definition 2.6. For example, in the factor graph in Figure 2.6,

both nodes labeled 2222 are the set of highest parameters and both nodes labeled 0000 are the set of lowest parameters. Each subfactor graph G_j^i has a unique node $e_{j,i} \in P \setminus L p_j$ and unique node $h_{j,i} \in P \setminus H p_j$. We will call these nodes the root and leaf of a subfactor graph G_j^i , respectively.

Definition 3.1. Let $G_j^i = (V_j^i; E_j^i)$ be a subfactor graph. We define a strict partial order on V_j^i by $p_j^s \prec p_j^t$ when $j^s p_j^s \prec j^t p_j^t$ for all $! \in P \setminus R_j$, with strict inequality for at least one $! \in P \setminus R_j$.

Theorem 1. Let $G_j^i = (V_j^i; E_j^i)$ be a subfactor graph. Then

- (a) for any $p_j^s, p_j^t \in V_j^i$ with $p_j^s \prec p_j^t$, there is a path from p_j^s to p_j^t in G_j^i . In other words, there is a sequence of vertices $p_j^s = p_j^0; p_j^1; \dots; p_j^n = p_j^t$ such that $p_j^k; p_j^{k+1} \in E_j^i$ for all $k = 0; \dots; n-1$.
- (b) if $p_j^s; p_j^t \in P \setminus E_j^i$, then either $p_j^s \prec p_j^t$ or $p_j^s \succ p_j^t$.

Proof of Theorem 1. (a) We prove the statement in two steps. Assume first that $j^s p_j^s \prec j^t p_j^t$ for all $! \in P \setminus R_j$, with strict inequality for exactly one $! \in P \setminus R_j$ and that $j^t p_j^t \prec j^s p_j^s$ for all other $! \in P \setminus R_j$. If $n = 1$ then $p_j^s; p_j^t \in P \setminus E_j^i$ by Definition 2.7. Assume now that $n \geq 1$. Since V_j^i contains all logic parameters then there exists $p_j^1; p_j^2 \in P \setminus E_j^i$ such that $j^1 p_j^1 \prec j^s p_j^s \prec 1$ and $j^1 p_j^1 \prec j^t p_j^t \prec 1$ otherwise. Similarly, there exists $p_j^1; p_j^2 \in P \setminus E_j^i$ such that $j^2 p_j^2 \prec j^t p_j^t \prec 1$ and $j^2 p_j^2 \prec j^s p_j^s \prec 1$ otherwise. Repeating this process n times, we construct a path

$$p_j^s \tilde{N} p_j^1 \tilde{N} p_j^2 \tilde{N} \dots \tilde{N} p_j^n$$

in G_j^i . Notice that $p_j^n \prec p_j^t$ since $j^n p_j^n \prec j^t p_j^t$ for all $! \in P \setminus R_j$, proving that this is a path from p_j^s to p_j^t .

Assume now that there are $!_1; \dots; !_q \in P \setminus R_j$ such that $p_j^s \prec p_j^t$ satisfies $j^!_u p_j^s \prec j^!_u p_j^t$ for $u = 1; \dots; q$, but $j^! p_j^s \prec j^! p_j^t$ for all other $! \in P \setminus R_j$. We now sequentially apply the construction in step one by adjusting the values of $j^!_u$ one at a time. The important

restriction on this construction is that the f_j^k functions need to remain monotone functions throughout this process; i.e. $f_j^k \leq f_j^l$ implies $f_j^k(p) \leq f_j^l(p)$ as in Definition 2.6. It is easy to see that if $f_j^k \leq f_j^l$ then increasing the values of $f_j^s(p)$ before adjusting values of $f_j^t(p)$ will preserve the monotonicity of f_j^s at all stages of the construction. Therefore, we adjust the values of $f_j^s(p)$ starting from the highest p and then proceed down the partial order. The concatenation of these paths gives a path γ_j between p_j^s and p_j^t . This proves (a).

To prove (b), suppose $p_j^s, p_j^t \in P E_j^i$. Then by Definition 2.7 there exists exactly one $f_j^i \in PR_j$ such that $f_j^s(p) = f_j^t(p) - 1$, with equality for all other $f_j^i \in PR_j$. If $f_j^s(p) = f_j^t(p) - 1$ then $p_j^s \leq p_j^t$ and if $f_j^t(p) = f_j^s(p) - 1$ then $p_j^s \geq p_j^t$.

The following Corollary is an immediate consequence of Theorem 1.

Corollary 3.1. Each subfactor graph G_j^i is connected.

We will now prove that a subfactor graph is a graded poset, with the immediate consequence that its rank function can be applied to the factor graph as a whole. This procedure will allow us to divide the factor graph into a linearly ordered sequence of layers. A monotone function imposed on these layers can then provide a direction to parameter changes within the factor graph.

A rank function [22] L is a map on a poset PO such that given $x, y \in PO$

(i) $x \leq y$ implies $L(x) \leq L(y)$ and

(ii) $L(y) = L(x) + 1$ if y covers x .

A poset PO is graded if it admits a rank function L , and is denoted as $gr(PO; L)$ [22]. A chain is a totally ordered subset of PO and a maximal chain is a chain that isn't contained in a larger chain in PO [41].

Recall that the subfactor graph $G_j^i = (V_j^i; E_j^i)$ has unique root $\rho_{j,i} \in P L_j$ and unique leaf $h_{j,i} \in P H_j$.

Theorem 2. Let $L_j : V_j^i \rightarrow \mathbb{N} \cup \{0\}$ be a function on the vertex set of a subfactor graph $G_j^i = (V_j^i; E_j^i)$ defined as follows

$$L_j(p_j^q) = \sum_{! \in PR_j} L_j(p_j^{q-1}) \tag{3.1}$$

Then

- (a) $(V_j^i; q)$ is a graded poset with rank function L_j , and
- (b) $L_j(p_j^0) = 0$ and $L_j(p_j^q) = |R_j| - |T(p_j^q)|$.

Proof. Since the subfactor graph $G_j^i = (V_j^i; E_j^i)$ has unique root $p_j^0 \in PR_j$ and unique leaf $h_j^i \in PH_j^i$, to show (a) it is sufficient to prove that all maximal chains in $(V_j^i; q)$ have the same length [41]. We will show that this length is $|R_j| - |T(p_j^q)|$.

Consider a maximal chain in $(V_j^i; q)$

$$p_j^1 \prec p_j^2 \prec \dots \prec p_j^n$$

By Theorem 1(a) if $p_j^1 \notin \text{PR}_j$ the chain can be extended by an element smaller than p_j^1 and therefore the chain is not maximal. A similar argument applies to p_j^n and therefore $p_j^1 \in \text{PR}_j$ and $p_j^n = h_j^i$. Similarly, by Theorem 1, $p_j^q \prec p_j^{q-1}$ must satisfy $p_j^q \in PE_j^i$, otherwise a path in G_j^i could be inserted between p_j^q and p_j^{q-1} , contradicting maximality. Thus, for each $q = 1, \dots, n-1$, we have

$$L_j(p_j^{q-1}) = L_j(p_j^q) - 1$$

for exactly one $! \in PR_j$. Note that for any $! \in PR_j$, since

$$L_j(p_j^0) = 0 \text{ and } L_j(p_j^q) = |R_j| - |T(p_j^q)|$$

then there must be exactly $|T(p_j^q)|$ inequalities $p_j^q \prec p_j^{q-1}$ such that

$$L_j(p_j^q) = L_j(p_j^{q-1}) - 1$$

Thus, each $! \in PR_j$ requires $|T(p_j^q)|$ distinct inequalities in the maximal chain showing that the length must be $|R_j| - |T(p_j^q)|$. Since we chose an arbitrary maximal chain, we have shown

that all maximal chains in $\rho V_j^i; q$ have the same length, proving it is a graded poset.

Now we show that L_j (equation 3.1) is a rank function on $\rho V_j^i; q$. Let $p_j^s; p_j^t \in \rho V_j^i; q$ and suppose that $p_j^s \prec p_j^t$, then by Definition 3.1,

$$\begin{array}{c} \xrightarrow{\quad} \\ \text{! PR}_j \end{array} \quad \begin{array}{c} \xrightarrow{\quad} \\ \text{! PR}_j \end{array}$$

Additionally, when $p_j^s; p_j^t \in PE_j^i$ then p_j^t covers p_j^s which by Definition 2.7 implies $L_j(p_j^t) - L_j(p_j^s) = 1$. This proves part (a).

The proof of (b) follows directly from the definition of L_j .

Since the subfactor graphs of a factor graph F_j are isomorphic, the rank function L_j is the same for all G_j^i . We will use this rank function to define layers of F_j .

Remark 3.1. There is a subtle difference between our definition of the parameter graph as the set of all pairs of order and logic parameters, and our definition of the function ρ_j in the switching ODE model in (2.10). Not every logic parameter can be realized by a function and differences start for functions with 3 inputs [5]. The realizable parameters ρ are subset of all parameters P and these are encoded in the software DSGRN. We refer readers to Section 3.2 for the proof of Theorem 2 for realizable parameters.

Definition 3.2. Let $F_j = \rho V_j; E_j$ be the factor graph for node v_j , with decomposition into subfactor graphs $G_j^i = \rho V_j^i; E_j^i$ for $i \in \{1, \dots, |\text{TP}_j|\}$. The factor graph layer of $\rho_j \in \rho V_j^i$ is $L_j(\rho_j)$. The k -th factor graph layer of F_j is the node set

$$\{ \rho_j \in \rho V_j \mid L_j(\rho_j) = k \};$$

for $k \in \{0, \dots, |R_j| - |\text{TP}_j|\}$.

We say that the highest factor graph layer is the set H_{F_j} which is factor graph layer $|R_j| - |\text{TP}_j|$. Likewise, the lowest factor graph layer is the set L_{F_j} , which is factor graph layer 0.

To illustrate the concept of factor graph layers, consider a network node with one in-edge with $0 \leq u$ and two out-edges with thresholds τ_1 and τ_2 . There are 12 factor parameters in the factor graph, see Figure 3.1(a). This factor graph has five layers. Each node label ab represents a logic parameter p_j^q with $a = j^q$ and $b = j^u$. Since $L_j(p_j^q) = \text{wPR}_j(p_j^q)$ by equation 3.1, then the factor layer number is $a + b$. Another factor graph example is in Figure 2.6. This factor graph has 9 layers arranged horizontally and numbered by the sum of the labels $abcd$. The factor graph layers allow us to define an idea of monotonicity of paths through a factor graph.

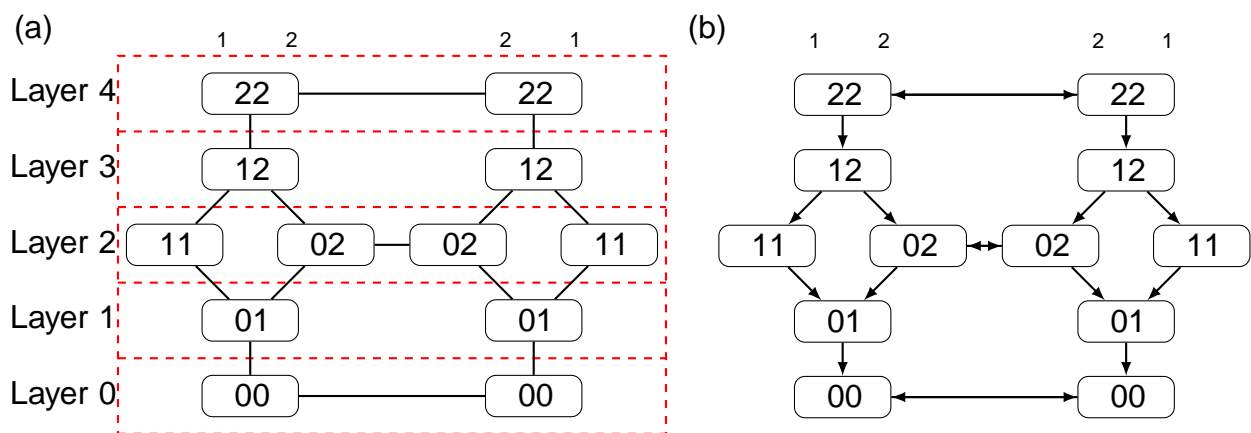


Figure 3.1: Factor graph layers and modeling external variables. (a) The factor graph and factor graph layers for aRN node with one in-edge and two out-edges with thresholds τ_1 and τ_2 and input values $0 \leq u$. The red dashed lines depict which nodes in the factor graph belong to each factor graph layer. (b) The factor graph from (a) with an activating external variable imposed, with directed edges depicting the direction of motion when $u_j = 1$ (violet). Notice that the external variable is inducing decreasing monotonicity through the factor graph.

Definition 3.3. Consider a path through the factor graph $\mathbb{F}_j = \{p_j^q; E_j^q\}$ with sequence of nodes $p_1; p_2; \dots; p_m \in V_j$. The path is said to be monotone increasing if for all p_i, p_k in the path, we have $L_j(p_i^q) \leq L_j(p_k^q)$ if and only if $i \leq k$, i.e. factor graph layers increase along the path. Similarly, the path is said to be monotone decreasing if, for all p_i, p_k in the path, we have $L_j(p_i^q) \geq L_j(p_k^q)$ if and only if $i \geq k$. A monotone increasing or monotone

decreasing path in F_j is called a monotone path .

3.2 Factor Graph Layers for DSGRN Realizable Parameters

Recall from Section 2.4.2 that a logic parameter for node v_j in a regulatory network is a function $\rho_j : R_j \rightarrow \tilde{N} \times X_j$, where

$$R_j = \{ (x_1, \dots, x_k) \mid x_m \in P_{j,m}; u_{j,m} \in U \}$$

with $k = |S(v_j)|$ the number of source nodes of v_j . Further recall that an order parameter for v_j is a bijective map $\sigma_j : \tilde{N} \times \{0, 1, \dots, M\} \rightarrow U$, where $\sigma_j = (i_{1,j}, \dots, i_{M,j})$ is the collection of thresholds for v_j and $M = |T(v_j)|$ is the number of targets of v_j . Lastly, recall that a factor graph $F_j = (V_j; E_j)$ for v_j has $M!$ isomorphic subfactor graphs $G_j^i = (V_j^i; E_j^i)$ where the V_j^i partition V_j , i.e., $\bigcup_j V_j^i = V_j$ [7]. Each of these subgraphs G_j^i is associated with a particular threshold order σ_j^i . Each subfactor graph G_j^i has unique lowest parameter node $\ell_{i,j}$ given by

$$\ell_{i,j} = \rho_j^0; \sigma_j^i \text{ where } \rho_j^0(x) = 0 \text{ for all } x \in P_{R_j}$$

unique highest parameter

$$h_{i,j} = \rho_j^M; \sigma_j^i \text{ where } \rho_j^M(x) = M \text{ for all } x \in P_{R_j}$$

Definition 3.4. A factor parameter node $\rho_j = (\rho_j; \sigma_j)$ is DSGRN realizable if there exist sets of real, positive values $\{l_{j,m}; u_{j,m}\}_{m=1}^k$ and σ_j , where the elements of σ_j are all distinct, and a function $g : R^k \rightarrow \tilde{N} \times R$ which has the form of product of sums

$$g(x_1, \dots, x_k) = \prod_{m=1}^k (x_m + l_{j,m}) \quad (3.2)$$

such that for all $x \in P_{R_j}$ and all n_j

$$\rho_j(x) = n_j \text{ if and only if } \rho_j(x) = \sigma_j^{-1}(n_j) \in U$$

and an equality $g(x) = u_{n,j}$ never occurs. We call the collection

$$w : \{t\} \cup \{l_{j,m}; u_{j,m}\} \cup \{u_{m-1}^k\} \cup Y_j$$

a witness of the parameter node ρ under function g .

Remark 3.2. Note that the nodes s_{ij} and h_{ij} are realizable for any function g of the form in (3.2). To see this, choose an arbitrary set of real, positive values $u_{j,m}$ and set

$$\hat{m} : \min\{g(x) \mid x \in P_{j,u}\} \quad \text{and} \quad \hat{M} : \max\{g(x) \mid x \in P_{j,u}\}$$

Note that $0 < \hat{m} < \hat{M}$. Then if we select a set $\{j,p\}$ with $\max\{j,p\} < \hat{m}$ then $U \cup Y_j$ is witness for $h_{j,i}$ and if we select $\{j,p\}$ with $\min\{j,p\} > \hat{M}$ then $U \cup Y_j$ is witness for s_{ij} .

We denote the set of \mathbb{R} real-valued inputs evaluated on the witness w by $R_{j,p\{q\}}$ and the set of threshold values in witness w by $\{j,p\}$ and let

$$Y_{j,p\{q\}} : \{g(x) \mid x \in P_{j,p\{q\}}\}$$

where repeated elements are permitted.

Lemma 3.1. Given parameter node ρ and function g , for a generic choice of w , the set $Y_{j,p\{q\}}$ is totally ordered. That is, there exists an open and dense set $\mathcal{D} \subseteq V$ where V is an open subset of $\mathbb{R}^{|w|}$ of those values that satisfy

1. $0 < l_{j,m} < u_{j,m}$ (i.e., (2.11)),
2. distinct thresholds in $\{j,p\}$,
3. $Y_{j,p\{q\}} \cap X_j \cap H_j = \emptyset$, and
4. the inequality constraints of the parameter node ρ ,

such that $w \in \mathcal{D}$ implies all values of $Y_{j,p\{q\}}$ are distinct.

Proof. Notice that the requirement that $0 < |j; m| < u_{j; m}$ induces the condition $g(x) = g(x^1)$ for $x; x^1 \in R_j$ whenever $x_m = x_m^1$ and $x_s = x_s^1$ for all $s \neq m$. The problem of potential equality, $g(x) = g(x^1)$, can only occur when $x_{m_1} = x_{m_1}^1$ and $x_{m_2} = x_{m_2}^1$ for some $m_1 \neq m_2$.

Suppose for a witness w for parameter p , there are two values $x; x^1 \in R_j(p, w)$ such that $g(x) = g(x^1)$. Choose a position m such that $x_m = x_m^1$, and assume without loss of generality that $x_m = |j; m|$.

Define w' as a witness under g of some parameter node q by taking the witness w and changing exactly one value: $|j; m| = |j; m| + \epsilon$ for some $\epsilon > 0$. In particular, ϵ must be small enough to ensure $|j; m| + \epsilon < u_{j; m}$ and $Y(p, w) \cap X_j(p, w') = \emptyset$. The latter can be accomplished since $Y(p, w) \cap X_j(p, w)$ has a finite number of points. Notice then that $g(x) = g(x^1) = g(x^1)$, where $x = x + \epsilon e_m$, and e_m is the unit vector in the m^{th} direction. Note that these conditions remain true for any $0 < \epsilon < \epsilon_0$.

We would like to ensure that w' is a witness for the same parameter p as w , i.e. $q = p$. It is sufficient to satisfy $|j; m| \leq |j; m| + \epsilon$ for all $y \in R_j(p, w')$. Clearly this holds true for any y where $y_m = u_{j; m}$, since $y = y$. So consider y with $y_m = |j; m|$ and suppose $|j; m| + \epsilon > |j; m|$ for some $|j; m|$. Since g is continuous, ϵ can be chosen sufficiently small so that $|j; m| + \epsilon \leq |j; m|$ as well. Repeat for all $y \in R_j(p, w')$ with $y_m = |j; m|$ to choose an ϵ sufficiently small to simultaneously satisfy all these constraints, and ensure $q = p$.

We must also avoid introducing new equalities, i.e. we additionally require $g(y) = g(z)$ whenever $g(y) = g(z)$ for $y; z \in R_j(p, w')$. Since g is a continuous function and $g(y)$ and $g(z)$ are isolated, taking ϵ sufficiently small ensures that for each such pair $y; z$, it remains true that $g(y) = g(z)$.

After all the adjustments to w have been made, the new witness w' for parameter p now ensures that $g(x) = g(x^1)$ without introducing any new duplicates in $Y(p, w')$. However, there may be other pairs $y; y^1 \in R_j(p, w')$ that satisfy $g(y) = g(y^1)$. Since there are at most a finite number, the procedure above may be repeated until some final witness w'' for p

under g is constructed such that all elements of Y_{pwq} are distinct. Since at each step, the corresponding ϵ may be taken arbitrarily small, it is true that given any witness w , there is another witness w' arbitrarily close to w where Y_{pwq} is totally ordered. This proves that the property of total ordering of Y_{pwq} is dense in U .

Since g is continuous, there is an open neighborhood of witnesses in $R^{|W|}$ whenever Y_{pwq} is totally ordered, since Y_{pwq} has a finite number of isolated values. Call the neighborhood N_{pwq} . Then under the subspace topology $W \times N_{pwq} \in U$ is relatively open in V . Since U is covered by $\bigcup_w V \times N_{pwq}$, U is open in V .

Definition 3.5. A DSGRN realizable sub-factor graph $G_j^i = (V_j^i; E_j^i)$ under g is a node induced subgraph G_j^i , where the collection of nodes $v_j^i \in V_j^i$ are those nodes that have a witness w under g . A DSGRN realizable factor graph F_j^i under g is the product

$$F_j^i = \prod_{i=1}^n G_j^i$$

Lemma 3.2. Assume $p = p_{ij}; j \in q \in P G_j^i; p = p_{ij}$, and let w be a witness of p under function g . Then there exists a path from p to p_{ij} within G_j^i . Likewise, there exists a path from p to h_{ij} within G_j^i .

Proof. Assume without loss of generality that the witness w induces a totally ordered set Y_{pwq} see Lemma 3.1. Define the sets

$$\begin{aligned} Q_0 &= \{t \in P R_j(pwq) \mid j \mu q = 0\} \\ Q_1 &= \{t \in P R_j(pwq) \mid j \mu q = 1\} \\ &\vdots \\ Q_M &= \{t \in P R_j(pwq) \mid j \mu q = M\} \end{aligned}$$

where recall $M = |T(p_j, q)|$. Since $p = p_{ij}$, there exists at least one nonempty Q_n with $n \geq 0$. Since Q_n is a finite set, it has a smallest element $r_0 \in j \mu_0 q$. Let $r_1 \in P Q_n$ be the smallest element in $Q_n \cap r_0 u$, if it exists, and let $r_1 = p_j q^{-1} p n q$ if such a smallest element does not

exist. Let $r_1 = r_0$ and let

$$\tau_{n-1,j}^1 := r_0 - \frac{1}{2}$$

We define a new witness where we replace $\tau_{n-1,j}$ by the new value of the threshold $\tau_{n-1,j}^1$

$$w^1 = \tau_{l,j;m}^1; u_{j;m}^k u_{m-1}^1 u_{j-1}^1$$

with $\tau_j^1 = p_{j-1} z_{n-1,j} u_{j-1}^1 u_{j-1}^1 u_{j-1}^1$. Then w^1 is a witness for the parameter q where the value of τ_j changes from $n-1$ to value $n-1$. Therefore q is a node in \hat{G}_j^i that is the immediate neighbor of node p and therefore there is an edge between p and q . Repeating this argument it is easy to see that eventually only the set Q_0 is non-empty, which occurs only at the node $\tau_{i,j}^1$. Therefore, every node p in the subfactor graph has a path to $\tau_{i,j}^1$.

The analogous argument proves the existence of a path from p to $h_{i,j}$.

Corollary 3.2. Every DSGRN realizable subfactor graph \hat{G}_j^i is connected.

Proof. By the Remark 3.2 every realizable subfactor graph \hat{G}_j^i contains both $\tau_{i,j}^1$ and $h_{i,j}$. Then every node $p \in \tau_{i,j}^1$, including $h_{i,j}$, in the \hat{G}_j^i is connected to $\tau_{i,j}^1$ by Lemma 3.2.

Corollary 3.3. Every DSGRN realizable factor graph F_j^A is connected.

Proof. Recall that $\tau_{i,j}^1; h_{i,j} \in P \hat{G}_j^i$ for any subfactor graph \hat{G}_j^i . Consider a subfactor graph \hat{G}_j^k such that τ_j^j and τ_j^k are identical except for two adjacent thresholds. That is, $\tau_{n,j}^j q$ $\tau_{n,j}^k q$ $\tau_{n,j}^j q$ $\tau_{n,j}^k q$ $\tau_{n,j}^j q$ $\tau_{n,j}^k q$ $\tau_{n,j}^j q$ $\tau_{n,j}^k q$ otherwise. Then the nodes $\tau_{i,j}^1$ and $\tau_{i,k}^1$ are connected in F_j , and therefore they are connected in F_j^A .

Since there is a sequence of such adjacent swaps that connects any two permutations of τ_j , the set of nodes $\tau_{s,i}^m u_{s-1}^m$ is connected in F_j^A . Since each \hat{G}_j^i is connected, F_j^A is connected.

Finally, note that F_j^A inherits the structure of graded poset from F_j .

3.3 Interpreting External Variables as Parameter Changes

Regulatory networks do not operate in isolation; they are subject to environmental factors that can impact their function. We choose to model the impact of a spatially monotone but temporally constant environmental variable as a directed sequence of parameter changes induced in a targeted subset of nodes, see Figure 2.3(c).

Definition 3.6. A monotone external variable $c : Y \rightarrow \mathbb{R}$, is an external variable not included in the RN that satisfies either $c^1(p) \neq 0$ (increasing) or $c^1(p) \leq 0$ (decreasing) on Y .

For the purposes of this manuscript, one can imagine the domain to be a spatial dimension. We assume the following properties of the external variable:

1. If $c(p)$ is an activator of a network node v_j , then the abundance of v_j qualitatively matches the abundance of $c(p)$, i.e. high levels of $c(p)$ induce high levels of v_j and low levels of $c(p)$ are associated to low levels of v_j .
2. If $c(p)$ is a repressor of a network node v_j , high levels of $c(p)$ induce low levels of v_j and low levels of $c(p)$ induce high levels of v_j .
3. Monotone changes in $c(p)$ induce a corresponding monotone response in v_j .

We elaborate on the last point. Let $\sigma_j = 1$, where $\sigma_j = 1$ means that $c(p)$ is an activator and $\sigma_j = -1$ means $c(p)$ is a repressor to the target node v_j . Let F_j be the factor graph of v_j . We model the effect of $c(p)$ on v_j as a monotone path over the layers of F_j : $c(p)$ induces monotone increasing paths in F_j when $\sigma_j \cdot \text{sign}(c^1(p)) = 1$ and monotone decreasing paths when $\sigma_j \cdot \text{sign}(c^1(p)) = -1$. This monotonicity condition on the factor graph of v_j is a model of the continuously changing abundance of v_j as a function of changing $c(p)$. In Figure 3.1(b), an activating external variable that is monotone decreasing in v_j (violet) is imposed on the

factor graph. This induces decreasing monotonicity on the factor graph shown as directed edges.

We will make use of a stricter condition on the modeling of external forcing that requires target nodes v_j in PRN to not only exhibit consistently high and low expression but to operate at the most extreme factor graph layers.

Definition 3.7. A maximal monotone path in the factor graph F_j is either

1. a monotone increasing path that starts in the lowest factor graph layer and ends in the highest factor graph layer, or
2. a monotone decreasing path that starts in the highest factor graph layer and ends in the lowest factor graph layer.

In Chapter 4, we show how to apply this modeling framework to match observations along a spatial domain under external variable control and apply it to the example of the *Drosophila melanogaster* gap gene network. Additionally, we show how biological observations may be translated into the language of Morse graphs, and apply this translation to *Drosophila melanogaster* development. In Chapter 5, we construct paths in the DSGRN parameter graph, making use of concepts developed in this chapter.

CHAPTER FOUR

EXPRESSING EXPERIMENTAL DATA AS MORSE GRAPHS

In this chapter, we interpret spatial data as a sequence of fixed points of a dynamical system and translate these into DSGRN Morse graphs. We then demonstrate this technique on gene expression data from the gap gene network.

4.1 Descriptive Pattern and Phenotype Pattern Graph

We formally describe a methodology for interpreting any spatial data in a DSGRN framework. For a given network model $\mathcal{N} = (V; E; \varphi)$, we consider paths $\varphi_1, \dots, \varphi_k$ in the parameter graph $\mathcal{P}(\mathcal{N})$ and the corresponding sequences of fixed point Morse sets $\{FP_1, \dots, FP_k\}$ as the output of the network model. Since the number of out-edges of $v \in V$ determines the highest integer state of X_j (see Definition 2.6), the highest value of an FP annotation will vary across network topologies. This complicates the comparison of network models to each other and to the data. Therefore, in order to match experimental data to the model output in X_j , we first transform experimental data to qualitative data, using the descriptors "high", "intermediate" and "low". Then for each network under consideration, we transform this qualitative data to integer values in X_j .

Definition 4.1. Consider a finite set L of qualitative expression level labels, for example $L = \{l; h\}$ for "low" and "high", that admits a (not necessarily strict) total order, such as $l \leq h$. Further, consider a spatial data set of M genes and N spatial locations. Then an $N \times M$ matrix D with $D_{ij} \in L$ is the descriptive pattern of the spatial data.

We desire to match a DSGRN model of a regulatory network with M vertices $V = \{v_1; \dots; v_M\}$ to the descriptive pattern at spatial locations $\{1; \dots; N\}$. In order to perform this matching, we map the n -th row of the descriptive pattern D (denoted D_n)

onto a collection of DSGRN fixed points (FPs), whose annotations match D_n . We then organize this data into a phenotype pattern graph, that is, a DSGRN representation of the observed data. We will say that there is a match between the data and the DSGRN model if there is a path in the phenotype pattern graph $\{m_1, \dots, m_k\}$ and a path in the DSGRN parameter graph $\{p_1, \dots, p_k\}$ such that at every position i , there is at least one Morse node FP PMG_{pp} such that $FP = m_i$.

Definition 4.2. A pattern label $\{p_1, \dots, p_M\}$ where $p_j \in X_j$ (see Definition 2.6 for definition of X_j), is a collection of integer states, one for each variable $x_j; \dots; v_M$. Let D be the descriptive pattern for a spatial data set with M variables and N spatial locations. We say a pattern label is consistent with D_n if $D_{n;j} = D_{n;k}$ implies $p_j = p_k$. The set of pattern labels associated with spatial location n is

$$P_n = \{p \mid p \text{ is consistent with } D_n\}.$$

The phenotype pattern of D is

$$P = \{p \mid p_1, p_2, \dots, p_N\}$$

A parameter node $p \in PG$ has a relevant phenotype if there is an $n \in \{1, \dots, N\}$ and a pattern label $p \in P_n$ such that there exists a Morse node $FP \in PMG_{pp}$. If PMG_{pp} is monostable (see Definition 2.11), then we say p is a strict phenotype of p . Lastly, we use the notation PMG_{pp} to denote the pattern label of a monostable fixed point Morse node in PMG_{pp} .

Note that a phenotype pattern is a coarse representation of spatial data that we want to match by a sequence of FPs along a path that represents a continuous path in the DSGRN parameter graph PG . However, consecutive pattern labels between p_n and p_{n+1} may differ at two or more elements. We make the reasonable assumption that continuity permits the insertion of intermediate pattern labels when seeking paths through the DSGRN parameter graph.

Definition 4.3. For a network $RN = (V; E)$ with $|V| = M$ and two vectors $c, d \in \mathbb{R}^M$, a set of transition vectors between c and d is

$$T_{c,d} = \{a \in \mathbb{R}^M \mid a_i = \pm d_i \text{ for } i = 1, \dots, M\}$$

where

$$I_i = \begin{cases} \{d_i, -d_i\} & \text{if } c_i = d_i; \\ \{d_i, -c_i\} & \text{if } c_i \neq d_i; \end{cases}$$

Define

$$T_{pq} = \{T_{c,d} \mid c \in P_{pq} \text{ and } d \in P_{n-1,q}\}$$

to be the set of transition pattern labels from position n to position $n-1$, and let

$$\hat{T}_{pq} = \bigcup_{n=1}^{n-1} T_{pq}$$

Definition 4.4. A phenotype pattern graph for D is a directed graph $PPG = (P; E)$ where $p, q \in P$ if $p \neq q$ or the following are simultaneously satisfied

$$\hat{T}_{pq} \neq \emptyset \text{ and } \hat{T}_{qp} \neq \emptyset \text{ and}$$

$$\hat{T}_{pq} \neq \emptyset \text{ implies } D_{n,j} \leq D_{n-1,j} \text{ and } \hat{T}_{qp} \neq \emptyset \text{ implies } D_{n,j} \geq D_{n-1,j}.$$

4.2 Drosophila Melanogaster Example

As an example, we describe the construction of a descriptive pattern for the melanogaster data. Figure 4.1 shows the protein concentration data of the trunk gap genes along the A-P axis of the embryo. These data are taken late in the segmentation process when protein concentrations have equilibrated to a fixed distribution across the A-P axis (see Section 2.2). We therefore assume that these concentrations correspond to steady state values of the segmentation dynamics.

At most positions along the A-P axis, the protein expression levels of the four genes are ordered, with the expression of two genes having very low protein concentration. Furthermore, there are sections where this ordering doesn't change. For example, at every point between positions 40% and 45% egg length the protein expression levels are ordered, from highest to lowest, Hb, Kr, Gt, Kni. Using these observations we divide the A-P axis into eight regions R_n (see the dashed lines in Figure 4.1), where the protein expression levels are consistently ordered. Region boundaries are at crossings between two protein concentrations.

R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8

Figure 4.1: Data of protein concentration along the anterior-posterior position % egg length for the trunk gap gene proteins Hunchback (Hb) in yellow (peak in R_3), Giant (Gt) in blue (peak in R_1, R_8), Krüppel (Kr) in green (peak in R_4) and Knirps (Kni) in red (peak in R_6). The gap gene protein expression pattern data ([SData.ods](#)) was obtained from supplementary information in [46].

We discretize the experimental values in each of the 8 regions by one of the descriptive labels L or H ; $?$:

1. H : Protein expression level is high,
2. L : Protein expression level is low,
3. $?$: Protein expression level is indeterminate.

Since there are large regions where it is unclear whether protein expression levels should be regarded as high or low, we introduce the third character. The order on the label set \mathcal{L} is $L \prec \cdot \prec H$ and $L \prec H$.

Protein expression levels in Figure 4.1 show that *kni* is inactive between A-P positions 35-47%, *gt* is inactive between 49-59%, and *hb* is inactive between 61-75%, which is consistent with the interpretation of Verd et al [48]. Thus, in these regions, these protein expression levels will be labeled L. We further assign label H to each gene whose protein expression level is highest in a given region in Figure 4.1. Therefore in each region, we will have one gene labeled H, two genes labeled L and the remaining gene with intermediate protein expression will be assigned \cdot . We arrive at the descriptive pattern seen in Table 4.1.

Region	A-P	Hb	Gt	Kr	Kni
R_1	35-37	\cdot	H	L	L
R_2	37-40	H	\cdot	L	L
R_3	40-45	H	L	\cdot	L
R_4	45-51	\cdot	L	H	L
R_5	51-57	L	L	H	\cdot
R_6	57-63	L	L	\cdot	H
R_7	63-67	L	\cdot	L	H
R_8	67-75	L	H	L	\cdot

Table 4.1: Descriptive pattern for the *D. melanogaster* protein expression pattern data seen in Figure 4.1.

We now transform the descriptive pattern in Table 4.1 into a phenotype pattern graph. We use the following FP assignment for a pattern label $p \in \{Hb; Gt; Kr; Kni\}$ at spatial position $n \in \{1; \dots; 8\}$. For every gene v_j with $j \in \{Hb; Gt; Kr; Kni\}$, we assign

$$\hat{v}_j = 0 \text{ if } D_{n,j} = L,$$

$$\hat{v}_j = \max\{1; |T(p_j, q)|\} \text{ if } D_{n,j} = H,$$

$$\hat{v}_j = |T(p_j, q)| \text{ if } |T(p_j, q)| > 0 \text{ and } D_{n,j} = \cdot, \text{ then } \hat{v}_j \in \{1; \dots; |T(p_j, q)|\} \text{ and}$$

if $|T_{p_j, q}| = 0$ and $D_{n,j} = 1$, then $p_j \in P_{t_0; 1u}$.

Note that the pattern labels are consistent with D by Definition 4.2. We use this assignment to construct a phenotype pattern graph for a regulatory network with proteins Hb, Gt, Kr and Kni, as illustrated in Figure 4.2. The network, called the strong edges network (StrongEdges) consists of edges with the strongest predicted interaction between the trunk gap genes from the original gap gene network in Figure 2.2(a), except the self-loops (see Figure 4.2(left)). Additionally, Figure 4.2(right) shows a table representing all possible pattern labels for each region (compare to Table 4.1).

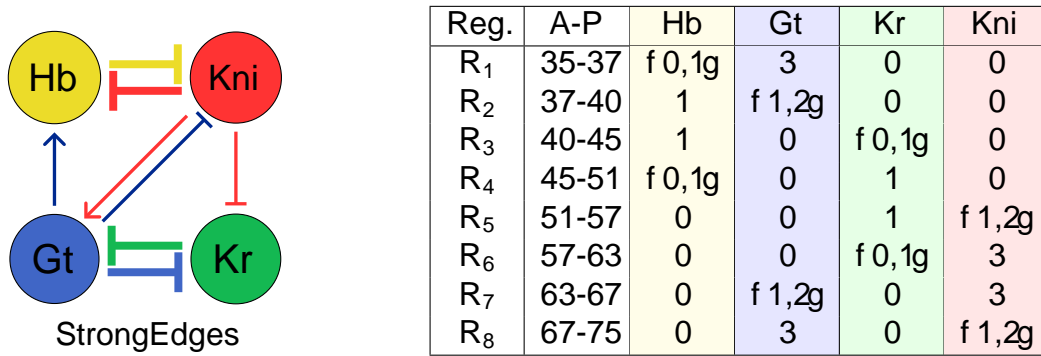


Figure 4.2: StrongEdges network (left) and the phenotype pattern for StrongEdges (right).

Consider regions R_1 and R_2 , where $D_1 = (p, H, L, L)$ and $D_2 = (p, H, L, L)$. The collections of pattern labels are

$$c = (p_1, q_1) = (0; 3; 0; 0) \text{ and } d = (p_2, q_2) = (1; 1; 0; 0)$$

For pattern labels $c = (p_1, q_1)$ and $d = (p_2, q_2)$ we have

$$T_{c;d} = (p_1, q_1; p_2, q_2) = (0; 3; 0; 0; 1; 1; 0; 0)$$

Doing this for each $c \in P_1$ and $d \in P_2$ we find that $T_{c;d} \in T_{c;d}$ (note that in general this need not be true). The entire phenotype graph for StrongEdges can be seen in Figure 4.3.

The phenotype pattern describes the annotations of the DSGRN fixed points that we say match the data in each of the eight regions along the A-P axis. We are interested in

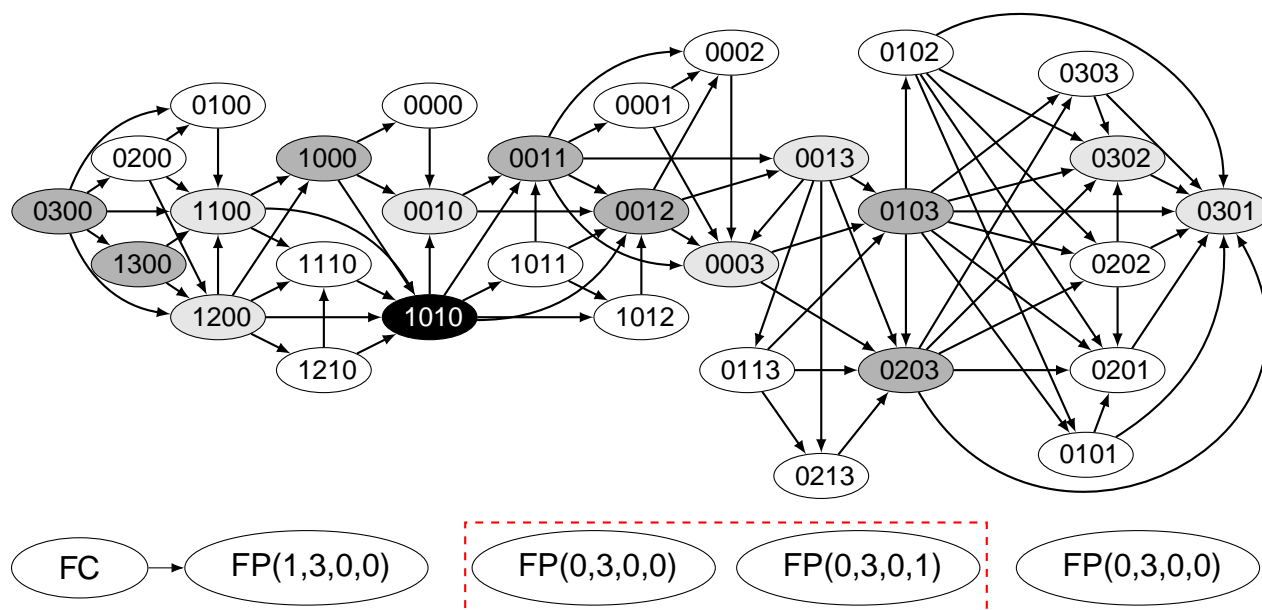


Figure 4.3: StrongEdges phenotype pattern graph. (Top) StrongEdges phenotype pattern graph, where labels $abcd$ correspond to pattern labels $a; b; c; d$. Note each node has a self-loop which is omitted for clarity. Nodes are shaded if the pattern label is in p_nq for some region R_n , $n = 1; 2; \dots; 8$. Dark gray is specific for $n = 1; 3; 5; 7$, light gray is specific for $n = 2; 4; 6; 8$, and white indicates that the pattern label is a transition label only. The black node is a pattern label that is consistent both region R_3 and R_4 . (Bottom) Three examples of associated StrongEdges relevant phenotypes for p_1q . The nodes boxed in dashed red depict a bistable Morse graph, which is the only example that isn't a strict phenotype for StrongEdges.

finding a sequence of parameter graph nodes with Morse graphs that exhibit fixed points determined by the phenotype pattern. In our example, the associated StrongEdges fixed points for p_1q are in the set

$$\{FP(0; 3; 0; 0), FP(1; 3; 0; 0)\}$$

Any Morse graph containing one of these fixed points is consistent with the data in R_1 . For example, a Morse graph having a full cycle connected to $FP(3; 0; 0; 0)$ as shown in Figure 4.3, a bistable Morse graph shown boxed in red, and the monostable Morse graph $FP(3; 0; 0; 0)$ all exhibit relevant phenotypes for R_1 . Additionally, the two monostable Morse graphs are strict phenotypes of R_1 .

CHAPTER FIVE

MODELING SPATIAL GRADIENTS AND MATCHING OBSERVATIONS WITH
DSGRN

We now describe how spatial data can be compared to DSGRN network model predictions while respecting external variables. The basic construction is a chemical gradient graph, whose name is inspired by the spatial distributions of the *D. melanogaster* maternal gradients Bcd and Cad. It is constructed as a subgraph of the DSGRN parameter graph with directed edges imposed by external variables on the factor graphs of network nodes affected by these external variables. Within the chemical gradient graph, we identify developmental paths, with this name again motivated by our example, that are consistent with both the external variables and the phenotype patterns derived from the spatial data.

The collection of developmental paths is the subgraph of the chemical gradient graph composed of all matches between the DSGRN model and the data. The shape and other features of this subgraph represent the DSGRN prediction of the robustness of the developmental program. However, the chemical gradient graph is prohibitively large, and therefore computing and investigating all paths is prohibitive as well. Therefore we compress the chemical gradient graph into smaller graphs that retain the information about the developmental paths. We first create a condensed chemical gradient graph, and then from that a path graph whose structure contains information about the quality and quantity of matching between the model and the data. There is one path graph per network model and comparisons of robustness between path graphs permit a ranking of network models, a subject that is discussed in Chapter 6.

5.1 Chemical Gradient Graph

The translation of an experimental spatial dataset into a phenotype pattern graph allows us to study the collection of paths in the DSGRN parameter graph that (1) are consistent with the action of external variables, and (2) match the phenotype pattern by the sequence of annotated Morse nodes. To facilitate this investigation we construct a subgraph of parameter graph PG where we add orientation to the edges that match the effect of external variables. We call this directed graph the chemical gradient graph \mathcal{G} .

Let $PG = (P; A; q)$ be the DSGRN parameter graph of $RN = (p; V; E; q)$ where $|V| = M$. Let $\{v_1, \dots, v_m\} \subseteq V$ be the maximal subset of network nodes where each v_j , $j = 1, \dots, m$, is affected by an external variable c_j for $y \in Y$, see Definition 3.6. We allow one external variable to affect multiple nodes, but do not consider the case when one node is controlled by multiple external variables. Let $\sigma_j = \pm 1$, where $\sigma_j = 1$ (-1) denotes that c_j is an activator (repressor) of v_j . Additionally, recall that L_j is the factor graph layer of v_j for a parameter node $p \in P$. Lastly, recall that $D_{N;M}$ is the descriptive pattern for a spatial data set with N regions and M genes.

Definition 5.1. The chemical gradient graph $\mathcal{G} = (p; V; E; q)$ is a directed graph constructed from PG and descriptive pattern $D_{N;M}$ in two steps. First, for $p \in P$ we have $p \in V$ if $p \in MG$; where \mathcal{P} is the set of nodes in the phenotype pattern graph $\mathcal{P}(PG)$ as defined in Definition 4.4. Then for each $p; q \in V$, we have $p; q \in E$ (directed) if $p; q \in PA$ and one of the following is satisfied

1. L_j is a subgraph of L_k for all $j = 1, \dots, m$, or
2. L_k is a subgraph of L_j for some $k \in \{1, \dots, m\}$ and L_j is a subgraph of L_k for all $j = 1, \dots, m$ such that $j = k$.

Remark 5.1. Notice that condition (1) implies $p \rightarrow q$; $q \rightarrow p$ PE, while condition (2) implies $p \rightarrow q$ PE, but $q \rightarrow p$ RE, since $L_{k \rightarrow p \rightarrow q} = \text{sign}(\alpha_k^1) \alpha_k = L_{k \rightarrow p \rightarrow q}$ flows against the gradient. This fact will be important later.

We now again turn to our example of the gap gene network *D. melanogaster*. Consider the maternal gradients Bcd and Cad, which we model as external variables to the gap gene network. Concentration of these proteins varies monotonically along the A-P axis; Bcd is increasing and Cad is decreasing. Since Figure 2.2(a) indicates that both Bcd and Cad may affect Gt and Kr, the spatial variance of this effect along the A-P axis is not clear. Therefore, we chose to only model effects of material gradient that have clear spatial differences along the A-P axis, namely the impact of Bcd on Hb and Cad on Kni. Let $x \in [0; 100]$ represent the A-P axis, where $x = 0$ represents the start of the anterior region and $x = 100$ the end of the posterior region. Then using the notation $\langle p \rightarrow q \rangle$ as in Definition 3.6 with $c = \text{Bcd}; \text{Cad}$ and $j = \text{Hb}; \text{Kni}$, we have $\langle \text{Bcd} \rightarrow \text{Hb} \rangle$ modeling the effect of Bcd on Hb as an activating ($\alpha_{\text{Hb}} = 1$) decreasing ($\text{sign}(\alpha_{\text{Hb}}^1) = -1$) external variable and $\langle \text{Cad} \rightarrow \text{Kni} \rangle$ modeling the effect of Cad on Kni as an activating ($\alpha_{\text{Kni}} = 1$) increasing ($\text{sign}(\alpha_{\text{Kni}}^1) = 1$) external variable (see the violet and cyan gradients in Figure 2.2(b)). Thus, p and q are nodes in the parameter graph $\mathcal{P} \rightarrow \mathcal{P}; \mathcal{A} \rightarrow \mathcal{Q}$ that satisfy $\langle p \rightarrow q \rangle \in \mathcal{M} \rightarrow \mathcal{P}$ and $\langle p \rightarrow q \rangle \in \mathcal{P}$ and $\langle p \rightarrow q \rangle \in \mathcal{P}$, then $\langle p \rightarrow q \rangle$ PE as well if

1. $L_{\text{Hb} \rightarrow p \rightarrow q} = L_{\text{Hb} \rightarrow p \rightarrow q}$ and $L_{\text{Kni} \rightarrow p \rightarrow q} = L_{\text{Kni} \rightarrow p \rightarrow q}$, or
2. $L_{\text{Hb} \rightarrow p \rightarrow q} = -L_{\text{Hb} \rightarrow p \rightarrow q}$ and $L_{\text{Kni} \rightarrow p \rightarrow q} = L_{\text{Kni} \rightarrow p \rightarrow q}$, or
3. $L_{\text{Kni} \rightarrow p \rightarrow q} = -L_{\text{Kni} \rightarrow p \rightarrow q}$ and $L_{\text{Hb} \rightarrow p \rightarrow q} = L_{\text{Hb} \rightarrow p \rightarrow q}$.

5.2 Developmental Paths

Having constructed the chemical gradient graph \mathcal{G} , we will now describe paths in \mathcal{G} that are consistent with the data.

Definition 5.2. Let $G = (V; E; q)$ be the chemical gradient graph of some $\mathcal{N} = (V; E; q)$. Let $v_1; \dots; v_m \in V$ be network nodes under influence of the corresponding external variables c_j and regulation types τ_j . Let F_j be the factor graphs of v_j , with L_j and H_j the sets of lowest and highest factor graph layers of F_j respectively. Let $n = 1; \dots; N$ denote the spatial regions of the dataset.

We say that $PS \in V$ is a starting node if

1. $s \in L_j$ for all j such that $\tau_j = 1$ and $c_j^1 \neq 0$ or $\tau_j = 1$ and $c_j^1 = 0$,
2. $s \in H_j$ for all j such that $\tau_j = 1$ and $c_j^1 = 0$ or $\tau_j = 1$ and $c_j^1 \neq 0$, and
3. the annotation of the Morse set at s must match the phenotype pattern in one of the first ℓ regions, i.e., $\text{pMG}(s) \in \mathcal{P}^{\ell, \tau, \tau_1, \dots, \tau_\ell}$ where ℓ is a modeling choice.

We say that $PT \in V$ is a stopping node if

1. $t \in H_j$ for all j such that $\tau_j = 1$ and $c_j^1 \neq 0$ or $\tau_j = 1$ and $c_j^1 = 0$,
2. $t \in L_j$ for all j such that $\tau_j = 1$ and $c_j^1 = 0$ or $\tau_j = 1$ and $c_j^1 \neq 0$, and
3. the annotation of the Morse set at t must match the phenotype pattern in one of the last k regions, i.e., $\text{pMG}(t) \in \mathcal{P}^{\ell, \tau, \tau_{N-k+1}, \dots, \tau_N}$ where k is a modeling choice.

From now on we will assume that the chemical gradient graph $G = (V; E; S; T; q)$ comes equipped with the designated set of starting nodes S and stopping nodes T .

Definition 5.3. Let $G = (V; E; S; T; q)$ be the chemical gradient graph. A developmental path is a path $p_1 \tilde{N} \dots \tilde{N} p_k$ in G such that

1. $p_1 \in S$ and $p_k \in T$, and
2. $\text{pMG}(p_1) \tilde{N} \dots \tilde{N} \text{pMG}(p_k)$ is a path in the phenotype pattern graph.

By construction, any path in the chemical gradient graph will be a monotone path with respect to factor graph layers for each gene R_N affected by an external variable. Our goal is to quantify features of the set of all developmental paths to characterize the robustness of the developmental program as predicted across network models.

Notice in Definition 5.2 we allow starting nodes to be in the first spatial regions and stopping nodes to be in the last regions to account for boundary conditions that may not be included in the model. While it would be ideal to find paths in the chemical gradient graph that follow the entire phenotype pattern, we found in the gap gene network of *D. melanogaster* example that nodes with the annotated MG for regions R_1 and R_8 are often disconnected from nodes with the annotated MG for regions R_2 through R_7 . We hypothesize that this is a consequence of additional regulation of gene expression in these regions by genes from the positions external to the A-P positions 35-70% where gap genes are active and that are modeled in this thesis, see Figure 4.1. In particular, there are gene-protein interactions in the anterior between 0-30% and region R_1 [19] and in the posterior between 80-100% and region R_8 [18, 2]. The lack of accounting for these boundary regulations may impact the ability of network models consisting of only trunk gap genes to recapitulate the data at the extremes of the A-P axis.

5.3 Condensed Chemical Gradient Graph

The DSGRN parameter graph size grows rapidly with the size of the network [6]. For example, networks with four nodes and eight edges have between 1.440 million and 23.064 million parameter graph nodes. Unfortunately, this means that the chemical gradient graph can be quite large. For example, the chemical gradient graph of the StrongEdges network has over 1.4 million nodes and 12 million edges. Graph algorithms such as path-finding rapidly reach computational limits in common chemical gradient graph sizes. Hence, directly finding all developmental paths is impractical. To overcome this limitation, we take two actions.

- ^ First, we consider only strict phenotypes, i.e., monostable fixed points, when constructing the chemical gradient graph. This is a conservative decision because it requires the elements of the phenotype pattern to be as dynamically stable as possible.
- ^ Second, we contract the chemical gradient graph using strongly connected components and annotated MGs into a condensed chemical gradient graph. Then we construct a subgraph of the condensed chemical gradient graph that contains all developmental paths, called the path graph, and study its structure rather than computing all developmental paths directly.

Consider the chemical gradient graph $G = (V; E; S; T; q)$. We define an equivalence relation on the set of vertices V by the requirement that they lie in the same strongly connected component of G and that the corresponding Morse graphs are the same.

Definition 5.4. Let $G = (V; E; S; T; q)$ be the chemical gradient graph and $H = (V; E')$ be a subgraph of G where $E' = \{ (u; v) \in E \mid MG(u; v) = MG(v; u) \}$. Additionally, let \mathcal{H} be the collection of strongly connected components of H (see Definition 2.2). Define the following equivalence relation over V

$$u \sim v \text{ if and only if } u; v \in \mathcal{H} \text{ for some } \mathcal{H} \in \mathcal{H} :$$

We say

$$V_u = \{ v \in V \mid v \sim u \}$$

is a strong MG equivalence class of V and call u the representative of V_u :

Definition 5.5. Let $G = (V; E; S; T; q)$ be the chemical gradient graph. We construct a weighted directed graph named the condensed chemical gradient graph $cG = (cV; cE; W; q)$ as follows. The nodes cV are the collection of all strong MG equivalence classes V_u of V , i.e.,

$$cV = \{ V_{u_1}; V_{u_2}; \dots; V_{u_k} \}$$

where $V = \bigcup_{i=1}^k V_{u_i}$. Additionally, there is an edge $(V_{u_i}; V_{u_j}) \in E$ for $i < j$ if and only if there exist nodes $u \in V_{u_i}$ and $v \in V_{u_j}$ such that $(u; v) \in E$. Let $M_{i;j}$ be the number of edges from any node in V_{u_i} to any node in V_{u_j} , i.e.,

$$M_{i;j} = |\{(u; v) \in E \mid u \in V_{u_i} \text{ and } v \in V_{u_j}\}|$$

and let N_i be the total number of edges from nodes in V_{u_i} to nodes outside of V_{u_i} , i.e.,

$$N_i = |\{(u; v) \in E \mid u \in V_{u_i} \text{ and } v \notin V_{u_i}\}|$$

Then we assign $(V_{u_i}; V_{u_j}) \in E$ the weight

$$W(V_{u_i}; V_{u_j}) = \frac{M_{i;j}}{N_i};$$

making G a weighted directed graph with weights in the range $[0, 1]$. Lastly, we say a node $V_{u_i} \in V$ is a starting or stopping node if there exists $u \in V_{u_i}$ such that $u \in S$ or $u \in T$, respectively. We label the set of starting nodes and stopping nodes of G by S and T respectively.

Figure 5.1: (left) The small nodes (colored blue, red and yellow) and black edges depict a directed graph G , where nodes of the same color indicate they have the same Morse graph. The large green nodes and thick green edges depict the condensation graph \bar{G} of (right). The condensed form of G where nodes represent clusters of strongly connected components that share the same Morse graph.

We note that there is a slight abuse of notation in the previous definition, where V_u denotes both a strong MG equivalence class and a node in G . We took a similar liberty for the nodes of the regulatory network RN and do so here again for clarity.

Notice that the condensed chemical gradient graph $c(G)$ is the condensation of the chemical gradient graph G (see Definition 2.3), where the strongly connected components of G are further decomposed by Morse graphs, see Figure 5.1 for a small example. The following lemma is an immediate consequence of Definition 5.1 and Remark 5.1.

Lemma 5.1. Let $G = (V; E; S; T)$ be the chemical gradient graph and V^i be a strongly connected component of G where V^i denotes the nodes of the subgraph G^i . Let v_j with $j = 1; \dots; m$ be the externally controlled network nodes with associated factor graph layers. Then for all $p; q \in V^i$ we have

$$L_j(p; q) = L_j(p; q)$$

for all $j = 1; \dots; m$.

The strong MG equivalence classes partition each strongly connected component of G and so Lemma 5.1 guarantees that each one can be unambiguously (though non-uniquely) labeled with a collection $L_j(p; q)$ for $j = 1; \dots; m$. This leads to an immediate Corollary.

Corollary 5.1. If $V_u \in S$ is a starting node in V and $p \in V_u$, then p is a starting node in $S \in V$. Similarly for stopping nodes.

Proof. Since all $p; q \in V_u$ have matching factor graph layers by Lemma 5.1 and matching Morse graphs by Definition 5.4, if one node in V_u satisfies the criteria of Definition 5.2, they all must.

Lemma 5.1 and Corollary 5.1 justify searching for developmental paths in the (much) smaller condensed chemical gradient graph $c(G)$ instead of in G , since paths in $c(G)$ follow the externally imposed gradients from starting nodes to stopping nodes.

Definition 5.6. Let $cG = (cV; cE; W; cS; cT)$ be the condensed chemical gradient graph. A condensed developmental path is a path $V_{u_1} \tilde{N} \tilde{N} V_{u_k}$ in cG such that

1. $V_{u_1} \in cS$ and $V_{u_k} \in cT$, and
2. $(\mu_1, \mu_2, \dots, \mu_k)$ is a path in the phenotype pattern graph.

In the worst case $|V| = |cV|$ but in practice $|cV|$ is much smaller than $|V|$. For example, while the StrongEdges chemical gradient graph has over 1.4 million nodes, its condensed chemical gradient graph has 14,832 nodes (see Figure 5.2). Importantly, we show that every condensed developmental path in the condensed chemical gradient graph contains at least one developmental path in the chemical gradient graph.

Figure 5.2: Condensed chemical gradient graph (all nodes) for FullConn (left) and StrongEdges (right). FullConn has 12,398 and 100,179 nodes and edges respectively. StrongEdges has 14,832 and 123,407 nodes and edges respectively. Purple, red, and green nodes depict nodes in the path graph, starting nodes are shown in green, and stopping nodes are shown in red. Graph visualization done using Gephi [3] with the OpenOrd layout algorithm [26].

Lemma 5.2. For every condensed developmental path $U : V_{u_1} \tilde{N} V_{u_2} \tilde{N} \dots \tilde{N} V_{u_k}$ there exists at least one developmental path

$$p_1 \tilde{N} p_2 \tilde{N} \dots \tilde{N} p_{j_1} \tilde{N} p_{j_2} \tilde{N} \dots \tilde{N} p_{j_2} \tilde{N} \dots \tilde{N} p_{j_n}^n$$

with the consecutive sets of vertices $\{p_1, \dots, p_{j_i}\} \cup PV_{u_i}$ in the chemical gradient graph.

Proof. Let $U : V_{u_1} \tilde{N} V_{u_2} \tilde{N} \dots \tilde{N} V_{u_k}$ be a condensed developmental path in the condensed chemical gradient graph $G = (V; E; W; CS; CT)$. Since there is an edge $V_{u_i} \tilde{N} V_{u_{i-1}}$, there must exist nodes $p_i \in PV_{u_i}$ and $q_{i-1} \in PV_{u_{i-1}}$ such that $p_i; q_{i-1} \in E$. Furthermore, by definition of a connected component, when $|V_{u_i}| \geq 1$ there exists a path between any two nodes in V_{u_i} . Thus, if $p_i; q_{i-1} \in E$ is an edge between V_{u_i} and $V_{u_{i-1}}$, and likewise $p_{i-1}; q_{i-2} \in E$ is an edge between $V_{u_{i-1}}$ and $V_{u_{i-2}}$, then for $|V_{u_{i-1}}| \geq 1$ there exists a path in $V_{u_{i-1}}$

$$P : p_i \tilde{N} q_{i-1} \tilde{N} v_{j_1} \tilde{N} \dots \tilde{N} v_{j_n} \tilde{N} p_{i-1} \tilde{N} q_{i-2}$$

for $v_{j_1}; \dots; v_{j_n} \in PV_{u_{i-1}}$. If $|V_{u_{i-1}}| = 1$, then $q_{i-1} = p_{i-1}$ so clearly there is a path

$$p_i \tilde{N} q_{i-1} = p_i \tilde{N} q_{i-2}$$

It follows that there exists a path in G through $V_{u_1}; V_{u_2}; \dots; V_{u_k}$, of the form

$$p_1 \tilde{N} q_2 \tilde{N} \dots \tilde{N} q_{i-1} \tilde{N} v_{j_1} \tilde{N} \dots \tilde{N} v_{j_n} \tilde{N} p_{i-1} \tilde{N} q_{i-2} \tilde{N} \dots \tilde{N} p_k$$

By Definition 5.6, path U induces a path $p \text{MG} p_{u_1} q q \tilde{N} p \text{MG} p_{u_2} q q \tilde{N} \dots \tilde{N} p \text{MG} p_{u_k} q q$ in the phenotype pattern graph. Observe that since $u_i; p_i \in PV_{u_i}$ and $u_{i-1}; q_{i-1}; v_{j_1}; \dots; v_{j_n}; p_{i-1} \in PV_{u_{i-1}}$ then $\text{MG} p_{u_i} q = \text{MG} p p_i q$ and

$$\text{MG} p_{u_{i-1}} q = \text{MG} p q_{i-1} q = \text{MG} p v_{j_1} q \dots \text{MG} p v_{j_n} q = \text{MG} p p_{i-1} q$$

Thus, $p \text{MG} p_{u_i} q q \tilde{N} p \text{MG} p_{u_{i-1}} q q$ implies

$$p \text{MG} p p_i q q \tilde{N} p \text{MG} p q_{i-1} q q \tilde{N} p \text{MG} p v_{j_1} q q \tilde{N} \dots \tilde{N} p \text{MG} p v_{j_n} q q \tilde{N} p \text{MG} p p_{i-1} q q$$

is a path in the phenotype pattern graph. It follows that

$$p \text{MG} p p_1 q q \tilde{N} p \text{MG} p p_2 q q \tilde{N} \dots \tilde{N} p \text{MG} p q_{i-1} q q \tilde{N} p \text{MG} p v_{j_1} q q \tilde{N} \dots \tilde{N} p \text{MG} p p_k q q$$

is a path in the phenotype pattern graph.

Lastly, U is a condensed developmental path s_{u_1} is a starting node and v_{u_k} is a stopping node in cG , and therefore by Corollary 5.1 $p_1 \in V_{u_1}$ is a starting node in $S \in V$ and $p_k \in V_{u_k}$ is a stopping node in $T \in V$. Therefore, the path P is a developmental path in G by Definition 5.3. Since U was arbitrary, we have shown there exists at least one developmental path for every condensed developmental path.

Lemma 5.3. Every developmental path $p_1 \tilde{N} \tilde{N} p_n$ in the chemical gradient graph can be projected uniquely onto a condensed developmental path $p_1 \tilde{N} \tilde{N} v_{u_k}$ in the condensed chemical gradient graph. In other words, there is a partition of p_1, \dots, p_n into k consecutive groups of vertices, each of which belongs to one component.

Proof. Let $G = (V; E; S; T)$ be the chemical gradient graph and let $cG = (cV; cE; W; cS; cT)$ be the condensed chemical gradient graph.

Let $p_1 \tilde{N} \tilde{N} p_n$ be a developmental path in G with induced phenotype pattern path

$$pMG(p_1, q) \tilde{N} \tilde{N} pMG(p_n, q) \quad p_1 \tilde{N} \tilde{N} p_n \quad (\text{for brevity}).$$

Our goal is to uniquely construct a condensed developmental path $v_{u_1} \tilde{N} \tilde{N} v_{u_k}$ in cG from \tilde{N} and \tilde{N} .

We partition the parameter nodes in \tilde{N} into sets A_{ij} that are the maximal sets of sequential elements in the path that all belong to the same strong MG equivalence class.

Formally,

$$A_{ij} = \{ p_i; p_{i-1}; \dots; p_j \}$$

where

1. $p_i = p_{i-1} \quad p_i = p_j$,
2. $p_i = p_{i-1}$ if $i > 1$, and

$pV_u; sq$ that satisfy

$$pMGpuqq \ s \quad (5.1)$$

which we call the matching graph \mathcal{M} , and denote the projection of \mathcal{H} onto the first component by $c\mathcal{H}$. In other words, $c\mathcal{H}$ is the subgraph of \mathcal{G} such that a path exists in $c\mathcal{H}$ if and only if that path induces a path in $\mathcal{P}\mathcal{P}\mathcal{G}$. We call $c\mathcal{H}$ the projected matching graph \mathcal{M} .

Given a regulatory node v affected by some monotone external variable e , we have that any path in the projected matching graph $c\mathcal{H}$ follows a monotone path in the factor graph of v (see Definition 3.3), i.e., the path must be consistent with external variable e . In the gap gene network, the external variables Bcd and Cad impose a requirement that Hb decreases and, at the same time, Kni increases. However, our construction of $c\mathcal{H}$ allows Hb to decrease entirely before Kni increases and vice-versa. This does not capture the biological reality where the maternal gradients change simultaneously, i.e., Hb is decreasing while Kni is increasing along the A-P axis of the embryo. We capture this behavior in our final graph construction, the path graph. However, we must first define what it means for external variables to be changing simultaneously in the context of DSGRN. In order to do that we develop the concept of a diagonal subgraph in a product of oriented graphs.

Definition 5.8. Consider $RN = \{v_i; \dots; v_m\} \subseteq V$, with each v_j affected by a corresponding external variable e_j . Let $L_j = |R_j| = |T\{v_j\}| \geq 1$, i.e., L_j is the number of factor graph layers for v_j . We call

$$r_{ij} := \max \left(\frac{L_j}{L_i}, \frac{L_i}{L_j} \right)$$

the size ratio between v_i and v_j .

For example, in the StrongEdges network, we have $L_{Hb} = 5$ and $L_{Kni} = 13$ where L_{Hb} and L_{Kni} denote the number of factor graph layers for Hb and Kni respectively. Then

$$r_{Hb;Kni} = \max \left(\frac{L_{Hb}}{L_{Kni}}, \frac{L_{Kni}}{L_{Hb}} \right) = 3$$

Returning to the general case, let two regulatory nodes $v_i, v_j \in \{v_1, \dots, v_m\}$. In order to simplify notation we set $r = r_{ij}$. The number r will be used to describe a subregion within a rectangle $R = [0, a_i] \times [0, a_j] \subseteq \mathbb{Z}^2$, where $a_i \in \mathbb{Z}$, $a_j \in \mathbb{Z}$. To be explicit we assume $a_i \neq a_j$.

When $\text{sign}(c_i) = \text{sign}(c_j)$ then we define a neighborhood of a diagonal in \mathbb{R}^2 , $S_{ij}^d \subseteq \mathbb{R}^2$ by

$$S_{ij}^d : \{ (x, y) \in \mathbb{R}^2 \mid x \in [0, a_i] \text{ and } \frac{1}{r}x - \frac{r-1}{r} \leq y \leq \frac{1}{r}x + a_j - \frac{r-1}{r} \} \quad (5.2)$$

Note that the lower bound of S_{ij}^d is a line passing through point $(0, 0)$ and the upper bound through the point $(a_i - r + 1, a_j)$ both with the same slope $\frac{1}{r}$, see Figure 5.3(left).

For the case when $\text{sign}(c_i) = -\text{sign}(c_j)$ we set

$$S_{ij}^a : \{ (x, y) \in \mathbb{R}^2 \mid x \in [0, a_i] \text{ and } \frac{1}{r}x + a_j - \frac{r-1}{r} \leq y \leq \frac{1}{r}x + a_j - \frac{r-1}{r} \} \quad (5.3)$$

see Figure 5.3(right).

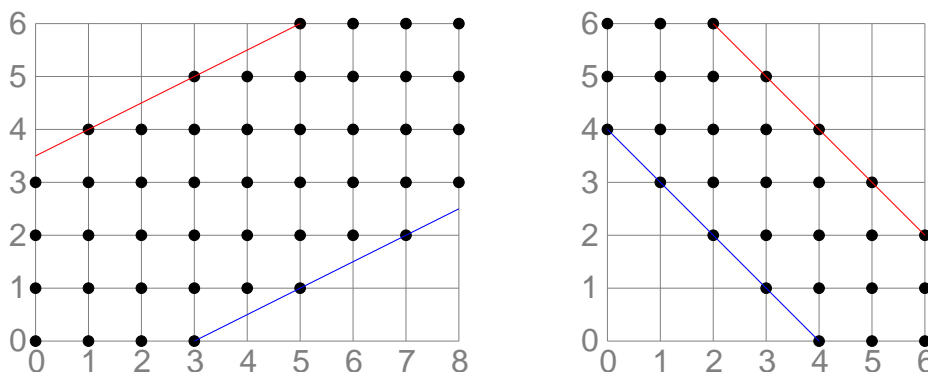


Figure 5.3: Example of sets S_{ij}^d and S_{ij}^a . (left) All black points are the points in the set S_{ij}^d for $a_i = 8$, $a_j = 6$ and $r = 2$ and (right) S_{ij}^a with $a_i = a_j = 6$ and $r = 1$. In both grids, the red lines depict the upper bound of y in S_{ij}^d and S_{ij}^a while the blue lines depict the lower bound of y .

Definition 5.9. Consider $RN = (V, E)$ and regulatory nodes $v_1, \dots, v_m \in V$, with each v_j for $j = 1, \dots, m$ affected by a corresponding external variable q_j and let \mathcal{H} be the graph

as constructed in Definition 5.7. The diagonal subgraph D is a node-induced subgraph of cH where a node $v_p \in cH$ belongs to D if and only if the corresponding factor graph layers satisfy

$$L_i \neq L_j \iff \begin{cases} S_{i,j}^d & \text{if } \text{sign}(c_i^1) = \text{sign}(c_j^1) \\ S_{i,j}^a & \text{if } \text{sign}(c_i^1) \neq \text{sign}(c_j^1) \end{cases}$$

for all pairs $v_i, v_j \in cV$; $v_m, v_n \in cV$ where $L_i \neq L_j$:

We restrict our attention to paths in the diagonal subgraph to exclude paths that are not consistent with our interpretation of a simultaneous change in the external variables. For our final construction, we further enforce that nodes in D must be connected to the collection of starting and stopping nodes.

Definition 5.10. Let $cG = (cV; cE; W; cS; cT)$ be the condensed chemical gradient graph, and consider the diagonal subgraph D of cH . A node $v_u \in D$ has terminal reach if one of the following is satisfied.

1. If $v_u \in cS \cup cT$, then there is both a path in D from at least one $v_s \in cS$ to v_u and a path in D from v_u to at least one $v_t \in cT$.
2. If $v_u \in cS$, then there is a path in D from v_u to at least one $v_t \in cT$.
3. If $v_t \in cT$, then there is a path in D from at least one $v_s \in cS$ to v_u .

Definition 5.11. The path graph $P = (V; E)$ is the node-induced subgraph D where we have removed all nodes v_u (and incident edges) if v_u does not have terminal reach in D . The edge $(v_u; v_w) \in E$ inherits the weight of the edge $(v_u; v_w) \in cE$. A node in V is a starting (stopping) node for P if it is a starting (stopping) node in cG .

The next theorem summarizes our construction.

Theorem 3. Every path in the path graph \mathcal{P} from a starting node to a stopping node is a condensed developmental path with simultaneous changes in external variables. Moreover, every developmental path in G with simultaneous changes in external variables can be projected onto a path in \mathcal{P} .

We showed that every condensed developmental path c in \mathcal{C} represents a developmental path in G and that every developmental path in G can be projected uniquely onto a condensed developmental path in \mathcal{C} . Since the path graph contains all condensed developmental paths that satisfy the biologically motivated constraint of simultaneous external variable change, it is sufficient to analyze the path graph in order to understand the structure of the developmental paths in G . See Figure 5.4 for a visualization of \mathcal{P} for both the example gap gene models FullConn and StrongEdges.

Figure 5.4: Path graph (all nodes) for FullConn (left) and StrongEdges (right). Colored by optimum 2-clustering separating starting nodes (green) and stopping nodes (red). FullConn has 1,576 and 5,767 nodes and edges respectively. StrongEdges has 2,393 and 10,037 nodes and edges respectively. Note that visually one might say that FullConn has a larger bottleneck than StrongEdges which we found not to be the case. This is due to the lack of visualization of edge weights, as well as a two-dimensional graph layout of a multi-dimensional graph. Recall any path from green to red nodes is a matching developmental path. Graph visualization done using Gephi [3] with the OpenOrd layout algorithm [26].

CHAPTER SIX

ROBUSTNESS SCORES

In this chapter, we use the fact that every path in a path graph $P = (V; E)$ matches spatial data represented by the phenotype pattern graph. Our quantification of robustness will rely on features of the shape and wiring of P , the extent of P lifted into the chemical gradient graph G , and the attractiveness of P as a subgraph of condensed chemical gradient graph cG . For the last, we assume that a random perturbation that redirects a developmental path out into cG is undesirable. We assert that a high likelihood that such a perturbation does not permanently divert paths out of P is a sign of a network model that robustly matches the data. With this in mind, we say that a network model is robust to perturbations if each of the following properties are satisfied.

- (P1) The path graph does not contain bottlenecks. Informally, a bottleneck in the path graph is a node, or a set of nodes, where a significant portion of perturbations result in a path in $cG \setminus P$. This property measures the fragility of the collection of developmental paths in the path graph.
- (P2) P is an attractor within the condensed chemical gradient graph. This means that if a local perturbation of a node in P leads to a node q outside of P , then paths starting at q will re-enter P after a few transitions. This permits the resumption of the phenotype pattern after a local break.
- (P3) A path in P is unlikely to be perturbed in such a way as to cause portions of the phenotype pattern to be skipped. This means that if there is a local perturbation of a node in P to another node in P , then the new path will still be a proper developmental path.

(P4) The nodes and edges of P constitute a large portion of nodes and edges of the chemical gradient graph G when P is lifted back into G . This means that for any given sequence of parameter changes that respects the external variables, the corresponding sequence of phenotypes is highly likely to reproduce the phenotype pattern.

Figure 6.1: Example of a path graph P that illustrates properties P1-4. (left) All nodes and edges represent G , the green nodes and edges are P . The nodes of P are annotated with a number $n = 1, \dots, 8$ indicating that the Morse graph of the node represents the data region R_n . Any path of green nodes and edges from the node labeled 1 to the node labeled 8 is a matching developmental path. Node 5 represents a bottleneck, the edge from 2 to 4 represents region skipping and any blue edge emanating from a green node contradicts attractivity of P within G . (right) All nodes and edges represent G , the green nodes and edges are P when lifted back into G . The lifted path graph of this example only constitutes 29% of the nodes and edges of G .

An example of a bottleneck in the path graph can be seen in Figure 6.1. In this figure, all nodes and edges represent G , with the green nodes and edges representing P . Additionally, the nodes of P are annotated with a number $n = 1, \dots, 8$ indicating the Morse graph the node would represent for data region R_n . Hence, any path from the node labeled 1 to the node labeled 8 consisting of only green nodes and edges would be a matching developmental path, and thus match the data. In this example, the node labeled 5 is the only node in the graph representing data for region R_5 . Thus, any matching developmental path must contain this

node. The issue arises when considering a local perturbation of a path near node 5, which will always result in the path having to exit P . The blue edge between the nodes labeled 2 and 4 shows an example of region skipping. Here, a matching developmental path could be perturbed in such a way as to take that edge, resulting in a path that matches the data for each region except region R_3 and is no longer a matching developmental path. Lastly, this example also shows issues with the attraction σ where there are blue edges from green nodes.

Lets consider properties P1-4 in context of the $D. melanogaster$ example. A path graph having property P1 suggests that any perturbation will still result in the proper development of the embryo. For spatially localized perturbations, property P2 would imply that the new path is still a proper development path and property P3 would allow for natural development of the embryo with only local laws. Finally, having property P4 means that chemical gradients Bcd and Cad robustly give rise to the phenotype pattern. We now introduce scores to quantify properties P1-4.

6.1 Bottlenecks Scored by Optimized Weighted Cut (OWCut) of the Path Graph

A common understanding of a bottleneck is a restriction point for traffic. In our case, the path graph describes a set of developmental paths that represent a sequence of parameter changes consistent with external variables that recapitulate qualitative observations about variable expression across spatial regions. A bottleneck indicates a set of parameter regions where a random perturbation will likely disrupt the phenotype pattern. Our goal is to develop a mathematical definition of a bottleneck that captures this behavior in the setting of a directed weighted graph (see Definition 2.1). To do so, we utilize the concept of weighted graph cut from [27].

Definition 6.1. Let W be the weight matrix of G . Let D be a diagonal matrix where

$$D_i = \begin{cases} \sum_{v_j \in V} w_{ij} & \text{if } v_j \in V \\ 0 & \text{otherwise.} \end{cases}$$

where $w_{ij} \in \mathbb{R}$. Notice if all edges have weight 1, then $\sum_{v_j \in V} w_{ij}$ is the out-degree of v_i . We call D the weighted out-degree matrix of the graph G [27].

Definition 6.2. Let $\{C_1, C_2, \dots, C_K\}$ be a collection of disjoint subsets of V , i.e., $C_k \cap C_j = \emptyset$ for $k \neq j$ and $\bigcup_{k=1}^K C_k = V$. If

$$C_1 \cup C_2 \cup \dots \cup C_K = V$$

then $\{C_1, C_2, \dots, C_K\}$ is a K -clustering of G . We call

$$D_k = \sum_{v_i \in C_k} D_i$$

the degree of cluster k [27], where D_i was defined in Definition 6.1.

We associate to each set of nodes $C_j \subseteq V$ all the edges $(u, v) \in E$ that connect vertices within C_j . K -clustering imposes a cut on the graph G ; the cut contains the set of edges in E that connect vertices in different clusters.

Definition 6.3. Let W and D be the weight and weighted out-degree matrices of weighted directed graph G respectively and let $\{C_1, C_2, \dots, C_K\}$ be a K -clustering of G . Then the weighted cut of C is given by [27]

$$WCut(C) = \sum_{k=1}^K \frac{1}{D_k} \sum_{v_i \in C_k} \sum_{v_j \in C_l} w_{ij} \mathbb{1}_{k \neq l}$$

Further, let

$$s_k = \sum_{v_i \in C_k} \sum_{v_j \in C_k} w_{ij} \quad \text{and} \quad d_k = \sum_{v_i \in C_k} \sum_{v_j \in V} w_{ij}$$

Notice that s_k is the sum of weights for every edge in the node-induced subgraph of G with node set C_k , denoted $G[C_k]$, while d_k is the sum of weights of edges departing $G[C_k]$.

If each $w_{i,j}$ represents a probability of taking the edge from node v_i to v_j , then in general a path starting in C_k has a higher probability of staying in C_k whenever s_k is higher than d_k .

Definition 6.4. Let $G = (V; E; W)$ be a weighted directed graph with K -cluster C . Given a cluster $C_k \in C$, we say that G has a bottleneck from C_k into $V \setminus C_k$ whenever $s_k < d_k$. i.e., whenever a path starting in C_k has a better chance of staying in C_k than it does of departing C_k .

We detect the existence of bottlenecks in a graph and score their strength using the weighted cut.

Theorem 4. Let $G = (V; E; W)$ be a weighted directed graph with weights $w_{i,j} \in [0, 1]$ such that $w_{i,j} \geq 0$ and $\sum_{j \in V} w_{i,j} = 1$. Given some cluster C_k of G , if

$$W \text{Cut}(C_k; V \setminus C_k) \geq \frac{1}{2}$$

then G has a bottleneck from C_k into $V \setminus C_k$.

Proof. Let $G = (V; E; W)$ be a weighted directed graph with weights $w_{i,j} \in [0, 1]$, let D be the weighted out-degree matrix, and let $C = \{C_1, \dots, C_K\}$ be a K -clustering of G . For every k notice that

$$\sum_{v_i \in C_k} \sum_{v_j \in V} w_{i,j} = \sum_{v_i \in C_k} \sum_{v_j \in C_k} w_{i,j} + \sum_{v_i \in C_k} \sum_{v_j \in V \setminus C_k} w_{i,j} = s_k = d_k;$$

where the first equality is because $C_k \cup (V \setminus C_k) = V$. Then

$$D_k = \sum_{v_i \in C_k} \sum_{v_j \in V} w_{i,j} = s_k = d_k$$

with equality when $D_k = 0$. We can also express $W \text{Cut}(C)$ of the clustering in terms of weights of internal edges s_k and external edges d_k .

$$\begin{aligned}
 W_{Cut}(C) &= \sum_{k=1}^K \frac{1}{D_k} \sum_{i \in C_k} \sum_{j \in V \setminus C_k} W_{i,j} \\
 &= \sum_{k=1}^K \frac{1}{D_k} D_k s_k \\
 &= \sum_{k=1}^K \frac{s_k}{D_k} \\
 &= \sum_{k=1}^K \frac{s_k}{s_k d_k} :
 \end{aligned}$$

We note that

$$1 - \frac{s_k}{s_k d_k} = \frac{1}{2} \text{ if and only if } d_k = s_k :$$

Applying the W_{Cut} to clustering consisting of two clusters $C; V \setminus C$, we observe that if $W_{Cut}(C; V \setminus C) = \frac{1}{2}$; then both $d_1 = s_1$ and $d_2 = s_2$. Therefore there is a bottleneck between C and $V \setminus C$.

Notice that if the cluster C_k has no edge to $V \setminus C_k$ then $d_k = 0$ and we have

$$1 - \frac{s_k}{s_k d_k} = 0 :$$

On the other hand if there are no edges connecting nodes within C_k then $s_k = 0$ and

$$1 - \frac{s_k}{s_k d_k} = 1 :$$

Therefore the closer $W_{Cut}(C; V \setminus C)$ is to zero, the stronger the bottleneck between C and $V \setminus C$.

Thus, to score property P1 we will find the optimal 2-clustering of the path graph P that minimizes the W_{Cut} of P , with the additional condition that one cluster contains all the starting nodes while the other contains all the stopping nodes. We compute an optimal 2-clustering as follows using methods inspired by the normalized Laplacian from [27] and a grouping algorithm for image segmentation [37] based on spectral clustering. Let L and D be the weighted matrix and weighted degree matrix of P , respectively. Computing the

Hermitian part of the Laplacian $L = \frac{1}{2}(D + W + W^T)$ and normalizing by D [27], we have

$$L : D^{-\frac{1}{2}} H p L \phi D^{-\frac{1}{2}} = \frac{1}{2} D^{-\frac{1}{2}} p 2 D^{-\frac{1}{2}} W + W^T \phi D^{-\frac{1}{2}} :$$

The key insight from spectral theory of harmonic maps is that while the smallest eigenvalue of L has the eigenvector with positive entries, the eigenvector corresponding to the second eigenvalue has entries of both signs, and the nodes corresponding to each sign form two clusters which minimize $WCut$. Since the value zero does not have a privileged position in our matrix L , we follow [37] to find an optimum value to define the clusters. Let $x = (x_1, x_2, \dots, x_{|V|})$ be the eigenvector corresponding to the second smallest eigenvalue of L . We reorder elements in x in ascending order and let $v_1, v_2, \dots, v_{|V|}$ be the corresponding ordering of nodes in V . Consider a 2-clustering of nodes $C_1^i = \{v_1, \dots, v_i\}$ and $C_2^i = \{v_{i+1}, \dots, v_{|V|}\}$ constructed by partitioning nodes using $x_{v_i} = (x_{v_{i-1}} + x_{v_{i+1}})/2$ as a splitting point. We say that C_1^i and C_2^i satisfy the starting and stopping node condition if

$$(S \times C_1^i = S \text{ and } T \times C_2^i = T) \text{ or } (T \times C_1^i = T \text{ and } S \times C_2^i = S) \quad (6.1)$$

Then we define an optimum two clustering as the index i such that [27, 37]

$$WCut(pC_1^i; C_2^i) = \min_i (WCut(pC_1^i; C_2^i) \mid C_1^i; C_2^i \text{ satisfy (6.1)})$$

We denote the $WCut$ of the optimum 2-clustering of P (with starting/stopping node conditions satisfied) by

$$OWCut(pP) = WCut(pC_1; C_2)$$

We remark that the $OWCut(pP)$ is not computable for some path graphs P because it is not possible to separate the starting and stopping nodes into separate clusters using this method. See Appendix Figure 5.4 for a visualization of the optimal clustering of P for both Fullconn and StrongEdges.

6.2 Using Absorbing Markov Chains (AMC) to Score Leak (P2) and Skip (P3)

To score properties P2 and P3, we will score how likely a random path that starts in the path graph $P = (V; E)$ will veer away from the path graph within the condensed chemical gradient graph $cG = (cV; cE; W; cS; cT)$. By construction, P is a subgraph of cG , i.e., $V \subseteq cV$ and $E \subseteq cE$. We will consider two types of edges in $cE \setminus E$ that may lead to different disruptions of the phenotype pattern. Let

$$O = \{ (V_u; V_v) \in cE \mid V_u \in P \text{ and } V_v \notin P \}$$

$$J = \{ (V_u; V_v) \in cE \mid V_u, V_v \in P \text{ and } (V_u; V_v) \notin E \}$$

The edges in O capture paths that leave the path graph P ; this represents a leak from the set of developmental paths to those which do not recapitulate the observed phenotype pattern. On the other hand, both vertices of the edges in J lie in V , but the edge is not in E . Therefore edges in J represent paths that skip a portion of the phenotype pattern. We say an edge $(V_u; V_v) \in cE$ is a leak edge if $(V_u; V_v) \in O$ and a skip edge if $(V_u; V_v) \in J$.

We use absorbing Markov chains (see Section 2.3.1) to quantify the amount of leak from P into cG and skip within P .

Definition 6.5. Let $P = (V; E)$ be the path graph of the condensed chemical gradient graph $cG = (cV; cE; W; cS; cT)$. The absorbing Markov chain expansion of P , denoted by $AMC(P; l; s; U; M)$ with transition matrix W , is defined as follows. The nodes are $U = V \cup \{l; s\}$, where l and s are nodes that represent all states that are targets of edges in O and J , respectively. Consider the following sets of weighted edges

$$E = \{ (V_{u_i}; V_{u_j}; w_{ij}) \mid (V_{u_i}; V_{u_j}) \in E \}$$

$$O = \{ (V_{u_i}; l; w_{ij}) \mid (V_{u_i}; V_{u_j}) \in O \}$$

$$J = \{ (V_{u_i}; s; w_{ij}) \mid (V_{u_i}; V_{u_j}) \in J \}$$

where w_{ij} is the weight of edge (i, j) in G . Then the set of edges of $AMC(p; l; s)$ is

$$M = \{ (i, j) \in E(G) \mid i \in P, j \in P \cup \{l, s\} \}$$

The entries w_{ij} in M are given by $w_{ij} = w_{ij}$ for each $(i, j) \in M$ and 0 otherwise.

The interpretation of edge weights in G as transition probabilities allows us to view $AMC(p; l; s)$ as a Markov chain. It is easy to see that the transition matrix of $AMC(p; l; s)$ satisfies the Markov property. Observe that the stopping nodes P , along with the nodes l and s , are the absorbing nodes of $AMC(p; l; s)$. Then the probabilities p_l and p_s are the probability of a random walk in $AMC(p; l; s)$ ending in nodes l or s from a starting node. By construction of $AMC(p; l; s)$, p_l and p_s are then the probability of a random walk beginning at a starting node of P and leaving P or skipping a region respectively. The probability p_l quantifies the lack of attractiveness of P within G (property P2) while the probability p_s quantifies region skipping (property P3).

6.3 Size of Lifted Path Graph in Chemical Gradient Graph (P4)

Recall from Section 5.4 that the nodes of the path graph $\mathcal{P}(V; E)$ are strong MG equivalence classes of the chemical gradient graph $\mathcal{G}(V; E)$ and the edges represent collections of edges between these components. Let

$$V_P = \{ (v, P) \mid v \in V_u \text{ for some } V_u \in \mathcal{P}(V; E) \}$$

Similarly, for $(V_u; V_v) \in \mathcal{P}(V; E)$, consider the associated collection of edges $\mathcal{E}_{u,v}$

$$\mathcal{E}_{u,v} = \{ (x, y) \in E \mid x \in V_u, y \in V_v \}$$

and let

$$E_P = \bigcup_{(V_u; V_v) \in \mathcal{P}(V; E)} \mathcal{E}_{u,v}$$

Then the size of the lifting of the path graph into the chemical gradient graph is given by

$$gp^P; Gq = \frac{|V_P| + |E_P|}{|V| + |E|}.$$

The rationale for using $gp^P; Gq$ as an indicator of robustness is that if P lifts to a large subgraph of G , then the collection of (non-condensed) developmental paths that respect the external variables is also large. A larger collection of developmental paths means that perturbations are more likely to cause a shift to another developmental path, thus ensuring that the phenotype pattern is preserved despite the disruption.

6.4 Scoring

Let N be a set of regulatory networks and let $P \in N$. Additionally, let D denote the descriptive pattern describing the spatial data. Let P^N and G^N be the path graph and chemical gradient graph respectively for N and D . We combine the following bottleneck, path size, and leak and skip scores

$$B^N q = \text{OWCut}^P q, \quad PS^N q = gp^P; G^N q \text{ and } LS^N q = 1 - p_{pp} q - p_{sq} q$$

respectively, in such a way as to score the robustness of

First, we normalize each score to be between 0 and 1 to equalize their impacts on an overall score. For $f \in \{B, PS, LS\}$, let

$$\hat{f}^N q = \frac{f^N q - \min_{N, P^N} f^N q}{\max_{N, P^N} f^N q - \min_{N, P^N} f^N q}$$

be our normalization, then we give N the robustness score

$$S^N q = \frac{\hat{B}^N q + \hat{PS}^N q + \hat{LS}^N q}{3}.$$

Note that we choose to apply an equal weighting of each of the scores B , PS and LS (i.e., each have a weight of $\frac{1}{3}$) because we don't know which, if any, of the scores has the most impact on the robustness of the network.

CHAPTER SEVEN

RESULTS

We consider two candidate network models for *D. melanogaster* development: StrongEdges (introduced in Section 4.2) and FullConn (introduced in Section 2.2.3). We found that both networks were capable of capturing the protein expression data from regions R_2 to R_7 seen in Figure 4.1. The remainder of the results are dedicated to evaluating and comparing their robustness. While values of the robustness score for some networks N are hard to interpret in a physical sense, we can use these values to compare network models. In particular, we wish to compare the networks StrongEdges and FullConn with a class of random networks. If these networks are valid representations of the gap gene network (which is known to be robust) then they should, in theory, have higher robustness scores than the average randomly generated network. Additionally, we would like to know if any network properties impact a network's robustness score. To accomplish these tasks, we must define a set of random networks, as well as define network properties we wish to evaluate. Given that both StrongEdges and FullConn have four nodes (Hb, Gt, Kr, and Kni) and eight edges, we restrict our attention to networks with nodes Hb, Gt, Kr, and Kni as well as eight edges. These can be any combination of edges between the nodes, with either activating or repressing signs. There are 126,720 networks in this class, with DSGRN parameter graph sizes ranging between 1.44 and 23.064 million nodes. For computational reasons, we only consider networks with a DSGRN parameter graph size up to 4.32 million nodes. We note that FullConn and StrongEdges have DSGRN parameter graph sizes of 52 and 324 million nodes respectively, so this range allows comparison with networks that have both smaller and larger parameter graphs. This class has 58,366 networks, which will be our network population, denoted by N .

Calculating the score $s_{\mathcal{N}}^q$ for a single network \mathcal{N} takes between 5 and 30 minutes, with time heavily dependent on the chemical gradient graph size. Using 3 threads in parallel, with this number limited by memory constraints, the computation time for 100 networks takes approximately one day. Hence we expect computing a score for all 58,366 networks would take nearly a year with our available resources. Thus, we selected nearly 1000 networks from \mathcal{N} and computed the robustness score $s_{\mathcal{N}}^q$ (sampling details are described below). The ability to collect comprehensive data about the dynamics of such a large set of networks is a unique characteristic of the DSGRN approach.

We used a mixed random sampling method to generate our sample of networks from the network population \mathcal{N} . We started by collecting a sample of 752 networks from \mathcal{N} using a simple random sampling method, i.e, there was an equal probability of every network in \mathcal{N} being selected during the sampling process. We call this simple random sample of 752 networks the *baseline group* that we denote by \mathcal{B} . We then asked if the following features of a regulatory network impact the robustness score.

1. Subnetworks of the gap gene network from Verd *et al.* [48], as seen in Figure 7.1 (and Figure 2.2). Networks $\mathcal{N} \in \mathcal{P}(\mathcal{N})$ that satisfy this condition are said to have the feature Verd.
2. Subnetworks of the gap gene network from Reinitz [25], see Figure 7.1. Since the Reinitz gap gene network is a subgraph of Verd, any \mathcal{N} that is a subgraph of the Reinitz gap gene network is also a subgraph of Verd. We call this feature Reinitz. When a \mathcal{N} has the Verd feature but not the Reinitz feature, we say this \mathcal{N} has the strict Verd feature.
3. The ACDC 1-3 motifs from left to right in Figure 2.3. We call these the ACDC1, ACDC2 and ACDC3 features.

4. Networks that have all four repressing edges Hb to Kni, Kni to Hb, Gt to Hb and Hb to Gt, which are the edges with the most biological evidence. We call this the Ultra Strong feature, see Figure 7.1.
5. The number of repressing edges M . We call this integer-valued property the RE feature.
6. The number of negative and positive feedback loops M (see Section 2.1). These are the NFL and PFL features, respectively.

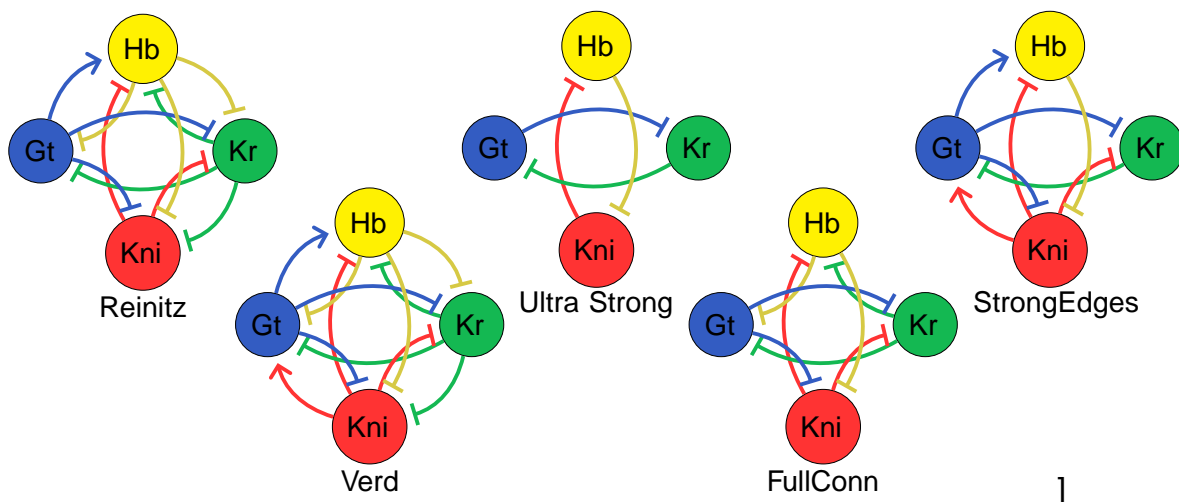


Figure 7.1: Networks labeled Reinitz and Verd are the gap gene networks as described in Reinitz et al. [25] and Verdet al. [48] respectively. Ultra Strong is a feature of interest and the FullConn and StrongEdge Networks are our networks we wish to compare to random networks.

We call categories (1)-(4) subgraph features and categories (5)-(6) number features. While all networks have number features, only 2122 networks M have at least one subgraph feature. We denote this set of networks by F . Given that there are less than 4% of networks with a subgraph feature, the random sample B did not produce many networks from F in the baseline group.

In order to evaluate how subgraph features impact our score, we used a stratified simple random sampling method to select more networks. Using the stratification of M into two

disjoint groups, F and $N \setminus F$ we randomly sampled 200 additional networks from \bar{N} , denoted B_F . We remark that $B \times B_F = H$. We call B_F the feature group and Table 7.1 shows a breakdown of the number of networks in N ; B ; and B_F , the number of networks in N we attempted to score, and the number of networks we were able to score.

Subgraph Feature	N						B						B _F						B ∪ B _F					
	Total		Disjoint				Sampled			Scored			Sampled			Scored			Total Scored					
ACDC1	468		387				7			7			44			43			50					
ACDC2	468		411				3			3			34			32			35					
ACDC3	468		387				9			7			41			40			47					
Ultra Strong	704		583				10			10			75			72			82					
Strict Verd	170		116				0			0			18			17			17					
Reinitz	85		37				1			1			14			13			14					
Count of Number Feature	RE		PFL		NFL		RE			PFL			NFL			RE			PFL			NFL		
	N	F	N	F	N	F	RE	PFL	NFL	RE	PFL	NFL	RE	PFL	NFL	RE	PFL	NFL	RE	PFL	NFL			
0	255	0	1680	0	2160	120	1	31	33	1	31	33	0	0	17	0	0	16	1	31	49			
1	1824	0	7680	128	5952	21713	119	82	13	113	81	0	17	21	0	16	20	13	129	101				
2	6060	0	14256	488	15888	75283	191	183	82	188	175	0	43	69	0	41	66	82	229	241				
3	12120	0	17712	687	18672	66841	227	214	137	218	208	0	57	53	0	56	52	137	274	260				
4	15690	119	12912	692	10512	28207	131	173	201	125	167	9	68	30	9	65	30	210	190	197				
5	13200	510	3168	87	4128	71179	42	51	172	42	49	48	9	9	47	9	9	219	51	58				
6	6924	850	720	36	1008	796	7	15	90	7	14	91	6	1	86	6	0	176	13	14				
7	2040	534	240	4	48	0	29	4	1	29	4	1	42	0	0	41	0	70	4	1				
8	255	109	0	0	0	0	3	0	0	3	0	0	10	0	0	10	0	13	0	0				
Total	58368						752			728			200			193			914					

Table 7.1: The sizes of the groups of networks in the statistical analysis. The pair of numbers for each subgraph feature under "N" is the population size of networks containing the specified subgraph feature in N and by construction in F (first number), and the number of networks containing only the specified subgraph feature (second number). The pairs of numbers for the number features are the population sizes of networks available in N and F respectively for the counts of edges/loops in the row. No networks with greater than 7 positive or negative loops exist in N . In columns labeled B and B_F , "Scored" are the subsets of the "Sampled" sets that could be scored. Under $B \cup B_F$, "Total Scored" indicates the population size of networks used in our statistical analysis. The integers beneath the number features for the sampled groups of networks indicate the counts of networks with each number feature in the specified sample.

Of the networks we attempted to score, only 11 were unable to reproduce the data, i.e., they did not contain a developmental path. One of these networks (network 21283 shown in Figure 7.3) was unable to reproduce the data due to having no stopping nodes in the chemical gradient graph. The other 10 had path graphs that became disconnected after imposing the requirement that developmental paths follow maternal gradient flow simultaneously (see Definition 5.9). Additionally, there were 19 networks where OWCut could not be calculated

with the condition that the starting and stopping nodes be in separate clusters. Due to not having a calculated score, we left these 30 networks out of our statistical analysis. Now we would like to answer the following questions:

1. Is FullConn or StrongEdges more robust than the average network from our population?
2. Is there evidence of a difference in robustness between our baseline and feature groups? Specifically, we ask if networks containing at least one of our noted features will be more or less robust than average.
3. Is there a relationship between any of the features and the network robustness score? Specifically, we want to know if there is evidence of a specific feature having an impact on the robustness score.

Figure 7.2 shows a summary of our analysis for both the baseline (grey) and the subgraph feature group (blue) B_F , together with means and the 95% mean confidence intervals for each of the normalized scores and the overall robustness score. The numerical values can be seen in Appendix Table A.01, along with the results for FullConn and StrongEdges. Additionally, see Appendix Figure B.01 for a summary of score results before applying the normalization defined in Section 6.4. We see that both FullConn and StrongEdges have robustness scores that lie outside of the 95% confidence intervals for the baseline group. In particular, FullConn exceeds the 95% mean confidence intervals for the baseline group in all robustness scores and exceeds the 95% mean confidence interval for the feature group in all scores but StrongEdges is a worse performer, with StrongEdges below the 95% mean confidence intervals in the baseline and feature groups, and the overall score StrongEdges below the 95% mean confidence interval in the feature group. Comparing the baseline and feature distributions, we see evidence that a network containing a subgraph feature, on average, has a higher robustness score than a random network, primarily

Figure 7.2: Violin plots comparing normalized robustness scores between the baseline (grey) and feature (blue) groups. White bars depict the 95% mean confidence intervals. The red dot and black square indicate the score for FullConn and StrongEdges respectively. Note in particular when the scores for FullConn and StrongEdges lie outside the 95% mean confidence intervals for each group.

due to higher performing LS and PS scores. Figure 7.3 shows the network topology of the best and worst scoring networks in both the baseline and subgraph feature groups.

To address question 3, we used multiple linear regression (MLR) on all networks scored, see Section 2.3.2 for model details. The subgraph and number features were set as the explanatory (independent) variables and the network score as the response (dependent) variable. All assumptions to ensure the validity of this model were checked, see Appendix A.

Our results are summarized in Figure 7.4. Each explanatory variable coefficient is depicted by a black dot, and the 95% confidence interval is depicted by the blue line on either side of the coefficients. For example, for S in the lower right panel, we can see that when a network contained ACDC2 as a subnetwork, we are 95% confident that the true mean of

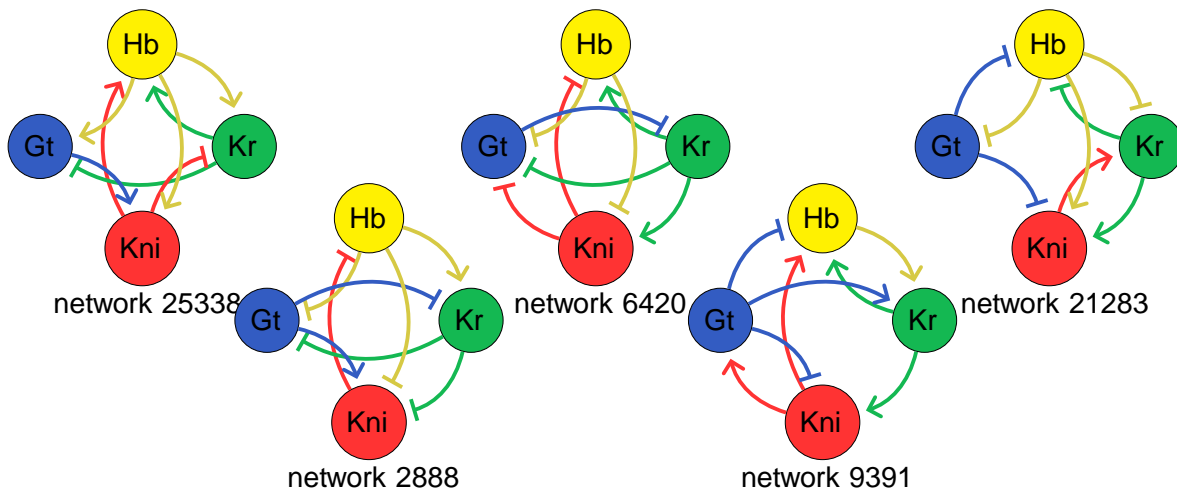


Figure 7.3: First 4 networks are ordered from highest to lowest score, with networks 25338 and 2888 being the best scoring networks for the Baseline and Feature group respectively and networks 6420 and 9391 being the worst scoring networks for the Feature and Baseline group respectively. Network 21283 is the only network that had an empty stopping set

the network score is increased by between 0.073 and 0.123 in our population, after adjusting for the additional possible presence of ACDC1, ACDC3, Ultra Strong, Strict Verd, Reinitz, NFL, PFL, and RE. Additionally, for each repressive edge added to a regulatory network, we are 95% confident that the true mean of the network score is increased by between 0.0008 and 0.0086 in our population, after adjusting for the presence of the remaining features. From this, we saw evidence that 5 of the 9 features (ACDC2, Reinitz, Strict Verd, RE and NFL) had an increasing or decreasing effect on our overall network robustness score. The impacts of feature groups on the component scores PS and LS are also shown.

In more detail, let F denote our set of explanatory variables. Given an explanatory variable $f \in F$, let $H_{0,f}$ be the null hypothesis, which states that there is no linear relationship between the network score and f , once we have accounted for all explanatory variables in F other than f . To determine if we can reject the null hypothesis, we use a t-statistic and a p-value. Since our model has 906 degrees of freedom (number of observations minus number of variables), then a t-statistic (denoted t_{906}) above 1963 or below -1963 is considered

Figure 7.4: Regression coefficients for each explanatory variable in our model and model intercept, plotted with 95% confidence intervals. See Appendix Table A.02 for coefficients, 95% confidence intervals, t and p-values for each of the response variables (top left), PS (top right), LS (bottom left) and S (bottom right).

significant at the 95% level, along with a p-value less than .05 [16]. During our analysis, we found that there was evidence for a positive linear relationship between network robustness and feature ACDC2, once we accounted for the remaining features; i.e., there was evidence against $H_{0;ACDC2}$ ($t_{906} = 7.4970$, p-value = 0.0001). That is, the presence of ACDC2 on average increased the score of any network of which it is a subnetwork. We also found evidence for positive linear relationships in Reinitzt ($t_{906} = 3.5587$, p-value = 0.0004), Strict Verd ($t_{906} = 2.9821$, p-value = 0.0029), and RE ($t_{906} = 2.3844$, p-value = 0.0173). On the other hand, we saw evidence for a negative relationship in NFI ($t_{906} = -5.8644$, p-value = 0.0001). Lastly, we saw little to no evidence against the null hypotheses for ACDC1 (t_{906}

= 1.4800, p-value = 0.1392), ACDC3 ($t_{906} = -0.4808$, p-value = 0.6308), Ultra Strong ($t_{906} = 0.5861$, p-value = 0.5580) or PFL ($t_{906} = -0.4890$, p-value = 0.6250).

Figure 7.4 shows the impact of the three constituent robustness scores as well. For example, we see evidence to suggest that there exists a positive linear relationship between the path size score P_S and feature Reinitz once we account for all other features ($t_{906} = 3.7818$, p-value = 0.0002). However, we see little to no evidence that there exists a linear relationship between Reinitz and the bottleneck score B ($t_{906} = 1.0178$, p-value = 0.3090), as well as the leak-skip score L_S ($t_{906} = 1.2341$, p-value = 0.2175), once we account for all other features. This suggests that the Reinitz feature impacted the overall robustness score by increasing the size of the lifted path graph in the chemical gradient graph.

CHAPTER EIGHT

DISCUSSION

In this manuscript, we reinterpreted the output of the network modeling tool DSGRN to accommodate a linear array of identical networks that are impacted by spatially varying external factors. We were motivated by the developmental program of *D. melanogaster*, particularly by the stabilizing influence of maternal protein gradients on gap gene network models in late-stage segmentation. We used the new modeling framework to quantify the robustness of various such models.

Our main mathematical contributions are three-fold. First, we conceptually reinterpreted the output of the DSGRN methodology to enable modeling of spatially arranged cells that are impacted by monotone control variables. This was done by proving that DSGRN factor parameter graphs can be represented as graded posets. Second, we developed a path graph based on the graded posets that permits a DSGRN network model to match spatial experimental data subject to constraints from monotone control variables. The path graph summarizes all the ways in which a network model is capable of matching the data under these constraints. Lastly, we developed evaluation criteria for the robustness of the match between model and data by devising three robustness scores that quantify the structural fragilities of the path graph. These structural fragilities can be interpreted as obstacles to correct development.

Our major biological contributions are a rank ordering of proposed gap gene network models in *D. melanogaster* according to robustness score, a quantification of their performance over random networks, and a characterization of the impact of various network motifs on network model performance. In particular, we showed that while it is common for a network model to be able to match experimental data, a network model inspired by Veri

al. [48] (FullConn) shows strong robustness scores compared to a random sample of network models. We also identified a motif (ACDC2) within FullConn that, on average, improves the robustness scores of network models that contain it.

The network FullConn is an alternative view of the dynamic module approach in Verd et al. [48], with which we showed that it is possible to model observed data using a single network functioning at different parameter regimes across spatial locations, as opposed to modeling the observed data using different networks across spatial locations. The FullConn network is a combination of the modules proposed by Verd et al., and our analysis showed that FullConn had a better robustness score than both the average random network with 4 nodes and 8 edges, as well as the average random network containing other subgraph features of interest. The FullConn network even had a higher robustness score than the StrongEdges network, which was constructed using only strong edges from the gap gene network from Figure 2.2. This suggests that the modules proposed by Verd et al. [48] are a reasonable hypothesis for dynamic control in the late-stage segmentation process of *Drosophila melanogaster*, although it is unnecessary to view them as distinct networks. Moreover, we found that networks that are subnetworks of the proposed (large) gap gene network from Verd et al. [48] and from Reinitz et al. [25], which is the same gap gene network but without the edge $Kni \rightarrow Gt$ (see Figure 7.1), have higher robustness scores, suggesting that both models contain subnetworks important to the function of the gap gene network. Furthermore, the motif ACDC2 had the most impact on our robustness score suggesting this motif may be particularly biologically relevant for robustness in the gap gene network.

We also found that nearly all of the randomly sampled networks with 4 nodes and 8 edges can reproduce the phenotype pattern derived from the developmental data between regions R_2 and R_7 . While our score allows rank-ordering these networks, it may be desirable to constrain the potential network models further. Our framework is capable of incorporating additional datasets that may help reduce the number of networks that fit the phenotype

pattern. In particular, measuring expression of the gap genes in embryos where the spatial expression of Bcd and Cad was experimentally manipulated could lead to, for example, non-diagonal developmental paths that any network model would be required to match along with the wild-type data, resulting in additional restrictions on network structure. A similar process of using additional data to reduce the space of hypotheses was used in the context of DSGRN models of yeast cell cycle network [10].

Moreover, while the biologically and mathematically motivated network models FullConn and StrongEdges scored well in comparison to the random sample, there were plenty of networks that optimized the robustness score even more. We hypothesize that optimizing for robustness is a constrained optimization problem, where factors such as evolutionary and environmental constraints may cause a network to be selected during evolution even if another network may provide more robustness for developmental or other highly conserved genetic programs.

The DSGRN approach that we present in this thesis is a powerful tool for the exploration of network models under different parameter regimes across spatial domains. It enables the comprehensive description of (coarse) dynamical behavior across parameter space, enabling the quantification of features such as robustness. Moreover, the computational efficiency permits the exploration of very large samples of network topologies, lending more credence to rank orderings of possible network models.

There is an immediate application of our methods to insects with a similar developmental system, such as embryonic development in *Episyrphus balteatus* [24] and *Megaselia abdita* [50]. We could also apply our work to other network models, such as the pair-rule gene network in *D. melanogaster*, where the gap gene protein concentrations determine pair-rule gene transcription [13]. Hence, in this model, the gap genes would be the external variables to the pair-rule gene network, though we would need to extend our work to non-monotone external variables. Finally, we can further extend our approach to modeling

late-stage dynamic shifts in domain boundaries along the A-P axis of gap gene protein concentrations.

REFERENCES CITED

- [1] Elizabeth Andreas, Breschine Cummins, and Tomàs Gedeon. Quantifying robustness of the gap gene network. *Journal of Theoretical Biology* 580:111720, 2024.
- [2] Maksat Ashyraliyev, Ken Siggins, Hilde Janssens, Joke Blom, Michael Akam, and Johannes Jaeger. Gene circuit analysis of the terminal gap gene huckebein. *PLoS Computational Biology* 5(10), 2009.
- [3] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks, 2009.
- [4] Attila Becskei and Luis Serrano. Engineering stability in gene networks by autoregulation. *Nature*, 405(6786):590{593, 2000.
- [5] Peter Crawford-Kahrl, Bree Cummins, and Tomàs Gedeon. Joint realizability of monotone boolean functions. *Theoretical Computer Science* 922:447{474, 2022.
- [6] Bree Cummins, Tomas Gedeon, Shaun Harker, and Konstantin Mischaikow. Dsgrn: Examining the dynamics of families of logical models. *Frontiers in Physiology*, 9:549, 2018.
- [7] Bree Cummins, Tomas Gedeon, Shaun Harker, Konstantin Mischaikow, and Kafung Mok. Combinatorial representation of parameter space for switching networks. *SIAM Journal on Applied Dynamical Systems* 15(4):2176{2212, 2016.
- [8] H de Jong, JL Gouze, C Hernandez, M Page, T Sari, and J Geiselmann. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull Math Biol*, 66(2):301{340, 2004.
- [9] Stefano Ermon, Carla Gomes, Ashish Sabharwal, and Bart Selman. Designing fast absorbing Markov chains. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 849{855. AAAI Press, 2014.
- [10] Erika Fox, Bree Cummins, William Duncan, and Tomàs Gedeon. Modeling Transport Regulation in Gene Regulatory Networks. *Bulletin of Mathematical Biology*, 84(8):89, July 2022.
- [11] Marcio Gameiro. Dynamic signatures generated by regulatory networks, 2018. Version 1.1.0. <https://github.com/marciogameiro/DSGRN> .
- [12] Scott Gilbert. *Developmental Biology* Sunderland (MA): Sinauer Associates, 6th edition, 2000.
- [13] Scott Gilbert and Michael Barresi. *Developmental biology* Oxford University Press, 7 edition, 2018.
- [14] L Glass and S a Kau man. Co-operative components, spatial localization and oscillatory cellular dynamics. *Journal of Theoretical Biology* 34(2):219{37, February 1972.

- [15] L Glass and S a Kau man. The logical analysis of continuous, non-linear biochemical control networks. *Journal of Theoretical Biology*, 39(1):103{29, April 1973.
- [16] Mark Greenwood. *Intermediate Statistics with R*, volume 3(1). Montana State University, 2022.
- [17] Vivian Irish, Ruth Lehmann, and Michael Akam. The *Drosophila* posterior group gene nanos functions by repressing hunchback activity. *Nature*, 338:646{648, 1989.
- [18] Johannes Jaeger. The gap gene network. *Cellular and Molecular Life Sciences*, 68(2):243{274, 01 2011.
- [19] Johannes Jaeger, Maxim Blagov, David Kosman, Konstantin Kozlov, Manu, Ekaterina Myasnikova, Svetlana Surkova, Carlos Vanario-Alonso, Maria Samsonova, David Sharp, and John Reinitz. Dynamical analysis of regulatory interactions in the gap gene system of *Drosophila melanogaster*. *Genetics*, 167(4):1721{1737, 2004.
- [20] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning: With applications in R*. Boston: Springer, 2 edition, 2021.
- [21] David Jobson. *Applied Multivariate Data Analysis: Regression and Experimental Design*. Springer-Verlag, 1st edition, 1991.
- [22] David Klarner. The number of graded partially ordered sets. *Journal of Combinatorial Theory*, 6(1):12{19, 1969.
- [23] Paul Lehrer and David Eddie. Dynamic processes in regulation and some implications for biofeedback and biobehavioral interventions. *Applied Psychophysiology and Biofeedback*, 38(2):143{155, 2013.
- [24] Ste en Lemke, Stephanie Busch, Dionysios Antonopoulos, Folker Meyer, Marc Domanus, and Urs Schmidt-Ott. Maternal activation of gap genes in the hover y episyrrhus. *Development*, 137(15):1709{1719, 2010.
- [25] Manu, Svetlana Surkova, Alexander Spirov, Vitaly Gursky, Hilde Janssens, Ah-Ram Kim, Ovidiu Radulescu, Carlos Vanario-Alonso, David Sharp, Maria Samsonova, and John Reinitz. Canalization of gene expression and domain shifts in the *Drosophila* blastoderm by dynamical attractors. *PLOS Computational Biology*, 5(3):1{15, 03 2009.
- [26] Shawn Martin, W. Michael Brown, Richard Klavans, and Kevin Boyack. *Openord: an open-source toolbox for large graph layout*. In *Electronic imaging* 2011.
- [27] Marina Meib and William Pentney. Clustering by weighted cuts in directed graphs. *SIAM International Conference on Data Mining* 2007.
- [28] Hans Meinhardt. Hierarchical inductions of cell states: a model for segmentation in *Drosophila*. *J Cell Sci (Supplement)*, 4:357{381, 1986.

- [29] Konstantin Mischaikow. Topological techniques for efficient rigorous computation in dynamics. *Acta Numerica*, 11:435{477, 2002.
- [30] Alexander Mitrophanov and Eduardo Groisman. Positive feedback in cellular control systems. *BioEssays* 30(6):542{555, 2008.
- [31] Christiane Nüsslein-Volhard, Hans Georg Frohnhöfer, and Ruth Lehmann. Determination of anteroposterior polarity in *Drosophila*. *Science* 238:1675{1687, 1987.
- [32] Christiane Nüsslein-Volhard and Eric Wieschaus. Mutations affecting segment number and polarity in *Drosophila*. *Nature*, 287(5785):795{801, 1980.
- [33] Jasmina Panovska-Gri ths, Karen Page, and James Briscoe. A gene regulatory motif that generates oscillatory or multiway switch outputs. *Journal of The Royal Society Interface*, 10(79):20120826, 2013.
- [34] Theodore J Perkins, Johannes Jaeger, John Reinitz, and Leon Glass. Reverse engineering the gap gene network of *drosophila melanogaster*. *PLOS Computational Biology*, 2(5):1{12, 05 2006.
- [35] Nicolas Privault. Understanding Markov chains: Examples and applications. *Mathematics Series*. Springer, 2nd edition, 2018.
- [36] Fred Ramsey and Dan Schafer. *The statistical sleuth: A course in methods of data analysis* Thomson Brooks/Cole, 3 edition, 2013.
- [37] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2000.
- [38] Marcia Simpson-Brose, Jessica Treisman, and Claude Desplan. Synergy between the hunchback and bicoid morphogens is required for anterior patterning in *drosophila*. *Cell*, 78(5):855{865, 1994.
- [39] El Houssine Snoussi. Qualitative dynamics of piecewise-linear differential equations: a discrete mapping approach. *Dynamics and stability of Systems* 4(3-4):565{583, 1989.
- [40] Alexander Spirov, Khalid Fahmy, Martina Schneider, Erich Frei, Markus Noll, and Stephan Baumgartner. Formation of the bicoid morphogen gradient: an mrna gradient dictates the protein gradient. *Development (Cambridge, England)* 136(4):605{614, 2009.
- [41] Richard Stanley. *Enumerative combinatorics Mathematics Series*. Cambridge: Cambridge Univ. Press, 2 edition, 2012.
- [42] Denis Thie ry and Rere Thomas. Qualitative analysis of gene networks. *Pacific Symposium on Biocomputing*. Pacific Symposium on Biocomputing 998:77{88, 1998.

- [43] René Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42:563{585, 1973.
- [44] René Thomas. Regulatory networks seen as asynchronous automata: A logical description. *Journal of Theoretical Biology*, 153:1{23, 1991.
- [45] William Thompson and James McNeal. Sales planning and control using absorbing Markov chains. *Journal of Marketing Research*, 4(1):62{66, 1967.
- [46] Berta Verd, Erik Clark, Karl Wotton, Hilde Janssens, Eva Jimenez-Guri, Anton Crombach, and Johannes Jaeger. A damped oscillator imposes temporal order on posterior gap gene expression in *Drosophila*. *PLoS Biology*, 16(2):1{24, 2018.
- [47] Berta Verd, Anton Crombach, and Johannes Jaeger. Dynamic maternal gradients control timing and shift-rates for *drosophila* gap gene expression. *PLoS Computational Biology*, 13(2):1{23, 2017.
- [48] Berta Verd, Nicholas Monk, and Johannes Jaeger. Modularity, criticality, and evolvability of a developmental gene regulatory network. *Life*, 8(2):243{274, 2019.
- [49] Charlotte Wang, Laura Dickinson, and Ruth Lehmann. Genetics of nanos localization in *Drosophila*. *Developmental Dynamics*, 199(2):103{115, 1994.
- [50] Karl R Wotton, Eva Jimenez-Guri, Anton Crombach, Hilde Janssens, Anna Alcaine-Colet, Ste en Lemke, Urs Schmidt-Ott, and Johannes Jaeger. Quantitative system drift compensates for altered maternal inputs to the gap gene network of the scuttle y *Megaselia abdita*. *eLife*, 4:e04785, 2015.
- [51] Karl Åström and Richard Murray. *Feedback Systems: An Introduction for Scientists and Engineers* Princeton University Press, 2021.

APPENDICES

APPENDIX A

MODEL RESULTS AND MLR VALIDITY

Results of the difference in means model used to assess if there is a difference between the average network in the baseline group versus the average network in the features group can be seen in Table A.01.

	Baseline Group (B) Results				Feature Group (B_F) Results				FullConn Results	StrongEdges Results
	mean	std dev	mean	95% CI	mean	std dev	mean	95% CI		
B_{Pq}	0.134	0.106	[0.127, 0.142]	0.139	0.120	[0.121, 0.156]	0.184	0.222		
B_{Spq}	0.352	0.162	[0.340, 0.364]	0.449	0.178	[0.424, 0.475]	0.381	0.387		
L_{Spq}	0.109	0.107	[0.102, 0.117]	0.155	0.116	[0.139, 0.172]	0.357	0.056		
S_{Pq}	0.199	0.078	[0.193, 0.204]	0.248	0.087	[0.235, 0.260]	0.307	0.222		

Table A.01: Mean, standard deviation, and confidence intervals for the random samples B and B_F and the specific networks FullConn and StrongEdges.

We provide the data of the multiple linear regression (MLR) used in Section 7 (see Table A.02) as well as verification of all assumptions needed to use the MLR model.

B_{Pq}	coef	std dev	t-value	p-value	95% ci	B_{Spq}	coef	std dev	t-value	p-value	95% ci
intercept	0.1542	0.0217	7.1	0.0001	[0.1116, 0.1969]	intercept	0.4358	0.03	14.5109	0.0001	[0.3768, 0.4947]
ACDC3	-0.0091	0.0167	-0.5416	0.5882	[-0.0419, 0.0238]	ACDC3	-0.0469	0.0232	-2.0243	0.0432	[-0.0923, -0.0014]
ACDC2	0.0497	0.0188	2.6433	0.0084	[0.0128, 0.0866]	ACDC2	0.2104	0.026	8.0959	0.0001	[0.1594, 0.2614]
ACDC1	-0.0115	0.0167	-0.6904	0.4901	[-0.0442, 0.0212]	ACDC1	0.0306	0.023	1.3308	0.1836	[-0.0145, 0.0758]
Ultra Strong	-0.0125	0.0137	-0.9152	0.3603	[-0.0393, 0.0140]	Ultra Strong	-0.0129	0.0189	-0.6841	0.4941	[-0.05, 0.0241]
Strict Verd	0.0923	0.0263	3.508	0.0005	[0.0407, 0.1439]	Strict Verd	0.0982	0.0364	2.7011	0.007	[0.0269, 0.1696]
Reinitz	0.0298	0.0293	1.0178	0.309	[-0.0277, 0.0872]	Reinitz	0.153	0.0405	3.7818	0.0002	[0.0736, 0.2325]
NFL	0.0007	0.0034	0.2073	0.8359	[-0.0061, 0.0075]	NFL	-0.0451	0.0048	-9.4607	0.0001	[-0.0544, -0.0357]
PFL	-0.0089	0.0035	-2.5902	0.0097	[-0.0157, -0.0022]	PFL	-0.005	0.0048	-1.0543	0.292	[-0.0144, 0.0043]
RE	0.0005	0.0028	0.1609	0.8722	[-0.0051, 0.006]	RE	0.014	0.0039	3.6062	0.0003	[0.0064, 0.0216]
L_{Spq}	coef	std dev	t-value	p-value	95% ci	S_{Pq}	coef	std dev	t-value	p-value	95% ci
intercept	0.0782	0.0219	3.5723	0.0004	[0.0352, 0.1211]	intercept	0.2227	0.0152	14.616	0.0001	[0.1928, 0.2526]
ACDC3	0.039	0.0169	2.311	0.0211	[0.0059, 0.0721]	ACDC3	-0.0056	0.0117	-0.4808	0.6308	[-0.0287, 0.0174]
ACDC2	0.0365	0.0189	1.9277	0.0542	[-0.0007, 0.0737]	ACDC2	0.0989	0.0132	7.497	0.0001	[0.073, 0.1248]
ACDC1	0.0327	0.0168	1.9507	0.0514	[-0.0002, 0.0657]	ACDC1	0.0173	0.0117	1.48	0.1392	[-0.0056, 0.0402]
Ultra Strong	0.0423	0.0138	3.0714	0.0022	[0.0153, 0.0692]	Ultra Strong	0.0056	0.0096	0.5861	0.558	[-0.0132, 0.0244]
Strict Verd	-0.0254	0.0265	-0.9589	0.3378	[-0.0774, 0.0266]	Strict Verd	0.055	0.0185	2.9821	0.0029	[0.0188, 0.0912]
Reinitz	0.0364	0.0295	1.2341	0.2175	[-0.0215, 0.0943]	Reinitz	0.0731	0.0205	3.5587	0.0004	[0.0328, 0.1134]
NFL	0.0018	0.0035	0.5261	0.5989	[-0.005, 0.0086]	NFL	-0.0142	0.0024	-5.8644	0.0001	[-0.0189, -0.0094]
PFL	0.0104	0.0035	2.9961	0.0028	[0.0036, 0.0172]	PFL	-0.0012	0.0024	-0.489	0.625	[-0.0059, 0.0086]
RE	-0.0004	0.0028	-0.1275	0.8986	[-0.0059, 0.0052]	RE	0.0047	0.002	2.3844	0.0173	[0.0008, 0.0086]

Table A.02: The regression coefficients, standard errors, t-values, p-values lower and upper confidence intervals for our MLR model for the bottleneck score (B_{Pq}), path size score (B_{Spq}), leak-skip score (L_{Spq}) and the robustness score (S_{Pq}).

First, checking multicollinearity between our explanatory variables, we consider the variance inflation factors (VIFs) [16], which are shown in Table A.03. Since none of the VIFs

are greater than 5, and in fact small (close to 1), we have evidence that multicollinearity between our explanatory variables is not a problem [16].

Variable	ACDC1	ACDC2	ACDC3	Ultra Strong	Strict Verd	Reinitz	NFL	PFL	RE
VIF	1.105	1.050	1.074	1.182	1.049	1.086	1.615	1.566	1.417

Table A.03: Each explanatory variable VIF for the MLR model.

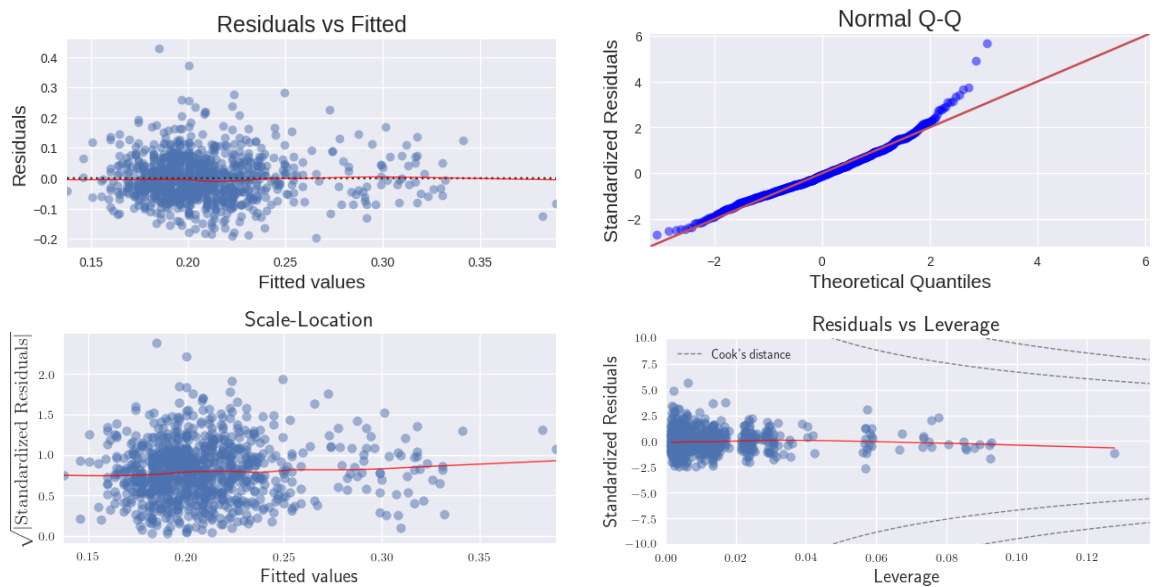


Figure A.01: MLR diagnostic plots.

We see no clear pattern in the Residuals vs Fitted plot, showing it is reasonable to assume linearity of relationships [16]. The Scale-Location plot shows weak to moderate evidence against equal variance, as indicated by having higher variance for the middling fitted values [16]. In general, our normal QQ-plot is showing a deviation from the line of normality, though only a slight right-skew. Hence, we see no indication of a violation of the normality assumption. Lastly, our average leverage is approximately 0.01, meaning all points with leverage greater than 0.02 have high leverage. However, since no points have a Cook's distance greater than 0.5 then we can conclude no points are overly influential to our model [16].

APPENDIX B

NETWORK MEASURE RESULTS BEFORE NORMALIZATION

