COMPARING NETWORK MODELS OF GAP GENE INTERACTION DURING *DROSOPHILA MELANOGASTER* DEVELOPMENT.

by

Elizabeth Anne Andreas

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

of

Master of Science

in

Mathematics

MONTANA STATE UNIVERSITY
Bozeman, Montana

April 2021

TABLE OF CONTENTS

TABLE OF CONTENTS – CONTINUED

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

LIST OF ALGORITHMS

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. In this thesis we plan to investigate this robustness using a previously developed computational framework called Dynamic Signatures Generated by Regulatory Networks (DSGRN). The principal result of this research has been in extending DSGRN to study how tissue-scale behavior arises from network behavior in individual cells, such as gap gene expression along the anterior-posterior (A-P) axis of the *Drosophila* embryo. Essentially, we extend DSGRN to study cellular systems where each cell contains the same network structure but operates under a parameter regime that changes continuously from cell to cell. We then use this extension to study the robustness of two different models of the gap gene network by looking at the number of paths in each network that can produce the observed gap gene expression. While we found that both networks are capable or replicating the data, we hypothesize that one network is a better fit than the other. This is significant in two ways; finding paths shows us that the spatial data can be replicated using a single network with different parameters along the A-P axis, and that we may be able to use this extension of DSGRN to rank network models.

1
INTRODUCTION

Gap genes involved in driving segmentation of *Drosophila melanogaster* (the fruit fly) during early development have been extensively studied. Studies using the quantitative spatial expression patterns have shown that there can be significant variability between cells within an embryo as well as from embryo to embryo, yet the flies develop and when hatched, function reliably [6]. Thus, it is known that early development of *Drosophila* is stable or robust against perturbation.

In this thesis we plan to investigate the robustness of early development of *Drosophila* using a previously developed computational framework called Dynamic Signatures Generated by Regulatory Networks (DSGRN) [2]. DSGRN associates a finite combinatorial object called a *parameter graph* to a regulatory network, which captures the qualitative properties of the network dynamics across global parameter space. Each node of the parameter graph corresponds to a region of high dimensional parameter space, where the dynamics are described by a particular *state transition graph* (STG) representing behavior across all of phase space. Each STG is generated as an asynchronous update of a particular collection of monotone Boolean functions that is compatible with the network dynamics at the associated parameter node. The dynamics of the STG are summarized by a reachability graph of the strongly connected components, called a *Morse graph.* Leaves of the Morse graph correspond to attracting regions of the dynamics that can be associated to experimentally observable states. Every node of the parameter graph is associated to a (not necessarily distinct) Morse graph.

Up until now, DSGRN was a tool used for time-dependent intracellular modeling. However, we extend DSGRN to a time- and space-dependent tissue-level modeling, which is a significant advancement in the utility of DSGRN. Thus, the principal goal of this research is to extend DSGRN to study how tissue-scale behavior arises from network behavior in individual cells. We do this by formalizing a way to discretize spatial data, which we call the *phenotype pattern.* We then find a subset of the parameter graph, called the *phenotype graph*, consisting only of nodes that match the phenotype pattern. Then we develop algorithms for searching the phenotype graph for paths that match the phenotype pattern, as well as additional graph theoretic techniques for handling several emerging computations challenges. We believe that we can study robustness of a regulatory network by searching for paths in the phenotype graph because we hypothesize that the spatial expression patterns represents dynamics of a single network, but at different parameters along the spatial domain. Therefore, we can represent the spatial expression pattern as a path in the DSGRN parameter graph.

We then use the new tools to study two different network models of gap genes, which we call the fully connected (Fullconn) network and the strong edges (StrongEdges) network, by looking at the number of paths that can produce the observed dynamics in each network. We found that both networks are capable of replicating the the discretized data. We then hypothesis that we can *score* network robustness, showing that the StrongEdges network is a better candidate to match the data than the Fullconn network. Our results in this area are preliminary and their extension will be the focus of future work.

BACKGROUND

Graph theory

The central idea to the methodology that we use, DSGRN, is that important features of the dynamics of gene regulatory networks can be expressed using the language of graph theory. This *combinatorialization* of the dynamics allows effective computation of quantities that characterize dynamics, where the corresponding quantities for continuous dynamical systems parameterized by real-valued parameters would be inaccessible due to high dimensionality of the corresponding phase space and parameter space. Since in this thesis we will use repeatedly the language of graph theory to describe dynamics of gene regulatory networks, we now introduce some of the concepts we will need, starting with the common definitions for graphs.

Definition 2.1. *Let $G = (V, E)$ be an ordered pair where V is a finite set of vertices, called nodes, and*

$$E = \{\{x, y\} \mid x \in V, y \in V, x \neq y\}$$

is a finite set of undirected edges between nodes x and y. Then G is called a ***undirected graph*** *where the order $|V|$ is the number of nodes and the size $|E|$ is the number of edges.*

Definition 2.2. *Let $G = (V, E)$ be an ordered pair where V is a finite set of vertices, called nodes, and*

$$E = \{(x, y) \mid (x, y) \in V^2\}$$

is a finite set of directed edges from x (the source) to y (the target). Then G is called a ***directed graph*** *where the order $|V|$ is the number of nodes and the size $|E|$ is the number of edges. Notice in this definition we are allowing for* ***loops****, (i.e., where $x = y$).*

Notice the difference in notation between undirected edges, $\{x, y\}$, and directed edges, (x, y).

Definition 2.3. *A* ***path*** *in an undirected graph $G = (V, E)$ from $x \in V$ to $y \in V$ is a sequence of distinct edges $\{x, u_1\}, \{u_1, u_2\}, \ldots, \{u_{n-1}, u_n\}, \{u_n, y\}$ in E from x to y. A* ***path*** *in a directed graph $G = (V, E)$ from $x \in V$ to $y \in V$ is a sequence of distinct, directed edges $(x, u_1), (u_1, u_2), \ldots, (u_{n-1}, u_n), (u_n, y)$ in E from x to y. We denote a directed path from x to y by $x \to \cdots \to y$.*

Although nonstandard, we will assume for this manuscript that any path $x \to \cdots \to y$ satisfies the condition that no node in the path is ever revisited, unless otherwise noted.

Definition 2.4. *Given a directed graph $G = (V, E)$, if every pair of nodes $x, y \in V$ has a path from x to y and from y to x following directed edges, then G is said to be* ***strongly connected****. A strongly connected subgraph H of G is said to be* ***maximal*** *if there is no subgraph H' in G containing H that is also strongly connected. A maximal strongly connected subgraph is called a* ***strongly connected component****. This includes singleton vertices that are not strongly connected to any other node. A* ***strongly connected path component*** *excludes the singleton cases.*

Strongly connected components are important for the graph theoretic techniques that we develop. Strongly connected path components are important for understanding the Morse graph of DSGRN mentioned in the Introduction, and discussed in detail in the following section.

Definition 2.5. *A graph is **acyclic** if it contains no directed paths from a node to itself, including loops. The **transitive reduction** of an acyclic graph $G = (V, E)$ is the unique subgraph $H = (V, E')$ with the smallest subset of edges $E' \subseteq E$ that satisfies the condition that a path $x \to \cdots \to y$ exists in G if and only if a (possibly distinct) path $x \to \cdots \to y$ exists in H.*

Definition 2.6. *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 $u \neq v$, 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.*

In Definition 2.19 we will discuss the analogous definition for strongly connected path components.

One of the key outputs of DSGRN is an object called a partially ordered set, which can be represented as an acyclic graph.

Definition 2.7. *Given a set P with the binary operation $\leqslant$, $(P, \leqslant)$ is called a **partially ordered set**, or **poset**, if*

- *$p \leqslant p$ for all $p \in P$ (reflexivity)*
- *if $p_0 \leqslant p_1$ and $p_1 \leqslant p_0$, then $p_0 = p_1$ (antisymmetry)*
- *if $p_0 \leqslant p_1$ and $p_1 \leqslant p_2$ then $p_0 \leqslant p_2$ for all $p_0, p_1, p_2 \in P$ (transitivity).*

*An object $(P, <)$ is called a **strict partially ordered set** if*

- *$p \nless p$ for all $p \in P$ (irreflexivity)*
- *$p_0 < p_1$ implies $p_1 \nless p_0$ for all $p_0 \neq p_1 \in P$ (asymmetry)*
- *if $p_0 < p_1$ and $p_1 < p_2$ then $p_0 < p_2$ for all $p_0 \neq p_1 \neq p_2 \in P$ (transitivity).*

*The **Hasse diagram** of a poset is the transitive reduction H of the acyclic graph $G = (P, E)$ with directed edge $(u, v) \in E$ for $u \neq v$ if and only if $u < v$.*

The input to DSGRN is also a special kind of graph. The main conceptual model of gene cell regulation in systems biology is that of a regulatory network. 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.

Definition 2.8. *A **Regulatory Network** is a directed graph, denoted **RN**=(V,E), where the vertices are network nodes and the edges are interactions, annotated by* $\to$ *or* $\dashv$*. An annotated edge* $(x, y) \in E$*,* $x \to y$ *is denoting that* x *is an **activator** of* y *and* $x \dashv y$ *that* x *is an **inhibitor** of* y*. The edge* (x, y) *is annotated by either* $\to$ *or* $\dashv$*, but not both.*

Definition 2.9. *Given a regulatory network* $\boldsymbol{RN} = (V, E)$*, a **source** of a node* v_j *is some node* v_i *such that* v_i *activates or inhibits* v_j*. A **target** of* v_j *is some node* v_n *such that* v_j *activates or inhibits* v_n*. The set of sources and targets of a node* v_j *are given by*

$$S(v_j) := \{v_i \mid (v_i, v_j) \in E\} \quad \text{and} \quad T(v_j) := \{v_n \mid (v_j, v_n) \in E\}.$$

The simplest conceptual model of the dynamics of a regulatory network is *Boolean network.*

Definition 2.10. *Let* $B := \{0, 1\}$ *be a set of binary values. A Boolean function is a map* $g : B^k \to B$*. A Boolean map* $g(x_1, \ldots, x_k)$ *is non-decreasing in variable* x_j *if*

$$g(x_1 \ldots, x_{j-1}, 0, x_{j+1} \ldots, x_k) \leqslant g(x_1 \ldots, x_{j-1}, 1, x_{j+1}, \ldots, x_k)$$

for any values $x_1, \ldots, x_k$*. The map* g *is non-increasing in variable* x_j *if*

$$g(x_1 \ldots, x_{j-1}, 0, x_{j+1} \ldots, x_k) \geqslant g(x_1 \ldots, x_{j-1}, 1, x_{j+1}, \ldots, x_k).$$

*A Boolean network model associated to **RN**=(V,E) is a collection* $f = (f_1, \ldots, f_N)$ *of* $N := |V|$ *Boolean maps* $f_i : B^{|S_i|} \to B$ *where* f_i *is non-decreasing in* x_j *if, and only if* $x_j \to x_i$*, and non-increasing in* x_j *if, and only if* $x_j \dashv x_i$*.*

DSGRN

In this section, we discuss a previously developed tool called Dynamic Signatures Generated by Regulatory Networks (DSGRN) [2] that associates a finite combinatorial object called a parameter graph to a regulatory network, which captures the qualitative properties of the network dynamics across global parameter space. DSGRN is a useful tool in studying the global dynamics of a network across all its parameters. In this thesis, we will need to fully understand this tool, thus we will go over all necessary definitions in order to describe it.

Switching systems

We will proceed by associating to a regulatory network **RN** a set of differential equations that are compatible with biological assumptions used to construct such a network.

Given a regulatory network $\mathbf{RN} = (V, E)$, the rate at which the species concentration of v_i in a cell decays is called the **decay rate** and will be denoted by $\gamma_i > 0$. A decay constant is associated to every node v_i in the network. Additionally, we associate a set of non-negative parameters l_{ji}, u_{ji} and θ_{ji} in $\mathbb{R}^+$ to each edge $(v_i, v_j) \in E$. Here l_{ji} and u_{ji} and are called the lower (low) and upper (high) expression levels of the regulated node v_j induced by v_i with $0 \leqslant l_{ji} < u_{ji}$. A threshold $\theta_{ji} > 0$ for node v_i is where the expression levels of the regulator node v_i change from low to high or high to low with respect to regulatory activity at node v_j. We assume that the values of θ_{ki} for any node i are distinct. In other words, if $\theta_{ji} = \theta_{ki}$ then $j = k$. See Figure 2.1 (a) for an example **RN**.

Definition 2.11. *We call the collection of all parameters $\bar{\mathcal{P}} = (l, u, \theta, \gamma)$ the* ***parameter space*** *for* ***RN****, where $\gamma = (\gamma_1, \ldots, \gamma_N)$ for a network with N nodes, and*

$$\begin{aligned} \theta &= \{\theta_{ji} \textit{ for all } (v_i, v_j) \in E\} \\ l &= \{l_{ji} \textit{ for all } (v_i, v_j) \in E\} \\ u &= \{u_{ji} \textit{ for all } (v_i, v_j) \in E\}. \end{aligned}$$

The dimension of parameter space is then $|V| + 3|E|$.

Given a regulatory network $\mathbf{RN} = (V, E)$ and a parameter (l, u, θ, γ), the rate of activation and inhibition of v_j is given by a vector of step functions $\sigma_j : \mathbb{R}^{|V|} \to \mathbb{R}^{|S(v_j)|}$, where

$$\pi_{v_i}(\sigma_j) =: \sigma_{ji}(v) = \begin{cases} l_{ji} & \text{if } v_i \to v_j \text{ and } v_i < \theta_{ji} \text{ or } v_i \dashv v_j \text{ and } v_i > \theta_{ji} \\ u_{ji} & \text{if } v_i \to v_j \text{ and } v_i > \theta_{ji} \text{ or } v_i \dashv v_j \text{ and } v_i < \theta_{ji} \end{cases} \tag{2.1}$$

for each $v_i \in S(v_j)$.

Definition 2.12. *Given a regulatory network* $\boldsymbol{RN} = (V, E)$ *of order N, a* ***switching system*** *is a system of N ordinary differential equations of the form*

$$\dot{v}_j = -\gamma_j v_j + \Lambda_j(v) \tag{2.2}$$

where γ_j is the decay constant and $\Lambda_j(v)$ is a piecewise constant function

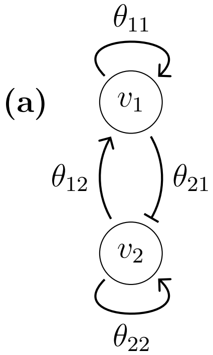
$$\Lambda_j(v) := M_j \circ \sigma_j. \tag{2.3}$$

Here M_j is a multilinear function $M_j : \mathbb{R}^{|S(v_j)|} \to \mathbb{R}$ such that

$$M_j(\sigma_j) = \prod \sum \sigma_{ji} \tag{2.4}$$

and each σ_{ji} appears exactly once in the expression. The map M_j is called the ***algebraic expression*** *of the node v_j.*

See Figure 2.1 (b) for an example. Most often, we use the following rule of thumb when constructing the algebraic expression at a node v_j: we choose to add the functions σ_{ji} for all activators v_i of node v_j and multiply the resulting sum by σ_{jk} over all repressors v_k of v_j, as is discussed in section 2.



$$\textbf{(b)}\ \dot{v}_1 = -\gamma_1 v_1 + \left(\begin{cases} l_{12} & \text{if } v_2 < \theta_{12} \\ u_{12} & \text{if } v_2 > \theta_{12} \end{cases} \right) + \left(\begin{cases} l_{11} & \text{if } v_1 < \theta_{11} \\ u_{11} & \text{if } v_1 > \theta_{11} \end{cases} \right)$$

$$\dot{v}_2 = -\gamma_2 v_2 + \left(\begin{cases} u_{21} & \text{if } v_1 < \theta_{21} \\ l_{21} & \text{if } v_1 > \theta_{21} \end{cases} \right) \cdot \left(\begin{cases} l_{22} & \text{if } v_2 < \theta_{22} \\ u_{22} & \text{if } v_2 > \theta_{22} \end{cases} \right)$$

Figure 2.1: (a) An example a regulatory network **RN** with two nodes v_1 and v_2, and four edges. To each edge we associate expression levels $l_{11} < u_{11}$, $l_{21} < u_{21}$, $l_{22} < u_{22}$ and $l_{12} < u_{12}$, as well as thresholds θ_{11}, θ_{21}, θ_{22} and θ_{12}. (b) The set of ordinary differential equations for **RN**, with an algebraic expression that adds σ_{12} and σ_{11} and multiplies σ_{21} and σ_{22}.

Phase space

Next, we show that the choice of piecewise constant functions σ_{ij} allows us to define a finite decomposition of the phase space into domains, in which the dynamics of the switching system is particularly simple. This will allow us later to assemble the information from these domains into a state transition graph, which will describe global dynamics of a switching system.

Definition 2.13. *Given a regulatory network* $\boldsymbol{RN} = (V, E)$ *with* $|V| = N$ *and a parameter space* $\bar{\mathcal{P}}$ *of* $\boldsymbol{RN}$*, consider the finite collection of thresholds* $\theta \in \bar{\mathcal{P}}$*. Each* $v_i \in V$ *has a finite number of thresholds* $\theta_{j_1 i}, \theta_{j_2 i}, ..., \theta_{j_n i} \in \theta$*. These thresholds divide the interval* $[0, \infty)$ *into* $n+1$ *intervals* I*. Choosing* $v_k \neq v_i$*,* v_k *also has a finite number of thresholds* $\theta_{\ell_1 k}, \theta_{\ell_2 k}, ..., \theta_{\ell_m k} \in \theta$ *that together with the thresholds of* v_i *decomposes* $[0, \infty)^2$ *into* $(n+1)(m+1)$ *rectangles. Continuing through all nodes in* $\boldsymbol{RN}$*, the thresholds in* θ *divide* $[0, \infty)^N$ *into a finite number of* N*-dimensional rectangles called* ***domains****. Let* $\mathcal{K}$ *denote the collection of all domains created by the collection of thresholds* θ*.*

See Figure 2.2 (a)-(c) for an example. Note that each domain k has the form $k := \Pi_{i=1}^{N}[\theta_{a_i i}, \theta_{b_i i}]$, where the i-th projection $\pi_i(k) := [\theta_{a_i i}, \theta_{b_i i}]$ and $\theta_{a_i i} < \theta_{b_i i}$ are *consecutive thresholds*, i.e. there is no threshold θ_{ji} with $\theta_{a_i i} < \theta_{ji} < \theta_{b_i i}$. Then

$$\Pi_{j=1}^{i-1}[\theta_{a_j j}, \theta_{b_j j}] \times \{\theta_{a_i i}\} \times \Pi_{j=i+1}^{N}[\theta_{a_j j}, \theta_{b_j j}]$$

is the i-th **left face** of k and

$$\Pi_{j=1}^{i-1}[\theta_{a_j j}, \theta_{b_j j}] \times \{\theta_{b_i i}\} \times \Pi_{j=i+1}^{N}[\theta_{a_j j}, \theta_{b_j j}]$$

is the i-th **right face** of k. Notice that a face is either a left face or a right face. A **wall** is pair (τ, k) where τ is a face of the domain k.

The following definition defines a generic subset of all parameters that avoids several degeneracies.

Definition 2.14. *A parameter $p \in \bar{\mathcal{P}}$ is **regular** if, in addition to distinct $\{\theta_{j,i}\}$ for every v_i, we have*

1. $0 < l_{ji} < u_{ji}$, $0 < \gamma_i$, *and* $0 < \theta_{ji}$,
2. *for each domain $k \in \mathcal{K}$, $\Lambda_i(k) \neq \gamma_i \theta_{ji}$.*

We denote the set of regular parameters by $\mathcal{P}$.

Definition 2.15. *Given a regulatory network $\boldsymbol{RN} = (V, E)$ and collection of algebraic expressions of $\Lambda_i, i = 1, \ldots, N$, let $\mathcal{P}$ be the associated set of regular parameters. Take a parameter $p \in \mathcal{P}$.Then p uniquely determines a switching system and a set of domains $\mathcal{K}(p)$. For each domain $k \in \mathcal{K}(p)$, the function $\Lambda_j(v)$ has a constant value for each $v \in k$, $j \in 1, \ldots, N$. Let $\Lambda(k) := (\Lambda_1(k), \ldots, \Lambda_N(k))$ denote the vector of these values. Note that the flow in each domain k converges to an point, that can be found by*

$$\dot{v}|_k = -\Gamma v + \Lambda(k) = 0. \tag{2.5}$$

*Here Γ is a diagonal matrix, with decay rates γ_j as its diagonal entries. Then a **target point** for k is*

$$TP(k) = \Gamma^{-1}\Lambda(k). \tag{2.6}$$

*When $TP(k) \in k$, we call k an **attracting domain**.*

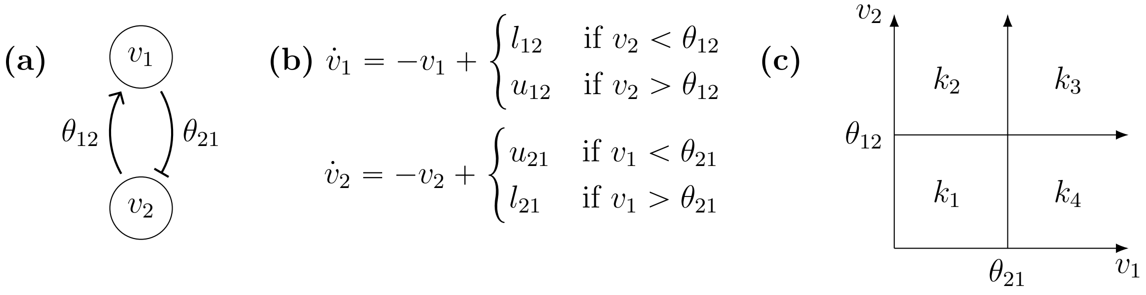
See Figure 2.2 for an example.

Definition 2.16. *Given a regulatory network $\boldsymbol{RN}$ and a choice Λ, let $\mathcal{P}$ be the associated set of regular parameters. We say that $\boldsymbol{RN} = (V, E, \Lambda, \mathcal{P})$ is a **parameterized regulatory network.***

The next definition assigns to each wall a direction, expressed as a wall label, that indicates in which direction is the wall crossed by the solutions of the switching system. Since the vector field of the switching system is not defined on faces of the domains, one can only consider a closure of the solutions defined on both open domain that border a particular wall. We avoid the intricacies of trying to define these directions using the local solutions in each domain by defining the wall labels directly using a relative position of the target point of a domain and the domain itself.

Definition 2.17. *Let $\boldsymbol{RN} = (V, E, \Lambda, \mathcal{P})$ be a parameterized regulatory network and let $p \in \mathcal{P}$. Let $\mathcal{K}$ be the set of domains created by the collection of thresholds θ in p. Given a wall (τ, k) for $k \in \mathcal{K}$, we assign a **wall label** by*

$$sgn(\tau, k) = \begin{cases} 1 & \text{if } \tau \text{ is a left face of } k \\ -1 & \text{if } \tau \text{ is a right face of } k. \end{cases}$$



(b)
$$\dot{v}_1 = -v_1 + \begin{cases} l_{12} & \text{if } v_2 < \theta_{12} \\ u_{12} & \text{if } v_2 > \theta_{12} \end{cases}$$
$$\dot{v}_2 = -v_2 + \begin{cases} u_{21} & \text{if } v_1 < \theta_{21} \\ l_{21} & \text{if } v_1 > \theta_{21} \end{cases}$$

(d) $TP(k_1) = (l_{12}, u_{21}), \quad TP(k_2) = (u_{12}, u_{21}), \quad TP(k_3) = (u_{12}, l_{21}), \quad TP(k_4) = (l_{12}, l_{21})$

Figure 2.2: (a) An example of a regulatory network **RN**. (b) The set of ordinary differential equations, with decay rates γ_1, γ_2 chosen to equal 1. (c) The thresholds θ_{12} and θ_{21} break up phase space into four domains k_1 through k_4. (d) The target points for each domain k_i. The position of target points within the domains $k_1, \ldots, k_4$ determines the state transition graph; the collection of DSGRN parameters, organized in the parameter graph, are formed by the inequalities that delineate all possible configurations of target points within the four domains.

The collection of wall labels for p is denoted $\mathcal{W}(p)$. Given a wall (τ, k) such that τ is created by the threshold θ_{ji}, we can define a logic function $\ell : \mathcal{W}(p) \to \{-1, 0, 1\}$ by

$$\ell(\tau, k) := sgn(\tau, k) sgn(\dot{v}_i|_k) = sgn(\tau, k) sgn(-\gamma_i \theta_{ji} + \Lambda_i(k)).$$

*Notice that a wall is an **incoming wall** if $\ell(\tau, k) = 1$, an **outgoing wall** if $\ell(\tau, k) = -1$, and a tangential wall if $\ell(\tau, k) = 0$. Additionally, k can only be an attracting domain if every wall of k is an incoming wall.*

Finally, we assemble the information about the domains and wall labels into an abstract object, called state transition graph that captures global dynamics of a switching system at a fixed parameter p.

Definition 2.18. *Let **RN** $= (V, E, \Lambda, \mathcal{P})$ be a parameterized regulatory network, let $p \in \mathcal{P}$ and consider the wall labeling ℓ induced by p. We define a **state transition graph** STG=($\mathcal{V}, \mathcal{E}$). Each domain k will be represented by a vertex $u(k)$. Vertices $u(k_1)$ and $u(k_2)$ will be connected by a directed edge $u(k_1) \to u(k_2)$ if*

1. $k_1 \cap k_2 = \tau$,

2. $\ell(\tau, k_1) = -1$ *and* $\ell(\tau, k_2) = 1$.

*In addition, vertex $u(k)$ has a self-edge,if, and only if k is an attracting domain. Alternatively, we can view the STG assigned to p as a multivalued map $\mathcal{F}(p) : V \rightrightarrows V$. Finally, we denote the set of all state transition graphs for RN by **STG($\mathcal{P}$)**.*

See Figure 2.3 for an example of how the piece-wise vector field of a switching system is transformed into an STG.

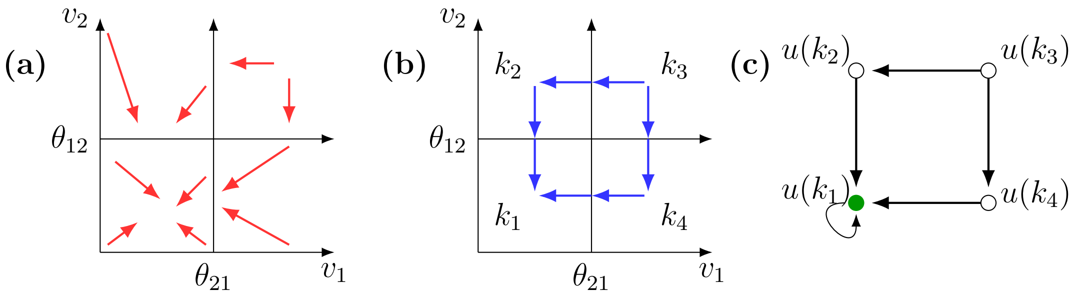


Figure 2.3: (a) A set of trajectories for some parameter $p \in \mathcal{P}$ in the domains for **RN** from the example in Figure 2.2 and (b) is a depiction of the wall labeling where an arrow pointing away from a wall is incoming and an arrow pointing to the wall is outgoing. (c) A depiction of the state transition graph associated to the domains and wall labeling from (b).

Dynamics

State transition graphs are finite objects that capture global dynamics of a switching system of differential equations and hence provide a computationally tractable means of interrogation of such systems. Nevertheless, the state transition graph size grows rapidly with the number of nodes in the network. Therefore, we compute a summary of recurrent behavior of the state transition graph by considering structure of its strongly connected components.

Definition 2.19. *Let **RN** $= (V, E, \Lambda, \mathcal{P})$ be a parameterized regulatory network, let $p \in \mathcal{P}$ and consider the state transition graph $\mathcal{F}(p)$. The **Morse decomposition** of $\mathcal{F}(p)$, denoted as MD($\mathcal{F}(p)$), is the set of all strongly connected path components of $\mathcal{F}(p)$. Consider any two strongly connected path components $s_1, s_2 \in$MD($\mathcal{F}(p)$), if there is no path in STG from s_1 to s_2 then we say $s_1 \leqslant s_2$, defining a partial order (PO, $\leqslant$) on MD($\mathcal{F}(p)$). The **Morse graph** of $\mathcal{F}(p)$, denoted MG($\mathcal{F}(p)$) is the Hasse diagram of (PO, $\leqslant$), and the vertices of MG($\mathcal{F}(p)$) are called **Morse Nodes**.*

In order for the Morse graph to provide interpretable information, we label Morse nodes in a way that suggest the dynamics that can be associated to it.

Definition 2.20. *Let **RN** $= (V, E, \Lambda, \mathcal{P})$ be a parameterized regulatory network, and consider the Morse graph MG($\mathcal{F}(p)$). Each Morse node component is annotated FP(w), FC, or XC($v_{i_1}, ..., v_{i_n}$) according to the underlying strongly connected path component of the state transition graph dynamics in the following way.*

1. *Suppose a strongly connected path component of a state transition graph is a single vertex $u(k)$ such that k is the domain associated to $u(k)$. Choose some v_i and let*

*$n = |T(v_i)|$. The thresholds of v_i can be ordered $0 < \theta_{a_1 i} < \theta_{a_2 i} < \cdots < \theta_{a_n i}$, where each a_m is some $j \in T(v_i)$. The projection of the domain k onto the i-th axis is then an interval between a pair of consecutive thresholds $(\theta_{a_{m_i} i}, \theta_{a_{m_i+1} i})$. There is one such interval for each of the N vertices in **RN**. So for the domain k, we can define a vector $w = (m_1, m_2, \ldots, m_N)$, where each $0 \leqslant m_j \leqslant |T(v_j)|$ is the index of the left endpoint of the j-th interval. Notice that this is the number of thresholds to the left of the domain k in the j-th direction. The notation FP(w) is used to label a Morse node where the corresponding strongly connected path component (and therefore the attracting region) consists of a single domain k.*

2. *Suppose a strongly connected path component of a state transition graph has a path $P := u(k_1) \to u(k_2) \to \ldots \to u(k_s) \to u(k_1)$. Each transition is associated with crossing a threshold θ_{ij} for some i, j. If the collection of thresholds within a path P contains θ_{ij} for every $i \in \{1, \ldots, N\}$, then we label the Morse node by FC for "full cycle". Notice that here the nodes of the path need not be distinct.*

3. *If a strongly connected path component M of STG contains a path P, as described in (2), where the collection of thresholds θ_{ij} contains $i \in S \subsetneq \{1, \ldots, N\}$, then we label the corresponding Morse node by XC(S), where $S \subsetneq V$ is a maximal such set across all paths in M.*

*The collection of annotated Morse graphs is denoted **AnnMG**.*

Definition 2.21. *A **monostable Morse Graph** is a Morse graph containing a single node with no targets. A **strict monostable Morse Graph** is a Morse graph containing only a single Morse node. A **monostable fixed point** is the unique attracting Morse node in a monostable Morse graph and has FP annotation. A **strict monostable fixed point** is the unique Morse node in a strict monostable Morse graph and has FP annotation.*

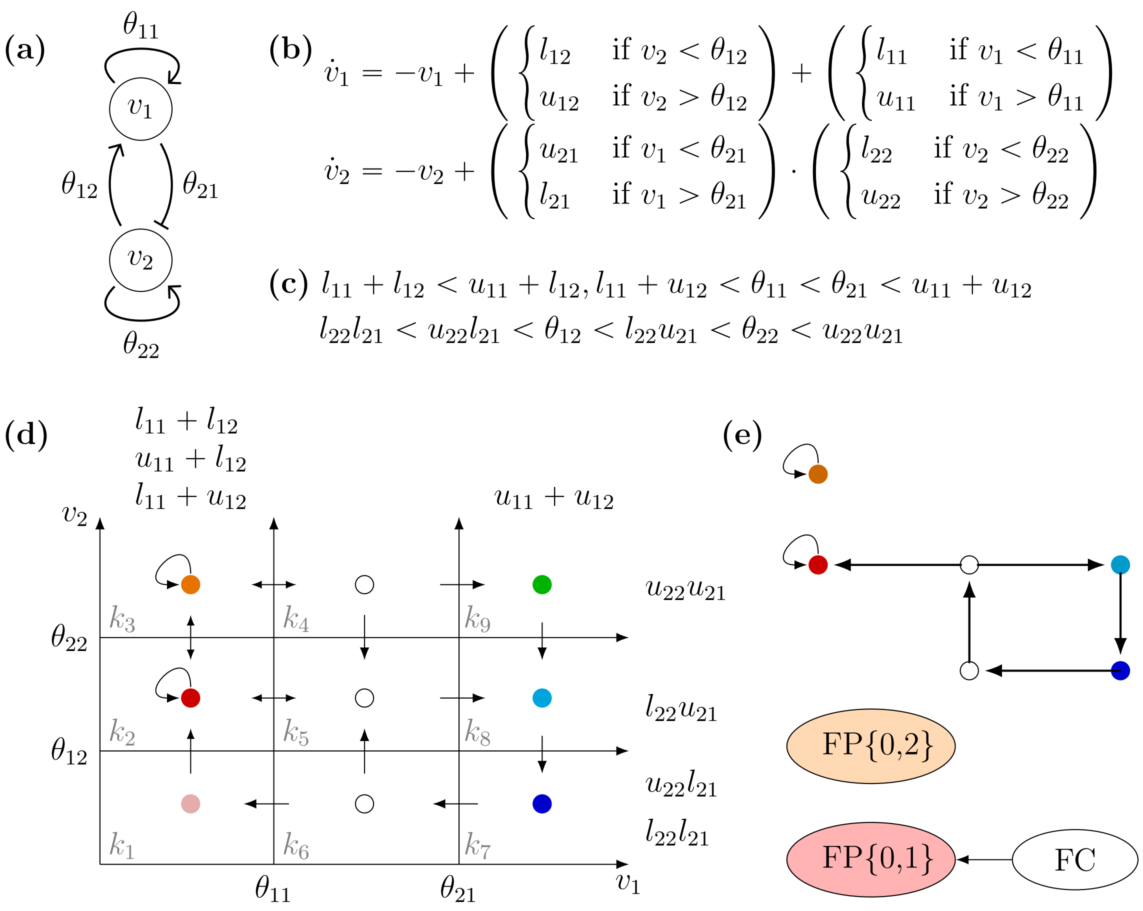
Consider a parameterized regulatory network $\mathbf{RN} = (V, E, \Lambda, \mathcal{P})$ with two nodes, v_1 and v_2, with activating self loops, an activating edge $v_2 \to v_1$, and a repressing edge $v_1 \dashv v_2$. Additionally, we assume that the functions σ_{12} and σ_{11} are added together and functions σ_{21} and σ_{22} are multiplied together when affecting the node v_1, respectively v_2; see Figure 2.4 (a).

Consider a parameter $p \in \mathcal{P}$ such that $\gamma_1 = \gamma_2 = 1$, and

$$\begin{aligned} p =\{&l_{11} + l_{12} < u_{11} + l_{12}, l_{11} + u_{12} < \theta_{11} < \theta_{21} < u_{11} + u_{12}, \\ &l_{22}l_{21} < u_{22}l_{21} < \theta_{12} < l_{22}u_{21} < \theta_{22} < u_{22}u_{21}.\} \end{aligned}$$

Then, as we can see in Figure 2.4, p determines the state transition graph, as superimposed on phase space in part (d). For example, consider the domain k_1, which is the bottom left domain in Figure 2.4(d). Any point in domain k_1 is below all of the thresholds, thus the ordinary differential equations in this domain are

$$\dot{v}_1 = -\gamma_1 v_1 + (l_{12} + u_{11})$$



(b)
$$\dot{v}_1 = -v_1 + \left(\begin{cases} l_{12} & \text{if } v_2 < \theta_{12} \\ u_{12} & \text{if } v_2 > \theta_{12} \end{cases}\right) + \left(\begin{cases} l_{11} & \text{if } v_1 < \theta_{11} \\ u_{11} & \text{if } v_1 > \theta_{11} \end{cases}\right)$$
$$\dot{v}_2 = -v_2 + \left(\begin{cases} u_{21} & \text{if } v_1 < \theta_{21} \\ l_{21} & \text{if } v_1 > \theta_{21} \end{cases}\right) \cdot \left(\begin{cases} l_{22} & \text{if } v_2 < \theta_{22} \\ u_{22} & \text{if } v_2 > \theta_{22} \end{cases}\right)$$

(c) $l_{11} + l_{12} < u_{11} + l_{12}, l_{11} + u_{12} < \theta_{11} < \theta_{21} < u_{11} + u_{12}$
$l_{22}l_{21} < u_{22}l_{21} < \theta_{12} < l_{22}u_{21} < \theta_{22} < u_{22}u_{21}$

$TP(k_3) = (u_{12} + l_{11}, u_{21}u_{22}),\ TP(k_4) = (u_{12} + u_{11}, u_{21}u_{22}),\ TP(k_9) = (u_{12} + u_{11}, l_{21}u_{22}),$
$TP(k_2) = (u_{12} + l_{11}, u_{21}l_{22}),\ TP(k_5) = (u_{12} + u_{11}, u_{21}l_{22}),\ TP(k_8) = (u_{12} + u_{11}, l_{21}l_{22}),$
$TP(k_1) = (l_{12} + l_{11}, u_{21}l_{22}),\ TP(k_6) = (l_{12} + u_{11}, u_{21}l_{22}),\ TP(k_7) = (l_{12} + u_{11}, l_{21}l_{22}).$

Figure 2.4: (a) The **RN** and the (b) associated ordinary differential equations from Figure 2.1, with decay rates $\gamma_1 = \gamma_2 = 1$. (c) A choice of DSGRN parameter. (d) Phase space decomposition into the nine domains by thresholds $\theta_{11}, \theta_{12}, \theta_{22}$ and θ_{21}. Each domain is represented by a circular vertex inside the domain. Arrows are the depiction of wall labelings. The choice of DSGRN parameter has $u_{11} + u_{22} > \theta_{21}$, which is depicted above domain k_9. The vertices and wall labelings form the state transition graph. Below the phase space diagram is the list of target points, where colors of the TP(k_i) match the color of the vertex of the domain where that target point falls (for example, TP(k_1), TP(k_2) and TP(k_6) all have target points in the domain k_2). (e) The Morse graph associated to the state transition diagram (below) and the strongly connected components, or Morse nodes, associated to each node of the Morse graph (above).

$$\dot{v}_2 = -\gamma_2 v_2 + (l_{22} u_{21})$$

and the target point for this domain is

$$TP(k_1) = (l_{12} + l_{11}, l_{22} u_{21}),$$

assuming $\gamma_1 = \gamma_2 = 1$ for simplicity. Notice the parameter p tells us the value $l_{12} + l_{11}$ is below θ_{11}, while $l_{22} u_{21}$ is between θ_{12} and θ_{22}. Therefore the target point $TP(k_1)$ is in domain k_2, which is the domain directly above domain k_1 in Figure 2.4. The arrow from domain k_1 to domain k_2 is depicting the corresponding wall label. Notice that every arrow from the domains sharing a wall with k_2 are going into k_2, showing that k_2 is an attracting domain. In Figure 2.4, this is depicted by a red circular vertex with a self loop. Recall from Definition 2.18 that we can denote the Morse nodes associated to these circular vertices as $u(k_i)$ for $i \in \{1, ..., 9\}$. The combination of all arrows between the domains and the vertices $u(k_i)$ is the the state transition graph, and for this example can be seen in part (d) of Figure 2.4. In this state transition graph there is a path $P := u(k_6) \to u(k_5) \to u(k_8) \to u(k_7) \to u(k_6)$ and this path crosses the thresholds θ_{12} and θ_{21}. Since at least one threshold for each node v_1 and v_2 has been crossed, then P is a full cycle by Definition 2.20. Additionally, there is a path from $u(k_5) \to u(k_2)$ that connects the full cycle to the attracting domain k_2, indicating reachability between the corresponding Morse nodes. Lastly, k_3 is also an attracting domain, so we have 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. The strongly connected path components in the state transition diagram, as well as the Morse graph, can be seen in Figure 2.4(e).

DSGRN parameter graph

Our final construction is the decomposition of the regular parameter space $\mathcal{P}$ into a finite set of semi-algebraic sets, such that for all parameters p in one of these domains, the STG is the same. Identifying each such domain of parameters with a node in a parameter graph (PG), we complete the construction of PG by connecting edges to nodes that correspond to domains that share a codimension-1 boundary in $\mathcal{P}$.

Definition 2.22. *Let* $\boldsymbol{RN} = (V, E, \Lambda, \mathcal{P})$ *be a parameterized regulatory network and let* $p \in \mathcal{P}$. *Notice that by construction, the functions* Λ_i *each have a finite number of values dependent on* p *that fulfill some set of inequalities wherein either* $\Lambda_i(k) < \gamma_i \theta_{j_n,i}$ *or* $\Lambda_i(k) > \gamma_i \theta_{j_n,i}$ *for each* $i \in 1, ..., |V|$ *and for each* $v_{j_n} \in T(v_i)$ *and domain* k. *Notice also that* p *defines an explicit threshold order* $\theta_{j_1,i} < ... < \theta_{j_{|T(v_i)|},i}$. *This collection of abstract inequalities defines a semi-algebraic region* $\mathcal{R}$ *of parameter space where every* $p \in \mathcal{R}$ *induces the same set of inequalities. We call such a collection of inequalities a* ***DSGRN parameter****, denoted* $\mathcal{D}$.

Definition 2.23. *The set of all DSGRN parameters for some parameterized regulatory network* $\boldsymbol{RN} = (V, E, \Lambda, \mathcal{P})$ *will be represented as the set of nodes of a connected* ***parameter graph (PG)****, with an undirected edge between nodes if there is a single inequality change between one of the orderings in the collection corresponding to a* ***PG*** *node. Each* $d \in \mathcal{D}$ *defines the location of the target points for each of the domains* $k \in \mathcal{K}$. *Thus, each* ***PG****(d) is* *uniquely associated to a state transition graph and Morse graph. We will therefore assign to d the corresponding Morse graph annotations as described in Definition 2.20.*

*The parameter graph is a product of graphs, one associated to each v_i. The **factor graph** of v_i is an undirected graph that is the projection of **PG** onto the i-th collection of parameter inequalities. That is, the factor graph of v_i is an undirected, connected graph where each node is associated to a choice of inequalities consistent with the algebraic expression M*

$$\{\Lambda_i(k) < \gamma_i\theta_{j_n,i} \text{ or } \Lambda_i(k) > \gamma_i\theta_{j_n,i} \text{ for each } i \in 1,...,|V|, v_{j_n} \in T(v_i), k \in \mathcal{K}\}$$
$$\text{and } \theta_{j_1,i} < ... < \theta_{j_{|T(v_i)|},i}$$

with edges between the inequality nodes occur whenever there is a single swap in order (see Figure 2.5).

The class of possible factor graphs is determined by the topology of network node v_i and the algebraic expression M_i. There is a unique factor graph for a node with an expression M_i, $|S(v_i)|$ in-edges, and $|T(v_i)|$ out-edges, regardless of the identity of the source or target nodes, and regardless of activation or repression on the edges [2].

An example of a parameter graph can be seen in Figure 2.5 for a two node regulatory network such that v_1 represses v_2 and v_2 activates v_1. The node v_1 has partial order of inequalities $l_{12} < u_{12}$, with three ways to order θ_{21}. Additionally, v_2 has partial order of inequalities $l_{21} < u_{21}$, with three ways to order θ_{12}. Then there are nine DSGRN parameters, which implies there are nine parameter graph nodes. Notice that there are no edges in the nodes diagonal from each other in this figure, since there are two inequality changes over a diagonal. We already saw this regulatory network and its system of equations in Figure 2.2. Additionally, in Figure 2.3 we saw a depiction of one of the state transition graphs, associated with FP{0,0}∈ AnnMG. In Figure 2.5 we see the state transition graph associated with FP{1,1}∈ AnnMG.

Computational considerations

DSGRN represents parameter space by the undirected parameter graph and represents the dynamics by the annotated Morse graph, which is a condensed version of the state transition graph. The output of DSGRN is called the DSGRN database, which is organized in a SQL database. The SQL database contains each node of the parameter graph and the annotated Morse graph that represents the dynamics. The DSGRN repository, at the time of writing, is stored on GitHub at github.com/marciogameiro/DSGRN. The version used for this thesis is 1.1.0 and was accessed on March 24th, 2020.

The parameter graph scales rapidly. For example, a regulatory network with two nodes and four edges has a parameter graph with a few thousand nodes [1]. However, as we will see later, a network with four nodes and eight edges has millions of parameter graph nodes. Even using a depth-first algorithm to find the existence of a single path in a parameter graph becomes a computational challenge at this size.

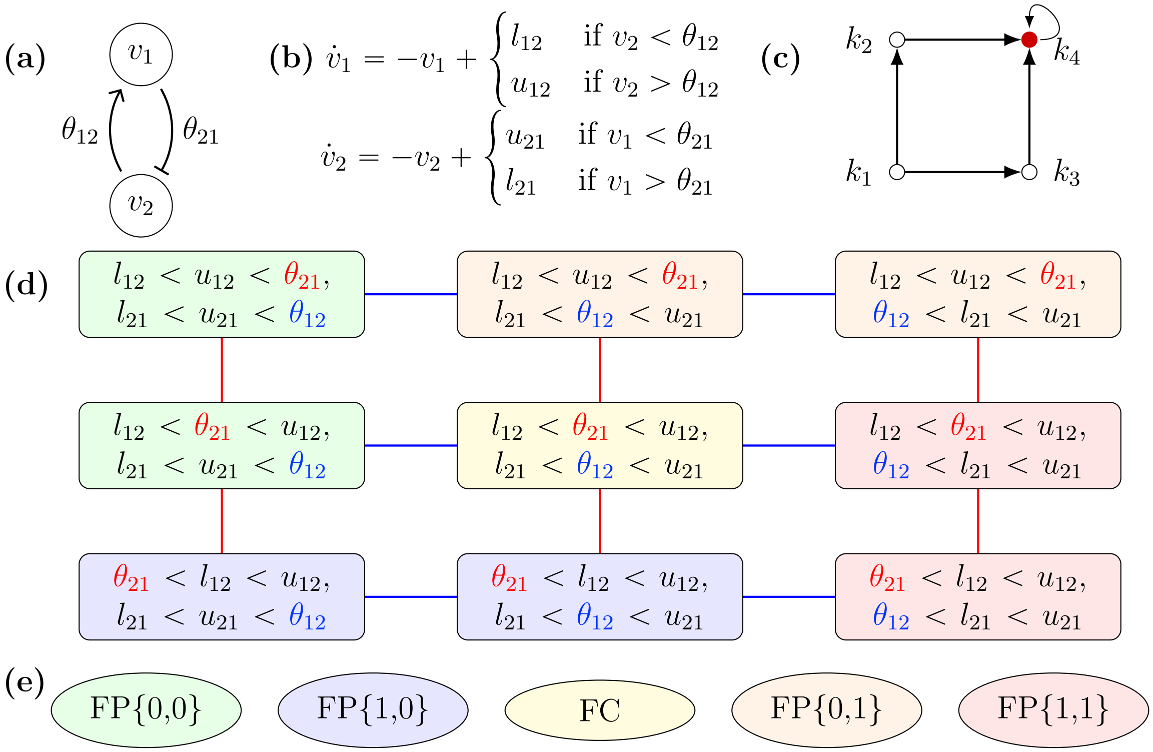

(a)
v_1
θ_{12}
θ_{21}
v_2
(b) $\dot{v}_1 = -v_1 + \begin{cases} l_{12} & \text{if } v_2 < \theta_{12} \\ u_{12} & \text{if } v_2 > \theta_{12} \end{cases}$
$\dot{v}_2 = -v_2 + \begin{cases} u_{21} & \text{if } v_1 < \theta_{21} \\ l_{21} & \text{if } v_1 > \theta_{21} \end{cases}$
(c)
k_2
k_4
k_1
k_3
(d)
$l_{12} < u_{12} < \theta_{21}$, $l_{21} < u_{21} < \theta_{12}$
$l_{12} < u_{12} < \theta_{21}$, $l_{21} < \theta_{12} < u_{21}$
$l_{12} < u_{12} < \theta_{21}$, $\theta_{12} < l_{21} < u_{21}$
$l_{12} < \theta_{21} < u_{12}$, $l_{21} < u_{21} < \theta_{12}$
$l_{12} < \theta_{21} < u_{12}$, $l_{21} < \theta_{12} < u_{21}$
$l_{12} < \theta_{21} < u_{12}$, $\theta_{12} < l_{21} < u_{21}$
$\theta_{21} < l_{12} < u_{12}$, $l_{21} < u_{21} < \theta_{12}$
$\theta_{21} < l_{12} < u_{12}$, $l_{21} < \theta_{12} < u_{21}$
$\theta_{21} < l_{12} < u_{12}$, $\theta_{12} < l_{21} < u_{21}$
(e)
FP{0,0}
FP{1,0}
FC
FP{0,1}
FP{1,1}

Figure 2.5: (a) An example of a RN and, (b) ordinary differential equations from Figure 2.2, with γ_1, γ_2 chosen to equal 1. (c) The state transition graph associated with FP{1,1}$\in$ AnnMG. (d) The parameter graph where blue edges are showing a change to θ_{12} and red edges are showing a change to θ_{21}. Each row of the grid is isomorphic to the factor graph of v_1, and each column is isomorphic to the factor graph of v_2. (e) The annotated Morse graphs associated to the parameter graph in d. Notice each colored region in d is depicting which annotated Morse graph that is associated to it. Figure adapted from [3].

DROSOPHILA MELANOGASTER

Early *Drosophila* melanogaster (fruit fly) development

Early development (i.e., development of the embryo) of *Drosophila* has three main stages, classified by the number of nuclear divisions called cycles: syncytial cleavage stage, syncytial blastoderm stage, and gastrulation. During the syncytial cleavage stage, cycles one through nine (see Figure 3.1), the *Drosophila* embryo undergoes nine nuclei divisions called cleavage 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 and nearly instantaneously, at the end of the syncytial blastoderm stage, cycle 14A. 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, the endoderm, mesoderm, and ectoderm. The ectoderm forms the exoskeleton and nervous system, the endoderm forms the intestinal organs and the mesoderm forms the rest of the organs. 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 [4].

Anterior-posterior (A-P) patterning in *Drosophila*

While the germ band is extended, different segments start appearing, segmenting from the head (**anterior region**) to the posterior of the embryo. After the head at the anterior, the next three segments form the thorax and the rest form the abdomen. Each segment is responsible for distinct portions of the adult *Drosophila*. Interestingly, the genes responsible for this segmentation were found experimentally by mutating the embryo and describing the resulting phenotypes. It was found that phenotypes typically affected the embryo either along the dorso-ventral axis (D-V) or the antero-posterior (A-P) axis, but rarely both. In particular, there is a class of phenotypes called gap genes that causes entire regions of the A-P axis to be missing.

Gap Genes

The gap genes affecting the Drosophila embryo along the A-P axis are *hunchback* (*hb*), giant (*gt*), *Krüpple* (*kr*) and Knirps (*kni*); as well as *tailless* (*tll*) and *huckebein* (*hkb*) that specifically affect the extreme posterior (terminal) end of the embryo. However, there are two networks (D-V and Terminal) that have a strong affect on the regulation of *tll* and *hkb* in the anterior and posterior regions of the embryo and are not affected at all by *hb*, *gt*, *Kr*, and *kni*. Thus, we will not focus on *tll* or *hbk* and restrict our study to exclude regions of the A-P axis regulated by them. With this in mind, the term **trunk gap genes** will refer to only the genes *hb*, *gt*, *kr*, and *kni*.

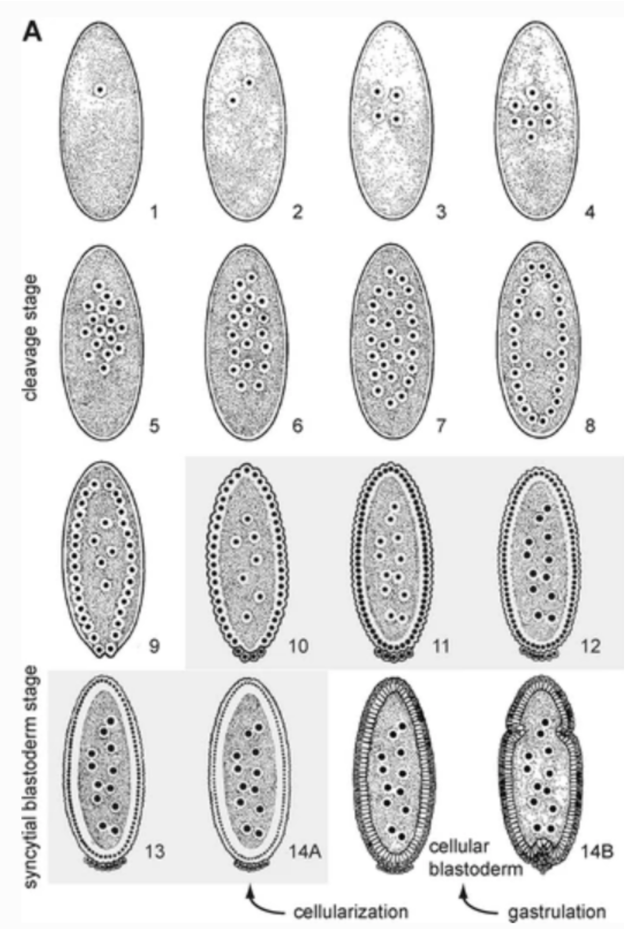


Figure 3.1: Early development of the *Drosophila* embryo. (a) The numbers indicate cleavage cycle. Figure from [5], reproduced/adapted with permission from jcs.biologists.org.

The trunk gap genes are, in part, regulated by maternal protein gradients, *bicoid* (*bcd*), *caudal* (*cad*), *nanos* (*nos*) and maternal *hb*. 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. Since Bcd represses Cad it creates a gradient in the opposite direction i.e. decreasing from posterior to anterior. The Bcd and Cad gradients will be referred to as the **maternal gradients**. Similarly, maternal Hb proteins are uniformly spread throughout the egg and Nos proteins are concentrated at the posterior. After fertilization, Nos diffuses creating a gradient while repressing maternal Hb creating a gradient opposite of Nos. However, Nos has only been found to regulate maternal Hb and maternal Hb is mostly gone by cellularization.

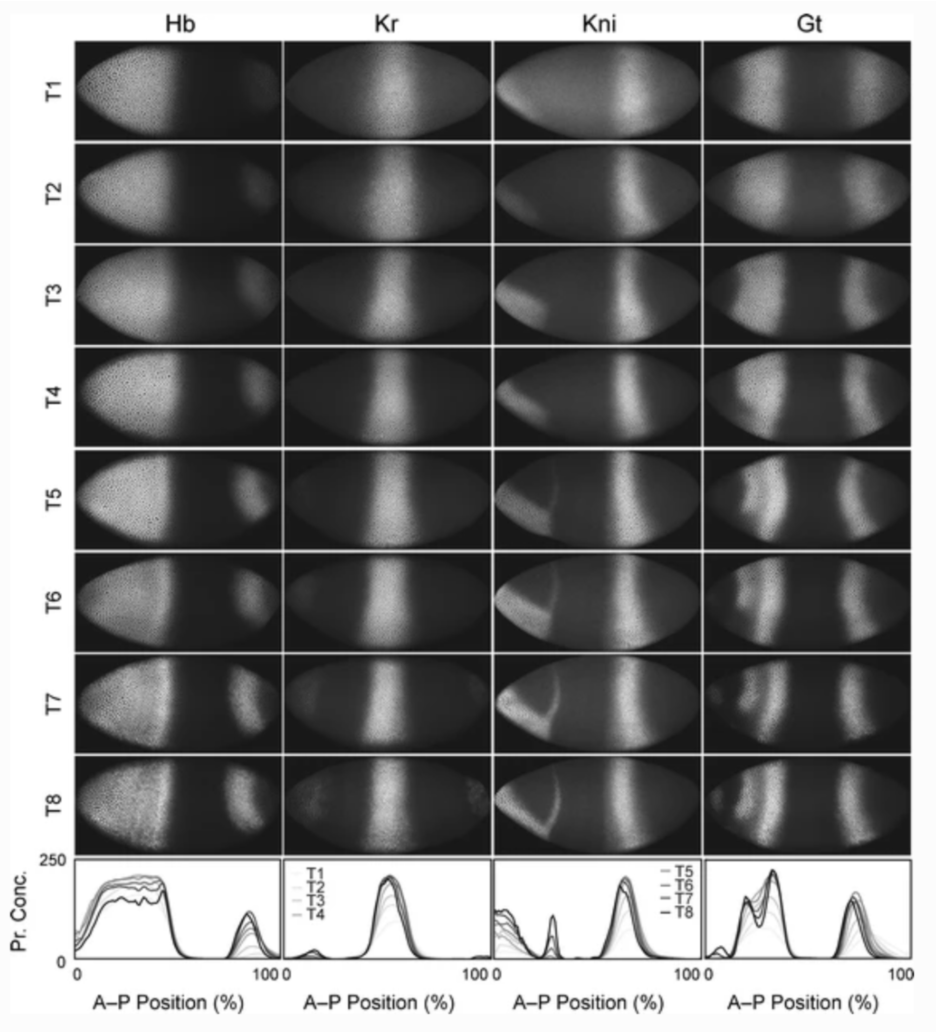


Figure 3.2: Trunk gap gene expressions during each time class T1 through T8 during cycle 14A, which shows gap gene domain boundaries sharpening. The light areas show the high protein expression for each of the trunk gap genes along the A-P axis. The plots shown in the bottom row shows the protein expression data for time classes T1 through T8. Figure from [5].

Trunk Gap Gene Regulation

In the blastoderm stage of the embryo, regulation of the trunk gap genes happens in two phases. Initially, expression of the genes is regulated by the maternal gradients. These gradients give rise to **protein expression patterns**, which are regions along the A-P axis where each gene has high or low concentration. These regions have variable domain boundaries (i.e., where a region is transitioning from high concentration to low and vise versa) from embryo to embryo. Additionally, domain boundaries are not well defined and sharp, but more closely resemble gradients in concentration. Beginning in cycle 13, however, interactions, mostly repressive, between the genes start to develop and variability (from embryo to embryo) in trunk gap gene expression starts to decrease. Cycle 14A shows the most dramatic decrease in this variability. This cycle is typically broken down in eight time classes (T1-T8). During this time, as seen in Figure 3.2, the domain boundary regions sharpen along the A-P axis as time progresses from T1 to T8.

The late stage gap gene regulation (time class T1-T8) can be described by four main **regulatory mechanisms**: (1) activation by maternal gradients, (2) auto-activation, (3) strong repressive feedback between complementary genes and (4) weak regulation between genes with overlapping overlapping boundaries. We describe these mechanisms in more detail.

1. *Activation by maternal gradients*: During early gap gene expression, *bcd* and *cad* play a major role in beginning to form the boundaries. However, by cycle 14A, *bcd* and *cad* activation only contributes to maintaining gap gene expression while the trunk gap genes are the controlling factor in sharpening the boundaries.

2. *Auto-activation*: Many early models of the gap gene network showed that auto-activation of each gene was essential, 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. Experimentally, *hb* has the strongest evidence for auto-activation.

3. *Strong repressive feedback between complementary genes*: The strongest experimental evidence for trunk gap gene regulation during cycle 14A is between *hb* and *kni*, and *kr* and *gt*. Both pairs exhibit a mutual strong repression with each other, called **repressive feedback**.

4. *Weak regulation between overlapping genes*: There is also experimental evidence that there are interactions between the genes that have overlapping boundaries, though the exact type and strength of these interactions has only been examined by mathematical models.

The gap gene regulatory network

Though many models have been shown to faithfully replicate the expression of the trunk gap genes, we will focus our study to the gap gene regulatory network and as described in [9] shown in Figure 3.3(a). A spatial representation of the gap gene regulatory network (Figure 3.3(b)) shows the interactions between the trunk gap genes along the A-P axis and shows each of the regulatory mechanisms. Most notably, the regulation between overlapping genes only happens in specific domains which is visible in the spatial representation but not the regulatory network. Additionally, the spatial representation gives a better understanding of how the maternal gradients are affecting the network at different A-P positions. Recall from section 3 that there is evidence for strong repressive feedback between *hb* and *kni* as well as *gt* and *kr*. This strong repressive feedback is depicted as bold edges in Figure 3.3. Additionally, there is experimental evidence of weak regulation between overlapping genes. The strength of these interactions has been assessed using mathematical models [5]. The stronger ones are depicted as solid edges in Figure 3.3 while the weaker ones are dashed. For the remainder of this thesis, we will call the dotted edges the *weak edges* and the rest the *strong edges*. We will make the reasonable assumption that stronger edges with more experimental evidence are more likely to be the dominating regulatory factors in the protein expression levels.

Dynamic Modules

During their study of the gap gene regulatory network in Figure 3.3(a), Verd et al [9] partitioned a reduced version of the spatial representation into three subnetworks they described as dynamic modules. For our purposes, a **dynamic module** of the gap gene regulatory network is a subgroup of the genes that controls protein expression in a region of the A-P axis. They showed 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. 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. Thus, they create three dynamic modules called AC/DC1, AC/DC2 and AC/DC3 which are active in regions 1, 2 and 3 respectively, which are shown in Figure 3.4(d). They showed that these subnetworks were capable of reproducing the boundary shifts seen biologically from time class T1 to T8. One of the purposes of this study is to show that the decomposition into dynamic modules is an unnecessary step to reproduce the data.

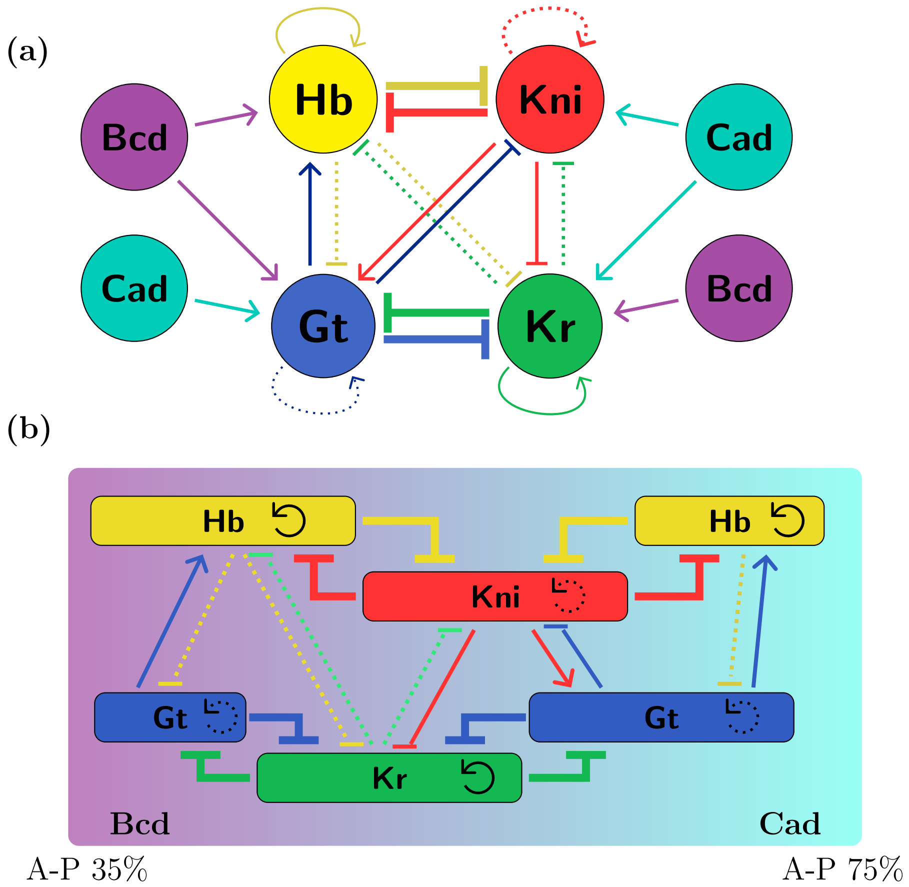


Figure 3.3: (a) The gap gene network used in [9]. The edge widths depict the strength of the interaction; dotted edges are the weakest interactions and the bold edges are the strongest. (b) Spatial representation of gene regulation from anterior - posterior (A-P) position 35% to 75% for the gap gene network in (a). 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 protein expression in time class T8. Figure adapted from [9].

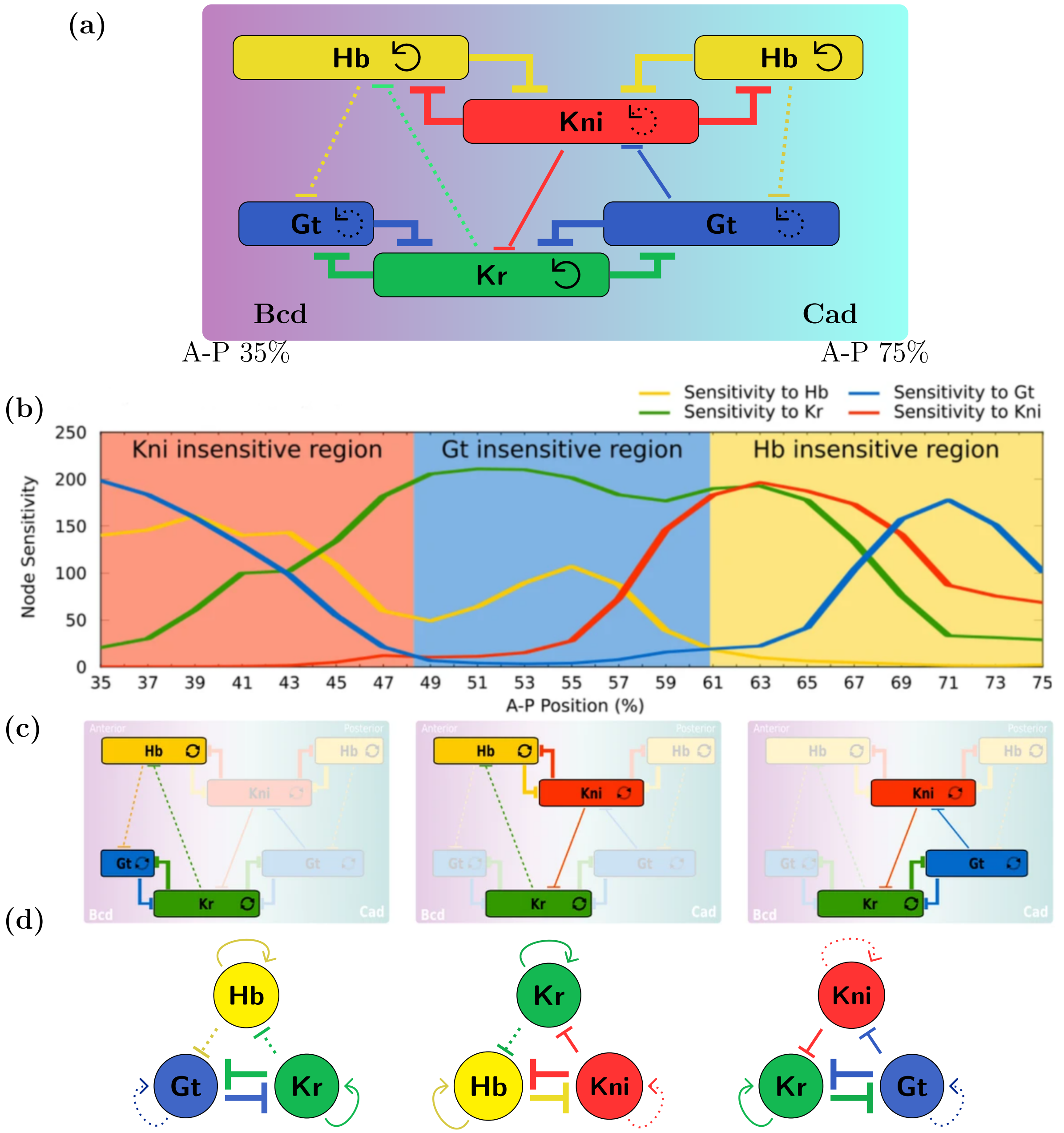


Figure 3.4: (a) The Verd et al [9] representation of high protein expression along the A-P axis. (b) Node insensitivity regions from Verd et al [9] (c) Identification of active nodes along spatial domains 35-47%, 49-59% and 61-75% identified in Verd et al [9]. (d) The AC/DC subnetworks. Figure adapted from [9].

INNOVATIONS

Early development of *Drosophila* 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. Experimental data of the segmentation process show that the outcome of the segmentation dynamics does not depend on the precise value of the parameters [6]. We plan to use the DSGRN parameter graph to investigate this robustness, which we hypothesize arises from the structure of the underlying gene regulatory networks. The Verd et al [9] analysis of the gap gene network represented the dynamics of the gap gene expression using different networks that are responsible for generating different patterns in different segments along the A-P axis. We hypothesize that the expression pattern represents dynamics of a single network, but at different parameters in different regions of the A-P axis. Therefore, we represent a spatial pattern of expression as a path in the DSGRN parameter graph. We will study the robustness of two different network models by looking at the number of paths that can produce the observed dynamics in each network. The extension of DSGRN from a tool for time-dependent intracellular modeling to time- and space-dependent tissue-level modeling is a significant advancement in the utility of DSGRN.

Specifically, a *developmental path* is a path in the DSGRN parameter graph where each vertex is labeled according to its associated Morse graph. A *matching developmental path* is a path whose labels match the phenotypes observed in the experimental data. In the following, we first discuss how we can interpret the spatial chemical gradients of Bcd and Cad across a *Drosophila* embryo as a sequence of DSGRN parameter changes. Using this interpretation, we then construct a graph, called the *phenotype graph*, that organizes all possible developmental paths. While this graph contains all matching developmental paths, the number of all such paths is prohibitively large to allow computationally effective search. Therefore, we develop methods of analysis that use the condensation of the phenotype graph, which allows efficient searching.

Factor graph layers

In this section we discuss how to define a direction on the factor graph by imposing a partial order. We begin by discussing the stereotyped structure of the factor graph.

Let $\Theta_{j,i} = \{\theta_{j1,i}, \theta_{j2,i}, \cdots \theta_{j_{|T(i)|},i}\}$ be the collection of the thresholds of v_i. Let $O_{j,i} = \theta_{j1,i} < \theta_{j2,i} < \cdots < \theta_{j_{|T(i)|},i}$ be a threshold order for node v_i in $\mathbf{RN}$ and let O_i be the set of all such orders.

The factor graph $F_i = (V_i, E_i)$ of v_i (see Definition 2.23) has an interesting structure, whereby it contains $|T(i)|!$ isomorphic subgraphs. The union of the nodes of these subgraphs is the entire set V_i. The key observation is that orders O_i are related by a group of permutations $\mathcal{P}_{|T(i)|}$, where each $\eta \in \mathcal{P}_{|T(i)|}$ permutes threshold labels. As a consequence each for each parameter $p \in V_i$ with a threshold order $O_{j,i}$ there are $|T(i)|!$ other parameters p_η, $\eta \in \mathcal{P}_{|T(i)|}$, where threshold labels are swapped based on η. Therefore each factor graph $F_i = (V_i, E_i)$ contains a collection of $|T(i)|!$ isomorphic subgraphs $G_{j,i} = (V_{j,i}, E_{j,i})$ where the $V_{j,i}$ partition V_i, i.e., $\bigsqcup_j V_{j,i} = V_i$ [2]. We call each of these subgraphs $G_{j,i}$, associated to a particular threshold order $O_{j,i}$, a **subfactor graph**.

Notice that each factor graph $F_i = (V_i, E_i)$ has a set of "lowest parameters" composed of the collection of parameter nodes

$$Lp_i := \{p \in V_i \mid M_i(u_{i,k_1}, u_{i,k_2}, \ldots, u_{i,k_{|S(i)|}}) < \theta_{j,i} \text{ for all } \theta_{j,i} \in \Theta_{j,i}\},$$

where recall that M_i is the algebraic expression for gene product v_i and u_{i,k_n} are the higher expression levels of v_i, as opposed to the lower expression levels l_{i,k_n}. Similarly, there is a set of "highest parameters" composed of the nodes

$$Hp_i := \{p \in V_i \mid M_i(l_{i,k_1}, l_{i,k_2}, \ldots, l_{i,k_{|S(i)|}}) > \theta_{j,i} \text{ for all } \theta_{j,i} \in \Theta_{j,i}\}.$$

We say that a node $\ell_{j,i} \in Lp_i$ is a **root** of a subfactor graph and a node $h_{j,i} \in Hp_i$ is a **leaf** of a subfactor graph. Ideally, we would like to have a number of nice properties with regard to the subfactor graph, particularly:

1. connectedness,
2. a unique root $\ell_{j,i} \in Lp_i$ and a unique leaf $h_{j,i} \in Hp_i$ for the subfactor graph $G_{j,i}$, and
3. every $v_i \in V_{j,i}$ participates in a minimal-length path from $\ell_{j,i}$ to $h_{j,i}$.

If (1)-(3) hold, then we are able to define a partial order on the graph that gives a well-defined direction of motion from root to leaf. Proving these or similar results is a work in progress.

For now, we have constructed by hand a few low-dimensional cases, including those used in this thesis. In these cases, M_i may be any product of sums of $|S(v_i)|$ inputs. The constructed subfactor graphs are

- 1 in-edge, 2 out-edges or, isomorphically, 2 in-edges, 1 out-edge;
- 2 in-edges, 2 out-edges or, isomorphically, 3 in-edges, 1 out-edge;
- 2 in-edges, 3 out-edges.

We verified that conditions (1)-(3) hold in these cases. An example of a factor graph with 1 in-edge and 2 out-edges can be seen in Figure 4.1. This factor graph example has two subfactor graphs, where all the nodes with threshold order $\theta_1 < \theta_2$ belong to the subfactor graph on the left and all the nodes with threshold order $\theta_2 < \theta_1$ belong to the subfactor graph on the right. The factor graph for 2 in-edges and 1 out-edge is in Figure 4.2 (a).

Definition 4.1. *Any path between $\ell_{m,i}$ and $h_{n,i}$ for $n, m \in V_{j,i}$ will be called a* ***maximal path*** *in the factor graph. Let $p_0, \ldots, p_N$ be a shortest path between $\ell_{j,i}$ and $h_{j,i}$ within the same subfactor graph. Call such a path a* ***minimax path*** *in the factor graph. Call $N + 1$ the* ***minimax path length*** *of the factor graph.*

For our special cases, we define the layers of the factor graph by the minimum path length to a root of the factor graph.

Definition 4.2. *Let $F_i = (V_i, E_i)$ be the factor graph for node i, with subfactor graphs $G_{j,i} = (V_{j,i}, E_{j,i})$ for $j \in \{1, \ldots, |T(i)|!\}$. The shortest path length, $\mathcal{L}_n$, between $p_n \in V_{j,i}$ and $\ell_{j,i}$ is called the* ***factor graph layer of*** $\mathbf{p_n}$. *The* ***k-th factor graph layer of*** $\mathbf{F_i}$ *is the node set*

$$\{p_n \in V_i \mid \text{ with shortest path length } \mathcal{L}_k\},$$

for $k \in \{0, \ldots, N\}$, where $N + 1$ is the minimax path length.

We say that the **highest factor graph layer** is the set Hp_i, any element of which has path length N to a root $\ell_{j,i}$, and is therefore factor graph layer N. Likewise, the **lowest factor graph layer** is the set Lp_i, which has path length 0 to itself, and is therefore factor graph layer 0.

For example, consider a node v with one in-edge and two out-edges with thresholds θ_1 and θ_2. Then there are twelve ways to order θ_1 and θ_2 together with the partial order $0 < l < u$ as shown in Figure 4.1. Note that factor graph has five layers. Additionally, the highest and lowest factor graph layers are $\theta_1, \theta_2 < l$ and $u < \theta_1, \theta_2$ respectively. The idea of factor graph layers allows us to define an idea of monotonicity of paths through a factor graph.

Definition 4.3. *Consider a path through the factor graph $F_i = (V_i, E_i)$ of node v_i, represented as a sequence of nodes $p_1, p_2, \ldots, p_m$, where each $p_j \in V_i$. The path is said to be* ***monotone increasing*** *if for all p_j, p_k in the path, we have $\mathcal{L}_j \leqslant \mathcal{L}_k$ if and only if $j \leqslant k$, i.e. factor graph layers increase along the path. Similarly, the path is said to be* ***monotone decreasing*** *if for all p_j, p_k in the path, we have $\mathcal{L}_j \leqslant \mathcal{L}_k$ if and only if $j \geqslant k$, i.e. factor graph layers decrease along the path.*

Interpreting maternal gradients as parameter changes

In ODE systems, observed dynamics can change based on varying initial conditions or varying parameters. In DSGRN, initial conditions are represented as nodes in the STG, while parameters are nodes in the PG. The question here is whether changing initial conditions or changing parameters is a more appropriate model for spatially distributed dynamics, both in terms of a biological interpretation and within the DSGRN context. Given a gene regulatory network $\mathbf{RN} = (V, E)$, we can model the impact of a control variable (or network regulator), such as one of the maternal gradients of *Drosophila*, as an **explicit** network node or as an **implicit** environmental effect.

If a control variable c has an activating (inhibiting) effect on node $v_i \in V$, the explicit representation would add the node c and edge $c \to v_i$ ($c \dashv v_i$) to the network $\mathbf{RN}$. After adding all appropriate target edges, the control variable c would be capable of expressing discretized initial conditions anywhere in the range $\{0, 1, \ldots, |T(c)|\}$. After adding all control variables explicitly to the network, the goal would be to locate parameters for which the

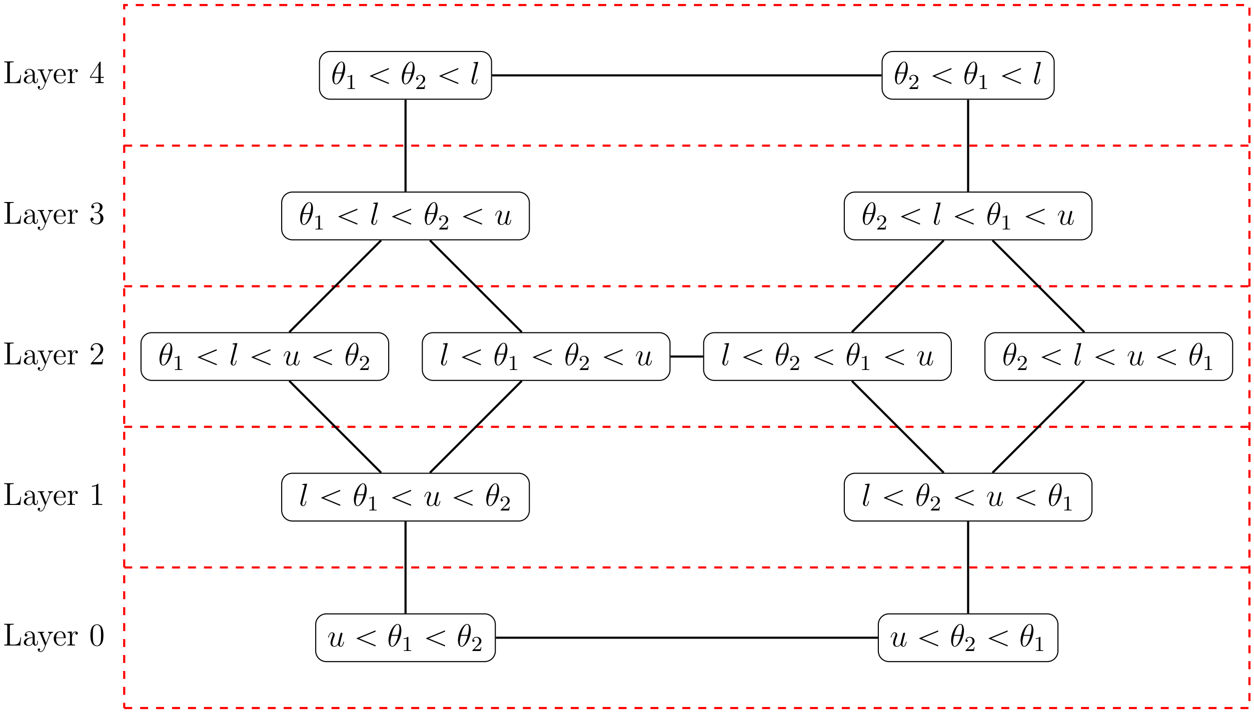

Layer 4
Layer 3
Layer 2
Layer 1
Layer 0
$\theta_1 < \theta_2 < l$
$\theta_2 < \theta_1 < l$
$\theta_1 < l < \theta_2 < u$
$\theta_2 < l < \theta_1 < u$
$\theta_1 < l < u < \theta_2$
$l < \theta_1 < \theta_2 < u$
$l < \theta_2 < \theta_1 < u$
$\theta_2 < l < u < \theta_1$
$l < \theta_1 < u < \theta_2$
$l < \theta_2 < u < \theta_1$
$u < \theta_1 < \theta_2$
$u < \theta_2 < \theta_1$

Figure 4.1: Factor graph for a node v with one in-edge and two out-edges with thresholds θ_1 and θ_2. Inputs have a partial order $l < u$. Layers 0-4 denote factor graph layers. Layer 4 is composed of the *highest factor graph nodes* for v. Similarly, layer 0 is composed of the *lowest factor graph nodes* for v.

Morse graph contains all elements of the phenotype pattern simultaneously, each associated to a different set of control variable initial conditions.

For an implicit representation of a control variable c, we introduce here an important subset of possible behaviors of c.

Definition 4.4. *A* ***monotone control variable*** *c, is a control variable that is either $c' \geqslant 0$ (increasing) or $c' \leqslant 0$ (decreasing) everywhere on its domain.*

An implicit representation of a control variable is implemented as a parameter change at a target node v_i and does not add c explicitly as a node to $\mathbf{RN}$. We assume that the control variable is the dominating effect on parameter changes in v_i through the following modeling rules.

1. If c is an activator (repressor) and is present in high amounts (low amounts), then v_i is present in high amounts.
2. If c an activator (repressor) and is present only in low amounts (high amounts), then v_i is present in low amounts.
3. A monotone control variable c induces a monotone response in its targets.

We elaborate on the last point. Let $r_i = \pm 1$, where $+1$ means that c is an activator and -1 means c is a repressor to a target node v_i. Let F_i be the factor graph of v_i. We model the changing behavior of c as monotone paths in F_i (see Definition 4.3) as follows; c induces monotone increasing paths in F_i when $r_i \cdot \text{sign}(c') = +1$ and monotone decreasing paths when $r_i \cdot \text{sign}(c') = -1$. This monotonicity condition on the factor graph of v_i (see Figure 4.2) is a model of continuously changing abundance of v_i as a function of changing c.

There are computational advantages to using an implicit modeling scheme because the network $\mathbf{RN}$ is smaller without the explicit control variable modeling. In addition, there is a solid biological argument for choosing implicit modeling of maternal gradients. Recall that during late stage gap gene regulation, the maternal gradients are only maintaining gap gene expression while the interactions between the trunk gap genes are controlling the dynamics. Additionally, note that at any point along the A-P axis, the maternal gradients are relatively constant. That is, within the context of a single cell there is not a significant continuous change in the gradients. An interpretation of this is that the control variables of the network, *bcd* and *cad*, are a part of the environmental conditions of the cell and not an active participator of the network. In this interpretation, it is reasonable to model the spatially distributed dynamics as a change in parameters of the trunk gap genes rather than a change of initial conditions of the maternal gradients across the entire A-P region.

As an example of implicit modeling in the gap gene network, consider the activation of *bcd* on *hb* at A-P position 40% Egg Length. At that position, Bcd is high so we would expect *hb* to exhibit consistently high expression. However, at A-P position 75% Egg Length, Bcd is at it lowest so we would expect *hb* to be at consistently low expression levels. In between these two extremes, we expect monotone behavior. Moreover, we can impose a stricter condition on the modeling of the maternal gradients that requires *hb* to not only exhibit consistently high and low expression, but to operate at the most extreme factor graph layers.

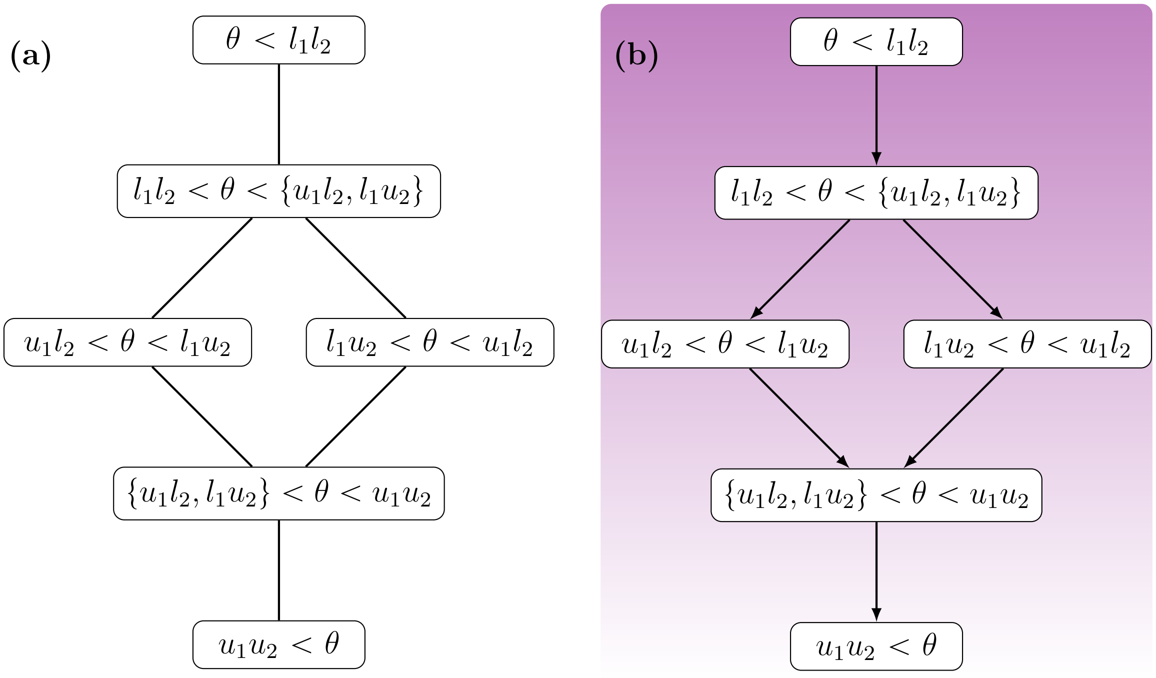

(a)
$\theta < l_1l_2$
$l_1l_2 < \theta < \{u_1l_2, l_1u_2\}$
$u_1l_2 < \theta < l_1u_2$
$l_1u_2 < \theta < u_1l_2$
$\{u_1l_2, l_1u_2\} < \theta < u_1u_2$
$u_1u_2 < \theta$
(b)
$\theta < l_1l_2$
$l_1l_2 < \theta < \{u_1l_2, l_1u_2\}$
$u_1l_2 < \theta < l_1u_2$
$l_1u_2 < \theta < u_1l_2$
$\{u_1l_2, l_1u_2\} < \theta < u_1u_2$
$u_1u_2 < \theta$

Figure 4.2: (a) Factor graph for a node v_i with two in edges (one activating and one repressing) and one out edge. Inputs have a partial order $l_1l_2 < \{u_1l_2, l_1u_2\} < u_1u_2$ and every node of the factor graph has all inputs, though only position of thresholds between neighboring inputs is depicted. (b) The same factor graph with an activating control variable c imposed, depicted as a decreasing gradient (violet). Then, as c decreases it induces decreasing monotonicity through the factor graph (directed arrows).

Definition 4.5. *A **maximal monotone path** in the factor graph F_i is either*

1. *an increasing monotone path (see Definition 4.3) that starts in the lowest factor graph layer and ends in the highest factor graph layer, or*

2. *a decreasing monotone path that starts in the highest factor graph layer and ends in the lowest factor graph layer.*

*A **direct maximal monotone path** M is a maximal monotone path that is also a minimax path (see Definition 4.1).*

We will impose the condition of a maximal monotone path in our simulations presented in Section 6.

Interpreting data combinatorially

At each position of the A-P axis the experimental data collected during times T1-T8 suggest that the concentration of the gap gene proteins evolve in time until they reach equilibrium value at time point T8, this can be seen graphically in Figure 3.2. Figure 4.3 shows this data at T8 across all positions of interest where the trunk gap genes are thought to be driving this behavior. Note that there is significant correlation between values at nearby positions. Therefore, in agreement with Verd et al [9], we break down the A-P axis into eight regions R_n (see Figure 4.3), where the limiting values at T8 are distinct. It is a coincidence that there are both eight temporal regions and eight spatial regions; they are not correlated in any way.

Since each region R_n has approximately achieved equilibrium, it is reasonable to model the dynamics of each region as a fixed point. We must express these fixed points in the language of DSGRN. In order to match the experimental values to coordinates of equilibria in DSGRN, we discretize the experimental values to classes that match the resolution of DSGRN equilibrium labels. We label the concentration of each gene product in each of the 8 regions by one of the labels $\{H, *, L\}$:

1. H : Gene expression is high,

2. L : Gene expression is low,

3. $*$: Gene expression is indeterminate.

This discretization is to some extent arbitrary, but Boolean modeling (i.e. H = 1, L = 0) is a common way of discretizing data. Since in our case, there are large regions where it is unclear whether gene expression should be regarded as high or low, we introduce the third character $*$, which is taken to be either one.

Recall that Verd et al [9] showed *kni* is inactive between A-P positions 35-47%, *gt* is inactive between 49-59% and *hb* is inactive between 61-75%. Thus, in these regions, the gene expression would be below all thresholds and labeled **L**. Finally, the labels **H**, $*$, **L** will be

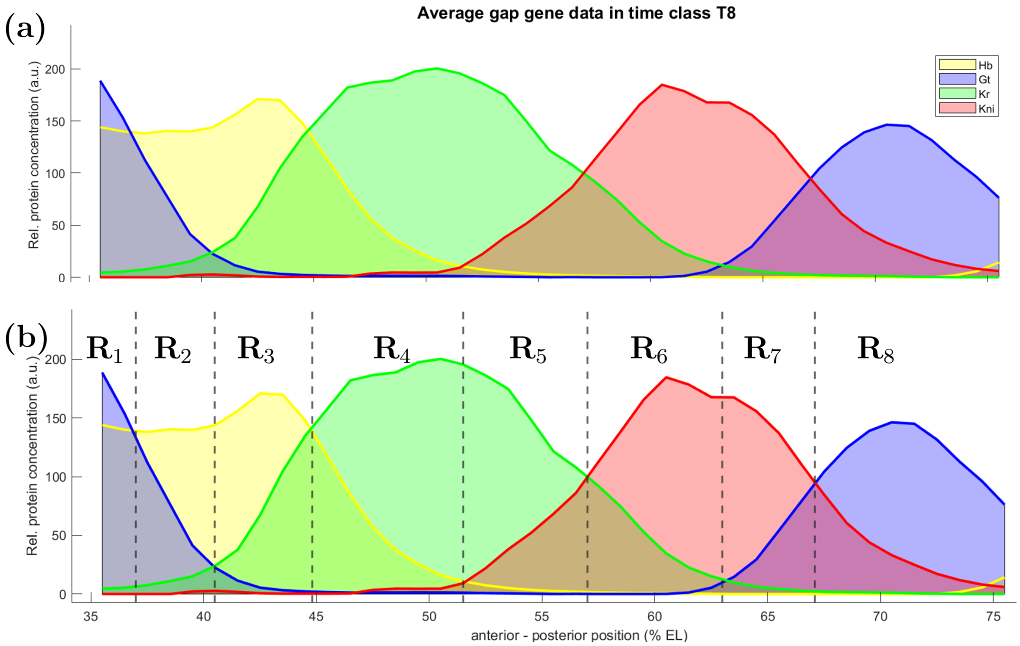


Figure 4.3: Data of protein expression expression along the anterior-posterior position % Egg Length (EL) for the trunk gap genes *hunchback* (*hb*) in yellow, *giant* (*gt*) in blue, *Krüpple* (*Kr*) in green and *Knirps* (*kni*) in red for time class T8. The gap gene expression data (S1_Data.ods) was obtained from the supplementary information in [8].

given to genes when their expression is high, transitioning between high and low or low and high, and low, respectively. See Figure 4.3. Therefore, we have the data discretization seen in Figure 4.4.

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

Figure 4.4: Data discretization.

This discretization can then be used to look for matching developmental paths through the DSGRN parameter graph of the gap gene regulatory network. In order to match the experimental data to labels of the DSGRN equilibria, we interpret the data discretization labels $\{H, *, L\}$ as follows:

1. H : Gene v_i concentration is in its highest numerical state, $|T(v_i)|$, i.e. above all of its thresholds.

2. L : Gene v_i concentration is in its lowest numerical state, 0, i.e. is below all of its thresholds,

3. $*$: Gene v_i concentration may achieve any value in the integer set $\{0, \ldots, |T(v_i)|\}$.

In each of the eight spatially distinct regions, we specify the discretized expression level of each of the four trunk gap genes, *hb*, *gt*, *kr*, and *kni*. Then we can represent each region as a labeled node.

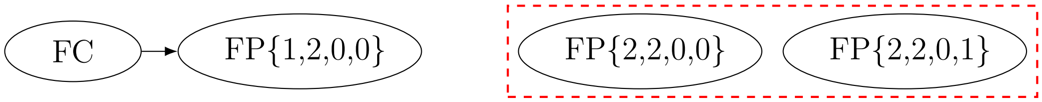
Definition 4.6. *A* ***pattern label*** *is a discretized relative expression level for each gap gene of the form* $\Sigma = (\Sigma_{Hb}, \Sigma_{Gt}, \Sigma_{Kr}, \Sigma_{Kni})$*, where* $\Sigma_A \in \{0, 1, \ldots, |T(v_A)|\}$*. A* ***phenotype pattern*** *is an ordered collection of pattern labels, one for each* R_n *position along the A-P axis. A* ***phenotype*** *of* Σ *is any DSGRN Morse graph with a fixed point with label* Σ*. A* ***strict phenotype*** *of* Σ *is any DSGRN Morse graph with a monostable fixed point with label* Σ*. We use the notation* $\Sigma(R_n)$ *to identify the set of phenotypes corresponding to the pattern label* Σ *for the A-P region* R_n*.*

Phenotype graph

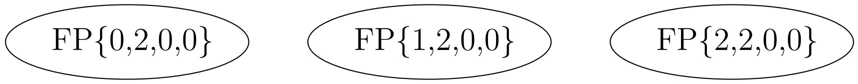
The phenotype pattern describes the annotations of the DSGRN Morse graphs that would match the data in each of the eight regions along the A-P axis. Thus, we are only interested in parameter graph nodes that are associated to the annotated Morse graphs (the phenotypes) described by the phenotype pattern. For example, in region R_1, where the discretized data are described by ($*$, H, L, L), then the associated phenotypes for the network in Figure 6.1 (a) are the annotated Morse graphs that contain any fixed point in the set

$$\{\mathrm{FP}\{0, 2, 0, 0\}, \mathrm{FP}\{1, 2, 0, 0\}, \mathrm{FP}\{2, 2, 0, 0\}\},$$

when all of the nodes in the regulatory network have two in-edges and two out-edges. Then any Morse graph containing one of the above fixed points are associated phenotypes for R_1. For example, a Morse graph having a full cycle to FP{ 1, 2, 0, 0} as shown below, or a multistable Morse graph shown below boxed in red, are associated phenotypes for R_1.



Additionally, the strict phenotypes are the annotated Morse graphs with any of the fixed points as the unique monostable element of the Morse graph, thus any of the following three Morse graphs are the associated strict phenotypes for R_1.



The subset of the parameter graph associated to the desired phenotypes is formalized in the following definition.

Definition 4.7. *Let $V_i := \{v \in PG \mid MG(v) \in \Sigma(R_i)\}, \quad i = 1, \ldots, 8,$ be the sets of parameter nodes with a Morse graph that contains a FP-labeled Morse node that matches some phenotype in $\Sigma(R_i)$. A* ***phenotype graph*** *is a graph $PhG = (V, E)$ with nodes*

$$V := \bigsqcup_{i=1}^{8} V_i.$$

Note that the same node $v \in PG$ may be included in the set V multiple times, if the phenotype (or Morse graph) of v exists in multiple regions, i.e. $MG(v) \in \Sigma(R_i)$ for multiple values of i.

The nodes $v_i, v_j \in V$, where v_i, v_j represent distinct nodes in PG, are connected by a directed edge $(v_i \to v_j)$, if and only if both of the following are satisfied:

1. *there is an undirected edge between v_i and v_j in the parameter graph PG, and*

2. *there exists a region R_n, for $n \in \{1, ..., 8\}$, such that $MG(v_i) \in \Sigma(R_n)$ and $MG(v_j) \in \Sigma(R_n) \cup \Sigma(R_{n+1})$.*

The nodes $v_i, v_j \in V$, where v_i, v_j are copies of the same node $w \in PG$, are connected by a directed edge $(v_i \to v_j)$ if and only if

1. *$MG(w) \in \Sigma(R_n) \cap \Sigma(R_{n+1})$,*

2. *$v_i \in V_n$, and*

3. *$v_j \in V_{n+1}$.*

A ***strict phenotype graph*** *is defined exactly the same except that all $\Sigma(R_n)$ are composed of strict phenotypes.*

A ***developmental path*** *is any path in the phenotype graph. A* ***matching developmental path*** *is a path in the phenotype graph $v_{i_1} \to \cdots \to v_{i_m}$ such that*

1. *the nodes $\{v_{i_j}\}$ are distinct,*

2. *$MG(v_{i_1}) \in \Sigma(R_1)$ and $MG(v_{i_m}) \in \Sigma(R_8)$, and*

3. *for each $k = 2, ..., 7$, there exists at least one v_{i_j} such that $MG(v_{i_j}) \in \Sigma(R_k)$.*

Notice that by construction, a matching developmental path $v_{i_1} \to \cdots \to v_{i_m}$ satisfies the condition that a representative phenotype from each region is passed through *in order* along the path. For a given network model, it is not guaranteed that there are any matching developmental paths. In the worst case, the phenotype graph is empty, indicating that no phenotypes seen in the data can be reproduced by the network model.

Recall that we plan to model the maternal gradients as control variables and therefore we want to enforce monotone paths through any factor graph of a regulatory target of the maternal gradients. The phenotype pattern is constructed based on the discretized experimental data, which were collected in the presence of maternal gradients. Therefore the information about material gradients is implicitly present in the phenotype pattern. However, this information is not enough to guarantee that all matching developmental paths have the desired monotone property when projected onto a factor graph of interest. Thus, we also consider a subgraph of the phenotype graph in which every matching developmental path satisfies a monotonicity requirement.

Definition 4.8. *Let $F_{i_1}, \ldots, F_{i_m}$ be a subset of the set of factor graphs associated to m target nodes of a collection of monotone control variables $c_1, \ldots, c_q$. A* ***monotone (strict) phenotype graph*** *with respect to $F_{i_1}, \ldots, F_{i_m}$ is a subgraph $PhG' = (V', E')$ of the (strict) phenotype graph $PhG = (V, E)$ such that the projection of every matching developmental path in PhG' onto F_{i_j} is a maximal monotone path (see Definition 4.5) in F_{i_j} for all $j \in \{1, \ldots, m\}$.*

Let $r_{i_j,k} = +1$ ($r_{i_j,k} = -1$) when c_k is an activator (repressor) of $v_{i,j}$. The maximal monotone path in F_{i_j} is increasing monotone (decreasing monotone) when $r_{i_j,k} \cdot sign(c'_k) = +1$ ($r_{i_j,k} \cdot sign(c'_k) = -1$). We also assume that the paths in the monotone phenotype graph are consistent; i.e.

$$r_{i_j,k} \cdot sign(c'_k) = r_{i_j,m} \cdot sign(c'_m)$$

whenever c_k and c_m are both regulators of the same node v_{i_j}.

For example, let's consider a node v_i with two in-edges, one activating ($l_1 < u_1$) and one repressing ($l_2 < u_2$), as well as one out-edge with threshold θ and its factor graph F_i with. Then there are six ways to form a partial order of $l_1l_2 < \{l_1u_2, u_1l_2\} < u_1u_2$ and the threshold θ, see Figure 4.2(a). Let c be an activating monotone control variable with $c' \leqslant 0$. Then a matching developmental path in the monotone phenotype graph projected onto F_i is required to be a maximal monotone path. Since c is an activator but decreasing, then the maximal monotone path is decreasing monotone (i.e., it starts at the highest factor graph layer and ends in the lowest factor graph layer). A visualization of how this matching developmental path must behave in the projection onto F_i can be seen in Figure 4.2(b). In this figure, we are representing c as the violet monotone decreasing gradient, the directed arrows show the only directions the decreasing monotone path can go.

Ideally, one could find all matching developmental paths in the phenotype graph and monotone phenotype graphs, then use it as a measure of the robustness of the network model to produce the observed A-P positional dynamics. In practice, the size of phenotype graph scales poorly with increasing regulatory network size, much as the DSGRN parameter graph does, and it becomes computationally impractical even to find a single path through the phenotype graph. To address this issue, we considered two different coarsenings of the phenotype graph that conserve the order of the phenotype pattern.

Definition 4.9. *The* ***coarsened (strict) phenotype graph*** *and* ***coarsened monotone (strict) phenotype graph*** *are the condensations of the (strict) phenotype graph and the monotone (strict) phenotype graph, respectively.*

Definition 4.10. *Let u be a node in the coarsened phenotype graph, possibly monotone and/or strict, and denote by $scc(u)$ the strongly connected component of PhG associated to u. A* ***phenotype cluster path*** *$u_1 \to u_2 \to ... \to u_m$ is a path in a coarsened (monotone, strict) phenotype graph such that*

1. *all $p_1 \in scc(u_1)$ satisfy $MG(p_1) \in \Sigma(R_1)$,*

2. *for each $i = 2, ..., 7$, there exists at least one u_i with $p_i \in scc(u_i)$ such that $MG(p_i) \in \Sigma(R_i)$, and*

3. *all $p_m \in scc(u_m)$ satisfy $MG(p_m) \in \Sigma(R_8)$.*

Notice that by construction, a phenotype cluster path $u_1 \to u_2 \to ... \to u_m$ satisfies the condition that a representative phenotype from each region is passed through in order along the path. Notice also that the phenotype graph and monotone phenotype graphs have a minimum of eight strongly connected components, one for each region R_n, due to the partition of the nodes V of the phenotype graph PhG into eight disjoint sets, $V_1, \dots, V_8$. If $p_i \in V_i$ and $p_j \in V_{i+1}$, then the edge $p_i \to p_j$ may exist in the phenotype graph PhG, but the edge $p_j \to p_i$ may not exist in PhG. Therefore, no two distinct V_i and V_j may co-exist in the same strongly connected component, and each V_i has at least one distinct strongly connected component.

Searching the coarsened phenotype graph for phenotype cluster paths can provide significant computational savings compared to searching a phenotype graph for matching developmental paths, if the coarsened phenotype graph is smaller. In the worst case scenario, the number of strongly connected components is equal to the number of nodes in the phenotype graph but it is typically much smaller. We will now show the desirable property that phenotype cluster paths within a coarsened (monotone, strict) phenotype graph are associated to matching developmental paths in the (monotone, strict) phenotype graph.

Lemma 4.1. *There exists at least one matching developmental path in the (monotone, strict) phenotype graph for every distinct phenotype cluster path in a coarsened (monotone, strict) phenotype graph.*

Proof. Let $u_1 \to u_2 \to ... \to u_m$ be a path in the coarsened (monotone, strict) phenotype graph. Recall for $i = 1, ..., m$ that for each u_i, $scc(u_i) \subset V$ is a strongly connected component of the (monotone, strict) phenotype graph $PhG = (V, E)$. Additionally, for there to be an edge $u_i \to u_{i+1}$, there must exist distinct nodes $p_i \in scc(u_i)$ and $q_{i+1} \in scc(u_{i+1})$ such that $(p_i, q_{i+1}) \in E$. Furthermore, by definition of a strongly connected component, when $|scc(u_i)| > 1$ there exists a path between any two nodes in $scc(u_i)$. Thus, if (p_i, q_{i+1}) is an edge between $scc(u_i)$ and $scc(u_{i+1})$, and likewise (p_{i+1}, q_{i+2}) is an edge between $scc(u_{i+1})$ and $scc(u_{i+2})$, then for $|scc(u_{i+1})| > 1$ there exists a path of distinct nodes in PhG

$$p_i \to q_{i+1} \to v_{j_1} \to ... \to v_{j_n} \to p_{i+1} \to q_{i+2}$$

for $v_{j_1}, ..., v_{j_n} \in scc(u_{i+1})$. If $|scc(u_{i+1})| = 1$, then $q_{i+1} = p_{i+1}$ so clearly there is a path

$$p_i \to q_{i+1} = p_{i+1} \to q_{i+2}.$$

Since we constructed a path in the phenotype graph from $scc(u_i)$ to $scc(u_{i+1})$ for arbitrary i, then we can construct a path in this way for all $scc(u_{i+1})$. Thus, there exists a path in PhG, through $scc(u_1), scc(u_2), \ldots, scc(u_m)$, of the form

$$p_1 \to q_2 \to ... \to q_{i+1} \to v_{j_1} \to ... \to v_{j_n} \to p_{i+1} \to q_{i+2} \to ... \to p_m.$$

By construction of a phenotype cluster path, $p_1 \in scc(u_1)$ with $\mathrm{MG}(p_1) \in \Sigma(R_1)$, $p_m \in scc(u_m)$ with $\mathrm{MG}(p_m) \in \Sigma(R_8)$, and for each $k = 2, \ldots, 7$, there exists a u_i with $p_i \in scc(u_i)$ such that $MG(p_i) \in \Sigma(R_k)$. Lastly, all nodes in a path are distinct by our construction. Thus the path

$$p_1 \to q_2 \to ... \to q_{i+1} \to v_{j_1} \to ... \to v_{j_n} \to p_{i+1} \to q_{i+2} \to ... \to p_m$$

satisfies all the conditions of a matching developmental path in Definition 4.7. ■

SOFTWARE

In this section we give an overview of the algorithm written to find all nodes in DSGRN database that match the phenotype pattern, both the algorithm for constructing the phenotype graph, as well as the path finding algorithm. The algorithm used to compute the condensation graph of the phenotype graphs is a non-recursive version of Tarjan's Algorithm [7] and is modified from the code hosted, at the time of writing this thesis, on GitHub at github.com/alviano/python/blob/master/rewrite_aggregates/scc.py. The path finding algorithm is a depth first search. All of the software developed for this Thesis is hosted on GitHub at github.com/Eandreas1857/dsgrn_acdc.

Algorithm 1 Find all parameter nodes associated to phenotype pattern.

```
procedure PARAMSLIST($PG = (V, E)$, phenotype pattern)
    for $R_n \in$ phenotype pattern do
        paramslist← all $p_i \in V$ such that $MG(p_i) \in \Sigma(R_n)$
    end for
return paramslist
end procedure
```

Algorithm 2 Construct phenotype graph

```
procedure PHENOTYPEGRAPH($PG = (V, E)$, paramslist)
    for $R_p \in$ paramslist do
        while $R_p$ is not empty do
            Pop $v$ from $R_p$
            $V' \leftarrow v$
            $N = \{u : (v, u) \in E$ and $u \in R_p \cup R_{p+1}\}$
            for $u \in N$ do
                $E' \leftarrow (v, u)$
            end for
        end while
    end for
return $PhG = (V', E')$
end procedure
```

Algorithm 3 Path finding.

```
procedure FindAllPaths($PhG = (V', E')$, start set, stop set, max length)
    procedure FindPath(path, max length)
        if last element in path in stop set then
            Paths← path
        else
            for $n$ in $N$ do
                if path length $\leqslant$ max length then
                    return FindPath(path+[n], max length)
                else
                    Break
                end if
            end for
        end if
    end procedure
    for $s$ in start set do
        if path length $\leqslant$ max length then
            $N = \{u : (s, u) \in E'\}$
            return FindPath([s], max length)
        else
            Break
        end if
return Paths
```

RESULTS

We hypothesize that the spatial differences in expression levels of gap genes along the A-P axis result from the same network but under changing parameters. These parameters are imposed by the maternal gradients. To examine this hypothesis we represent both the phenotype and the spatial dependence of parameters in terms of discretized structures used by DSGRN, and we aim to address these questions.

A

1. Can we find network models that are capable of reproducing the discretized data of the spatial gap gene profile?
2. In other words, is there a matching developmental path in the phenotype graph of a network model at which the DSGRN fixed point sequence matches discretized experimental gene expression levels?
3. How many such paths are there? Can we use the abundance of such paths and perhaps their structure to assess robustness of the developmental program within a specific network model?

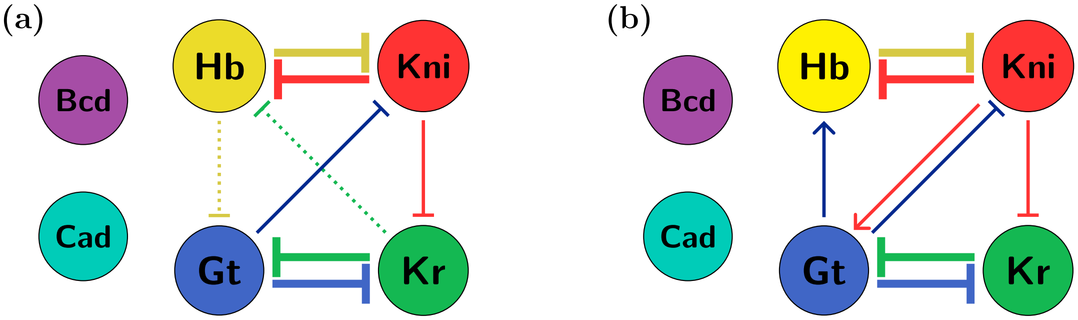
B

1. If we impose monotone control variable modeling on the maternal gradients *bcd* and *cad*, are there still matching developmental paths that reproduce experimental expression levels in all regions R_1-R_8?
2. How does the imposition of the monotone control variable modeling change the number and structure of matching developmental paths?

C Do different candidate networks give different predictions? Is one network better able to match the data than others?

To answer these questions we will examine two networks. One network, which we call the fully connected network (Fullconn), consists of the union of nodes and edges from the three dynamical modules from the Verd et al [9], except the self-loops see Figure 6.1 (a)). The other network, called the strong edges network (StrongEdges) consists of strong edges between the trunk gap genes from the original gap gene network in Figure 3.3, except the self-loops, also pictured in Figure 6.1 (b). We made the choice to exclude self-loops due to computational constraints. This choice is partly justified by the fact that self-regulation was previously found to be unnecessary to reproduce experimental data (see Section 3).

Recall that along the A-P axis, that *bcd* is a decreasing gradient and that *cad* is an increasing gradient. Thus, we will model the maternal gradients *bcd* and *cad* as monotone control variables. Since different nodes in different networks have a variable number of output edges, and output edges determine the number of thresholds in DSGRN for the corresponding variable, the number of output edges affects the range of annotation of each fixed point Morse set. In general, if a network node X has m output edges, the X component of a fixed point (FP) can have values in $\{0, \ldots, m\}$. In our analysis, we always interpret Low (L) experimental level of expression of gene X as a 0 component of a FP, high (H) experimental level of expression of gene X as a m (i.e., the highest component of a FP), and transitioning ($*$) experimental level of expression as any level $\{0, \ldots, m\}$ of a FP. For



Reg.	A-P	Hb	Gt	Kr	Kni
R_1	35-37	[0,2]	2	0	0
R_2	37-40	2	[0,2]	0	0
R_3	40-45	2	0	[0,2]	0
R_4	45-51	[0,2]	0	2	0
R_5	51-57	0	0	2	[0,2]
R_6	57-63	0	0	[0,2]	2
R_7	63-67	0	[0,2]	0	2
R_8	67-75	0	2	0	[0,2]

Reg.	A-P	Hb	Gt	Kr	Kni
R_1	35-37	[0,1]	3	0	0
R_2	37-40	1	[0,3]	0	0
R_3	40-45	1	0	[0,1]	0
R_4	45-51	[0,1]	0	1	0
R_5	51-57	0	0	1	[0,3]
R_6	57-63	0	0	[0,1]	3
R_7	63-67	0	[0,1]	1	3
R_8	67-75	0	3	0	[0,3]

Figure 6.1: (a) The Fullconn network (above), and Fullconn phenotype pattern (below). (b) The StrongEdges network (above) with StrongEdges phenotype pattern (below).

example, in the Fullconn network, each node has exactly two out edges. Then **H**=2, **L**=0 and $*$ could be any value in $\{0, 1, 2\}$ which we annotate as $[0, 2]$. For region R_1 of the data (Figure 4.3), we are looking for all parameter graph nodes with a Morse graph containing fixed points of the form FP{$*$, H, L, L}, which are FP{0, 2, 0, 0}, FP{1, 2, 0, 0}, and FP{2, 2, 0, 0}. When imposing strict phenotypes, each fixed point is required to be monostable, so that the set of strict phenotypes associated to FP{$*$, H, L, L} must have exactly one of the following fixed points and no other stable Morse nodes:

FP{0,2,0,0}   FP{1,2,0,0}   FP{2,2,0,0}

In the first step of the analysis we construct the phenotype graph using strict phenotypes and its condensation, the coarsened strict phenotype graph. In the second step, we will impose monotone control variable modeling of the maternal gradients on *hb* and *kni*, as they are the only trunk gap genes that are affected exclusively by either *bcd* or *cad*, and not both, as *gt* and *kr* are. This is necessary because when both maternal gradients are targeting the same variable the control variable consistency condition is not met (see Definition 4.8). In other words, *bcd* and *cad* are both activators but they have opposing gradients and so the impact on the dually regulated nodes is not clear.

In this way, we construct two separate coarsened monotone strict phenotype graphs, one where we model the effect of *bcd* on *hb* and another where we model the effect of *cad* on *kni*. Thus, we will analyze four different graphs for each network. As shorthand, we will use the following notation for the Fullconn network graphs:

1. $PhG_F = (\mathcal{V}_F, \mathcal{E}_F)$ is the strict phenotype graph;
2. $CPhG_F = (V_F, E_F)$ is the coarsened strict phenotype graph;
3. $CPhG_F^{Hb} = (V_F^{Hb}, E_F^{Hb})$ is the coarsened (hb) monotone strict phenotype graph;
4. $CPhG_F^{Kni} = (V_F^{Kni}, E_F^{Kni})$ is the coarsened (kni) monotone strict phenotype graph;

We will perform the same steps for the StrongEdges network graphs:

1. $PhG_S = (\mathcal{V}_S, \mathcal{E}_S)$ is the strict phenotype graph;
2. $CPhG_S = (V_S, E_S)$ is the coarsened strict phenotype graph;
3. $CPhG_S^{Hb} = (V_S^{Hb}, E_S^{Hb})$ is the coarsened (hb) monotone strict phenotype graph;
4. $CPhG_S^{Kni} = (V_S^{Kni}, E_S^{Kni})$ is the coarsened (kni) monotone strict phenotype graph.

Fully connected network (Fullconn)

The parameter graph of the Fullconn network has 2.56 million nodes while the strict phenotype graph PhG_F has 694,476 nodes. Therefore over 25% of the parameter graphs nodes participate in at least one developmental path. How many of these developmental paths match the phenotype pattern? To address this question we consider the coarsened strict phenotype graph. Interestingly, this graph has exactly eight nodes and a single phenotype cluster path. This shows that all the parameter nodes that match $\Sigma(R_n)$ are connected and form one connected component for every $n = 1, \ldots, 8$. The connectedness of all nodes that match $\Sigma(R_n)$ can be interpreted as robustness of the phenotype pattern, in the sense that there are "wide" areas in parameter space for developmental paths to traverse. However, by Lemma 4.1, we only know that there is guaranteed to be one matching developmental path in the strict phenotype graph. This does not give us much idea about the robustness of the developmental program in the network model.

As indicated in the overview, in the second step we individually model the maternal gradients *bcd* and *cad*. To model *bcd* we require that the projection of a matching developmental path in the strict phenotype graph be a maximal monotone path in the factor graph of *hb* (see Definition 4.8). Recall that *bcd* is an activator of *hb* and that Bcd is decreasing along the A-P axis, thus a maximal monotone path must be a decreasing monotone path. Additionally, *cad* is an activator of *kni* but Cad is increasing along the A-P axis, thus a maximal monotone path in the factor graph of kni must be an increasing monotone path.

These constraints impact the number of strongly connected components, which we see in the coarsened (hb) monotone strict phenotype graph $CPhG_F^{Hb}$ with 137 strongly connected components and in the coarsened (kni) monotone strict phenotype graph $CPhG_F^{Kni}$ with 123 strongly connected components. For each of the graphs we present the number of nodes of the coarsened (hb) monotone strict phenotype graph that correspond to a single node of the coarsened strict phenotype graph and all of which match a particular label $\Sigma(\mathrm{R}_n)$. As can be seen in Table 6.1, for regions $n = 2, \ldots, 7$ there are between 20-30 nodes in the coarsened monotone strict phenotype graphs. Note that the effects of modeling the maternal gradient on *hb* and *kni* are not identical and that modeling of *bcd* on *hb* shatters the coarse structure more than *cad* on *kni* gradient. The increased number of connected components gives us an increased number of distinct phenotype cluster paths. By Lemma 4.1, this means we have increased the lower bound of the number of matching developmental paths in the strict phenotype graph, which allows us to better gauge the robustness of the network model. We will discuss the change in the number of paths later on, with the data summarized in Table 6.2.

Region	$\lvert\mathcal{V}_F\rvert_{R_n}$	$\lvert V_F\rvert_{R_n}$	$\lvert V_F^{Hb}\rvert_{R_n}$	$\lvert V_F^{Kni}\rvert_{R_n}$
R_1	91,296	1	1	1
R_2	82,323	1	23	23
R_3	91,296	1	23	9
R_4	82,323	1	34	9
R_5	91,296	1	23	34
R_6	82,323	1	23	23
R_7	91,296	1	9	23
R_8	82,323	1	1	1

Table 6.1: Number of nodes in different categories. Column labels indicate (from left to right) nodes of strict phenotype graph $PhG_F = (\mathcal{V}_F, \mathcal{E}_F)$, coarsened strict phenotype graph $CPhG_F = (V_F, E_F)$, coarsened *hb* monotone strict phenotype graph $CPhG_F^{Hb} = (V_F^{Hb}, E_F^{Hb})$ and coarsened *kni* monotone strict phenotype graph $CPhG_F^{Kni} = (V_F^{Kni}, E_F^{Kni})$. Row labels indicate in which region the phenotypes, $\Sigma(R_j)$, of the nodes lie.

Analysis of phenotype cluster paths

Recall that $CPhG_F$ has a single phenotype cluster path through it, thus we know by Lemma 4.1 that there is at least one matching developmental path in the phenotype pattern graph. However, we do not know if a matching developmental path, when projected onto the factor graphs of *hb* or *kni*, is a maximal monotone path; i.e., we want to compare the behavior of paths before control variable modeling with path behavior after control variable modeling is applied. Thus, to visualize the potential behavior of a matching developmental path, we create a heat map that allows us to see the worst case scenario of how matching developmental paths could behave in the factor graphs of each gene. Essentially, what we want to do is look at what factor graph layers (see Definition 4.2) for each gene are showing up in the phenotype cluster paths for each region in the phenotype pattern. However, since different strongly connected components don't necessarily have the same cardinality, and because there can be more than one strongly connected component being assessed per region in the phenotype pattern, then we need a way to normalize our results.

Definition 6.1. *Let $DC_F \subseteq U$ be the subset of $U = V_F, V_F^{Hb}$, or V_F^{Kni} for the coarsened (monotone) strict phenotype graphs $CPhG_F$, $CPhG_F^{Hb}$, $CPhG_F^{Kni}$, respectively, that belong to at least one phenotype cluster path. Let $U(R_n) \subsetneq U$ be the subset of nodes of the coarsened (monotone) strict phenotype graph that are associated to the phenotypes in $\Sigma(R_n)$. Let $C_{R_n} = DC_F \cap U(R_n)$ be the subset of nodes of $U(R_n)$ that also exist in at least one phenotype cluster path.*

Let

$$N_n = \bigcup_{u_i \in C_{R_n}} scc(u_i)$$

be the set of nodes from the phenotype graph $PhG_F = (\mathcal{V}_F, \mathcal{E}_F)$ that are in a strongly connected component associated to some $u_i \in C_{R_n}$. Let $\pi_i(p)$ for $p \in N_n$ be the projection of the parameter node p onto the factor graph F_i for the i-th regulatory network variable, v_i. The factor graph layers are denoted $X_{i,m}$. For each region R_n, define the collection of sets

$$S_m := \{p \in N_n \mid \pi_i(p) \in X_{i,m}\}.$$

Then, $|S_m|$ is the number of nodes in N_n with a projection onto F_i that lies in the factor graph layer $X_{i,m}$.

For example, $CPhG_F$ has a single phenotype cluster path $u_1 \to u_2 \to ... \to u_8$. Let $scc(u_i)$ be the associated strongly connected component in the phenotype pattern graph. Recall that because we only have eight strongly connected components, then for all $p \in scc(u_i)$, we have $MG(p) \in \Sigma(R_i)$. So $|DC_F| = |C_{\mathrm{R}_\mathrm{i}}| = 1$ for all $i = 1, ..., 8$. Additionally, there are 91,296 nodes of PhG_F with Morse graphs associated to region R_1, and all of these nodes are in the strongly connected component $scc(u_1)$, so $|N_1| = 91,296$. Of the nodes in N_1, $|S_8| = 4744$ of them are in the highest layer of the factor graph for *hb*, while $|S_4| = 18,610$ are precisely in the middle layer of the factor graph for *hb*. Additionally, $|N_8| = 82,323$, as we can see in Table 6.1. Of the nodes in N_8, $|S_0| = 34,612$ of them are in the lowest layer of the factor graph for *hb*.

In order to compare across regions as well as the different graphs, we consider the ratio $|S_m|/|N_n|$ for m indexing the layers of a factor graph and n indexing the regions of the A-P axis. In the previous examples, the ratios are $|S_8|/|N_1| = 0.05$ and $|S_0|/|N_8| = 0.42$. Doing this calculation for each gene and the corresponding factor graph layers, as well as for each strongly connected component in the phenotype cluster paths creates a heat map. Thus, this heat map gives a visual representation of how the matching developmental path *could* behave in its projection onto the factor graphs of each gene. For $CPhG_F$ and its single phenotype cluster path, the heat maps for each gene *hb* (yellow), *gt* (blue), *kr* (green) and *kni* (red) are shown in Figure 6.2. Each column represents the phenotype pattern layer, one for each region R_n and the rows represent the factor graph layers. The colorbars represent the ratios $|S_m|/|N_n|$ as described above and the ratios themselves are given in each square of the heat map. Recall that a matching developmental path must have a node $v_{i_k} \in \Sigma(R_k)$ for each $k \in 1, ..., 8$. As the matching developmental path goes from $k = 1$ to $k = 8$, the color indicates how likely it is that a matching developmental path passes through each factor graph layer.

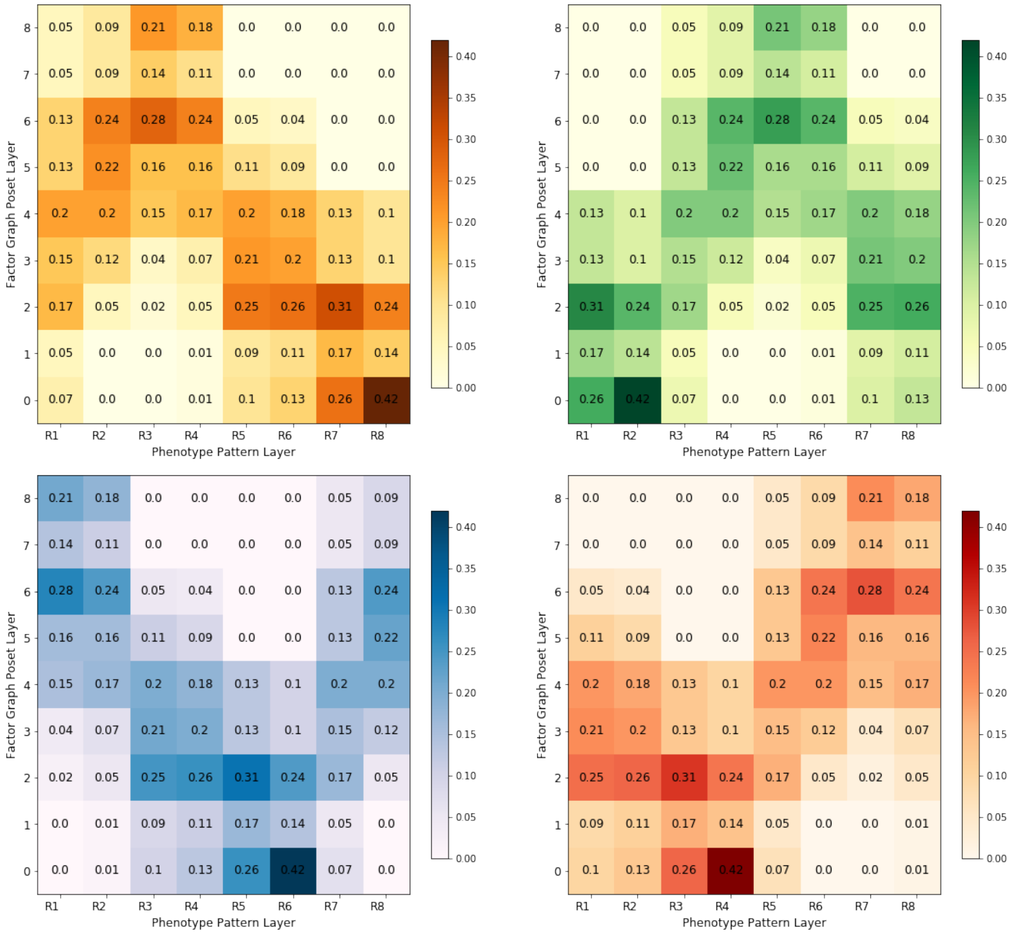


Figure 6.2: The Fullconn network strict phenotype graph heat maps: From top left, clockwise: *hb* (yellow), *kr* (green), *kni* (red) and *gt* (blue). Each column represents the phenotype pattern layers, one for each region R_n, and the rows represent the factor graph layers. Furthermore, the colorbars represent the ratios $|S_m|/|N_n|$ as described in the text. The exact ratios are shown in each square of the heat maps.

The heat maps coarsely reflect phenotype restrictions that we imposed on the set of developmental paths. For example, in region R_1 the phenotype pattern is FP$\{*, \text{H}, \text{L}, \text{L}\}$. The first component, Hb, is labeled as transitioning ($*$), meaning we put no restrictions on its Morse graph. This is reflected in the heat map, since each factor graph layer contains some parameter nodes, i.e., the first column of the yellow heat map in the upper left of Figure 6.2 has a nonzero ratio in every row. However, Gt is labeled high (H) in this region, meaning we did put restrictions on its Morse graph. Again, this seems to reflect what we see in the heat map, as there are no parameter nodes in the two lowest layers of the factor graph of gt, as can be seen in the lowest two rows in the first column of the blue heat map in the lower left of Figure 6.2.

However, now we want to know how modeling the impact of the maternal gradients on *hb* or *kni* will impact the heat map for all genes. Recall that we are concerned with maximal monotone paths in the factor graph when modeling maternal gradients as monotone control variables. This implies that when we model the impact of Bcd on Hb, in the heat map we should see Hb be in its highest factor graph layer in region R_1 of the phenotype pattern, and have a decreasing monotone behavior down to the lowest factor graph layer for region R_8. This is exactly what is being modeled when we construct the (*hb*) monotone strict phenotype graph. We are interested in seeing the impact of this constraint on the heat maps of the other three gene, *kni*, *gt*, and *kr*. We are interested in seeing if the implicit modeling of *bcd* alone can cause sharper localization of the phenotype cluster paths in the factor graphs of the other three gene products. This would allow us to conclude that the extra information about *bcd* above and beyond that which is implicitly present in the fixed point sequence is transmitted to the other gene products through the action of the network model. Similarly, when we model the impact of *cad* on *kni*, in the heat map we should see Kni be in its lowest factor graph layer in region R_1 of the phenotype pattern, and have an increasing monotone behavior to its highest factor graph layer for region R_8. We would again seek a sharpened localization in the factor graphs of the other three gene, *hb*, *gt*, and *kr*, due solely to extra information about the impact of *cad* on *kni*. Ideally, we could impose both control variables simultaneously, but this is currently computationally infeasible.

Once the maternal gradients are modeled on *hb* or *kni*, the number of phenotype cluster paths increases significantly. Even though each coarsened monotone strict phenotype graph is much smaller than the strict phenotype graph, we find that we must search for paths limited to one specific path length at a time to avoid memory problems. Thus our results are path-length dependent. We choose to start with the minimax path length of each factor graph (see Definition 4.1), which is the minimal path length required to achieve a maximal monotone path in the factor graph. This length depends on number of layers in the factor graph of either *hb* or *kni* depending on the gradient being modeled and on existence of phenotype cluster paths. For example, since the factor graph of *kni* in the Fullconn network has 9 layers, then by design a phenotype cluster path must have at least length 9. Thus, we initially restrict our search to paths of length 9. If there were none, then we increase the search to phenotype cluster paths of length 10. If we are able to find paths of length 10, we stop our search. If not, we increase the path length again by one node, until we are able to find phenotype cluster paths.

We found that $CPhG_F^{Hb}$ had 96 phenotype cluster paths of length 9, and $CPhG_F^{Kni}$ had 582 phenotype cluster paths of length 11 (see Table 6.2). Notice that this means that in each of the paths, there exists at least one region R_n whose phenotype is repeated. In other words, there are multiple parameter nodes representing different strongly connected components with phenotypes in $\Sigma(R_n)$.

Of the 127 nodes in $CPhG_F^{Hb}$, we found that only 29 ($\approx$ 23%) of them actually

$\lvert CPhG_F \rvert$	# of paths	max length	$\lvert CPhG_F^{Hb} \rvert$	# of paths	max length	$\lvert CPhG_F^{Kni} \rvert$	# of paths	max length
8	1	8	127	96	9	123	582	11

Table 6.2: Fullconn network: The number of nodes, number of paths, and their lengths in the coarsened strict phenotype graph $CPhG_F$, as well as coarsened monotone strict phenotype graphs $CPhG_F^{Hb}$ and $CPhG_F^{Hb}$.

participate in a phenotype cluster path. Similarly, of the 123 nodes in $CPhG_F^{Hb}$, only 30 ($\approx 24\%$) participate in a phenotype cluster path. While approximately the same percentage of nodes are participating in paths for both $CPhG_F^{Hb}$ and $CPhG_F^{Kni}$, the increased number of phenotype cluster paths in $CPhG_F^{Kni}$ is an artifact of the path length. The difference is that there are direct maximal monotone paths (see Definition 4.5) through the *hb* factor graph but not the *kni* factor graph.

For comparison, we also want to look at the percentage of nodes in the strict phenotype graph that could be in a matching developmental path. Thus, consider the following; the 29 nodes in the condensation $CPhG_F^{Hb}$ are associated to a total of 152,392 parameter nodes from the strict phenotype graph PhG_F. Recall that PhG_F has 694,476 parameter nodes, so approximately 22% of the nodes are in a phenotype cluster path and therefore potentially participate in a matching development path. Additionally, the 30 nodes in $CPhG_F^{Kni}$ are associated to a total of 174,063 nodes in PhG_F, so approximately 25% of parameter nodes in PhG_F potentially participate in a matching development path.

Region	$\lvert V_F^{Hb} \rvert_{R_n}$	in-path	A	$\lvert V_F^{Kni} \rvert_{R_n}$	in-path	A
R_1	1	1	1	1	1	1
R_2	23	5	0.22	23	2	0.09
R_3	23	5	0.22	9	2	0.22
R_4	34	9	0.27	9	1	0.11
R_5	23	3	0.13	34	10	0.29
R_6	23	3	0.13	23	11	0.48
R_7	9	2	0.22	23	12	0.52
R_8	1	1	1	1	1	1

Table 6.3: Fullconn network: Analysis of coarsened monotone strict phenotype graphs $CPhG_F^j = (V_F^j, E_F^j)$ for $j = Hb, Kni$. Numbers indicate how many nodes match phenotype label from $\Sigma(R_n), n = 1, \dots, 8$ indicated by the row label. The column in-path shows the number of nodes that participate in a strict phenotype path. Each column A is the ratio of column in-path over the column $\lvert V_F^j \rvert_{R_n}$.

Recall that we are not only interested in the effect of modeling the maternal gradients on the coarsened monotone strict phenotype graph structure, but also how it effects the heat map of the other genes. We discovered that modeling the maternal gradient *bcd* on *hb* had very little effect on the heat map of the other genes. Similarly, modeling the maternal gradient *cad* on *kni* had very little effect on the heat maps of the other three genes. Thus we conclude that the monotone control variable modeling of single maternal gradients does not impact our conclusions about the likelihood of maximal monotone paths in the factor graph. The heat maps for $CPhG_F$, $CPhG_F^{Hb}$ and $CPhG_F^{Kni}$ can be seen side by side for comparison in Figure A.2 from the appendix.

Strong edges network (StrongEdges)

The parameter graph of the StrongEdges network has 3.24 million nodes while the strict phenotype graph PhG_S has 983,144 nodes, which equates to approximately 30% of the parameter graph nodes participating in at least one developmental path. As we saw with the Fullconn network, the coarsened phenotype graph for the StrongEdges network graph has exactly eight nodes and a single phenotype cluster path. Thus, we know that there exists at least one matching developmental path in PhG_S by Lemma 4.1.

Region	$\lvert\mathcal{V}_S\rvert_{R_n}$	$\lvert V_S\rvert_{R_n}$	$\lvert V_S^{Hb}\rvert_{R_n}$	$\lvert V_S^{Kni}\rvert_{R_n}$
R$_1$	71,636	1	1	1
R$_2$	134,774	1	5	172
R$_3$	71,636	1	3	36
R$_4$	79,816	1	6	36
R$_5$	205,346	1	5	279
R$_6$	79,816	1	5	163
R$_7$	205,346	1	5	119
R$_8$	134,774	1	1	1

Table 6.4: StrongEdges network: Number of nodes in different categories. Column labels indicate (from left to right) nodes of strict phenotype graph $PhG_S = (\mathcal{V}_S, E_S)$, coarsened strict phenotype graph $CPhG_S = (V_S, E_S)$, coarsened *hb* monotone strict phenotype graph $CPhG_S^{Hb} = (V_S^{Hb}, E_S^{Hb})$, and coarsened *kni* monotone strict phenotype graph $CPhG_S^{Kni} = (V_S^{Kni}, E_S^{Kni})$. Row labels indicate in which region the phenotypes, $\Sigma(R_j)$, of the nodes lie.

Analysis of phenotype cluster paths

Again, to visualize the behavior of the phenotype cluster paths, we create heat maps to see the worst case scenario of how a matching developmental path could behave in the factor graphs of each gene. We again use the normalization factor S_m derived in Definition 6.1, although for the StrongEdges network coarsened (monotone) strict phenotype graphs instead of those for Fullconn. Recall that the heat maps give a visual representation of how the matching developmental paths *could* behave in their projections onto the factor graphs of each gene. For $CPhG_S$ and its phenotype cluster path, the heat maps for each gene *hb* (yellow), *gt* (blue), *kr* (green) and *kni* (red) are shown in Figure 6.3. As we saw in the Fullconn network, the heat maps coarsely reflect phenotype restrictions that we imposed on the set of developmental paths.

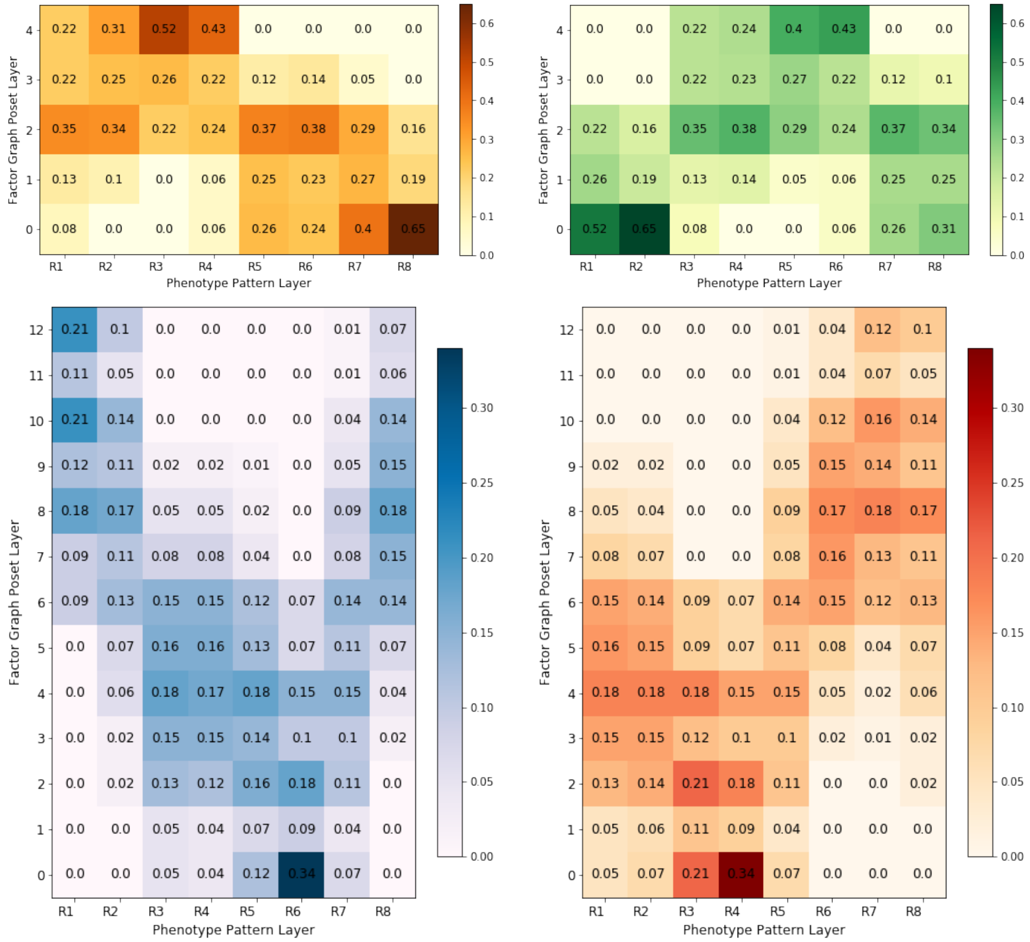


Figure 6.3: The StrongEdges network strict phenotype graph heat maps: Clockwise from top left: *hb* (yellow), *kr* (green), *kni* (red) and *gt* (blue). Each column represents the phenotype pattern layers, one for each region R_n, and the rows represent the factor graph layers. Furthermore, the colorbars are depicting the ratios $|S_m|/|N_n|$ as described in the text. The exact ratios are shown in the boxes in the heat maps.

The impact of modeling the maternal gradients *bcd* or *cad* degrades the strong connectedness of the coarsened strict phenotype graph, with $CPhG_S^{Hb}$ and $CPhG_S^{Kni}$ having 31 and 807 strongly connected components respectively (see Table 6.5). In this network, as seen in Table 6.4, modeling *cad* on *kni* shatters the coarse structure more than modeling *bcd* on *hb*. For example, notice that there are only five strongly connected components in $V_S^{Hb}|_{R_2}$, but 172 in $V_S^{Kni}|_{R_2}$. This means that modeling the gradient of *cad* on *kni* broke up the singular strongly connected component in $V_S|_{R_2}$ far more than modeling *bcd* on *hb*.

Actually, $CPhG_S^{Kni}$ is by far the largest coarsened (monotone) strict phenotype graph we have seen, and $CPhG_S^{Hb}$ is the smallest. The graphs $CPhG_S$ and $CPhG_S^{Hb}$ can be seen in Figure A.1 from the appendix.

We attribute this to the number of layers in the factor graphs for *hb* and *kni*, since we are requiring maximal monotone paths in the projection of phenotype cluster paths onto the factor graph of *hb* or *kni*. The Fullconn network is symmetrical in the number of in and out edges for each gene, thus the factor graphs for each gene all have nine layers. However, this is not the case in the StrongEdges network. Though the StrongEdges network has some symmetry, the number of in and out edges is not the same for *hb* and *kni*. This creates a difference in the number of layers for their factor graphs, five and 13 respectively. We found 34 phenotype cluster paths in $CPhG_S^{Hb}$ of length 8 and 358,370 phenotype cluster paths in $CPhG_S^{Kni}$ of length 14. Notice that neither coarsened monotone strict phenotype graph has a direct maximal monotone path (see Definition 4.5).

$\lvert CPhG_S\rvert$	# of paths	max length	$\lvert CPhG_S^{Hb}\rvert$	# of paths	max length	$\lvert CPhG_S^{Kni}\rvert$	# of paths	max length
8	1	8	31	34	8	807	358370	14

Table 6.5: StrongEdges network: The number of nodes, number of paths, and their lengths in the coarsened strict phenotype graph $CPhG_S$ and coarsened (*hb* or *kni*) monotone strict phenotype graphs $CPhG_S^{Hb}$ and $CPhG_S^{Hb}$.

Of the 31 nodes in $CPhG_S^{Hb}$, we found 19 ($\approx 61\%$) of them actually participate in a phenotype cluster path. Furthermore, of the 807 nodes in $CPhG_S^{Kni}$ we found 367 ($\approx 45\%$) of them participate in a phenotype cluster path. In this network, the difference in the number of phenotype cluster paths is an artifact of the size difference, we believe the important takeaway is that approximately 61% and approximately 45% of $CPhG_S^{Hb}$ and $CPhG_S^{Kni}$ respectively, are participating in the most direct phenotype cluster paths.

As we did with the Fullconn Network results, we compare the percentage of nodes in the phenotype graph that could be in a matching developmental path. The 19 nodes from $CPhG_S^{Hb}$ participating in a phenotype cluster path contain 642,449 nodes from the phenotype graph PhG_S. Recall that PhG_S has 983,144 nodes, so approximately 65% of the nodes potentially participate in a matching development path. The 367 nodes from $CPhG_S^{Kni}$ participating in a phenotype cluster path contain 295,616 nodes from the phenotype graph PhG_S. So approximately 30% of the nodes potentially participate in a matching development path.

As with the Fullconn network, we discovered that modeling the maternal gradient on *hb* had very little effect on the heat map of the other genes. Similarly, modeling the maternal gradient on *kni* had very little effect on the heat map of the other genes. The heat maps for $CPhG_S$, $CPhG_S^{Hb}$ and $CPhG_S^{Kni}$ can be seen side by side for comparison in Figure **??** from the appendix.

Region	$\vert V_S^{Hb}\vert_{Rn}$	in-path	A	$\vert V_S^{Kni}\vert_{Rn}$	in-path	A
R$_1$	1	1	1	1	1	1
R$_2$	5	2	0.4	172	6	0.03
R$_3$	3	3	1	36	5	0.14
R$_4$	6	4	0.67	36	4	0.11
R$_5$	5	3	0.6	279	137	0.49
R$_6$	5	3	0.6	163	116	0.71
R$_7$	5	2	0.4	119	97	0.82
R$_8$	1	1	1	1	1	1

Table 6.6: Analysis of coarsened monotone strict phenotype graphs $CPhG_S^j = (V_S^j, E_S^j)$ for $j = Hb, Kni$. Numbers indicate how many nodes match phenotype label from $\Sigma(R_n), n = 1, \ldots, 8$ indicated by the row label. The column in-path shows the number of nodes that participate in a strict phenotype path. Each column A is the ratio of column in-path over the column $|V_F^j|_{R_n}$.

DISCUSSION

Theory

As mentioned in section 2, DSGRN is a tool that has been developed in order to understand the global dynamics of gene regulatory networks, i.e., across all parameters. The realization that the global dynamics of a gene regulatory network can be represented finitely in terms state transition graphs is the main advantage to DSGRN. This realization naturally leads to a finite decomposition of phase space into regions where the state transition graphs are the same. DSGRN represents the parameter space in terms of a graph where nodes are the regions and edges between pair of regions represent boundaries between them. In this thesis we use the proximity information about parameter space to find paths in the with particular properties, that is, we find paths where each vertex is constrained to a particular Morse graph. While we initially apply this new theory and computational tool to paths that that match network dynamics for a spatially distributed developmental problem, this tool may have potential applications elsewhere.

Computational tool

The principal result of this research has been in extending DSGRN to study how tissue-scale behavior arises from network behavior in individual cells. We used this extension of DSGRN to study cellular systems where each cell contains the same network structure but operates under a parameter regime that changes continuously from cell to cell. In other words, we have started developing a multi-scale version of DSGRN. This generalization of DSGRN has resulted in a computational tool with a number of low-level algorithms that can be combined to model many biological situations, not just a spatially linear arrangement of cells.

In particular, the algorithm developed for constructing the phenotype graph can be used in a more general setting then what we have done. It is capable of constructing any subgraph of the parameter graph with desired fixed point dynamics, though it is straightforward to extend this tool to allow for construction of the subgraph when the desired dynamics are any dynamics representable by a DSGRN Morse graph. Theoretically, the path finding algorithm can be used on any DSGRN parameter graph where the user is searching for a path of continuous parameter changes along a particular sequence of bifurcations. Though most networks scale poorly in DSGRN, the construction of the condensation of the phenotype graph has the ability to deal with the computational challenges associated to path finding directly in the parameter graph.

Applications

As discussed in the results section, the goal of our analysis was to answer a set of questions about *Drosophila* development. We now discuss our current interpretation of these results.

Can we find network models that are capable of reproducing the discretized data of the spatial gap gene profile?

We have shown that both the Fullconn and StrongEdges networks are capable of reproducing the discretized data. Therefore, we have shown that it is unnecessary to consider sequential network modules at different points along the A-P axis, as in Verd et al. [9]. Furthermore, we have shown that it is even unnecessary to choose weaker edges to construct the network. The network with the most strongly justified biological edges is a perfectly adequate model of the observed experimental data.

In our analysis, in the coarsened strict phenotype graphs for both the Fullconn network and StrongEdges network, we found a single phenotype cluster path. Thus, we know that there is at least one matching developmental path in the phenotype graphs by Lemma 4.1. When modeling maternal gradients as control variables, the single strongly connected component associated to each region R_n breaks into a number of smaller strongly connected components under the restrictions on the factor graphs of *hb* or *kni*, giving us a larger number of paths through the coarsened monotone strict phenotype graph compared to the non-monotone graph. Since each of these paths is associated to a matching development path by Lemma 4.1, we have a greater lower bound for the number of matching developmental paths in the phenotype graph. For Fullconn constrained in the *hb* factor graph, this bound is 96 paths and Fullconn constrained under *kni* gives 582 paths. There is no overlap between these paths because of the difference in path length, and therefore the new lower bound is 96 + 582. Similarly, the StrongEdges network has a new lower bound of 34 + 358,370 of matching developmental paths.

Before modeling the maternal gradients, we don't know directly how a matching developmental path behaves when projected onto each gene's factor graph. Interestingly, though, the heat maps for the Fullconn and StrongEdges networks seen in Figures 6.2 and 6.3 show us that a matching developmental path cannot behave too badly in any of the gene factor graphs. For example, consider the node *hb* in the Fullconn network and its heat map (yellow) in the upper left of Figure 6.2. Recall that Hb has high expressions in R_2 an R_3 or is transitioning otherwise and has low expression level in the regions R_5 to R_8 (see Figures 4.3 and 4.4). Notice that for phenotype pattern layers R_2 and R_3 in the heat map that a matching developmental path, when projected onto the *hb* factor graph, can never be in the lower two factor graph layers for phenotype pattern layers R_2 and R_3. Additionally, for phenotype pattern layers R_5 through R_8 in the heat map, a matching developmental path when projected onto the *hb* factor graph can never be in the upper two factor graph layers for R_5 and R_6, as well as the upper four factor graph layers for R_7 and R_8. Furthermore, in phenotype pattern layer R_8 of the heat map, 42% the matching developmental paths end in the layer 0 of the factor graph. Thus, this heat map is showing us that any matching developmental path is following the expression patterns for the regions of the A-P axis, though there is no expectation it does so monotonically in *hb* and *kni* factor graphs. We see this phenomenon in every heat map for the Fullconn and StrongEdges networks, including the non-monotonic variables *gt* and *kr*.

Now, when we modeled the maternal gradients on *hb* or *kni*, we found many phenotype cluster paths for all the coarsened monotone phenotype graphs. Recall that the number of phenotype cluster paths is a lower bound on the number of matching developmental paths. When we modeled *bcd* onto *hb* (or *cad* on *kni*), each of these matching developmental paths has monotonic behavior in the *hb* (*kni*) factor graph. Aside from *hb* (*kni*), we saw little change in the heat maps for the other genes. Thus, these heat maps are showing us that any matching developmental path is following the expression patterns for the regions of the A-P axis, and does so monotonically for *hb*.

All of this suggests that both the Fullconn and StrongEdges network models are capable of reproducing the discretized data of the spatial gap gene profile and important biological information.

Can we use the abundance of matching developmental paths and perhaps their structure to assess robustness of the developmental program within a specific network model? How does the imposition of the monotone control variable modeling change the number and structure of matching developmental paths?

The results for this question are still preliminary, but we discuss some interesting results and their potential interpretations, as well as ideas for future work.

Due to the different sizes of the coarsened graphs and factor graphs, comparing the number of phenotype cluster paths is challenging. Therefore, we compare how many of the nodes potentially participate in a matching developmental path within its own respective phenotype graph and parameter graph. Recall the following results, which we will call **scores**.

1. The comparison of sizes of nodes sets in graphs $CPhG_F$, $CPhG_F^{Hb}$ and $CPhG_F^{Kni}$ for the Fullconn network showed that 100%, 22% and 25% of the nodes in the phenotype graph could participate in a matching developmental path, respectively.

2. The comparison of sizes of nodes sets in graphs $CPhG_S$, $CPhG_S^{Hb}$ and $CPhG_S^{Kni}$ for the StrongEdges network showed that 100%, 65% and 30% of the nodes in the phenotype graph could participate in a matching developmental path, respectively.

We note that for both the Fullconn and StrongEdges coarsened strict phenotype graphs, there is a single strongly connected component of phenotype nodes matching $\Sigma(R_n)$. Therefore all the parameter nodes that match this description of the phenotypes are connected. This may be interpreted as indication of robustness of the matching developmental paths: if we perturb any parameter node to a neighboring parameter node anywhere along the path and this new node belongs to the phenotype graph, there is a matching developmental path through this new node as well.

As discussed earlier (section 7), the heat maps for graphs $CPhG_F$ (Figure 6.2) and $CPhG_S$ (Figure 6.3) show an interesting correlation between the gene expression level imposed at the particular phenotype pattern layer R_n and the layers of the corresponding factor graph. Since we interpret the effect of the external maternal gradients as parameter node position in the factor graph for this gene, this observation suggests that the phenotypic data on levels of the gene expression and position in the parameter factor graph are closely related. This is an exciting observation, since it indicates that we will be able to detect on the level of DSGRN the effect in the opposite direction: modeling of maternal gradients determines the level of expression of all genes of the network. We will pursue this line of inquiry in the near future.

We examine the role of maternal gradient on number of developmental paths. When we model *bcd* on *hb*, the unique strongly connected component of the phenotype graph that matches $\Sigma(R_n), n = 1, \ldots, 8$ decomposes (shatters) into several strongly connected components. The number of nodes in the phenotype graph that could participate in a matching developmental path decreases to 22% and 65% for $CPhG^{Hb}_F$ and $CPhG^{Hb}_S$ respectively. The fact that the StrongEdges network had higher scores across the board may be the first indication we have that the StrongEdges network is a more robust model of the gap gene network then the Fullconn network. We will continue this line of inquiry in future work.

Do different candidate networks give different predictions? Is one network better able to match the data than others?

Again, the results for this question are preliminary and require more analysis in the future. However, the observations so far show that both the Fullconn network and the StrongEdges networks are capable of replicating the data. This is significant because it shows that the spatial data can be replicated using a single network with different parameters along the A-P axis, rather than different networks for different regions of the A-P axis. This observation is our most important result. This matches biological expectation of each cell having the same network structure, since these cells are very closely related being in their 14th round of the division and therefore it is likely that an identical network operates in each cell.

Additionally, if we hypothesize that having higher scores (see Section 7) means that the network better replicates the data, then the StrongEdges network is the better candidate than the Fullconn network. Since the Fullconn network constructed by Verd et al [9] took several weak edges from the original gap gene network in Figure 3.3, over suspected stronger edges, our results indicate that these additions do not make Fullconn network a better model for this developmental process.

Future Work

We would like to continue the work we started in this thesis in order to test the hypothesis we made during the discussion. Does having a higher score (as mentioned in the subsection 7) imply robustness and therefore a better gap gene network model? To answer this question, we plan to construct an artificial network by starting with either Fullconn and StrongEdges and perturb signs, directions and endpoints of edges. We will then analyze such a network in the same way we did on the Fullconn and StrongEdges, and see how the scores are affected.

Additionally, we would like to continue our analysis of the StrongEdges and Fullconn networks. So far, we have required strict phenotypes and maximal monotone paths in the factor graphs. What effect does relaxing this strictness have on our results? Will we see more of our coarsened phenotype graphs participating in direct phenotype cluster paths? We could also impose more strictness on the phenotype pattern and place additional constraints when the gene expressions transition. Currently, this is represented by $*$. Notice from Figure 4.3, that in the regions where gene expression was labeled $*$, each gene is either increasing or decreasing. We will also be interested in searching for matching developmental paths that match the monotonicity of this behavior in each region marked $*$. For example, in region R_2, gt is labeled $*$ in our current phenotype pattern. However, it is transitioning from H in R_1 to L in R_3. In the Fullconn network, this is represented by fixed points FP{2, 0, 0, 0}, FP{2, 1, 0, 0} and FP{2, 2, 0, 0}, and we allow all edges in the phenotype graph between them. In the future we will restrict this by only allowing paths that are decreasing in gt, i.e., we can only have edges in the phenotype graph from FP{2, 2, 0, 0} to FP{2, 1, 0, 0} to FP{2, 0, 0, 0}.

We did not model any of the gene self loops from the original gap gene network seen in Figure 3.3. We have shown that it is possible to replicate the data without them, however, it has been shown experimentally that these self loops exists. What is the role of the self-loops in the network? Would adding the self loops on the genes increase robustness?

There is also more work to be done on implicitly modeling the maternal gradients. So far we have only been able to implicitly model one gradient at a time. When we modeled the maternal gradients on both *hb* and *kni* at the same time, it shattered the coarsened gradient (*hb* and *kni*) phenotype graphs into so many nodes that we were unable to find a full set of phenotype cluster paths. Thus, in order to model more than one gradient at a time we will need to develop a way to estimate the number of phenotype cluster paths. Our current idea is to do this using an algorithm for estimating the number of linear extensions of a graph. A different method to numerically estimate the number of paths large graphs is by sampling paths.

Finally, recall that during time classes T1-T8 for cycle 14A, it has been experimentally determined that the maternal gradients are maintaining gap gene expression, not driving it. This was the reason we decided to model their effect implicitly as an effect on parameters, rather than directly on expression levels of genes. We are curious if we recover the gap gene expression by first modeling the gradient information on the parameter graph, find paths that respect monotonicty of the maternal gradients, and then look at the collection of fixed points at the parameters along these paths. This is the inverse problem of the work we have done in this thesis. Doing this analysis could help us determine whether the maternal gradients are sufficient to determine the gap gene expression and thus robustly drive the developmental program of *Drosophila* at this stage.

REFERENCES CITED

[1] Bree Cummins, Tomas Gedeon, Shaun Harker, and Konstantin Mischaikow. Dsgrn: Examining the dynamics of families of logical models. *Frontiers in Physiology*, 9:549, 2018.

[2] 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.

[3] Tomáš Gedeon, Bree Cummins, Shaun Harker, and Konstantin Mischaikow. Identifying robust hysteresis in networks. *PLOS Computational Biology*, 14(4):1–23, 04 2018.

[4] Scott F. Gilbert. *Developmental Biology*. 6th edition, 2000.

[5] Johannes Jaeger. The gap gene network. *Cellular and Molecular Life Sciences*, 68(2):243–274, 01 2011.

[6] Manu, Svetlana Surkova, Alexander V. Spirov, Vitaly V. Gursky, Hilde Janssens, Ah-Ram Kim, Ovidiu Radulescu, Carlos E. Vanario-Alonso, David H. Sharp, Maria Samsonova, and John Reinitz. Canalization of gene expression and domain shifts in the drosophila blastoderm by dynamical attractors. *PLOS Computational Biology*, 5(3):1–15, 03 2009.

[7] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[8] Berta Verd, Erik Clark, Karl R. Wotton, Hilde Janssens, Eva Jiménez-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, 02 2018.

[9] Berta Verd, Nicholas Monk, and Johannes Jaeger. Modularity, criticality, and evolvability of a developmental gene regulatory network. *eLife*, 8(2):243–274, 2019.

APPENDICES

APPENDIX A

SUPPLEMENTAL FIGURES

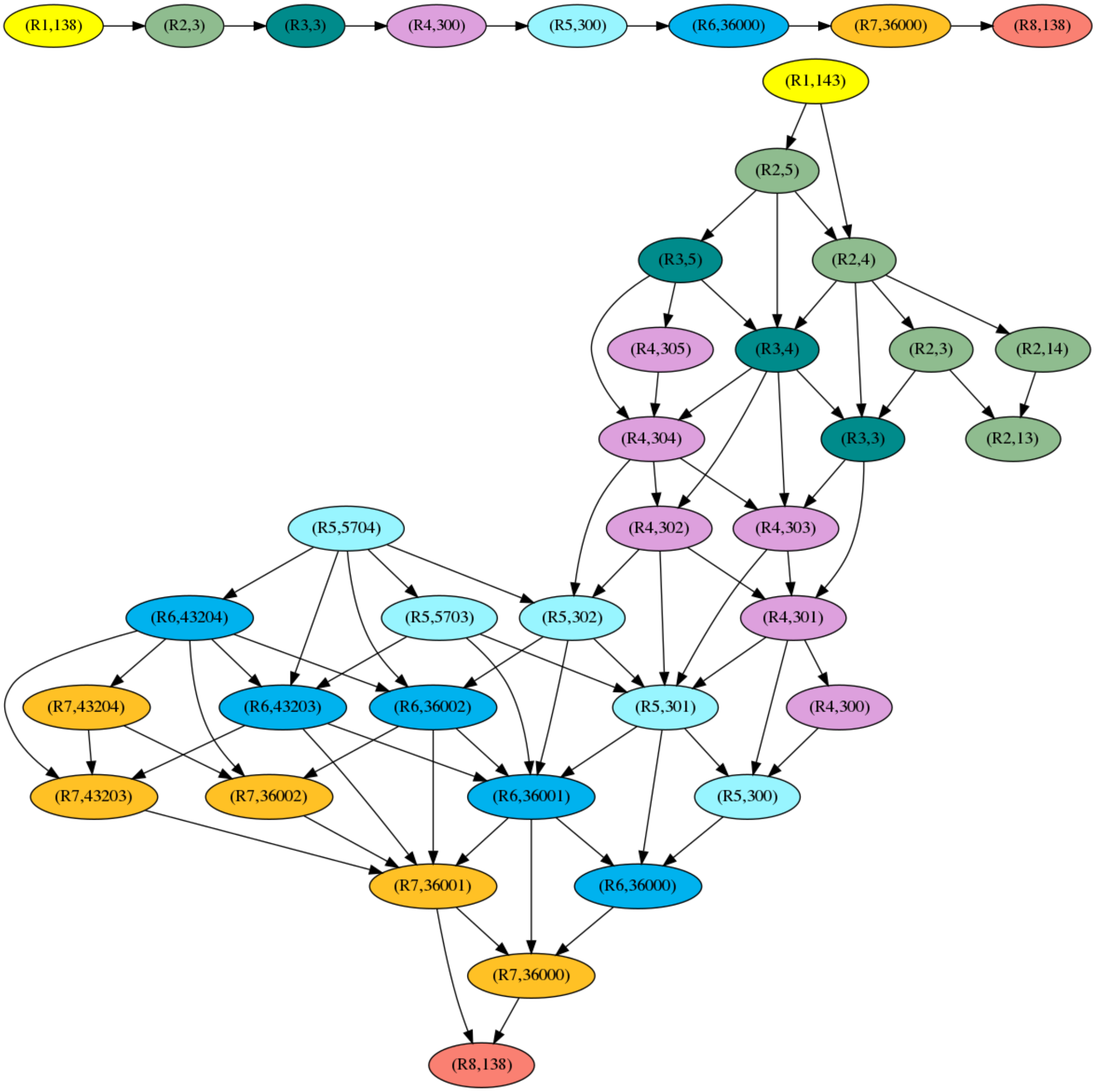


Figure A.1: Top: StrongEdges network coarsened strict phenotype graph, i.e., the condensation of the strict phenotype graph where no maternal gradients are implicitly modeled. Bottom: StrongEdges coarsened (hb) monotone strict phenotype graph, i.e., the condensation of the strict phenotype graph with bcd modeled on hb. For both graphs, each node v_i has a label $(R_n$, PI) where R_n is the region such that any $p_i \in scc(v_i)$ has $MG(v_i) \in \Sigma(R_n)$ and PI is the representative index of the strongly connected component $scc(v_i)$. There is a different color for each R_n for emphasis. Notice how modeling bcd on hb breaks up the strongly connected components of the strict phenotype graph. For example, R_4 (purple) in the coarsened phenotype graph (top) breaks up into six nodes in the coarsened (hb) monotone strict phenotype graph.

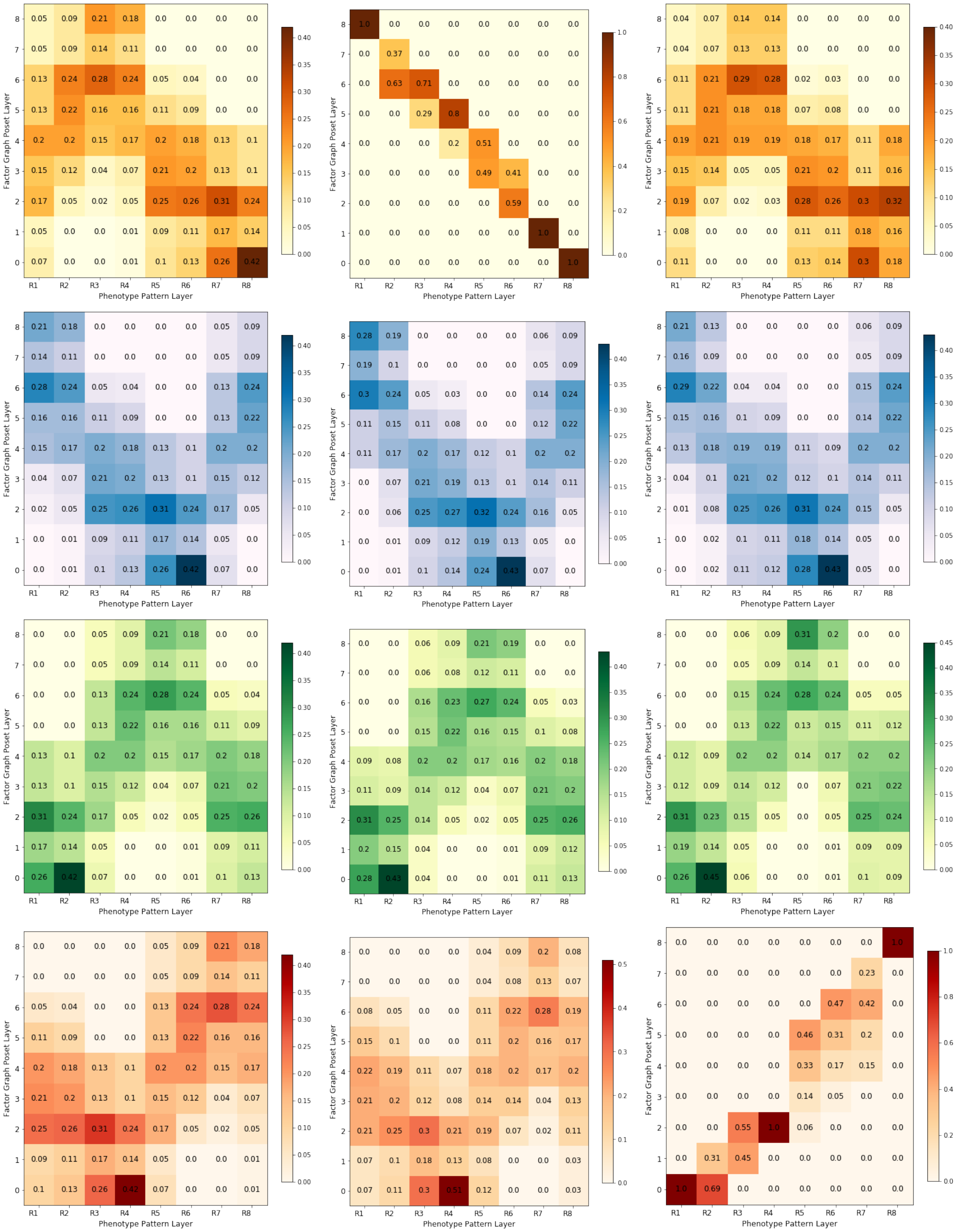


Figure A.2: Fullconn: Left column is the strict phenotype graph, middle is the monotone (*hb*) phenotype graph and right is the monotone (*kni*) phenotype graph. From top to bottom: *hb* (yellow), *gt* (blue), *kr* (green) and *kni* (red). A detailed explanation of the heat map construction can be found in the text in section 6.

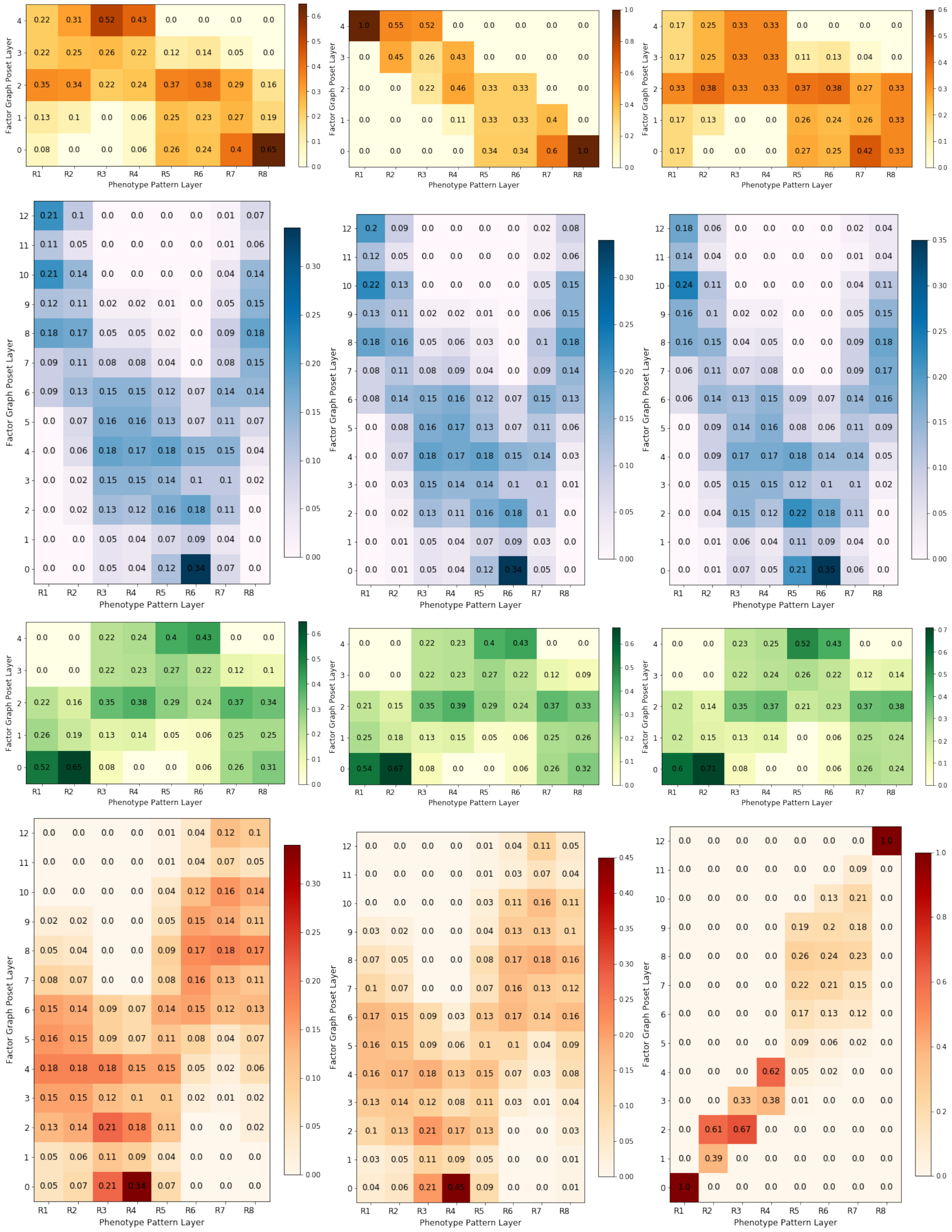


Figure A.3: StrongEdges: Left column is the strict phenotype graph, middle is the monotone (*hb*) phenotype graph and right is the monotone (*kni*) phenotype graph. From top to bottom: *hb* (yellow), *gt* (blue), *kr* (green) and *kni* (red). A detailed explanation of the heat map construction can be found in the text in section 6.