



Quantifying robustness of the gap gene network

Elizabeth Andreas*, Breschine Cummins, Tomáš Gedeon

Department of Mathematical Sciences, Montana State University, Bozeman, 59718, MT, USA

ARTICLE INFO

Dataset link: https://github.com/Eandreas1857/2023_GGN_Robustness

Keywords:

Dynamics
Regulatory network
Drosophila
Weighted directed graph
Finite directed paths

ABSTRACT

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

1. Introduction

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

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

anterior–posterior (A–P) axis, respectively (Jaeger et al., 2004; Jaeger, 2011).

This system has been modeled by several research groups (Verd et al., 2019; Manu et al., 2009; Perkins et al., 2006; Jaeger et al., 2004). To explain the experimental data, Verd et al. (2019) assume that there are different subnetworks, called ACDC dynamic modules, active in different regions along the A–P axis. They showed that each module could reproduce the data observed in each particular region at the end of the late stages of gap gene expression.

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

To answer these questions we set up an ambitious goal of constructing several robustness scores that we use to quantify the robustness

* Corresponding author.

E-mail address: eandreas1857@gmail.com (E. Andreas).

of a parameterized network fit to spatial experimental data that is computationally efficient enough to be evaluated over hundreds of networks. In order to accomplish this, we use the DSGRN (Dynamic Signatures Generated by Regulatory Networks) (Cummins et al., 2016; Gameiro, 2018) approach previously used for assessing network model fit with time series data and use it to model spatial data. For a regulatory network $\mathbf{RN}$, DSGRN constructs a parameter graph $PG(\mathbf{RN})$ which represents a finite decomposition of the parameter space for an ODE model of $\mathbf{RN}$, where paths in PG represent a continuous change of parameters in this ODE. For each parameter node $p \in PG(\mathbf{RN})$, DSGRN computes the summary of network dynamics from which one can extract a qualitative description of the stable equilibria of the system. To match a network model $\mathbf{RN}$ to spatial data, we seek paths in $PG(\mathbf{RN})$ along which the qualitative description of stable equilibria matches experimentally observed expression levels of gene products. If such a path exists, we say the network $\mathbf{RN}$ is capable of reproducing the data.

To address the robustness question, for each network $\mathbf{RN}$ we study the shape of the subgraph P of all such matching paths. We evaluate to what extent this subgraph has bottlenecks (indicating the fragility of development at some spatial position), we score how many paths can leave the subgraph P without completing the developmental program, we score how many paths can skip a segment, and we evaluate the overall size of P as a subgraph of the graph of all paths. We evaluate these scores on a class of nearly 1000 networks that have the same number of nodes (4) and edges (8) as two “canonical” network models. One of them is the network that is the union of the three ACDC submodules proposed by Verd et al. (2019). The second network is a subnetwork consisting of stronger regulatory interactions from the gap gene network derived by Verd et al. (2017) using work by Ashyraliyev et al. (2009).

The DSGRN approach to modeling network dynamics is an essential tool without which evaluating the global dynamics of hundreds of networks with 4 nodes and 8 edges would not be possible. However, even with this approach, the number of paths in $PG(\mathbf{RN})$ that has to be examined is astronomical. We develop graph constructions based on condensation graphs that allow the computation and handling of these large sets.

Our analysis produces evidence suggesting that previously explored network models and motifs tend to have higher robustness scores when compared to randomly generated networks, indicating consistency of our results with previous work (Verd et al., 2019). On the other hand, more local features such as the number of positive loops, number of negative loops, or number of negative edges does not seem to have a significant effect on robustness scores. Importantly, our work implies that particular features of network structure are capable of imparting robustness independent of the specific genes involved, which suggests that network structure itself may be subject to evolutionary pressure.

The organization of the paper is as follows. In Section 2, we provide enough background on *D. melanogaster* and DSGRN for the reader to obtain a solid understanding of modeling choices and the DSGRN parameter graph PG , respectively. The interpretation of certain paths in PG as spatial expression patterns is presented in Section 3. The background section on DSGRN also introduces Morse graphs, which are the “dynamic signatures” of DSGRN describing network behavior. These are used in Section 4 to provide a mechanism for matching DSGRN predictions to experimental data in *D. melanogaster* development. In Section 5, we introduce carefully constructed subgraphs of PG that incorporate information about spatial gradients of proteins important for proper segmentation of the *D. melanogaster* embryo. Particularly important is a subgraph called the path graph. In Section 6, we quantify features of the path graph that permit us to assess the robustness of *D. melanogaster* development in terms of the breadth and quality of the match between DSGRN predictions and experimental observations. In Section 7, we apply these scores to nearly 1000 networks to compare robustness across network topology. We conclude with a discussion in Section 8.

2. Background

2.1. *Drosophila melanogaster*

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

2.1.1. Anterior–posterior (A–P) patterning in *Drosophila melanogaster*

During *D. melanogaster* development, the embryo undergoes segmentation from the head (anterior region) to the tail (posterior region) of the embryo. We will refer to the linear array of segments as the A–P axis. The genes responsible for this segmentation were found experimentally by inducing genetic mutations and describing the resulting phenotypes (Nüsslein-Volhard and Wieschaus, 1980; Gilbert and Barresi, 2018). These experiments resulted in the discovery of a class of so-called gap genes whose knockouts cause entire regions of the A–P axis to be missing. We focus on the **trunk gap genes** (Jaeger, 2011) *hunchback* (*hb*), *giant* (*gt*), *Krüppel* (*Kr*) and *Knirps* (*kni*), which play a central role in the formation of the middling part of the A–P axis, namely between 35% and 75% egg length (Manu et al., 2009).

The trunk gap genes are, in part, regulated by maternal protein gradients, Bicoid (*Bcd*) and Caudal (*Cad*) in addition to Nanos (*Nos*) and existing maternal *Hb*, the gene product of *hb*. However, *Nos* and maternal *Hb* are less important to development than *Bcd* and *Cad* (Wang et al., 1994; Irish et al., 1989; Jaeger, 2011; Gilbert and Barresi, 2018), therefore we will limit our discussion of maternal gradients to *Bcd* and *Cad*.

The interaction of *Bcd* and *Cad* creates opposing gradients from anterior to posterior; *Bcd* has high concentration at the anterior region and smoothly decreases to a low concentration at the posterior, while *Cad* smoothly increases from anterior to posterior (Nüsslein-Volhard et al., 1987; Spirov et al., 2009; Gilbert and Barresi, 2018). These gradients give rise to **protein expression patterns** of the trunk gap gene proteins, which are regions along the A–P axis where each protein has high or low concentration (Nüsslein-Volhard et al., 1987; Nüsslein-Volhard and Wieschaus, 1980; Gilbert and Barresi, 2018). **Domain boundaries** for a particular protein are where the protein expression pattern is transitioning from high concentration to low or vice versa. Domain boundaries sharpen during late-stage development of the embryo, a process controlled by trunk gap genes rather than maternal gradients (Jaeger, 2011). Trunk gap gene regulation associated with the late-stage segmentation process can be described by four main **regulatory mechanisms**, as articulated by Jaeger (2011):

1. **Activation by maternal gradients:** *Bcd* and *Cad* maintain gap gene expression (Nüsslein-Volhard et al., 1987) as domain boundaries sharpen.
2. **Auto-activation:** Many early models of the gap gene network showed that auto-activation of each gene was essential (Meinhardt, 1986), 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 (Jaeger et al., 2004; Perkins et al., 2006). Experimentally, *hb* has the strongest evidence for auto-activation (Simpson-Brose et al., 1994; Jaeger et al., 2004; Perkins et al., 2006).
3. **Strong repressive feedback between complementary genes:** The strongest experimental evidence for late-stage trunk gap gene regulation is between the pair *hb* and *kni*, and the pair *kr* and *gt* (Jaeger et al., 2004). Both pairs exhibit mutual strong repression with each other, called **repressive feedback**.

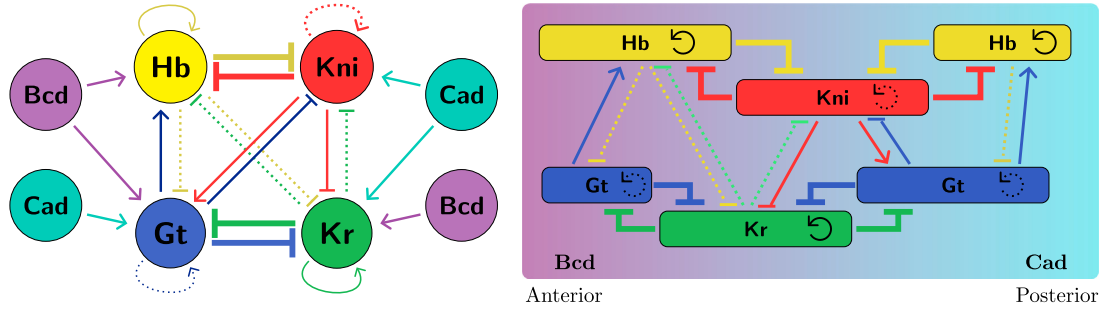


Fig. 1. Gap gene network. (left) The gap gene network used in Verd et al. (2019). The edge widths depict the strength of the interaction; dotted edges are the weakest interactions and the bold edges are the strongest. (right) Simplified spatial representation of gene regulation from anterior–posterior (A–P) position 35% to 75% for the gap gene network. The violet gradient indicates the concentration of Bcd, and the cyan gradient indicates the concentration of Cad. The horizontal extent of the boxes represents spatial positions with high late-stage protein expression levels.

Source: Figure adapted from (Verd et al., 2019).

4. *Regulation between non-complementary genes:* There is also experimental evidence that there are interactions between the other genes that are not complementary (Jaeger et al., 2004), though the exact type, strength, and potential effect of these interactions have only been examined by mathematical models (Jaeger, 2011).

2.1.2. The gap gene regulatory network

Though many models have been shown to faithfully replicate the protein expression of the trunk gap genes (Jaeger et al., 2004; Verd et al., 2019; Manu et al., 2009; Perkins et al., 2006), we will focus our study on the gap gene network as described in Verd et al. (2019) and shown in Fig. 1. Edges are color-coded according to their source protein. Dotted lines indicate weak regulatory interaction while bold lines indicate stronger regulatory interaction (Verd et al., 2019, 2017). We call these **weak edges** and **strong edges**, respectively. We will make the reasonable assumption that strong edges are more likely to be the dominating regulatory factors in the protein expression levels. The left panel in Fig. 1 is a regulatory network representation of the gap gene network while on the right is a spatial representation, which shows the extent of protein expression along the A–P axis.

We hypothesize that a sequence of parameter changes representing the impact of Bcd and Cad within a single network is capable of recapitulating the protein expression level data (see sections 5 and 3.2). There is a solid biological argument for choosing to model maternal gradients as a change in network parameters. In late-stage gap gene regulation, at any point along the A–P axis, the maternal gradients are relatively constant. That is, within a single cell there is not a significant change in the level of Bcd and Cad. Therefore Bcd and Cad can be viewed as part of the environmental conditions of the cell that help determine network parameters and not as active participants of the network.

2.1.3. ACDC dynamic modules of the gap gene network

During their study of the gap gene regulatory network in Fig. 1(left), Verd et al. (2019) partitioned a slightly reduced version of the spatial representation of the gap gene network shown in Fig. 1(right) into three subnetworks they described as dynamic modules. According to their definition, a dynamic module of the gap gene regulatory network is a subgroup of the genes that control protein expression in a region of the A–P axis. They postulate that the A–P axis can be split into three regions, each of which has a single gene that does not participate in network dynamics (i.e. is inactive) in that region. They assume that between A–P positions 35%–47% (region 1) *kni* is inactive, between 49%–59% (region 2) *gt* is inactive and between 61%–75% (region 3) *hb* is inactive. Thus, they create three dynamic modules (Fig. 2), all isomorphic to the ACDC signaling motif (Panovska-Griffiths et al.,

2013), that are active in regions 1, 2 and 3, respectively. Verd et al. (2019) showed that these subnetworks were capable of reproducing the protein expression levels between 35%–75% egg length.

One of the purposes of our study is to show that the decomposition into dynamic modules is unnecessary to reproduce the data. To do so, consider the network constructed as the union of nodes and edges from the three ACDC dynamical modules without the self-loops. We call this network the fully connected network (FullConn), see Fig. 2. We will evaluate this network using our methods to see if it can faithfully capture the protein expression data, which we describe in Section 4.2.

2.2. DSGRN

In this section, we discuss a modeling approach called DSGRN (Dynamic Signatures Generated by Regulatory Networks) (Cummins et al., 2016) that captures network dynamics across global parameter space. The structures that will be especially important are the **parameter graph** constructed from **factor graphs** (Section 2.2.2) and the **Morse graph** capturing the dynamics at each DSGRN parameter (Section 2.2.3).

2.2.1. Regulatory networks and switching systems

A **Regulatory Network** is a directed graph, denoted $\text{RN} = (V, E)$, where V is the set of nodes and the edges $E \subset V \times V \times \{1, -1\}$ denote interactions between the network nodes: the edge $(v_i, v_j, 1) \in E$ indicates that v_i is an **activator** of v_j (denoted by $v_i \rightarrow v_j$), while the edge $(v_i, v_j, -1) \in E$ (denoted $v_i \dashv v_j$), indicates that v_i is an **inhibitor** of v_j . An ordered pair $(v_i, v_j) \in E$ represents either $v_i \rightarrow v_j$ or $v_i \dashv v_j$.

Definition 2.2.1. Given a regulatory network $\text{RN} = (V, E)$, a **source** of a node v_j is a node v_i such that $(v_i, v_j) \in E$. A **target** of v_j is a node v_k such that $(v_j, v_k) \in E$. The set of sources and targets of a node v_j are given by

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

We associate to a regulatory network $\text{RN} = (V, E)$ with $|V| = M$ a system of M ordinary differential equations (ODEs) with piecewise constant nonlinearities called a **switching system** (Glass and Kauffman, 1972, 1973; Thomas, 1973, 1991; Thieffry and Thomas, 1998; de Jong et al., 2004; Snoussi, 1989). With a slight abuse of notation in the interest of clarity, we use v_j to denote either a node in V or the corresponding variable in a dynamical system that evolves according to

$$\dot{v}_j = -\gamma_j v_j + A_j(v), \quad j = 1, \dots, M \quad (1)$$

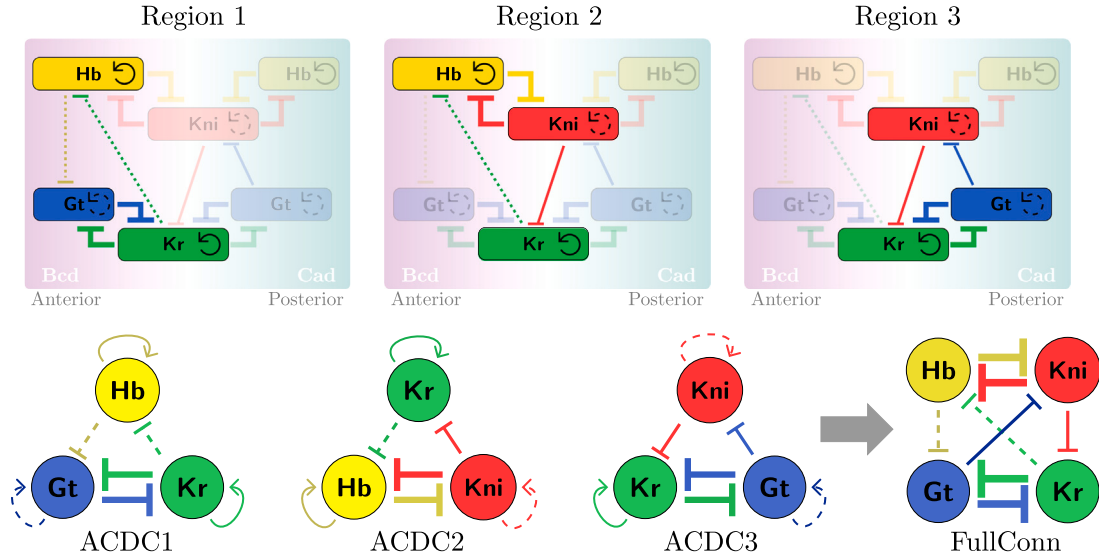


Fig. 2. ACDC modules of the gap gene network and the FullConn network. (top) Identification of active nodes along spatial domains 35%–47% (region 1), 49%–59% (region 2) and 61%–75% (region 3) identified in Verd et al. (2019). (bottom) The ACDC modules 1, 2, and 3, assumed by Verd et al. (2019) to be active in region 1, 2, and 3 respectively. The fully connected (FullConn) network is a union of nodes and edges from the ACDC modules (without the self-loops). Figure adapted from (Verd et al., 2019).

where $\gamma_j > 0$ is the decay rate of v_j and $\Lambda_j(v)$ is a product of sums of step functions $\sigma_{j,i}^\pm(v_i)$ for each $v_i \in S(v_j)$ given by

$$\sigma_{j,i}^+(v_i) = \begin{cases} l_{j,i} & \text{if } v_i < \theta_{j,i} \\ u_{j,i} & \text{if } v_i > \theta_{j,i} \end{cases} \quad (2)$$

if $v_i \rightarrow v_j$ and

$$\sigma_{j,i}^-(v_i) = \begin{cases} l_{j,i} & \text{if } v_i > \theta_{j,i} \\ u_{j,i} & \text{if } v_i < \theta_{j,i} \end{cases} \quad (3)$$

if $v_i \dashv v_j$. Here $l_{j,i}$ and $u_{j,i}$ are called the lower (low) and upper (high) level of effect of node v_i on node v_j , where $0 < l_{j,i} < u_{j,i}$. The threshold $0 < \theta_{j,i}$ for node v_i is where the effect on target v_j of the regulator node v_i changes. We assume that the values of $\theta_{j,i}$ for any node v_i are distinct. Suppose $|S(v_j)| = K$, where the nodes $v_{i_1}, \dots, v_{i_\ell}$ are activators of v_j and the nodes $v_{i_{\ell+1}}, \dots, v_{i_K}$ are inhibitors of v_j , then for the computations in this paper we choose the expression

$$\Lambda_j(v) = (\sigma_{j,i_1}^+(v_{i_1}) + \dots + \sigma_{j,i_\ell}^+(v_{i_\ell})) \sigma_{j,i_{\ell+1}}^-(v_{i_{\ell+1}}) \dots \sigma_{j,i_K}^-(v_{i_K}). \quad (4)$$

This form, often used in switching systems (Glass and Kauffman, 1972; Thomas, 1973; Thieffry and Thomas, 1998), was motivated by the fact that transcriptional activators often act additively, and that the transcriptional repressors physically block transcription initiation. This choice is not conceptually necessary, but is currently implemented in the DSGRN software (Gameiro, 2018). See Fig. 3(a, b) for an example of an RN and its associated switching system.

2.2.2. Factor graph and DSGRN parameter graph

The values $\{\theta_{j,i}, l_{j,i}, u_{j,i}\}$ are non-negative parameters of system (1), where we assume decay rates of 1 for simplicity. Traditionally, to characterize the behavior of the ODE system over parameter space, a (necessarily sparse) parameter sampling would be performed. DSGRN takes a different approach and divides parameter space into a finite number of regions defined by inequalities, and evaluates coarse but informative signatures of dynamic behaviors of the network that are invariant within each region (Cummins et al., 2016). Since the number of regions is finite, it is in principle possible to compute these coarse signatures over all of parameter space for a switching system associated to RN, although the number of regions grows combinatorially and exhaustive computations become rapidly intractable. In this section, we

introduce the inequalities that define DSGRN parameter regions and arrange them into a parameter graph that reflects region adjacency in the parameter space.

To do so, we define *order parameters* and *logic parameters*. For a node v with $|T(v)|$ thresholds, one for each $v_k \in T(v)$, an order parameter defines an ordering of these thresholds. A logic parameter defines how a finite collection of possible inputs to node v is related to the $|T(v)|$ thresholds of v .

Definition 2.2.2. Let $v_j \in V$ be a node in RN with source nodes $S(v_j) = \{v_{s_1}, \dots, v_{s_K}\}$ and target nodes $T(v_j) = \{v_{i_1}, \dots, v_{i_T}\}$. The thresholds associated with v_j are $\Theta_j = \{\theta_{i_1,j}, \dots, \theta_{i_T,j}\}$. An **order parameter** for v_j is a bijective map $\alpha_j : \Theta_j \rightarrow \{0, 1, \dots, |T(v_j)| - 1\}$ that induces a total ordering of the thresholds associated to v_j . We call O_j the set of all order parameters for vertex v_j .

Let

$$R_j = \{l_{j,s_1}, u_{j,s_1}\} \times \dots \times \{l_{j,s_K}, u_{j,s_K}\}$$

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

$$l_{j,s_k} < u_{j,s_k} \quad \text{for all } s_k \in S(v_j). \quad (5)$$

That is, we will write $w < \bar{w} \in R_j$ whenever, for $w = (a_1, \dots, a_K)$ and $\bar{w} = (\bar{a}_1, \dots, \bar{a}_K)$, we have $a_k \leq \bar{a}_k$ for all $k = 1, \dots, K$ and at least one of the inequalities is strict. Let $X_j = \{0, 1, \dots, |T(v_j)|\}$ be the set of $|T(v_j)| + 1$ integers that enumerates the intervals between the thresholds. A **logic parameter** ξ_j for v_j is a map $\xi_j : R_j \rightarrow X_j$, which satisfies

$$w < \bar{w} \Rightarrow \xi_j(w) \leq \xi_j(\bar{w});$$

i.e. it is monotone. We call L_j the set of all logic parameters for vertex v_j . A **factor parameter** for a node $v_j \in V$ is a pair $p_j := (\xi_j, \alpha_j) \in L_j \times O_j$.

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

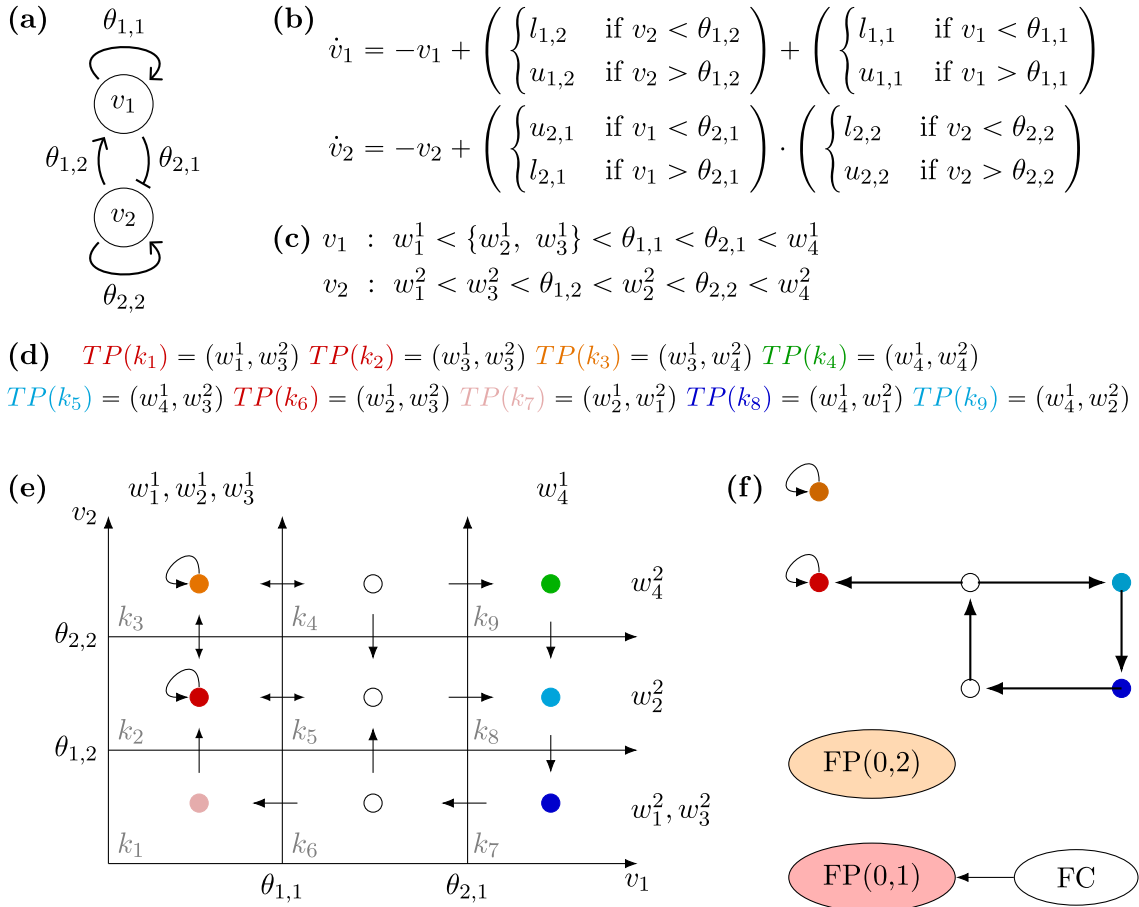


Fig. 3. Example of a DSGRN computation. (a) The RN and the (b) associated ordinary differential equations with decay rates $\gamma_1 = \gamma_2 = 1$. (c) A choice of DSGRN parameter where $w_1^i = (l_{i,1}, l_{i,2})$, $w_2^i = (l_{i,1}, u_{i,2})$, $w_3^i = (u_{i,1}, l_{i,2})$, and $w_4^i = (u_{i,1}, u_{i,2})$ for $i = 1, 2$, see Section 2.2.2 for more details. (d) The list of target points for each domain k_i of the phase space in (e). Note the 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). See Appendix A for details. (e) Phase space decomposition into the nine domains by thresholds $\theta_{1,1}$, $\theta_{1,2}$, $\theta_{2,1}$ and $\theta_{2,2}$. Each domain is represented by a circular vertex inside the domain. Arrows are the depiction of the direction of trajectories of a switching ODE system model in (b). The choice of DSGRN parameter is depicted above and to the right of phase space. For example, we have $w_4^1 > \theta_{2,1}$, since domains k_7 , k_8 and k_9 are all above $\theta_{2,1}$ we write w_4^1 above these. The vertices and arrows form the state transition graph. See Appendix A for details. (f) The Morse graph (below) associated with the state transition graph in (e) and the strongly path connected components, or Morse nodes, associated with each node of the Morse graph (above), where the strongly connected path components are associated to domains k_2, k_3, k_5, k_6, k_7 and k_8 in phase space and the edges are reachability conditions.

Table 1

Logic parameter examples (right) and corresponding inequality descriptions (left) for a network node with two in-edges and two out-edges and order parameter $\alpha_1(\theta_{1,1}) = 0$, $\alpha_1(\theta_{1,2}) = 1$.

Logic Parameter	Inequality Description
$\xi_1(w_1^1) = 0$, $\xi_1(w_2^1) = 0$, $\xi_1(w_3^1) = 0$, $\xi_1(w_4^1) = 0$	$w_1^1 < \{w_2^1, w_3^1\} < w_4^1 < \theta_{1,1} < \theta_{1,2}$
$\xi_1(w_1^1) = 0$, $\xi_1(w_2^1) = 0$, $\xi_1(w_3^1) = 1$, $\xi_1(w_4^1) = 2$	$w_1^1 < w_2^1 < \theta_{1,1} < w_3^1 < \theta_{1,2} < w_4^1$
$\xi_1(w_1^1) = 0$, $\xi_1(w_2^1) = 2$, $\xi_1(w_3^1) = 1$, $\xi_1(w_4^1) = 2$	$w_1^1 < \theta_{1,1} < w_3^1 < \theta_{1,2} < w_2^1 < w_4^1$

See Table 1 for example parameters for node v_1 in Fig. 3, which has two in-edges and two out-edges. The lattice of inputs

$$R_1 = \{l_{1,1}, u_{1,1}\} \times \{l_{1,2}, u_{1,2}\} = \{(l_{1,1}, l_{1,2}), (l_{1,1}, u_{1,2}), (u_{1,1}, l_{1,2}), (u_{1,1}, u_{1,2})\}$$

is partially ordered $(l_{1,1}, l_{1,2}) < \{(l_{1,1}, u_{1,2}), (u_{1,1}, l_{1,2})\} < (u_{1,1}, u_{1,2})$ with respect to the product order. The out-edges have thresholds $\theta_{1,1}$ and $\theta_{1,2}$ with the set of order parameters O_1 consisting of two functions

$$O_1 = \{(\alpha_1^1(\theta_{1,1}) = 0, \alpha_1^1(\theta_{1,2}) = 1), (\alpha_1^2(\theta_{1,2}) = 0, \alpha_1^2(\theta_{1,1}) = 1)\},$$

while the set of logic parameters is the set of functions $\xi_1 : R_1 \rightarrow \{0, 1, 2\}$. Select without loss one of the two order parameters $\alpha_1 := \alpha_1^1$ which we interpret as $\theta_{1,1} < \theta_{1,2}$. Using the notation $w_1^1 = (l_{1,1}, l_{1,2})$, $w_2^1 =$

$(l_{1,1}, u_{1,2})$, $w_3^1 = (u_{1,1}, l_{1,2})$, and $w_4^1 = (u_{1,1}, u_{1,2})$, we list in Table 1 three logic parameters with the corresponding description in terms of inequalities. While Table 1 only shows 3 logic parameters for our example, there are 20 in total. Hence, $|L_1 \times O_1| = 40$, showing v_1 has 40 factor parameters.

Remark 2.2.1. In the special case where $|T(v_j)| = 0$, i.e. node v_j has no out-edges, DSGRN assumes that v_j still can attain high and low levels of expression. To implement this, a “ghost” threshold is assigned, and the parameters for v_j are taken to be the same as if $|T(v_j)| = 1$.

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

Definition 2.2.3. Given a RN $= (V, E)$, the **factor graph** $F_j = (V_j, E_j)$ for a regulatory node $v_j \in V$ is an undirected graph with a node $p_j \in V_j$ for each factor parameter of v_j and edges between nodes whenever there is a single inequality change between two factor parameters; i.e., there is an edge between (ξ_j^k, α_j^k) and (ξ_j^l, α_j^l) if exactly one of the following is satisfied:

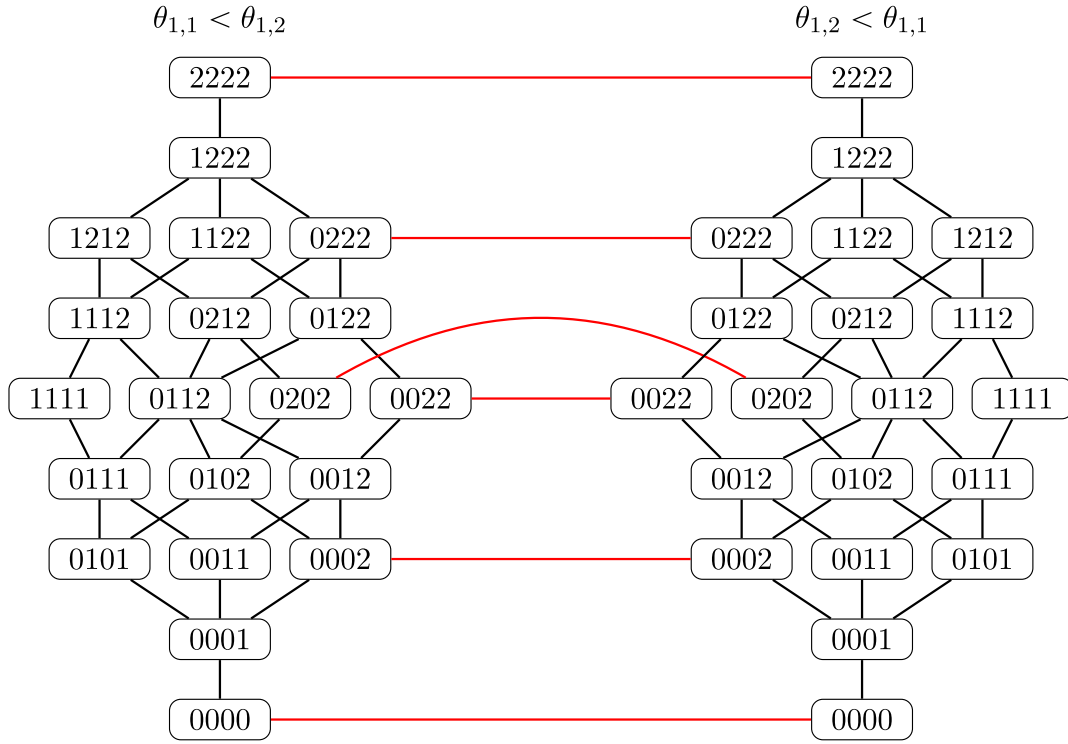


Fig. 4. Factor graph for a node v_1 from Fig. 3. Each node label $abcd$ represents a logic parameter, with $a = \xi_1(w_1)$, $b = \xi_1(w_2)$, $c = \xi_1(w_3)$ and $d = \xi_1(w_4)$. The left-hand side of the graph has factor parameters associated with the order parameter $\theta_{1,1} < \theta_{1,2}$ while the right-hand side is associated with $\theta_{1,2} < \theta_{1,1}$. Black edges show logical adjacencies and red edges show order adjacencies.

1. (Logical adjacency) $\xi_j^i(\omega) = \xi_j^k(\omega)$ for $\omega \in R_j$ except for exactly one $\tilde{\omega} \in R_j$, in which case

$$\xi_j^i(\tilde{\omega}) = \xi_j^k(\tilde{\omega}) + 1,$$

and $\alpha_j^i(\theta_{i_m,j}) = \alpha_j^k(\theta_{i_m,j})$ for all $\theta_{i_m,j} \in \Theta_j$, or

2. (Order adjacency) $\xi_j^i = \xi_j^k$ and there exists exactly one set of integers s, m, n such that $\alpha_j^i(\theta_{i_\ell,j}) = \alpha_j^k(\theta_{i_\ell,j})$ for $\ell \neq m, n$ and

$$\begin{aligned} \alpha_j^i(\theta_{i_m,j}) &= s, & \alpha_j^i(\theta_{i_n,j}) &= s + 1 \\ \alpha_j^k(\theta_{i_m,j}) &= s + 1, & \alpha_j^k(\theta_{i_n,j}) &= s. \end{aligned}$$

The factor graph of node v_1 from Fig. 3 can be seen in Fig. 4. Note that this graph has all 40 factor parameters, where the order parameter is depicted above the nodes and the logic parameter is encoded in the labels, where a label $abcd$ corresponds to the logic parameter with $\xi_1(w_1) = a$, $\xi_1(w_2) = b$, $\xi_1(w_3) = c$, and $\xi_1(w_4) = d$. Logical adjacencies are shown as black edges and order adjacencies are shown as red edges.

Order adjacencies exist only between **subfactor graphs**, or isomorphic subgraphs of a factor graph that contain only logical adjacencies. For a regulatory node $v_j \in V$, the order parameters $\alpha_j^i \in O_j$ are related by a group of permutations $\pi_{|T(v_j)|}$ that permute threshold labels. As a consequence, for each factor parameter $p_j \in V_j$ with a threshold order α_j^i there are $|T(v_j)|!$ parameters p_η , $\eta \in \pi_{|T(v_j)|}$, where threshold labels are permuted by η . Therefore, each factor graph contains a collection of $|T(v_j)|!$ subfactor graphs. An example of this can be seen in Fig. 4 factor graph, which has two subfactor graphs.

A **DSGRN parameter** is the choice of one factor parameter for each $v \in V$. Fig. 3(c) shows an example of a DSGRN parameter for the RN shown in (a).

Definition 2.2.4. Let $\text{RN} = (V, E)$ be a regulatory network with $|V| = M$ and let $F_j = (V_j, E_j)$ denote the factor graphs of each $v_j \in V$.

Then the **DSGRN parameter graph PG** has a vertex set $\mathcal{P}$ given by

$$\mathcal{P} = \prod_{j=1}^M V_j.$$

The nodes $(p_1, p_2, \dots, p_M)$ and $(q_1, q_2, \dots, q_M)$ in $\mathcal{P}$ are connected by an edge if and only if there exists exactly one $j = 1, 2, \dots, M$ where $(p_j, q_j) \in E_j$ and $p_i = q_i$ otherwise. In other words, there exists an adjacent change in inequalities in exactly one factor parameter graph.

Consider the RN from Fig. 3(a) and the factor graph in Fig. 4 for node v_1 . Since v_2 also has two in-edges and two-out-edges, then it has a factor graph that is isomorphic to the factor graph for v_1 . The parameter graph for this RN is constructed by taking the product of the factor graph for v_1 with the factor graph of v_2 . Then this parameter graph has a total of 1600 nodes (a fully constructed parameter graph for a two-node and two-edge network can be seen in Gedeon et al. (2018)). While 1600 appears to be a large number of parameter nodes, it represents a finite decomposition of $\mathbb{R}^{3|E|} = \mathbb{R}^{12}$ and therefore is a great reduction in number of parameters that need to be examined.

2.2.3. Morse graphs

A **Morse graph** is a compact description of the global dynamics of a regulatory network **RN** at a specific parameter node in **PG**. That is, for every node in **PG**, there is a Morse graph that captures both stable and unstable dynamics. The Morse graph is computed from a **state transition graph** (STG), an example of which is shown in Fig. 3(e). The partition of phase space shown is due to the thresholds associated with **RN**. The set of thresholds Θ_j for $v_j \in V$ divide the interval $[0, \infty)$ into $|T(v_j)| + 1$ intervals, namely $(0, \theta_{i_1,j}), (\theta_{i_1,j}, \theta_{i_2,j}), \dots, (\theta_{i_{|T(v_j)|},j}, \infty)$, described by the previously introduced set $X_j = \{0, 1, \dots, |T(v_j)|\}$. The collection of thresholds $\Theta = (\Theta_1, \dots, \Theta_M)$ ref:Theta divides $[0, \infty)^M$ into a finite number of M -dimensional rectangles called **domains**. Let $\mathcal{K}$ denote the collection of all such domains. This collection is represented

by the M -tuples of integers $X = \prod_{j=1}^M X_j$ via a bijection $\phi : \mathcal{K} \rightarrow X$. For example, domain k_1 in the lower left of Fig. 3(e) has label $\phi(k_1) = (0, 0)$, since both v_1 and v_2 exceed 0 thresholds each. The arrows between the domains represent the direction of trajectories of a switching ODE system model that is consistent with network structure. The details of the construction of the arrows are not needed for an understanding of this work and are summarized in Appendix A based on the introduction in Cummins et al. (2016). Once the division of phase space and the flow across domains is calculated, the resulting information can be simplified to the STG, shown by the nodes and edges superposed over phase space in Fig. 3(e). The most important features to notice are that (1) there can be single cells that are attracting, corresponding to the presence of a stable equilibrium, and (2) stable or unstable cyclic behavior can be identified. A Morse graph is a summary of this recurrent behavior in the STG, as described by the arrangement of strongly connected path components, see Fig. 3(f).

Definition 2.2.5. A directed graph $G = (V, E)$ is **strongly connected** if every pair of nodes $u, v \in V$ has a directed path from u to v and from v to u . A strongly connected subgraph H of G is said to be **maximal** if there is no strongly connected subgraph $H' \subsetneq G$ with $H \subsetneq H' \subsetneq G$. A maximal strongly connected subgraph is called a **strongly connected component**. A **strongly connected path component** is a strongly connected component that has at least one edge (Mischaikow, 2002).

We summarize the following definition from (Cummins et al., 2016). The **Morse decomposition** $\text{MD}(p)$ of a STG for $p \in \mathcal{P}$ is the set of all strongly connected path components of the STG. Consider any two strongly connected path components $s_1, s_2 \in \text{MD}(p)$. If there is a path in the STG from s_2 to s_1 then we say $s_1 \leq s_2$, defining a partial order $\leq$ on $\text{MD}(p)$. The **Morse graph** of the STG, denoted $\text{MG}(p)$, is the Hasse diagram of $(\text{MD}(p), \leq)$, and the vertices of $\text{MG}(p)$ are called **Morse nodes**.

In order for the Morse graph to provide interpretable information, we label each Morse node in a way that suggests the dynamics associated with the underlying strongly connected path component of the STG. The notation $\text{FP}(w)$ is used to label a Morse node where the corresponding strongly connected path component consists of a single attracting domain $k \in \mathcal{K}$ with label $w = \phi(k) \in X$. For example, in Fig. 3(e) we see that the domain k_3 is an attracting region with label $(0, 2)$, and thus the corresponding Morse node will be labeled $\text{FP}(0, 2)$.

The full cycle label (FC) annotates Morse nodes where there is a closed path $\{k^0, k^2, \dots, k^m, k^0\}$ in $\mathcal{K}$ where each edge $k^i \rightarrow k^{i+1} \pmod{m+1}$ follows the directed edge in STG and crosses a threshold for each node v_j . For example, the STG in Fig. 3(e) gives rise to a Morse graph with two Morse nodes labeled FP, as well as a full cycle FC that has a path to one of the fixed points (Fig. 3(f)). The full cycle represents the path in phase space $k_6 \rightarrow k_5 \rightarrow k_8 \rightarrow k_7 \rightarrow k_6$.

Definition 2.2.6. The leaves of the Morse graph, i.e. the Morse nodes with no out-edges, are called **stable Morse nodes**. All others are **unstable Morse nodes**. A **monostable Morse graph** is a Morse graph containing a single stable Morse node. A **monostable fixed point** is the unique stable Morse node in a monostable Morse graph that has an FP annotation.

3. Spatial modeling using DSGRN

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

3.1. Factor graph layers

In this section, we define a partial order on the factor graph by first defining it on every subfactor graph. Recall $\Theta_j = \{\theta_{i_1,j}, \theta_{i_2,j}, \dots, \theta_{i_{|T(v_j)|},j}\}$ is the collection of thresholds of node $v_j \in V$, and O_j is the set of all order parameters for v_j . Given a factor graph $F_j = (V_j, E_j)$, let $G_j^i = (V_j^i, E_j^i)$ be a subfactor graph of F_j that is associated with a particular order parameter α_j^i .

Note F_j has a set of **lowest parameter nodes** Lp_j

$$Lp_j := \{p_j = (\xi_j, \cdot) \in V_j \mid \xi_j(w) = 0 \text{ for all } w \in R_j\}$$

and a set of **highest parameter nodes** Hp_j

$$Hp_j := \{p_j = (\xi_j, \cdot) \in V_j \mid \xi_j(w) = |T(v_j)| \text{ for all } w \in R_j\}$$

with $(\xi_j, \cdot)$ and R_j as defined in Definition 2.2.2. For example, in the factor graph in Fig. 4, both nodes labeled 2222 are the set of highest parameters and both nodes labeled 0000 are the set of lowest parameters. Each subfactor graph G_j^i has a unique node $\ell_{j,i} \in Lp_j$ and unique node $h_{j,i} \in Hp_j$. We will call these nodes the **root** and **leaf** of a subfactor graph G_j^i , respectively.

Definition 3.1.1. Let $G_j^i = (V_j^i, E_j^i)$ be a subfactor graph. We define a strict partial order $<$ on V_j^i by $p_j^s < p_j^t$ when $\xi_j^s(w) \leq \xi_j^t(w)$ for all $w \in R_j$, with strict inequality for at least one $w \in R_j$.

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

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

Proof of Theorem 1. (a) We prove the statement in two steps. Assume first that $\xi_j^s(w) \leq \xi_j^t(w)$ for all $w \in R_j$, with strict inequality for exactly one $\bar{w} \in R_j$ and that $\xi_j^t(\bar{w}) = \xi_j^s(\bar{w}) + n$. If $n = 1$ then $(p_j^s, p_j^t) \in E_j^i$ by Definition 2.2.3. Assume now that $n > 1$. Since V_j^i contains all logic parameters then there exists $(p_j^s, p_j^1) \in E_j^i$ such that $\xi_j^1(\bar{w}) = \xi_j^s(\bar{w}) + 1$ and $\xi_j^1(w) = \xi_j^s(w)$ otherwise. Similarly, there exists $(p_j^1, p_j^2) \in E_j^i$ such that $\xi_j^2(\bar{w}) = \xi_j^1(\bar{w}) + 1$ and $\xi_j^2(w) = \xi_j^1(w)$ otherwise. Repeating this process n times, we construct a path

$$p_j^s \rightarrow p_j^1 \rightarrow p_j^2 \rightarrow \dots \rightarrow p_j^n$$

in G_j^i . Notice that $p_j^n = p_j^t$ since $\xi_j^n(w) = \xi_j^t(w)$ for all $w \in R_j$, proving that this is a path from p_j^s to p_j^t .

Assume now that there are $\omega_1, \dots, \omega_q \in R_j$ such that $p_j^s < p_j^t$ satisfies $\xi_j^s(\omega_u) = \xi_j^t(\omega_u) + n_u$ for $u = 1, \dots, q$, but $\xi_j^s(w) = \xi_j^t(w)$ for all other $w \in R_j$. We now sequentially apply the construction in step one by adjusting the values of ξ_j^s one ω_u at a time. The important restriction on this construction is that the ξ_j^s functions need to remain monotone functions throughout this process; i.e. $\omega_\alpha < \omega_\beta$ implies $\xi_j^s(\omega_\alpha) \leq \xi_j^s(\omega_\beta)$ as in Definition 2.2.2. It is easy to see that if $\omega_\alpha < \omega_\beta$ then increasing the values of $\xi_j^s(\omega_\beta)$ before adjusting values of $\xi_j^s(\omega_\alpha)$ will preserve the monotonicity of ξ_j^s at all stages of the construction. Therefore, we adjust the values of $\xi_j^s(\omega_u)$ starting from the highest ω_u and then proceed down the partial order. The concatenation of these paths gives a path in G_j^i between p_j^s and p_j^t . This proves (a).

To prove (b), suppose $(p_j^s, p_j^t) \in E_j^i$. Then by Definition 2.2.3 there exists exactly one $\bar{w} \in R_j$ such that $\xi_j^s(\bar{w}) = \xi_j^t(\bar{w}) \pm 1$, with equality for all other $w \in R_j$. If $\xi_j^s(\bar{w}) = \xi_j^t(\bar{w}) + 1$ then $p_j^s > p_j^t$ and if $\xi_j^s(\bar{w}) = \xi_j^t(\bar{w}) - 1$ then $p_j^s < p_j^t$.

The following Corollary is an immediate consequence of Theorem 1.

Corollary 3.1.1. Each subfactor graph G_j^i is connected.

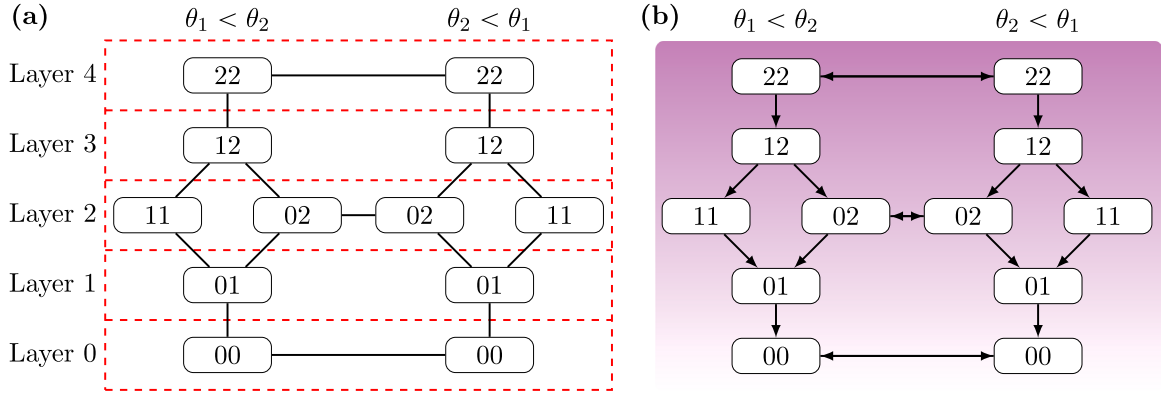


Fig. 5. Factor graph layers and modeling external variables. (a) The factor graph and factor graph layers for a **RN** node with one in-edge and two out-edges with thresholds θ_1 and θ_2 and input values $l < u$. The red dashed lines are depicting which nodes in the factor graph belong to each factor graph layer. (b) The factor graph from (a) with an activating external variable imposed, with directed edges depicting the direction of motion when $c'(y) = -1$ (violet). Notice that the external variable is inducing decreasing monotonicity through the factor graph.

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

A **rank function** (Klarnier, 1969) $\mathcal{L}$ is a map on a poset PO such that given $x, y \in PO$

- (i) $x < y$ implies $\mathcal{L}(x) < \mathcal{L}(y)$ and
- (ii) $\mathcal{L}(y) = \mathcal{L}(x) + 1$ if y covers x .

A poset PO is **graded** if it admits a rank function $\mathcal{L}$, and is denoted as $(PO, \mathcal{L})$ (Klarnier, 1969). A **chain** is a totally ordered subset of PO and a **maximal chain** is a chain that is not contained in a larger chain in PO (Stanley, 2012).

Recall that the subfactor graph $G_j^i = (V_j^i, E_j^i)$ has unique root $\ell_{j,i} \in Lp_j$ and unique leaf $h_{j,i} \in Hp_j$.

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

$$\mathcal{L}_j(p_j^q) = \sum_{\omega \in R_j} \xi_j^q(\omega).$$

Then

- (a) $(V_j^i, <)$ is a graded poset with rank function $\mathcal{L}_j$, and
- (b) $\mathcal{L}_j(\ell_{j,i}) = 0$ and $\mathcal{L}_j(h_{j,i}) = |R_j| \cdot |T(v_j)|$.

Proof. Since the subfactor graph $G_j^i = (V_j^i, E_j^i)$ has unique root $\ell_{j,i} \in Lp_j$ and unique leaf $h_{j,i} \in Hp_j$, to show (a) it is sufficient to prove that all maximal chains in $(V_j^i, <)$ have the same length (Stanley, 2012). We will show that this length is $|R_j| \cdot |T(v_j)|$.

Consider a maximal chain in $(V_j^i, <)$

$$p_j^1 < p_j^2 < \dots < p_j^n.$$

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

$$\xi_j^{q+1}(\omega) = \xi_j^q(\omega) + 1$$

for exactly one $\omega \in R_j$. Note that for any $\omega \in R_j$, since

$$\xi_j^1(\omega) = 0 \text{ and } \xi_j^n(\omega) = |T(v_j)|,$$

then there must be exactly $|T(v_j)|$ inequalities $p_j^q < p_j^{q+1}$ such that

$$\xi_j^q(\omega) = \xi_j^{q+1}(\omega) + 1.$$

Thus, each $\omega \in R_j$ requires $|T(v_j)|$ distinct inequalities in the maximal chain showing that the length must be $|R_j| \cdot |T(v_j)|$. Since we chose an arbitrary maximal chain, we have shown that all maximal chains in $(V_j^i, <)$ have the same length, proving it is a graded poset.

Now we show that $\mathcal{L}_j$ is a rank function on $(V_j^i, <)$. Let $p_j^s, p_j^t \in (V_j^i, <)$ and suppose that $p_j^s < p_j^t$, then by Definition 3.1.1,

$$\sum_{\omega \in R_j} \xi_j^s(\omega) < \sum_{\omega \in R_j} \xi_j^t(\omega).$$

Additionally, when $(p_j^s, p_j^t) \in E_j^i$ then p_j^t covers p_j^s which by Definition 2.2.3 implies $\mathcal{L}_j(p_j^t) = \mathcal{L}_j(p_j^s) + 1$. This proves part (a).

The proof of (b) follows directly from the definition of $\mathcal{L}_j$. ■

Since the subfactor graphs of a factor graph F_j are isomorphic, the rank function $\mathcal{L}_j$ is the same for all G_j^i . We will use this rank function to define layers of F_j .

Remark 3.1.1. There is a subtle difference between our definition of the parameter graph as the set of all pairs of order and logic parameters, and our definition of the function Λ_j in the switching ODE model in (4). Not every logic parameter can be realized by a function Λ and differences start for functions with 3 inputs (Crawford-Kahrl et al., 2022). The realizable parameters p are subset of all parameters P and these are encoded in the software DSGRN. We refer readers to Appendix B for the proof of Theorem 2 for realizable parameters.

Definition 3.1.2. Let $F_j = (V_j, E_j)$ be the factor graph for node v_j , with decomposition into subfactor graphs $G_j^i = (V_j^i, E_j^i)$ for $i \in \{1, \dots, |T(v_j)|\}$. The **factor graph layer** of $p_j \in V_j^i$ is $\mathcal{L}_j(p_j)$. The **k-th factor graph layer** of F_j is the node set

$$\{p_j \in V_j \mid \mathcal{L}_j(p_j) = k\},$$

for $k \in \{0, \dots, |R_j| \cdot |T(v_j)|\}$.

We say that the **highest factor graph layer** is the set Hp_j which is factor graph layer $|R_j| \cdot |T(v_j)|$. Likewise, the **lowest factor graph layer** is the set Lp_j , which is factor graph layer 0.

To illustrate the concept of factor graph layers, consider a network node v_j with one in-edge with $0 < l < u$ and two out-edges with thresholds θ_1 and θ_2 . There are 12 factor parameters in the factor graph, see Fig. 5(a). This factor graph has five layers. Each node label ab represents a logic parameter with $a = \xi_j(l)$ and $b = \xi_j(u)$. Since $\mathcal{L}_j = \sum_{\omega \in R_j} \xi_j(\omega)$ then the factor layer number is $a + b$. Another factor

graph example is in Fig. 4. This factor graph has 9 layers arranged horizontally and numbered by the sum of the labels $abcd$. The factor graph layers allow us to define an idea of monotonicity of paths through a factor graph.

Definition 3.1.3. Consider a path through the factor graph $F_j = (V_j, E_j)$, with sequence of nodes $p_1, p_2, \dots, p_m \in V_j$. The path is said to be **monotone increasing** if for all p_i, p_k in the path, we have $\mathcal{L}_j(p_i) \leq \mathcal{L}_j(p_k)$ if and only if $i \leq k$, i.e. factor graph layers increase along the path. Similarly, the path is said to be **monotone decreasing** if, for all p_i, p_k in the path, we have $\mathcal{L}_j(p_i) \leq \mathcal{L}_j(p_k)$ if and only if $i \geq k$. A monotone increasing or monotone decreasing path in F_j is called a **monotone path**.

3.2. Interpreting external variables as parameter changes

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

Definition 3.2.1. A **monotone external variable** $c : Y \rightarrow \mathbb{R}$, is an external variable not included in the network RN that satisfies either $c'(y) \geq 0$ (increasing) or $c'(y) \leq 0$ (decreasing) on Y .

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

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

We elaborate on the last point. Let $\sigma_j = \pm 1$, where $+1$ means that $c(y)$ is an activator and -1 means $c(y)$ is a repressor to the target node v_j . Let F_j be the factor graph of v_j . We model the effect of $c(y)$ on v_j as a monotone path over the layers of F_j : $c(y)$ induces monotone increasing paths in F_j when $\sigma_j \cdot \text{sign}(c'(y)) = +1$ and monotone decreasing paths when $\sigma_j \cdot \text{sign}(c'(y)) = -1$. This monotonicity condition on the factor graph of v_j is a model of the continuously changing abundance of v_j as a function of changing $c(y)$. In Fig. 5(b), an activating external variable that is monotone decreasing in y (violet) is imposed on the factor graph. This induces decreasing monotonicity on the factor graph shown as directed edges.

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

Definition 3.2.2. A **maximal monotone path** in the factor graph F_j is either

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

We now show how to apply this modeling framework to match observations along a spatial domain under external variable control using the example of the *D. melanogaster* gap gene network. In the next section, we show how biological observations may be translated into the

language of Morse graphs, and apply this translation to *D. melanogaster* development. In the subsequent section, we construct paths in the DSGRN parameter graph, making use of concepts developed in this section.

4. Expressing experimental data as morse graphs

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

4.1. Descriptive pattern and phenotype pattern graph

We formally describe a methodology for interpreting any spatial data in a DSGRN framework. For a given network model $\text{RN} = (V, E)$, we consider paths $p_1 \rightarrow \dots \rightarrow p_k$ in the parameter graph PG and the corresponding sequences of fixed point Morse sets $\text{FP}_1, \dots, \text{FP}_k$ as the output of the network model. Since the number of out-edges of $v_j \in V$ determines the highest integer state of X_j (see Definition 2.2.2), the highest value of an FP annotation will vary across network topologies. This complicates the comparison of network models to each other and to the data. Therefore, in order to match experimental data to the model output in X_j , we first transform experimental data to qualitative data, using the descriptors “high”, “intermediate” and “low”. Then for each network under consideration, we transform this qualitative data to integer values in X_j .

Definition 4.1.1. Consider a finite set $\mathcal{L}$ of qualitative expression level labels, for example $\mathcal{L} = \{L, H\}$ for “low” and “high”, that admits a (not necessarily strict) total order, such as $L < H$. Further, consider a spatial data set of M genes and N spatial locations. Then an $N \times M$ matrix D with $D_{i,j} \in \mathcal{L}$ is the **descriptive pattern** of the spatial data.

We desire to match a DSGRN model of a regulatory network with M vertices $V = \{v_1, \dots, v_M\}$ to the descriptive pattern at spatial locations $\{1, \dots, N\}$. In order to perform this matching, we map the n th row of the descriptive pattern D (denoted $D_{n,\cdot}$) onto a collection of DSGRN fixed points (FPs), whose annotations match $D_{n,\cdot}$. We then organize this data into a **phenotype pattern graph**, that is, a DSGRN representation of the observed data. We will say that there is a *match between the data and the DSGRN model* if there is a path in the phenotype pattern graph $(m_1, \dots, m_k)$ and a path in the DSGRN parameter graph $(p_1, \dots, p_k)$ such that at every position i , there is at least one Morse node $\text{FP} \in \text{MG}(p_i)$ such that $\text{FP} = m_i$.

Definition 4.1.2. A **pattern label** $\rho = (\rho_1, \dots, \rho_M)$, where $\rho_j \in X_j$ (see Definition 2.2.2), is a collection of integer states, one for each variable $v_1, \dots, v_M$. Let D be the descriptive pattern for a spatial data set of M variables and N spatial locations. We say a pattern label ρ is **consistent with** $D_{n,\cdot}$ if $D_{n,j} < D_{n,k}$ implies $\rho_j < \rho_k$. The set of pattern labels associated with spatial location n is

$$\rho(n) = \{\rho \mid \rho \text{ is consistent with } D_{n,\cdot}\}.$$

The **phenotype pattern** of D is

$$\mathcal{P} = (\rho(1), \rho(2), \dots, \rho(N)).$$

A parameter node $p \in PG$ has a **relevant phenotype** if there is an $n \in \{1, \dots, N\}$ and a pattern label $\rho \in \rho(n)$ such that there exists a Morse node $\text{FP}(\rho) = \text{FP}((\rho_1, \dots, \rho_M)) \in \text{MG}(p)$. If $\text{MG}(p)$ is monostable (see Definition 2.2.6), then we say ρ is a **strict phenotype** of p . Lastly, we use the notation $\rho(\text{MG}(p))$ to denote the pattern label of a monostable fixed point Morse node in $\text{MG}(p)$.

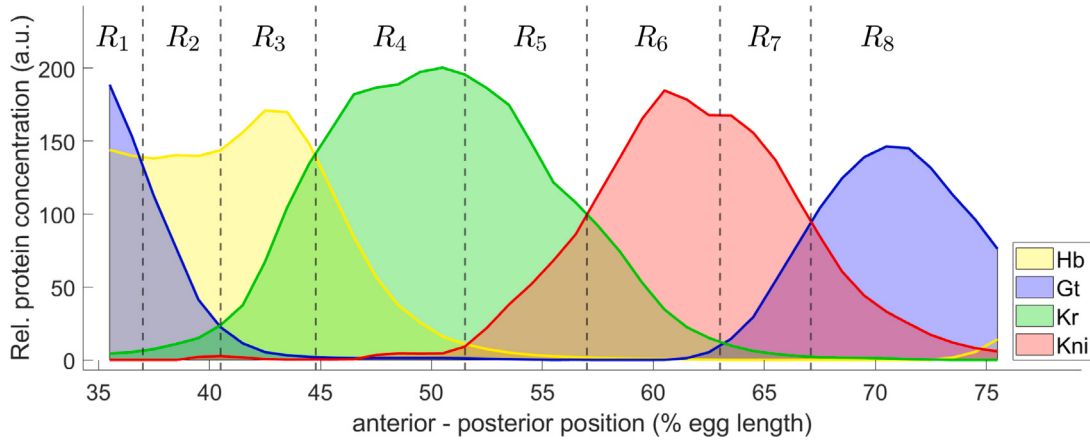


Fig. 6. Data of protein concentration along the anterior–posterior position % egg length for the trunk gap gene proteins Hunchback (Hb) in yellow (peak in R_3), Giant (Gt) in blue (peak in R_1 , R_8), Krüppel (Kr) in green (peak in R_4) and Knirps (Kni) in red (peak in R_6). The gap gene protein expression pattern data (S1_Data.ods) was obtained from supplementary information in Verd et al. (2018).

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

Definition 4.1.3. For a network $RN = (V, E)$ with $|V| = M$ and two vectors $c, d \in X = \prod_{v \in V} X_v$ a **set of transition vectors between c and d** is

$$\mathcal{T}_{c,d} := \{a \in X \mid a_i \in I_i \text{ for } i = 1, \dots, M\}$$

where $I_i = [c_i, d_i]$ if $c_i \leq d_i$ and $I_i = [d_i, c_i]$ if $c_i \geq d_i$ is the interval of integers between c_i and d_i . Define

$$\mathcal{T}(n) = \bigcup \{\mathcal{T}_{c,d} \mid c \in \rho(n) \text{ and } d \in \rho(n+1)\}$$

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

$$\Sigma = \bigcup_{n=1}^{N-1} \mathcal{T}(n).$$

Definition 4.1.4. A **phenotype pattern graph** for D is a directed graph $PPG = (\Sigma, E_\Sigma)$, where $(\rho, \rho') \in E_\Sigma$ if $\rho = \rho'$ or the following are simultaneously satisfied

- $\rho \in \rho(n)$ and $\rho' \in \mathcal{T}(n)$; and
- the paths are strictly monotone in the descriptive pattern, i.e., $\rho_j > \rho'_j$ implies $D_{n,j} > D_{n+1,j}$ and $\rho_j < \rho'_j$ implies $D_{n,j} < D_{n+1,j}$.

4.2. *D. melanogaster* example

As an example, we describe the construction of a descriptive pattern for the *D. melanogaster* data. Fig. 6 shows the protein concentration data of the trunk gap genes along the A–P axis of the embryo. These data are taken late in the segmentation process when protein concentrations have equilibrated to a fixed distribution across the A–P axis. We therefore assume that these concentrations correspond to steady state values of the segmentation dynamics.

At most positions along the A–P axis, the protein expression levels of the four genes are ordered, with the expression of two genes having very low protein concentration. Furthermore, there are sections where this ordering does not change. For example, at every point between positions 40% and 45% egg length the protein expression levels are

Table 2

Descriptive pattern for the *D. melanogaster* protein expression pattern data seen in Fig. 6.

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	*

ordered, from highest to lowest, Hb, Kr, Gt, Kni. Using these observations we divide the A–P axis into eight regions R_n (see the dashed lines in Fig. 6), where the protein expression levels are consistently ordered. Region boundaries are at crossings between two protein concentrations.

We discretize the experimental values in each of the 8 regions by one of the descriptive labels $\mathcal{L} = \{H, *, L\}$:

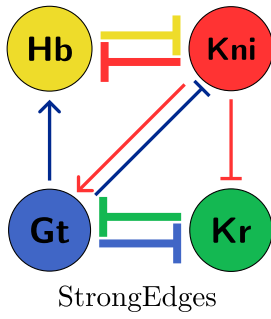
1. H : Protein expression level is high,
2. L : Protein expression level is low,
3. * : Protein expression level is indeterminate.

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

Protein expression levels in Fig. 6 show that *kni* is inactive between A–P positions 35%–47%, *gt* is inactive between 49%–59%, and *hb* is inactive between 61%–75%, which is consistent with the interpretation of Verd et al. (2019). Thus, in these regions, these protein expression levels will be labeled L. We further assign label H to each gene whose protein expression level is highest in a given region in Fig. 6. Therefore in each region, we will have one gene labeled H, two genes labeled L and the remaining gene with intermediate protein expression will be assigned *. We arrive at the descriptive pattern seen in Table 2.

We now transform the descriptive pattern in Table 2 into a phenotype pattern graph. We use the following FP assignment for a pattern label $\rho = (\rho_{Hb}, \rho_{Gt}, \rho_{Kr}, \rho_{Kni})$ at spatial position $n = 1, \dots, 8$. For every gene v_j with $j \in \{Hb, Gt, Kr, Kni\}$, we assign

- $\rho_j = 0$ if $D_{n,j} = L$,
- $\rho_j = \max\{1, |T(v_j)|\}$ if $D_{n,j} = H$,
- if $|T(v_j)| > 0$ and $D_{n,j} = *$, then $\rho_j \in \{1, \dots, |T(v_j)| - 1\}$, and
- if $|T(v_j)| = 0$ and $D_{n,j} = *$, then $\rho_j \in \{0, 1\}$.



Reg.	A-P	Hb	Gt	Kr	Kni
R_1	35-37	$\{0,1\}$	3	0	0
R_2	37-40	1	$\{1,2\}$	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	$\{1,2\}$
R_6	57-63	0	0	$\{0,1\}$	3
R_7	63-67	0	$\{1,2\}$	0	3
R_8	67-75	0	3	0	$\{1,2\}$

Fig. 7. StrongEdges network (left) and phenotype pattern (right).

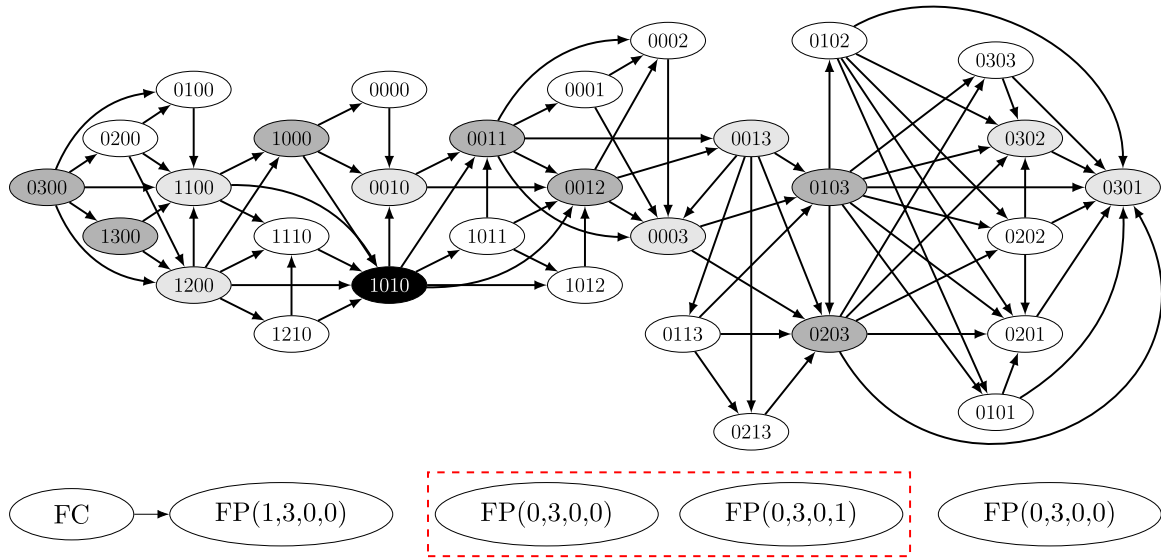


Fig. 8. StrongEdges phenotype pattern graph. (Top) StrongEdges phenotype pattern graph, where labels $abcd$ correspond to pattern label (a,b,c,d) . Note each node has a self-loop which is omitted for clarity. Nodes are shaded if the pattern label is in $\rho(n)$ for some region R_n , $n = 1, 2, \dots,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

Note that the pattern labels are consistent with D by Definition 4.1.2. We use this assignment to construct a phenotype pattern graph for a regulatory network with proteins Hb, Gt, Kr and Kni, as illustrated in Fig. 7. The network, called the strong edges network (StrongEdges) consists of edges with the strongest predicted interaction between the trunk gap genes from the original gap gene network in Fig. 1(a), except the self-loops (see Fig. 7(left)). Additionally, Fig. 7(right) shows a table representing all possible pattern labels for each region (compare to Table 2).

Consider regions R_1 and R_2 , where $D_{1,\cdot} = (*, H, L, L)$ and $D_{2,\cdot} = (H, *, L, L)$. The collections of pattern labels are

$$\rho(1) = \{(0, 3, 0, 0), (1, 3, 0, 0)\} \text{ and } \rho(2) = \{(1, 1, 0, 0), (1, 2, 0, 0)\}.$$

For pattern labels $c = (0, 3, 0, 0)$ and $d = (1, 1, 0, 0)$ we have

$$\mathcal{T}_{c,d} = \{(0, 3, 0, 0), (0, 2, 0, 0), (0, 1, 0, 0), (1, 3, 0, 0), (1, 2, 0, 0), (1, 1, 0, 0)\}.$$

Doing this for each $c' \in \rho(1)$ and $d' \in \rho(2)$ we find that $\mathcal{T}_{c',d'} \subset \mathcal{T}_{c,d}$. (Note that in general this need not be true.) The entire phenotype graph for StrongEdges can be seen in Fig. 8.

The phenotype pattern describes the annotations of the DSGRN fixed points that we say match the data in each of the eight regions

along the A-P axis. We are interested in finding a sequence of parameter graph nodes with Morse graphs that exhibit fixed points determined by the phenotype pattern. In our example, the associated StrongEdges fixed points for $\rho(1)$ are in the set

$$\{\text{FP}(0, 3, 0, 0), \text{FP}(1, 3, 0, 0)\}.$$

Any Morse graph containing one of these fixed points is consistent with the data in R_1 . For example, a Morse graph having a full cycle connected to $\text{FP}(1, 3, 0, 0)$ as shown in Fig. 8, a bistable Morse graph shown boxed in red, and the monostable Morse graph $\text{FP}(1, 3, 0, 0)$ all exhibit relevant phenotypes for R_1 . Additionally, the two monostable Morse graphs are strict phenotypes of R_1 .

5. Modeling spatial gradients and matching observations with DSGRN

We now describe how spatial data can be compared to DSGRN network model predictions while respecting external variables. The basic construction is a **chemical gradient graph**, whose name is inspired by the spatial distributions of the *D. melanogaster* maternal gradients Bcd and Cad. It is constructed as a subgraph of the DSGRN parameter

graph with directed edges imposed by external variables on the factor graphs of network nodes affected by these external variables. Within the chemical gradient graph, we identify **developmental paths**, with this name again motivated by our example, that are consistent with both the external variables and the phenotype patterns derived from the spatial data.

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

5.1. Chemical gradient graph

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

Let $PG = (P, A)$ be the DSGRN parameter graph of $RN = (V, E)$, where $|V| = M$. Let $\{v_1, \dots, v_m\} \subseteq V$ be the maximal subset of network nodes where each v_j , $j = 1, \dots, m$, is affected by an external variable $c_j(y)$ for $y \in Y$, see Definition 3.2.1. We allow one external variable to affect multiple nodes, but do not consider the case when one node is controlled by multiple external variables. Let $\sigma_j = \pm 1$, where $+1$ (-1) denotes that $c_j(y)$ is an activator (repressor) of v_j . Additionally, recall that $\mathcal{L}_j(p)$ is the factor graph layer of v_j for a parameter node $p \in P$. Lastly, recall that $D_{N,M}$ is the descriptive pattern for a spatial data set with N regions and M genes.

Definition 5.1.1. The **chemical gradient graph**, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, is a directed graph constructed from PG and descriptive pattern $D_{N,M}$ in two steps. First, for $p \in P$ we have $p \in \mathcal{V}$ if $\rho(MG(p)) \in \Sigma$, where Σ is the set of nodes in the phenotype pattern graph (PPG) as defined in Definition 4.1.4. Then for each $p, q \in \mathcal{V}$, we have $(p, q) \in \mathcal{E}$ (directed) if $(p, q) \in A$ and one of the following is satisfied

1. $\mathcal{L}_j(p) = \mathcal{L}_j(q)$ for all $j = 1, \dots, m$, or
2. $\mathcal{L}_k(p) + \text{sign}(c'_k)\sigma_k = \mathcal{L}_k(q)$ for some $k \in \{1, \dots, m\}$ and $\mathcal{L}_j(p) = \mathcal{L}_j(q)$ for all $j = 1, \dots, m$ such that $j \neq k$.

Remark 5.1.1. Notice that condition (1) implies $(p, q), (q, p) \in \mathcal{E}$, while condition (2) implies $(p, q) \in \mathcal{E}$, but $(q, p) \notin \mathcal{E}$, since $\mathcal{L}_k(q) - \text{sign}(c'_k)\sigma_k = \mathcal{L}_k(p)$ flows against the gradient. This fact will be important later.

We now again turn to our example of gap gene network in *D. melanogaster*. Consider the maternal gradients Bcd and Cad, which we model as external variables to the gap gene network. Concentration of these proteins varies monotonically along the A–P axis; Bcd is increasing and Cad is decreasing. Since Fig. 1(a) indicates that both Bcd and Cad may affect Gt and Kr, the spatial variance of this effect along the A–P axis is not clear. Therefore, we chose to only model effects of material gradient that have clear spatial differences along the A–P axis, namely the impact of Bcd on Hb and Cad on Kni. Let $Y = [0, 100]$

represent the A–P axis, where $y = 0$ represents the start of the anterior region and $y = 100$ the end of the posterior region. Then using the notation $c_j(y)$ as in Definition 3.2.1 with $c = Bcd, Cad$ and $j = Hb, Kni$, we have $Bcd_{Hb}(y)$ modeling the effect of Bcd on Hb as an activating ($\sigma_{Hb} = +1$) decreasing ($\text{sign}(Bcd'_{Hb}(y)) = -1$) external variable and $Cad_{Kni}(y)$ modeling the effect of Cad on Kni as an activating ($\sigma_{Kni} = +1$) increasing ($\text{sign}(Cad'_{Kni}(y)) = 1$) external variable (see the violet and cyan gradients in Fig. 1(b)). Thus, if p and q are nodes in the parameter graph $PG = (P, A)$ that satisfy $\rho(MG(p)), \rho(MG(q)) \in \Sigma$ and $(p, q) \in A$, then $(p, q) \in \mathcal{E}$ as well if

1. $\mathcal{L}_{Hb}(p) = \mathcal{L}_{Hb}(q)$ and $\mathcal{L}_{Kni}(p) = \mathcal{L}_{Kni}(q)$, or
2. $\mathcal{L}_{Hb}(p) - 1 = \mathcal{L}_{Hb}(q)$ and $\mathcal{L}_{Kni}(p) = \mathcal{L}_{Kni}(q)$, or
3. $\mathcal{L}_{Kni}(p) + 1 = \mathcal{L}_{Kni}(q)$ and $\mathcal{L}_{Hb}(p) = \mathcal{L}_{Hb}(q)$.

5.2. Developmental paths

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

Definition 5.2.1. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the chemical gradient graph of some $RN = (V, E)$. Let $\{v_1, \dots, v_m\} \subset V$ be network nodes under influence of the corresponding external variables c_j and regulation types σ_j . Let F_j be the factor graphs of v_j , with Lp_j and Hp_j the sets of lowest and highest factor graph layers of F_j respectively. Let $n = 1, \dots, N$ denote the spatial regions of the dataset.

We say that $s \in S \subset \mathcal{V}$ is a **starting node** if

1. $s \in Lp_j$ for all j such that $\sigma_j = +1$ and $c'_j(y) \geq 0$ or $\sigma_j = -1$ and $c'_j(y) \leq 0$,
2. $s \in Hp_j$ for all j such that $\sigma_j = +1$ and $c'_j(y) \leq 0$ or $\sigma_j = -1$ and $c'_j(y) \geq 0$, and
3. the annotation of the Morse set at s must match the phenotype pattern in one of the first ℓ regions, i.e., $\rho(MG(s)) \in \left(\bigcup_{n=1}^{\ell} \mathcal{T}(n)\right) \setminus \rho(\ell + 1)$, where ℓ is a modeling choice.

We say that $t \in \mathcal{T} \subset \mathcal{V}$ is a **stopping node** if

1. $t \in Hp_j$ for all j such that $\sigma_j = +1$ and $c'_j(y) \geq 0$ or $\sigma_j = -1$ and $c'_j(y) \leq 0$,
2. $t \in Lp_j$ for all j such that $\sigma_j = +1$ and $c'_j(y) \leq 0$ or $\sigma_j = -1$ and $c'_j(y) \geq 0$, and
3. the annotation of the Morse set at t must match the phenotype pattern in one of the last k regions, i.e., $\rho(MG(t)) \in \bigcup_{n=N-k}^{N-1} \mathcal{T}(n)$, where k is a modeling choice.

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

Definition 5.2.2. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}; S, \mathcal{T})$ be the chemical gradient graph. A **developmental path** is a path $p_1 \rightarrow \dots \rightarrow p_k$ in $\mathcal{G}$ such that

1. $p_1 \in S$ and $p_k \in \mathcal{T}$, and
2. $\rho(MG(p_1)) \rightarrow \dots \rightarrow \rho(MG(p_k))$ is a path in the phenotype pattern graph.

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

Notice in Definition 5.2.1 we allow starting nodes to be in the first ℓ spatial regions and stopping nodes to be in the last k regions to account for boundary conditions that may not be included in the model. While it would be ideal to find paths in the chemical gradient graph that follow the entire phenotype pattern, we found in the gap gene network *D.*

melanogaster example that nodes with the annotated MG for regions R_1 and R_8 are often disconnected from nodes with the annotated MG for regions R_2 through R_7 . We hypothesize that this is a consequence of additional regulation of gene expression in these regions by genes from the positions external to the A–P positions 35 – 70% where gap genes are active and that are modeled in this paper, see Fig. 6. In particular, there are gene–protein interactions in the anterior between 0%–30% and region R_1 (Jaeger et al., 2004) and in the posterior between 80%–100% and region R_8 (Jaeger, 2011; Ashyraliyev et al., 2009). The lack of accounting for these boundary regulations may impact the ability of network models consisting of only trunk gap genes to recapitulate the data at the extremes of the A–P axis.

5.3. Condensed chemical gradient graph

The DSGRN parameter graph size grows rapidly with the size of the network (Cummins et al., 2018). For example, networks with four nodes and eight edges have millions of parameter graph nodes. Unfortunately, this means that the chemical gradient graph can be quite large. For example, the chemical gradient graph of the StrongEdges network has over 1.4 million nodes and 12 million edges. Graph algorithms such as path-finding rapidly reach computational limits in common chemical gradient graph sizes. Hence, directly finding all developmental paths is impractical. To overcome this limitation, we take two actions.

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

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

Definition 5.3.1. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}; S, \mathcal{T})$ be the chemical gradient graph and let $\mathcal{H}$ be the collection of strongly connected components of $\mathcal{G}$. Define the following equivalence relation over $\mathcal{V}$

$$u \sim v \text{ if and only if } MG(v) = MG(u) \text{ and } u, v \in H \text{ for some } H \in \mathcal{H}.$$

We say

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

is a **strong MG equivalence class** of $\mathcal{V}$ and call u the representative of V_u .

Definition 5.3.2. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}; S, \mathcal{T})$ be the chemical gradient graph. We construct a weighted directed graph named the **condensed chemical gradient graph** $c\mathcal{G} = (c\mathcal{V}, c\mathcal{E}, W)$ as follows. The nodes $c\mathcal{V}$ are the collection of all strong MG equivalence classes of $\mathcal{V}$, i.e.,

$$c\mathcal{V} := \{V_{u_1}, V_{u_2}, \dots, V_{u_k}\},$$

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

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

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

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

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

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

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

We note that there is a slight abuse of notation in the previous definition, wherein V_{u_i} denotes both a strong MG equivalence class and a node in $c\mathcal{G}$. We took a similar liberty for the nodes of the regulatory network $\mathbf{RN}$ and do so here again for clarity.

Notice that the condensed chemical gradient graph ($c\mathcal{G}$) is the condensation of the chemical gradient graph ($\mathcal{G}$), where the strongly connected components of $\mathcal{G}$ are further decomposed by Morse graphs. The following lemma is an immediate consequence of Definition 5.1.1 and Remark 5.1.1.

Lemma 5.3.1. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}; S, \mathcal{T})$ be the chemical gradient graph and let H^i be a strongly connected component of $\mathcal{G}$ where V^i denotes the nodes of the subgraph H^i . Let v_j with $j = 1, \dots, m$ be the externally controlled network nodes with associated $\mathcal{L}_j$ factor graph layers. Then for all $p, q \in V^i$ we have

$$\mathcal{L}_j(p) = \mathcal{L}_j(q)$$

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

The strong MG equivalence classes partition each strongly connected component H^i of $\mathcal{G}$ and so Lemma 5.3.1 guarantees that each one can be unambiguously (though non-uniquely) labeled with a collection $\{\mathcal{L}_j(p)\}$ for $j = 1, \dots, m$. This leads to an immediate Corollary.

Corollary 5.3.1. If $V_u \in cS$ is a starting node in $c\mathcal{V}$ and $p \in V_u$, then p is a starting node in $S \subset \mathcal{V}$. Similarly for stopping nodes.

Proof. Since all $p, q \in V_u$ have matching factor graph layers by Lemma 5.3.1 and matching Morse graphs by Definition 5.3.1, if one node in V_u satisfies the criteria of Definition 5.2.1, they all must. ■

Lemma 5.3.1 and Corollary 5.3.1 justify searching for developmental paths in the (much) smaller condensed chemical gradient graph $c\mathcal{G}$ instead of in $\mathcal{G}$, since paths in $c\mathcal{G}$ follow the externally imposed gradients from starting nodes to stopping nodes.

Definition 5.3.3. Let $c\mathcal{G} = (c\mathcal{V}, c\mathcal{E}, W; cS, c\mathcal{T})$ be the condensed chemical gradient graph. A **condensed developmental path** is a path $V_{u_1} \rightarrow \dots \rightarrow V_{u_k}$ in $c\mathcal{G}$ such that

1. $V_{u_1} \in cS$ and $V_{u_k} \in c\mathcal{T}$, and
2. $\rho(MG(u_1)) \rightarrow \dots \rightarrow \rho(MG(u_k))$ is a path in the phenotype pattern graph.

In the worst case $|\mathcal{V}| = |c\mathcal{V}|$ but in practice $|c\mathcal{V}|$ is much smaller than $|\mathcal{V}|$. For example, while the StrongEdges chemical gradient graph has over 1.4 million nodes, its condensed chemical gradient graph has 14,832 nodes (see Fig. C.1). Importantly, we show that every condensed developmental path in the condensed chemical gradient graph contains at least one developmental path in the chemical gradient graph.

Lemma 5.3.2. For every condensed developmental path $V_{u_1} \rightarrow V_{u_2} \rightarrow \dots \rightarrow V_{u_k}$ there exists at least one developmental path

$$p_1^1 \rightarrow p_2^1 \rightarrow \dots \rightarrow p_{j_1}^1 \rightarrow p_1^2 \rightarrow \dots \rightarrow p_{j_2}^2 \rightarrow \dots \rightarrow p_{j_n}^n$$

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

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

$$P := p_i \rightarrow q_{i+1} \rightarrow v_{j_1} \rightarrow \dots \rightarrow v_{j_n} \rightarrow p_{i+1} \rightarrow q_{i+2}$$

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

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

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

$$p_1 \rightarrow q_2 \rightarrow \dots \rightarrow q_{i+1} \rightarrow v_{j_1} \rightarrow \dots \rightarrow v_{j_n} \rightarrow p_{i+1} \rightarrow q_{i+2} \rightarrow \dots \rightarrow p_k.$$

By Definition 5.3.3, path U induces a path $\rho(MG(u_1)) \rightarrow \rho(MG(u_2)) \rightarrow \dots \rightarrow \rho(MG(u_k))$ in the phenotype pattern graph. Observe that since $u_i, p_i \in V_{u_i}$ and $u_{i+1}, q_{i+1}, v_{j_1}, \dots, v_{j_n}, p_{i+1} \in V_{u_{i+1}}$ then $MG(u_i) = MG(p_i)$ and

$$MG(u_{i+1}) = MG(q_{i+1}) = MG(v_{j_1}) = \dots = MG(v_{j_n}) = MG(p_{i+1}).$$

Thus, $\rho(MG(u_i)) \rightarrow \rho(MG(u_{i+1}))$ implies

$$\rho(MG(p_i)) \rightarrow \rho(MG(q_{i+1})) \rightarrow \rho(MG(v_{j_1})) \rightarrow \dots \rightarrow \rho(MG(v_{j_n})) \rightarrow \rho(MG(p_{i+1}))$$

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

$$\rho(MG(p_1)) \rightarrow \rho(MG(q_2)) \rightarrow \dots \rightarrow \rho(MG(q_{i+1})) \rightarrow \rho(MG(v_{j_1})) \rightarrow \dots \rightarrow \rho(MG(p_k))$$

is a path in the phenotype pattern graph.

Lastly, U is a condensed developmental path so V_{u_1} is a starting node and V_{u_k} is a stopping node in $c\mathcal{G}$, and therefore by Corollary 5.3.1, $p_1 \in V_{u_1}$ is a starting node in $S \subset \mathcal{V}$ and $p_k \in V_{u_k}$ is a stopping node in $\mathcal{T} \subset \mathcal{V}$. Therefore, the path P is a developmental path in $\mathcal{G}$ by Definition 5.2.2. Since U was arbitrary, we have shown there exists at least one developmental path for every condensed developmental path. ■

Lemma 5.3.3. Every developmental path $p_1 \rightarrow \dots \rightarrow p_n$ in the chemical gradient graph can be projected uniquely onto a condensed developmental path $V_{u_1} \rightarrow \dots \rightarrow V_{u_k}$ in the condensed chemical gradient graph. In other words, there is a partition of $\{p_1, \dots, p_n\}$ into k consecutive groups of vertices, each of which belongs to one component V_{u_i} .

Proof. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}; S, \mathcal{T})$ be the chemical gradient graph and let $c\mathcal{G} = (c\mathcal{V}, c\mathcal{E}, W; cS, c\mathcal{T})$ be the condensed chemical gradient graph of $\mathcal{G}$.

Let $\gamma = p_1 \rightarrow \dots \rightarrow p_n$ be a developmental path in $\mathcal{G}$ with induced phenotype pattern path

$$\gamma_\rho = \rho(MG(p_1)) \rightarrow \dots \rightarrow \rho(MG(p_n)) = \rho_1 \rightarrow \dots \rightarrow \rho_n \text{ (for brevity).}$$

Our goal is to uniquely construct a condensed developmental path $\Gamma = V_{u_1} \rightarrow \dots \rightarrow V_{u_k}$ in $c\mathcal{G}$ from γ and γ_ρ .

We partition the parameter nodes in γ into sets $A_{i,j}$ that are the maximal sets of sequential elements in the path that all belong to the same strong MG equivalence class. Formally,

$$A_{i,j} = \{p_i, p_{i+1}, \dots, p_{i+j}\},$$

where

1. $p_i \sim p_{i+1} \sim \dots \sim p_{i+j}$,
2. $p_i \not\sim p_{i-1}$ if $i > 1$, and
3. $p_{i+j} \not\sim p_{i+j+1}$ if $i + j < n$.

In particular, if we denote the partition in order by $A_{i_1,j_1}, \dots, A_{i_k,j_k}$, where $k \leq n$ and $i_s < i_{s+1}$ for $s = 1, \dots, k-1$, then there exists a unique $V_{u_s} \in c\mathcal{V}$ for every A_{i_s,j_s} such that $A_{i_s,j_s} \subset V_{u_s}$. In fact, we may take the representative u_s to be p_{i_s} if desired. Since a developmental path follows a one-way gradient flow and the A_{i_s,j_s} are maximal, it follows that all V_{u_s} are distinct (i.e., no strong MG equivalence class can be revisited). Moreover, since there exists a $p_{\ell} \in A_{i_s,j_s}$ such that $p_{\ell+1} \in A_{i_{s+1},j_{s+1}}$ with $(p_{\ell}, p_{\ell+1}) \in \gamma$, then $(V_{u_s}, V_{u_{s+1}}) \in c\mathcal{E}$. Additionally, since $p_1 \in V_{u_1}$ is a starting node in $S \subset \mathcal{V}$, then $V_{u_1} \in cS$ is also a starting node in $c\mathcal{G}$ by Definition 5.3.2. Similarly, the stopping node $p_n \in V_{u_k}$ implies $V_{u_k} \in c\mathcal{T}$. Therefore $\Gamma = V_{u_1} \rightarrow \dots \rightarrow V_{u_k}$ is a path in $c\mathcal{G}$ from a starting node to a stopping node.

The partition of the nodes in γ into $\{A_{i,j}\}$ and the construction of $\sim$ induces a partition of the pattern labels in γ_ρ , i.e.,

$$\rho(MG(u_s)) = \rho(MG(p_{i_s})) = \dots = \rho(MG(p_{i_s+j_s})) \neq \rho(MG(p_{i_{s+1}+j_{s+1}})) = \rho(MG(u_{s+1}))$$

for $s = 1, \dots, k-1$. Given that γ is a developmental path, then it follows that the path Γ induces a path in the phenotype pattern graph. Uniqueness follows from Definition 5.3.1, completing the proof. ■

5.4. Path graph

We now identify the minimal subgraph of the condensed chemical gradient graph that contains all condensed developmental paths subject to an additional constraint on near simultaneous change of external variables along the paths.

Definition 5.4.1. Given the condensed chemical gradient graph $c\mathcal{G} = (c\mathcal{V}, c\mathcal{E}, W; cS, c\mathcal{T})$ and the phenotype pattern graph $PPG = (\Sigma, E_\Sigma)$, the **tensor product graph** $c\mathcal{G} \times PPG$ contains the nodes $\{(V_u, s) \mid V_u \in c\mathcal{V}, s \in \Sigma\}$ and edges $((V_u, s), (V_{u'}, s'))$ if $(V_u, V_{u'}) \in c\mathcal{E}$ and $(s, s') \in E_\Sigma$. Let $H \subset c\mathcal{G} \times PPG$ be the subgraph induced by nodes (V_u, s) that satisfy

$$\rho(MG(u)) = s. \quad (6)$$

Finally, let $c\mathcal{H}$ be a projection of H onto the first component. In other words, $c\mathcal{H}$ is the subgraph of $c\mathcal{G}$ such that a path exists in $c\mathcal{H}$ if and only if that path induces a path in PPG .

Given a regulatory node v affected by some monotone external variable $c(y)$, we have that any path in $c\mathcal{H}$ follows a monotone path in the factor graph of v (see Definition 3.1.3), i.e., the path must be consistent with external variable effect. In the gap gene network, the external variables Bcd and Cad impose a requirement that Hb decreases and, at the same time, Kni increases. However, our construction of $c\mathcal{H}$ allows Hb to decrease entirely before Kni increases and vice-versa. This does not capture the biological reality where the maternal gradients change simultaneously, i.e., Hb is decreasing while Kni is increasing along the A-P axis of the embryo. We capture this behavior in our final graph construction, the **path graph**. However, we must first define what it means for external variables to be changing simultaneously in the context of DSGRN. In order to do that we develop the concept of a **diagonal subgraph** in a product of oriented graphs.

Definition 5.4.2. Consider $RN = (V, E)$ and regulatory nodes $\{v_1, \dots, v_m\} \subseteq V$, with each v_j affected by a corresponding external variable $c_j(y)$. Let $\mathcal{L}_j^* = |R_j| \cdot |T(v_j)| + 1$, i.e., $\mathcal{L}_j^*$ is the number of factor graph layers for v_j . We call

$$r_{i,j} := \left\lceil \max \left\{ \frac{\mathcal{L}_j^*}{\mathcal{L}_i^*}, \frac{\mathcal{L}_i^*}{\mathcal{L}_j^*} \right\} \right\rceil$$

the **size ratio** between v_i and v_j .

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

$$r_{Hb,Kni} = \left\lceil \max \left\{ \frac{\mathcal{L}_{Hb}^*}{\mathcal{L}_{Kni}^*}, \frac{\mathcal{L}_{Kni}^*}{\mathcal{L}_{Hb}^*} \right\} \right\rceil = 3.$$

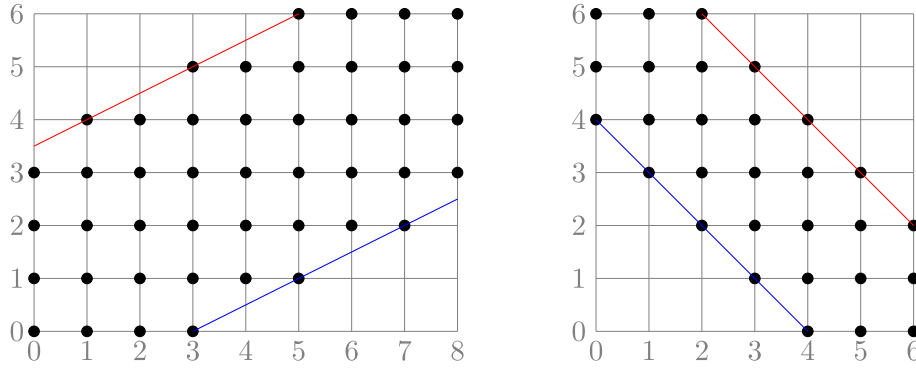


Fig. 9. Example of sets $S_{i,j}^d$ and $S_{i,j}^a$. (left) All black points are the points in the set $S_{i,j}^d$ for $a_i = 8$, $a_j = 6$ and $r = 2$ and (right) $S_{i,j}^a$ with $a_i = a_j = 6$ and $r = 1$. In both grids, the red lines depict the upper bound of y in $S_{i,j}^d$ and $S_{i,j}^a$ while the blue lines depict the lower bound of y .

Returning to the general case, fix two regulatory nodes $v_i, v_j \in \{v_1, \dots, v_m\}$. In order to simplify notation we set $r := r_{i,j}$. The number r will be used to describe a subregion S within a rectangle $R = [0, a_i] \times [0, a_j] \subset \mathbb{Z} \times \mathbb{Z}$, where $a_i := \mathcal{L}_i^* - 1$, $a_j := \mathcal{L}_j^* - 1$. To be explicit we assume $a_i \geq a_j$.

When $\text{sign}(c'_i) = \text{sign}(c'_j)$ then we define a neighborhood of a diagonal in R , $S_{i,j}^d \subset R$ by

$$S_{i,j}^d := \{x, y \in R \mid x \in [0, a_i] \text{ and } \frac{1}{r}x - \frac{r+1}{r} \leq y \leq \frac{1}{r}(x - a_i) + a_j + \frac{r+1}{r}\}. \quad (7)$$

Note that the lower bound of $S_{i,j}^d$ is a line passing through point $(r+1, 0)$ and the upper bound through the point $(a_i - r - 1, a_j)$, both with the same slope $\frac{1}{r}$, see Fig. 9(left). For the case when $\text{sign}(c'_i) \neq \text{sign}(c'_j)$ we set

$$S_{i,j}^a := \{x, y \in R \mid x \in [0, a_i] \text{ and } -\frac{1}{r}(x - a_i) - \frac{r+1}{r} \leq y \leq -\frac{1}{r}x + a_j + \frac{r+1}{r}\}, \quad (8)$$

see Fig. 9(right).

Definition 5.4.3. Consider $\text{RN} = (V, E)$ and regulatory nodes $\{v_1, \dots, v_m\} \subseteq V$, with each v_j for $j = 1, \dots, m$ affected by a corresponding external variable $c_j(y)$, and let $c\mathcal{H}$ be the graph as constructed in Definition 5.4.1. The **diagonal subgraph** $\mathcal{D}$ is a node-induced subgraph of $c\mathcal{H}$ where a node $V_p \in c\mathcal{H}$ belongs to $\mathcal{D}$ if and only if the corresponding factor graph layers satisfy

$$(\mathcal{L}_i(p), \mathcal{L}_j(p)) \in \begin{cases} S_{i,j}^d & \text{if } \text{sign}(c'_i) = \text{sign}(c'_j) \\ S_{i,j}^a & \text{if } \text{sign}(c'_i) \neq \text{sign}(c'_j) \end{cases}$$

for all pairs $v_i, v_j \in \{v_1, \dots, v_m\}$ where $\mathcal{L}_i^* \geq \mathcal{L}_j^*$.

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

Definition 5.4.4. Let $c\mathcal{G} = (c\mathcal{V}, c\mathcal{E}, W; c\mathcal{S}, c\mathcal{T})$ be the condensed chemical gradient graph, and consider the diagonal subgraph $\mathcal{D}$ of $c\mathcal{H}$. A node $V_u \in \mathcal{D}$ has **terminal reach** if one of the following is satisfied.

1. If $V_u \notin c\mathcal{S} \cup c\mathcal{T}$, then there is both a path in $\mathcal{D}$ from at least one $V_s \in c\mathcal{S}$ to V_u and a path in $\mathcal{D}$ from V_u to at least one $V_t \in c\mathcal{T}$.
2. If $V_u \in c\mathcal{S}$, then there is a path in $\mathcal{D}$ from V_u to at least one $V_t \in c\mathcal{T}$.
3. If $V_t \in c\mathcal{T}$, then there is a path in $\mathcal{D}$ from at least one $V_s \in c\mathcal{S}$ to V_u .

Definition 5.4.5. The **path graph** $P = (\mathcal{V}, \mathcal{E})$ is the node-induced subgraph of $\mathcal{D}$ where we have removed all nodes V_u (and incident edges) if V_u does not have terminal reach in $\mathcal{D}$. The edge $(V_u, V_w) \in \mathcal{E}$

inherits the weight of the edge $(V_u, V_w) \in c\mathcal{E}$. A node in $\mathcal{V}$ is a starting (stopping) node for P if it is a starting (stopping) node in $c\mathcal{G}$.

The next theorem summarizes our construction.

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

We showed that every condensed developmental path in $c\mathcal{G}$ represents a developmental path in $\mathcal{G}$ and that every developmental path in $\mathcal{G}$ can be projected uniquely onto a condensed developmental path in $c\mathcal{G}$. Since the path graph contains all condensed developmental paths that satisfy the biologically motivated constraint of simultaneous external variable change, it is sufficient to analyze the path graph in order to understand the structure of the developmental paths in $\mathcal{G}$. See Appendix Fig. C.1 for a visualization of $c\mathcal{G}$ and P for both the example gap gene models Fullconn and StrongEdges.

6. Robustness scores

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

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

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

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

6.1. Bottlenecks scored by optimized weighted cut (OWCut) of the path graph

A common understanding of a bottleneck is a restriction point for traffic. In our case, the path graph describes a set of developmental paths that represent a sequence of parameter changes consistent with external variables that recapitulate qualitative observations about variable expression across spatial regions. A bottleneck indicates a set of parameter regions where a random perturbation will likely disrupt the phenotype pattern. Our goal is to develop a mathematical definition of a bottleneck that captures this behavior in the setting of a directed weighted graph. To do so, we utilize the concept of weighted graph cut from (Meilă and Pentney, 2007).

Definition 6.1.1. A **weighted** directed graph $G = (V, E, W)$ is a directed graph equipped with a non-negative weight $w_{i,j}$ assigned to each directed edge (v_i, v_j) . We organize weights in a $|V| \times |V|$ **weight matrix** $W = [w_{i,j}]$. If $(v_i, v_j) \notin E$, then $w_{i,j} = 0$.

Definition 6.1.2. Let W be the weight matrix of G . Let D be a diagonal matrix where

$$D_i = \begin{cases} 1 & \text{if } \sum_{v_j \in V} w_{i,j} = 0, \\ \sum_{v_j \in V} w_{i,j} & \text{otherwise.} \end{cases}$$

where $w_{i,j} \in W$. Notice if all edges have weight 1, then $\sum_{v_j \in V} w_{i,j}$ is the out-degree of v_i . We call D the **weighted out-degree matrix** of the graph G (Meilă and Pentney, 2007).

Definition 6.1.3. def:Clusteret $\{C_1, C_2, \dots, C_K\}$ be a collection of disjoint subsets of V , i.e., $C_k \subset V$ for $k = 1, 2, \dots, K$ and $C_i \cap C_j = \emptyset$ for $i \neq j$. If

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

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

$$\mathcal{D}_k = \sum_{v_i \in C_k} D_i$$

the **degree of cluster k** (Meilă and Pentney, 2007).

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

Definition 6.1.4. Let W and D be the weight and weighted out-degree matrices of weighted directed graph G respectively and let $C = \{C_1, C_2, \dots, C_K\}$ be a K-clustering of G . Then the **weighted cut** of C is

given by Meilă and Pentney (2007)

$$WCut(C) = \sum_{k=1}^K \frac{1}{\mathcal{D}_k} \sum_{v_i \in C_k} (D_i - \sum_{v_j \in C_k} w_{i,j}).$$

Further, let

$$s_k := \sum_{v_i \in C_k} \sum_{v_j \in C_k} w_{i,j} \quad \text{and} \quad d_k := \sum_{v_i \in C_k} \sum_{v_j \in V \setminus C_k} w_{i,j}.$$

Notice that s_k is the sum of weights for every edge in the node-induced subgraph of G with node set C_k , denoted $G[C_k]$, while d_k is the sum of weights of edges departing $G[C_k]$. If each $w_{i,j}$ represents a probability of taking the edge from node v_i to v_j , then in general a path starting in C_k has a higher probability of staying in C_k whenever s_k is higher than d_k .

Definition 6.1.5. Let $G = (V, E, W)$ be a weighted directed graph with a K-cluster C . Given a cluster $C_k \in C$, we say that G has a **bottleneck** from $G[C_k]$ into $G[V \setminus C_k]$ whenever $s_k > d_k$, i.e., whenever a path starting in $G[C_k]$ has a better chance of staying in $G[C_k]$ than it does of departing $G[C_k]$.

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

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

$$WCut(\{C_k, V \setminus C_k\}) < \frac{1}{2}$$

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

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

$$\begin{aligned} \sum_{v_i \in C_k} \sum_{v_j \in V} w_{i,j} &= \sum_{v_i \in C_k} \left(\sum_{v_j \in C_k} w_{i,j} + \sum_{v_j \in V \setminus C_k} w_{i,j} \right) \\ &= \sum_{v_i \in C_k} \sum_{v_j \in C_k} w_{i,j} + \sum_{v_i \in C_k} \sum_{v_j \in V \setminus C_k} w_{i,j} = s_k + d_k, \end{aligned}$$

where the first equality is because $C_k \sqcup V \setminus C_k = V$. Then

$$\mathcal{D}_k \geq \sum_{v_i \in C_k} \sum_{v_j \in V} w_{i,j} = s_k + d_k$$

with equality when $D_i := \sum_{v_j \in V} w_{i,j} \neq 0 \forall i$. We can also express $WCut(C)$ of the clustering in terms of weights of internal edges s_k and external edges d_k .

$$\begin{aligned} WCut(C) &= \sum_{k=1}^K \frac{1}{\mathcal{D}_k} \sum_{v_i \in C_k} \left(D_i - \sum_{v_j \in C_k} w_{i,j} \right) \\ &= \sum_{k=1}^K \frac{1}{\mathcal{D}_k} \left(\sum_{v_i \in C_k} D_i - \sum_{v_i \in C_k} \sum_{v_j \in C_k} w_{i,j} \right) \\ &= \sum_{k=1}^K \frac{1}{\mathcal{D}_k} (\mathcal{D}_k - s_k) \\ &= \sum_{k=1}^K \left(1 - \frac{s_k}{\mathcal{D}_k} \right) \\ &\leq \sum_{k=1}^K \left(1 - \frac{s_k}{s_k + d_k} \right). \end{aligned}$$

We note that

$$1 - \frac{s_k}{s_k + d_k} < \frac{1}{2} \quad \text{if and only if } d_k < s_k.$$

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

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

$$1 - \frac{s_k}{s_k + d_k} = 0.$$

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

$$1 - \frac{s_k}{s_k + d_k} = 1.$$

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

Thus, to score property P1 we will find the optimal 2-clustering of the path graph P that minimizes the $WCut$ of P , with the additional condition that one cluster contains all the starting nodes while the other contains all the stopping nodes. We compute an optimal 2-clustering as follows using methods inspired by the normalized Laplacian from (Meilă and Pentney, 2007) and a grouping algorithm for image segmentation (Shi and Malik, 2000) based on spectral clustering. Let W and D be the weighted matrix and weighted degree matrix of P , respectively. Computing the Hermitian part of the Laplacian $L = (D - W)$ and normalizing by D (Meilă and Pentney, 2007), we have

$$\mathcal{L} := D^{-\frac{1}{2}} H(L) D^{-\frac{1}{2}} = \frac{1}{2} D^{-\frac{1}{2}} (2D - W - W^T) D^{-\frac{1}{2}}.$$

The key insight from spectral theory of harmonic maps is that while the smallest eigenvalue of $\mathcal{L}$ has the eigenvector with positive entries, the eigenvector corresponding to the second eigenvalue has entries of both signs, and the nodes corresponding to each sign form two clusters which minimize $WCut$. Since the value zero does not have a privileged position in our matrix $\mathcal{L}$, we follow (Shi and Malik, 2000) to find an optimum value to define the clusters. Let $\mathbf{x} = (x_1, x_2, \dots, x_{|V|})$ be the eigenvector corresponding to the second smallest eigenvalue of $\mathcal{L}$. We reorder elements in $\mathbf{x}$ in ascending order and let $(v_1, v_2, \dots, v_{|V|})$ be the corresponding ordering of nodes in V . Let j be the index such that

$$|x_j - x_{j+1}| = \max_{i \leq |V|-1} \{|x_i - x_{i+1}|\}.$$

Then we define a 2-clustering of nodes by setting $C_1 := (v_1, \dots, v_j)$ and $C_2 = (v_{j+1}, \dots, v_{|V|})$ (Meilă and Pentney, 2007; Shi and Malik, 2000). If the starting and stopping nodes are not contained in separate clusters, we find a clustering based on the next largest consecutive distance $|x_j - x_{j+1}|$, until this condition is satisfied. We denote the **optimum 2-clustering of P** (with starting/stopping node conditions satisfied) by

$$OWCut(P) := WCut(\{C_1, C_2\}).$$

We remark that the $OWCut(P)$ is not computable for some path graphs P because it is not possible to separate the starting and stopping nodes into separate clusters. See Appendix Fig. C.2 for a visualization of the optimal clustering of P for both Fullconn and StrongEdges.

6.2. Using absorbing Markov chains (AMC) to score leak (P2) and skip (P3)

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

$$\begin{aligned} \mathcal{O} &:= \{(V_u, V_v) \in c\mathcal{E} \mid V_u \in \mathcal{V} \text{ and } V_v \notin \mathcal{V}\} \\ \mathcal{J} &:= \{(V_u, V_v) \in c\mathcal{E} \mid V_u, V_v \in \mathcal{V} \text{ and } (V_u, V_v) \notin \mathcal{E}\}. \end{aligned}$$

The edges in $\mathcal{O}$ capture paths that leave the path graph P ; this represents a *leak* from the set of developmental paths to those which do not recapitulate the observed phenotype pattern. On the other hand, both vertices of the edges in $\mathcal{J}$ lie in $\mathcal{V}$, but the edge is not in $\mathcal{E}$. Therefore edges in $\mathcal{J}$ represent paths that *skip* a portion of the phenotype pattern.

We say an edge $(V_u, V_v) \in c\mathcal{E}$ is a **leak** edge if $(V_u, V_v) \in \mathcal{O}$ and a **skip** edge if $(V_u, V_v) \in \mathcal{J}$.

We use absorbing Markov chains to quantify the amount of leak from P into $c\mathcal{G}$ and skip within P . A **Markov chain** (Privault, 2018) is a discrete stochastic process with a finite number of states $\{s_i\}_{i=0}^n$ for $n \in \mathbb{N}$ where the probability of transitioning from state s_{i_k} to state $s_{i_{k+1}}$ depends only on the current state s_{i_k} and not on previous states $s_{i_{k-1}}, \dots, s_{i_0}$, i.e.,

$$p(s_{i_{k+1}} | s_{i_k}, s_{i_{k-1}}, \dots, s_{i_0}) = p(s_{i_{k+1}} | s_{i_k}).$$

This is called the Markov property. As a consequence, a Markov chain can be represented by a **transition matrix** (Thompson and McNeal, 1967) $\mathcal{W}$ where $\omega_{i,j}$ is given by $\omega_{i,j} := p(s_j | s_i)$ and

$$\sum_{j=1}^n \omega_{i,j} = 1 \quad (9)$$

for each $i = 1, \dots, n$ (Thompson and McNeal, 1967). An **absorbing state** is a state s_i with $p(s_i | s_i) = 1$ and a **transient state** is any state that is not absorbing. An **absorbing Markov chain** is a Markov chain where each state can reach an absorbing state in a finite number of steps (Ermon et al., 2014; Thompson and McNeal, 1967). The probabilities of transitioning from transition state s_i to absorbing state s_j in an AMC can be calculated from its transition matrix $\mathcal{W}$ (Thompson and McNeal, 1967).

Definition 6.2.1. Let $P = (\mathcal{V}, \mathcal{E})$ be the path graph of the condensed chemical gradient graph $c\mathcal{G} = (c\mathcal{V}, c\mathcal{E}, W; cS, c\mathcal{T})$. The **absorbing Markov chain expansion of P** , denoted by $AMC(P, l, s) = (\mathcal{U}, \mathcal{M})$ with transition matrix $\mathcal{W}$, is defined as follows. The nodes are $\mathcal{U} = \mathcal{V} \cup \{l, s\}$, where l and s are nodes that represent all states that are targets of edges in $\mathcal{O}$ and $\mathcal{J}$, respectively. Consider the following sets of weighted edges

$$\mathcal{E}^* = \{(V_u, V_v, w_{i,j}) \mid (V_u, V_v) \in \mathcal{E}\},$$

$$\mathcal{O}^* = \{(V_u, l, w_{i,j}) \mid (V_u, V_v) \in \mathcal{O}\},$$

$$\mathcal{J}^* = \{(V_u, s, w_{i,j}) \mid (V_u, V_v) \in \mathcal{J}\},$$

where $w_{i,j} \in W$. Then the set of edges of $AMC(P, l, s)$ is

$$\mathcal{M} = \mathcal{E}^* \cup \mathcal{O}^* \cup \mathcal{J}^* \cup \{(V_u, V_u, 1) \mid V_u \in c\mathcal{T}\} \cup \{(l, l, 1) \cup (s, s, 1)\}.$$

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

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

6.3. Size of lifted path graph in chemical gradient graph (P4)

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

Let

$$\mathcal{V}_P = \{v \in \mathcal{V} \mid v \in V_u \text{ for some } V_u \in \mathcal{V}\}.$$

Similarly, for $(V_u, V_v) \in \mathcal{E}$, consider the associated collection of edges in $\mathcal{G}$

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

and let

$$\mathcal{E}_P = \bigcup_{(V_u, V_v) \in \mathcal{E}} \mathcal{E}_{u,v}.$$

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

$$g(P, \mathcal{G}) = \frac{|\mathcal{V}_P| + |\mathcal{E}_P|}{|\mathcal{V}| + |\mathcal{E}|}.$$

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

6.4. Scoring

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

$$B(N) := OWCut(P^N), PS(N) := g(P^N, \mathcal{G}^N) \text{ and } LS(N) := 1 - (p(l) + p(s)).$$

in such a way as to score the robustness of N .

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

$$\hat{f}(N) := \frac{f(N) - \min_{\tilde{N} \in \mathcal{N}}(f(\tilde{N}))}{\max_{\tilde{N} \in \mathcal{N}}(f(\tilde{N})) - \min_{\tilde{N} \in \mathcal{N}}(f(\tilde{N}))}$$

be our normalization, then we give N the **robustness score**

$$S(N) = \frac{\hat{B}(N) + \hat{PS}(N) + \hat{LS}(N)}{3}.$$

7. Results

We consider two candidate network models for *D. melanogaster* development: StrongEdges (introduced in Section 4.2) and FullConn (introduced in Section 2.1.3). We found that both networks were capable of capturing the protein expression data from regions R_2 to R_7 seen in Fig. 6. The remainder of the results are dedicated to evaluating and comparing their robustness. While values of the robustness score $S(N)$ for some networks N are hard to interpret in a physical sense, we can use these values to compare network models. In particular, we wish to compare the networks StrongEdges and FullConn with a class of random networks. If these networks are valid representations of the gap gene network (which is known to be robust) then they should, in theory, have higher robustness scores than the average randomly generated network. Additionally, we would like to know if any network properties impact a network's robustness score. To accomplish these tasks, we must define a set of random networks, as well as define network properties we wish to evaluate. Given that both StrongEdges and FullConn have four nodes (Hb, Gt, Kr, and Kni) and eight edges, we restrict our attention to networks with nodes Hb, Gt, Kr, and Kni as well as eight edges. These can be any combination of edges between the nodes, with either activating or repressing signs. There are 126,720 networks in this class, with DSGRN parameter graph sizes ranging between 1.44 and 23.064 million nodes. For computational reasons, we only consider networks with a DSGRN parameter graph size up to 4.32 million nodes. We note that FullConn and StrongEdges have DSGRN parameter graph sizes of 2.56 and 3.24 million nodes respectively, so this range allows comparison with networks that have both smaller and larger parameter graphs. This class has 58,366 networks, which will be our **network population**, denoted by $\mathcal{N}$.

Calculating the score $S(N)$ for a single network N takes between 5 and 30 min, with time heavily dependent on the chemical gradient

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

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

1. Subnetworks of the gap gene network from Verd et al. (2019), as seen in Fig. 1. Networks $N \in \mathcal{N}$ that satisfy this condition are said to have the feature *Verd*.
2. Subnetworks of the gap gene network from Reinitz (Manu et al., 2009), see Appendix Fig. D.1. Since the Reinitz gap gene network is a subgraph of Verd, a N that is a subgraph of the Reinitz gap gene network is also a subgraph of Verd. We call this feature *Reinitz*. When a N has the Verd feature but not the Reinitz feature, we say this N has the *strict Verd* feature.
3. The ACDC 1-3 motifs from left to right in Fig. 2. We call these the *ACDC1*, *ACDC2* and *ACDC3* features.
4. Networks that have all four repressing edges Hb to Kni, Kni to Hb, Gt to Hb and Hb to Gt, which are the edges with the most biological evidence. We call this the *Ultra Strong* feature.
5. The number of repressing edges in N . We call this integer-valued property the *RE* feature.
6. The number of negative and positive feedback loops in N . These are the *NFL* and *PFL* features, respectively.

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

In order to evaluate how subgraph features impact our score, we used a stratified simple random sampling method to select more networks. Using the stratification of $\mathcal{N}$ into two disjoint groups, F and $\mathcal{N} \setminus F$ we randomly sampled 200 additional networks from F , denoted B_F . We remark that $B \cap B_F = \emptyset$. We call B_F the **feature group** and Table 3 shows a breakdown of the number of networks in $\mathcal{N}$, B , and B_F , the number of networks in $\mathcal{N}$ we attempted to score, and the number of networks we were able to score.

Of the networks we attempted to score, only 11 were unable to reproduce the data, i.e., they did not contain a developmental path. One of these networks (network 21,283) was unable to reproduce the data due to having no stopping nodes in the chemical gradient graph. The topology of this network can be seen in Appendix Fig. D.1. The other 10 had path graphs that became disconnected after imposing the requirement that developmental paths follow maternal gradient flow simultaneously (see Definition 5.4.3). Additionally, there were 19 networks where *OWCut* could not be calculated with the condition that the starting and stopping nodes be in separate clusters. Due to not having a calculated score, we left these 30 networks out of our statistical analysis.

We would like to answer the following questions:

1. Is FullConn or StrongEdges more robust than the average network from our population?
2. Is there evidence of a difference in robustness between our baseline and feature groups? Specifically, we ask if networks

Table 3

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

Subgraph feature	$\mathcal{N}$						B						B_F						$B \cup B_F$					
	Total		Disjoint		Sampled		Scored		Sampled		Scored		Sampled		Scored		Total scored							
ACDC1	468		387		7		7		44		43						50							
ACDC2	468		411		3		3		34		32						35							
ACDC3	468		387		9		7		41		40						47							
Ultra Strong	704		583		10		10		75		72						82							
Strict Verd	170		116		0		0		18		17						17							
Reinitz	85		37		1		1		14		13						14							
Count of number feature	RE	$\mathcal{F}$	PFL	$\mathcal{F}$	NFL	$\mathcal{F}$	RE	PFL	NFL	RE	PFL	NFL	RE	PFL	NFL	RE	PFL	NFL	RE	PFL	NFL	RE	PFL	NFL
0	255	0	1680	0	2160	120	1	31	33	1	31	33	0	0	17	0	0	16	1	31	49			
1	1824	0	7680	128	5952	217	13	119	82	13	113	81	0	17	21	0	16	20	13	129	101			
2	6060	0	14256	488	15888	752	83	191	183	82	188	175	0	43	69	0	41	66	82	229	241			
3	12120	0	17712	687	18672	668	141	227	214	137	218	208	0	57	53	0	56	52	137	274	260			
4	15690	119	12912	692	10512	287	207	131	173	201	125	167	9	68	30	9	65	30	210	190	197			
5	13200	510	3168	87	4128	71	179	42	51	172	42	49	48	9	9	47	9	9	219	51	58			
6	6924	850	720	36	1008	7	96	7	15	90	7	14	91	6	1	86	6	0	176	13	14			
7	2040	534	240	4	48	0	29	4	1	29	4	1	42	0	0	41	0	0	70	4	1			
8	255	109	0	0	0	0	3	0	0	3	0	0	10	0	0	10	0	0	13	0	0			
Total	58368						752						200						193					

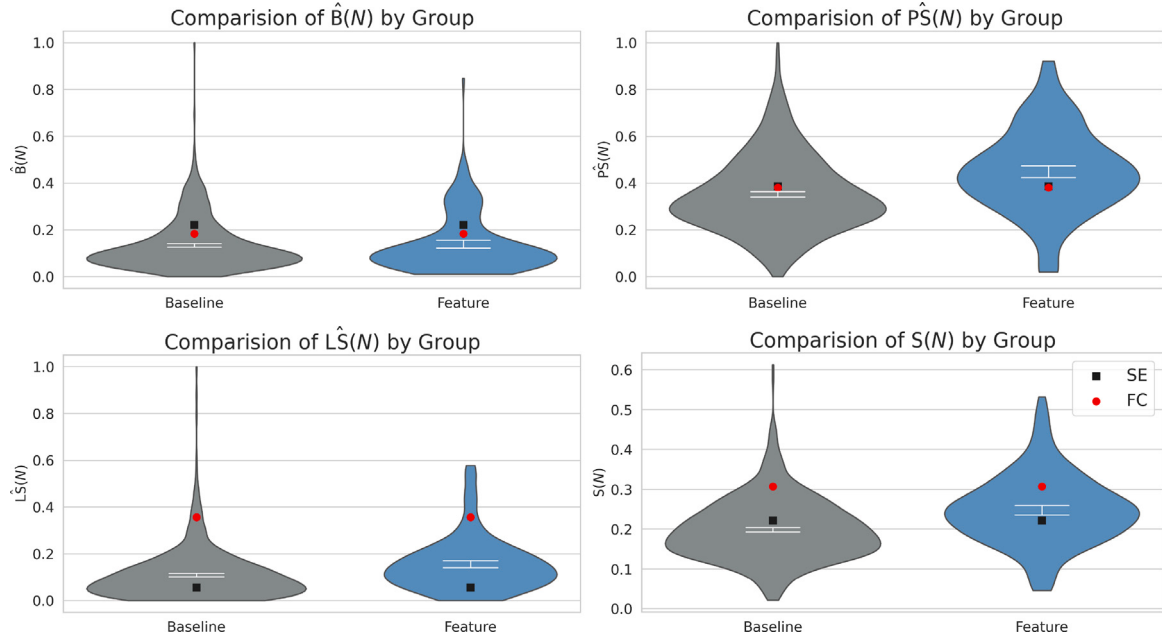


Fig. 10. Violin plots comparing normalized robustness scores between the baseline (gray) and feature (blue) groups. White bars depict the 95% mean confidence intervals. The red dot and black square indicate the score for FullConn and StrongEdges respectively. Note in particular when the scores for FullConn and StrongEdges lie outside the 95% mean confidence intervals for each group.

containing at least one of our noted features will be more or less robust than average.

3. Is there a relationship between any of the features and the network robustness score? Specifically, we want to know if there is evidence of a specific feature having an impact on the robustness score.

Fig. 10 shows a summary of our analysis for both the baseline (gray) B and the subgraph feature group (blue) B_F , together with means and the 95% mean confidence intervals for each of the normalized scores and the overall robustness score. The numerical values can be seen in Appendix Table F.1, along with the results for FullConn and

StrongEdges. Additionally, see Appendix Fig. E.1 for a summary of score results before applying the normalization defined in Section 6.4. We see that both FullConn and StrongEdges have robustness scores that lie outside of the 95% confidence intervals for the baseline group. In particular, FullConn exceeds the 95% mean confidence intervals for the baseline group in all robustness scores and exceeds the 95% mean confidence interval for the feature group in all scores but $\hat{P}S$. StrongEdges is a worse performer, with $\hat{L}S$ (StrongEdges) below the 95% mean confidence intervals in the baseline and feature groups, and the overall score S (StrongEdges) below the 95% mean confidence interval in the feature group. Comparing the baseline and feature distributions, we see evidence that a network containing a subgraph feature, on

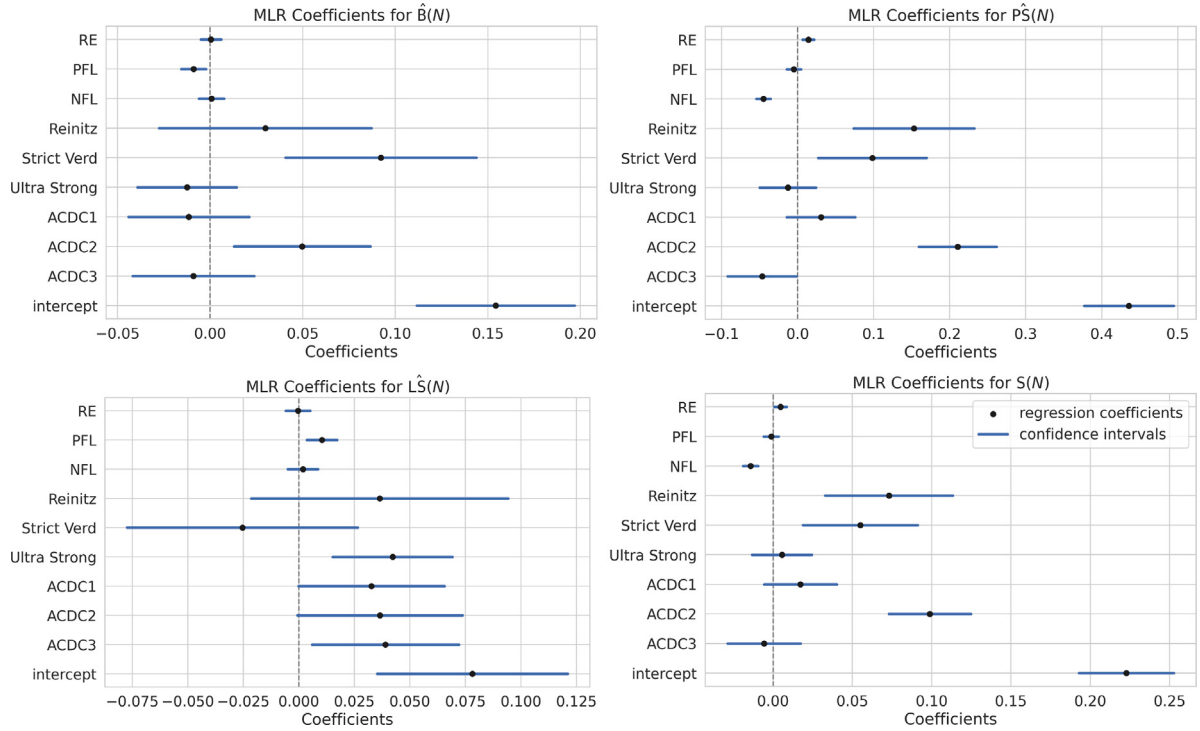


Fig. 11. Regression coefficients for each explanatory variable in our model and model intercept, plotted with 95% confidence intervals. See Appendix Table F.2 for coefficients, 95% confidence intervals, t and p-values for each of the response variables $\hat{B}$ (top left), $\hat{P}$ (top right), $\hat{L}$ (bottom left) and S (bottom right).

average, is more robust than a random network, primarily due to higher performing $\hat{L}$ and $\hat{P}$ scores.

To address question 3, we used multi-linear regression (MLR) (Greenwood, 2022) on all networks scored. The subgraph and number features were set as the explanatory (independent) variables and the network score as the response (dependent) variable. All assumptions to ensure the validity of this model were checked, see Appendix F.

Our results are summarized in Fig. 11. Each explanatory variable coefficient is depicted by a black dot, and the 95% confidence interval is depicted by the blue line on either side of the coefficients. For example, for S in the lower right panel, we can see that when a network contained ACDC2 as a subnetwork, we are 95% confident that the true mean of the network score is increased by between 0.073 and 0.123 in our population, after adjusting for the additional possible presence of ACDC1, ACDC3, Ultra Strong, Strict Verd, Reinitz, NFL, PFL, and RE. Additionally, for each repressive edge added to a regulatory network, we are 95% confident that the true mean of the network score is increased by between 0.0008 and 0.0086 in our population, after adjusting for the presence of the remaining features. From this, we saw evidence that 5 of the 9 features (ACDC2, Reinitz, Strict Verd, RE and NFL) had an increasing or decreasing effect on our overall network robustness score S . The impacts of feature groups on the component scores $\hat{B}$, $\hat{P}$, and $\hat{L}$ are also shown.

In more detail, let F denote our set of explanatory variables. Given an explanatory variable $f \in F$, let $H_{0,f}$ be the null hypothesis of f , which states that there is no linear relationship between the network score and f , once we have accounted for all explanatory variables in $F \setminus \{f\}$. To determine if we can reject the null hypothesis, we use a t-statistic and a p-value. Since our model has 906 degrees of freedom (number of observations minus number of variables), then a t-statistic (denoted t_{906}) above 1.963 or below -1.963 is considered significant at the 95% level, along with a p-value less than 0.05 (Greenwood, 2022). During our analysis, we found that there was evidence for a positive linear relationship between network robustness and feature ACDC2, once we accounted for the remaining features; i.e., there was evidence

against $H_{0,ACDC2}$ ($t_{906} = 7.4970$, $p\text{-value} < 0.0001$). That is, the presence of ACDC2 on average increased the score of any network of which it is a subnetwork. We also found evidence for positive linear relationships in Reinitz ($t_{906} = 3.5587$, $p\text{-value} = 0.0004$), Strict Verd ($t_{906} = 2.9821$, $p\text{-value} = 0.0029$), and RE ($t_{906} = 2.3844$, $p\text{-value} = 0.0173$). On the other hand, we saw evidence for a negative relationship in NFL ($t_{906} = -5.8644$, $p\text{-value} < 0.0001$). Lastly, we saw little to no evidence against the null hypotheses for ACDC1 ($t_{906} = 1.4800$, $p\text{-value} = 0.1392$), ACDC3 ($t_{906} = -0.4808$, $p\text{-value} = 0.6308$), Ultra Strong ($t_{906} = 0.5861$, $p\text{-value} = 0.5580$) or PFL ($t_{906} = -0.4890$, $p\text{-value} = 0.6250$).

Fig. 11 shows the impact of the three constituent robustness scores of S as well. For example, we see evidence to suggest that there exists a positive linear relationship between the path size score $\hat{P}$ and feature Reinitz once we account for all other features ($t_{906} = 3.7818$, $p\text{-value} = 0.0002$). However, we see little to no evidence that there exists a linear relationship between Reinitz and the bottleneck score $\hat{B}$ ($t_{906} = 1.0178$, $p\text{-value} = 0.3090$), as well as the leak-skip score $\hat{L}$ ($t_{906} = 1.2341$, $p\text{-value} = 0.2175$), once we account for all other features. This suggests that the Reinitz feature impacted the overall robustness score by increasing the size of the lifted path graph in the chemical gradient network.

8. Discussion

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

Our main mathematical contributions are three-fold. First, we conceptually reinterpreted the output of the DSGRN methodology to enable modeling of spatially arranged cells that are impacted by monotone

control variables. This was done by proving that DSGRN factor parameter graphs can be represented as graded posets. Second, we defined a *path graph* based on the graded posets that permits a DSGRN network model to match spatial experimental data subject to constraints from monotone control variables. The path graph summarizes all the ways in which a network model is capable of matching the data under these constraints. Lastly, we developed evaluation criteria for the robustness of the match between model and data by devising three *robustness scores* that quantify the structural fragilities of the path graph. These structural fragilities can be interpreted as obstacles to correct development.

Our major biological contributions are a rank ordering of proposed gap gene network models in *D. melanogaster* according to robustness score, a quantification of their performance over random networks, and a characterization of the impact of various network motifs on network model performance. In particular, we showed that while it is common for a network model to be able to match experimental data, a network model inspired by Verd et al. (2019) (FullConn) shows strong robustness scores compared to a random sample of network models. We also identified a motif (ACDC2) within FullConn that, on average, improves the robustness scores of network models that contain it.

The network FullConn is an alternative view of the dynamic module approach in Verd et al. (2019), with which we showed that it is possible to model observed data using a single network functioning at different parameter regimes across spatial locations, as opposed to modeling the observed data using different networks across spatial locations. The FullConn network is a combination of the modules proposed by Verd et al., and our analysis showed that FullConn had a better robustness score than both the average random network with 4 nodes and 8 edges, as well as the average random network containing other subgraph features of interest. The FullConn network even had a higher robustness score than the StrongEdges network, which was constructed using only strong edges from the gap gene network from Fig. 1. This suggests that the modules proposed by Verd et al. (2019) are a reasonable hypothesis for dynamic control in the late-stage segmentation process of *D. melanogaster*, although it is unnecessary to view them as distinct networks. Moreover, we found that networks that are subnetworks of the proposed (large) gap gene network from Verd et al. (2019) and from Reinitz et al. (Manu et al., 2009), which is the same gap gene network but without the edge $\text{Kni} \rightarrow \text{Gt}$ (see Appendix D), have higher robustness scores, suggesting that both models contain subnetworks important to the function of the gap gene network. Furthermore, the motif ACDC2 had the most impact on our robustness score suggesting this motif may be particularly biologically relevant for robustness in the gap gene network.

We also found that nearly all of the randomly sampled networks with 4 nodes and 8 edges can reproduce the phenotype pattern derived from the developmental data between regions R_2 and R_7 . While our score allows rank-ordering these networks, it may be desirable to constrain the potential network models further. Our framework is capable of incorporating additional datasets that may help reduce the number of networks that fit the phenotype pattern. In particular, measuring expression of the gap genes in embryos where the spatial expression of Bcd and Cad was experimentally manipulated could lead to, for example, non-diagonal developmental paths that any network model would be required to match along with the wild-type data, resulting in additional restrictions on network structure. A similar process of using additional data to reduce the space of hypotheses was used in the context of DSGRN models of yeast cell cycle network (Fox et al., 2022).

Moreover, while the biologically and mathematically motivated network models FullConn and StrongEdges scored well in comparison to the random sample, there were plenty of networks that optimized the robustness score even more. We hypothesize that optimizing for robustness is a constrained optimization problem, where factors such

as evolutionary and environmental constraints may cause a network to be selected during evolution even if another network may provide more robustness for developmental or other highly conserved genetic programs.

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

There is an immediate application of our methods to insects with a similar developmental system, such as embryonic development in *Episyrphus balteatus* (Lemke et al., 2010) and *Megaselia abdita* (Wotton et al., 2015). We could also apply our work to other network models, such as the pair-rule gene network in *D. melanogaster*, where the gap gene protein concentrations determine pair-rule gene transcription (Gilbert and Barresi, 2018). Hence, in this model, the gap genes would be the external variables to the pair-rule gene network, though we would need to extend our work to non-monotone external variables. Finally, we can further extend our approach to modeling late-stage dynamic shifts in domain boundaries along the A–P axis of gap gene protein concentrations.

CRedit authorship contribution statement

Elizabeth Andreas: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. **Breschne Cummins:** Conceptualization, Formal analysis, Investigation, Methodology, Resources, Supervision, Writing – original draft, Writing – review & editing. **Tomáš Gedeon:** Conceptualization, Formal analysis, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

None

Data availability

All scripts and processed data used to produce the figures and results in this manuscript can be found at https://github.com/Eandreas1857/2023_GGN_Robustness.

Acknowledgments

We thank Dennis Mortiz (Montana State University) for his help with the statistical analysis. EA was partially supported by National Science Foundation DMS-1748883, as well as the SMART Scholarship funded by OUSD/R&E (The Under Secretary of Defense-Research and Engineering), National Defense Education Program (NDEP) / BA-1, Basic Research. TG and BC were partially supported by National Science Foundation DMS-1839299 and NIH, United States 5R01GM126555-01. The funders had no role in the study design, execution, analysis, interpretation of results, or decision to submit.

Appendix A. State transition graph (STG)

Given a regulatory network $\text{RN} = (V, E)$, the set $X = \prod_{j=1}^{|V|} X_j$, where $X_j = \{0, \dots, |T(v_j)|\}$ for $v_j \in V$, is the set of nodes of the *state transition graph* (STG). The directed edges between the nodes in STG indicate the direction of flow between neighboring (non-diagonal) domains. As we will now show, the edge directions are uniquely determined by choice of DSGRN parameter. We first define the domain target points.

Definition A.1. Given a regulatory network $\mathbf{RN} = (V, E)$ with $|V| = N$, let $\mathcal{P}$ be the associated set of DSGRN parameters. Fix a parameter $p \in \mathcal{P}$. For each domain $k \in \mathcal{K}$ (see Section 2.2.3 of main text), the function $A_j(x)$ is constant for all $x \in k$. Let $A(k) := (A_1(k), \dots, A_N(k))$ denote the vector of these values. Note that the flow in each domain k converges to a point determined by

$$\dot{v}|_k = -\Gamma v + A(k) = 0. \quad (\text{A.1})$$

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

$$TP(k) = \Gamma^{-1}A(k). \quad (\text{A.2})$$

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

We now translate the map $k \rightarrow \phi(TP(k))$, which is the map $\mathcal{K} \rightarrow X$ to a map on the space X . For $p \in \mathcal{P}$, the map $F^0 : X \times \mathcal{P} \rightarrow X$ defined by

$$F^0(x; p) := y \quad \text{where } y = \phi(TP(\phi^{-1}(x))).$$

That is, $y \in X$ is an integer signature of the domain where the target point of domain $\phi^{-1}(x)$ lies.

We are ready to define a multi-valued map F on X that gives rise to the STG.

Definition A.2. The multi-valued map $F : X \times \mathcal{P} \rightrightarrows X$ is generated by F^0 and defined by

- If $F^0(x; p) = x$ then $F(x; p) = \{x\}$.
- For any component $j = 1, \dots, N$ and $\beta \in \{-1, 1\}$ satisfying $\beta F^0(x; p) > \beta x_j$ the state

$$\bar{x}_j = x_j + \beta, \quad \bar{x}_i = x_i \text{ for } i \neq j$$

satisfies $\bar{x} \in F(x; p)$.

Note that $x \in X$ is a fixed point of F if and only if x is a fixed point of F^0 . The multivalued map F can be represented as a state transition graph (STG), see Fig. 3(e) in the main text.

As an example, we construct a STG for the regulatory network in Fig. 3(a) at a particular DSGRN parameter. Suppose $\gamma_1 = \gamma_2 = 1$ and consider the DSGRN parameter $p = (p_1, p_2)$ with $p_1 = (\xi_1, \alpha_1)$, $p_2 = (\xi_2, \alpha_2)$. Assume that order parameters are

$$\alpha_1(\theta_{1,1}) = 0, \quad \alpha_1(\theta_{2,1}) = 1$$

$$\alpha_2(\theta_{1,2}) = 0, \quad \alpha_2(\theta_{2,2}) = 1,$$

and logic parameters are

$$\xi_1 : l_{1,1} + l_{1,2} < \{u_{1,1} + l_{1,2}, l_{1,1} + u_{1,2}\} < \theta_{1,1} < \theta_{2,1} < u_{1,1} + u_{1,2}$$

$$\xi_2 : l_{2,2}l_{2,1} < u_{2,2}l_{2,1} < \theta_{1,2} < l_{2,2}u_{2,1} < \theta_{2,2} < u_{2,2}u_{2,1}.$$

This choice of parameter p determines STG in Fig. 3(e), which is superimposed on phase space. For example, consider the domain k_1 , which is the bottom left domain. All the regulatory nodes in domain k_1 are below their thresholds, thus the ordinary differential equations in this domain are

$$\dot{v}_1 = -v_1 + (l_{1,2} + u_{1,1}), \quad \dot{v}_2 = -v_2 + (l_{2,2}u_{2,1})$$

with $TP(k_1) = (l_{1,2} + l_{1,1}, l_{2,2}u_{2,1})$. Notice the choice of DSGRN parameter p implies that the value $l_{1,2} + l_{1,1} < \theta_{1,1}$, while $\theta_{1,2} < l_{2,2}u_{2,1} < \theta_{2,2}$. Therefore the target point $TP(k_1)$ is in domain k_2 . Repeating this for every domain k_j , $j = 1, \dots, 9$ we construct STG in 3(e).

Appendix B. Factor graph layers for DSGRN realizable parameters

Recall from Section 2.2.2 that a logic parameter for node v_j in a regulatory network is a function $\xi_j : R_j \rightarrow X_j$, where

$$R_j := \{(x_1, \dots, x_k) \mid x_m \in \{l_{j,m}, u_{j,m}\}\}$$

with $k = |S(v_j)|$ the number of source nodes of v_j . Further recall that an order parameter for v_j is a bijective map $\alpha_j : \Theta_j \rightarrow \{0, 1, \dots, M-1\}$, where $\Theta_j = \{\theta_{1,j}, \dots, \theta_{M,j}\}$ is the collection of thresholds for v_j and $M = |T(v_j)|$ is the number of targets of v_j . Lastly, recall that a factor graph $F_j = (V_j, E_j)$ for v_j has $M!$ isomorphic subgraphs $G_j^i = (V_j^i, E_j^i)$ where the V_j^i partition V_j , i.e., $\bigsqcup_j V_j^i = V_j$ (Cummins et al., 2016). Each of these subgraphs G_j^i , called **subfactor graphs**, is associated with a particular threshold order α_j^i . Each subfactor graph G_j^i has unique lowest parameter node $\ell_{i,j}$ given by

$$\ell_{i,j} = (\xi_j^0, \alpha_j^i) \quad \text{where } \xi_j^0(x) = 0 \text{ for all } x \in R_j$$

unique highest parameter

$$h_{i,j} = (\xi_j^H, \alpha_j^i) \quad \text{where } \xi_j^H(x) = M \text{ for all } x \in R_j.$$

Definition B.1. A factor parameter node $p = (\xi_j, \alpha_j^i)$ is **DSGRN realizable** if there exist sets of real, positive values $\{\{l_{j,m}, u_{j,m}\}_{m=1}^k\}$ and Θ_j , where the elements of Θ_j are all distinct, and a function $g : \mathbb{R}^k \rightarrow \mathbb{R}$ which has the form of product of sums

$$g(x_1, \dots, x_k) = \prod \sum x_m, \quad (\text{B.1})$$

such that for all $x \in R_j$ and all $\theta_{n,j}$

$$g(x) < \theta_{n,j} \quad \text{if and only if } \xi(x) < \alpha_j^i(\theta_{n,j}) + 1$$

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

$$w := \{\{l_{j,m}, u_{j,m}\}_{m=1}^k\} \cup \Theta_j$$

a **witness** of the parameter node p under function g .

Remark B.1. Note that the nodes $\ell_{i,j}$ and $h_{i,j}$ are realizable for any function g of the form in (B.1). To see this, choose an arbitrary set of real, positive values $U := \{\{l_{j,m}, u_{j,m}\}_{m=1}^k\}$ and set

$$\hat{m} := \min\{g(x) \mid x \in R_j\} \quad \text{and} \quad \hat{M} := \max\{g(x) \mid x \in R_j\}.$$

Note that $0 < \hat{m} < \hat{M}$. Then if we select a set $\Theta_j(w)$ with $\max \Theta_j(w) < \hat{m}$ then $U \cup \Theta_j$ is witness for $h_{j,i}$ and if we select $\Theta_j(w)$ with $\min \Theta_j(w) > \hat{M}$ then $U \cup \Theta_j(w)$ is witness for $\ell_{j,i}$.

We denote the set of 2^k real-valued inputs evaluated on the witness w by $R_j(w)$ and the set of threshold values in witness w by $\Theta_j(w)$ and let

$$Y(w) := \{g(x) \mid x \in R_j(w)\},$$

where repeated elements are permitted.

Lemma B.1. Given parameter node p and function g , for a generic choice of w , the set $Y(w)$ is totally ordered. That is, there exists an open and dense set $U \subset V$ where V is an open subset of $\mathbb{R}^{|w|}$ of those values that satisfy

1. $0 < l_{j,m} < u_{j,m}$ (i.e., (5)),
2. distinct thresholds in Θ_j ,
3. $Y(w) \cap \Theta_j = \emptyset$, and
4. the inequality constraints of the parameter node p ,

such that $w \in U$ implies all values of $Y(w)$ are distinct.

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

Suppose for a witness w for parameter p , there are two values $x \neq x' \in R_j(w)$ such that $g(x) = g(x') \in Y(w)$. Choose a position m such that $x_m \neq x'_m$, and assume without loss of generality that $x_m = l_{j,m}$.

Define $\tilde{w}$ as a witness under g of some parameter node q by taking the witness w and changing exactly one value: $\tilde{l}_{j,m} = l_{j,m} + \epsilon$ for some $\epsilon > 0$. In particular, ϵ must be small enough to ensure $l_{j,m} + \epsilon < u_{j,m}$ and

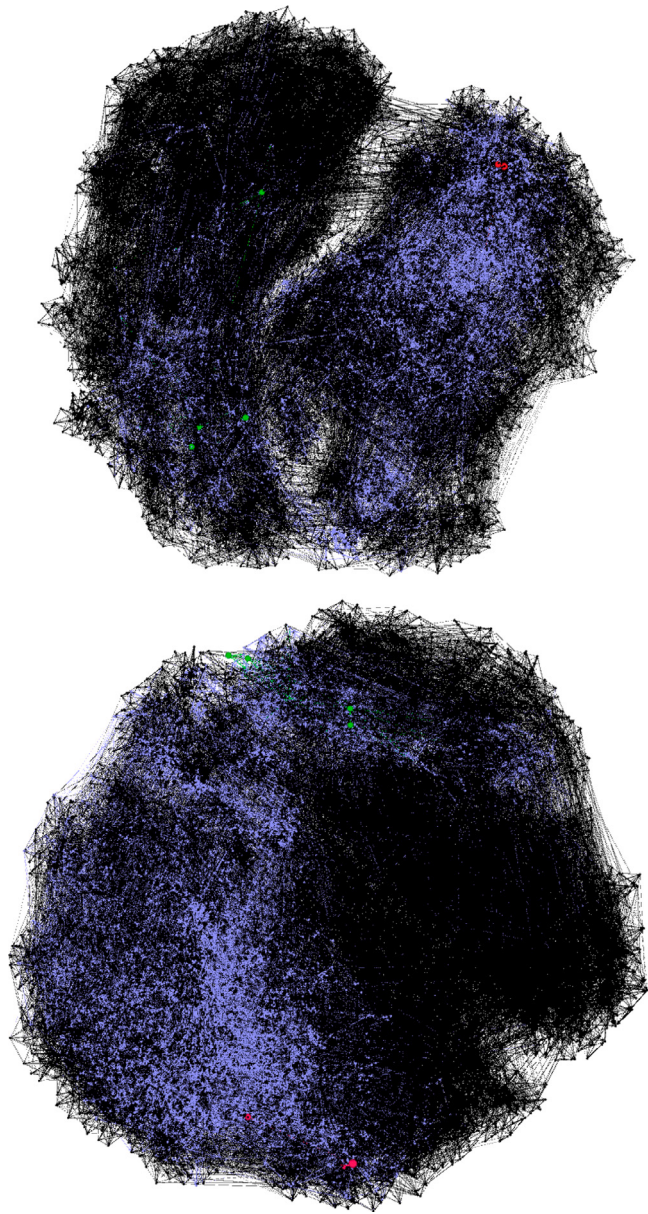


Fig. C.1. Condensed chemical gradient graph (all nodes) for FullConn (top) and StrongEdges (bottom). FullConn has 12,398 and 100,179 nodes and edges respectively. StrongEdges has 14,832 and 123,407 nodes and edges respectively. Purple, red, and green nodes depict nodes in the path graph, starting nodes are shown in green, and stopping nodes are shown in red. Graph visualization done using Gephi (Bastian et al., 2009) with the OpenOrd layout algorithm (Martin et al., 2011).

$Y(\bar{w}) \cap \Theta_j(\bar{w}) = \emptyset$. The latter can be accomplished since $Y(\bar{w}) \cup \Theta_j(\bar{w})$ has a finite number of points. Notice then that $g(\bar{x}) \neq g(x') = g(\bar{x}')$, where $\bar{x} = x + \epsilon e_m$, and e_m is the unit vector in the m^{th} direction. Note that these conditions remain true for any $0 < \delta < \epsilon$.

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

We must also avoid introducing new equalities, i.e. we additionally require $g(\bar{y}) \neq g(\bar{z})$ whenever $g(y) \neq g(z)$ for $y, z \in R_j(w)$. Since g is a continuous function and $g(y)$ and $g(z)$ are isolated, taking ϵ

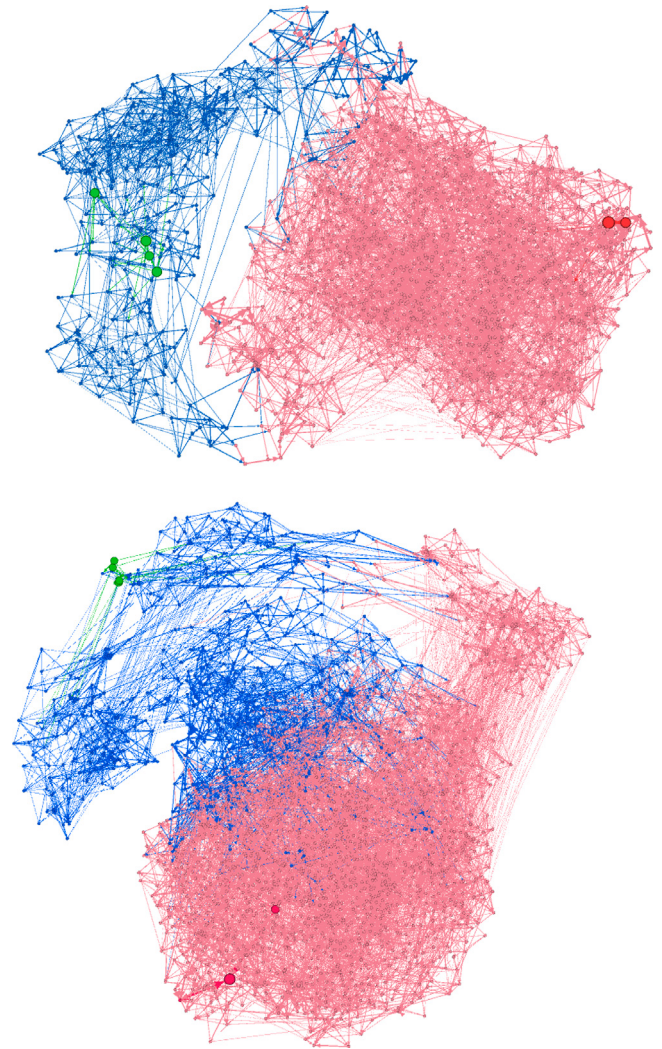


Fig. C.2. Path graph (all nodes) for FullConn (top) and StrongEdges (bottom). Colored by optimum 2-clustering separating starting nodes (green) and stopping nodes (red). FullConn has 1576 and 5767 nodes and edges respectively. StrongEdges has 2393 and 10,037 nodes and edges respectively. Note that visually one might say that FullConn has a larger bottleneck than StrongEdges which we found not to be the case. This is due to the lack of visualization of edge weights, as well as a two-dimensional graph layout of a multi-dimensional graph. Recall any path from green to red nodes is a matching developmental path. Graph visualization done using Gephi (Bastian et al., 2009) with the OpenOrd layout algorithm (Martin et al., 2011).

sufficiently small ensures that for each such pair y, z , it remains true that $g(\bar{y}) \neq g(\bar{z})$.

After all the adjustments to ϵ have been made, the new witness $\bar{w}$ for parameter p now ensures that $g(\bar{x}) \neq g(\bar{x}')$ without introducing any new duplicates in $Y(\bar{w})$. However, there may be other pairs $y \neq y' \in R_j(\bar{w})$ that satisfy $g(y) = g(y')$. Since there are at most a finite number, the procedure above may be repeated on $\bar{w}$ until some final witness $\hat{w}$ for p under g is constructed such that all elements of $Y(\hat{w})$ are distinct. Since at each step, the corresponding ϵ may be taken arbitrarily small, it is true that given any witness w , there is another witness $\hat{w}$ arbitrarily close to w where $Y(\hat{w})$ is totally ordered. This proves that the property of total ordering of $Y(\hat{w})$ is dense in U .

Since g is continuous, there is an open neighborhood of witness w in $R^{|w|}$ whenever $Y(w)$ is totally ordered, since $Y(w)$ has a finite number of isolated values. Call the neighborhood $N_\delta(w)$. Then under the subspace topology, $V \cap N_\delta(w) \subset U$ is relatively open in V . Since U is covered by $\bigcup_w V \cap N_\delta(w)$, U is open in V . ■

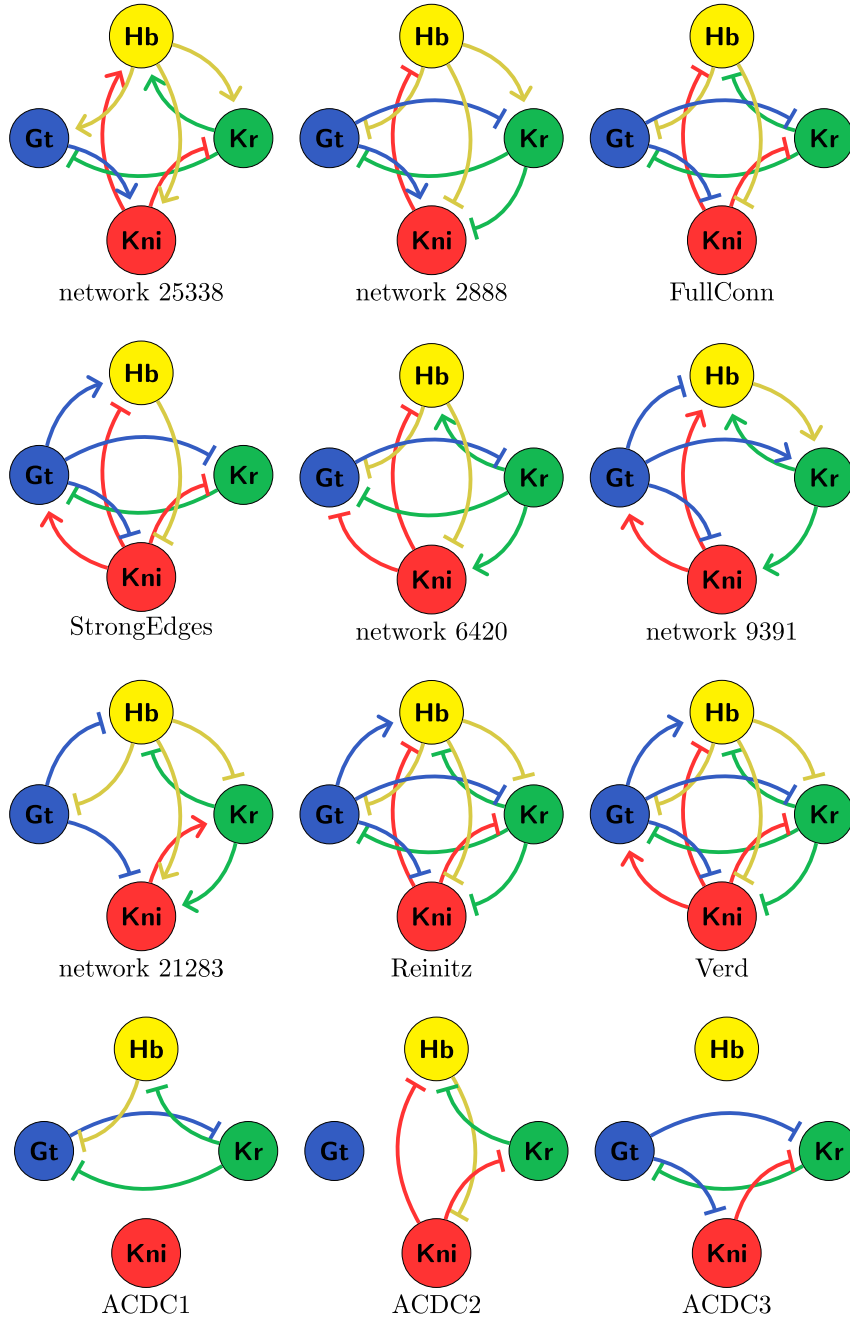


Fig. D.1. First 6 networks are ordered from highest to lowest score, with networks 25,338 and 2888 being the best scoring networks for the Baseline and Feature group respectively and networks 6420 and 9391 being the worst scoring networks for the Feature and Baseline group respectively. Network 21,283 is the only network that had an empty stopping set S . Next are the gap gene networks as described in Reinitz et al. (Manu et al., 2009) and Verd et al. (2019) respectively. Lastly are the ACDC submodules derived by Verd et al. (2019).

Definition B.2. A DSGRN realizable sub-factor graph $\hat{G}_j^i = (\hat{V}_j^i, \hat{E}_j^i)$ under g is a node induced subgraph of G_j^i , where the collection of nodes $\hat{V}_j^i \subset V_j^i$ are those nodes that have a witnesses under g . A DSGRN realizable factor graph $\hat{F}_j$ under g is the product

$$\hat{F}_j = \prod_i \hat{G}_j^i.$$

Lemma B.2. Assume $p = (\xi_j, \alpha_j^i) \in \hat{G}_j^i$, $p \neq \ell_{i,j}$, and let w be a witness of p under function g . Then there exists a path from p to $\ell_{i,j}$ within $\hat{G}_j^i$. Likewise, there exists a path from p to $h_{i,j}$ within $\hat{G}_j^i$.

Proof. Assume without loss of generality that the witness w induces a totally ordered set $Y(w)$, see Lemma B.1. Define the sets

$$\begin{aligned} Q_0 &= \{u \in R_j(w) \mid \xi_j(u) = 0\} \\ Q_1 &= \{u \in R_j(w) \mid \xi_j(u) = 1\} \\ &\vdots \\ Q_M &= \{u \in R_j(w) \mid \xi_j(u) = M\}, \end{aligned}$$

where recall $M = |T(v_j)|$. Since $p \neq \ell_{i,j}$, there exists at least one nonempty Q_n with $n > 0$. Since Q_n is a finite set, it has a smallest element $r_0 := \xi_j(u_0)$. Let $r_1 \in Q_n$ be the smallest element in $Q_n \setminus \{r_0\}$, if it exists, and let $r_1 = (\alpha_j^i)^{-1}(n)$, if such a smallest element does not

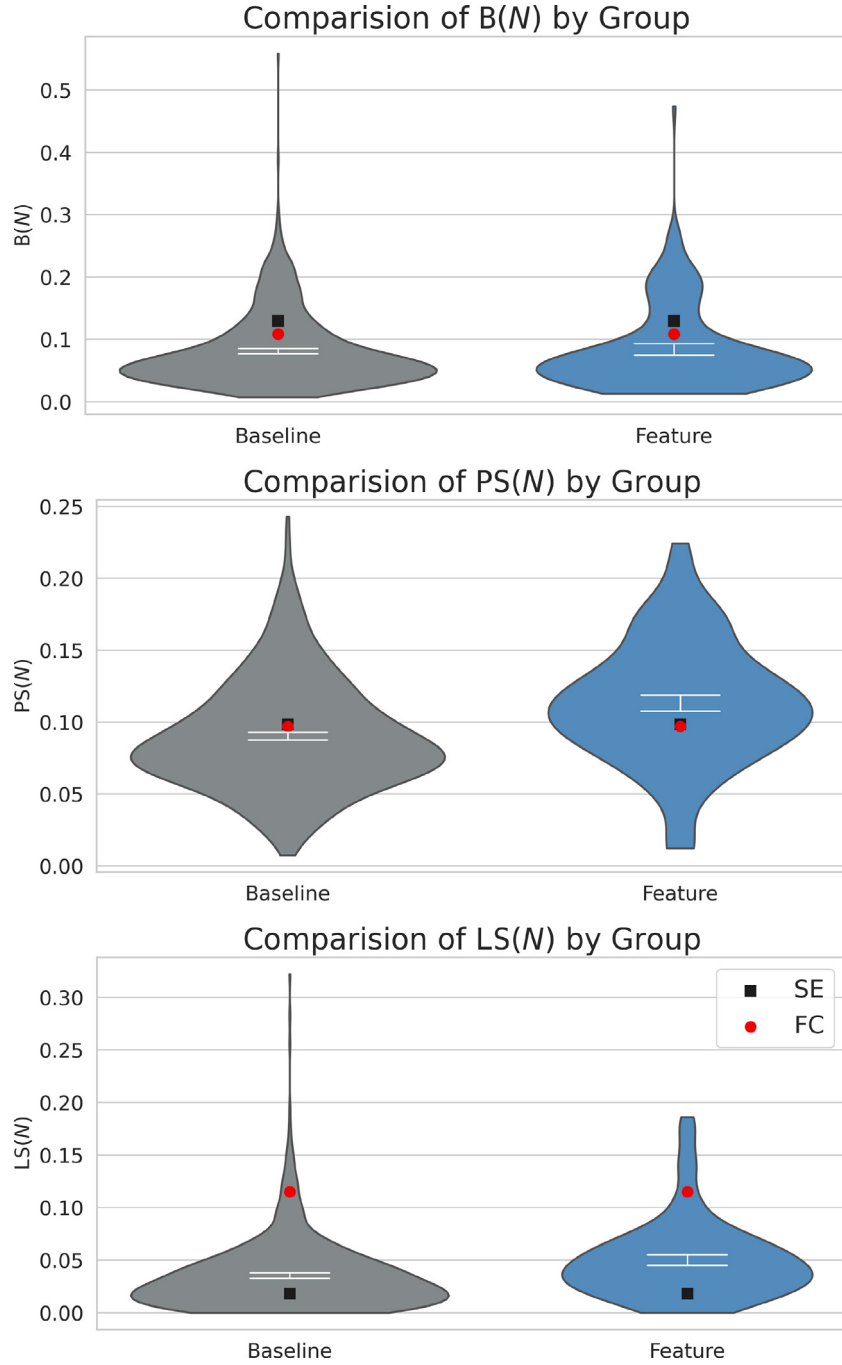


Fig. E.1. Score results for each network in the baseline and features group before applying the normalization detailed in Section 6.4. Notice that all but one network have $B(N) < 0.5$ indicating a bottleneck existence per Theorem 4.

exist. Let $\Delta = r_1 - r_0$ and let

$$\theta'_{n-1,j} := r_0 + \frac{\Delta}{2}$$

We define a new witness where we replace $\theta_{n-1,j}$ by the new value of the threshold $\theta'_{n-1,j}$

$$w' = \{\{l_{j,m}, u_{j,m}\}_{m=1}^k\} \cup \Theta'_j$$

with $\Theta'_j = (\Theta_j \setminus \{\theta_{n-1,j}\}) \cup \{\theta'_{n-1,j}\}$. Then w' is a witness for the parameter q where the value of $\xi_j(u_0)$ changes from n to value $n-1$. Therefore q is a node in $\hat{G}_j^i$ that is the immediate neighbor of node p and therefore there is an edge between p and q . Repeating this argument it is easy to see that eventually only the set $\mathcal{Q}_0$ is non-empty, which occurs only at

the node $\ell_{i,j}$. Therefore, every node p in the subfactor graph has a path to $\ell_{i,j}$.

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

Corollary B.1. Every DSGRN realizable subfactor graph $\hat{G}_j^i$ is connected.

Proof. By Remark B.1 every realizable subfactor graph $\hat{G}_j^i$ contains both $\ell_{i,j}$ and $h_{i,j}$. Then every node $p \neq \ell_{i,j}$, including $h_{i,j}$, in the $\hat{G}_j^i$ is connected to $\ell_{i,j}$ by Lemma B.2. ■

Table F.1Mean, standard deviation, and confidence intervals for the random samples B and B_F and the specific networks FullConn and StrongEdges.

	Baseline Group (B) Results				Feature Group (B_F) Results				FullConn	StrongEdges
	Mean	Std dev	Mean 95% CI		Mean	Std dev	Mean 95% CI		Results	Results
$\hat{B}(N)$	0.134	0.106	[0.127,	0.142]	0.139	0.120	[0.121,	0.156]	0.184	0.222
$\hat{P}\hat{S}(N)$	0.352	0.162	[0.340,	0.364]	0.449	0.178	[0.424,	0.475]	0.381	0.387
$\hat{L}\hat{S}(N)$	0.109	0.107	[0.102,	0.117]	0.155	0.116	[0.139,	0.172]	0.357	0.056
$S(N)$	0.199	0.078	[0.193,	0.204]	0.248	0.087	[0.235,	0.260]	0.307	0.222

Table F.2The regression coefficients, standard errors, t-values, p-values lower and upper confidence intervals for our MLR model for the bottleneck score ($\hat{B}(N)$), path size score ($\hat{E}(N)$), leak-skip score ($\hat{L}\hat{S}(N)$) and the robustness score ($S(N)$).

$\hat{B}(N)$	Coef	Std dev	t-value	p-value	95% ci	$\hat{P}\hat{S}(N)$	Coef	Std dev	t-value	p-value	95% ci
intercept	0.1542	0.0217	7.1	< 0.0001	[0.1116, 0.1969]	intercept	0.4358	0.03	14.5109	< 0.0001	[0.3768, 0.4947]
ACDC3	-0.0091	0.0167	-0.5416	0.5882	[-0.0419, 0.0238]	ACDC3	-0.0469	0.0232	-2.0243	0.0432	[-0.0923, -0.0014]
ACDC2	0.0497	0.0188	2.6433	0.0084	[0.0128, 0.0866]	ACDC2	0.2104	0.026	8.0959	< 0.0001	[0.1594, 0.2614]
ACDC1	-0.0115	0.0167	-0.6904	0.4901	[-0.0442, 0.0212]	ACDC1	0.0306	0.023	1.3308	0.1836	[-0.0145, 0.0758]
Ultra Strong	-0.0125	0.0137	-0.9152	0.3603	[-0.0393, 0.0143]	Ultra Strong	-0.0129	0.0189	-0.6841	0.4941	[-0.05, 0.0241]
Strict Verd	0.0923	0.0263	3.508	0.0005	[0.0407, 0.1439]	Strict Verd	0.0982	0.0364	2.7011	0.007	[0.0269, 0.1696]
Reinitz	0.0298	0.0293	1.0178	0.309	[-0.0277, 0.0872]	Reinitz	0.153	0.0405	3.7818	0.0002	[0.0736, 0.2325]
NFL	0.0007	0.0034	0.2073	0.8359	[-0.0061, 0.0075]	NFL	-0.0451	0.0048	-9.4607	< 0.0001	[-0.0544, -0.0357]
PFL	-0.0089	0.0035	-2.5902	0.0097	[-0.0157, -0.0022]	PFL	-0.005	0.0048	-1.0543	0.292	[-0.0144, 0.0043]
RE	0.0005	0.0028	0.1609	0.8722	[-0.0051, 0.006]	RE	0.014	0.0039	3.6062	0.0003	[0.0064, 0.0216]
$\hat{L}\hat{S}(N)$	Coef	Std dev	t-value	p-value	95% ci	$S(N)$	Coef	Std dev	t-value	p-value	95% ci
intercept	0.0782	0.0219	3.5723	0.0004	[0.0352, 0.1211]	intercept	0.2227	0.0152	14.616	< 0.0001	[0.1928, 0.2526]
ACDC3	0.039	0.0169	2.311	0.0211	[0.0059, 0.0721]	ACDC3	-0.0056	0.0117	-0.4808	0.6308	[-0.0287, 0.0174]
ACDC2	0.0365	0.0189	1.9277	0.0542	[-0.0007, 0.0737]	ACDC2	0.0989	0.0132	7.497	< 0.0001	[0.073, 0.1248]
ACDC1	0.0327	0.0168	1.9507	0.0514	[-0.0002, 0.0657]	ACDC1	0.0173	0.0117	1.48	0.1392	[-0.0056, 0.0402]
Ultra Strong	0.0423	0.0138	3.0714	0.0022	[0.0153, 0.0692]	Ultra Strong	0.0056	0.0096	0.5861	0.558	[-0.0132, 0.0244]
Strict Verd	-0.0254	0.0265	-0.9589	0.3378	[-0.0774, 0.0266]	Strict Verd	0.055	0.0185	2.9821	0.0029	[0.0188, 0.0912]
Reinitz	0.0364	0.0295	1.2341	0.2175	[-0.0215, 0.0943]	Reinitz	0.0731	0.0205	3.5587	0.0004	[0.0328, 0.1134]
NFL	0.0018	0.0035	0.5261	0.5989	[-0.005, 0.0086]	NFL	-0.0142	0.0024	-5.8644	< 0.0001	[-0.0189, -0.0094]
PFL	0.0104	0.0035	2.9961	0.0028	[0.0036, 0.0172]	PFL	-0.0012	0.0024	-0.489	0.625	[-0.0059, 0.0036]
RE	-0.0004	0.0028	-0.1275	0.8986	[-0.0059, 0.0052]	RE	0.0047	0.002	2.3844	0.0173	[0.0008, 0.0086]

Table F.3

Each explanatory variable VIF for the MLR model.

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

Corollary B.2. Every DSGRN realizable factor graph $\hat{F}_j$ is connected.

Proof. Recall that $\ell_{i,j}, h_{i,j} \in \hat{G}_j^i$ for any subfactor graph $\hat{G}_j^i$. Consider a subfactor graph $\hat{G}_j^k$ such that α_j^i and α_j^k are identical except for two adjacent thresholds. That is, $\alpha_j^i(\theta_{n,j}) = \alpha_j^k(\theta_{m,j}) + 1$ and $\alpha_j^k(\theta_{n,j}) = \alpha_j^i(\theta_{m,j}) + 1$, but $\alpha_j^i(\theta_{s,j}) = \alpha_j^k(\theta_{s,j})$ otherwise. Then the nodes $\ell_{i,j}$ and $\ell_{k,j}$ are connected in F_j , and therefore they are connected in $\hat{F}_j$.

Since there is a sequence of such adjacent swaps that connects any two permutations of Θ_j , the set of nodes $\{\ell_{s,i}\}_{s=1}^m$ is connected in $\hat{F}_j$. Since each $\hat{G}_j^i$ is connected, $\hat{F}_j$ is connected. ■

Finally, note that $\hat{F}_j$ inherits the structure of graded poset from F_j .

Appendix C. Graph visualization

See Figs. C.1 and C.2.

Appendix D. Network topology

See Fig. D.1.

Appendix E. Network measure results before normalization

See Fig. E.1.

Appendix F. Model results and MLR validity

Results of the difference in means model used to assess if there is a difference between the average network in the baseline group versus the average network in the features group can be seen in Table F.1.

We provide the data of the multilinear regression (MLR) used in Section 7 (see Table F.2) as well as verification of all assumptions needed to use the MLR model.

First, checking multicollinearity between our explanatory variables, we consider the variance inflation factors (VIFs) (Greenwood, 2022), which are shown in Table F.3. Since none of the VIFs are greater than 5, and in fact small (close to 1), we have evidence that multicollinearity between our explanatory variables is not a problem (Greenwood, 2022).

We see no clear pattern in the Residuals vs Fitted plot, showing it is reasonable to assume linearity of relationships (Greenwood, 2022). The Scale-Location plot shows weak to moderate evidence against equal variance, as indicated by having higher variance for the middling fitted values (Greenwood, 2022). In general, our normal QQ-plot is showing a deviation from the line of normality, though only a slight right-skew. Hence, we see no indication of a violation of the normality assumption. Lastly, our average leverage is approximately 0.01, meaning all points with leverage greater than 0.02 have high leverage. However, since no points have a Cook's distance greater than 0.5 then we can conclude no points are overly influential to our model (Greenwood, 2022) (see Fig. F.1).

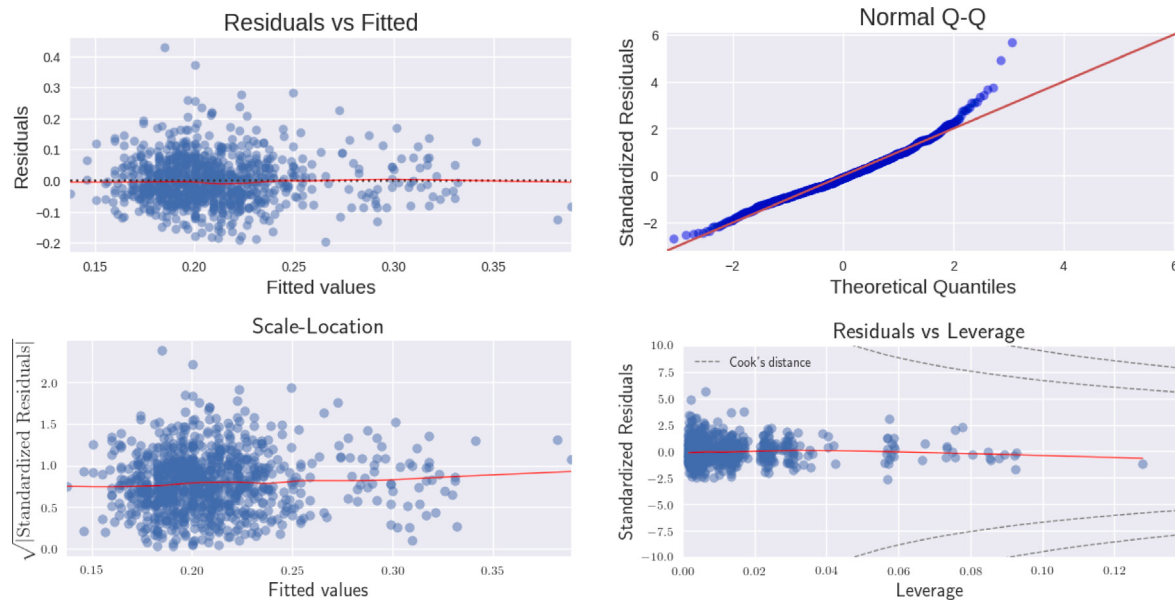


Fig. F.1. MLR diagnostic plots.

References

- Ashyraliyev, Maksat, Siggins, Ken, Janssens, Hilde, Blom, Joke, Akam, Michael, Jaeger, Johannes, 2009. Gene circuit analysis of the terminal gap gene *huckebein*. *PLoS Comput. Biol.* 5 (10).
- Bastian, Mathieu, Heymann, Sebastien, Jacomy, Mathieu, 2009. Gephi: An open source software for exploring and manipulating networks.
- Crawford-Kahrl, Peter, Cummins, Bree, Gedeon, Tomáš, 2022. Joint realizability of monotone Boolean functions. *Theoret. Comput. Sci.* 922, 447–474.
- Cummins, Bree, Gedeon, Tomas, Harker, Shaun, Mischaikow, Konstantin, 2018. DSGRN: Examining the dynamics of families of logical models. *Front. Physiol.* 9, 549.
- Cummins, Bree, Gedeon, Tomas, Harker, Shaun, Mischaikow, Konstantin, Mok, Kafung, 2016. Combinatorial representation of parameter space for switching networks. *SIAM J. Appl. Dyn. Syst.* 15 (4), 2176–2212.
- de Jong, H., Gouze, J.L., Hernandez, C., Page, M., Sari, T., Geiselmann, J., 2004. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull. Math. Biol.* 66 (2), 301–340.
- Ermon, Stefano, Gomes, Carla, Sabharwal, Ashish, Selman, Bart, 2014. Designing fast absorbing Markov chains. In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. AAAI '14*, AAAI Press, pp. 849–855.
- Fox, Erika, Cummins, Bree, Duncan, William, Gedeon, Tomáš, 2022. Modeling transport regulation in gene regulatory networks. *Bull. Math. Biol.* 84 (8), 89.
- Gameiro, Marcio, 2018. Dynamic signatures generated by regulatory networks. Version 1.1.0. <https://github.com/marciogameiro/DSGRN>.
- Gedeon, Tomáš, Cummins, Bree, Harker, Shaun, Mischaikow, Konstantin, 2018. Identifying robust hysteresis in networks. *PLoS Comput. Biol.* 14 (4), 1–23.
- Gilbert, Scott, Barresi, Michael, 2018. *Developmental Biology*, seventh ed. Oxford University Press.
- Glass, L., Kauffman, S.a., 1972. Co-operative components, spatial localization and oscillatory cellular dynamics. *J. Theoret. Biol.* 34 (2), 219–237.
- Glass, L., Kauffman, S.a., 1973. The logical analysis of continuous, non-linear biochemical control networks. *J. Theoret. Biol.* 39 (1), 103–129.
- Greenwood, Mark, 2022. *Intermediate Statistics with R*. 3.1, Montana State University.
- Irish, Vivian, Lehmann, Ruth, Akam, Michael, 1989. The drosophila posterior group gene nanos functions by repressing hunchback activity. *Nature* 338, 646–648.
- Jaeger, Johannes, 2011. The gap gene network. *Cellular and Molecular Life Sciences* 68 (2), 243–274.
- Jaeger, Johannes, Blagov, Maxim, Kosman, David, Kozlov, Konstantin, Manu, Myasnikova, Ekaterina, Surkova, Svetlana, Vanario-Alonso, Carlos, Samsonova, Maria, Sharp, David, Reinitz, John, 2004. Dynamical analysis of regulatory interactions in the gap gene system of *Drosophila melanogaster*. *Genetics* 167 (4), 1721–1737.
- Klarner, David, 1969. The number of graded partially ordered sets. *J. Combin. Theory* 6 (1), 12–19.
- Lemke, Steffen, Busch, Stephanie, Antonopoulos, Dionysios, Meyer, Folker, Domanus, Marc, Schmidt-Ott, Urs, 2010. Maternal activation of gap genes in the hover fly *episyrphus*. *Development* 137 (15), 2604.
- Manu, Surkova, Svetlana, Spirov, Alexander, Gursky, Vitaly, Janssens, Hilde, Kim, Ah-Ram, Radulescu, Ovidiu, Vanario-Alonso, Carlos, Sharp, David, Samsonova, Maria, Reinitz, John, 2009. Canalization of gene expression and domain shifts in the drosophila blastoderm by dynamical attractors. *PLoS Comput. Biol.* 5 (3), 1–15.
- Martin, Shawn, Brown, W. Michael, Klavans, Richard, Boyack, Kevin, 2011. OpenOrd: an open-source toolbox for large graph layout. In: *Electronic Imaging*.
- Meilă, Marina, Pentney, William, 2007. Clustering by weighted cuts in directed graphs. *SIAM Int. Conf. Data Min.*
- Meinhardt, Hans, 1986. Hierarchical inductions of cell states: a model for segmentation in drosophila. *J. Cell Sci. (Suppl.)* 4, 357–381.
- Mischaikow, Konstantin, 2002. Topological techniques for efficient rigorous computation in dynamics. *Acta Numer.* 2002 435–478.
- Nüsslein-Volhard, Christiane, Frohnhofer, Hans Georg, Lehmann, Ruth, 1987. Determination of anteroposterior polarity in drosophila. *Science* 238, 1675–1687.
- Nüsslein-Volhard, Christiane, Wieschaus, Eric, 1980. Mutations affecting segment number and polarity in drosophila. *Nature* 287 (5785), 795–801.
- Panovska-Griffiths, Jasmina, Page, Karen, Briscoe, James, 2013. A gene regulatory motif that generates oscillatory or multiway switch outputs. *J. R. Soc. Interface* 10 (79), 20120826.
- Perkins, Theodore, Jaeger, Johannes, Reinitz, John, Glass, Leon, 2006. Reverse engineering the gap gene network. *PLoS Comput. Biol.* 2:e51.
- Privault, Nicolas, 2018. *Understanding Markov Chains: Examples and Applications*, second ed. In: *Mathematics Series*, Springer.
- Shi, Jianbo, Malik, Jitendra, 2000. Normalized cuts and image segmentation. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
- Simpson-Brose, Marcia, Treisman, Jessica, Desplan, Claude, 1994. Synergy between the hunchback and bicoid morphogens is required for anterior patterning in drosophila. *Cell* 855–865.
- Snoussi, El Houssine, 1989. Qualitative dynamics of piecewise-linear differential equations: a discrete mapping approach. *Dyn. Stab. Syst.* 4 (3–4), 565–583.
- Spirov, Alexander, Fahmy, Khalid, Schneider, Martina, Frei, Erich, Noll, Markus, Baumgartner, Stephan, 2009. Formation of the bicoid morphogen gradient: an mRNA gradient dictates the protein gradient. *Development (Cambridge, England)* 136 (4), 605–614.
- Stanley, Richard, 2012. *Enumerative Combinatorics*, second ed. In: *Mathematics Series*, Cambridge Univ. Press, Cambridge.
- Thieffry, D., Thomas, R., 1998. Qualitative analysis of gene networks. *Pac. Symp. Biocomput.* 77–88.
- Thomas, R., 1973. Boolean formalization of genetic control circuits. *J. Theoret. Biol.* 42, 563–585.
- Thomas, R., 1991. Regulatory networks seen as asynchronous automata: A logical description. *J. Theoret. Biol.* 153, 1–23.
- Thompson, William, McNeal, James, 1967. Sales planning and control using absorbing Markov chains. *J. Mar. Res.* 4 (1), 62–66.
- Verd, Berta, Clark, Erik, Wotton, Karl, Janssens, Hilde, Jiménez-Guri, Eva, Crombach, Anton, Jaeger, Johannes, 2018. A damped oscillator imposes temporal order on posterior gap gene expression in *Drosophila*. *PLoS Biol.* 16 (2), 1–24.

- Verd, Berta, Crombach, Anton, Jaeger, Johannes, 2017. Dynamic maternal gradients control timing and shift-rates for *Drosophila* gap gene expression. *PLoS Comput. Biol.* 13 (2).
- Verd, Berta, Monk, Nicholas, Jaeger, Johannes, 2019. Modularity, criticality, and evolvability of a developmental gene regulatory network. *eLife* 8 (2), 243–274.
- Wang, Charlotte, Dickinson, Laura, Lehmann, Ruth, 1994. Genetics of nanos localization in *drosophila*. *Dev. Dyn.* 199 (2), 103–115.
- Wotton, Karl R., Jiménez-Guri, Eva, Crombach, Anton, Janssens, Hilde, Alcaine-Colet, Anna, Lemke, Steffen, Schmidt-Ott, Urs, Jaeger, Johannes, 2015. Quantitative system drift compensates for altered maternal inputs to the gap gene network of the scuttle fly *Megaselia abdita*. In: Barkai, Naama (Ed.), *eLife* 4, e04785.