Extending Combinatorial Regulatory Network Modeling to Include Activity Control and Decay Modulation

Cummins, Bree; Gameiro, Marcio; Gedeon, Tomas; Kepley, Shane; Mischaikow, Konstantin; Zhang, Lun

published in
SIAM Journal on Applied Dynamical Systems
2022

DOI (link to publisher)
10.1137/21M1456832

document version
Publisher's PDF, also known as Version of record

SIAM J. Applied Dynamical Systems
Vol. 21, No. 3, pp. 2096–2125

Extending Combinatorial Regulatory Network Modeling to Include Activity Control and Decay Modulation*

Bree Cummins†, Marcio Gameiro‡, Tomas Gedeon†, Shane Kepley§, Konstantin Mischaikow¶, and Lun Zhang¶

Abstract. Understanding how the structure of within-system interactions affects the dynamics of the system is important in many areas of science. We extend a network dynamics modeling platform DSGRN, which combinatorializes both dynamics and parameter space to construct finite but accurate summaries of network dynamics, to new types of interactions. While the standard DSGRN assumes that each network edge controls the rate of abundance of the target node, the new edges may control either activity level or a decay rate of its target. While motivated by processes of post-transcriptional modification and ubiquitination in systems biology, our extension is applicable to the dynamics of any signed directed network.

Key words. network dynamics, gene regulation, mathematical biology

MSC codes. 92C42, 37N25, 37B35, 37C25

DOI. 10.1137/21M1456832

1. Introduction.

Networks have become a paradigm for organizing relational information: each node is associated with a particular object or quantity and edges indicate a relation between these objects or quantities. In many cases these relations are meant to capture causality, e.g., a directed edge from node m to node n indicates that the product associated with node m has an impact on the product associated with node n. We refer to such a network as a *regulatory network*. Our main goal is to identify possible dynamics of a given regulatory network. In order to better specify the problem and make it relevant for applications the following challenges need to be addressed.

*Received by the editors November 2, 2021; accepted for publication (in revised form) by T. Wanner May 3, 2022; published electronically August 11, 2022.
https://doi.org/10.1137/21M1456832
Funding: The work of the first and third authors was partially supported by NSF grant DMS-1839299, DARPA FA8750-17-C-0054, and NIH 5R01GM126555-01. The work of the second, fourth, fifth, and sixth authors was partially supported by NSF grant DMS-1839294 and HDR TRIPODS award CCF-1934924, DARPA contract HR0011-16-2-0033, and National Institutes of Health award R01 GM126555. The work of the third author was also partially supported by FAPESP grant 2019/06249-7 and by CNPq grant 309073/2019-7. The work of the fifth author was also supported by a grant from the Simons Foundation.
†Department of Mathematical Sciences, Montana State University, Bozeman, MT 59717 USA (breschine.cummins@gmail.com, gedeon@math.montana.edu).
‡Department of Mathematics, Rutgers University, Piscataway, NJ 08854 USA, and Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, São Paulo, Brazil (gameiro@math.rutgers.edu).
§Department of Mathematics, VU Amsterdam, 1081 HV Amsterdam, The Netherlands (s.kepley@vu.nl).
¶Department of Mathematics, Rutgers University, Piscataway, NJ 08854 USA (mischaik@math.rutgers.edu, lz210@rutgers.edu).

C1. The network structure must encode a sufficiently broad range of meaningful causal interactions so that the range of dynamics relevant to applications can be realized.
C2. The computational framework should take the network structure as input and output an identification and characterization of global dynamics.
C3. There needs to be a theoretical framework that ties the outputs of the computations back to the dynamics of the application of interest.

While the focus of this paper is on C2, the motivation for this work is the analysis of regulatory networks arising from systems biology. Thus, we will partially address C1 in the context of gene regulatory networks that include post-transcriptional modifications such as phosphorylation and ubiquitination. The theoretical validation of our approach, i.e., C3, is based on previous and ongoing work [20, 5, 14, 11, 13] and is discussed at relevant points in the paper.

We make two assumptions that are maintained throughout the paper.

A1. An ordinary differential equation (ODE) provides an adequate model for the dynamics.
A2. Let x_n denote the quantity of product associated with node n. Then, the rate of change of x_n can be expressed as

$$-\Gamma_n(x)x_n + \Lambda_n(x), \tag{1.1}$$

where $\Gamma_n(x)$ and $\Lambda_n(x)$ quantify the rate of decay and the rate of production of x_n, respectively. The network encodes the coordinates of the state x upon which Γ_n and Λ_n are dependent, but we do not assume a particular functional form for this dependency.

Observe that we have refrained from writing (1.1) in the form of an ODE. This is to emphasize the fact that we use explicit functions for Γ and Λ only for computational purposes, but we are not interested in the dynamics in the traditional sense, e.g., trajectories or equilibria of the resulting differential equations. Instead, the appropriate interpretation of the dynamics obtained via our computations should be derived indirectly from associated lattice structures and algebraic topology, which is part of C3.

Our approach to C2 is an extension to an earlier approach based on a combinatorial representation of the dynamics [5]. At its foundation lies the perspective that because we do not know the precise nonlinearities we should not try to identify dynamics on the level of trajectories. Instead the goal is to provide a computationally efficient robust combinatorial representation of the dynamics that, at a minimum, is capable of accurately identifying existence and structure of attractors. Furthermore, since the dynamics that can be exhibited by a network is parameter dependent, it is desirable that there is a clear correspondence between parameters and global dynamics. With this in mind we developed the Dynamic Signatures Generated by Regulatory Networks (DSGRN) software [5, 4, 13] that takes a network as input, creates an appropriate parameter space along with an explicit finite decomposition thereof, and for each region of parameter space computes a combinatorial/algebraic topological description of the global dynamics (see section 2 for further details).

This approach has been applied to a variety of regulatory networks associated with questions and challenges from systems and synthetic biology including identification of oscillatory behavior in a simple model of the p53 network [5], identification of minimal models for the switching behavior of the mammalian Rb-E2F system [14], EMT [27], oocyte [7], and design of optimal 3 node hysteretic switches [11]. This variety of applications suggests that DSGRN

is a potentially powerful tool for the global analysis of networks. However, in the above mentioned biological contexts the current version of DSGRN imposes two significant constraints. The first is that the decay rate Γ_n is assumed to be constant, i.e., not controlled by other nodes (representing protein concentrations) within the network. However, the fact that the 2004 Nobel Prize in Chemistry was awarded for the discovery that ubiquitination leads to protein decay [18, 22] indicates that this is a rather severe assumption. The second limitation is that, once produced, a protein has a constant efficacy with which it controls the activation or repression of its targets. This ignores the common phenomenon of protein-protein interactions that can have a dramatic effect on the regulatory capabilities of the targeted protein and a decisive effect on network function [23]. In other words, the current DSGRN does not allow for general enough interpretations/expression of both Γ_n and Λ_n, i.e., it is lacking with respect to C1.

Observe that the abovementioned biochemical constraints are associated with how the output of one node affects the activity of another node and thus are expressed via edges within the regulatory network. With this in mind in this paper we extend the DSGRN framework and the corresponding software to allow for regulatory networks with edges of the form described in Figure 1. More detailed descriptions of the meaning of this notation are provided in sections 2 and 3. For the moment it is sufficient to know that the diagrams indicate edges with the following properties.

1. The *solid (type* 0*)* edges of Figure 1(a) indicate that an increase in x_1 leads to a higher rate of production of x_2, while an increase in x_3 leads to a decrease in production of x_4. Regulatory networks consisting only of these types of edges can be handled by the original DSGRN software.
2. The *dashed (type* I*)* edges indicated in Figure 1(b) indicate that an increase in x_1 increases the decay rate of x_2, while an increase in x_3 decreases the decay rate of x_4.
3. The *type* II edges are pairs of edges where one edge connects a node to the other edge. A solid dot on the edge in Figure 1(c) indicates that the product from node 1 needs to be modified to become active. The pointed arrow from node 3 to the edge indicates that an increase in x_3 leads to an increase in the fraction of x_1 that is modified, while the blunt arrow from node 2 indicates that an increase in x_2 leads to a decrease in the fraction of x_1 that is modified. The pointed arrow from node 1 indicates that once the modification occurs the product of 1 acts as an activator. The blunt arrow from node $1'$ indicates that once modified the output from node $1'$ acts as a repressor.
4. The empty dot on the receiving edge in Figure 1(d), also type II edges, indicates that the product from node 1 is active, but that it can be modified to be deactivated. In particular, the blunt edge of node 3 indicates that x_3 decreases the fraction of x_1 that is modified, while the pointed arrow of node 2 indicates that x_2 maintains or increases the fraction of x_1 that is modified.
5. We conclude the description by noting that activity modifications described in Figures 1(c)–(d) can also be applied to dashed arrows from Figure 1(b). That is, the solid and empty dots can also be placed on dashed edges.

The biologically minded reader may wish to identify Figure 1(b) with the process of ubiquitination and processes in Figures 1(c) and (d) with phosphorylation and dephosphorylation, or other post-transcriptional and post-translational protein modification.

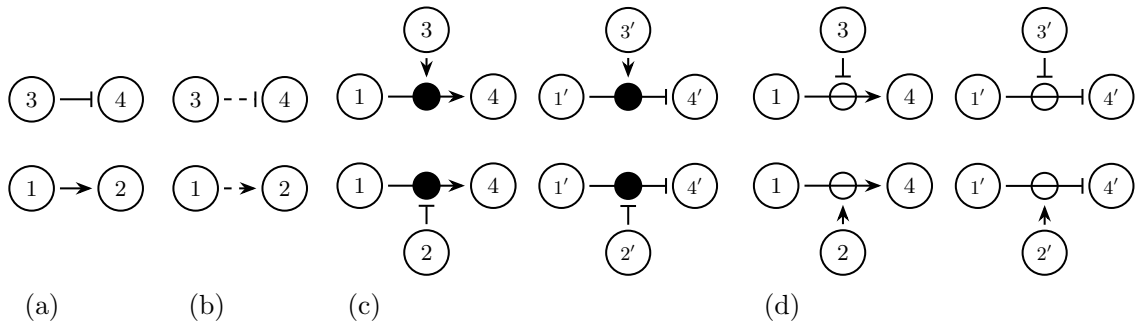

Figure 1. (a) **Type 0 edges.** *Direct up-regulation of* 2 *by* 1 *and direct down regulation of* 4 *by* 3. (b) **Type I edges.** *Node* 1 *increases decay rate of* 2 *and* 3 *decreases decay rate of* 4. (c) **Type II edges.** *Output of* 1 *has two states, unmodified and modified. The solid dot indicates that the product from node* 1 *needs to be modified to become active. The pointed arrow from* 3 *leads to modification of* 1 *(thus activates) and* 2 *leads to demodification of* 1 *(thus represses).* (d) **Type II edges**. *Output of* 1 *has two states, unmodified and modified. The hollow dot indicates that the product from node* 1 *is active, but it can be modified to be deactivated. The blunt arrow from* 3 *leads to modification of* 1 *(thus represses) and* 2 *leads to demodification of* 1 *(thus activates).*

Keeping in mind that the focus of this paper is on C2, the outline for this paper is as follows. In section 2 we review the mathematics and combinatorics that provide the foundation for the current version of the DSGRN software. As explained in this section, the most significant constraint on the software is the number of in-and out-edges, as well as their *interaction type* at any given node. The interaction type describes how the inputs at each node are combined to inform the node's output. In section 3 we describe the extension to DSGRN and via Tables 1 and 2 we indicate the interaction types that the extended software can currently handle. In principle these two sections provide sufficient information for a user to have a conceptual understanding of how the DSGRN software functions.

Of course, the DSGRN software has been constructed with the aim of being a useful tool in the analysis of regulatory networks that arise from applications. Thus in section 3 we consider C1 and discuss ODE models arising from post-transcriptional regulation of gene regulatory networks and demonstrate how the computations that DSGRN performs can be identified with a singular limit of these systems. In section 4 we compare the range of global dynamics for networks based on edges of the type in Figure 1(a) against similar networks that allow for the full range of edge interaction, i.e., including Figures 1(b)–(d).

2. DSGRN.

The computational utility of the DSGRN software arises from two distinct combinatorial abstractions. The first is a combinatorial representation of the dynamics. Furthermore, the expansion of DSGRN presented in this paper requires no adjustments to the previous versions regarding the representation of the dynamics and therefore we refer the reader to [5, 11] for details on the DSGRN approach. The second is an explicit decomposition of parameter space into a finite collection of semialgebraic sets, with the property that the combinatorial dynamics is constant for all parameters within each such semialgebraic set. A key point of this paper is the derivation of the proper decompositions for networks that contain edges of the form in Figures 1(b)–(d), and thus we review the decomposition in this section.

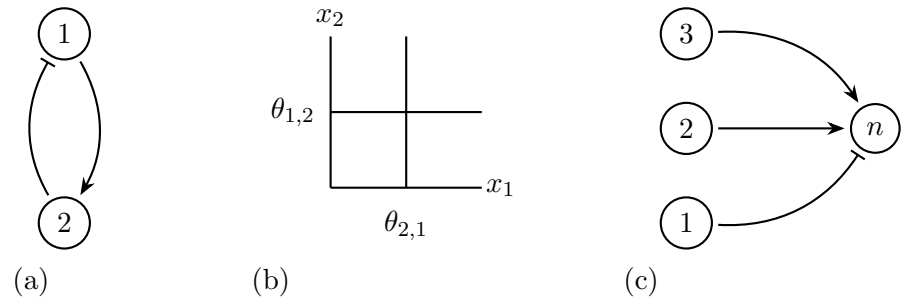

Figure 2. (a) *A* 2*-node network.* (b) *Decomposition of phase space of the network in* (a) *into four domains by thresholds.* (c) *Nodes* 1, 2, *and* 3 *affect node* n *(see text for details).*

To explain how the DSGRN software decomposes parameter space we begin by considering the particularly simple network in Figure 2(a) (for complete details the reader is referred to [5, 11, 21]). Notice that the edges are type 0 edges shown in Figure 1(a). As indicated in the introduction and compatible with the discussion of Figure 1, the decay rate for x_n in (1.1) is assumed to be a positive constant that we denote by γ_n. The rate of production of x_1 is given by $\Lambda_1(x_2)$ since the unique in-edge to node 1 comes from node 2. Similarly, the rate of production of x_2 is given by $\Lambda_2(x_1)$. The edge $2 \dashv 1$ indicates that x_2 represses the production of x_1, while the edge $1 \to 2$ indicates that x_1 activates the production of x_2. Step functions provide the simplest characterization of these phenomena, thus we introduce the functional expressions

(2.1)
$$\lambda^+_{n,m}(x_m) := \begin{cases} \ell_{n,m} & \text{if } x_m < \theta_{n,m}, \\ \ell_{n,m} + \delta_{n,m} & \text{if } x_m > \theta_{n,m}, \end{cases} \quad \text{and} \quad \lambda^-_{n,m}(x_m) := \begin{cases} \ell_{n,m} + \delta_{n,m} & \text{if } x_m < \theta_{n,m}, \\ \ell_{n,m} & \text{if } x_m > \theta_{n,m}, \end{cases}$$

where the parameters $\theta_{n,m}$, $\ell_{n,m}$, and $\delta_{n,m}$ are assumed to be positive. In particular, for the regulatory network of Figure 2(a) the rate of production of x_1 and x_2, given by (1.1), takes the form

(2.2)
$$\begin{aligned} &-\gamma_1 x_1 + \lambda^-_{1,2}(x_2), \\ &-\gamma_2 x_2 + \lambda^+_{2,1}(x_1). \end{aligned}$$

We associate four positive parameters to each node: $(\gamma_1, \theta_{2,1}, \ell_{1,2}, \delta_{1,2})$ to node 1, and $(\gamma_2, \theta_{1,2}, \ell_{2,1}, \delta_{2,1})$ to node 2.

Thus, the parameter space is $(0,\infty)^8 = (0,\infty)^4 \times (0,\infty)^4$. The phase space is $(0,\infty)^2$.

The functions $\lambda^-_{1,2}$ and $\lambda^+_{2,1}$ are constant off the hyperplanes $x_1 = \theta_{2,1}$ and $x_2 = \theta_{1,2}$, and thus there is a natural decomposition of phase space into rectangular regions (see Figure 2(b)). We focus for the moment on the behavior of x_1, restricting our attention to whether within one of these regions x_1 is increasing or decreasing, i.e., on the sign of

(2.3)
$$-\gamma_1 x_1 + \begin{cases} \ell_{1,2} + \delta_{1,2} & \text{if } x_2 < \theta_{1,2}, \\ \ell_{1,2} & \text{if } x_2 > \theta_{1,2}. \end{cases}$$

Note that since the first term is linear in x_1 it is sufficient to determine this at the threshold $x_1 = \theta_{2,1}$. Therefore, the answer depends entirely on the parameters, i.e., on one of the three possible relationships between parameters,

$$\gamma_1\theta_{2,1} < \ell_{1,2} < \ell_{1,2} + \delta_{1,2}, \quad \ell_{1,2} < \gamma_1\theta_{2,1} < \ell_{1,2} + \delta_{1,2}, \quad \ell_{1,2} < \ell_{1,2} + \delta_{1,2} < \gamma_1\theta_{2,1}. \tag{2.4}$$

Remark 2.1. As will be made clear shortly, it is useful to derive the relations of (2.4) by beginning with the linear order $\ell_{1,2} < \ell_{1,2} + \delta_{1,2}$ and then considering all possible relative values of $\gamma_1\theta_{2,1}$.

Observe that (2.4) provides an explicit decomposition of $(0,\infty)^4$, the parameter space associated with node 1, that we codify as a graph:

$$\boxed{\gamma_1\theta_{2,1} < \ell_{1,2} < \ell_{1,2} + \delta_{1,2}} \text{——} \boxed{\ell_{1,2} < \gamma_1\theta_{2,1} < \ell_{1,2} + \delta_{1,2}} \text{——} \boxed{\ell_{1,2} < \ell_{1,2} + \delta_{1,2} < \gamma_1\theta_{2,1}} \tag{2.5}$$

The nodes (represented by the rectangles) represent the regions in parameter space and edges indicate a single equality that defines adjacency of the regions. We refer to this graph as the *factor graph* associated to node 1 and denote it by $PG(1)$. There is a similar factor graph for node 2 and the full *parameter graph* is given by $PG := PG(1) \times PG(2)$. Observe that this provides a decomposition of the full parameter space $(0,\infty)^8$ into 9 parameter domains.

We now consider a general regulatory network with edges of the form of Figure 1(a). The most significant difference is that a node may have multiple in-edges. For the sake of notational simplicity assume that node n has K in-edges coming from nodes $1,\dots,K$, in which case by (1.1) we are interested in

$$-\gamma_n x_n + \Lambda_n(x_1,\dots,x_K).$$

We make use of the following definition to express the allowable form of Λ_n.

Definition 2.2. Following [21, Definition 1.1], an *interaction function* of order K is a polynomial in K variables $z = (z_1,\dots,z_K)$ of the form

$$f(z) := \prod_{j=1}^{q} f_j(z),$$

where each factor has the form

$$f_j(z) = \sum_{i\in I_j} z_i$$

and the indexing sets $\{I_j \mid 1 \le j \le q\}$ form a partition of $\{1,\dots,K\}$. The interaction type of f is $(k_1,\dots,k_q) \in \mathbb{N}^q$, where k_j denotes the number of elements of I_j with $k_j \le k_{j+1}$.

Remark 2.3. We denote the interaction type $(k_1,\dots,k_q)$ with the convention that $k_j \le k_{j+1}$ to match with the indexing used by DSGRN ([21] uses the convention $k_j \ge k_{j+1}$).

We assume that the function Λ_n is given by

$$\Lambda_n(x_1,\dots,x_K) = f\Big(\lambda^{\pm}_{n,1}(x_1), \lambda^{\pm}_{n,2}(x_2),\dots,\lambda^{\pm}_{n,K}(x_K)\Big), \tag{2.6}$$

where f is an interaction function of order K and the sign of $\lambda^{\pm}_{n,j}$ is determined by the edge from node j to node n.

In a slight abuse of notation, we sometimes also refer to the polynomial expression of the interaction function f as its interaction type. As an example for the interaction function $f(z) = z_1(z_2 + z_3)$ of type $(1, 2)$, we denote this interaction type as $z_1(z_2 + z_3)$. This notation conveys the same information in a less compressed format and will be useful when describing type I and II edges.

Consider the following example. Let node n have three in-edges as shown in Figure 2(c). Given that $1 \dashv n$, $2 \to n$, and $3 \to n$, a biologically motivated choice for Λ_n is

$$\begin{aligned}\Lambda_n(x_1, x_2, x_3) &= \lambda^{-}_{n,1}(x_1)\left(\lambda^{+}_{n,2}(x_2) + \lambda^{+}_{n,3}(x_3)\right)\\ &= \left(\begin{cases}\ell_{n,1} + \delta_{n,1} & \text{if } x_1 < \theta_{n,1}\\ \ell_{n,1} & \text{if } x_1 > \theta_{n,1}\end{cases}\right)\left(\begin{cases}\ell_{n,2} & \text{if } x_2 < \theta_{n,2}\\ \ell_{n,2} + \delta_{n,2} & \text{if } x_2 > \theta_{n,2}\end{cases} + \begin{cases}\ell_{n,3} & \text{if } x_3 < \theta_{n,3}\\ \ell_{n,3} + \delta_{n,3} & \text{if } x_3 > \theta_{n,3}\end{cases}\right).\end{aligned}$$

The associated interaction function of type $(k_1, k_2) = (1, 2)$ is

$$f(z) = z_1(z_2 + z_3).$$

Observe that the values that Λ_n can assume are given by the following eight polynomials in parameters ℓ and δ:

$$\begin{aligned}
p_0 &= \ell_{n,1}(\ell_{n,2} + \ell_{n,3}), & p_4 &= \ell_{n,1}(\ell_{n,2} + \ell_{n,3} + \delta_{n,3}),\\
p_1 &= (\ell_{n,1} + \delta_{n,1})(\ell_{n,2} + \ell_{n,3}), & p_5 &= (\ell_{n,1} + \delta_{n,1})(\ell_{n,2} + \ell_{n,3} + \delta_{n,3}),\\
p_2 &= \ell_{n,1}(\ell_{n,2} + \delta_{n,2} + \ell_{n,3}), & p_6 &= \ell_{n,1}(\ell_{n,2} + \delta_{n,2} + \ell_{n,3} + \delta_{n,3}),\\
p_3 &= (\ell_{n,1} + \delta_{n,1})(\ell_{n,2} + \delta_{n,2} + \ell_{n,3}), & p_7 &= (\ell_{n,1} + \delta_{n,1})(\ell_{n,2} + \delta_{n,2} + \ell_{n,3} + \delta_{n,3}).
\end{aligned}$$

To put this into proper perspective we return to Remark 2.1. For Λ_1, arising in the case where node 1 has a single in-edge, we have the polynomials

$$p_0 = \ell_{1,2} \quad \text{and} \quad p_1 = \ell_{1,2} + \delta_{1,2}.$$

Observing that $p_0 < p_1$ is the only admissible linear order for the values of these polynomials allows us to determine the decomposition of parameter space. The reader can check that the complete list of admissible linear orders associated with Λ_n is exactly

$$\begin{array}{cccc}
(0,1,2,3,4,5,6,7) & (0,1,2,4,6,3,5,7) & (0,1,4,5,2,3,6,7) & (0,2,1,4,6,3,5,7)\\
(0,1,2,3,4,6,5,7) & (0,1,4,2,5,3,6,7) & (0,1,4,5,2,6,3,7) & (0,2,4,1,6,3,5,7)\\
(0,1,2,4,3,5,6,7) & (0,1,4,2,5,6,3,7) & (0,2,1,3,4,6,5,7) & (0,2,4,6,1,3,5,7)\\
(0,1,2,4,3,6,5,7) & (0,1,4,2,6,5,3,7) & (0,2,1,4,3,6,5,7) & (0,4,1,2,5,6,3,7)\\
(0,4,2,6,1,5,3,7) & (0,4,1,5,2,6,3,7) & (0,4,2,1,6,5,3,7) & (0,4,1,2,6,5,3,7)
\end{array}$$

where $(0,1,2,3,4,5,6,7)$ corresponds to $p_0 < p_1 < p_2 < p_3 < p_4 < p_5 < p_6 < p_7$. To determine the associated factor graph we consider all possible values of $\{\gamma_n\theta_{m,n}\}_m$ relative to

each linear order where m ranges over all out-edges of node n. This produces the factor graph $PG(n)$.

As indicated in [5] and [21, Table 1] the complete list of admissible linear orders have been determined for Λ_n associated with interaction functions of type $(k_1, \dots, k_q)$ given by

$$
\begin{array}{l}
(1) \\
(1,1),\ (2) \\
(1,1,1),\ (1,2),\ (3) \\
(1,1,1,1),\ (1,1,2),\ (2,2),\ (1,3),\ (4) \\
(1,1,1,1,1),\ (1,1,1,2),\ (5) \\
(1,1,1,1,1,1),\ (6)
\end{array}
\tag{2.7}
$$

Thus, the DSGRN software can, in principle, take as input any network consisting of edges of type 0 in Figure 1(a) with the restriction that at any node n the production function Λ_n is given by one of the interaction functions of the type listed above. Using precomputed factor graphs $PG(n)$, DSGRN software constructs the parameter graph as the product $PG = \prod_{n=1}^{N} PG(n)$. In applications, the most serious restriction is that size of the factor graphs grows rapidly as a function of the number of variables, e.g., for a node with interaction function of type (6) with one out-edge the factor graph has 89,414,640 elements.

A goal of this paper is to identify the factor graphs, and hence make the DSGRN software applicable, for networks that involve edges of type Figures 1(b)–(d).

Remark 2.4. There is an alternative characterization of parameter nodes in a factor parameter graph as collections of Boolean functions. Any input polynomial to node n can be represented as a Boolean string of length k—the number of inputs to node n. As an example consider node n with three in-edges as shown in Figure 2(c) and discussed above. Then each of the polynomials $p_0, \dots, p_7$ can be represented as a Boolean string (b_1, b_2, b_3), $b_i \in B = \{0, 1\}$ by assigning $b_i = 0$ if p contains $\ell_{n,i}$ and $b_i = 1$ if p contains $\ell_{n,i} + \delta_{n,i}$. Since for each threshold $\theta_{m,n}$ of node n, the parameter node contains a linear order

$$p_0 < \cdots < p_i < \theta_{m,n} < p_j < \cdots < p_7,$$

we can associate to it a Boolean function $g_{m,n} : B^3 \to B$ by setting $g_{m,n}(p_s) = 0$ if and only if $p_s < \theta_{m,n}$. Therefore a node of a factor parameter graph that corresponds to a node n of a regulatory network with k inputs and m outputs corresponds to a collection of m Boolean functions with k inputs. For a more detailed description see [3].

2.1. State transition graph. To close the review of DSGRN, we discuss how dynamics is associated to each parameter node. This association is the same for new types of interactions presented in this paper. As indicated in (2.3), for all parameters that belong to a domain in the parameter space represented by a parameter node, each x_n is either increasing or decreasing on the boundaries of domains bounded by thresholds $\theta_{j,n}$. This leads to a construction of a state transition graph that represents the dynamics of the system.

To simplify the explanation we assume that the gene regulatory network does not contain repressing self-edges, as was assumed in the original DSGRN [5]. This ensures that the state

transition graph, as we describe below, is well defined. However, this is not an essential limitation. In [11] this problem was fully resolved by replacing the threshold corresponding to a repressing self-edge by two thresholds and thus extending the state transition graph. In this paper we apply the same method used in [11] to replace the thresholds corresponding to all self-edges in the network (both repressing and activating) by two thresholds.

We also allow networks with multiple edges between pairs of nodes and use the DSGRN extension described in [12] to construct the state transition graph in that case.

The collection of hyperplanes $\{x_m = \theta_{*,m}\}$, $m = 1, \ldots, N$, provides a cubical decomposition of the phase space $(0, \infty)^N$. We refer to the associated N-dimensional cubes as *domains*, and these domains define the vertices of the state transition graph $\mathcal{F}$. To define the edges of $\mathcal{F}$, let κ be a domain, with an $(N-1)$-dimensional face σ that is a subset of the hyperplane $x_n = \theta_{m,n}$. The sign of $-\gamma_n\theta_{m,n} + \Lambda_n(\kappa)$ determines whether σ points into or away from κ. Note that for all parameters belonging to a parameter node of PG this sign is the same and thus the following construction produces the same $\mathcal{F}$. The edges of the state transition graph $\mathcal{F}$ are now defined by the following two rules.

R1. If all the faces of κ point into κ, then κ has a self-edge.

R2. If σ is an $(N-1)$-dimensional face of two domains κ and κ' and σ points away from κ' and into κ, then there is an edge from κ' to κ.

Given a directed graph $\mathcal{F}$, the condensation graph $\mathcal{F}^{\mathrm{SCC}}$ can be identified in linear time [2]. Recall that $\mathcal{F}^{\mathrm{SCC}}$ is a directed acyclic graph with one node for each strongly connected component (SCC) of $\mathcal{F}$ and hence a poset. We define an SCC to be *nontrivial* if it contains at least one edge. We define $(\mathsf{M}(\mathcal{F}), \leq)$, the *Morse poset* of $\mathcal{F}$, to be the subposet of $\mathcal{F}^{\mathrm{SCC}}$ consisting of the nontrivial SCCs. The Haase diagram for $\mathsf{M}(\mathcal{F})$ is called the *Morse graph*.

An obvious question about the DSGRN state transition graph dynamics, characterized by Morse graphs, is whether this characterization applies to network ODE models where smooth nonlinearities approximate the piecewise constant functions $\Lambda_n(x)$. The correspondence between individual solutions of such a smooth perturbed system and a system generated by ODEs with right-hand sides (1.1) was studied by Ironi et al. [19]; the correspondence between the lattices of attractors for ODEs in $\mathbb{R}^2$ was established in [15]. As is demonstrated in [9] there is a close correspondence between equilibria and their stability for the smooth perturbed systems and the characterization provided by DSGRN. Finally, bifurcations of DSGRN equilibria within a class of ramp function perturbations have been examined in [8].

We conclude this section with a summary:

1. For a given regulatory network and a type of interaction between inputs at each node of the network, there is a finite decomposition of the parameter space, encoded as the parameter graph PG, such that all parameters in a single parameter node admit the same state transition graph.
2. The long-term dynamics of the state transition graph is represented by its Morse graph.
3. The parameter graph PG with a Morse graph at each vertex encodes a finite representation of the dynamics of the regulatory network.

3. Extension of DSGRN.

In this section we consider three generalizations of the parameter space decomposition (PSD) problem originally described in [21] which we review briefly.

Suppose that node n has K input edges which are labeled $\{1,\dots,K\}$ and assume that the rate expression for x_n is given by $-\gamma_n x_n + \Lambda_n(x)$, which we refer to as a *classical* rate expression. Assume that f is an interaction function as defined in Definition 2.2 with order K and type $(k_1,\dots,k_q)$, and Λ_n is given by the formula (2.6). Observe that the image of Λ_n consists of 2^K values

$$\mathcal{P} := \{f(z_1,\dots,z_K) \mid z_j \in \{\ell_{n,j}, \ell_{n,j}+\delta_{n,j}\},\ 1 \le j \le K\},$$

regardless of the sign of $\lambda^{\pm}_{n,j}$ for $1 \le j \le K$, and independent of the interaction type of f. The elements of $\mathcal{P}$ are polynomial expressions in the $2K$ parameters $\{\ell_{n,j}, \delta_{n,j} \mid 1 \le j \le K\}$.

The *PSD problem* associated with f is to compute all possible linear orders of $\mathcal{P}$ subject to the constraints $\ell_{n,j}, \delta_{n,j} > 0$ for all $1 \le j \le K$. In [21] it is shown that given a solution of the PSD problem, the factor graph $PG(n)$ can be recovered with a trivial amount of post-processing. In general, rigorously solving the PSD problem is difficult, but it needs to be solved only once for each interaction type and the results stored; see (2.7).

In the remainder of this section we introduce the generalization of the PSD problem for the type I and type II edges defined in section 1. These new rate expressions give rise to new PSD problems which we define below. Then, we demonstrate that the solutions of these new PSD problems can be obtained by creatively using classical PSD solutions.

3.1. Type I edges. We start with the PSD for a rate expression that includes type I edges shown in Figure 1(b). As is the case throughout this paper the mathematical expression governing the production of x_n has the form (1.1) with $x = (x_1,\dots,x_N) \in \mathbb{R}^N$.

However, to emphasize that we assume that a type I edge impacts decay as opposed to production we highlight the distinction of the variables and write

$$-\Gamma_n(\tilde{x})x_n + \Lambda_n(x), \tag{3.1}$$

where Λ_n has the form (2.6) and involves only type 0 edges, that is, it is defined by an interaction function f of order K and depends on the state variables $x_1,\dots,x_K$. We will refer to Λ_n as the *classical production rate expression*. The function Γ_n also has the form (2.6), but involving only type I edges, that is, it is an interaction function composed of step functions that depend on the state variables $\tilde{x}_j \in \{x_1,\dots,x_N\}$ for $j = 1,\dots,\tilde{K}$, where the state variables $\{x_1,\dots,x_K\}$ and $\{\tilde{x}_1,\dots,\tilde{x}_{\tilde{K}}\}$ form disjoint sets. Specifically,

$$\Gamma_n(\tilde{x}) = \tilde{f}\Big(\tilde{\lambda}^{\pm}_{n,1}(\tilde{x}_1), \tilde{\lambda}^{\pm}_{n,2}(\tilde{x}_2), \dots, \tilde{\lambda}^{\pm}_{n,\tilde{K}}(\tilde{x}_{\tilde{K}})\Big),$$

where $\tilde{f}$ is an interaction function of order $\tilde{K}$, and interaction type $(\tilde{k}_1,\dots,\tilde{k}_{\tilde{q}})$. For $1 \le j \le \tilde{K}$, $\tilde{\lambda}^{\pm}_{n,j}$ is a step function depending on parameters $\{\tilde{\ell}_{n,j}, \tilde{\delta}_{n,j}\}$ and the sign of $\tilde{\lambda}^{\pm}_{n,j}$ is determined by the type of edge in Figure 1(b). By a similar observation as in the classical rate expression we note that Γ_n is a simple function which takes the values

$$\tilde{\mathcal{P}} := \Big\{\tilde{f}\big(z_1,\dots,z_{\tilde{K}}\big) \mid z_j \in \Big\{\tilde{\ell}_{n,j}, \tilde{\ell}_{n,j}+\tilde{\delta}_{n,j}\Big\},\ 1 \le j \le \tilde{K}\Big\},$$

regardless of signs of each step function or the interaction type of $\tilde{f}$.

Analogous to the classical rate expression, we are interested in determining the sign of a *type* I *rate expression* of the form

$$-\Gamma_n(\tilde{x})\theta_{*,n} + \Lambda_n(x) \tag{3.2}$$

where $\theta_{*,n}$ is an arbitrary threshold associated to an outgoing edge whose source is node n.

A crucial observation is that the sign determination of (3.2) depends only on the value of $\theta_{*,n}$ relative to the expression $\Lambda_n(x)/\Gamma_n(\tilde{x})$. The problem of sign determination for (3.2) over all parameter values can be reduced to the problem of determining all possible linear orders of the finite set

$$\mathcal{R} := \left\{ \frac{p}{\tilde{p}} \mid p \in \mathcal{P},\ \tilde{p} \in \tilde{\mathcal{P}} \right\}, \tag{3.3}$$

subject to the constraints $\ell_{n,i}, \delta_{n,i} > 0$ for $1 \le i \le K$ and $\tilde{\ell}_{n,j}, \tilde{\delta}_{n,j} > 0$ for $1 \le j \le \tilde{K}$. This is the *joint PSD problem* associated to the pair $(\tilde{f}, f)$ with a *joint interaction type* denoted by $(\tilde{k}_1, \dots, \tilde{k}_{\tilde{q}}; k_1, \dots, k_q)$. As in the classical case, given a solution of the joint PSD problem the sign of (3.2) can be determined for the entire parameter space with trivial postprocessing.

Similarly to the case of a type 0 rate expression, we use a shorthand notation to describe the interaction type of a type I rate expression. In this case we put the polynomial expression of $\tilde{f}$ within $\langle\cdot\rangle$ and add to that the polynomial expression of f, that is, we denote the interaction type by $\langle \tilde{f} \rangle + f$. As an example, assuming we are using a rate expression with interaction type $(1;2)$ for node n in Figure 3(a), we express this fact by saying that the interaction type of node n is

$$\langle x \rangle + y + z.$$

Before presenting the general result, we demonstrate how to obtain the linear orders of a type I rate expression in a simple example.

3.2. Example. Consider the simplest possible type I rate expression for x_n,

$$-\Gamma_n(x_1)x_n + \Lambda_n(x_2),$$

where x_1 and x_2 are state variables. For simplicity, we suppress the dependence on n which usually appears in parameter subscripts and assume that x_1 and x_2 are both promoters of x_n so that we have the formulas

$$\Gamma_n(x_1) = \begin{cases} \ell_1 + \delta_1 & \text{if } x_1 < \theta_1, \\ \ell_1 & \text{if } x_1 > \theta_1, \end{cases} \qquad \Lambda_n(x_2) = \begin{cases} \ell_2 & \text{if } x_2 < \theta_2, \\ \ell_2 + \delta_2 & \text{if } x_2 > \theta_2, \end{cases}$$

where $\{\ell_1, \delta_1, \theta_1, \ell_2, \delta_2, \theta_2\}$ are positive parameters. The interaction type associated with this example is $(1;1)$ and it is denoted by $\langle z_1 \rangle + z_2$.

The associated joint PSD problem is to determine all admissible linear orders of the rational expressions

$$\mathcal{R} := \left\{ \frac{\ell_2}{\ell_1}, \frac{\ell_2 + \delta_2}{\ell_1}, \frac{\ell_2}{\ell_1 + \delta_1}, \frac{\ell_2 + \delta_2}{\ell_1 + \delta_1} \right\}$$

subject to the constraints that all parameters must be strictly positive.

Define the polynomials

$$p_0 = \ell_2, \quad p_1 = \ell_2 + \delta_2, \quad \tilde{p}_0 = \ell_1, \quad \tilde{p}_1 = \ell_1 + \delta_1,$$

which we collect into distinct subsets $\mathcal{P} := \{p_0, p_1\}$ and $\tilde{\mathcal{P}} := \{\tilde{p}_0, \tilde{p}_1\}$. We let the set $\mathcal{R}$ inherit this indexing by writing $\mathcal{R} = \left\{ r_{ij} \mid r_{ij} = \frac{p_i}{\tilde{p}_j},\ 0 \le i, j, \le 1 \right\}$. Let $\mu = (\ell_1, \delta_1, \ell_2, \delta_2)$ denote an arbitrary parameter vector and observe that for any $\mu \in (0, \infty)^4$ we have

$$r_{01}(\mu) < r_{00}(\mu) < r_{10}(\mu) \qquad \text{and} \quad r_{01}(\mu) < r_{11}(\mu) < r_{10}(\mu),$$

which induces a partial order on $\mathcal{R}$ given by $r_{01} \prec \{r_{00}, r_{11}\} \prec r_{10}$. One easily checks that μ can be chosen such that either linear order of the set $\{r_{00}(\mu), r_{11}(\mu)\}$ is possible. Thus, there are exactly two admissible linear extensions of $(\mathcal{R}, \prec)$ given by

$$(r_{01}, r_{00}, r_{11}, r_{10}), \qquad (r_{01}, r_{11}, r_{00}, r_{10}). \tag{3.4}$$

We compare this result to a PSD problem for interaction type $(1, 1)$ which corresponds to a classical rate expression of the form

$$-\gamma_n x_n + \Lambda_n(x_1, x_2),$$

where γ_n is a positive parameter and Λ_n is given by the formula

$$\Lambda_n(x_1, x_2) = \lambda_1^+(x_1)\lambda_2^+(x_2).$$

The classical PSD is to determine all possible linear orders of the polynomials

$$\mathcal{Q} := \{\ell_2\ell_1, \ell_2(\ell_1 + \delta_1), (\ell_2 + \delta_2)\ell_1, (\ell_2 + \delta_2)(\ell_1 + \delta_1)\} = \{p_0\tilde{p}_0, p_0\tilde{p}_1, p_1\tilde{p}_0, p_1\tilde{p}_1\}.$$

In the previous version of DSGRN these four polynomials are endowed with a partial order by assigning labels via the indexing

$$q_0 := \ell_1\ell_2, \quad q_1 := (\ell_1 + \delta_1)\ell_2, \quad q_2 := \ell_1(\ell_2 + \delta_2), \quad q_3 := (\ell_1 + \delta_1)(\ell_2 + \delta_2)$$

with the partial order $q_0 \prec \{q_1, q_2\} \prec q_3$. One easily checks that both possible orders between q_1 and q_2 can be satisfied. Thus, the solution to this classical PSD problem is the two possible linear orders

$$(0, 1, 2, 3), \qquad (0, 2, 1, 3), \tag{3.5}$$

which are stored in DSGRN. However, if instead we indexed the same four polynomials as

$$\bar{q}_0 := p_0\tilde{p}_0, \quad \bar{q}_1 := p_1\tilde{p}_0, \quad \bar{q}_2 := p_0\tilde{p}_1, \quad \bar{q}_3 := p_1\tilde{p}_1,$$

then we observe that

$$\bar{q}_1(\mu) < \bar{q}_2(\mu) \iff \frac{p_1(\mu)}{\tilde{p}_1(\mu)} < \frac{p_0(\mu)}{\tilde{p}_0(\mu)} \iff r_{11}(\mu) < r_{00}(\mu).$$

It follows that finding all admissible linear orders of $\mathcal{R}$ is equivalent to finding all admissible linear orders of $\mathcal{Q}$ with respect to the new indexing. However, this has already been computed and stored in DSGRN and we only require the correct change of indexing on these polynomials in order to reuse it and this is trivial to store in a table and look up as needed.

Intuitively, choosing the relative order of the two "free" polynomials for the $(1,1)$ PSD problem is equivalent to choosing the relative order of the two "free" rational expressions for the $(1;1)$ PSD problem. Moreover, given the two possible linear orders for $\mathcal{Q}$, the two admissible orders for $\mathcal{R}$ are immediately recovered.

Next, we prove that solving any joint PSD problem is equivalent to solving a related classical PSD problem. In particular, the joint PSD problem can be solved using the algorithms described in [21].

Theorem 3.1. *Suppose x_n is governed by a type* I *rate expression $-\Gamma_n(\tilde{x})x_n + \Lambda_n(x)$ with associated interaction functions $\tilde{f}$ and f of orders $\tilde{K}$ and K and interaction types $(\tilde{k}_1, \dots, \tilde{k}_{\tilde{q}})$ and $(k_1, \dots, k_q)$, respectively, with $\{x_1, \dots, x_K\}$ and $\{\tilde{x}_1, \dots, \tilde{x}_{\tilde{K}}\}$ disjoint sets of state variables.*

Then, the joint PSD problem associated to $(\tilde{f}, f)$ is equivalent to the classical PSD problem associated to the interaction function $f \cdot \tilde{f}$ in the following sense.

1. *There is bijection between solutions of the joint PSD problem for $(\tilde{f}, f)$ and solutions of the classical PSD problem associated with $f \cdot \tilde{f}$.*
2. *This bijection can be efficiently and explicitly constructed, i.e., the admissible linear orders for the joint PSD problem can be efficiently recovered given the admissible linear orders for the classical PSD problem.*

Proof. Let $g = f \cdot \tilde{f}$ and observe that g is an interaction function of order $K + \tilde{K}$ and interaction type $(k_1, \dots, k_q, \tilde{k}_1, \dots, \tilde{k}_{\tilde{q}})$.

Observe that this interaction type need not satisfy our previously stated convention that the summand sizes are in increasing order. For instance, it is possible that $\tilde{k}_1 < k_q$. However, this convention is simply a notational convenience since any interaction function is invariant under permutation of the summands. In particular, the set of admissible linear orders for an interaction type depends only on the summand sizes and does not depend on their order in the vector defining the interaction type.

The PSD problem associated with g is to determine all possible linear orders for a collection of $2^{K+\tilde{K}}$ polynomials denoted by

$$\mathcal{Q} \subset \mathbb{R}\Big[\ell_{n,1}, \dots, \ell_{n,K}, \delta_{n,1}, \dots, \delta_{n,K}, \tilde{\ell}_{n,1}, \dots, \tilde{\ell}_{n,\tilde{K}}, \tilde{\delta}_{n,1}, \dots, \tilde{\delta}_{n,\tilde{K}}\Big],$$

subject to the constraint that each of the $2(K + \tilde{K})$ indeterminates is positive. Observe that from the construction of g and the definition of the classical PSD problem, each $q \in \mathcal{Q}$ admits a unique factorization of the form $q = p \cdot \tilde{p}$ where

$$p \in \mathcal{P} := \{f(z_1, \dots, z_K) \mid z_i \in \{\ell_{n,i}, \ell_{n,i} + \delta_{n,i}\},\ 1 \le i \le K\},$$
$$\tilde{p} \in \tilde{\mathcal{P}} := \Big\{\tilde{f}\big(z_1, \dots, z_{\tilde{K}}\big) \mid z_j \in \Big\{\tilde{\ell}_{n,j}, \tilde{\ell}_{n,j} + \tilde{\delta}_{n,j}\Big\},\ 1 \le j \le \tilde{K}\Big\}.$$

We note that the collection of polynomials defining $\mathcal{Q}$ is closely related to the rational expressions appearing in the PSD problem associated to $(\tilde{f}, f)$. Specifically, the joint PSD problem

associated to $(f, \tilde{f})$ is to determine all possible linear orders for the collection of $2^{K+\tilde{K}}$ rational functions defined over the same $2(K+\tilde{K})$ indeterminate parameters denoted by

$$\mathcal{R} := \left\{ \frac{p}{\tilde{p}} \mid p_i \in \mathcal{P},\ \tilde{p}_j \in \tilde{\mathcal{P}} \right\} \subset \mathbb{R}\Big[\ell_{n,1}, \ldots, \delta_{n,K}, \tilde{\ell}_{n,1}, \ldots, \tilde{\delta}_{n,\tilde{K}}\Big], \tag{3.6}$$

subject to the same positivity constraints on the parameters.

In order to establish the bijection between linear orders of $\mathcal{Q}$ and $\mathcal{R}$ we note the following useful fact. Suppose $(\xi, \tilde{\xi}) \in (0,\infty)^{2K} \times (0,\infty)^{2\tilde{K}}$ where

$$\xi := (\ell_{n,1}, \ldots, \delta_{n,K}) \quad \text{and} \quad \tilde{\xi} := \Big(\tilde{\ell}_{n,1}, \ldots, \tilde{\delta}_{n,\tilde{K}}\Big).$$

Then for any $p_1, p_2 \in \mathcal{P}$ and $\tilde{p}_1, \tilde{p}_2 \in \tilde{\mathcal{P}}$ we have the equivalence,

$$\frac{p_1(\xi)}{\tilde{p}_1(\tilde{\xi})} < \frac{p_2(\xi)}{\tilde{p}_2(\tilde{\xi})} \quad \text{if and only if} \quad p_1(\xi)\cdot\tilde{p}_2(\tilde{\xi}) < \tilde{p}_1(\tilde{\xi})\cdot p_2(\xi). \tag{3.7}$$

The proof of this claim is a trivial consequence of the fact that elements of $\mathcal{P}, \tilde{\mathcal{P}}$ have only positive coefficients due to Definition 2.2 and all coordinates of $\xi, \tilde{\xi}$ are strictly positive. Therefore, the quantities $p_1(\xi), \tilde{p}_1(\tilde{\xi}), p_2(\xi), \tilde{p}_2(\tilde{\xi})$ are strictly positive for any choices of $\xi \in (0,\infty)^{2K}$ and $\tilde{\xi} \in (0,\infty)^{2\tilde{K}}$ and the equivalence claimed in (3.7) follows.

To complete the proof we assume that $\mathcal{P}$ and $\tilde{\mathcal{P}}$ are indexed by unspecified but fixed indexing maps onto the integers $I := \{0, \ldots, 2^K - 1\}$ and $J := \left\{0, \ldots, 2^{\tilde{K}} - 1\right\}$, respectively. We use subscripts to denote these indices so that the sets $\mathcal{P}$ and $\tilde{\mathcal{P}}$ can be expressed as

$$\mathcal{P} = \{p_i \mid i \in I\}, \qquad \tilde{\mathcal{P}} = \{\tilde{p}_j \mid j \in J\}.$$

These indices induce associated indexing maps on $\mathcal{Q}$ and $\mathcal{R}$ defined by

$$q_{ij} := p_i \cdot \tilde{p}_j \in \mathcal{Q}, \quad r_{ij} := \frac{p_i}{\tilde{p}_j} \in \mathcal{R},$$

for any $i \in I, j \in J$. Observe that with these indices for $\mathcal{Q}$ and $\mathcal{R}$ fixed, any linear order of $\mathcal{Q}$ or $\mathcal{R}$ can be uniquely identified with a permutation on $2^{K+\tilde{K}}$ symbols, i.e., an element of $S_{2^{K+\tilde{K}}}$. Consequently, the solution to the PSD problems associated with either g or $(\tilde{f}, f)$ can be identified with subsets of $S_{2^{K+\tilde{K}}}$ denoted by $\mathcal{T}_g$ and $\mathcal{T}_{(f,\tilde{f})}$, respectively.

Let us consider $\sigma \in \mathcal{T}_g$ which defines a linear order on $\mathcal{Q}$ which we interpret as a function, $\sigma : \left\{0, \ldots, 2^{K+\tilde{K}} - 1\right\} \to I \times J$. The assumption that σ defines an admissible linear order implies there exists a witness

$$\mu = \Big(\ell_{n,1}, \ldots, \ell_{n,K}, \delta_{n,1}, \ldots, \delta_{n,K}, \tilde{\ell}_{n,1}, \ldots, \tilde{\ell}_{n,\tilde{K}}, \tilde{\delta}_{n,1}, \ldots, \tilde{\delta}_{n,\tilde{K}}\Big) \in (0,\infty)^{2(K+\tilde{K})}$$

such that

$$q_{\sigma(0)}(\mu) < q_{\sigma(1)}(\mu) < \cdots < q_{\sigma(2^{K+\tilde{K}}-1)}(\mu),$$

or equivalently,

$$q_{\sigma(m-1)}(\mu) < q_{\sigma(m)}(\mu) \qquad \text{for all } 1 \le m \le 2^{K+\tilde{K}} - 1.$$

Fix an arbitrary $m \in \left\{1, \ldots, 2^{K+\tilde{K}} - 1\right\}$ and assume that $\sigma(m-1) = (i, j)$ and $\sigma(m) = (i', j')$. We split μ as

$$\mu = (\xi, \tilde{\xi}) \in (0, \infty)^{2K} \times (0, \infty)^{2\tilde{K}}$$

and recalling that $\mathcal{P}, \tilde{\mathcal{P}}$ are polynomials over disjoint parameters, we have

$$q_{\sigma(m-1)}(\mu) = p_i(\xi) \cdot \tilde{p}_j(\tilde{\xi}) < p_{i'}(\xi) \cdot \tilde{p}_{j'}(\tilde{\xi}) = q_{\sigma(m)}(\mu) \tag{3.8}$$

and applying (3.7) we conclude that

$$r_{ij'}(\mu) = \frac{p_i(\xi)}{\tilde{p}_{j'}(\tilde{\xi})} < \frac{p_{i'}(\xi)}{\tilde{p}_j(\tilde{\xi})} = r_{i'j}(\mu). \tag{3.9}$$

Applying the same argument for each $1 \leq m \leq 2^{K+\tilde{K}} - 1$ yields a distinct linear order on the elements of $\mathcal{R}$. Observe that this is not the same linear order as σ, but it is induced directly by σ and the same μ is a witness for both. In other words, the existence of $\sigma \in \mathcal{T}_g$ implies existence of a related linear order $\tau \in \mathcal{T}_{(\tilde{f},f)}$ such that

$$r_{\tau(0)}(\mu) < r_{\tau(1)}(\mu) < \cdots < r_{\tau(2^{K+\tilde{K}}-1)}(\mu),$$

where μ is any witness for σ. It follows that $\#\mathcal{T}_g \leq \#\mathcal{T}_{(\tilde{f},f)}$. However, a similar argument shows that any linear order $\tau \in \mathcal{T}_{(\tilde{f},f)}$ yields a distinct admissible linear order on $\mathcal{Q}$ by applying the converse of (3.7) to successive pairs of elements ordered by τ, which completes the proof of the first claim.

To prove the second claim, we simply observe that if $\mathcal{T}_g$ has been computed and stored, then $\mathcal{T}_{(\tilde{f},f)}$ is recovered by a small number of trivial lookup operations. Specifically, for each $\sigma \in \mathcal{T}_g$, the inverse indexing map for $\mathcal{Q}$ which decomposes $q_{\sigma(m)}$ into factors as in (3.8) must be evaluated, followed by an evaluation of the indexing map for $r_{\tau(m)}$ as in (3.9). This must be done for each $1 \leq m \leq 2^{K+\tilde{K}}$, in order to completely construct $\tau \in \mathcal{T}_{(\tilde{f},f)}$ from a given linear order $\sigma \in \mathcal{T}_g$. ∎

Theorem 3.1 demonstrates that solving the PSD problem for type I rate expressions is equivalent to solving the PSD problem for a related classical rate expression. In particular, for many network topologies of interest these classical PSD problems have already been solved and implemented in the current version of DSGRN, implying that the implementation of type I rate expressions often requires only minor modifications to the existing DSGRN library. See Table 1 for a list of all interaction types available.

3.3. Type II edges. Type II edges are pairs of edges where one edge connects a node to the other edge as shown in Figures 1(c)–(d); however, they are implemented as pairs of edges from the source node to the target node. For example, the pair of type II edges in Figure 3(b) is implemented as the pair of edges from nodes 1 and 2 to node 3 as shown in Figure 3(c) and the type II edges in Figure 3(d) are implemented as depicted in Figure 3(e).

The mathematical expression governing the production of x_n that includes type II edges is called a *type* II *rate expression* and has the form

$$-\gamma_n x_n + \Lambda_n(\tilde{w}_1, \ldots, \tilde{w}_{\tilde{K}}, x_1, \ldots, x_K), \tag{3.10}$$

Table 1

Complete list of available type I *interaction types involving type* I *and type* 0 *edges (first column). The size of the parameter factor graph for each of the interaction types is the same as the size of the corresponding original DSGRN interaction types listed (middle column). PTM = post-transcriptional modification.*

PTM interaction type	DSGRN interaction type	Interaction type
$\langle x\rangle + y$	xy	$(1;1)$
$\langle x\rangle + yz$	xyz	$(1;1,1)$
$\langle xy\rangle + z$	xyz	$(1,1;1)$
$\langle x\rangle + y + z$	$x(y+z)$	$(1;2)$
$\langle y+z\rangle + x$	$x(y+z)$	$(2;1)$
$\langle x\rangle + yzw$	$xyzw$	$(1;1,1,1)$
$\langle xy\rangle + zw$	$xyzw$	$(1,1;1,1)$
$\langle xyz\rangle + w$	$xyzw$	$(1,1,1;1)$
$\langle x\rangle + y(z+w)$	$xy(z+w)$	$(1;1,2)$
$\langle xy\rangle + z + w$	$xy(z+w)$	$(1,1;2)$
$\langle z+w\rangle + xy$	$xy(z+w)$	$(2;1,1)$
$\langle x(z+w)\rangle + y$	$xy(z+w)$	$(1,2;1)$
$\langle x+y\rangle + z + w$	$(x+y)(z+w)$	$(2;2)$
$\langle x\rangle + y + z + w$	$x(y+z+w)$	$(1;3)$
$\langle y+z+w\rangle + x$	$x(y+z+w)$	$(3;1)$
$\langle x\rangle + yzuw$	$xyzuw$	$(1;1,1,1,1)$
$\langle xy\rangle + zuw$	$xyzuw$	$(1,1;1,1,1)$
$\langle xyz\rangle + uw$	$xyzuw$	$(1,1,1;1,1)$
$\langle xyzu\rangle + w$	$xyzuw$	$(1,1,1,1;1)$
$\langle x\rangle + yz(u+w)$	$xyz(u+w)$	$(1;1,1,2)$
$\langle xy\rangle + z(u+w)$	$xyz(u+w)$	$(1,1;1,2)$
$\langle xyz\rangle + u + w$	$xyz(u+w)$	$(1,1,1;2)$
$\langle u+w\rangle + xyz$	$xyz(u+w)$	$(2;1,1,1)$
$\langle x(u+w)\rangle + yz$	$xyz(u+w)$	$(1,2;1,1)$
$\langle xy(u+w)\rangle + z$	$xyz(u+w)$	$(1,1,2;1)$
$\langle x\rangle + yzuvw$	$xyzuvw$	$(1;1,1,1,1,1)$
$\langle xy\rangle + zuvw$	$xyzuvw$	$(1,1;1,1,1,1)$
$\langle xyz\rangle + uvw$	$xyzuvw$	$(1,1,1;1,1,1)$
$\langle xyzu\rangle + vw$	$xyzuvw$	$(1,1,1,1;1,1)$
$\langle xyzuv\rangle + w$	$xyzuvw$	$(1,1,1,1,1;1)$

where $x_1,\dots,x_K$ represent nodes $1,\dots,k$ connected to node n by type 0 edges and $\tilde{w}_1,\dots,\tilde{w}_{\tilde{K}}$ represent pairs of nodes connected to n by type II edges, that is, $\tilde{w}_k = (\tilde{x}_{i_k}, \tilde{y}_{j_k})$, with $\tilde{x}_{i_k}, \tilde{y}_{j_k} \in \{x_1,\dots,x_N\}$, represents a pair of type II edges for $k = 1,\dots,\tilde{K}$. We assume that $\{x_1,\dots,x_K\}$, $\{\tilde{x}_{i_1},\dots,\tilde{x}_{i_{\tilde{K}}}\}$, and $\{\tilde{y}_{i_1},\dots,\tilde{y}_{i_{\tilde{K}}}\}$ are disjoint sets.

The production rate function Λ_n is given in terms of an interaction function f as defined in Definition 2.2 as follows.

In contrast to type 0 and I edges, to a pair of type II edges we associate a single ℓ and δ. More precisely, if $(\tilde{x}_{i_k}, \tilde{y}_{j_k})$ represents a pair of type II edges terminating at the node n, to

this pair we associate the positive parameters $\tilde{\ell}_{n,(i_k,j_k)}$, $\tilde{\delta}_{n,(i_k,j_k)}$. We still, however, associate one threshold per edge, that is, we associate $\tilde{\theta}_{n,i_k}$ and $\tilde{\theta}_{n,j_k}$ to the edges from i_k and j_k to n, respectively.

To simplify the presentation we denote

$$[\tilde{x}_{i_k}, \tilde{y}_{j_k}] := \min\{\lambda^{\pm}_{n,i_k}(\tilde{x}_{i_k}; \tilde{\ell}_{n,(i_k,j_k)}, \tilde{\delta}_{n,(i_k,j_k)}, \tilde{\theta}_{n,i_k}), \lambda^{\pm}_{n,j_k}(\tilde{y}_{j_k}; \tilde{\ell}_{n,(i_k,j_k)}, \tilde{\delta}_{n,(i_k,j_k)}, \tilde{\theta}_{n,j_k})\},$$

where $\lambda^{\pm}_{n,i_k}(\tilde{x}_{i_k}; \tilde{\ell}_{n,(i_k,j_k)}, \tilde{\delta}_{n,(i_k,j_k)}, \tilde{\theta}_{n,i_k})$ and $\lambda^{\pm}_{n,j_k}(\tilde{y}_{j_k}; \tilde{\ell}_{n,(i_k,j_k)}, \tilde{\delta}_{n,(i_k,j_k)}, \tilde{\theta}_{n,j_k})$ are regular step functions (see (2.1)) corresponding to the edges from i_k and j_k to n, respectively. When convenient we also use the notation $[\lambda^{\pm}_{n,i_k}(\tilde{x}_{i_k}), \lambda^{\pm}_{n,j_k}(\tilde{y}_{j_k})] := [\tilde{x}_{i_k}, \tilde{y}_{j_k}]$.

Note that we included the parameters to emphasize that $\tilde{\ell}$ and $\tilde{\delta}$ are the same between the functions $\lambda^{\pm}_{n,i_k}$ and $\lambda^{\pm}_{n,j_k}$. As a consequence, both have values in the set $\{\tilde{\ell}_{n,(i_k,j_k)}, \tilde{\ell}_{n,(i_k,j_k)} + \tilde{\delta}_{n,(i_k,j_k)}\}$, and therefore

$$[\tilde{x}_{i_k}, \tilde{y}_{j_k}] = \begin{cases} \tilde{\ell}_{n,(i_k,j_k)} & \text{if } \lambda^{\pm}_{n,i_k}(\tilde{x}_{i_k}) = \tilde{\ell}_{n,(i_k,j_k)} \text{ or } \lambda^{\pm}_{n,j_k}(\tilde{y}_{j_k}) = \tilde{\ell}_{n,(i_k,j_k)}, \\ \tilde{\ell}_{n,(i_k,j_k)} + \tilde{\delta}_{n,(i_k,j_k)} & \text{otherwise.} \end{cases} \tag{3.11}$$

Therefore $[\tilde{x}_{i_k}, \tilde{y}_{j_k}]$ acts like an AND operation on the values of $\lambda^{\pm}_{n,i_k}$ and $\lambda^{\pm}_{n,j_k}$, that is, $[\tilde{x}_{i_k}, \tilde{y}_{j_k}]$ is high if and only if both of those values are high.

The signs are determined by the type of edges as follows. Let $(\tilde{x}_{i_k}, \tilde{y}_{j_k})$ represent a pair of type II edges. If the pair of edges interact as in Figure 1(c) denoted by a solid dot, then

$$\lambda^{\pm}_{n,i_k}(\tilde{x}_{i_k}) = \begin{cases} \lambda^{+}_{n,i_k}(\tilde{x}_{i_k}) & \text{if } i_k \to n, \\ \lambda^{-}_{n,i_k}(\tilde{x}_{i_k}) & \text{if } i_k \dashv n, \end{cases} \quad \text{and} \quad \lambda^{\pm}_{n,j_k}(\tilde{y}_{j_k}) = \begin{cases} \lambda^{+}_{n,j_k}(\tilde{y}_{j_k}) & \text{if } j_k \to n, \\ \lambda^{-}_{n,j_k}(\tilde{y}_{j_k}) & \text{if } j_k \dashv n. \end{cases} \tag{3.12}$$

If the pair of edges interact as in Figure 1(d) denoted by an open dot, then the sign of the second function is flipped, that is,

$$\lambda^{\pm}_{n,i_k}(\tilde{x}_{i_k}) = \begin{cases} \lambda^{+}_{n,i_k}(\tilde{x}_{i_k}) & \text{if } i_k \to n, \\ \lambda^{-}_{n,i_k}(\tilde{x}_{i_k}) & \text{if } i_k \dashv n, \end{cases} \quad \text{and} \quad \lambda^{\pm}_{n,j_k}(\tilde{y}_{j_k}) = \begin{cases} \lambda^{-}_{n,j_k}(\tilde{y}_{j_k}) & \text{if } j_k \to n, \\ \lambda^{+}_{n,j_k}(\tilde{y}_{j_k}) & \text{if } j_k \dashv n. \end{cases} \tag{3.13}$$

Let $f(\tilde{z}_1, \dots, \tilde{z}_{\tilde{K}}, z_1, \dots, z_K)$ be an interaction function of type $(k_1, \dots, k_q)$ and define

$$\Lambda_n(\tilde{w}_1, \dots, \tilde{w}_{\tilde{K}}, x_1, \dots, x_K) = f([\tilde{x}_{i_1}, \tilde{y}_{j_1}], \dots, [\tilde{x}_{i_{\tilde{K}}}, \tilde{y}_{j_{\tilde{K}}}], \lambda^{\pm}_{n,1}(x_1), \dots, \lambda^{\pm}_{n,K}(x_K)). \tag{3.14}$$

Since Λ_n is given in terms of a classical interaction function of type $(k_1, \dots, k_q)$, solving the PSD problem for type II rate expressions reduces to solving the PSD problem corresponding to this classical interaction function.

Before presenting the general result, let us consider some examples. Consider the pair of type II edges from nodes 1 and 2 terminating on node 3 in Figure 3(b). The rate expression for node 3 is the result of an AND operation on the incoming values of nodes 1 and 2, that

is, the incoming value to node 3 is high if and only if the incoming values from nodes 1 and 2 are both high. More precisely,

$$\Lambda_3(x_1,x_2)=[\lambda^+_{3,1}(x_1),\lambda^+_{3,2}(x_2)]=\begin{cases}\ell_{3,(1,2)} & \text{if } x_1<\theta_{3,1} \text{ or } x_2<\theta_{3,2},\\ \ell_{3,(1,2)}+\delta_{3,(1,2)} & \text{otherwise,}\end{cases} \tag{3.15}$$

where $\ell_{3,(1,2)}$ and $\delta_{3,(1,2)}$ are the parameters corresponding to the pair (x_1,x_2). Note that while the network in Figure 3(b) is asymmetric, the net effect on the rate of change of node 2 is symmetric, as expressed in Figure 3(c).

The factor graph of a node having one or more pairs of type II edges as input is computed from a partial order on the input polynomials computed directly from the original DSGRN total orders. Consider, for example, the network in Figure 3(d). The inputs to node 4 are nodes 1, 2, and 3, and the rate expression for node 4 is given by

$$\begin{aligned}\Lambda_4(x_1,x_2,x_3)&=[\lambda^+_{4,1}(x_1),\lambda^+_{4,2}(x_2)]\lambda^-_{4,3}(x_3)\\ &=\left(\begin{cases}\ell_{4,(1,2)} & \text{if } x_1<\theta_{4,1} \text{ or } x_2<\theta_{4,2}\\ \ell_{4,(1,2)}+\delta_{4,(1,2)} & \text{otherwise}\end{cases}\right)\left(\begin{cases}\ell_{4,3}+\delta_{4,3} & \text{if } x_3<\theta_{4,3}\\ \ell_{4,3} & \text{if } x_3>\theta_{4,3}\end{cases}\right),\end{aligned}$$

which takes the following values:

$$\{\ell_{4,(1,2)}\ell_{4,3},\ (\ell_{4,(1,2)}+\delta_{4,(1,2)})\ell_{4,3},\ \ell_{4,(1,2)}(\ell_{4,3}+\delta_{4,3}),\ (\ell_{4,(1,2)}+\delta_{4,(1,2)})(\ell_{4,3}+\delta_{4,3})\}.$$

Notice that these are the values of an interaction function of type $(1,1)$ for the original DSGRN for which the PSD problem has been solved. In particular, define the polynomials

$$\begin{aligned}p_0&:=\ell_{4,(1,2)}\ell_{4,3}, & p_1&:=(\ell_{4,(1,2)}+\delta_{4,(1,2)})\ell_{4,3},\\ p_2&:=\ell_{4,(1,2)}(\ell_{4,3}+\delta_{4,3}), & p_3&:=(\ell_{4,(1,2)}+\delta_{4,(1,2)})(\ell_{4,3}+\delta_{4,3});\end{aligned}$$

then for an arbitrary parameter $\mu=(\ell_{4,(1,2)},\delta_{4,(1,2)},\ell_{4,3},\delta_{4,3})\in(0,\infty)^4$ one of the following orders must be satisfied:

$$p_0(\mu)<p_1(\mu)<p_2(\mu)<p_3(\mu) \quad \text{or} \quad p_0(\mu)<p_2(\mu)<p_1(\mu)<p_3(\mu). \tag{3.16}$$

Hence, the admissible linear orders are given by

$$(0,1,2,3) \quad \text{and} \quad (0,2,1,3),$$

which is identical to the solution of the classical PSD problem for interaction type $(1,1)$.

Even though the choice of identical parameters in $\lambda^+_{4,1}$ and $\lambda^+_{4,2}$ reduces the number of values of $\Lambda_4(x_1,x_2,x_4)$ from 8 to 4, in order to compute the state transition graph we need to be able to determine in which domains these values are attained. In other words, in order to construct the state transition graph we need to be able to determine the sign of (3.10) for the parameter combinations corresponding to all phase space domains. Therefore we need to keep track of all combinations of the values of $\lambda^+_{4,1}(x_1)$, $\lambda^+_{4,2}(x_2)$, and $\lambda^-_{4,3}(x_3)$ and for each determine the value of $\Lambda_4(x_1,x_2,x_3)$. We refer to these combinations as the *input polynomials*

corresponding to $\Lambda_4(x_1, x_2, x_3)$. Since $[\lambda^+_{4,1}(x_1), \lambda^+_{4,2}(x_2)] = \min\{\lambda^+_{4,1}(x_1), \lambda^+_{4,2}(x_2)\}$, the input polynomials corresponding to $\Lambda_4(x_1, x_2, x_3)$ are

$$
\begin{aligned}
q_0 &:= [\ell_{4,(1,2)}, \ell_{4,(1,2)}]\ell_{4,3} = \ell_{4,(1,2)}\ell_{4,3},\\
q_1 &:= [\ell_{4,(1,2)} + \delta_{4,(1,2)}, \ell_{4,(1,2)}]\ell_{4,3} = \ell_{4,(1,2)}\ell_{4,3},\\
q_2 &:= [\ell_{4,(1,2)}, \ell_{4,(1,2)} + \delta_{4,(1,2)}]\ell_{4,3} = \ell_{4,(1,2)}\ell_{4,3},\\
q_3 &:= [\ell_{4,(1,2)} + \delta_{4,(1,2)}, \ell_{4,(1,2)} + \delta_{4,(1,2)}]\ell_{4,3} = (\ell_{4,(1,2)} + \delta_{4,(1,2)})\ell_{4,3},\\
q_4 &:= [\ell_{4,(1,2)}, \ell_{4,(1,2)}](\ell_{4,3} + \delta_{4,3}) = \ell_{4,(1,2)}(\ell_{4,3} + \delta_{4,3}),\\
q_5 &:= [\ell_{4,(1,2)} + \delta_{4,(1,2)}, \ell_{4,(1,2)}](\ell_{4,3} + \delta_{4,3}) = \ell_{4,(1,2)}(\ell_{4,3} + \delta_{4,3}),\\
q_6 &:= [\ell_{4,(1,2)}, \ell_{4,(1,2)} + \delta_{4,(1,2)}](\ell_{4,3} + \delta_{4,3}) = \ell_{4,(1,2)}(\ell_{4,3} + \delta_{4,3}),\\
q_7 &:= [\ell_{4,(1,2)} + \delta_{4,(1,2)}, \ell_{4,(1,2)} + \delta_{4,(1,2)}](\ell_{4,3} + \delta_{4,3}) = (\ell_{4,(1,2)} + \delta_{4,(1,2)})(\ell_{4,3} + \delta_{4,3}).
\end{aligned}
$$

Since these polynomials only attain the four values p_0, p_1, p_2, p_3, from the linear orders (3.16) we get the following list of partial orders:

$$\{q_0, q_1, q_2\} < q_3 < \{q_4, q_5, q_6\} < q_7 \quad \text{and} \quad \{q_0, q_1, q_2\} < \{q_4, q_5, q_6\} < q_3 < q_7.$$

From these partial orders we can compute the factor graph of node 4 as before by considering all possible values of the thresholds $\{\gamma_4\theta_{j,4}\}$ relative to each of these partial orders, with the added restriction that valid configurations cannot have thresholds between q_0, q_1, q_2 or between q_4, q_5, q_6. Observe that the parameter decomposition is symmetric with respect to the input edges of the type II pair of edges.

We now present the general result for type II edges. The proof is a simple generalization of the ideas presented in the above example and it is left to the reader.

Theorem 3.2. *Suppose node n is governed by a type* II *rate expression defined by an interaction function f of order K and type $(k_1, \dots, k_q)$ as defined in (3.14). Then, the PSD problem associated to Λ_n is equivalent to the classical PSD problem of type $(k_1, \dots, k_q)$ associated to the interaction function f.*

Observe that the parameter decomposition and the DSGRN computations are symmetric with respect to the input edges to a type II pair of edges. For this reason internally DSGRN represents pairs of type II edges as in Figure 3(c) and (e).

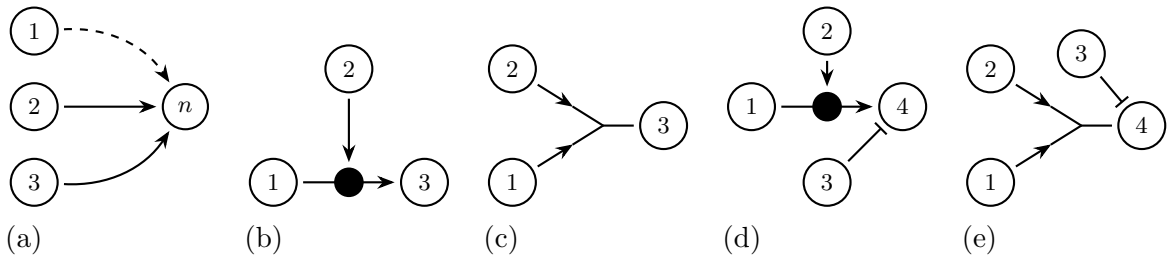

Figure 3. (a) *Example of type* I *regulation.* (b) *Type* II *edges terminating on node* 3. (c) *Implementation of interaction between type* II *edges starting from nodes* 1 *and* 2 *to node* 3. (d) *More complex interaction between type* II *and type* 0 *inputs to node* 4. (e) *Implementation of* (d).

For type II edges we refer to the polynomial expression of f as the *interaction type of a type* II *rate expression*. In addition, we enclose the pairs of type II edges in between square brackets $[\cdot]$ to denote the pair of edges used to define the rate expression (3.14). Notice that the notation $[x, y]$ is symmetric for a pair of type II edges x and y. As an example, the interaction type of the incoming edges to node 4 in Figure 3(e) is denoted by

$$[x_1, x_2]x_3.$$

Observe that the linear orders for this structure are the same as the linear orders (3.16) for the regular DSGRN interaction type

$$z_1 z_2.$$

By (3.12) and (3.13), the only effect of an open dot interaction, compared to full dot interaction, between a pair of type II edges is to change the sign of the second edge, that is, it changes the second edge from $i \to j$ to $i \dashv j$ or from $i \dashv j$ to $i \to j$. In DSGRN this change of sign is denoted by

$$[x_1, \sim x_2].$$

We can compute a complete list of interaction types involving type II edges supported by DSGRN by the following procedure: Take any interaction type from [21, Table 1] or from Table 1 (which in turn is derived from [21, Table 1]) and replace one or more nodes by a pair of nodes representing type II edges (see also section 3.4). The size of the factor graph thus generated is the same as the size of the factor graphs of the corresponding interaction type in the original DSGRN.

For example, the original DSGRN interaction type xy can be used to generate the type I interaction type

$$\langle x \rangle + y$$

and from these we can derive the following interaction types involving type II edges

$$[x_1, x_2]y, \quad [x_1, x_2][y_1, y_2], \quad \langle [x_1, x_2] \rangle + y, \quad \langle x \rangle + [y_1, y_2], \quad \langle [x_1, x_2] \rangle + [y_1, y_2],$$

all of which have factor graphs of the same size as xy.

Table 2 presents a complete list of the interaction types derived from $x(y + z)$ as well as some additional examples of interaction types involving type I and type II edges supported by DSGRN.

3.4. Combination of type I and type II edges. As indicated in the introduction, type I and II edges can be combined. In particular, a pair of type II edges can affect the decay of a node. As a result in a rate expression of type I we can replace one or more of the type I edges by pairs of type II edges. To solve a PSD problem and hence find the linear orders for this type of expression, we first get the linear orders for the corresponding type I expression and then apply the procedure described in section 3.3 that computes the partial orders for the type II edges. Consider, for example, the following interaction type involving type I expression:

$$\langle u + z \rangle + vw.$$

Table 2

Complete list of the interaction types derived from $x(y+z)$ (top) and some additional examples of interaction types with type I *and type* II *edges supported by DSGRN. The sizes of the factor graphs for each of the interaction types are the same as the size of the corresponding original DSGRN interaction type listed.*

PTM interaction type	DSGRN interaction type
$[x_1, x_2](y+z)$	$x(y+z)$
$x([y_1, y_2]+z)$	$x(y+z)$
$[x_1, x_2]([y_1, y_2]+z)$	$x(y+z)$
$x([y_1, y_2]+[z_1, z_2])$	$x(y+z)$
$[x_1, x_2]([y_1, y_2]+[z_1, z_2])$	$x(y+z)$
$\langle x\rangle + ([y_1, y_2]+z)$	$x(y+z)$
$\langle x\rangle + ([y_1, y_2]+[z_1, z_2])$	$x(y+z)$
$\langle [x_1, x_2]\rangle + (y+z)$	$x(y+z)$
$\langle [x_1, x_2]\rangle + ([y_1, y_2]+z)$	$x(y+z)$
$\langle [x_1, x_2]\rangle + ([y_1, y_2]+[z_1, z_2])$	$x(y+z)$
$\langle y+z\rangle + [x_1, x_2]$	$x(y+z)$
$\langle [y_1, y_2]+z\rangle + x$	$x(y+z)$
$\langle [y_1, y_2]+z\rangle + [x_1, x_2]$	$x(y+z)$
$\langle [y_1, y_2]+[z_1, z_2]\rangle + x$	$x(y+z)$
$\langle [y_1, y_2]+[z_1, z_2]\rangle + [x_1, x_2]$	$x(y+z)$
$[x, y]+z$	$x+z$
$[x, y]z$	xz
$\langle x\rangle + [y, z]$	xy
$\langle x\rangle + [y, z]w$	xyw
$\langle x\rangle + [y, z]uw$	$xyuw$
$\langle x\rangle + [y, z][u, v]w$	$xyuw$
$xy([z, w]+u)$	$xy(z+u)$
$\langle xy\rangle + [z, w]+u$	$xy(z+u)$
$\langle xy\rangle + [z, w]+[u, v]$	$xy(z+u)$

We can apply the procedure described in section 3.3 to the linear orders of this interaction type to get the partial orders of the interaction type where we replace the edge u by the type II pair $[x, y]$ to get the partial orders for the interaction type

$$\langle [x, y]+z\rangle + vw.$$

In a similar way we can obtain the linear orders for interaction types involving all three types of edges such as

$$\langle [x, y]+z\rangle + u[v, w].$$

4. Computational examples. In this section we present three examples of small networks with type I and type II edges; see the top row of Figure 4. The bottom row of Figure 4 shows the corresponding networks where type 1 and 2 edges were replaced by type 0 edges.

In Figure 4(a) the $3 \to 1$ and $3 \dashv 2$ are type 0 edges. The edge $1 \to 2$ is a type I edge where node 1 up-regulates the decay of 2. Node 2 changes activity of 1 via a type II edge. The fact that the edge $1 \to 3$ is up-regulating means that the active version of 1 up-regulates 3; the fact that the node on this edge is filled means that the effect of 2 is to change 1 from its inactive form to an active form.

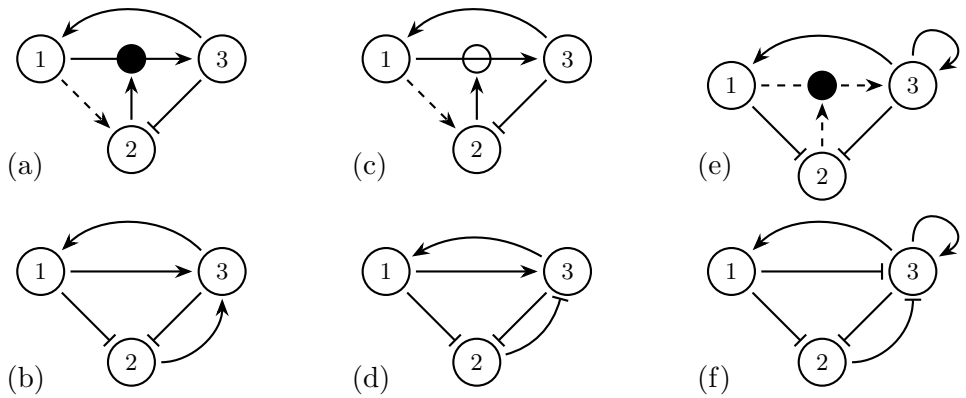

Figure 4. *In each column the top network has type* I *and type* II *edges and at the bottom is the same network interpreted with type* 0 *edges preserving the sign of the interaction. A dashed edge indicates that its end effect is on the decay rate of the receiving network node.*

In Figure 4(b) is the network from (a) with only type 0 edges that captures the same overall effect between the nodes. In particular, since type 1 edge $1 \to 2$ in (a) up-regulates decay, it is replaced by a negative type 0 edge. The type II edges from 1 and 2 that affect (3) are replaced by a pair of positive type 0 edges.

In Figure 4(c) the only change in comparison to (a) is that the node in the $1 \to 3$ edge is empty, which means that 2 facilitates transition from active to inactive form of 1. As a result, the overall effect from 2 to 3 is negative and therefore in Figure 4(d) this influence is represented by a negative type 0 edge.

The last pair of networks is in Figures 4(e) and (f). In (e) node 2 up-regulates the transition from the inactive to the active form of 1, which promotes decay of 3. Therefore the corresponding network in Figure 4(f) has two negative edges $2 \dashv 3$ and $1 \dashv 3$.

As described above for each network we compute the decomposition of parameter space into domains (i.e., parameter nodes) and for each of them we compute the state transition graph and the Morse graph. We characterize Morse nodes by their dynamic phenotypes:

- When a Morse node consists of a single domain κ with a self-edge, we designate it FP (for "Fixed Point").
- When a Morse node consists of a set of domains with a path such that along that path at least one threshold $\theta_{*,i}$ is crossed for every regulatory network vertex i, we designate the Morse node FC (for "Full Cycle").
- If the only paths within a Morse node are such that a proper subset of variables cross thresholds along a path, we designate such Morse node PC (for "Partial Cycle").

We are most interested in describing attractors in dynamics and therefore we concentrate on Morse nodes that are leaves in the Morse graph; we call such Morse nodes stable. Table 3 presents basic statistics on the number of stable Morse nodes (FC, PC, and FP) for the networks in Figure 4.

Comparing paired networks (a)–(b), (c)–(d), and (e)–(f), we observe that the size of the parameter graphs differs. In addition, the number of parameter nodes with stable FC, stable

Table 3

Computations for the networks in Figure 4. The second row from the top shows the number of parameter nodes in the parameter graph of the corresponding network. The remaining rows show the percentage of parameter nodes with the type of dynamics indicated in the first column.

Network	Fig. 4(a)	Fig. 4(b)	Fig. 4(c)	Fig. 4(d)	Fig. 4(e)	Fig. 4(f)
Number of parameters	864	2880	864	2880	21600	305424
Stable FC	1.39%	4.58%	0%	0%	0%	1.49%
Stable PC	4.17%	9.17%	0%	0%	3.83%	7.85%
0 stable FP	5.55%	13.19%	0%	0%	3.07%	7.62%
1 stable FP	90.74%	78.89%	87.04%	74.72%	61.59%	56.41%
2 stable FP	3.70%	7.92%	12.96%	23.61%	33.0%	31.16%
3 stable FP	0%	0%	0%	1.67%	2.33%	4.81%

PC, zero stable FP, one stable FP, bistability, and tristability between FPs are all different. In addition, if these counts are expressed as a percentage of all parameter nodes, the percentages are also all different.

We conclude that the new modality of DSGRN of modeling type I and type II edges in addition to type 0 edges can lead to different results in characterization of network dynamics. When the type I and type II edges describe more closely the underlying biological system, these results may offer a more detailed understanding of the network function.

5. Singular limits of ODE models.

Up to this point in the paper we have consciously avoided writing down explicit ODE models. The motivation for this is twofold: (i) the focus of the paper is on the challenge C2, developing a computational framework in which to identify the global dynamics of complex regulatory networks, and (ii) to emphasize that the results are independent of specific models or applications. However, as indicated in the introduction in the context of applications we must also address the challenge C1. Our primary motivation for developing the DSGRN software has come from systems biology and therefore we return to this subject to identify how a user may make use of the extended DSGRN tools to analyze networks involving transcriptional and post-transcriptional regulation.

Recall that a common ODE model describing the up-regulation of production of the protein of gene n by the protein of gene m has the form

$$\dot{x}_n = -\gamma_n x_n + \ell_{n,m} + \delta_{n,m} \frac{x_m^h}{\theta_{n,m}^h + x_m^h}, \tag{5.1}$$

where the nonlinearity is the classical Hill function. Even this minimal model has a multitude of parameters γ, θ, ℓ, δ, and h. In the search for qualitative insight, it makes sense to sacrifice functional form for ease of analysis. One direction is to consider the limit $h \to \infty$ that results in an equation of the form

$$\dot{x}_n = -\gamma_n x_n + \begin{cases} \ell_{n,m} & \text{if } x_m < \theta_{n,m} \\ \ell_{n,m} + \delta_{n,m} & \text{if } x_m > \theta_{n,m} \end{cases} = -\gamma_n x_n + \lambda^+(x_m; \ell_{n,m}, \delta_{n,m}, \theta_{n,m}), \tag{5.2}$$

where notation λ^+ extends definition (2.1) by including explicit parameter values in the argument. Similarly, a limit $h \to \infty$ of the equation

$$\dot{x}_n = -\gamma_n x_n + \ell_{n,m} + \delta_{n,m} \frac{\theta_{n,m}^h}{\theta_{n,m}^h + x_m^h} \tag{5.3}$$

results in an equation

$$\dot{x}_n = -\gamma_n x_n + \begin{cases} \ell_{n,m} & \text{if } x_m > \theta_{n,m} \\ \ell_{n,m} + \delta_{n,m} & \text{if } x_m < \theta_{n,m} \end{cases} = -\gamma_n x_n + \lambda^-(x_m; \ell_{n,m}, \delta_{n,m}, \theta_{n,m}). \tag{5.4}$$

Models of regulatory networks using equations of the form of (5.2)–(5.4) are typically called switching systems and have been analyzed and applied [24, 10, 6, 25] since their introduction by Glass and Kaufmann [16, 17]. The development of the current version of the DSGRN software was motivated by switching systems. However, there is a fundamental difference in that we are not interested in and do not solve for trajectories of switching system ODEs. Instead, as indicted in section 1 we retreat to a combinatorial model that only makes use of the sign of the right-hand side, e.g., (2.2). The more general Λ_n described in section 2 can all be obtained in a similar manner, by starting with a product of sums of Hill functions, each of which expresses an edge interaction, and allowing the Hill exponents h to become arbitrarily large.

As discussed in [11] and [7] the dynamics captured by the DSGRN computations is highly suggestive of the dynamics exhibited by the ODEs even for moderate levels of exponent h in the Hill functions.

To provide motivation for the expressions for the extended version of DSGRN developed in this paper we turn to the theory of multisite protein control. For simplicity of notation we assume that the protein being controlled is produced by node 1 and that the associated protein, Protein 1, is controlled by Proteins 2 and 3.

P1. Protein 1 has M sites and the activity level of Protein 1 is governed by how many sites are filled. In particular, Protein 1 is off if less than K of the sites are occupied and Protein 1 is on if K or more sites are occupied.

P2. Protein 2 and Protein 3 act as enzymes, with Protein 2 filling sites and Protein 3 emptying sites.

A guiding example for this arrangement may be multisite phosphorylation by a kinase (Protein 2) and dephosphorylation by a phosphatase (Protein 3). We denote the concentration of Proteins 1–3 by x_1, x_2, and x_3, respectively. To be more precise we let $x_1^{(m)}$, $m = 0, \ldots, M$, denote the concentration of Protein 1 that has m sites filled. Motivated by assumption P1 we set

$$x_1^{\mathsf{off}} = \sum_{m=0}^{K-1} x_1^{(m)} \quad \text{and} \quad x_1^{\mathsf{on}} = \sum_{m=K}^{M} x_1^{(m)}.$$

We model assumption P2 by the system of chemical reactions

$$x_1^{(m)} + x_2 \underset{k_-}{\overset{k_+}{\rightleftharpoons}} x_2 x_1^m \xrightarrow{k} x_1^{(m+1)} + x_2$$

$$x_1^{(m+1)} + x_3 \underset{l_-}{\overset{l_+}{\rightleftharpoons}} x_3 x_1^{m+1} \xrightarrow{l} x_1^{(m)} + x_3$$

and set

$$\alpha = \frac{kk_+(l+l_-)}{ll_+(k+k_-)}.$$

Following the analysis in [26][1] we obtain that for $K = 2M-1$, the concentration of the active version x_1^{on} of x_1 is given by

$$x_1^{\mathsf{on}} = \frac{(\alpha\frac{x_2}{x_3})^K}{1+(\alpha\frac{x_2}{x_3})^K}\, x_1. \tag{5.5}$$

Finally, taking the limit as $M \to \infty$ while keeping $M = 2K-1$ allows us to write x_1^{on} as a function of the variable $\alpha\frac{x_2}{x_3}$,

$$x_1^{\mathsf{on}}\left(\alpha\frac{x_2}{x_3}\right) = \sigma^+\left(\alpha\frac{x_2}{x_3};1\right)x_1 = \sigma^+\left(\frac{x_2}{x_3};\frac{1}{\alpha}\right)x_1, \tag{5.6}$$

where

$$\sigma^+(\xi;\zeta) := \begin{cases} 0 & \text{if } \xi < \zeta, \\ 1 & \text{if } \xi > \zeta. \end{cases}$$

5.1. Type I: Decay control edges. As indicated in the introduction we use the analysis from the previous section to model the modulation of the decay rate; see Figure 1(b). In the biological setting, the decay up-regulation is accomplished by ubiquitination by an enzyme ubiquitin ligase, and decay down-regulation by deubiquitination by ubiquitin-specific protease.

Let x_1 denote the total concentration of quantity associated to node 1 that is undergoing up-regulation of its decay rate by x_3 and down-regulation of its decay rate by x_2. We restrict our attention to processes where only one of x_2 and x_3 is actively controlled. We first focus on control of the decay up-regulation and assume the following:

A3.1. *The concentration of decay down-regulator x_3 is a constant y.*

Under assumptions A1, A2, and A3.1 we can rewrite (5.6) as

$$x_1^{\mathsf{on}}(x_2) = \sigma^+(x_2;\beta_{1,2})x_1, \tag{5.7}$$

where $\beta_{1,2} = y/\alpha$.

To incorporate this into DSGRN we assume that the decay rate for x_1 in the absence of x_2 is γ_1. Thus (5.2) becomes

$$\begin{aligned}\dot{x}_1 &= -\gamma_1 x_1 + \delta_{1,2}\sigma^+(x_2;\beta_{1,2})x_1 + \Lambda_1(x) = -\left(\begin{cases}\gamma_1 & \text{if } x_2 < \beta_{1,2} \\ \gamma_1+\delta_{1,2} & \text{if } x_2 > \beta_{1,2}\end{cases}\right)x_1 + \Lambda_1(x) \\ &= -\lambda^+(x_2;\gamma_1,\delta_{1,2},\beta_{1,2})x_1 + \Lambda_1(x).\end{aligned}$$

We model decay down-regulation in a similar manner. We make an assumption:

A3.2. *The concentration of decay up-regulator x_2 is constant y.*

[1]This paper also provides biological relevance for this modeling approach.

Under assumptions A1, A2, and A3.2 can rewrite (5.6) as

$$x_1^{\mathsf{off}}(x_3) = \left(1 - \sigma^+(x_3; \beta_{1,3})\right) x_1, \tag{5.8}$$

where $\beta_{1,3} = \alpha y$. We rewrite (5.8) one more time as

$$x_1^{\mathsf{off}}(x_3) = \sigma^-(x_3; \beta_{1,3}) x_1, \tag{5.9}$$

where

$$\sigma^-(\xi; \zeta) := \begin{cases} 1 & \text{if } \xi < \zeta, \\ 0 & \text{if } \xi > \zeta. \end{cases}$$

To incorporate this into DSGRN we assume that the decay rate for Protein 1 in the absence of x_3 is γ_1. Then (5.2) becomes

$$\begin{aligned} \dot{x}_1 &= -(\gamma_1 + \delta_{1,3}\sigma^-(x_3; \beta_{1,3}))x_1 + \Lambda_1(x) = -\left(\begin{cases} \gamma_1 & \text{if } x_3 > \beta_{1,3} \\ \gamma_1 + \delta_{1,3} & \text{if } x_3 < \beta_{1,3} \end{cases}\right) x_1 + \Lambda_1(x) \\ &= -\lambda^-(x_3; \gamma_1, \delta_{1,3}, \beta_{1,3}) x_1 + \Lambda_1(x). \end{aligned}$$

5.2. Type II: Activity control edges. We now turn our attention to modeling modulation of activity level; see Figures 1(c)–(d). In the biological context there are several modifications that affect activity of a protein. This is often achieved by binding an additional molecule or a group to an existing protein and modifying its properties. Examples include phosphorylation, methylation, glycosylation, lipidation, and other modifications.

We will first consider interactions shown in Figure 1(c) where the x_1^{on} is the active form of x_1. In the same way as we did for the decay control processes, we assume that only one of the processes controlling the activation versus deactivation is actively controlled. We start with the following:

A3.3. *The concentration x_3 remains constant.*

Under assumptions A1, A2, and A3.3 we can rewrite (5.6) as

$$x_1^{\mathsf{on}}(x_2) = \sigma^+(x_2; \beta_{1,2}) x_1, \tag{5.10}$$

where $\beta_{1,2} = x_3/\alpha$. On the other hand, if we assume

A3.4. *The concentration x_2 remains constant,*

then, under assumptions A1, A2, and A3.4 we can rewrite (5.6) as

$$x_1^{\mathsf{on}}(x_3) = \sigma^+\left(\frac{1}{x_3}; \frac{1}{\beta_{1,3}}\right) x_1, = \sigma^-(x_3; \beta_{1,3}) x_1, \tag{5.11}$$

where $\beta_{1,3} = \alpha x_2$.

To incorporate this into DSGRN we assume that the production rates for x_4 by x_1, when x_1 is in the inactive state off and active state on, are given by $\ell_{4,1}$ and $\ell_{4,1} + \delta_{4,1}$, respectively.

Then (5.2) for $n = 4$ is

$$\begin{aligned} \dot{x}_4 &= -\gamma_4 x_4 + \Lambda_4(x_1) \\ &= -\gamma_4 x_4 + \begin{cases} \ell_{4,1} & \text{if } x_1 < \theta_{4,1}, \\ \ell_{4,1} + \delta_{4,1} & \text{if } x_1 > \theta_{4,1}. \end{cases} \end{aligned} \tag{5.12}$$

Under the assumption A3.3 this becomes

$$\begin{aligned}\dot{x}_4 &= -\gamma_4 x_4 + \Lambda_4(\sigma^+(x_2;\beta_{1,2})x_1)\\ &= -\gamma_4 x_4 + \begin{cases}\ell_{4,1} & \text{if } x_2 < \beta_{1,2},\\ \ell_{4,1} & \text{if } x_1 < \theta_{4,1},\\ \ell_{4,1}+\delta_{4,1} & \text{if } x_1 > \theta_{4,1} \text{ and } x_2 > \beta_{1,2},\end{cases}\\ &= -\gamma_4 x_4 + [\lambda^+(x_1), \lambda^+(x_2)],\end{aligned}$$

where notation $[\cdot,\cdot]$ has been introduced in (3.15).

On the other hand, under the assumption A3.4 we have

$$\begin{aligned}\dot{x}_4 &= -\gamma_4 x_4 + \Lambda_4(\sigma^-(x_3;\beta_{1,3})x_1)\\ &= -\gamma_4 x_4 + \begin{cases}\ell_{4,1} & \text{if } x_3 > \beta_{1,3},\\ \ell_{4,1} & \text{if } x_1 < \theta_{4,1},\\ \ell_{4,1}+\delta_{4,1} & \text{if } x_1 > \theta_{4,1} \text{ and } x_3 < \beta_{1,3},\end{cases}\\ &= -\gamma_4 x_4 + [\lambda^+(x_1), \lambda^-(x_3)].\end{aligned}$$

For the second type of interactions in Figure 1(c), (5.12) has reversed inequalities with respect to threshold $\theta_{4,1}$,

$$\dot{x}_4 = -\gamma_4 x_4 + \begin{cases}\ell_{4,1} & \text{if } x_1 > \theta_{4,1},\\ \ell_{4,1}+\delta_{4,1} & \text{if } x_1 < \theta_{4,1}.\end{cases}$$

This change of inequality persists into the functions $\Lambda_4(\sigma^+(x_2;\beta_{1,2})x_1)$ under the assumption A3.3 and to function $\Lambda_4(\sigma^-(x_3;\beta_{1,3})x_1)$ under the assumption A3.4, resulting in

$$\dot{x}_4 = -\gamma_4 x_4 + [\lambda^-(x_1), \lambda^+(x_2)]$$

and

$$\dot{x}_4 = -\gamma_4 x_4 + [\lambda^-(x_1), \lambda^-(x_3)],$$

respectively.

For interactions in the left side of Figure 1(d) it is the x_1^{off} form of x_1 that activates production of x_4. Since $x_1^{\mathsf{on}} + x_1^{\mathsf{off}} = x_1$ under the assumption A3.3 we get

$$\dot{x}_4 = -\gamma_4 x_4 + \Lambda_4(\sigma^-(x_2;\beta_{1,2})x_1) = -\gamma_4 x_4 + [\lambda^+(x_1), \lambda^-(x_2)],$$

and under the assumption A3.3

$$\dot{x}_4 = -\gamma_4 x_4 + \Lambda_4(\sigma^+(x_3;\beta_{1,3})x_1 = -\gamma_4 x_4 + [\lambda^+(x_1), \lambda^+(x_3)].$$

Analogous to Figure 1(c), interactions indicated by the right half of Figure 1(d) will produce functions

$$[\lambda^-(x_1), \lambda^-(x_2)] \quad \text{and} \quad [\lambda^-(x_1), \lambda^+(x_3)],$$

respectively.

We close this section with a remark that when y_1^{on} regulates activity, rather than abundance, of protein y_2, which in turn regulates activity of y_3 along a chain of interactions until y_{n-1}, which eventually regulates transcription of y_n, then the equation for y_n will be

$$\dot{y}_n = -\gamma_n y_n + \Lambda((\sigma_{n-1}^{\pm} \circ \cdots \circ \sigma_1^{\pm})(y_1^{\mathsf{on}}; \beta_{1,2}) y_n), \tag{5.13}$$

where $(\sigma_{n-1}^{\pm} \circ \cdots \circ \sigma_1^{\pm})(y_1^{\mathsf{on}}; \beta_{1,2})$ is the $(n-1)$-fold composition of functions $\sigma^+(\,\cdot\,; \beta_{1,2})$ or $\sigma^-(\,\cdot\,; \beta_{1,2})$, as functions of the first argument for a fixed value of the parameter $\beta_{1,2}$, depending on the type of control in Figure 1(c) or (d).

We generalize the function

$$\Gamma(\zeta_1, \ldots, \zeta_n) := \min\{\zeta_1, \ldots, \zeta_n\}$$

and (note identical values of ℓ and δ) set

$$[\lambda(z_1), \lambda(z_2), \ldots, \lambda(z_n))] := \Gamma(\lambda(z_1, \ell, \delta, \theta_1), \lambda(z_2, \ell, \delta, \theta_2), \ldots, \lambda(z_n, \ell, \delta, \theta_n)).$$

Then (5.13) becomes

$$\dot{y}_n = -\gamma_n y_n + [\lambda^{\pm}(y_1), \lambda^{\pm}(y_2), \ldots, \lambda^{\pm}(y_n)],$$

where the sign of the function $\lambda^{\pm}(y_j)$ matches the sign of $\sigma_j^{\pm}(y_j; \beta_{j,j+1})$ for $j = 1, \ldots, n-1$, and the sign of $\lambda^{\pm}(y_n)$ matches the sign of the function $\Lambda(y_n)$. Similarly, the threshold parameters θ_j, $j = 1, \ldots, n-1$, are the activation parameters $\beta_{j,j+1}$ and θ_n is the threshold of the function $\Lambda(y_n)$.

6. Conclusions.

Networks are a useful abstraction expressing partial knowledge about internal correlation (undirected edges) or causal (directed edges) structure of complex systems. In addition, in gene regulatory networks edges are directed and signed, where sign denotes up- or down-regulation. While networks sometimes express static information like correlations, often one is interested in the dynamical behavior of the system described by the network. One of the ways to associate dynamics to a network is to represent each vertex by a continuous variable with a linear decay rate and each edge by a (nonlinear) monotone function, where the sign of the derivative matches the sign of the edge, and study a set of ODEs with this structure. One can view such a system as one where variables represent *abundance* with a rate of change responding to associated in-edges. In the context of gene regulation, this type of model corresponds to transcriptional regulation of genes, where an increase in the concentration of activators increases the rate of production from a particular gene, and an increase in the concentration of repressors decreases such a rate.

Understanding the dynamics of such systems, especially in systems with several to dozens of variables, is notoriously difficult. In particular, the dynamics of network models can vary widely with selection of nonlinearities as well as parameters, which are mostly unknown and lie in high dimensional space. Motivated by gene networks, and building on previous work on Boolean networks and switching systems [1, 17, 24, 19, 10, 6], we developed DSGRN. DSGRN assigns combinatorial dynamics to a network, of a type depending on a finite decomposition of the parameter space [5, 4, 15, 11, 13, 14]. The finiteness of this calculation allows complete

enumeration of types of dynamics compatible with the network. However, in spite of the finiteness of representation, the computed invariant, the Morse graph, is valid for a large class of ODE models. In particular, it has been shown that the Morse graph provides information about a Morse decomposition of nearby smooth systems of differential equations in two dimensions [15]; the generalization of this result to higher dimensions is forthcoming. In the case where the Morse nodes indicate the presence of stable equilibria, these equilibria do exist for nearby smooth differential equations [9, 8].

Again motivated by gene regulation, this paper extends the combinatorial DSGRN approach to a significantly larger class of network interactions. In cellular regulatory networks, the abundance of a particular protein may be constant, but its *activity* may be carefully regulated by, say, phosphorylation, methylation, or other type of post-transcriptional or post-translational regulation. In addition, the decay of a protein may be actively regulated as well. We use modeling work on multisite phosphorylation and ubiquitination to derive an appropriate combinatorial model for activity and decay regulation. DSGRN relies on precomputed logic files that encode all linear orders of a set of polynomials associated to a network vertex with a particular number of in- and out-edges. Based on our analysis we show how we modify and then use these precomputed logic files to support combinatorial model for activity and decay regulation.

Finally, we provide a comparison of dynamics between networks that include either activity or decay regulation, and the corresponding networks with the same types of edges but that only regulate abundance. It is clear from these results that the dynamics can be very different and therefore the new capability will allow a more precise delineation of network dynamics in a wide range of applications.

Acknowledgments. B.C. and T.G. acknowledge the Indigenous nations and peoples who are the traditional owners and caretakers of the land on which this work was undertaken at Montana State University.

REFERENCES

[1] R. Albert, J. J. Collins, and L. Glass, *Introduction to Focus Issue: Quantitative approaches to genetic networks*, Chaos, 23 (2013), 025001.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed., MIT Press, Cambridge, MA, 2001.

[3] P. Crawford-Kahrl, B. Cummins, and T. Gedeon, *Joint Realizability of Monotone Boolean Functions*, under review.

[4] B. Cummins, T. Gedeon, S. Harker, and K. Mischaikow, *Database of dynamic signatures generated by regulatory networks (DSGRN)*, in Computational Methods in Systems Biology - 2017, J. F. H. Koeppl, ed., Springer, New York, 2017, pp. 300–308, https://doi.org/10.1007/978-3-319-67471-1.

[5] B. Cummins, T. Gedeon, S. Harker, K. Mischaikow, and K. Mok, *Combinatorial representation of parameter space for switching networks*, SIAM J. Appl. Dyn. Syst., 15 (2016), pp. 2176–2212, https://doi.org/10.1137/15m1052743.

[6] H. de Jong, *Modeling and simulation of genetic regulatory systems: A literature review*, J. Comput. Biol., 9 (2002), pp. 67–103, https://doi.org/10.1089/10665270252833208.

[7] R. Diegmiller, L. Zhang, M. Gameiro, J. Barr, J. Imran Alsous, P. Schedl, S. Y. Shvartsman, and K. Mischaikow, *Mapping parameter spaces of biological switches*, PLOS Comput. Biol., 17 (2021), pp. 1–19, https://doi.org/10.1371/journal.pcbi.1008711.

[8] W. Duncan, and T. Gedeon, *Stability and bifurcations of equilibria in networks with piecewise linear interactions*, Internat. J. Bifur. Chaos, 31 (2021), 2130032.
[9] W. Duncan, T. Gedeon, H. Kokubu, K. Mischaikow, and H. Oka, *Equilibria and their stability in networks with steep sigmoidal nonlinearities*, SIAM J. Appl. Dyn. Syst., 20 (2021), pp. 2108–2141.
[10] R. Edwards, *Chaos in neural and gene networks with hard switching*, Differ. Equ. Dyn. Syst., 9 (2001), pp. 187–220.
[11] M. Gameiro, T. Gedeon, S. Kepley, and K. Mischaikow, *Rational Design of Complex Phenotype via Network Models*, https://arxiv.org/abs/2010.03803, 2020.
[12] M. Gameiro, K. Mischaikow, and A. Zheleznyak, *Parameter Decomposition of Regulatory Networks with Multiple Thresholds*, in preparation.
[13] T. Gedeon, *Multi-parameter exploration of dynamics of regulatory networks*, BioSystems, 190 (2020), p. 1045113.
[14] T. Gedeon, B. Cummins, S. Harker, and K. Mischaikow, *Identifying robust hysteresis in networks*, PLOS Comput. Biol., 14 (2018), pp. 1–23, https://doi.org/10.1371/journal.pcbi.1006121.
[15] T. Gedeon, S. Harker, H. Kokubu, K. Mischaikow, and H. Oka, *Global dynamics for steep sigmoidal nonlinearities in two dimensions*, Phys. D, 339 (2017), pp. 18–38.
[16] V. L. Ginzburg and L. D. Landau, *On the theory of superconductivity*, J. ETP, 20 (1965), pp. 1064–1082.
[17] L. Glass and S. A. Kauffman, *Co-operative components, spatial localization and oscillatory cellular dynamics*, Journal of Theoretical Biology, 34 (1972), pp. 219–237, https://doi.org/10.1016/0022-5193(72)90157-9.
[18] A. Hershko, A. Ciechanover, and I. Rose, *Resolution of the ATP-dependent proteolytic system from reticulocytes: A component that interacts with ATP*, Proc. Natl. Acad. Sci. USA, 76 (1979), pp. 3107–3110.
[19] L. Ironi, L. Panzeri, E. Plahte, and V. Simoncini, *Dynamics of actively regulated gene networks*, Phys. D, 240 (2011), pp. 779–794, https://doi.org/10.1016/j.physd.2010.12.010.
[20] W. Kalies, K. Mischaikow, and R. Vandervorst, *Lattice structures for attractors* I, J. Comput. Dyn., 1 (2014).
[21] S. Kepley, K. Mischaikow, and L. Zhang, *Computing linear extensions for polynomial posets subject to algebraic constraints*, SIAM J. Appl. Algebra Geom., 5 (2021), pp. 388–416, https://doi.org/10.1137/20M1343208.
[22] *Nobel Prize in Chemistry,* 2004*, Aaron Ciechanover, Avram Hershko and Irwin Rose*, Indian J. Physiol. Pharmacol., 49 (2005), p. 121.
[23] C. Ricci-Tam, I. Been-Zion, J. Wang, J. Palme, A. Li, Y. Savir, and M. Springer, *Decoupling transcription factor expression and activity enables dimmer switch regulation*, Science, 372 (2021), pp. 292–295.
[24] R. Thomas, *Regulatory networks seen as asynchronous automata: A logical description*, J. Theoret. Biol., 153 (1991), pp. 1–23, https://doi.org/10.1016/S0022-5193(05)80350-9.
[25] S. R. Veflingstad and E. Plahte, *Analysis of gene regulatory network models with graded and binary transcriptional responses*, Biosystems, 90 (2007), pp. 323–339, https://doi.org/10.1016/j.biosystems.2006.09.036.
[26] L. Wang, Q. Nie, and G. Enciso, *Nonessential sites improve phosphorylation switch*, Biophys. J., 99 (2010), pp. L41–L43.
[27] Y. Xin, B. Cummins, and T. Gedeon, *Multistability in the epithelial-mesenchymal transition network*, BMC Bioinformatics, 21 (2020), https://doi.org/10.1186/s12859-020-3413-1.