Published in final edited form as:

SIAM J Appl Dyn Syst. 2018 ; 17(2): 1589–1616. doi:10.1137/17M1134548.

Model Rejection and Parameter Reduction via Time Series

Bree Cummins[†], **Tomas Gedeon**[†], **Shaun Harker**[‡], **Konstantin Mischaikow**[‡]

Bree Cummins: cummins@math.montana.edu; Tomas Gedeon: gedeon@math.montana.edu; Shaun Harker: sharker@math.rutgers.edu; Konstantin Mischaikow: mischaik@math.rutgers.edu

[†]Department of Mathematical Sciences, Montana State University, Bozeman, MT 59715.

[‡]Department of Mathematics, Hill Center-Busch Campus, Rutgers, The State University of New Jersey, Piscataway, NJ 08854-8019.

Abstract

We show how a graph algorithm for finding matching labeled paths in pairs of labeled directed graphs can be used to perform model invalidation for a class of dynamical systems including regulatory network models of relevance to systems biology. In particular, given a partial order of events describing local minima and local maxima of observed quantities from experimental time series data, we produce a labeled directed graph we call the *pattern graph* for which every path from root to leaf corresponds to a plausible sequence of events. We then consider the regulatory network model, which can itself be rendered into a labeled directed graph we call the *search graph* via techniques previously developed in computational dynamics. Labels on the pattern graph correspond to experimentally observed events, while labels on the search graph correspond to mathematical facts about the model. We give a theoretical guarantee that failing to find a match invalidates the model. As an application we consider gene regulatory models for the yeast *S. cerevisiae*.

Keywords

regulatory networks; switching systems; time series; dynamics

AMS subject classifications.

37N25; 37N30

1. Introduction.

One of the fundamental challenges, as we move toward an era of data driven science, is how to make use of imprecise data to select or reject models and parameters that cannot be derived from first principles. Motivated by problems from systems biology we address this challenge in the context of oscillatory data under the assumption that reasonable models prescribe appropriate local behavior of trajectories. We adopt the following strategy. From experimental time series data we extract a partial order of events describing minima and maxima of observed quantities. On the modeling side, as a function of parameters, we construct a directed graph to catalogue the possible dynamics. The main result of this paper is an efficient algorithm to identify if the model dynamics is capable of exhibiting sequences of minima and maxima that are consistent with the experimental data. Failure can then be used for model rejection or parameter reduction.

To provide more detail we consider a particular example. High throughput experimental technology is making the collection of time series of gene expression a routine process. However, this data is noisy, often contains significant measurement error, is typically collected at a coarse time scale, and generally is collected over a relatively short time span. In an attempt to extract robust information from such data we focus on the ordering of extremal events. This paper does not address the difficulty of detecting and eliminating spurious pairs of extrema in data—a challenging problem in its own right. Instead, we assume that a statistically valid procedure is used to identify or impose time intervals during which a local maximum or minimum has occurred. This renders the time series into a set of extremal events where the error bounds determine the time intervals associated with the individual maxima and minima. If two intervals do not overlap, then we can distinguish the relative timing between the associated events, but if they do overlap, we cannot. Therefore, we represent relative timing as a partially ordered set (poset) that we call the *poset of extrema* (see Definition 3.1). We assume, however, that there is a linear temporal ordering along which the extrema occur and our lack of knowledge is due to experimental constraints. Consequently, we adopt the hypothesis that one of the linear extensions of the poset of extrema represents the correct sequence of events.

Because gene expression data is noisy and often collected at a coarse time scale, in practice there are many nodes in the poset of extrema that are incomparable (i.e., the time ordering cannot be resolved from the data). Roughly speaking, each such ambiguity leads to an additional multiplicative factor in the number of possible linear extensions. Therefore, in general we expect that the set of all linear extensions will be large—in fact, exponential in the number of genes and exponential in the length of the time series. To overcome this ostensibly intractable situation we construct a labeled directed acyclic graph that we call the *pattern graph* (see Definition 3.2), which gives a compressed representation of the set of all linear extensions. This directed graph is given by the transitive reduction of the lattice of down sets of the poset of extrema, with labels on nodes and edges identifying which variables are increasing or decreasing or have reached extrema. As explained in section 2.3.1, for a fixed number of genes the time to compute the pattern graph is only polynomial in the length of the time series, and the resulting labeled directed acyclic graph may be stored in linear space. This gives rise to efficient (i.e., polynomial time) algorithms for pattern matching among the exponential number of linear extensions.

The poset of extrema and the pattern graph represents the codification of the experimental data. Viewed abstractly this is just a means of formally capturing potential temporal ordering of experimentally observable phenomena and therefore these ideas are potentially applicable to a wide variety of problems within and outside of the life sciences.

Returning to the example of gene regulation, we observe that this is an extremely complex multiscale process and thus it is not reasonable to postulate a precise nonlinear model that describes its behavior. However, as indicated above, we assume that we can make assumptions concerning the local qualitative behavior of the dynamics. With this in mind we introduce the following notion.

Definition 1.1.

A system of trajectories $\mathcal{ST}$ *on a space X is a collection of continuous functions from closed intervals to X, i.e., x*: $[a, b] \to X$ *for some* $a, b \in R$, *called trajectories, such that*

1. *the restriction of any trajectory to a smaller closed interval is again a trajectory*;
2. *a concatenation of trajectories is again a trajectory (more precisely, if* $x: [a, b] \to X$ *and* $y: [c, d] \to X$ *are trajectories such that* $x(b) = y(c)$, *then there exists a trajectory* $z: [0, (b - a) + (d - c)] \to X$ *and any time translation of z, that agrees with x on* $[0, b - a]$ and y on $[b - a, (b - a) + (d - c)]$);
3. *the time translation, i.e.,* $x(t - \Delta t)$, *of any trajectory is again a trajectory; and*
4. *every map* $x: \{0\} \to X$ *is a trajectory.*

Notice that the set of solutions to a smooth system of ordinary differential equations can be written as a system of trajectories.

A heuristic description of how we employ this concept (see section 3.2 for formal definitions) is as follows. The phase space X is decomposed into a finite number of rectangular domains $\mathcal{X}$. Restricted to each of these rectangular domains each individual trajectory is monotone in every variable (note that even within a domain we are not assuming the same directions of monotonicity for each trajectory). The boundaries of the domains are called *walls*. On the walls the trajectories are allowed to achieve a local extremum with respect to at most one variable, which we call an *extremal event*. A system of trajectories that satisfies these conditions is called *extrema-pattern-matchable* with respect to $\mathcal{X}$. The dynamics is then recorded as a labeled directed graph, called the *search graph* (see Definition 3.6), where vertices correspond to domains and edges are determined by wall trajectories. The labeling acts on nodes and edges and is used to codify our knowledge as to which variables are increasing or decreasing or have reached extrema. Thus, the search graph formalizes the structure of the dynamics that can be expressed by a model that generates the system of trajectories.

The content of this paper is the development of an efficient means of comparing the model dynamics against the experimentally observed dynamics. The fundamental result is Theorem 2.5, which provides a polynomial time algorithm for matching maximal paths in the pattern graph, i.e., a specific ordering of extrema events that is compatible with the experimental data, to paths in the search graph, i.e., trajectories that are realizable by the model for the dynamics, where the matching preserves the labeling. A simplistic description of the applicability of this result is as follows: if the algorithm fails to produce a matching, then this provides a guarantee that the dynamics incorporated in the system of trajectories is incapable of reproducing the sequences of extrema that are compatible with the data. As a consequence we can reject the associated model of dynamics.

Of course, to apply these ideas to realistic problems is much more challenging. While we do not know a particular model that describes the observed dynamics we assume that the dynamics can be modeled by an unknown nonlinear system. One of the fundamental lessons of the theory of dynamical systems is that structure of invariant sets of nonlinear systems can change dramatically as a function of parameters. Thus, to achieve the claims of the title of this paper, we need both a systematic method for generating parameterized models and their associated search graphs, and a robust finite characterization of dynamics that can be computed over all parameter values. For this we make use of the recently developed Dynamics Signatures Generated by Regulatory Network (DSGRN) framework and software [4, 15].

The starting point for DSGRN is a regulatory network **RN** (see Definition 4.1). In the context of gene regulation this is an annotated directed graph in which the nodes represent genes, edges indicate the interaction between the genes, and the annotation indicates if the interaction involves activation or repression. It is also assumed that a logic, dictating how information received at each node is processed, is given. Given a regulatory network with N nodes and $|E|$ edges DSGRN represents dynamics occurring on the phase space $X = (0, \infty)^N$ where the dynamics is parameterized by an $N + 3 \cdot |E|$ dimensional set $Z \subset (0, \infty)^{N+3\cdot|E|}$. In particular, DSGRN computes a finite decomposition of Z where the individual regions are given by explicit semialgebraic sets. DSGRN represents parameter space via an undirected graph PG, called the *parameter graph*, where the nodes correspond to the above mentioned regions of Z and edges provide adjacency information. The dynamics is represented by a directed graph, called a *state transition graph*, derived from a parameter dependent rectangular decomposition of X. A fundamental fact is that as a function of parameters the state transition graph is constant over nodes of PG, i.e., it does not change on the individual regions of the decomposition of parameter space. The state transition graph can be large, and therefore DSGRN condenses the information into a directed acyclic graph MG, called a *Morse graph*. The nodes of the Morse graph correspond to maximal recurrent subgraphs of the state transition graph and the edges indicate reachability, via the state transition graph, from one Morse node to another. The output of DSGRN is called the DSGRN database, which is organized around the parameter graph PG. In particular, for each node in PG the database provides the explicit semialgebraic set in parameter space and the associated Morse graph MG.

Observe that, as desired, DSGRN provides us with a finite description of global dynamics over parameter space. However, the dynamics is described in combinatorial terms and we would like to argue that we are comparing the ordering extrema of continuous trajectories against experimental data. To make this comparison as transparent as possible we use the information encoded in the state transition graph to construct a particularly simple system of trajectories $\mathcal{ST}_{\mathrm{sw}}$ based on classical switching system models [5, 6, 7, 8, 11, 12, 13, 14].

However, we claim that the results for the switching systems immediately extend to a broader class of models. In particular, as is made explicit in the proof of Theorem 4.8, the qualitative properties of the trajectories of $\mathcal{ST}_{\mathrm{sw}}$ alone are sufficient to obtain an extremepattern-matchable system of trajectories. Thus, our results apply any model that produces trajectories with the same monotonicity properties as that of the switching system.

This paper is organized as follows. In section 2 we begin by recalling ideas from and establishing notation associated with graphs and posets. We present the algorithms that underlie our approach to matching model dynamics with experimental data and provide worst case complexity bounds for these algorithms.

In section 3 we provide combinatorial formalizations of the experimental data, the dynamics of the models, and the relation that allows us to compare them. To be more specific, in section 3.1 we show how experimental data can furnish a *poset of extrema P*. Interest in the set of all linear extensions leads us (by Theorem 2.10) to construct the down set graph of the poset of extrema. We label each down set according to whether a function that has experienced the events in the down set but not the events not included in the down set is increasing or decreasing in each variable. We label the edges in the down set graph (which are of the form $A \to A \cup \{p\}$) according to the extremal event associated with $p \in P$. We also introduce self-edges that are labeled as not experiencing any extremal events. We call the resulting labeled directed graph the *pattern graph*.

In section 3.2, we describe a class of dynamical models for which we can characterize possible trajectories in a combinatorial manner via a *domain graph*. A domain graph discretizes a dynamical system by giving a finite set of domains separated by codimension-1 walls. Vertices in the domain graph correspond to domains, and edges correspond to flow from one domain to another via a wall. We label the vertices of the domain graph according to whether the coordinate functions $x_i(t)$ are increasing, decreasing, or possibly both, in the associated domains. We label the edges of the graph according to which local minima or local maxima could occur on the associated walls. We call the resulting labeled directed graph a *search graph*.

In section 3.3, we present a matching relation between the pattern graph and the search graph. We prove (Theorem 3.8) that if there does not exist a match between a path from root to leaf of the pattern graph and a path in the search graph, then the dynamical model underlying the search graph is incompatible with the experimental observations leading to the pattern graph. Theorem 2.5 shows that we can decide whether such a match exists in polynomial time.

Finally, in section 4 we show how these ideas can be applied. We begin in section 4.1 with a brief review of the mathematical structure underlying DSGRN. In section 4.2 we provide a simple example of how one can pass from experimental time series data to a labeled pattern graph. Finally, in section 4.3 we apply these techniques to a simple wavepool model [19] for the metabolic cycle in *S. cerevisiae*. Courtesy of the Haase lab [18] we have experimental time series data for mRNA sequences associated with the genes SWI4, HCM1, NDD1, and YOX1 collected at time intervals of 5 minutes (see Figure 4). We take a biologically implausible model and show that our proposed techniques reject it and we take a model that is biologically acceptable and use our techniques to greatly constrain relations between parameters.

2. Graph theory and algorithms.

2.1. Matching paths in labeled graphs.

Definition 2.1.—*Given a finite set* Σ, *we denote by* Σ^n *the set of n-tuples consisting of elements of* Σ. *We denote by* Σ^* *the set of all finite tuples of elements of* Σ, *i.e.,* $\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$.

For the purpose of this paper a *directed graph* $G = (V, E)$ consists of a finite set of *vertices* V and *edges* $E \subset V \times V$. A *path in G* from $s \in V$ to $t \in V$ is a finite sequence of vertices $(s = v_1, v_2, \ldots, v_n = t)$ such that $v_i \in V$ and $(v_i, v_{i+1}) \in E$. We denote the set of all such paths by $G[s \rightsquigarrow t]$.

Definition 2.2.—*A labeled directed graph G is a quadruple* (V, E, Σ, ℓ) *where V and E denote the vertices and edges of G,* Σ *is a finite set called labels, and* $\ell: V \cup E \to \Sigma$ *is called a labeling function. Given a path* $p = (v_1, \ldots, v_n)$ *in G, the associated labeling is defined to be*

$$L(p) := \left(\ell(v_1), \ell((v_1, v_2)), \ell(v_2), \ldots, \ell((v_{n-1}, v_n)), \ell(v_n)\right) \in \Sigma^*.$$

Definition 2.3.—*A matching relation between two labeled directed graphs* $G = (V, E, \Sigma, \ell)$ *and* $G' = (V', E', \Sigma', \ell')$ *is a choice of a relation between the label sets* Σ *and* Σ'. *To indicate that the labels* $a \in \Sigma$ *and* $b \in \Sigma'$ *match, i.e., are related, we write* $a \smile b$. *We extend the matching relation* $\smile$ *onto the tuples of labels* Σ^* *and* Σ'^* *via*

$$(a_1, a_2, \ldots, a_n) \smile (b_1, b_2, \ldots, b_m) \; iff \; n = m \; and \; for \; 1 \le i \le n, a_i \smile b_i.$$

Given a matching relation, a path $p = (v_1, \ldots, v_n)$ *in* $G[s \rightsquigarrow t]$, *and a path* $p' = (v'_1, \ldots, v'_m)$ *in* $G'[s' \rightsquigarrow t']$, *we say that p matches p′ and write* $p \smile p'$ *whenever* $L(p) \rightsquigarrow L(p')$. *Note that we are using the same symbol* $\smile$ *to refer to three matching relations: between* Σ *and* Σ', *between* Σ^* *and* Σ'^*, *and between paths in* $G[s \rightsquigarrow t]$ *and paths in* $G'[s' \rightsquigarrow t']$.

Definition 2.4.—*Let* $\smile$ *be a matching relation between two labeled directed graphs* $G = (V, E, \Sigma, \ell)$ *and* $G' = (V', E', \Sigma', \ell')$. *Suppose* $s, t \in V$ *and* $s', t' \in V'$. *The alignment problem Alignment*$(G, G', \smile, (s, t), (s', t'))$ *is the decision problem of determining if there is a pair of paths* $p \in G[s \rightsquigarrow t]$ *and* $p' \in G'[s' \rightsquigarrow t']$ *such that* $p \smile p'$.

Theorem 2.5.—*There exist polynomial time algorithms for the following decision problems*:

1. *Let* $s, t \in V$, $s', t' \in V'$. *Decide* Alignment$(G, G', \smile, (s, t), (s', t'))$.
2. *Let* $s, t \in V$. *Decide* $\exists s', t' \in V'$ Alignment$(G, G', \smile, (s, t), (s', t'))$.
3. *Let* $s, t \in V$. *Decide* $\exists s' \in V'$ Alignment$(G, G', \smile, (s, t), (s', s'))$.

We postpone the proof of Theorem 2.5 to section 2.3, where we give explicit algorithms.

2.2. Down set graph of a poset.

Definition 2.6.—*A poset* $(P, \leq)$ *is a set P equipped with a transitive, reflexive, antisymmetric relation* $\leq$ *called a partial order. A linear extension of* $\leq$ *is a total order* $\leq'$ *which extends* $\leq$*, i.e., for all* $p_0, p_1 \in P$, $p_0 \leq p_1$ *implies* $p_0 \leq' p_1$*. We will use the notation* $(P, <)$ *to denote a strict partial order.*

Definition 2.7.—*Let* $(P, \leq)$ *be a poset. A down set of P is a subset* $A \subset P$ *such that for all* $p, q \in P$, $p \leq q$ and $q \in A$ *implies* $p \in A$*. The collection of down sets of P is denoted by* $\mathcal{O}(P)$.

Definition 2.8.—*Let* $(P, \leq)$ *be a finite poset. The down set graph of* $(P, \leq)$*, denoted* P_D*, is the directed graph* $(\mathcal{O}(P), F)$ *with vertices* $\mathcal{O}(P)$ *and edges* $A \rightarrow A'$ *iff* $A \subsetneq A'$ *and there does not exist* $A'' \in \mathcal{O}(P)$ *such that* $A \subsetneq A'' \subsetneq A'$.

Remark 2.9.—If $A, A' \in \mathcal{O}(P)$ *and there exists an edge* $A \rightarrow A'$ *in* P_D*, then* $A' = A \cup \{p\}$ *where* $p \in A$ *and* $q \leq p$ *implies that* $q \in A$.

For completeness, we include an algorithm for constructing the down set graph of a poset in section 2.3.2.

Theorem 2.10.—*Given a finite poset* $(P, \leq)$*, the associated down set graph* $P_D = (\mathcal{O}(P), F)$ *is a directed acyclic graph with a unique root* $\varnothing$ *and a unique leaf P. Moreover, there is a bijection between the paths in* P_D *from the root* $\varnothing$ *to the leaf P and the linear extensions of* $\leq$.

<u>Proof.:</u> The directed acyclic property is inherited from the definition via proper set inclusion. $\varnothing$ and P are the unique root and leaf since $\varnothing$ and P are the unique maximal and minimal elements in $\mathcal{O}(P)$, respectively. Now we show the moreover part. Let $\leq'$ be a linear extension of $\leq$. Suppose $P = \{p_1, p_2, \ldots, p_n\}$, where the indexing has been chosen so that $p_1 \leq' p_2 \leq' \cdots \leq' p_n$. Define $P_k := \{p_1, p_2, \ldots, p_k\}$. Then $\varnothing \rightarrow P_1 \rightarrow P_2 \rightarrow \cdots \rightarrow P_{n-1} \rightarrow P_n = P$ gives a path in P_D from root to leaf unique to $\leq'$. Now the converse. Suppose $\varnothing \rightarrow P_1 \rightarrow P_2 \rightarrow \cdots \rightarrow P_{n-1} \rightarrow P_n = P$ is a path from root to leaf in P_D. We claim that $P_k \setminus P_{k-1}$ must be a singleton for each k. Suppose otherwise. Then let $a, b \in P_k \setminus P_{k-1}$ such that $a \neq b$. Without loss, assume either $a \leq b$ or a and b are incomparable. Then $P_{k-1} \cup \{a\}$ is a down set and $P_{k-1} \subsetneq P_{k-1} \cup \{a\} \subsetneq P_k$ which by Definition 2.8 contradicts $P_{k-1} \rightarrow P_k$. Accordingly, let $p_k = P_k \setminus P_{k-1}$ for $k = 1, \ldots, n$, and see that this sequence of elements completely characterizes the path from root to leaf in P_D. Define the total order $\leq'$ via $p_1 \leq' p_2 \leq' \cdots \leq' p_n$; since $p_k \leq p_{k+1}$ holds for all k, $\leq'$ is a linear extension of $\leq$. Thus a path from root to leaf in P_D uniquely determines a linear extension $\leq'$ of $\leq$. ■

2.3. Algorithms.

2.3.1. Alignment problem.

<u>Definition 2.11.:</u> *Let* $\smile$ *be a matching relation between two labeled directed graphs* $G = (V, E, \Sigma, \ell)$ *and* $G' = (V', E', \Sigma', \ell')$*. The alignment graph* AlignmentGraph$(G, G', \smile)$ *is defined to be the directed graph* (V'', E'') *given by*

$$V'' = \{(v, v') \in V \times V' : \ell(v) \smile \ell'(v')\},$$

$$E'' = \{(e, e') \in E \times E' : \ell(e) \smile \ell'(e')\}.$$

The alignment graph AlignmentGraph($G, G', \smile$) is a subset of the product graph $G \times G'$, and hence it has at most $|V||V'|$ vertices, $|E||E'|$ edges.

The following proposition follows immediately from the construction of the alignment graph and the definition of matching paths.

Proposition 2.12.—*Paths in* AlignmentGraph($G, G', \smile$) *are in one-to-one correspondence with pairs of matching paths in G and G'. In particular,* $p'' = ((v_1, v_1'), (v_2, v_2'), \ldots, (v_n, v_n'))$ *is a path in the alignment graph iff* $p = (v_1, v_2, \ldots, v_n)$ *and* $p' = (v_1', v_2', \ldots, v_n')$ *are a pair of matching paths in G and G', respectively.*

It immediately follows that the alignment problem is equivalent to a reachability query in the alignment graph.

Proposition 2.13.—The following are equivalent:

1. Alignment($G, G', \smile, (s, t), (s', t')$),
2. AlignmentGraph($G, G', \smile$) $[(s, s') \rightsquigarrow (t, t')] = \varnothing$.

Proposition 2.14.—*If the cost of checking whether labels match is constant, then*

$$\text{AlignmentGraph}(G, G', \smile)$$

can be constructed in $O(|V||V'| + |E||E'|)$ *time.*

Proof.: The vertices of the alignment graph may be determined by checking for each element of $(v, v') \in V \times V'$ whether $\ell(v) = \ell(v')$. The edges of the alignment graph may be determined by checking for each $(e, e') \in E \times E'$ whether $\ell(e) = \ell(e')$. The result follows. ■

Proposition 2.15.—*Let* $G = (V, E)$ *be a directed graph and let* $s \in G$. *Then, the set* Reachable(G, s) $:= \{t \in V : G[s \rightsquigarrow t] = \varnothing\}$ *can be computed in* $O(|V| + |E|)$ time.

Proof.: Depth- or breadth-first search of G beginning at s will find all vertices reachable from s in time linear in the number of vertices and edges of G [3]. For completeness we provide a standard depth-first-search algorithm as Reachable(G, s) in Algorithm 1. ■

Proof of Theorem 2.5.: We show that the procedures Match, PathMatch, and CycleMatch of Algorithm 1 are polynomial time algorithms which decide the decision problems (1), (2), and (3), respectively. Let $G'' =$ AlignmentGraph($G, G', \smile$). By Proposition 2.13, we may rewrite (1), (2), (3) as follows:

1. Let $s, t \in V$, $s', t' \in V'$. Decide if $(t, t') \in \text{Reachable}(G'', (s, s'))$.
2. Let $s, t \in V$. Decide $\exists s', t' \in V'$ $(t, t') \in \text{Reachable}(G'', (s, s'))$.
3. Let $s, t \in V$. Decide $\exists s' \in V'$ $(t, s') \in \text{Reachable}(G'', (s, s'))$.

The correctness of Match for deciding (1) is now immediate. For (2) and (3), we recognize we can handle the outermost $\exists s' \in V'$ algorithmically via a **for** loop over $s' \in V'$. The algorithms PathMatch and CycleMatch result. This gives correctness.

To see that the algorithms are polynomial time, we refer to Propositions 2.14 and 2.15. In particular, given these it is straightforward to verify (defining $|G| = |V| + |E|$) that Reachable executes in worst-case $O(|G|)$ time, Match executes in worst-case $O(|G||G'|)$ time, and PathMatch and CycleMatch execute in worst-case $O(|G||G'||V'|)$ time. ■

See Figure 8 for an example of a PathMatch along with the corresponding path giving reachability in the alignment graph.

2.3.2. Construction of down set graph.—Recall that given a poset P a subset $I \subset P$ is independent if no two elements of I are comparable.

Proposition 2.16.—Algorithm 2 *computes the down set graph of a poset P.*

Proof.: We first note there is a one-to-one correspondence between down sets of a poset and the independent sets of a poset. In particular given a down set D we can associate an independent set $I = \text{Maxima|E|ementsOf}(D)$, and given an independent set I we can associate a down set $D = \text{Downset}(I) := \{p \in P : p \leq q \text{ for some } q \in I\}$. It is straightforward to see that this is one to one. Now consider the following recursively defined function:

$$f(D) := \left\{(D, D\backslash\left\{v\right\}) : \text{for } v \in \text{MaximalElementsOf}(D)\right\} \cup \bigcup_{v \in \text{MaximalElementsOf}(D)} f(D\backslash\{v\}).$$

See first that the recursion terminates since each recursive function call operates on a smaller set. Notice that if the function operates on a down set, then removing the maximal vertices again results in down sets—in fact, precisely the adjacent down sets in the down set graph. Hence $E = f(P)$ is the set of edges in the down set graph. Writing this recursion in terms of independent sets, we have

$$g(I) := \left\{(I, \text{MaximalElementsOf}(\text{Downset}(I)\backslash\left\{v\right\})) : \text{for } v \in I\right\} \cup \bigcup_{v \in I} g(\text{MaximalElementsOf}(\text{Downset}(I)\backslash\left\{v\right\})).$$

Now from

$$\text{MaximalElementsOf}(\text{Downset}(I)\backslash\{v\}) = \text{MaximalElementsOf}((I \cup \text{Predecessors}(v))\backslash\{v\})$$

the correctness of Algorithm 2 follows: it is just an implementation of this recursion which prevents some redundant recursion paths to save time (by storing them in V). ■

Algorithm 1.

Alignment problem.

```
procedure REACHABLE($G$, $s$)
    Push $s$ onto stack $S$.
    while $S$ is not empty do
        Pop $u$ from stack $S$.
        $R \leftarrow R \cup \{u\}$
        $A \leftarrow \{v : (u, v) \in E\}$
        for $v \in A$ do
            if $v \notin R$ then
                Push $v$ into stack $S$
            end if
        end for
    end while
    return $R$
end procedure

procedure MATCH($G$, $G'$, ($s$, $t$), ($s$, $t'$))
    $G'' \leftarrow$ AlignmentGraph($G$, $G'$, $\smile$)
    $R \leftarrow$ Reachable($G''$, ($s$, $s'$))
    if ($t$, $t'$) $\in R$ then
        return True
    else
        return False
    end if
end procedure

procedure PATHMATCH($G$, $G'$, ($s$, $t$))
    $G'' \leftarrow$ AlignmentGraph($G$, $G'$, $\smile$)
    for $s' \in V'$ do
        $R \leftarrow$ Reachable($G''$, ($s$, $s'$))
        for ($v$, $v'$) $\in R$ do
            if $v = t$ then
                return True
            end if
        end for
    end for
        return False
end procedure

procedure CYCLEMATCH($G$, $G'$, ($s$, $t$))
    $G'' \leftarrow$ AlignmentGraph($G$, $G'$, $\smile$)
    for $s' \in V'$ do
        $R \leftarrow$ Reachable($G''$, ($s$, $s'$))
        for ($v$, $v'$) $\in R$ do
            if $v = t$ and $v' = s'$ then
                return True
            end if
        end for
    end for
    return False
end procedure
```

Algorithm 2 does not run in polynomial time in general, yet it does for the special case of interest for the application of this paper. In particular, as we will describe in the next section, we will consider posets for which each element is associated to one of a small number of variables x_1, x_2, …, x_d, and all elements in the poset associated to the same variable are comparable. Moreover, as we discuss in section 4, associated to poset elements are time intervals determining the partial order such that one time interval (a, b) compares less than another time interval (c, d) iff $b \le c$. Under these assumptions, the *incomparability graph of the poset P* (i.e., the graph with vertices P and edges $u \leftrightarrow v$ whenever u and v are incomparable) is an *interval graph* [9], which is a special kind of *chordal graph*. A chordal graph with n vertices has at most n maximal cliques [20, 10]. From these considerations we get the following bound.

Proposition 2.17.—*Assume that P is a finite poset. Let d be the cardinality of the maximum independent set in P. Assume that the incomparability graph of P is chordal. Then* *the down set graph D_P has at most $2^d n$ vertices, and for fixed d,* Algorithm 2 *executes in polynomial time.*

Algorithm 2.

Down set graph.

```
procedure POSETTODOWNSETGRAPH($P$)
    Let $S$ be an empty stack
    $V \leftarrow \varnothing$
    $E \leftarrow \varnothing$
    Push MaximalElementsOf($P$) onto $S$
    while $S$ is not empty do
        Pop $I$ from $S$
        $V \leftarrow V \cup \{I\}$
        for $v \in I$ do
            $I' \leftarrow$ MaximalElementsOf(($I \cup$ Predecessors($v$)) \ {$v$})
            if $I' \notin V$ then
                Push $I'$ onto $S$
            end if
            $E \leftarrow E \cup \{(I', I)\}$
        end for
    end while
    return ($V$, $E$)
end procedure
```

3. Matching posets of extrema against computational dynamics models.

3.1. Labeled directed graphs from posets of extrema.

Assume that we can measure N variables over a time interval $[0, T]$ for the system that we are interested in modeling. If the quantities of these variables change continuously, then there exists a continuous function $x : [0, T] \to \mathbb{R}^N$ that represents the dynamics. We will assume that over this time interval each variable attains finitely many local extrema. As discussed in the introduction, in applications we can only sample the system at finite time intervals and the measurements will be subject to noise. We use the following structure to codify the possible orderings of maxima and minima of the coordinates x_i of x.

Definition 3.1.—*A* poset of extrema $(P, <_\tau; \mu)$ *is a finite poset* $(P, <_\tau)$ *equipped with a surjective function* $\mu : P \to \{-, m, M\}^N$ *that satisfies the following conditions. For* $n = 1, \dots, N$, *define* $P_n = \{p : \mu(p)_n \in \{m, M\}\}$.

1. $P = \bigcup_{n=1}^{N} P_i$.

2. If $n \neq j$, *then* $P_n \cap P_j = \varnothing$.

3. *For each n,* $P_n \subset P$ *is totally ordered by* $<_\tau$.

4. *Let $u, v \in P_n$. If $u <_\tau v$ and $\mu(u) = \mu(v)$, then there exists $w \in P_n$ such that $u <_\tau w <_\tau v$ and $\mu(u) \neq \mu(w)$.*

It is worth commenting on the rationale behind Definition 3.1. The poset of extrema is designed to capture orderings with respect to time of minima and maxima of d variables. The symbols $-$, m, and M stand for not an extremum, local minimum, and local maximum, respectively, and in applications the ordering $<_\tau$ respects the direction of time. Condition1 implies that every vertex of P is associated with an extremal event, where we define an extremal event to be a local extremum in exactly one coordinate projection. Condition 2 implies that each vertex is associated to an extremal event of precisely one variable, i.e., $\mu(p)_n = -$ for all but precisely one $n \in \{1,2, \ldots, N\}$. The assumption that each P_n is totally ordered with respect to $<_\tau$ implies that for each variable the ordering (with respect to time) of the minima and maxima is known; any ambiguity arises from comparing across variables. The final condition prevents a variable experiencing two local maxima or two local minima consecutively. Note that this is an assumption about the sampling frequency of the experiment.

Returning to the unknown function x that represents the dynamics, one expects, generically, that the maxima and minima of the coordinates x_n occur at different times. In the context of the poset of extrema $(P, <_\tau)$ we interpret this to mean that the true dynamics corresponds to linear extension of $<_\tau$. Since, given the data, the linear extension is unknown we consider any linear extension to be a plausible sequence of events. Our goal is to use the machinery of section 2.1 in order to search for linear extensions of P and thus we construct, following Theorem 2.10, the down set graph P_D of P, which exhibits a one-to-one correspondence between paths from root to leaf and linear extensions of P.

In order to produce a labeled directed graph suitable for pattern matching algorithms, we require labels on the vertices and edges of P_D. We make use of a particular set of labels

$$\Sigma_{\text{ext}} := \left\{I, D, *, -, m, M\right\}$$

called the extrema labels which are intended to carry the following information:

- I: increasing;
- D: decreasing;
- m: minimum;
- M: maximum;
- $-$: transitioning;
- $*$: lack of knowledge.

Definition 3.2.—*Let $(P, <_\tau; \mu)$ be a poset of extrema with down set graph $P_D = (O(P), E)$. The pattern graph P induced by the poset of extrema P is the labeled directed graph $(O(P), E \cup \{(A, A) : A \in O(P)\}, \Sigma^N_{ext}, \ell)$, where the labeling of the vertices is given by*

$$\ell(A)_n = \begin{cases} I \; if \; \ell\left(\max\left(P_n \cap A\right)\right)_n = m \; or \; \ell\left(\min\left(P_n \backslash A\right)\right)_n = M, \\ D \; if \; \ell\left(\max\left(P_n \cap A\right)\right)_n = M \; or \; \ell\left(\min\left(P_n \backslash A\right)\right)_n = m, \\ * \; otherwise, \end{cases}$$

and the labeling of the edges is defined by

$$\ell(A \to A) := (-, -, \ldots, -)$$

and (see Remark 2.9*)*

$$\ell(A \to A \cup \{p\}) := \mu(p), \quad p \in P.$$

Although the pattern graph is (trivially) cyclic due to the presence of self-edges, we will continue to refer to the root and leaf nodes of $\mathscr{P}$ as $\varnothing$ and P, respectively.

We give an example of a poset of extrema and the associated pattern graph using two variables x_1 and x_2, i.e., $N = 2$. Later on, we shall relate this example to a yeast dataset [18] that is discussed in section 4.2. For now, assume that x_1 and x_2 first attain minima, then later attain maxima, but that the timing of the minima cannot be distinguished, and neither can the maxima. This leads to the poset in Figure 1 (left). The associated pattern graph is in Figure 1 (right). The down sets of the poset of extrema are mapped to integers via

$$0 \leftrightarrow \varnothing; 1 \leftrightarrow \left\{x_1 \min\right\}; 2 \leftrightarrow \left\{x_2 \min\right\}; 3 \leftrightarrow \left\{x_1 \min, x_2 \min\right\};$$

$$4 \leftrightarrow \left\{x_1 \min, x_2 \min, x_2 \max\right\}; 5 \leftrightarrow \left\{x_1 \min, x_2 \min, x_1 \max\right\};$$

$$6 \leftrightarrow \left\{x_1 \min, x_2 \min, x_1 \max, x_2 \max\right\}.$$

3.2. Labeled directed graphs from computational dynamics.

In this section we develop the notion of a search graph, a labeled directed graph suitable for pattern matching sequence of extrema for models arising in computational dynamics.

Definition 3.3.—*Let $X = (0, \infty)^N$. Suppose for each $n \in \{1, \ldots, N\}$* we have a finite set

$$\Theta_n := \left\{\theta_1 < \theta_2 < \cdots < \theta_{J_n}\right\} \subset (0, \infty)$$

that we call thresholds. The rectangular decomposition $\mathscr{X}$ of X induced by $(\Theta_1, \Theta_2, \ldots, \Theta_N)$ *is the partition of X into cells $\mathscr{X}$ such that each $\sigma \in \mathscr{X}$ is a product of intervals*

$$\sigma = \prod_{n=1}^{N} I_n,$$

where for each n,

$$I_n \in \left\{ (0, \theta_1), (\theta_j, \theta_{j+1}), (\theta_{J_n}, \infty), [\theta_j, \theta_j] \;\middle|\; j = 1, \dots, J_n \right\}.$$

Accordingly, each cell is homeomorphic to an open ball of some dimension, which we call the dimension of the cell. We denote k-dimensional cells $\mathcal{X}_k$. *We call the cells in* $\mathcal{X}_N$ *domains and we call the cells in* $\mathcal{X}_{N-1}$ *walls. Two domains are said to be adjacent if the intersection of their closures contains a wall. We denote by* X_k *the union of all k-dimensional cells, i.e.,* $X_k := \bigcup_{\sigma \in \mathcal{X}_k} \sigma$.

Definition 3.4.—*Consider* $X = (0, \infty)^N$ *with rectangular decomposition* $\mathcal{X}$ *and a system of trajectories* $\mathcal{ST}$ *on X. A trajectory* $x : [t_0, t_1] \to X$ *is a domain trajectory if* $x([t_0, t_1]) \subset \xi$ *for some domain* $\xi \in \mathcal{X}_N$. *A trajectory* $x : [t_0, t_1] \to X$ *is a wall trajectory from a domain* ξ *to a domain* ξ' *if there exists a wall* $\sigma \in \mathcal{X}_{N-1}$ *such that* $x([t_0, t_1]) \subset \xi \cup \sigma \cup \xi'$, *and* $x^{-1}(\xi) < x^{-1}(\sigma) < x^{-1}(\xi')$ (*in the sense of comparing sets, i.e.,* $A < B$ *iff for all* $a \in A$, $b \in B$, $a < b$). *The domain graph generated by* $\mathcal{ST}$ *on* $\mathcal{X}$ *is the directed graph where the vertices are domains and there is an edge* $\xi \to \xi'$ *iff there exists a wall trajectory from* ξ *to* ξ'.

Definition 3.4 indicates how a domain graph is generated from a system of trajectories. For the applications discussed in this paper we are interested in particular trajectories that can be defined in terms of the domain graph.

Definition 3.5.—*Let* $D = (V, E)$ *be the domain graph generated by* $\mathcal{ST}$ *on* $\mathcal{X}$. *A trajectory which is the finite concatenation of wall trajectories is said to be a* domain-wall trajectory. *The* associated domain graph path *of a domain-wall trajectory x is the path of the domain graph edges corresponding to the wall trajectories which comprise x.*

Definition 3.6.—*Let* $\mathcal{ST}$ *be a system of trajectories on* $X = (0, \infty)^N$ *with rectangular decomposition* $\mathcal{X}$. *If every domain trajectory is monotone with respect to every variable and every wall trajectory undergoes at most one extremal event, we say* $\mathcal{ST}$ *is extrema-pattern-matchable with respect to* $\mathcal{X}$. *In this case, the labeled directed graph* $\mathcal{S} = \left(V, E, \Sigma_{ext}^N, \ell\right)$ *is said to be a search graph if* (V, E) *is the domain graph and* ℓ *reflects, as follows, our level of knowledge of the behaviors of trajectories*:

$$\ell(\xi')_n = \begin{cases} I \text{ } if \text{ } we \text{ } know \text{ } x_n(t) \text{ } is \text{ } increasing \text{ } for \text{ } every \text{ } trajectory \text{ } x(t) \text{ } in \text{ } domain \text{ } \xi', \text{ } else \\ D \text{ } if \text{ } we \text{ } know \text{ } x_n(t) \text{ } is \text{ } decreasing \text{ } for \text{ } every \text{ } trajectory \text{ } x(t) \text{ } in \text{ } domain \text{ } \xi', \text{ } else \\ * \text{ } otherwise, \end{cases}$$

$$\ell(\xi \to \xi')_n = \begin{cases} - \; if\, we've\, ruled\, out\, local\, extrema\, for\, x_n \text{ on the wall between } \xi \text{ and } \xi', \text{ else} \\ m \; if\, we've\, ruled\, out\, local maxima\, for\, x_n \text{ on the wall between } \xi \text{ and } \xi', \text{ else} \\ M \; if\, we've\, ruled\, out\, local minima\, for\, x_n \text{ on the wall between } \xi \text{ and } \xi', \text{ else} \\ * \; \text{otherwise.} \end{cases}$$

Observe that * indicates a lack of knowledge. If a system of trajectories is extrema-pattern-matchable, we can always make its domain graph into a search graph by choosing all labels to be *. This would lead to a higher rate of false positives in matching; it is better to assign the strongest labels one can prove.

As an example, consider a system of trajectories over a rectangular decomposition of $\mathbb{R}^2$, with trajectories qualitatively depicted in Figure 2 (left). The walls are shown as dotted lines. The domains are labeled 1–4, and within each domain the trajectories are monotonic in each variable, satisfying the requirements to be extrema-pattern-matchable. Within domain 1, x_2 trajectories are decreasing, while x_1 trajectories either decrease or increase. The associated search graph is shown in Figure 2 (right); node 1 corresponding to domain 1 is labeled $*$D and likewise for the other nodes. The edges between nodes in the search graph correspond to concatenation of domain trajectories. Clearly, there cannot be a maximum in x_1 as we pass from domain 1 to domain 2, but there could be a minimum, whereas x_2 is constantly decreasing and cannot have either a minimum or maximum on that wall. The edge $(1 \to 2)$ in the search graph is therefore labeled m-, and similar arguments hold for the other edges.

3.3. Matching pattern graphs against search graphs.

As indicated in the introduction we are interested in identifying whether our model for dynamics is capable of producing sequences of maxima and minima that do not contradict the experimental data. The capability is equivalent to the existence of a matching between a pattern graph and the search graph. To do this we impose a particular matching relation.

Definition 3.7.—*The* extremal event matching relation $\smile_{ext}$ *on* Σ_{ext} *is given by*

1. (*vertices*) $(I \smile_{ext} *)$, $(I \smile_{ext} I)$, $(D \smile_{ext} *)$, $(D \smile_{ext} D)$, $(* \smile_{ext} *)$,
2. (*edges*) $(- \smile_{ext} -)$, $(- \smile_{ext} m)$, $(- \smile_{ext} M)$, $(- \smile_{ext} *)$, $(m \smile_{ext} m)$, $(m \smile_{ext} *)$, $(M \smile_{ext} M)$, $(M \smile_{ext} *)$.

Given a pattern graph $\mathcal{P}$ *and a search graph* $\mathcal{S}$, *we extend this relation to* Σ^*_{ext} *by defining a* $\smile_{ext}$ *b whenever for all* $1 \le i \le N$, $a_i \smile_{ext} b_i$.

Theorem 3.8.—*Let* $\mathcal{P}$ *be a pattern graph for a poset of extrema* $(P, <, \mu)$ *and let* $\mathcal{S}$ *be a search graph for a system of trajectories* $\mathcal{ST}$ *which is extrema-pattern-matchable with respect to a rectangular decomposition* $\mathcal{X}$ *of* $X = (0, \infty)^N$. *If* $\mathcal{ST}$ *admits a domain-wall trajectory with a sequence of extremal events corresponding to a linear extension of P, then there exists a path* $p \in \mathcal{P}$ *from root to leaf and a path s in* $\mathcal{S}$ *such that* $p \smile_{ext} s$.

Proof.: Let $<'$ be a linear extension of P and name the elements of P as $e_1 <' e_2 <' \cdots <' e_n$. Suppose that $\phi : [t_0, t_1] \to X$ is a domain-wall trajectory with the sequence of extremal events $e_1, e_2, \ldots, e_n$. We show there exists a path p from root to leaf in $\mathscr{P}$ and a path s in $\mathscr{S}$ such that $p \smile s$.

Step 1. We construct a path p from root to leaf in $\mathscr{P}$ and a path s in $\mathscr{S}$. By Definition 3.5, since ϕ is a domain-wall trajectory, it can be written as a concatenation of wall trajectories $\phi^i : [t_i, t_{i+1}]$ for $i = 1, 2, \ldots, m$. Because $\mathscr{S}$ is a search graph, Definition 3.6 implies that extremal events for $\phi(t)$ can only occur on walls (i.e., during times when $\phi(t) \in X_{d-1}$) and at most one kind of extremal event can occur on a given wall. Since it is impossible for the same extremal event to occur twice in a row (e.g., between any two local minima there must be an intervening local maximum), and wall trajectories intersect precisely one wall, it follows that each wall trajectory experiences at most one extremal event. If we denote the set of extremal events which occur on the wall trajectory ϕ^i as E_i, then card $E_i \le 1$. Therefore, $E_i = \varnothing$ or $\{e_j\}$ for some j. Since ϕ experiences the events $e_1, e_2, \ldots, e_n$ in order, and ϕ is the concatenation of the trajectories ϕ^i, it follows that there exists an increasing function $\mu : \{1, \ldots, n\} \to \{1, \ldots, m\}$ such that for $i \in \{1, \ldots, n\}$, $e_i \in E_{\mu(i)}$. Define $p_1 := \varnothing$, and for $i \in \{1, \ldots, m\}$ define $p_{i+1} := \bigcup_{j=1}^{i} E_j$.

We show $p_1 \to p_2 \to \cdots \to p_{m+1}$ is a path in $\mathscr{P}$ from root to leaf. To this end it suffices to show that: (1) for each $i \in \{1, \ldots, m+1\}$, p_i is a down set of P, (2) for each $i \in \{1, \ldots, m\}$ there is an edge $p_i \to p_{i+1}$ in $\mathscr{P}$, (3) $p_1 = \varnothing$, and (4) $p_{m+1} = P$.

Let $i \in \{1, \ldots, m+1\}$. Define $k = \max \mu^{-1}(\{1, \ldots, i\})$. Since μ is increasing it follows that $p_{i+1} = \bigcup_{j=1}^{i} E_j = \{e_1, e_2, \ldots, e_k\}$.

Since $e_1 <' e_2 <' \cdots <' e_n$ and $<'$ is a linear extension of P, it follows that p_i is a down set of P. This demonstrates (1). Now let $i \in \{1, \ldots, m\}$. We show $p_i \to p_{i+1}$ is an edge in $\mathscr{P}$. There are two cases: either (a) $E_i = \varnothing$ and $p_i = p_{i+1}$, or else (b) or else $E_i = \{e_k\}$ for some k and p_{i+1} = pi ∪ $\{e_k\}$. For case (a), $p_i \to p_{i+1}$ is an edge in $\mathscr{P}$ since the pattern graph admits all self-edges. For case (b), $p_i \to p_{i+1}$ an edge in $\mathscr{P}$ since $\mathscr{P}$ contains the edges present in the down set graph of P. This demonstrates (2). That $p_1 = \varnothing$ is by definition. This demonstrates (3). Finally, $p_{m+1} = \bigcup_{i=1}^{m} E_i = P$. This demonstrates (4). Since (1), (2), (3), and (4) hold we have that $p = p_1 \to p_2 \to \cdots \to p_{m+1}$ is a path from root to leaf in $\mathscr{P}$. Let s be the path in $\mathscr{S}$ corresponding to the sequence of wall trajectories ϕ_i (i.e., the path associated with the domain-wall trajectory ϕ). Denote the vertices of the path i in order as $s_1 \to s_2 \to \cdots \to s_{m+1}$. Note that the wall trajectories ϕ_i correspond to the edges $s_i \to s_{i+1}$ in s. We have constructed a path p from root to leaf in $\mathscr{P}$ and a path s in $\mathscr{S}$, completing Step 1.

Step 2. We show that for p and s so constructed, $p \smile s$ holds. By Definitions 2.3 and 3.7, it suffices to show that for each $i \in \{1, \ldots, N\}$, for each $j \in \{1, \ldots, m+1\}$, $\ell(p_j)_i \smile' \ell(s_j)_i$ (i.e., vertex labels match) and for each $i \in \{1, \ldots, N\}$ for each $j \in \{1, \ldots, m\}$, $\ell(p_j \to p_{j+1})_i \smile' \ell(s_j \to s_{j+1})_i$ (i.e., edge labels match).

Proof that edge labels match. Let $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, m\}$. We show

$$\ell\left(p_j \to p_{j+1}\right)_i \smile' \ell\left(s_j \to s_{j+1}\right)_i . \qquad (3.1)$$

There are two cases: either (1) $E_j = \varnothing$ or (2) $E_j = \{e_k\}$ for some k. For case (1), $E_j = \varnothing$ implies $p_j = p_{j+1}$ and hence $\ell(p_j \to p_{j+1})_i = -$. Meanwhile $\ell(s_j \to s_{j+1})_i \in \{-, m, M, *\}$. By Definition 3.7 it follows that (3.1) holds for case (1). For case (2), $E_j = \{e_k\}$ for some k, we distinguish three subcases: (a) e_k is local minimum for variable i, (b) e_k is a local maximum for variable i, or (c) e_k is not a local extremum for variable i. For subcase (a), $\ell(p_j \to p_{j+1})_i = m$. Since the wall trajectory ϕ^j experienced a local minimum for variable i, it follows that we could not have ruled out a local minimum on the wall corresponding to the edge $s_j \to s_{j+1}$. This eliminates the possibility that $\ell(s_j \to s_{j+1})_i$ is either $-$ or M, i.e., $\ell(s_j \to s_{j+1})_i \in \{m, *\}$. Since $m \smile' m$ and $m \smile' *$, (3.1) holds for subcase (a). Subcase (b) is similar. For subcase (c), $\ell(p_j \to p_{j+1})_i = -$, and the argument of case (1) again applies. Hence (3.1) holds in all cases.

<u>Proof that vertex labels match.:</u> Let $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, m+1\}$. We show $\ell(p_j)_i \smile' \ell(s_j)_i$.

Let $P_i = \{p \in P : \ell(p)_i \in \{I, D\}\}$. We consider two cases: (1) $P_i = \varnothing$ and (2) $P_i = \varnothing$. For case (1), by Definition 3.2, $P_i = \varnothing$ implies $\ell(p_j)_i = *$. By Definition 3.6, $\ell(s_j)_i \in \{I, D, *\}$, and by Definition 3.7 $* \smile' I$, $* \smile' D$, and $* \smile' *$. It follows that $\ell(p_j)_i \smile' \ell(s_j)_i$ for case (1). For case (2), we assume $P_i = \varnothing$. Then $\ell(p_j)_i \in \{I, D\}$. There are four subcases depending on whether (a) $\ell(p_j)_i = I$ or $\ell(p_j)_i = D$ and (b) whether $p_j \cap P_i = \varnothing$. As they are all similar, we only consider the subcase when $\ell(p_j)_i = I$ and and $p_j \cap P_i = \varnothing$. Let $\phi' : [t_1, t_j] \to X$ be the domain-wall trajectory ϕ' obtained by concatenating $\phi^1, \phi^2, \cdots, \phi^{j-1}$. By construction, p_j is the set of events in P which occur on ϕ'. Let e be the maximal element of $p_j \cap P_i$. By Definition 3.2, $\ell(p_j)_i = I$ implies that e is a local minimum. It follows that ϕ' is increasing in variable i after event e occurs. This implies that for sufficiently small $\epsilon > 0$, $\phi^{j-1}|_{\left[t_j - \epsilon, t_j\right]}$ is an increasing trajectory with image contained in the domain s_j. By Definition 3.6, it follows that $\ell(s_j)_i \in \{I, *\}$. By Definition 3.7, $I \smile' I$ and $I \smile' *$, and $\ell(p_j)_i \smile' \ell(s_j)_i$ follows. Similar arguments for the other three subcases show $\ell(p_j)_i \smile' \ell(s_j)_i$ for case (2). We have shown $p \smile s$, which completes Step 2.

Since in Step 1 we constructed a path p in $\mathscr{P}$ from root to leaf and a path s in $\mathscr{S}$ and in Step 2 we showed $p \smile s$, the proof is complete. ■

To continue our example, we take the pattern graph $\mathscr{P}$ from Figure 1 (right) and the search graph $\mathscr{S}$ from Figure 2 (right) and seek matching paths $p \in \mathscr{P}$, $s \in \mathscr{S}$. To do this, we form the alignment graph as in Definition 2.11 using the matching relation $\smile_{ext}$ given in Definition 3.7. We then apply Proposition 2.13 that states that finding paths in the alignment graph is equivalent to finding pairs of matching paths in the pattern and search graphs. In particular, we seek a match to a path $p \in \mathscr{P}$ that is a linear extension of the poset of extrema in Figure 1 (left), to verify that the system of trajectories $\mathscr{S}\mathscr{T}$ in Figure 2 (left) can support the constraints on the order of extrema summarized by the poset.

The alignment graph is given in Figure 3, where each node is labeled by a pair (a, b), where a is a node identifier for the search graph (integers 1–4) and b is a node identifier for the pattern graph (integers 0–6). The red path denotes a match between path $p = (0,1,3,5,6)$ in the pattern graph in Figure 1 (right) and cyclic path $s = (1, 2, 3, 4,1)$ in the search graph in Figure 2 (right). We notice that $p = (0,1, 3, 5, 6)$ corresponds to a linear extension of the poset of extrema in Figure 1 (left), since it is a path from root to leaf of the pattern graph (Theorem 2.10). Therefore $\mathcal{ST}$ has at least one trajectory with a sequence of extrema respecting the constraints of the poset of extrema.

4. Application to regulatory networks.

As indicated in the introduction, to provide a demonstration of how to apply the combinatorial tools described in the previous sections we make use of DSGRN. A complete description of the mathematical framework can be found in [4]; however, for the benefit of the reader we begin this section with a short review. We then present an application to a simple system associated with the cell cycle of *S. cerevisiae* using experimental time series data (provided courtesy of the Haase lab; see [18] for data collection methods) for mRNA sequences associated with SWI4, HCM1, NDD1, and YOX1 collected at time intervals of 5 minutes.

4.1. DSGRN model for regulatory networks.

We provide a mathematical definition of a regulatory network, its associated parameter space, and an explicit decomposition of parameter space into a finite set of regions. For the sake of clarity we begin with a discussion of switching systems and demonstrate that they provide a system of trajectories. We then observe that based on the monotonicity assumption of systems of trajectories, the results for switching systems are applicable to a much larger class of dynamical systems. We conclude by relating the system of trajectories to the output of DSGRN, which provides us with a means of analyzing specific data sets.

Definition 4.1.—*A regulatory network* $\mathbf{RN} = (V, E)$ *consists of vertices* $V = \{1, \ldots, N\}$ *called network nodes, annotated directed edges* $E \subset V \times V \times \{\rightarrow, \dashv\}$ *called interactions, and for each* $n \in V$*, polynomial monotone increasing functions* $M_n : \mathbb{R}^{|S_n|} \rightarrow \mathbb{R}$ *called node logics, where* $S_n := \{(i, n) \in E\}$ *is called the nth source set.*

An $\rightarrow$ *annotated edge is referred to as an activation and an* $\dashv$ *annotated edge is called a repression. We indicate that either* $i \rightarrow j$ or $i \dashv j$ *without specifying which by writing* $(i, j) \in E$*. We allow self-edges. From one node to another we admit at most one type of annotated edge, e.g., we cannot have both* $i \rightarrow j$ *and* $i \dashv j$ *simultaneously. The nth target set is given by* $T_n := \{(n, j) \in E\}$.

A parameterized family of dynamics is generated from the regulatory network.

Definition 4.2.—*A parameter for a regulatory network* $\mathbf{RN} = (V, E)$ *is a tuple* $z \in Z \subset (0, \infty)^{(N+3\cdot|E|)}$*. The coordinates of a parameter z are associated with the nodes and edges of* RN *and are given by the values of four functions* $\gamma : V \to (0, \infty)$, *and* $l, u, \Theta : E \to (0, \infty)$ *with the constraint that* $l(e) \le u(e)$ *for each* $e \in E$.

The functions γ, l, u, and Θ are used to decompose phase space and generate dynamics as follows. Define

$$\Theta_n := \left\{\Theta((n, j)) : (n, j) \in T_n\right\} \quad \text{for } n \in \left\{1, \dots, N\right\}.$$

and assume that for all $n = 1, \dots, N$,

$$\text{if } \Theta((n, j)), \Theta((n, k)) \in \Theta_n, \text{ then } \Theta((n, j)) \neq \Theta((n, k)). \tag{4.1}$$

Then, $(\Theta_1, \Theta_2, \dots, \Theta_N)$ defines a rectangular decomposition $\mathcal{X}$ (see Definition 3.3) on $X := (0, \infty)^N$.

Define Γ to be the diagonal $N \times N$ matrix with diagonal entries $\gamma(n)$ for $n \in \{1, \dots, N\}$. Define $W : E \times X \to (0, \infty)$ via

$$W((i, j), x) = \begin{cases} l((i, j)) & \text{if } x_i < \Theta((i, j)) \text{ and } x_i \to x_j, \\ l((i, j)) & \text{if } x_i > \Theta((i, j)) \text{ and } x_i \dashv x_j, \\ u((i, j)) & \text{if } x_i > \Theta((i, j)) \text{ and } x_i \to x_j, \\ u((i, j)) & \text{if } x_i < \Theta((i, j)) \text{ and } x_i \dashv x_j, \\ 0 & \text{otherwise.} \end{cases}$$

Finally, define $\Lambda : X \to \mathbb{R}^N$ componentwise by

$$\Lambda_n(x) := M_n \circ W|_{S_n \times X}.$$

The construction up to the point allows us to recall the classical switching system [12, 5, 14],

$$\dot{x} = -\Gamma x + \Lambda(x), \tag{4.2}$$

where Γ is a diagonal matrix with diagonal entries given by γ, and the parameters l, u, and Θ specify Λ. Thus, (4.2) is implicitly associated with a given parameter value $z \in Z$. A nice feature of the switching system is that the structure of the dynamics over the rectangular domains $\mathcal{X}_N$ is easily understood. Observe that if $\xi \in \mathcal{X}_N$, then Λ is constant on ξ and hence it makes sense to write $\Lambda(\xi)$.

Definition 4.3.—*A parameter value* $z \in Z$ *is regular if* (4.1) *holds,* $l(e) < u(e)$ *for all* $e \in E$, *and*

$$-\gamma(n)\Theta((n, k)) + \Lambda_n(\xi) \neq 0 \tag{4.3}$$

if an $N-1$ *dimensional face of* $\xi \in \mathcal{X}_N$ *lies in the hyperplane defined by* $x_n = \Theta(n, k)$. *The set of regular parameter values is denoted by* Z^R.

From now on we restrict our attention to switching systems for which the parameter value is regular.

Definition 4.4.—*An* **RN** switching system domain trajectory *is a function* x: $[t_0, t_1] \to \mathrm{cl}(\xi)$, *where* $\xi \in \mathcal{X}_N$, *that solves the differential equation*

$$\dot{x} = -\Gamma x + \Lambda(\xi). \tag{4.4}$$

For $z \in Z^R$, *the associated* **RN** switching system of trajectories *at* z *is denoted by* $\mathcal{ST}_{sw}(\mathbf{RN}, z)$ *and is defined to be the smallest system of trajectories (see* Definition 1.1*) which contains every* **RN** *switching system domain trajectory.*

<u>**Remark 4.5.:**</u> It is straightforward to verify that under Definition 1.1, the intersection of two systems of trajectories is again a system of trajectories. Thus the notion of the smallest system of trajectories containing some set of trajectories is well-defined.

Observe that (4.4) is a linear differential equation and for each ξ can be extended to all of $\mathbb{R}^N$. In this case,

$$P^{\xi} := \Gamma^{-1}\Lambda(\xi) \in \mathbb{R}^N$$

is a globally attracting fixed point.

Let $\pi_n : \mathbb{R}^N \to \mathbb{R}$ be the canonical projection map onto the nth coordinate.

Proposition 4.6.—*Let* **RN** *be a regulatory network and* $z \in Z^R$. *Consider the switching system of trajectories* $\mathcal{ST}_{sw}(\mathbf{RN}, z)$. Let $\xi, \xi' \in \mathcal{X}_N$ *be separated by the hyperplane* $x_n = \Theta((n, j))$ *for some* $(n, j) \in E$ *such that* $\pi_n(\xi) < \pi_n(\xi')$. *Then,*

i. *there exists a wall trajectory from* ξ *to* ξ' *iff* $\max\{P_n^{\xi}, P_n^{\xi'}\} > \Theta((n, j))$,

ii. *there exists a wall trajectory from* ξ' *to* ξ *iff* $\min\{P_n^{\xi}, P_n^{\xi'}\} < \Theta((n, j))$.

<u>**Proof.:**</u> We show (i) and leave (ii) to the reader. Suppose there exists a wall trajectory $x : [0, T] \to (0, \infty)^N$ from ξ to ξ'. By Definition 1.1, the restrictions $x|_A$ and $x|_B$ onto $A = x^{-1}(\bar{\xi})$ and $B = x^{-1}(\bar{\xi}')$ are again trajectories. By Definition 4.4, $x|_A$ and $x|_B$ are solutions to (4.4). Such solutions are monotonic, so it follows that $x|_A$ and $x|_B$ are increasing. This requires $\pi_n(P^{\xi}) > \Theta((n, j))$ and $\pi_n(P^{\xi'}) > \Theta((n, j))$ (with strictness since we reach or leave the wall in finite time), yielding $\max\{P_n^{\xi}, P_n^{\xi'}\} > \Theta((n, j))$ as desired.

To prove the converse suppose $\max\left\{P_n^{\xi}, P_n^{\xi'}\right\} > \Theta((n, j))$. Let $\hat{x} \in \sigma$, where $\sigma \in \mathscr{X}_{N-1}$ is the cell between ξ and ξ'. Solve the initial value problem (4.4) with initial value $\hat{x}$ in forward time in ξ' and in backward time in ξ to obtain solutions x: $[t_0,t_1] \to$ cl(ξ) and y: $[t_1, t_2] \to$ cl(ξ') such that $x(t_1) = y(t_1) = \hat{x}$. By Definition 4.4, x and y are trajectories in $\mathscr{ST}_{\mathrm{sw}}(\mathbf{RN}, z)$. By Definition 1.1 the concatenation of x and y is again a trajectory. This yields a wall trajectory from ξ to ξ'. ■

Proposition 4.7.—*Let* **RN** *be a regulatory network, $z \in Z^R$, and $\xi \in \mathscr{X}_N$ be a domain. Then the switching system of trajectories $\mathscr{ST}_{sw}(\mathbf{RN}, z)$ has the following properties*:

i. *Every trajectory $x(t)$ in ξ is monotonic in each variable.*

ii. *If $P_n^{\xi} > \pi_n(\xi)$, then for every trajectory $x(t)$ in ξ, $x_n(t)$ is an increasing function.*

iii. *If $P_n^{\xi} < \pi_n(\xi)$, then for every trajectory $x(t)$ in ξ, $x_n(t)$* is a decreasing function.

iv. *If $P_n^{\xi} \in \pi_n(\xi)$, there exist trajectories $x(t)$ in ξ where $x_n(t)$ may be either an increasing, decreasing, or constant function.*

v. *Let w be a wall associated with the hyperplane $x_n = \Theta((n, j))$ arising from the regulatory network interaction $x_n \to x_j$. Then, the only type of extremum a wall trajectory can undergo as it passes through w is a local minimum in the variable x_j.*

vi. *Let w be a wall associated with the hyperplane $x_n = \Theta((n, j))$ arising from the regulatory network interaction $x_n \dashv x_,$. Then, the only type of extremum a wall trajectory can undergo as it passes through w is a local maximum in the variable x_j.*

<u>Proof.:</u> Properties (i)-(iv) follow immediately from (4.4).

We show (v) and leave (vi) to the reader. Let w be a wall associated with the hyperplane $x_n = \Theta((n,j))$ arising from the regulatory network interaction $x_n \to x_j$. Let ξ, ξ' be the adjacent domains that w separates, such that $\pi_n(\xi) < \pi_n(\xi')$. Let $x : [t_0,t_1] \to X$ be a wall trajectory from ξ to ξ'. We show that x cannot undergo any kind of extremum except possibly a local minimum in the variable x_j.

Since $z \in Z^R$, $\Theta((n, j)) \neq \Theta((n, k))$ for $j \neq k$. This implies that $P_k^{\xi} = P_k^{\xi'}$ for all $k \neq j$. Define $x|_A$ and $x|_B$, where $A = x^{-1}(\bar{\xi})$ and $B = x^{-1}(\bar{\xi}')$. Since $x|_A$ and $x|_B$ each obey (4.4) on their respective domains, it follows that x_k obeys $\dot{x}_k = -\gamma_k(x_k - P_k^u)$ everywhere. Thus $x_k(t)$ is monotonic and hence experiences no extremal event. Now we show x cannot undergo a local maximum event in variable x_j. Since $l((i, j)) < u((i, j))$ it follows from the definitions that we must have $P_j^{\xi} < P_j^{\xi'}$. If $x|_A$ is constant or decreasing in the jth coordinate, then there cannot be a local maximum as we pass the wall. So we consider only the case where $x|_A$ is increasing in the jth coordinate. This case requires that $x_j(A) < P_j^u$. Hence,

$x_j(\min B) < P_j^{\xi} < P_j^{\xi'}$. Since $x|_B$ is a solution of the initial value problem corresponding to (4.4) with an initial condition for x_j less than $P_j^{\xi'}$, it follows that x_j is everywhere increasing. Therefore, x_j does not experience a local maximum. ■

While the action of the switching system on top dimensional domains is easy to understand, the matching results only make use of the qualitative properties of the system of trajectories described in the conclusions of Propositions 4.6 and 4.7. With this in mind let $\mathcal{ST}(\mathbf{RN}, z)$ denote any system of trajectories that satisfies Proposition 4.6(i)–(ii) and Proposition 4.7(i) –(vi).

Theorem 4.8.—*Let* $(\mathbf{RN}, z)$ *be a parameterized regulatory network with* $z \in Z^R$ *and let* $\mathcal{ST}(\mathbf{RN}, z)$ *be an associated system of trajectories. Let* $\mathcal{S} = (V, E, \Sigma, \ell)$ *be the labeled directed graph given by the following*:

i. $V = \mathcal{X}_N$.

ii. $E = \left\{(\xi, \xi') \in \mathcal{X}_N^2 : \xi \, and \, \xi' \, are \, adjacent, and \, for \, all \, n \in V, either. \right.$
$\left(\left(\pi_n(\xi) < \pi_n(\xi')\right) \wedge \left(\min\left\{P_n^{\xi}, P_n^{\xi'}\right\} > \pi_n(\xi)\right)\right) or$
$\left.\left(\left(\pi_n(\xi) > \pi_n(\xi')\right) \wedge \left(\max\left\{P_n^{\xi}, P_n^{\xi'}\right\} < \pi_n(\xi)\right)\right)\right\}$

iii. *For all* $n \in V$, $\ell(\xi)_n = D$ *whenever* $P_n^{\xi} < \pi_n(\xi)$.

iv. *For all* $n \in V$, $\ell(\xi)_n = *$ *whenever* $P_n^{\xi} \in \pi_n(\xi)$.

v. *For all* $n, j, k \in V$,

$\ell(\xi \to \xi')_n = -$ *whenever* $x_j \to x_k$, $n = k$, *and* $x_j = \Theta((j, k))$ *separates* ξ *and* ξ'.

vi. *For all* $n, j \in V$,

$\ell(\xi \to \xi')_n = m$ *whenever* $x_j \to x_n$ *and* $x_j = \Theta((j, n))$ *separates* ξ *and* ξ'.

vii. *For all* $n, j \in V$,

$\ell(\xi \to \xi')_n = M$ *whenever* $x_j \dashv x_n$ *and* $x_j = \Theta((j, n))$ *separates* ξ *and* ξ'.

Then, $\mathcal{S}$ *is a search graph for* $\mathcal{ST}(\mathbf{RN}, z)$ *with the rectangular decomposition* $\mathcal{X}$.

Proof.: By Proposition 4.6, it follows that $\mathcal{S}$ is the domain graph for $\mathcal{ST}_{\mathrm{sw}}(\mathbf{RN}, z)$ with the rectangular decomposition $\mathcal{X}$. By Proposition 4.7, it follows that the vertex and edge labels satisfy the requirements of Definition 3.6. ■

Theorem 4.8 guarantees that given a regulatory network and regular parameter value there exists a search graph. The next proposition indicates that parameter space admits a finite decomposition, where within each open component of the decomposition the parameters exhibit isomorphic search graphs.

Proposition 4.9.—For a fixed regulatory network the following hold:

i. *The regular parameter values Z^R form an open and dense subset of all parameter values Z.*

ii. *Z^R has finitely many connected components.*

iii. *The connected components of Z^R are semialgebraic sets which can be written as systems of strict inequalities involving polynomials of the parameters.*

iv. *If $z_1, z_2 \in Z^R$ are in the same connected component of Z^R, then the search graph for $\mathcal{ST}(\mathbf{RN}, z_1)$ is isomorphic to the search graph associated with $\mathcal{ST}(\mathbf{RN}, z_2)$.*

We do not provide a proof of Proposition 4.9 as it is a partial summary of results in [4] that describes the mathematical foundations for the DSGRN software [15]. Given a regulatory network for which $|S_n| \leq 3$ and $|T_n| \leq 3$ the key computational result of [4] is that DSGRN provides an efficient computational scheme for constructing an undirected graph PG, called the parameter graph, where each node represents one of the connected components described in Proposition 4.9(iii) and the edges correspond to a notion of adjacency of the parameter regions. In addition, for each node in the parameter graph DSGRN can be used to compute the associated domain graph, i.e., identify the set of vertices and the set of edges of S as described in Theorem 4.8(i) and (ii).

From the domain graph, it is possible to extract summary data, called *a Morse* graph, that provides information about the global dynamics. The association of a Morse graph to each node in the parameter graph PG gives rise to the notion of a database of dynamical information; the interested reader is referred to [1, 2, 4] for further details about Morse graphs and dynamical databases. For the purposes of this paper, the notion of the domain graph, and the search graph which arises from it, suffices.

We remark that the system of trajectories $\mathcal{ST}_{sw}(\mathbf{RN}, z)$ qualitatively depicted in Figure 2 (left) arises from the regulatory network $\mathbf{RN}(\{x_1,x_2\}, \{x_1 \rightarrow x_2, x_2 \dashv x_1\})$ for any regular parameter z satisfying

$$l((1,2)) < \theta((1,2)) < u((1,2)),$$

$$l((2,1)) < \theta((2,1)) < u((2,1)).$$

4.2. Labeled pattern graph from experimental data.

We now turn to the task of generating a labeled pattern graph from experimental data. The graph in Figure 4 (left) provides normalized expression level data for mRNA sequences associated with SWI4, HCM1, NDD1, and YOX1 from *S. cerevisiae* taken at time intervals of 5 minutes. Since we are only concerned with the orderings of the extrema, the normalization of the data makes it easier to identify these extrema.

As indicated in the introduction identifying extrema in data is a serious statistical endeavor that we do not address in this paper. While our techniques require a set of potential sequences of extrema, they are agnostic with respect to how the potential sequences are derived; therefore we are content for the purpose of this paper to use simple heuristics. In particular, the table in Figure 4 (right) provides intervals of time within which we declare a maximum or minimum value of expression has occurred. For example, to allow for noise in the data the tightest time bound we are willing to assume on the maxima for SWI4 and YOX1 is (15, 30). Similarly, we ignore the potential for a local minimum and maximum of NDD1 at time points 70 and 80 and instead assume that a minimum occurs somewhere within the time interval (70,85).

Because we are using intervals to quantify the occurrence in time of extrema we cannot expect to obtain a linear ordering. Instead we define a partial order $<_\tau$ by

$$(a, b) < \tau(c, d) \text{ whenever } b \le c\,. \qquad (4.5)$$

Note that the poset of extrema in Figure 1 (left) arises from using $<_\tau$ on rows 1, 4, 5, and 6 in the table; i.e., we form the poset consisting of the first minimum and first maximum of each of x_1 = SWI4 and x_2 = YOX1.

Using all of the rows in the table in Figure 4 results in the poset indicated in Figure 5. Note that the linear extensions of $<_\tau$ correspond to ordered sequences of extrema events.

Observe that we have constructed a poset of extrema $(P, <_\tau; \mu)$ (see Definition 3.1) where P consists of the entries of the time interval column in the table in Figure 4 (right), $<_\tau$ is as defined by (4.5), and the values of $\mu: P \to \{-, m, M\}^4$ are obtained from the event column of the table in Figure 4 (right). For example, if the first coordinate of μ corresponds to SWI4, then $P_1 = \{(-\infty, 10), (15, 30), (50, 60), (75, \infty)\}$. Following Definition 3.2 the associated pattern graph $\mathscr{P}$ is shown in Figure 6. We remark that Proposition 2.17 applies in this situation, i.e., Algorithm 2 can quickly compute the pattern graph $\mathscr{P}$.

4.3. Results for wavepool models.

The regulatory network $\mathbf{RN}_W$ shown in Figure 7 is perhaps the simplest representative of the family of wavepool models proposed by the Haase lab [19] for the metabolic cycle in *S. cerevisiae*. Our goal is to identify if, for a particular identification of the nodes {1, 2, 3,4} with the genes {SWI4, HCM1, NDD1, YOX1}, a DSGRN model of this form is consistent with the time series data shown in Figure 4 (left) and, if so, under what ranges of parameter values.

Applying the DSGRN database code to $\mathbf{RN}_W$ produces a parameter graph PG with 1080 nodes. As explained in section 4.1, the phase space of this network is $(0, \infty)^4$ and the parameter space is a subset of $(0, \infty)^{19}$. The nodes correspond to 1080 distinct regions of parameter space, which in turn give rise to 1080 distinct classes of state transition graphs which may arise from the regulatory network of Figure 7. For each node we may present the associated nonempty connected region of parameter space as the solution set of a system of polynomial inequalities. For each point z in this set, the $\mathscr{ST}_{sw}\left(\mathbf{RN}_W, z\right)$ system of trajectories gives rise to the same associated search graph.

4.3.1. Invalidating a model.

—As a simple test we begin by considering a model that can be ruled out based on known biological interactions. Consider the regulatory network $\mathbf{RN}_W$ where 1 ↔ NDD1, 2 ↔ HCM1, 3 ↔ SWI4, and 4 ↔ YOX1. The evidence table in the supplementary material of [17, Table S2] shows no known regulation of HCM1 by NDD1, despite numerous experiments of different types, so we would expect to see no pattern matches with this model. Applying our pattern matching methodology to the search graphs which arise for each of the 1080 parameter nodes corresponding to this instantiation of the regulatory network $\mathbf{RN}_W$ and the pattern graph of Figure 6 we obtain no matches. This indicates that no matter how parameters are chosen, the dynamical model cannot give rise to a solution trajectory exhibiting a behavior qualitatively similar to the collected experimental data of Figure 4. Accordingly, we reject the proposed regulatory network model.

4.3.2. Parameter learning.

—We now turn to an accepted version of the wavepool regulatory network model $\mathbf{RN}_W$ where 1 ↔ SWI4, 2 ↔ HCM1, 3 ↔ NDD1, and 4 o YOX1. For this network we expect to find matches (in fact, failure to find any matches would probably suggest that the DSGRN model was inappropriate).

Applying the pattern matching methodology to the search graphs which arise for each of the 1080 parameter nodes corresponding to this revised instantiation of the regulatory network $\mathbf{RN}_W$ and the pattern graph of Figure 6 results in matches for 22 parameter nodes. By Theorem 3.8, for any parameter z belonging to any of the other 1058 parameter nodes, the $\mathcal{ST}_{sw}(\mathbf{RN}_W, z)$ system of trajectories does not contain any trajectory passing only through domains and walls which exhibits a sequence of extrema matching a plausible total order of the experimentally observed extrema in the data. Hence, our analysis has dramatically reduced uncertainty about relationships between the underlying parameters.

Furthermore, we can explicitly describe the regions of parameter space that correspond to these 22 matching parameter nodes. For example, for one such parameter node the associated parameter region in $(0, \infty)^{19}$ is given by the inequalities

$$\begin{aligned} &0 < l_1 l_2 < \gamma_1\theta_3 < u_1 l_2 < \gamma_1\theta_5 < l_1 u_2 < u_1 u_2,\\ &0 < l_3 < \gamma_2\theta_4 < u_3,\\ &0 < l_4 < \gamma_3\theta_1 < u_4,\\ &0 < l_5 < \gamma_4\theta_2 < u_5, \end{aligned} \tag{4.6}$$

where

$l_1 := l((NDD1; SWI4))$ $u_1 := u((NDD1; SWI4))$ $\theta_1 := \Theta((NDD1; SWI4))$

$l_2 := l((YOX1; SWI4))$ $u_2 := u((YOX1; SWI4))$ $\theta_2 := \Theta((YOX1; SWI4))$

$l_3 := l((SWI4; HCM1))$ $u_3 := u((SWI4; HCM1))$ $\theta_3 := \Theta((SWI4; HCM1))$

$l_4 := l((HCM1; NDD1))$ $u_4 := u((HCM1; NDD1))$ $\theta_4 := \Theta((HCM1; NDD1))$

$l_5 := l((SWI4;\ YOX1))\ u_5 := u((SWI4;\ YOX1))\ \theta_5 := \Theta((SWI4;\ YOX1))$

$\gamma_1 := \gamma(SWI4)\ \gamma_2 := \gamma(HCM1)$

$\gamma_3 := \gamma(NDD1)\ \gamma_4 := \gamma(YOX1)$

A complete listing of such regions is available in supplementary material [16].

A pair of matching paths between the pattern graph and the search graph corresponding to this parameter region is shown in Figure 8.

5. Concluding remarks.

We presented a general method capable of rejecting models that cannot match coarse data generated by an experimentally measured time series. Our assumptions are very general; we expect that the time series is subject to substantial experimental error and therefore we only assume partial knowledge of the order of extrema of the components of the time series. This information is encoded in a poset of extrema, which we represent as a labeled directed acyclic graph called a *pattern graph*.

Coming from the modeling side, we start with a concept of *system of trajectories*. Such a system can be produced by decomposition of the phase space into disjoint domains in which all trajectories are monotone, and on each boundary between domains, at most one component can attain an extremum. Existence of such a decomposition allows extraction of the extremal behavior and its encoding into a search graph. On this level of generality we show that the problem of matching labeled paths between pattern graph and *search graph* can be solved in polynomial time.

We discuss the applicability of our approach in two directions. First, we provide an example of a class of models which can be used to construct search graphs. Second, we apply our method to expression time series data from cell cycle in yeast. We show how our method can be used to learn parameter regimes consistent with the experimental measurement by rejecting parameter regimes where the dynamics does not align with the data.

In order to ensure our results may be reproduced we adhere to the following recipe: (1) we release our code under an open-source license, (2) we host our code on a publicly available site using version control (i.e., history tracking), (3) we give the version numbers of the code used to produce the result, (4) we provide instructions for installing and running the code, and (5) we produce *digital object identifiers* (DOIs) of the versioned code for use in bibliographical entries.

The computer codes used to reproduce the results in this paper are stored in two code repositories. The first repository is the DSGRN project [15]. This is an open-source project which, as of writing, is hosted on the code-sharing website GitHub at https://github.com/shaunharker/DSGRN. The version utilized for this paper is 1.0.0. The second repository is the supplement to this paper [16] and houses the code (which relies on DSGRN) which is used to reproduce the above results. This again is open-source and is hosted at https://

github.com/shaunharker/2017-DSGRN-ModelRejection. The version utilized for this paper is 1.0.0. The DOIs for these can be found in the references.

Funding:

This work was partially supported by grant NIH-1R01GM126555-01 as part of the Joint DMS/NIGMS Initiative to Support Research at the Interface of the Biological and Mathematical Sciences. The work of the first author was partially supported by DARPA D12AP200025 and USDA 2015-51106-23970. The work of the second author was partially supported by NSF grants DMS-1226213, DMS-1361240, DARPA D12AP200025, and NIH R01 grant 1R01AG040020-01. The work of the third and fourth authors was partially supported by grants NSF-DMS-1125174, 1248071, and 1521771 and DARPA contract HR0011-16-2-0033.

REFERENCES

[1]. Arai Z, Kalies W, Kokubu H, Mischaikow K, Oka H, and Pilarczyk P, A database schema for the analysis of global dynamics of multiparameter systems, SIAM J. Appl. Dyn. Syst, 8 (2009), pp. 757–789.

[2]. Bush J, Gameiro M, Harker S, Kokubu H, Mischaikow K, Obayashi I, and Pilarczyk P, Combinatorial-topological framework for the analysis of global dynamics, Chao, 22 (2012), 047508.

[3]. Cormen TH, Stein C, Rivest RL, AND Leiserson CE, Introduction to Algorithms, 2nd ed., McGraw-Hill, New York, 2001.

[4]. Cummins B, Gedeon T, Harker S, Mischaikow K, and Mok K, Combinatorial representation of parameter space for switching networks, SIAM J. Appl. Dyn. Syst, 15 (2016), pp. 2176–2212. [PubMed: 30774565]

[5]. de Jong H, Modeling and simulation of genetic regulatory systems: A literature review, J. Comput. Biol, 9 (2002), pp. 67–103, 10.1089/10665270252833208. [PubMed: 11911796]

[6]. Edwards R, Analysis of continuous-time switching networks, Phys. D, 146 (2000), pp. 165–199, https://www.sciencedirect.com/science/article/pii/S0167278900001305.

[7]. Edwards R, Farcot E, and Foxall E, Explicit construction of chaotic attractors in Glass networks, Chaos Solitons Fractals, 45 (2012), pp. 666–680, 10.1016/j.chaos.2012.02.018.

[8]. Edwards R and Ironi L, Periodic solutions of gene networks with steep sigmoidal regulatory functions, Phys. D, 282 (2014), pp. 1–15, 10.1016/j.physd.2014.04.013.

[9]. Fulkerson D and Gross O, Incidence matrices and interval graphs, Pacific J. Math, 15 (1965), pp. 835–855.

[10]. Gavril F, Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and, maximum independent set of a chordal graph, SIAM J. Comput, 1 (1972), pp. 180–187.

[11]. Glass L and Kauffman SA, Co-operative components, spatial localization and oscillatory cellular dynamics, J. Theoret. Biol, 34 (1972), pp. 219–37, https://www.ncbi.nlm.nih.gov/pubmed/5015702. [PubMed: 5015702]

[12]. Glass L and Kauffman SA, The logical analysis of continuous, non-linear biochemical control networks, J. Theoret. Biol, 39 (1973), pp. 103–29, https://www.ncbi.nlm.nih.gov/pubmed/4741704. [PubMed: 4741704]

[13]. Glass L and Pasternack J, Stable oscillations in mathematical models of biological control systems, J. Math. Biol, 6 (1978), pp. 207–223, https://link.springer.com/article/10.1007/BF02547797.

[14]. Gouzé J-L and Sari T, A class of piecewise linear differential equations arising in biological models, Dyn. Syst, 17 (2002), pp. 299–316, 10.1080/1468936021000041681.

[15]. Harker S, DSGRN Software, , 2017.

[16]. Harker S and Cummins B, Code Supplemental for “Model Rejection and Parameter Reduction via Time Series”. , 2017.

[17]. Kovacs LAS, Mayhew MB, Orlando DA, Jin Y, Li Q, Huang C, Reed SI, Mukher-JEE S, and Haase SB, Cyclin-dependent kinases are regulators and effectors of oscillations driven by a transcription factor network, Molecular Cell, 45 (2012), pp. 669–679. [PubMed: 22306294]

[18]. Leman AR, Bristow SL, and Haase SB, Analyzing transcription dynamics during the budding yeast cell cycle, in Cell Cycle Control. Methods in Molecular Biology (Methods and Protocols), Vol. 1170, Noguchi E and Gadaleta M, eds., Humana Press, New York, NY, 2014, pp. 295–312.

[19]. Orlando DA, Lin CY, Bernard A, Wang JY, Socolar JE, Iversen ES, Hartemink AJ, and Haase SB, Global control of cell-cycle transcription by coupled cdk and, network oscillators, Nature, 453 (2008), pp. 944–947. [PubMed: 18463633]

[20]. Rose DJ, Tarjan RE, and Lueker GS, Algorithmic aspects of vertex elimination on graphs, SIAM J. Comput, 5 (1976), pp. 266–283.

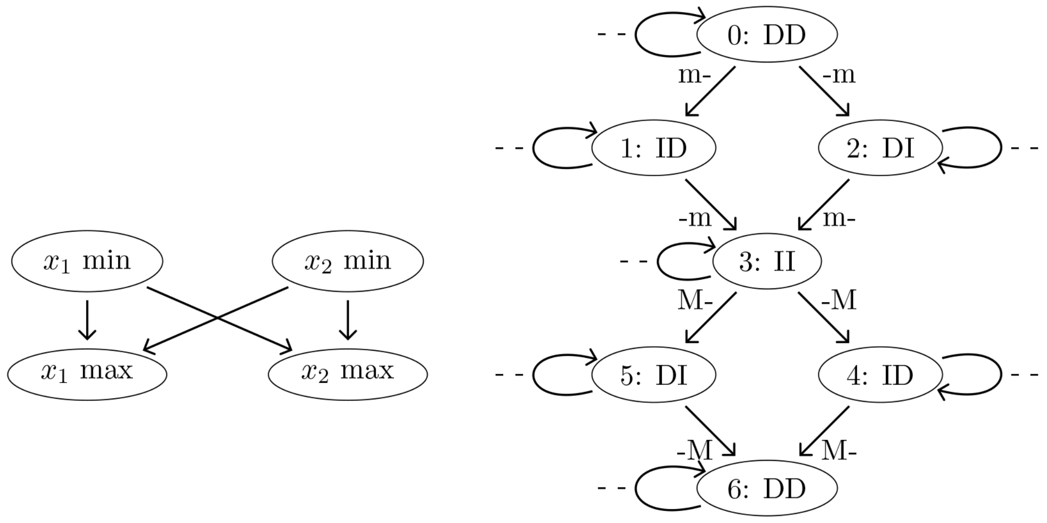


Figure 1.

Left: Example poset of extrema with four extrema. Right: Associated pattern graph.

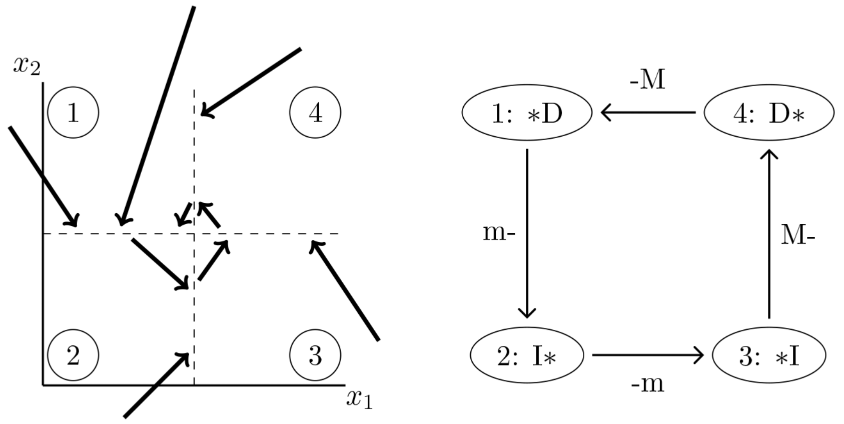


Figure 2.
Left: An example system of trajectories that is extrema-pattern-matchable over a rectangular decomposition of four components. Right: The associated search graph.

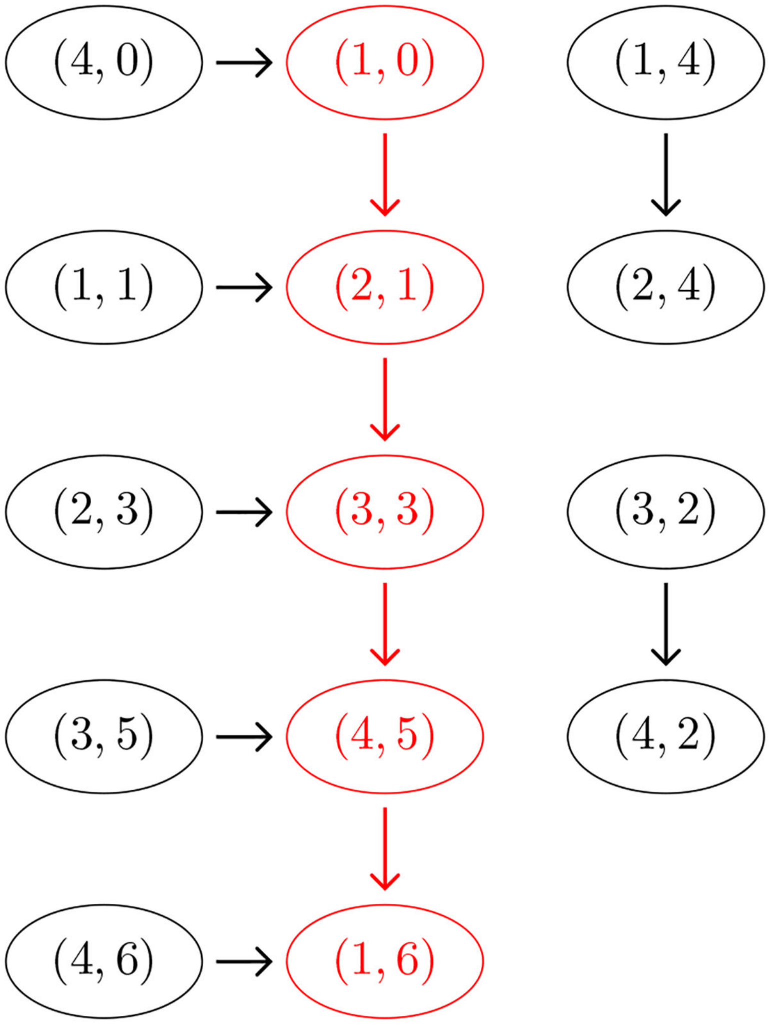
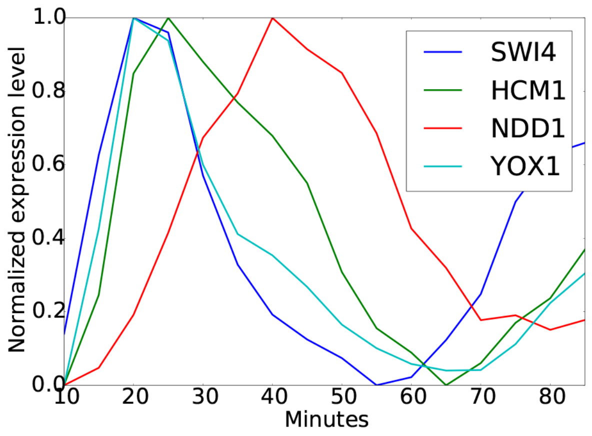


Figure 3.
Alignment graph for the pattern graph in Figure 1 (right) and the search graph in Figure 2 (right). The red path indicates a match between paths in the graphs.



event	time interval
SWI4 min	$(-\infty, 10)$
HCM1 min	$(-\infty, 10)$
NDD1 min	$(-\infty, 10)$
YOX1 min	$(-\infty, 10)$
SWI4 max	$(15, 30)$
YOX1 max	$(15, 30)$
HCM1 max	$(20, 35)$
NDD1 max	$(35, 45)$
SWI4 min	$(50, 60)$
YOX1 min	$(60, 75)$
HCM1 min	$(60, 75)$
NDD1 min	$(70, 85)$
SWI4 max	$(75, \infty)$
HCM1 max	$(85, \infty)$
YOX1 max	$(85, \infty)$
NDD1 max	$(85, \infty)$

Figure 4.

Left: Time series of RNA-seq data normalized to range from zero to one. Right: Table of time intervals associated to the extrema in the plot on the left.

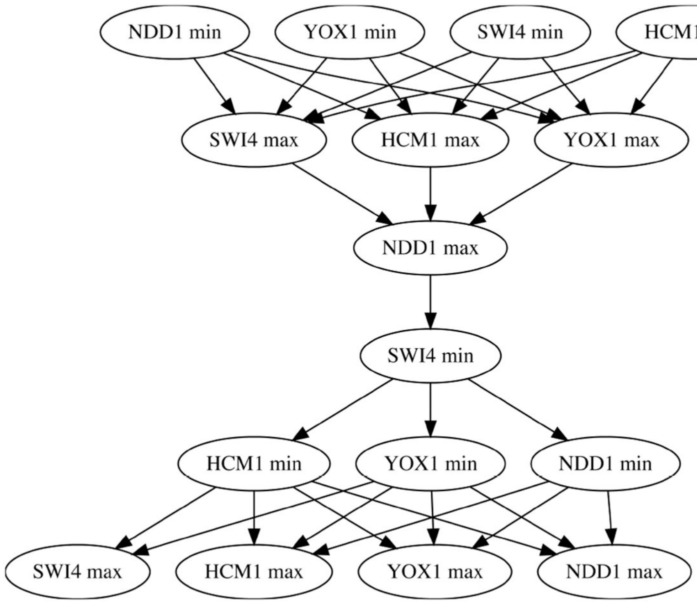


Figure 5.

The pattern (poset) arising from the choice of time intervals of extrema based on the table in Figure 4. Arrows indicate direction of time.

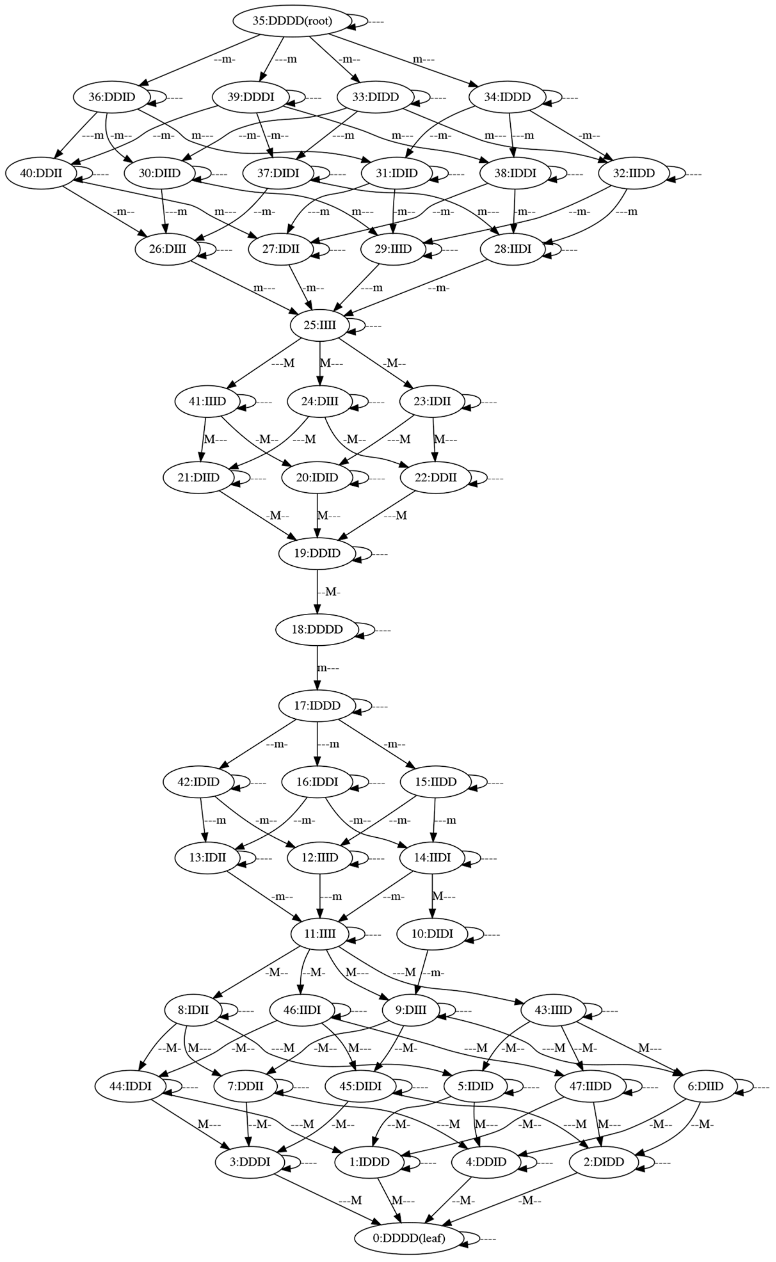


Figure 6.
Pattern graph associated to the pattern in Figure 5.

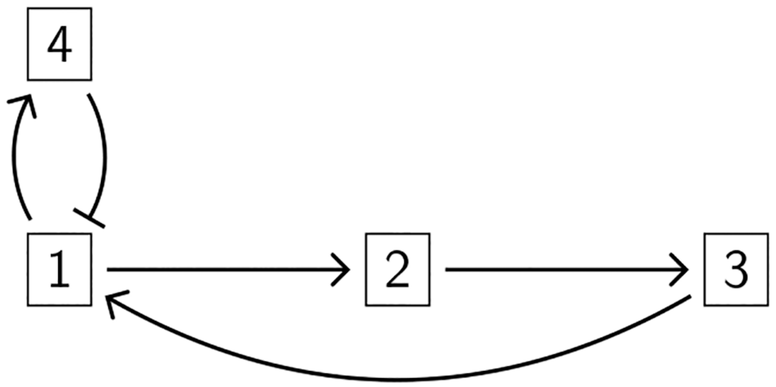


Figure 7.

The wavepool regulatory network $\mathbf{RN}_W$ where M_1 is multiplication.

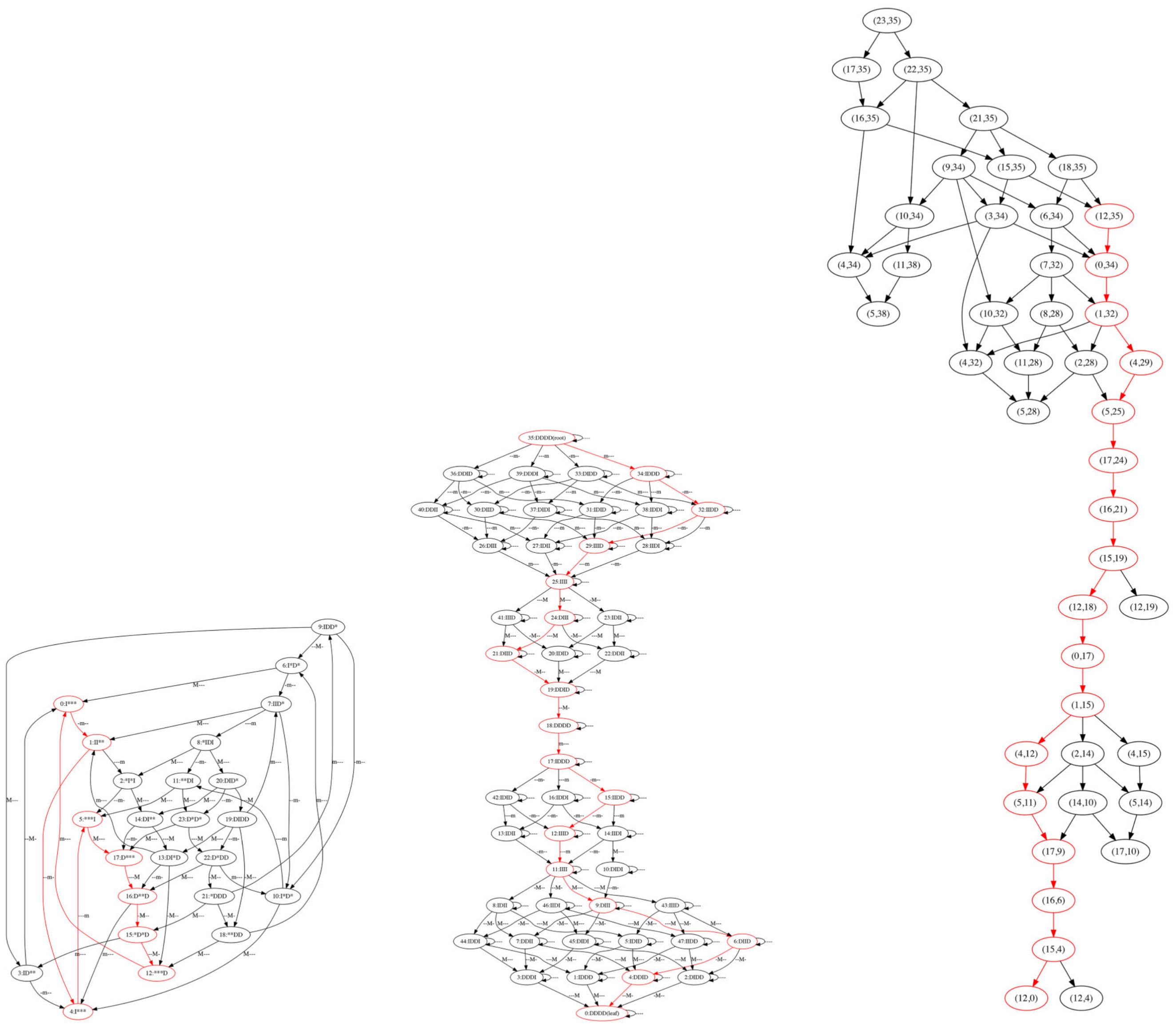


Figure 8.

Left and middle: Matching paths in search graph and pattern graph by Algorithm 1. Right: The corresponding path the algorithm found in the alignment graph.