Journal of Mathematical Biology (2020) 80:1523–1557
https://doi.org/10.1007/s00285-020-01471-4

Mathematical Biology

Using extremal events to characterize noisy time series

Eric Berry[1] · Bree Cummins[1] · Robert R. Nerem[1] · Lauren M. Smith[2] · Steven B. Haase[2] · Tomas Gedeon[1]

Received: 5 March 2019 / Revised: 13 January 2020 / Published online: 1 February 2020

Abstract
Experimental time series provide an informative window into the underlying dynamical system, and the timing of the extrema of a time series (or its derivative) contains information about its structure. However, the time series often contain significant measurement errors. We describe a method for characterizing a time series for any assumed level of measurement error ε by a sequence of intervals, each of which is guaranteed to contain an extremum for any function that ε-approximates the time series. Based on the merge tree of a continuous function, we define a new object called the normalized branch decomposition, which allows us to compute intervals for any level ε. We show that there is a well-defined total order on these intervals for a single time series, and that it is naturally extended to a partial order across a collection of time series comprising a dataset. We use the order of the extracted intervals in two applications. First, the partial order describing a single dataset can be used to pattern match against switching model output (Cummins et al. in SIAM J Appl Dyn Syst 17(2):1589–1616, 2018), which allows the rejection of a network model. Second, the comparison between graph distances of the partial orders of different datasets can be used to quantify similarity between biological replicates.

Keywords Time series · Merge trees · Order of extrema · Partial orders

Mathematics Subject Classification 05C12 · 06A06 · 37M10

1 Introduction

Time series data provide a discrete measurement of a dynamical system. By collecting simultaneous time series measuring different components of a dynamical system, we

✉ Bree Cummins
breschine.cummins@montana.edu

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

2 Biology Department, Duke University, Durham, NC, USA

can infer potentially causal relationships between components (Albert 2007; Sugihara et al. 2016; Cummins et al. 2015; McGoff et al. 2016). These relationships are represented in the form of a regulatory network, deduced from data experimentally or via *network learning* (McGoff et al. 2016; Akutsu et al. 2000; Brunton et al. 2016; Maucher et al. 2011; Lähdesmäki et al. 2003; Barker et al. 2011; Carré et al. 2017). Our recent work Cummins et al. (2016) has focused on the *post hoc* study of admissible dynamics for these network models. We associate to each network model a *switching system* (Albert et al. 2013; Edwards 2001; Glass and Kauffman 1973; Thomas 1991), in which each node is modeled by a piecewise linear ODE.

In Cummins et al. (2016), we introduced a method to describe the global dynamics of a regulatory network for all parameterizations of the switching system. This approach, named DSGRN (Dynamic Signatures Generated by Regulatory Networks) (Harker 2018), leverages the fact that the switching system admits a finite decomposition of phase space into domains, and each variable of the dynamical system can be assigned one of the finite number of states representing these domains. Furthermore, the dynamics of the switching system can be described by capturing transitions between these states via a *state transition graph*. Finally, the switching systems admit an explicit finite decomposition of parameter space in regions where the state transition graph is constant. DSGRN provides a complete combinatorialization of dynamics in both phase space and parameter space.

The solution trajectories of the parameterized switching system correspond to the sequences of discrete states determined by paths through the state transition graph. From the sequence of states, one can deduce the sequences of maxima and minima (extrema) of the admissible solution trajectories.

In a recent paper (Cummins et al. 2018), we compared a sequence of extrema generated by paths in the DSGRN switching model to experimentally observed sequences of maxima and minima in time series. We were seeking consistency between a network model and a time series dataset by checking if the ordering of extrema in the dataset is compatible with the ordering of extrema along a path in the state transition graph. If such a match could not be found, we proposed that the network model be rejected as a valid model of the underlying biological system producing time series data.

The sparseness of measurements in time series may not accurately capture the timing of an extremum. As a consequence, it may be difficult to ascertain differences in ordering between extrema in different time series with high confidence. To account for these issues, Cummins et al. (2018) proposed that each extremum in the dataset should be assigned a time interval representing the level of uncertainty in the timing of the extremum. If two time intervals overlapped, then the relative ordering of the extrema was taken to be unknown. If the intervals were disjoint, then the relative order was known. This relation formed a partial order of extrema representing the time series dataset.

In Cummins et al. (2018), the intervals were chosen in an ad hoc fashion. In this manuscript, we describe how to extract the desired time intervals about each extremum in a dataset $\mathcal{D}$ in a rigorous and automated fashion.

We consider our dataset $\mathcal{D}$ to be a collection of n time series

$$\mathcal{D} = (D_1, D_2, \ldots, D_n)$$

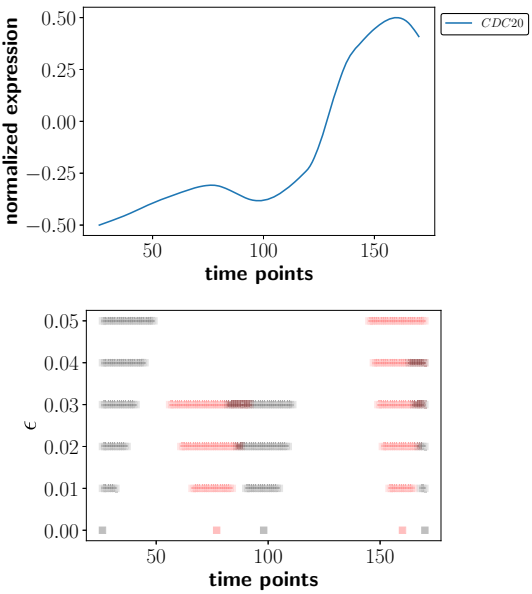

Fig. 1 Top: A smoothed and normalized time series. Bottom: The ε-extremal intervals as a function of increasing noise ε. The gray lines correspond to local minima and the reddish lines are associated to local maxima (color figure online)

measured at a common discrete set of time points $(z_1, \ldots, z_k)$. We assume that the measurement error of size ε is additive and thus the "true" time series T_i lies in a band of size 2ε about measured time series D_i. Our goal is to compute a collection time intervals $\mathcal{I}_\varepsilon(D_i)$, each of which is guaranteed to contain an extremum of the true time series T_i. Using the techniques of merge trees (Edelsbrunner and Harer 2010; Morozov et al. 2013), branch decompositions (Morozov and Weber 2013), and sublevel sets, we construct time intervals $\mathcal{I}_\varepsilon(D_i)$ that increase in length as a function of increasing ε (see Sect. 4). We call this construction the *ε-extremal interval method*.

As an example, see Fig. 1. On top is a time series curve D_1 that has been interpolated to smooth it (see "Appendix A") and normalized to the range $[-0.5, 0.5]$. It has five local extrema including the endpoints. In the second row of Fig. 1, there is a visualization of $\mathcal{I}_\varepsilon(D_1)$ as a function of increasing noise ε. The reddish lines are the intervals associated to local maxima and the gray lines are associated to the local minima.

The bottom row in the lower plot has five points marked, each of which is an interval of length zero corresponding to the case without noise, $\varepsilon = 0$. At a noise level of $\pm 1\%$, the intervals have widened but remain distinct. At a noise level of $\pm 2\%$, some of the intervals have started to overlap. Somewhere between $\pm 3\%$ and $\pm 4\%$, the two intervals corresponding to the maximum and minimum near time point 100 have widened so much that they coincide. When two intervals coincide, it means the ε-band has become so large that the true time series T_1 is not guaranteed to have any extrema within that interval, and the interval is removed from $\mathcal{I}_\varepsilon(D_1)$. Between $\pm 4\%$ and $\pm 5\%$, the interval associated to the right endpoint minimum has become a proper subset of the interval of the adjacent maximum. The interval containing the maximum continues to the 5% noise level, while the interval associated to the

rightmost minimum has been removed from $\mathcal{I}_\varepsilon(D_1)$. Every time an interval becomes a proper subset of another interval, the smaller interval is removed and the larger one is retained in $\mathcal{I}_\varepsilon(D_1)$. The reason is because the true time series T_1 is only guaranteed to attain the extremum associated to the larger interval.

The top row in the lower plot provides a description of the time series assuming a measurement error of $\pm 5\%$. At this level of precision, the time series is characterized by a global minimum well-separated from a global maximum. When the noise level increases enough that these last two intervals coincide, which they will do by a 50% noise level, then we say that the time series is *ε-constant*. At that point in time, all information about the extrema of the time series (if they exist) is is erased by noise.

As alluded to above, the key property that guides the construction of the collection of intervals $\mathcal{I}_\varepsilon(D_i)$ is the following. Let l be the linear interpolation of a time series D_i and let e be its local extremum. For a fixed measurement error level ε, let $I \in \mathcal{I}_\varepsilon(D_i)$ be the ε-extremal interval containing e. We prove in Sect. 4 that every continuous function whose graph lies within the band $l \pm \varepsilon$ must attain an extremum of the same type as e (maximum or minimum) within the interval I. Moreover, I is the minimal such interval about e with this property, up to the discretization level of the time series. This means that the intervals $\mathcal{I}_\varepsilon(D_i)$ describe the position of extrema of a time series in a way that is robust to noise in the most precise way possible given the information in the dataset. Naturally, if important extrema are missing from the dataset due to sparse time points, then this method cannot guess at their existence.

Using our approach, a dataset $\mathcal{D}$ is represented as a strict partial order $P_\varepsilon(\mathcal{D}) = (\mathcal{I}_\varepsilon(\mathcal{D}), \lhd)$ of the collection of intervals representing extrema for all time series in the dataset. If $I \in \mathcal{I}_\varepsilon(\mathcal{D})$ is an interval in time series D_i and $J \in \mathcal{I}_\varepsilon(\mathcal{D})$ is an interval in time series D_j with $i \neq j$, then $I \lhd J$ if and only if the right endpoint of I is less than or equal to the left endpoint of J. In other words, I is comparable to J if and only if the intervals are disjoint, except possibly at a single point. When intervals are disjoint, then the extrema they represent are well-separated and can be unambiguously distinguished. Since not all pairs of intervals in $\mathcal{I}_\varepsilon(\mathcal{D})$ are comparable, the intervals are only partially ordered.

Each partial order $P_\varepsilon(\mathcal{D})$ can be represented as a graph $H_\varepsilon^R(V, E)$, called the Hasse diagram, with nodes V corresponding to intervals in $\mathcal{I}_\varepsilon(\mathcal{D})$, and an edge in $E \subset V \times V$ whenever $I \lhd J$. This graph is a coarse representation of the information contained in the dataset, containing qualitative rather than quantitative phase shifts.

As ε increases, there are fewer disjoint intervals in $\mathcal{I}_\varepsilon(\mathcal{D})$, fewer well-ordered pairs of intervals, and therefore fewer restrictions on the ordering of extrema between time series. This results in a more permissive partial order, where the amount of trusted information decreases as assumed noise level increases. Eventually, ε increases to the point where some neighboring extrema become indistinguishable and the number of nodes in $H_\varepsilon^R(V, E)$ decreases. We illustrate this on an example in Fig. 2. In the upper left, there are two interpolated and normalized time series. The blue curve is the same as the one in Fig. 1 (top), and has five local extrema. The second orange curve has four local extrema including endpoints. We calculate the ε-extremal intervals for both curves for $\varepsilon = 0.0, 0.03,$ and 0.05 and get the Hasse diagrams of the partial orders shown in the figure. In every Hasse diagram, the arrow of time points downward.

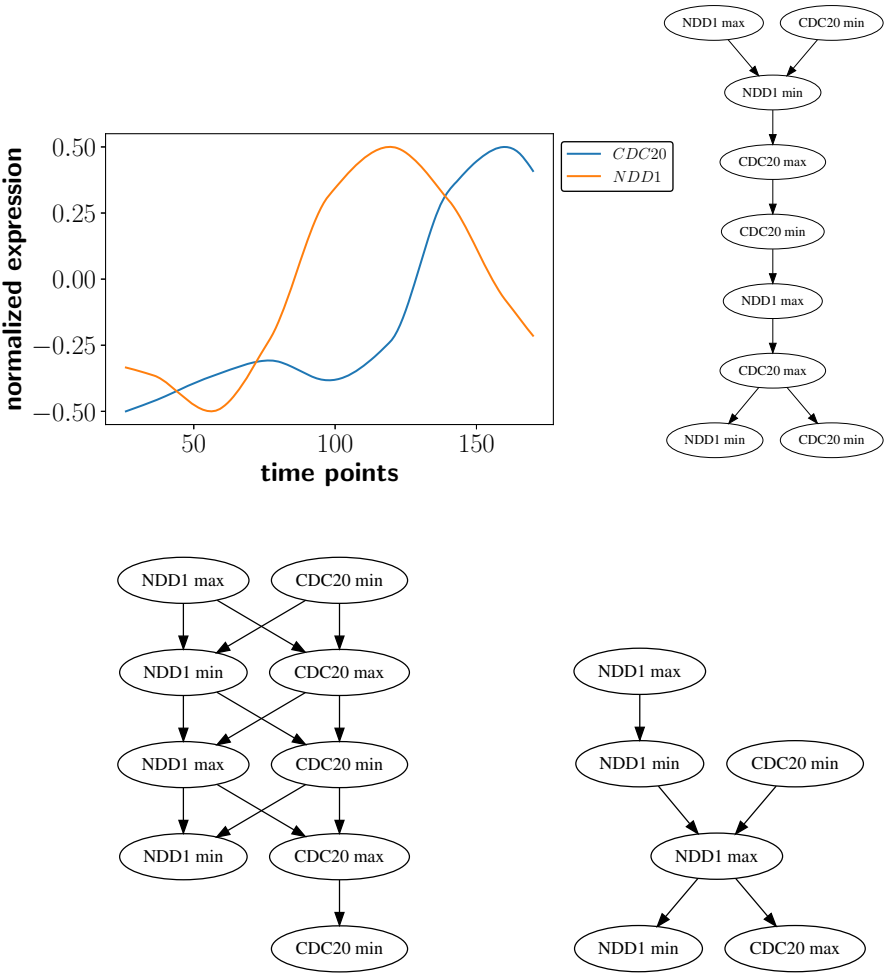

Fig. 2 Hasse diagrams of two time series as a function of ε. Upper left: The time series under consideration. Upper right: $\varepsilon = 0.0$. Lower left: $\varepsilon = 0.03$. Lower right: $\varepsilon = 0.05$

At the upper right in Fig. 2 is the Hasse diagram for $\varepsilon = 0.0$, which corresponds to the case without noise, and all nine local extrema are represented. At $\varepsilon = 0.03$ (lower left), all nine extrema are still present, but the greater number of incomparable ε-extremal intervals indicates a more permissive partial order. For example, at $\varepsilon = 0$, the first NDD1 minimum (the one closest to the top of the Hasse diagram) must occur before the first CDC20 maximum. But when $\varepsilon = 3\%$ those two extrema are incomparable, as indicated by the lack of an arrow between them in the Hasse diagram on the lower left. At the lower right with $\varepsilon = 5\%$, there are only six extrema. CDC20 has lost its two middle extrema and its rightmost minimum, as shown in Fig. 1 (bottom) where $\varepsilon = 0.05$.

As motivation for the work in Sect. 4, we discuss the applications of our method after a brief introduction to some standard definitions in graph theory. Application 1

demonstrates that the ε-extremal interval method can be used for consistency-checking a DSGRN model of network dynamics as suggested in Cummins et al. (2018). Application 2 shows that the technique can be used to quantify the similarity between different time series datasets.

For the purpose of the presentation of the applications we will ask the reader to accept that the collection $\mathcal{I}_\varepsilon(\mathcal{D})$, the partial order $P_\varepsilon(\mathcal{D}) = (\mathcal{I}_\varepsilon(\mathcal{D}), \lhd)$, and its representation $H^R_\varepsilon(V, E)$ can be unambiguously constructed for any dataset $\mathcal{D}$ and any ε. Rigorous technical detail for the method of ε-extremal intervals is given in depth in Sect. 4, along with information on the computational implementation (Cummins and Nerem 2019). Section 4 may be read before the applications in Sect. 3 if desired.

2 Graph theory preliminaries

Definition 1 A directed graph $G(V, E)$ is a set of nodes (or vertices) V, together with a collection of edges $E \subseteq V \times V$. A labeled, directed graph $G(V, E, \ell)$ has in addition a labeling function ℓ that assigns labels to the nodes and/or edges of a graph.

We will refer to all of unlabeled, node-labeled, and node- and edge-labeled graphs in this manuscript. One important example is a gene regulatory network, which is a node- and edge-labeled directed graph. Each node is labeled by the gene product that it represents, and every edge is labeled either as an activating ($\rightarrow$) or repressing ($\dashv$) edge. The gene regulatory network that we will explore in Application 1 is shown in Fig. 3.

Definition 2 A partial order $P = (S, \leq)$ is a binary relation $\leq$ on the set S that is reflexive, antisymmetric, and transitive. A strict partial order $P = (S, <)$ is a binary relation $<$ on the set S that is antisymmetric and transitive. A (strict) total order is a (strict) partial order where for any pair $(a, b) \in S \times S$ either $a \leq b$ ($a < b$) or $b \leq a$ ($b < a$). A linear extension of a (strict) partial order is a (strict) total order T such that if $a \leq b$ ($a < b$) in P, then $a \leq b$ ($a < b$) in T.

In this manuscript, we will only be concerned with strict partial and strict total orders. Our notation reflects this, but for brevity we will often refer only to partial and total orders.

Every partial order can be represented as a directed graph called a Hasse diagram. To explain Hasse diagrams, it is useful to know the concepts of transitive closures and transitive reductions. A transitive closure adds a direct edge wherever there is a path between two nodes, and a transitive reduction removes an edge whenever there is a longer path from one node to another.

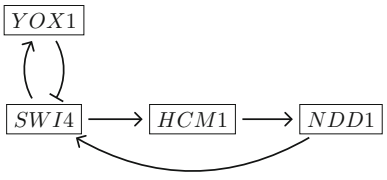

Fig. 3 Wavepool model of core genes involved in regulation of the yeast (*S. cerevisiae*) cell cycle (Cho et al. 2019)

Definition 3 Let $G(V, E)$ be a directed graph. A node j is reachable from a node i in the graph G if there exists a path

$$(i, i_1), (i_1, i_2), \ldots, (i_n, j)$$

such that each edge $(i_j, i_{j+1}) \in E$. The transitive closure of $G(V, E)$ is the directed graph $G^C(V, E')$ with $E \subseteq E'$ such that $(i, j) \in E'$ if and only if j is reachable from i in G The transitive reduction of $G(V, E)$ is the directed graph $G^R(V, E'')$ with the minimal set of edges $E'' \subseteq E$ such that if the vertex $j \in V$ is reachable from $i \in V$ in the graph G, then j is reachable from i in G^R.

Definition 4 Let $P = (S, <)$ be a strict partial order on a finite set S. Let $H(V, E)$ be a directed graph where the nodes V are in a bijection with the elements of the set S, and an edge $(i, j) \in E$ if and only if $i < j$. The Hasse diagram $H^R(V, E'')$ of P is the transitive reduction of H.

Note that the graph H defined in Definition 4 is its own transitive closure, $H(V, E) = H^C(V, E)$, because the partial order P is transitively closed. For clarity, we will refer to H^C rather than H in the text. The Hasse diagram H^R plays a role in both Applications 1 and 2, and the transitive closure of the Hasse diagram H^C is important in Application 2. We will use the notation $H^R_\varepsilon(\mathcal{D})$ and $H^C_\varepsilon(\mathcal{D})$ for a given dataset $\mathcal{D}$ and noise level ε.

Every partial order P is associated to a unique distributive lattice. This is the content of Birkhoff's Representation Theorem (see Chapter 5 in Davey and Priestley 2002).

Definition 5 Let $P = (S, <)$ be a strict partial order over a finite set S. A down-set is any set $Q \subseteq S$ such that if $a \in Q$ and $b < a$, then $b \in Q$. A down-set lattice is the partial order $P' = (\mathcal{O}(P), \subset)$, ordered by set inclusion, where $\mathcal{O}(P)$ is the collection of down-sets of P.

The down-set lattice has a minimal element, the empty set, and a maximal element, S. In the Hasse diagram of the down-set lattice, the two nodes associated to these sets are called the root and the leaf respectively. For brevity, we will use the term down-set lattice interchangeably with the Hasse diagram of the down-set lattice.

Remark 1 The paths from root to leaf in the down-set lattice are in a one-to-one correspondence with the collection of linear extensions of P. This observation plays a central role in Application 1.

Definition 6 A graph distance is a non-negative, real-valued, symmetric function $d(G_1, G_2)$ acting on two graphs G_1 and G_2 that satisfies the triangle inequality and that is zero if and only if $G_1 = G_2$.

The graph distance described in Sect. 4.3.3 for node-labeled, directed graphs is used in Application 2 to assess the similarity between two time series datasets.

3 Applications

3.1 Model rejection via pattern matching

3.1.1 DSGRN

In our previous work (Cummins et al. 2016) we introduced a method to describe the global dynamics of a regulatory network for all parameters. This approach, named DSGRN (Dynamic Signatures Generated by Regulatory Networks) (Harker 2018), uses combinatorial dynamics generated by switching systems (Albert et al. 2013; Edwards 2001; Glass and Kauffman 1973; Thomas 1991) to construct a database of all possible dynamics that a network may exhibit.

The core procedure of DSGRN (Cummins et al. 2016; Harker 2018) is the following:

1. A switching system ODE model is associated to a gene regulatory network. The dimension of phase space is the number of nodes in the network, N. The dimension of parameter space is $3M + N$, where M is the number of edges in the network.
2. The form of switching systems allows parameter space $\mathbb{R}_+^{3M+N}$ to be decomposed into a finite number of semi-algebraic regions. We call each region a *DSGRN parameter*.
3. The form of switching systems also admits a decomposition of phase space $\mathbb{R}_+^N$ into a finite number of rectangular regions (boxes). The number of boxes depends on the number of discrete states that each gene product can attain, which is exactly one more than the number of out-edges from the corresponding node.
4. The dynamics of the switching system for a DSGRN parameter are captured by a state transition graph, in which each node is associated to one of the boxes in phase space, and edges capture the direction of the normal component of the vector field at a boundary between two boxes. This assignment is consistent if there is no negative self-regulation in the regulatory network (Edwards 2001). A solution trajectory of the switching system then corresponds to a path in the state transition graph.

DSGRN produces a finite collection of state transition graphs that captures the parameter dependence of the dynamics across all of parameter space.

Important to the application here, the nodes of the state transition graph can be labeled by whether each gene product is increasing (I), decreasing (D), or both ($*$) in the corresponding phase space box. Likewise, the edges can be labeled by which variable is attaining an extremum between boxes (M for maximum, m for minimum, a dash for neither). By assumptions on the switching system, only one gene product may attain an extremum at a time. An example labeled state transition graph for a fixed DSGRN parameter for the network in Fig. 3 is shown in Fig. 4.

3.1.2 DSGRN model consistency with a dataset

In a recent paper (Cummins et al. 2018), we compared a DSGRN model of molecular regulation to experimentally observed time series. We proposed that a network can be rejected as a model of a biological system that produces the experimental dataset D

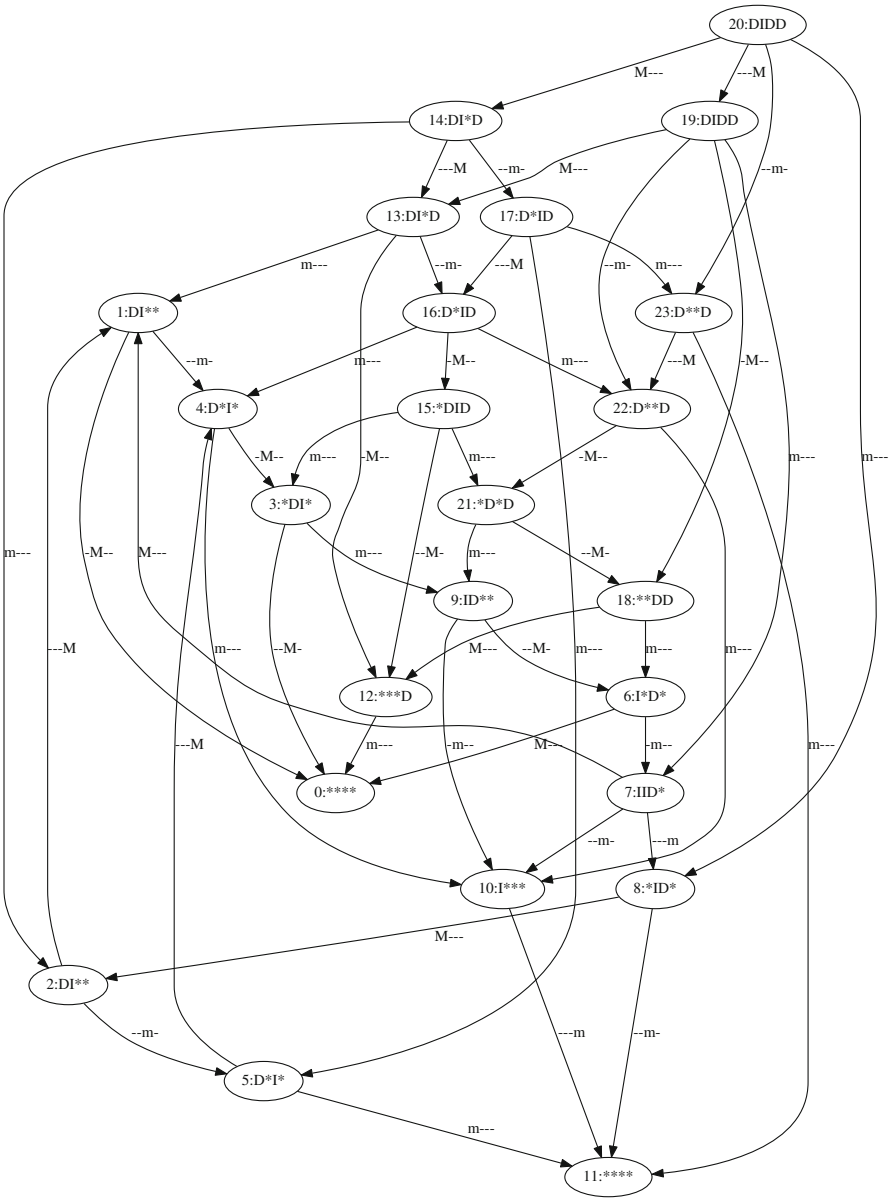

Fig. 4 Example state transition graph for the network in Fig. 3. This is a state transition graph as described in the text, with 4-symbol labels indicating whether a variable is increasing (I), decreasing (D), or both ($*$) in the corresponding partition of phase space. The order of the symbols is $SWI4$, $HCM1$, $NDD1$, and $YOX1$. The edge labels indicate the possibility of a local maximum (M) or minimum (m) for each variable. The dash indicates that no extremum is possible for that variable

if there is no DSGRN parameter at which a path in a DSGRN state transition graph is "consistent" with the partial order $P_\varepsilon(\mathcal{D})$ derived from a dataset.

A path is consistent with partial order $P_\varepsilon(\mathcal{D})$ if it is a linear extension of $P_\varepsilon(\mathcal{D})$. Any linear extension of $P_\varepsilon(\mathcal{D})$ is a total order of the extrema consistent with the time discretization and the measurement error level ε. Therefore a path in the state transition graph that is a linear extension of $P_\varepsilon(\mathcal{D})$ represents a solution trajectory that has an order of extrema consistent with noise level ε in the data. If such an extension exists, the network cannot be rejected as a valid model.

Because we are seeking a linear extension, we make use of the down-set lattice structure introduced in Definition 5, since it is a summary of all linear extensions of $P_\varepsilon(\mathcal{D})$. First, the lattice is augmented with labels. In the Hasse diagram of $P_\varepsilon(\mathcal{D})$, $H_\varepsilon^R(\mathcal{D})$, the nodes can be naturally labeled with extrema. Therefore one can unambiguously label the edges of $H_\varepsilon^R(\mathcal{D})$ based on whether the gene product is increasing or decreasing, which is opposite to the labeling on the state transition graph. We showed that there is a natural way to assign dual labeling to the down-set lattice. In other words, the lattice can be labeled unambiguously with extrema on the edges and increasing or decreasing behavior on the nodes. Second, the lattice is augmented with self-edges at every node. Since gene products may monotonically increase or decrease in concentration across multiple boxes before reaching an extremum, self-loops were added to represent this dwell time. The resulting graph is called a *pattern graph*, and has a unique root node and a unique leaf node. An example is shown in Fig. 6 based on the time series introduced in Fig. 5.

We seek a pair of paths with matching labels on both nodes and edges, where one path goes from root to leaf in the pattern graph and the other path is in the state transition graph. The matching relation between labels that we define allows for non-exact matches; namely the symbol * is allowed to match both I and D. This is a type of *approximate graph matching* (Bunke and Riesen 2011; Conte et al. 2004; Livi and Rizzi 2012; Fu 1996). If such a pair exists, then the model is consistent with the data and cannot be rejected. In Cummins et al. (2018), we formulate this consistency problem as a graph theory problem with a polynomial time algorithm.

In Figure 5 of Cummins et al. (2018), the partial order $P_\varepsilon(\mathcal{D})$ was calculated by hand via visual inspection of the data. We illustrate the approach developed in this paper on the example from Cummins et al. (2018).

3.1.3 Cell cycle data model

In Fig. 5 (upper right), we present microarray time series data from the yeast *S. cerevisiae*, which was normalized between -0.5 and 0.5. The data were published in Orlando et al. (2008), and have here undergone shifting via CLOCCS analysis (Orlando et al. 2007) and smoothing via polynomial splines, as described in "Appendix A". These data are similar to those in Cummins et al. (2018) (Section 4.2, data published in Kelliher et al. 2016), which are data collected on the same genes from the same organism in the same lab, but using RNAseq technology rather than the microarray platform.

We show two partial orders arising from this data at two noise levels using the ε-extremal interval method. The Hasse diagrams are shown in Fig. 5 at $\varepsilon = 0.0$ (left)

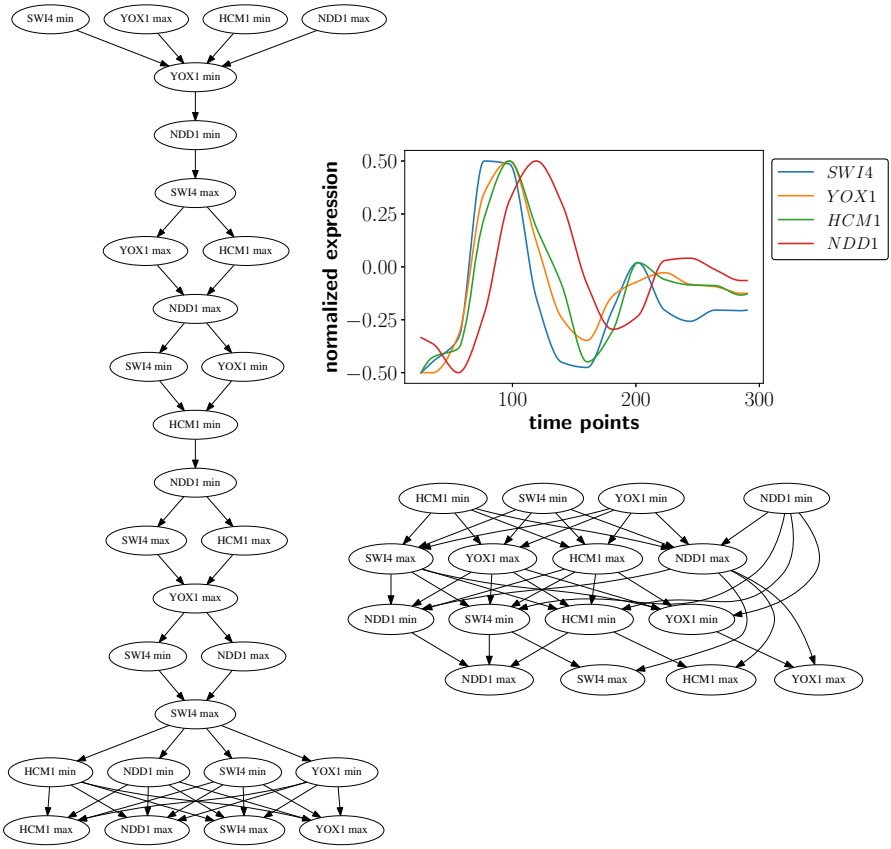

Fig. 5 *S. cerevisiae* time series data Orlando et al. (2008) (upper right) and example $H_\varepsilon^R(\mathcal{D})$ for the data at $\varepsilon = 0.0$ (left) and $\varepsilon = 0.15$ (lower right). The arrow of time points downward on the Hasse diagrams

and 0.15 (lower right). The number of extrema at a noise level of 0 is 28, and drops to 16 at a noise level of 0.15. The partial order at $\varepsilon = 0.0$ is far more restrictive (and thus looks closer to a total order), because few of the intervals overlap, and more order relations are known. The pattern graph associated to the partial order on the right is shown in Fig. 6.

The regulatory network in Fig. 3 is a simplified version of the *wavepool model*, introduced in Cho et al. (2019). This model has been corroborated experimentally (Kovacs et al. 2012; Bristow et al. 2014; Cho et al. 2017) and describes the mechanism for controlling the cell-cycle transcriptional program.

In Cummins et al. (2018), we verified that the wavepool model is consistent with the data. We find the same result here. In particular, there is a pair of matching paths between Figs. 4 and 6 of length 17:

$$(18, 38) \to (6, 37) \to (7, 34) \to (8, 32) \to (2, 31) \to (5, 42) \to (4, 25) \to (3, 22) \to (9, 21) \to (6, 19) \to (7, 65) \to (8, 63) \to (2, 13) \to (5, 80) \to (4, 7) \to (3, 4) \to (0, 0)$$

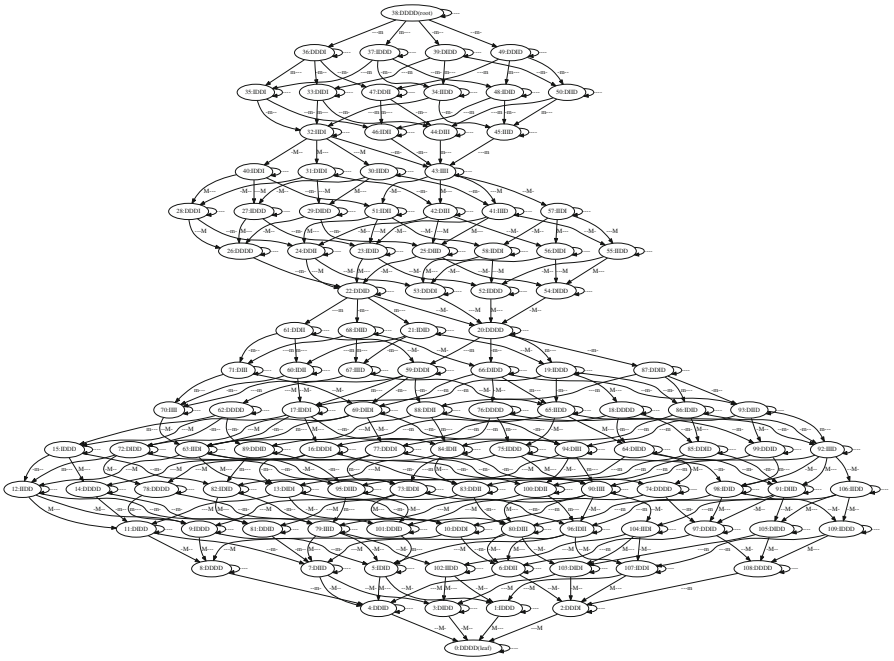

Fig. 6 Corresponding pattern graph for $H^R_{0.15}(\mathcal{D})$ in Fig. 5 (lower right). The 4-symbol node labels with I (increasing) and D (decreasing) and the edge labels with m (minimum) and M (maximum) symbols have the same meaning as in Fig. 4. Notice the unique root and unique leaf

The first number is the integer node label in Fig. 4, and the second is the integer node label in Fig. 6. The first node pair is at the root of the pattern graph (node 38) and the last node pair is at the leaf of the pattern graph (node 0). It can be verified that the labels at each pair of nodes match, remembering that the wild card character * matches itself, I, and D. Likewise, the edge labels match as well. So at the DSGRN parameter that produced the state transition graph in Fig. 4, the wavepool network model cannot be rejected.

3.1.4 Global assessment of the network model

We seek to characterize the performance of the wavepool model across parameter space and across various levels of noise. Recall that we use the variable ε to denote a band of noise around a time series. Recall also that the partial order representing the dataset $P_\varepsilon(\mathcal{D})$ depends on ε, so that the pattern graph depends on ε as well. We independently normalized each time series in the dataset between -0.5 and 0.5 so that ε represents a percentage of the distance between the global maximum and global minimum of each time series in the dataset.

The wavepool network in Fig. 3 has 1080 DSGRN parameters. For ε values ranging from 0.0 to 0.15, we searched for pairs of matching paths between the pattern graph and the state transition graph at each DSGRN parameter using the DSGRN pattern matching algorithm described in Cummins et al. (2018) and implemented in Harker

Table 1 Number of parameters with at least one match to $P_\varepsilon(\mathcal{D})$

ε	0.0	0.01	0.04	0.05	0.06	0.08	0.09	0.10	0.14	0.15
Figure 3	0	22	22	12	12	42	24	24	24	24
Figure 7	0	0	0	0	0	0	0	0	22	22

(2018). See Fig. 5 for the partial orders $P_{0.0}(\mathcal{D})$ and $P_{0.15}(\mathcal{D})$ showing the extremes of the representation of the dataset.

The results are summarized in the first row of Table 1. There are 22 DSGRN parameters in the wavepool network that exhibit consistent dynamics with the time series in Fig. 5 (upper right) for noise levels $\varepsilon = 0.01$ through 0.04. This is the same number of DSGRN parameters with path matches found in Cummins et al. (2018).

Although the number of matches is the same, none of the partial orders $P_\varepsilon(\mathcal{D})$ match the one computed by hand in Cummins et al. (2018). This is likely due to the difference in sampling times; the RNAseq data in Cummins et al. (2018) have a sampling interval of 5 min, while the microarray data have a sampling interval of 16 min. This discrepancy can be responsible for different resolutions in peak detection, and therefore change the representative partial orders. It is also possible that the CLOCCS shifting of the microarray dataset resulted in a small phase shift with respect to the RNAseq data, which could alter the relative locations of the extrema.

In addition to the difference in partial orders, we also do not know if the collection of parameters at which there are pattern matches is the same between this work and the previous one. However, the fact that the model has comparable performance under different data collection platforms and sampling intervals highlights both the reproducibility of the performance of the *S. cerevisiae* cell cycle, and the power of this technique to identify/reject models based on time series data.

We note that at $\varepsilon = 0$, the wavepool model has no matches to the partial order in Fig. 5 (left). Without considering noise, we would be tempted to reject the hypothesis that the wavepool model is consistent with the experimental data. However, the ability to scan through different potential noise levels reveals that the wavepool model cannot be rejected, as it consistently matches the data over a range of small noise levels.

Finally, we observe that the number of matches does not increase monotonically with ε. This may at first seem counterintuitive, since as intervals grow larger and overlap, the number of constraints in the partial order is reduced. However, nodes in the graph representing shallow extrema in the time series disappear as ε grows larger, which changes the size of the graph. Under these conditions, there is no guarantee of monotonicity. In our example here, the number of nodes in the partial order decreases from 28 to 16 as ε increases from 0 to 0.15. At $\varepsilon = 0.08$, we find the highest number of parameters with matches, 42.

3.1.5 Model rejection

In a second numerical experiment, we propose a different model for the same time series data, depicted in Fig. 7. Here we swapped the positions of the genes $NDD1$ and $SWI4$ compared with the first model, thus creating an "incorrect" model for the

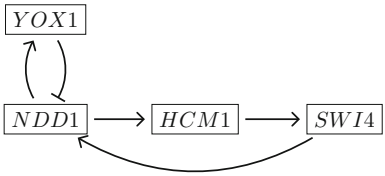

Fig. 7 A model in which two genes of the wavepool model are swapped, which our methodology invalidates

data. In Cummins et al. (2018), no matches were found for this model using the partial order computed by hand and shown in Figure 5 of that work. The number of matches that we find now for this network are shown in second row of Table 1. We observe that there are no matches in this network until noise level $\varepsilon = 0.14$, so that we match (Cummins et al. 2018) in the range $\varepsilon = 0.01$ through 0.04 as before. At a noise level of $\varepsilon = 0.14$, the band of uncertainty around the data is 28% of the difference between the global maximum and the global minimum in the normalized data. This a very high level of noise, and we hypothesize that the partial order at this noise level does not sufficiently constrain the model, leading to matches with models that are inconsistent with experimental results. The large range of noise levels with zero matches would lead us to reject the network in Fig. 7 as a description of the biological mechanism that produced the time series data.

3.2 Quantifying similarity between replicate experiments

In this application, we seek to quantify the similarity between two replicates of the same experiment with datasets $\mathcal{D}_1$ and $\mathcal{D}_2$. We construct for each dataset partially ordered sets of ε-extremal intervals $P_\varepsilon(\mathcal{D}_1)$ and $P_\varepsilon(\mathcal{D}_2)$, and represent them by the transitive closures of Hasse diagrams $H_\varepsilon^C(\mathcal{D}_1)$ and $H_\varepsilon^C(\mathcal{D}_2)$ as in Definitions 3 and 4. We then calculate a graph distance between $H_\varepsilon^C(\mathcal{D}_1)$ and $H_\varepsilon^C(\mathcal{D}_2)$, which is roughly the proportion of non-shared edges between the two graphs; see Sect. 4.3.3 for a precise definition. The distance $d(H_\varepsilon^C(\mathcal{D}_1), H_\varepsilon^C(\mathcal{D}_2))$ that we use is scaled so it has a range between 0 and 1. We say that the *similarity* between two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ is given by

$$s(\mathcal{D}_1, \mathcal{D}_2) = 1 - d(H_\varepsilon^C(\mathcal{D}_1), H_\varepsilon^C(\mathcal{D}_2)).$$

This similarity measure gives roughly the proportion of shared edges between $H_\varepsilon^C(\mathcal{D}_1)$ and $H_\varepsilon^C(\mathcal{D}_2)$.

In order for the comparison of the two datasets to be biologically relevant, they must be synchronized at the same point in the yeast cell cycle. In the datasets we consider, the time series were processed to align with the yeast cell cycle using the techniques described in "Appendix A". After this processing, we normalized the data to the range $[-0.5, 0.5]$ and truncated so that most of the time series exhibited one period, as shown in Fig. 8. The truncation limits the computation time of the graph distance, and focuses the analysis on the highest and most synchronized peaks.

To illustrate the properties of the similarity measure, we perform four different comparison experiments.

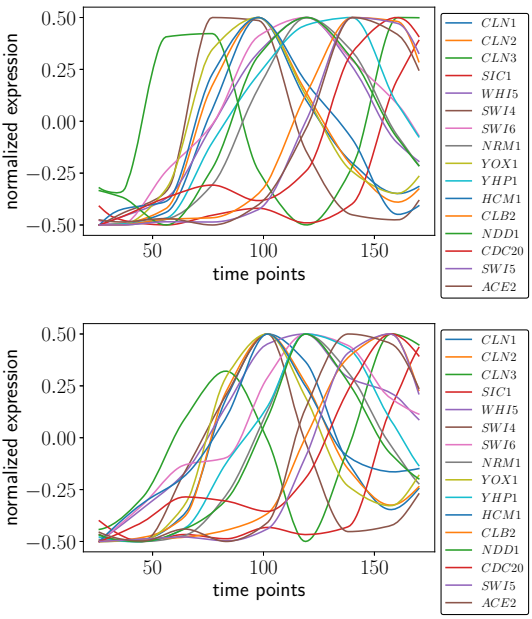

Fig. 8 Microarray yeast cell cycle data from Orlando et al. (2008), processed as described in "Appendix A". (Top) Replicate 1. (Bottom) Replicate 2

1. We first concentrate on only four genes, $SWI4$, $YOX1$, $NDD1$, and $HCM1$. We denote by $\mathcal{D}'_1$ and $\mathcal{D}'_2$ the time series of these four genes extracted from experiments $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively (see Fig. 9 top row). We calculate the similarity $s(\mathcal{D}'_1, \mathcal{D}'_2)$ over a range of ε.
2. We compute $s(\mathcal{D}'_1, \mathcal{D}''_2)$, where $\mathcal{D}''_2$ is the dataset formed by time series $SWI4$, $CLB2$, $NDD1$, and $HCM1$. Here we replace the time series of $YOX1$ in the second dataset by the time series of $CLB2$, where the $CLB2$ time series can be seen in Fig. 8.
3. We compare the same data as in (2), but we mislabel $CLB2$ in $\mathcal{D}''_2$ as $YOX1$. We call the mislabeled dataset $\mathcal{D}''_3$. The calculation of $s(\mathcal{D}'_1, \mathcal{D}''_3)$ shows the effect of replacing the $YOX1$ data in $\mathcal{D}'_2$ by a different time series. The comparison between experiments (2) and (3) gives an idea of the impact on the distance measure when there are non-matching gene labels in the partial orders.
4. Lastly, we compare all of the time series by randomly sampling four genes to construct datasets $\hat{\mathcal{D}}_1$ and $\hat{\mathcal{D}}_2$ one hundred times, and calculating $s(\hat{\mathcal{D}}_1, \hat{\mathcal{D}}_2)$ for each over a range of ε. The mean of these curves is taken to be representative of the full dataset. The same experiment is performed with random samples of eight genes to show the dependence of results on gene sample size.

Experiment 1:

We compute ε-extremal intervals to produce a partially ordered set of extrema for $\mathcal{D}'_1$ and $\mathcal{D}'_2$. As an example, the Hasse diagrams $H^R_{0.01}(\mathcal{D}'_1)$ and $H^R_{0.01}(\mathcal{D}'_2)$ for $\varepsilon = 0.01$ are shown in the bottom row of Fig. 9. Although we use the transitive closure $H^C_\varepsilon(\mathcal{D})$ to calculate distances, the transitive reductions are shown for simplicity, so that the structures of the partial orders are easier to compare.

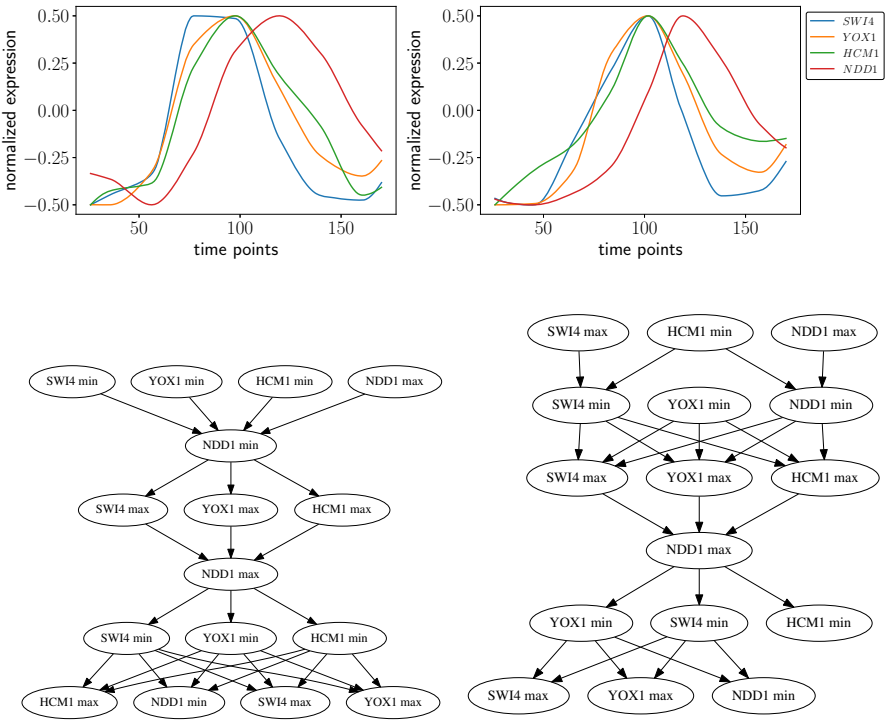

Fig. 9 (Top left) Time series for replicate 1. (Top right) Time series for replicate 2. (Bottom left) Hasse diagram $H^R_{0.01}(\mathcal{D}'_1)$ for replicate 1 at $\varepsilon = 0.01$. (Bottom right) Hasse diagram $H^R_{0.01}(\mathcal{D}'_2)$ for replicate 2 at $\varepsilon = 0.01$. The arrow of time points downward on the Hasse diagrams

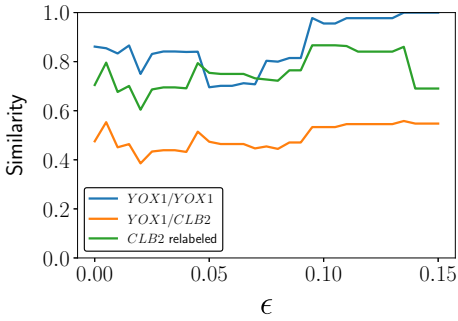

Fig. 10 Similarity as a function of ε. Blue curve: Experiment 1, $s(\mathcal{D}'_1, \mathcal{D}'_2)$. Orange curve: Experiment 2, $s(\mathcal{D}'_1, \mathcal{D}''_2)$. Green curve: Experiment 3, $s(\mathcal{D}'_1, \mathcal{D}''_3)$ (color figure online)

We repeat this procedure at noise levels $\varepsilon = 0$ to 0.15 at intervals of 0.005. The similarity $s(\mathcal{D}'_1, \mathcal{D}'_2)$ was calculated at each ε and represented as the blue curve in Fig. 10. Notice that the similarity between the partially ordered sets goes to 1 at larger values of ε and varies between 0.7 and 1.0 over the whole range of ε. To assess if this represents strong or weak similarity, we perform two other numerical experiments.

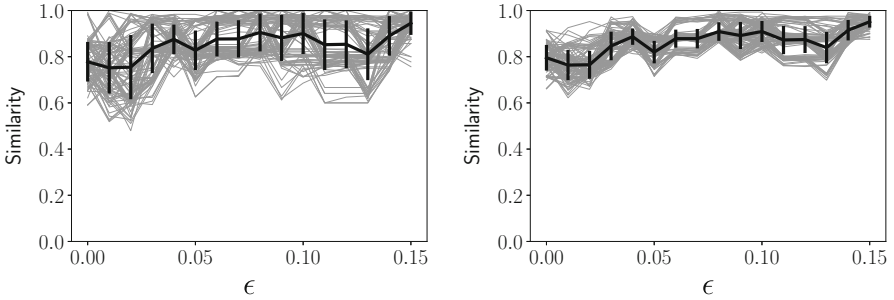

Fig. 11 Similarity $s(\hat{\mathcal{D}}_1, \hat{\mathcal{D}}_2)$ as a function of ε. Thin gray curves: one hundred samples of 4 genes each (left) and 8 genes each (right) from the 16 genes listed in the legend of Fig. 8. Thick black curve: Mean of the one hundred samples with $\pm$ 1 standard deviation

Experiment 2:

We replace the second data set $\mathcal{D}_2'$ by the dataset $\mathcal{D}_2''$ composed of genes $SWI4$, $CLB2$, $NDD1$, and $HCM1$, and we show the similarity $s(\mathcal{D}_1', \mathcal{D}_2'')$ in the orange curve in Fig. 10. Since the $YOX1$ extrema in replicate 1 are being compared with the $CLB2$ extrema in replicate 2, the distance between the partially ordered sets is larger than in Experiment 1. The similarity between the partial orders ranges between about 40–55%. This gives an idea of the distance when nodes cannot be matched across partial orders because of a single time series swap.

Experiment 3:

We relabeled all "$CLB2$" extrema in replicate 2 to "$YOX1$" labels to get dataset $\mathcal{D}_3''$. This allows us to compare the distance when the curve shape of $CLB2$ is used in place of the true $YOX1$ data. The resulting similarities $s(\mathcal{D}_1', \mathcal{D}_3'')$ are shown in the green curve in Fig. 10. By comparing the orange curve (Experiment 2) with the green curve (Experiment 3) in Fig. 10, it can be seen that having mismatched labels contributes substantially to the dissimilarity of time series. However, even when the mismatched labels are artificially removed with relabeling, we see by comparing the blue curve (Experiment 1) with the green curve (Experiment 3) in Fig. 10 that using the same set of time series in both replicates leads to a noticeably higher similarity over most of the range of ε.

Experiment 4:

We now consider all the time series in the dataset, as shown in Fig. 8, to quantify the similarity of the two replicates as a function of noise. Because our algorithm for graph distance does not scale favorably with the size of the graph, we chose to (a) sample ε more coarsely from 0 to 0.15 at intervals of 0.01, and (b) pick at random one hundred samples of four and eight genes each, and then calculate the mean similarity. The resulting curves are shown in Fig. 11 with samples of 4 genes on the left and 8 genes on the right. The thick black line indicates the mean and $\pm$ 1 standard deviation of the one hundred samples shown in grey. The mean on both curves ranges between about 75–95% similarity, with the bulk of the values over 80% similar. The standard deviation decreases substantially with increasing gene sample sizes, suggesting that

sampling subsets of genes is a reasonable proxy for calculating the similarity of the whole dataset.

4 Methods

The applications in the previous section are dependent on the representation of a time series dataset $\mathcal{D}$ by a partial order $P_\varepsilon(\mathcal{D})$ over time intervals $\mathcal{I}_\varepsilon(\mathcal{D})$ representing the location of extrema up to a noise level ε. We now present in detail the construction of the intervals $\mathcal{I}_\varepsilon(\mathcal{D})$.

We begin in Sect. 4.1 by establishing the theory for continuous functions. For $f : [x_1, x_2] \to \mathbb{R}$ a continuous function on a closed interval $[x_1, x_2]$, we present an approach that finds a collection of intervals $\mathcal{J}_\varepsilon(f)$, called ε-extremal intervals, with the property that any continuous function g whose values are within measurement error ε of f is guaranteed in each interval $I \in \mathcal{J}_\varepsilon(f)$ to attain a local extremum.

Our main tool is the notion of the *merge tree* of f (Edelsbrunner and Harer 2010; Morozov et al. 2013) that we use to define a new object, called the *normalized branch decomposition* of f on $[x_1, x_2]$. The normalized branch decomposition allows us construct a collection of ε-minimal intervals on $[x_1, x_2]$, such that every continuous function g that remains within the bounds $f - \varepsilon$ and $f + \varepsilon$ is guaranteed to achieve a minimum in each ε-minimal interval. By taking $-f$ and applying the same method, we construct ε-maximal intervals, in which every perturbation g of f bounded by ε is guaranteed to attain a local maximum. The union of ε-minimal and ε-maximal intervals forms the set $\mathcal{J}_\varepsilon(f)$.

The extension of merge trees and branch decompositions to discrete time series is straightforward (Smirnov and Morozov 2017). We show in Sect. 4.2 that ε-extremal intervals $\mathcal{I}_\varepsilon$ can also be assigned to a discrete time series by using the linear interpolation to construct a continuous function f. With this view, most of the same theorems hold for discrete time series as for general continuous functions.

In both discrete and continuous cases, there is a total order on the ε-extremal intervals for a single function f. In other words, the ε-extremal intervals represent a sequence of extrema of f that can be trusted up to measurement error level of ε. The total orders associated to a collection of functions $\{f_i\}$ derived from a time series dataset can be extended to a partial order on the extrema of $\{f_i\}$.

In Sect. 4.3, we discuss Algorithms 1 and 2 of Smirnov and Morozov (2017) that are used to compute merge trees and branch decompositions for a set of time series. We also discuss algorithms derived from Sect. 4.2 for constructing ε-extremal intervals, partial orders, and a graph distance for partial orders. We provide a repository (Cummins and Nerem 2019) in Python 3.7 that implements all the algorithms.

4.1 ϵ-Extremal intervals for continuous functions

Consider a continuous function, $f : [x_1, x_2] \to \mathbb{R}$, defined on a closed interval, $[x_1, x_2] \subset \mathbb{R}$.

Definition 7 Let $C([x_1, x_2])$ denote the space of continuous functions $g : [x_1, x_2] \to \mathbb{R}$, endowed with the supremum norm. For $\varepsilon \geq 0$, define

$$N_\varepsilon(f) := \{g \in C([x_1, x_2]) \ : \ |f - g| < \varepsilon\}$$

to be the ε-neighborhood of f. A function $g \in N_\varepsilon(f)$ will be called an ε-perturbation of f.

Given $\varepsilon \geq 0$, we would like to compute a collection of intervals, $\mathcal{J}_\varepsilon(f) := \{I_i^\varepsilon\}$, $I_i^\varepsilon \subset [x_1, x_2]$, not necessarily disjoint such that

1. every $g \in N_\varepsilon(f)$ attains either a minimum or a maximum in each I_i^ε, and
2. for any nonempty $J \subset I_i^\varepsilon$, there exists some $h \in N_\varepsilon(f)$ such that h does not attain a maximum or a minimum in J.

The collection $\mathcal{J}_\varepsilon(f)$ represents a set of extrema corresponding to a noise level of ε.

To construct $\mathcal{J}_\varepsilon(f)$ for all ε, we use merge trees (Edelsbrunner and Harer 2010; Morozov and Weber 2013; Morozov et al. 2013; Pascucci et al. 2004) to construct an associated object, which we call the normalized branch decomposition. We will show that the normalized branch decomposition provides the proper framework to allow us to associate a collection $\mathcal{J}_\varepsilon(f)$ to all $\varepsilon > 0$.

4.1.1 Merge trees

The merge tree of a real-valued function, f, captures the connectivity of the sublevel sets, $f^{-1}(-\infty, h]$, for each $h \in \mathbb{R}$, similar to how the Reeb graph (Edelsbrunner and Harer 2010) captures the connectivity of the level sets of a function. Here, we recall the definition of the merge tree associated to a function, and refer the reader to Edelsbrunner and Harer (2010) and Morozov et al. (2013) for more details.

Definition 8 The merge tree of f, denoted T_f, is defined to be the quotient space

$$T_f := \Gamma(f)/\sim,$$

where $\Gamma(f)$ denotes the graph of f, and for $x, y \in \Gamma(f)$, we declare $x \sim y$ if there exists an $h \in \mathbb{R}$ such that both x and y belong to the same level set of f, $f^{-1}(h)$, and also to the same connected component of the sublevel set, $f^{-1}(-\infty, h]$.

See Fig. 12b for an example of a merge tree. To visualize the construction of the merge tree, imagine a horizontal line sweeping upward from the bottom of the time series depicted in Fig. 14a. An intersection of such a line with a local minimum corresponds to a leaf of a merge tree, where the leaves are located at the bottom of the merge tree, and an intersection with a local maximum corresponds to an internal node of the merge tree. Notice that each node of T_f, whether maximum or minimum, is associated to a time at which the extremum is located, t_i, and a height of the extremum, $f(t_i)$. Denote by $\{\ell_i\}$ the leaves and by $\{m_i\}$ the internal nodes of the merge tree. For ℓ_i a leaf, denote its height by $a_i = f(t_i)$, and similarly for m_i an internal node, $b_i = f(t_i)$.

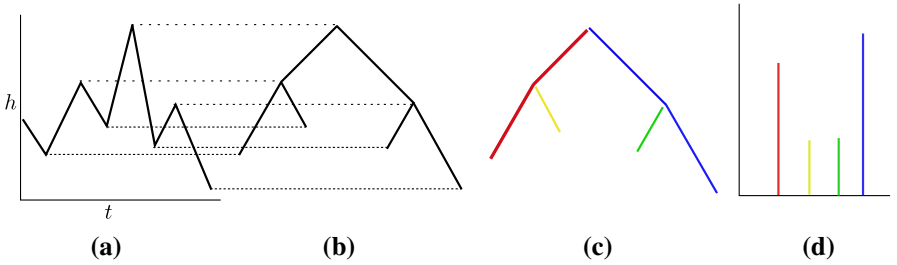

Fig. 12 **a** The graph of a function with height h versus time t, **b** its corresponding merge tree, **c** the branch decomposition represented as a colored graph, and **d** the normalized branch decomposition

Note that while the left hand endpoint in Fig. 1a is a local maximum, it does not play an important role in the merge tree, Fig. 1b, because no new topological features appear or merge with other features at that height. The endpoints are the only extrema that might not play a significant role in the merge tree.

4.1.2 Normalized branch decomposition

A branch decomposition is a way of partitioning a merge tree that pairs up the appearance (i.e. birth) and the disappearance (i.e. death) of minima as a function of sublevel height h (Morozov and Weber 2013). The birth heights are given by values $\{f(t_i)\}$ associated to leaves ℓ_i and the death heights $\{f(t_k)\}$ correspond to the internal nodes of the merge tree. At each internal node, at least two branches merge. We choose to continue the branch with lowest birth height and terminate all other branches.

This is unambiguous when the branches start at distinct heights. Although having distinct minima is a generic property in continuous functions, experimental time series may be measured only up to some finite resolution, in which case equal height values at which different minima appear may be common. Therefore, in the case that the branches do not start at distinct heights, we arbitrarily decide to continue the branch with the lowest height that also occurs first in time. Other choices are reasonable and would induce different branch decompositions.

Definition 9 We define a total order $\prec$ on the leaves of the merge tree $\{\ell_i\}$ by saying $\ell_i \prec \ell_j$ if one of the following holds:

1. $a_i < a_j$, or
2. $a_i = a_j$ and $t_i < t_j$.

This defines an indexing on the leaves of T_f that satisfies $\ell_i \prec \ell_{i+1}$ for all i, starting at $i = 0$.

Definition 10 Let m_k be an internal node of T_f, with associated height b_k. Define S_{m_k} to be the subtree of T_f that is rooted at m_k. We define the branch decomposition of f to be the collection of intervals

(a) $[a_0, b_0]$, where b_0 is the global maximum of f; and
(b) $[a_i, b_k)$ whenever S_{m_k} is the largest subtree satisfying

(1) ℓ_i is in the subtree S_{m_k} and
(2) $\ell_i \prec \ell_j$ for all $\ell_j \in S_{m_k}$ with $i \neq j$.

The branch decomposition can be viewed as a partition of the merge tree, as shown in Fig. 12c.

Definition 11 The normalized branch decomposition of $f : [x_1, x_2] \to \mathbb{R}$ is a collection of intervals

$$\mathcal{B}(f) := \bigsqcup_i J_i,$$

where $\bigsqcup$ denotes disjoint union, $J_0 := [0, b_0 - a_0]$, and $J_i := [0, b_k - a_i)$ for $i > 0$ are defined using the branch decomposition of f. Recall that each J_i is uniquely associated to a leaf ℓ_i, with value $a_i = f(t_i)$, and its time of occurrence t_i. We say that t_i is the representative of J_i.

Note that by representing the collection (J_i) as disjoint intervals, we can visualize the normalized branch decomposition as a "barcode"-like summary, which we show in Fig. 12d.

4.1.3 ϵ-Minimal intervals

We now establish properties of the normalized branch decomposition that allow us to associate a collection of intervals $\mathcal{J}_\varepsilon(f)$ to each parameter ε. First, we describe how to obtain the intervals corresponding to local minima, and then we dualize this procedure to obtain intervals corresponding to the local maxima.

Definition 12 Fix $\varepsilon > 0$. Let $B_\varepsilon \subset \mathcal{B}(f)$ be the collection of all intervals in the normalized branch decomposition $\mathcal{B}(f)$ that are longer than 2ε,

$$B_\varepsilon = \{J_i \in \mathcal{B}(f) \ : \ |J_i| > 2\varepsilon\}.$$

Let $J_i = [0, b_k - a_i) \in B_\varepsilon$ and consider its representative t_i. Define $\varphi(J_i)$ to be the connected component of $(f - \varepsilon)^{-1}(-\infty, a_i + \varepsilon)$ that contains t_i. Clearly, $\varphi(J_i)$ is a well-defined relatively open interval in $[x_1, x_2]$. We define the collection of ε-minimal intervals, denoted

$$\mathcal{J}_\varepsilon^{\min}(f) := \{\varphi(J_i) \mid J_i \in B_\varepsilon\}$$

to be the collection of all such intervals. For $I_i = \varphi(J_i)$, we say that t_i is the representative of I_i as well as of J_i.

See Fig. 13 for a depiction of the action of φ. The following Proposition shows that we cannot have overlapping ε-minimal intervals.

Proposition 1 *For any $I_i, I_j \in \mathcal{J}_\varepsilon^{\min}(f)$, $I_i \cap I_j = \emptyset$.*

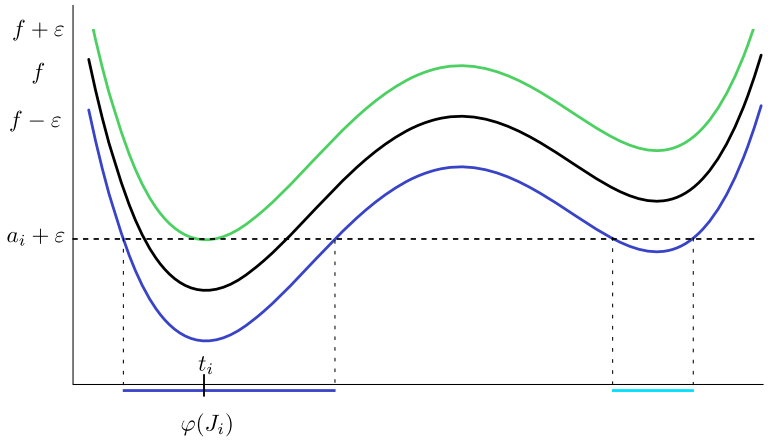

Fig. 13 A graph of a function, f, as well as $f \pm \varepsilon$. While $(f-\varepsilon)^{-1}(-\infty, a_i+\varepsilon)$ consists of both the dark blue and light blue intervals, $\varphi(J_i)$ is just the connected component that contains the representative, t_i (color figure online)

Proof Consider $I_i = \varphi(J_i)$ for some $J_i = [0, b_k - a_i) \in B_\varepsilon$, and $I_j = \varphi(J_j)$ for some $J_j = [0, b_\ell - a_j) \in B_\varepsilon$, their representatives t_i and t_j, and their leaves in the merge tree ℓ_i and ℓ_j, respectively. Since t_i and t_j are locations of minima of the continuous function f, there must exist at least one maximum of f between t_i and t_j. Let b_q denote the highest local maximum between the two minima, with location $t_q \in (t_i, t_j)$. Notice that if $b_q < b_k$ or $b_q < b_\ell$, then the two leaves ℓ_i and ℓ_j are in the same subtree rooted at one of the internal nodes m_k or m_ℓ associated to b_k and b_ℓ respectively. This contradicts the fact that I_i, I_j correspond to distinct branches in the branch decomposition. So $b_q \geq b_k$ and $b_q \geq b_\ell$. Then by Definition 12,

$$b_q - a_i \geq b_k - a_i > 2\varepsilon \text{ and } b_q - a_j \geq b_\ell - a_j > 2\varepsilon,$$

so that $t_q \notin (f-\varepsilon)^{-1}(-\infty, a_i+\varepsilon) = I_i$ and $t_q \notin (f-\varepsilon)^{-1}(-\infty, a_j+\varepsilon) = I_j$. Since $t_q \in (t_i, t_j)$ with $t_i \in I_i$ and $t_j \in I_j$, this establishes that $I_i \cap I_j = \emptyset$. □

Assume $f : [x_1, x_2] \to \mathbb{R}$ is a continuous function and assume $\mathcal{B}(f)$ is its corresponding normalized branch decomposition.

Proposition 2 *Fix $\varepsilon > 0$. Then any $g \in N_\varepsilon(f)$ attains a local minimum in the relative interior of every interval $I \in \mathcal{J}_\varepsilon^{min}(f)$.*

Proof Note that any interval $I \in \mathcal{J}_\varepsilon^{\min}(f)$ has a corresponding interval $J_i \in B_\varepsilon \subset \mathcal{B}(f)$ given by $I = \varphi(J_i)$. By construction of $\mathcal{B}(f)$, f attains a local minimum a_i in I, since $a_i = f(t_i)$ and $t_i \in I$.

We now consider several cases.

- First, assume $I = (y_1, y_2)$ away from the endpoints of $[x_1, x_2]$; i.e. $y_i \neq x_i, i = 1, 2$. Since $g \in N_\varepsilon(f)$, and $f(t_i) = a_i$, it follows that $g(t_i) < a_i + \varepsilon$. But by definition of I we have $a_i + \varepsilon = f(y_1) - \varepsilon$, because otherwise $y_1 \in I$. Therefore

$$a_i + \varepsilon = f(y_1) - \varepsilon < g(y_1),$$

and by similar argument $a_i + \varepsilon < g(y_2)$. It follows that

$$g(t_i) < g(y_1) \qquad \text{and} \qquad g(t_i) < g(y_2)$$

and by continuity g attains a local minimum in I.

- Next, assume that $I = [x_1, y)$, and $g \in N_\varepsilon(f)$. If g attains a local minimum at x_1, then the proof is complete; so assume that $g(x_1)$ is not a local minimum of g in I. As before we have that $g(y) > a_i + \varepsilon$, and $g(t_i) < a_i + \varepsilon$. If $g(x_1) \geq a_i + \varepsilon$, then by continuity, g must attain a local minimum in the interior of I. Finally, assume that $g(x_1) < a_i + \varepsilon$. Since g does not attain a local minimum at x_1, there exists some $\widetilde{x} \in I$ such that $g(\widetilde{x}) < g(x_1) < g(y)$. By continuity, g must attain a minimum in the interior of I.
- The case where $I = (y, x_2]$ follows from a similar argument to that of the previous case. □

Let $\bar{I}$ denote the closure of interval I.

Proposition 3 *Consider $I \in \mathcal{J}_\varepsilon^{min}(f)$. For any $\overline{J} \subsetneq \bar{I}$, there exists a strictly monotone function $g : \overline{J} \to \mathbb{R}$ such that $g \in N_\varepsilon(f|_{\overline{J}})$.*

Proof Let $\bar{I} = [y_1, y_2]$ and $\overline{J} := [z_1, z_2]$. We prove the case $z_1 \neq y_1$, with an analogous argument proving the case $z_2 \neq y_2$. Recall that f attains a minimum in I, $a_i = f(t_i)$. First assume that the minimum is located in the interior of the subinterval, $t_i \in (z_1, z_2)$. Note that $f(z_2) < a_i + 2\varepsilon$, and for all $z \in (t_i, z_2)$, we have $f(z) \geq a_i$. Therefore there exists $\delta_2 > 0$ such that

$$a_i + \varepsilon + \delta_2 \in N_\varepsilon(f(z_2)) = (f(z_2) - \varepsilon,\, f(z_2) + \varepsilon).$$

Since $z_1 \neq y_1$, there exists some $\delta_0 > 0$, such that

$$a_i + \varepsilon - \delta_0 \in N_\varepsilon(f(z_1)).$$

Finally, there is $\delta_1 < \delta_0$ such that

$$a_i + \varepsilon - \delta_1 \in N_\varepsilon(f(t_i)).$$

We define two linear increasing functions $g_1 : [z_1, t_i] \to \mathbb{R}$ and $g_2 : [t_i, z_2] \to \mathbb{R}$ by

$$\begin{aligned} g_1(z_1) &= a_i + \varepsilon - \delta_0, \quad g_1(t_i) = a_i + \varepsilon - \delta_1; \quad \text{and} \\ g_2(t_i) &= a_i + \varepsilon - \delta_1, \quad g_2(z_2) = a_i + \varepsilon + \delta_2. \end{aligned}$$

Function g_1 is increasing; since $\delta_1 < \delta_0$; g_2 is also clearly increasing. Most importantly, we have that the graph of g_1 is contained in $N_\varepsilon(f|_{[z_1,t_i]})$ and the graph of g_2 is contained in $N_\varepsilon(f|_{[t_i,z_2]})$. Since $g_1(t_i) = g_2(t_i)$ the continuous function, $g : \overline{J} \to \mathbb{R}$, defined by

$$g(x) = \begin{cases} g_1(x), & \text{if} \quad x \in [z_1, t_i] \\ g_2(x) & \text{if} \quad x \in [t_i, z_2] \end{cases}$$

is in $N_\varepsilon(f_{\bar{J}})$. By construction, g is a piecewise linear, strictly increasing function. This establishes the result for the case $t_i \in (z_1, z_2)$. To finish the proof, we note that if $t_i \leq z_1$, the construction of g_2 produces the desired function, and if $z_2 \leq t_i$, then the construction of g_1 produces the desired function. □

Corollary 1 *For any non-empty, proper subinterval, $J \subset I$ with $I \in \mathcal{J}_\varepsilon^{min}(f)$, such that $\overline{J} \subsetneq \overline{I}$, there exists some $g \in N_\varepsilon(f)$ such that g does not attain a local minimum in J.*

We conclude that $\mathcal{J}_\varepsilon^{\min}(f)$ consists of intervals on which every ε-perturbation of f attains a local minimum, and furthermore, that these intervals are the smallest such intervals for which this is true. This justifies the name and notation of $\mathcal{J}_\varepsilon^{\min}(f)$. This collection robustly represents the minima of f up to precision ε.

4.1.4 Dual construction for local maxima

The simple observation that the maxima of f are the minima of $-f$ leads to the following definition.

Definition 13 We define a collection of ε-maximal intervals $\mathcal{J}_\varepsilon^{\max}(f)$ by

$$\mathcal{J}_\varepsilon^{\max}(f) := \mathcal{J}_\varepsilon^{\min}(-f).$$

Given the above definition, we have the following corollary from Proposition 2 and Corollary 1.

Corollary 2 *Fix $\varepsilon > 0$. Then any $g \in N_\varepsilon(f)$ attains a local maximum in every interval $I \in \mathcal{J}_\varepsilon^{max}(f)$. Furthermore, for any nonempty, proper subinterval $J \subset I$ with $\overline{J} \subsetneq \overline{I}$, there exists some $h \in N_\varepsilon(f)$ such that h does not attain a local maximum in I.*

Therefore the collection $\mathcal{J}_\varepsilon^{\max}(f)$ robustly represents the maxima of f, which is a natural dual of $\mathcal{J}_\varepsilon^{\min}(f)$. The corresponding dual merge tree, dual branch decomposition, and dual normalized branch decomposition of the function f in Fig. 12 are shown in Fig. 14. To visualize the construction of the merge tree, imagine a horizontal line sweeping down from the top of the time series depicted in Fig. 14a. An intersection of such a line with a local maximum corresponds to a leaf of a merge tree, where

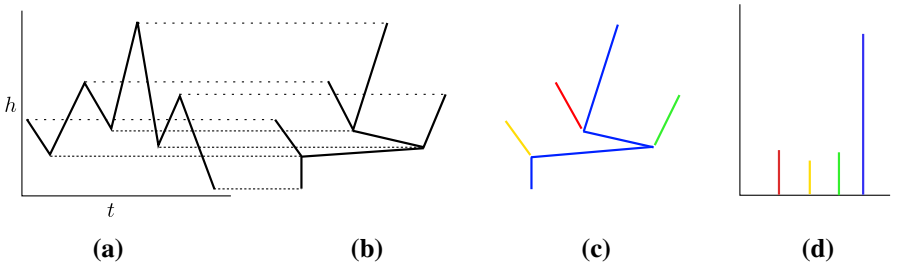

Fig. 14 From left to right: the graph of a function, its corresponding dual merge tree, the dual branch decomposition, and the normalized dual branch decomposition

the leaves are located at the top of the merge tree, and an intersection with a local minimum corresponds to an internal node of the merge tree.

Definition 14 The collection of ε-extremal intervals is the collection

$$\mathcal{J}_\varepsilon(f) := \mathcal{J}_\varepsilon^{\min}(f) \sqcup \mathcal{J}_\varepsilon^{\max}(f).$$

If $\mathcal{J}_\varepsilon(f) = \emptyset$, then we say f is ε-constant.

The motivation for the definition of ε-constant is that when $\mathcal{J}_\varepsilon(f) = \emptyset$, then $b - a \leq 2\varepsilon$ for any minimum a and maximum b, and so the extrema cannot be distinguished at ε. We now show that the ε-minimal intervals are distinct from ε-maximal intervals.

Proposition 4 *Consider $I \in \mathcal{J}_\varepsilon^{min}(f)$, represented by u and $J \in \mathcal{J}_\varepsilon^{max}(f)$, represented by v. Then $u \notin J$ and $v \notin I$.*

Proof By the definition of representative, $f(u) = a$ is a local minimum with $u \in I$ and $f(v) = b$ is a local maximum with $v \in J$. Then $v \notin I$ since otherwise

$$\begin{aligned} & v \in (f - \varepsilon)^{-1}(-\infty, a + \varepsilon) \\ \Rightarrow\ & b = f(v) \leq a + 2\varepsilon \\ \Rightarrow\ & b - a \leq 2\varepsilon \\ \Rightarrow\ & I, J \notin \mathcal{J}_\varepsilon(f). \end{aligned}$$

A similar argument shows $u \notin J$. □

Corollary 3 $\mathcal{J}_\varepsilon^{min}(f) \cap \mathcal{J}_\varepsilon^{max}(f) = \emptyset$.

4.1.5 Total ordering in $\mathcal{J}_\varepsilon(f)$

Since $f(x)$ is continuous, we expect that the ε-minimal and ε-maximal intervals must alternate. In this section, we define a total order on these intervals and prove that it is well-defined and that the extremal intervals alternate as expected. Technicalities occur, because the ε-minimal intervals can overlap with the ε-maximal intervals; see Fig. 15.

Definition 15 We define an order $\lhd$ on $\mathcal{J}_\varepsilon(f)$ as follows. Consider two relatively open intervals $I, J \in \mathcal{J}_\varepsilon(f)$, with $y_1 < y_2$ the endpoints of I and $z_1 < z_2$ the endpoints of J. Then $I \lhd J$ if and only if either $y_1 < z_1$ or $y_2 < z_2$.

Theorem 1 *The order $\lhd$ on $\mathcal{J}_\varepsilon(f)$ is a well-defined total order.*

Proof Consider two relatively open intervals $I, J \in \mathcal{J}_\varepsilon(f)$, with $y_1 < y_2$ the endpoints of I and $z_1 < z_2$ the endpoints of J. Let u represent I and let v represent J. To prove that $\lhd$ is well-defined, we must show that $y_1 \leq z_1$ iff $y_2 \leq z_2$.

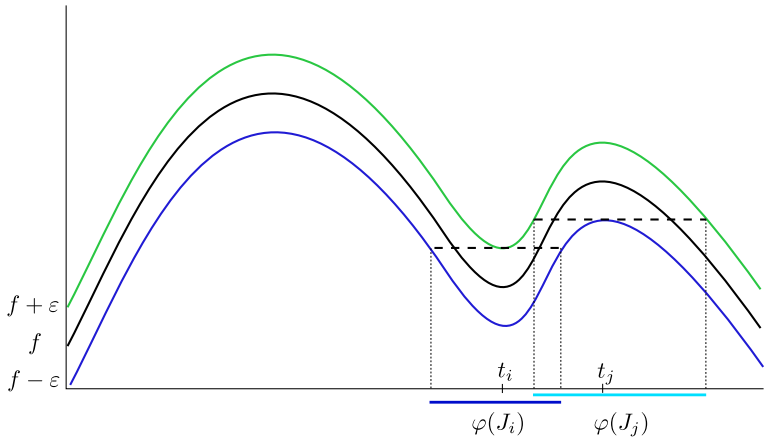

Fig. 15 A graph of a function, f, as well as $f \pm \varepsilon$. Here, t_i is a representative of a minimum with associated ε-minimum interval $\varphi(J_i)$, and t_j is a representative of the ε-maximum interval $\varphi(J_j)$. Notice that $\varphi(J_i) \cap \varphi(J_j) \neq \emptyset$

The case where $I \cap J = \emptyset$ is trivial, so we consider $I \cap J \neq \emptyset$. It follows from Proposition 1 that when $I \cap J \neq \emptyset$, then either $I \in \mathcal{J}_\varepsilon^{\min}(f)$ and $J \in \mathcal{J}_\varepsilon^{\max}(f)$ or vice versa. Now it follows from Proposition 4 that

$$u < v \Rightarrow y_1 \leq u \leq z_1 \text{ and } y_2 \leq v \leq z_2$$

and similarly

$$v < u \Rightarrow z_1 \leq v \leq y_1 \text{ and } z_2 \leq u \leq y_2.$$

This shows that the order is well defined. It now follows from Corollary 3 that the order is total. □

Theorem 2 *ε-minimal intervals alternate with ε-maximal intervals.*

Proof Consider two adjacent ε-minimal intervals $I_1 \triangleleft I_2$ with a_1 and a_2 the associated local minima. Assume without loss that $a_1 > a_2$. Then there exists $t \in (a_1, a_2)$ such that $t \notin (f - \varepsilon)^{-1}(-\infty, a_1 + \varepsilon)$, since otherwise I_1 and I_2 are not distinct. This implies that $f(t) > a_1 + 2\varepsilon$ and therefore $b := \max_{x \in [a_1, a_2]} f(x) > a_1 + \varepsilon > a_2 + \varepsilon$. Thus there is an ε-maximal interval J containing b. Since $\triangleleft$ is a total order this implies $I \triangleleft J \triangleleft K$.

The argument ruling out two adjacent maxima is similar. □

4.1.6 Partial ordering between $\mathcal{J}_\varepsilon(f_i)$

So far we have considered the representation of a single function $f(t)$ in terms of a noise-level dependent collection of ε-minimal and ε-maximal intervals. However, many datasets include M functions f_i on the same domain $[x_1, x_2]$. There is a natural partial order on the union of the ε-extremal intervals.

Definition 16 Define the partially ordered set

$$\left(\bigsqcup_{i}^{M} \mathcal{J}_\varepsilon(f_i), \lhd \right)$$

to be an extension of the total order of each $\mathcal{J}_\varepsilon(f_i)$ as follows. If $I, J \in \mathcal{J}_\varepsilon(f_i)$ for some i, define $\lhd$ as in Definition 15. Now consider $I \in \mathcal{J}_\varepsilon(f_i)$ and $J \in \mathcal{J}_\varepsilon(f_j)$ for $i \neq j$, with $y_1 < y_2$ the endpoints of I and $z_1 < z_2$ the endpoints of J. Then

$$I \lhd J \quad \text{if and only if } y_2 \leq z_1.$$

We choose this order because overlapping ε-extremal intervals *across* functions indicate that the order of the respective local extrema is not decidable at the noise level of ε.

4.2 ε-Extremal intervals for discrete time series

Time series have values measured at only a finite collection of times. In this section, we consider this case in more detail.

Definition 17 A set $\mathcal{D} := \{D_i\}_{i=1}^{M}$ is a dataset on the interval $[x_1, x_2]$ if $D_i := \{(z_k, h_k^i)\}_{k=1}^{N}$ where

$$Z := \{z_1 = x_1, z_2, \ldots, z_{N-1}, z_N = x_2\},$$

is an ordered set with $z_j < z_{j+1}$ and the heights h_k^i are measurements of the i^{th} variable at z_k. The components D_i will be referred to as time series.

The collection Z is independent of i. An example would be a collection of time series of gene expression, such as comes from RNAseq data. We note, however, that the independent variable is not required to be time.

Definition 18 Let f_i be the linear interpolation of D_i, and let $\mathcal{J}_\varepsilon(f_i)$ be the ε-extremal intervals of the linear interpolation. Define $\mathcal{K}_\varepsilon^{\min}(D_i)$ to be the set of relatively open intervals in $[x_1, x_2]$ with endpoints in the set Z such that for each $I_i \in \mathcal{J}_\varepsilon^{\min}(f_i)$, there exists $J_i \in \mathcal{K}_\varepsilon^{\min}(D_i)$ satisfying

(1) $J_i \supseteq I_i$, and
(2) J_i is the minimal such interval; i.e. there does not exist an interval K_i with endpoints in Z such that $J_i \supsetneq K_i \supseteq I_i$.

We define a function $\beta^{\min} : \mathcal{K}_\varepsilon^{\min}(D_i) \to \mathcal{J}_\varepsilon^{\min}(f_i)$ by $\beta^{\min}(J_i) = I_i$ that captures this relationship. Analogously, considering the linear interpolation $-f_i$, there is an analogous set of intervals $\mathcal{K}_\varepsilon^{\max}(D_i)$ and a map $\beta^{\max}$.

Since every local extremum of the linear interpolation f_i occurs at one of the points $z_j \in Z$, it is easy to see that the proof of Proposition 1 is still valid. Therefore we have

Proposition 5 *For any* $I, J \in \mathcal{K}_\varepsilon^{\min}(D_i)$, $I \cap J = \emptyset$.

Note that choosing f_i to be the linear interpolation of D_i is critical in order for this proposition to hold.

Definition 18 is a conservative definition in the sense that a minimum is guaranteed to occur within each interval I of $\mathcal{K}_\varepsilon^{\min}(D_i)$ by restricting to $\beta^{\min}(I) \in \mathcal{J}_\varepsilon^{\min}(f_i)$. In other words, Proposition 2 still holds for $\mathcal{K}_\varepsilon^{\min}(D_i)$. However, minimality is lost in discretization and Proposition 3 does not hold for $\mathcal{K}_\varepsilon^{\min}(D_i)$.

This widening of the ε-extremal intervals means that an ε-minimal interval and an ε-maximal interval can coincide. In other words, given $I \in \mathcal{K}_\varepsilon^{\min}(D_i)$ and $J \in \mathcal{K}_\varepsilon^{\max}(D_i)$, we can have that $I = J$, so that Proposition 3 does not hold. To address this issue, we remove these intervals from the sets $\mathcal{K}_\varepsilon^{\min}(D_i)$ and $\mathcal{K}_\varepsilon^{\max}(D_i)$.

Definition 19 Let $\mathcal{K}_\varepsilon^{\cap}(D_i)$ be the intersection $\mathcal{K}_\varepsilon^{\min}(D_i) \cap \mathcal{K}_\varepsilon^{\max}(D_i)$. Then define

$$\begin{aligned}\mathcal{I}_\varepsilon^{\min}(D_i) &:= \mathcal{K}_\varepsilon^{\min}(D_i) \setminus \mathcal{K}_\varepsilon^{\cap}(D_i)\\ \mathcal{I}_\varepsilon^{\max}(D_i) &:= \mathcal{K}_\varepsilon^{\max}(D_i) \setminus \mathcal{K}_\varepsilon^{\cap}(D_i)\\ \mathcal{I}_\varepsilon(D_i) &:= \mathcal{I}_\varepsilon^{min}(D_i) \sqcup \mathcal{I}_\varepsilon^{\max}(D_i).\end{aligned}$$

We say that a time series D_i is ε-constant if and only if

$$\mathcal{I}_\varepsilon(D_i) = \emptyset.$$

In a slight abuse of nomenclature, we will refer to $\mathcal{I}_\varepsilon(D_i)$ as the collection of ε-extremal intervals of D_i, and $\mathcal{I}_\varepsilon^{\max}(D_i)$ and $\mathcal{I}_\varepsilon^{\min}(D_i)$ will be called the ε-maximal and ε-minimal intervals of D_i, respectively. We say that

$$\mathcal{I}_\varepsilon(\mathcal{D}) := \bigsqcup_i^M \mathcal{I}_\varepsilon(D_i)$$

is the set of ε-extremal intervals of the dataset.

Since $\mathcal{I}_\varepsilon(\mathcal{D}) \subseteq \mathcal{K}_\varepsilon(\mathcal{D})$, all the results for $\mathcal{K}_\varepsilon(\mathcal{D})$ hold on $\mathcal{I}_\varepsilon(\mathcal{D})$.

The total order described in Definition 15, now applied to $\mathcal{I}_\varepsilon(D_i)$, and the associated Theorem 1 proving the total order is well-defined, hold without changes—provided we again use the fact that all extrema of f_i occur at some z_j in the discretized time interval. Using that observation, we remark that the ε-minimal and ε-maximal intervals of $\mathcal{I}_\varepsilon(D_i)$ still alternate, as in Theorem 2. We are now free to apply Definition 16 for the partial order $\lhd$ on $\mathcal{I}_\varepsilon(\mathcal{D})$, as restated here.

Definition 20 Define a partial order $\lhd$ on the set $\mathcal{I}_\varepsilon(\mathcal{D})$ as follows: Let $I \in \mathcal{I}_\varepsilon(D_i)$ and $J \in \mathcal{I}_\varepsilon(D_j)$ with $y_1 < y_2$ the endpoints of I and $z_1 < z_2$ the endpoints of J. If $i = j$, then $I \lhd J$ if and only if either $y_1 < z_1$ or $y_2 < z_2$. If $i \neq j$, then $I \lhd J$ if and only if $y_2 \leq z_1$.

4.3 Algorithms and software

4.3.1 Merge trees and branch decompositions

To calculate merge trees and branch decompositions for a discrete time series $D_i = \{z_j, h^i_j\}$, we follow Smirnov and Morozov (2017), who provide pseudocode for Kruskals algorithm and a helper function called FindDeepest. This combination of algorithms has $O(m \log n)$ complexity for a merge tree with n nodes and m edges. We briefly summarize these algorithms here.

Let Z be the ordered set of time points $\{z_1, \ldots, z_N\}$ and let f_i be the linear interpolation as before, with $f_i(z_j) = h^i_j$. We form a linear graph G, where the vertices of G are labeled by the elements of Z, and edges in G connect and z_j and z_{j+1} for all i. For brevity, we will drop subscripts where the context allows, and we will refer to a vertex in G as an element $z \in Z$.

Definition 21 (Smirnov and Morozov 2017) For $h \in \mathbb{R}$, the sublevel graph at h, denoted G_h, is the subgraph induced by the vertices $Z' \subseteq Z$ whose function values $f_i(z)$ for $z \in Z'$ do not exceed h. The representative of vertex z at level $h \geq f_i(z)$ is the vertex $y \in Z$ with the minimum function value in the connected component of G_h containing z.

Definition 22 (Smirnov and Morozov 2017) The merge tree of f_i on G is the tree on the vertex set of G that has an edge (z, y), with $f_i(z) < f_i(y)$, if the connected component of z in $G_{f_i(z)}$ is a subset of the connected component of y in $G_{f_i(y)}$, and there is no vertex x with $f_i(z) < f_i(x) < f_i(y)$ such that the connected component of z is a subset of the connected component of x in $G_{f_i(x)}$.

The algorithm of Smirnov and Morozov (2017) is based on a representation of each vertex z in the merge tree by a triplet of vertices (z, s, y), where vertex z represents itself at levels $h \in [f_i(z), f_i(s))$, and y becomes its representative at level $f_i(s)$. We make the following observations:

- if z is a vertex representing the global minimum of f_i, then the triplet attached to z will be (z, z, z);
- if z is any other local minimum of f_i and hence a leaf of the merge tree, then the triplet associated to z is (z, s, y) with $z \neq s \neq y$ and $f_i(y) < f_i(z) < f_i(s)$;
- if z is any other non-leaf vertex of G, then its triplet representation is (z, z, y) with $f_i(y) < f_i(z)$.

Note that for any leaf z of the merge tree with triplet (z, s, y), the interval $[f_i(z), f_i(s))$ represents a branch in the branch decomposition. For z with a triplet (z, z, z), the branch is $[f_i(z), b_0]$, where b_0 is the global maximum of f_i. The normalized branch decomposition is then the collection $\{[0, f_i(s) - f_i(z))\} \cup \{[0, b_0 - f_i(z)]\}$ for every triplet in the branch decomposition. Therefore the algorithms of Smirnov and Morozov (2017) calculate both the merge tree and the normalized branch decomposition simultaneously.

We implemented Algorithms 1 and 2 of Smirnov and Morozov (2017) in Python 3 (Cummins and Nerem 2019), along with a post-processing function to isolate the

leaves of the merge tree using the fact that non-leaf vertices always have the triplet form (z, z, y) with $f_i(y) < f_i(z)$. When there are two minima of identical depth, the one that occurs first in time is chosen to represent the triplet as in Definition 9.

4.3.2 ϵ-Extremal intervals

Once the leaves are isolated, we calculate, for some specified noise level ε, the ε-minimal interval associated to each minimum. First, we remove all normalized branches that are not greater than 2ε. Then we take each representative z in the remaining triplets (z, s, y), and grow a ball around z until the associated function values $f_i(z - \delta_1)$ and $f_i(z + \delta_2)$ meet or exceed a distance of 2ε from $f_i(z)$. This constraint arises because we are constructing the connected component of the sublevel set $(f_i - \varepsilon)^{-1}(-\infty, f_i(z) + \varepsilon)$ that contains z. So we are seeking the largest set of vertices $Z' \subseteq Z$ such that the subgraph of G induced by Z' is connected, and any $v \in Z'$ satisfies $f_i(v) - \varepsilon \in (-\infty, f_i(z) + \varepsilon)$, or equivalently, $f_i(v) - \varepsilon < f_i(z) + \varepsilon$.

In order to find the dual merge tree, dual branch decomposition, and associated ε-maximal intervals, we simply reflect the curve f_i over the z-axis to get $-f_i$, and repeat exactly the same procedure. The calculation of ε-extremal intervals given the triplets is at worst linear in the number of time points in the time series.

Once we have all of the ε-extremal intervals for a dataset, $\mathcal{I}_\varepsilon(\mathcal{D})$, we impose the partial order in Definition 20. All of this functionality is in the open source software (Cummins and Nerem 2019), along with Jupyter notebooks that generate the figures for the applications in Sect. 3.

4.3.3 Graph distance

In Application 2, we use a graph distance to calculate similarity between the transitive closures of Hasse diagrams. This graph distance gives roughly the proportion of dissimilar edges between $H_\varepsilon^C(\mathcal{D}_1)$ and $H_\varepsilon^C(\mathcal{D}_2)$.

Definition 23 For two node-labeled graphs $G(V, E, \ell)$ and $G'(V', E', \ell')$, a bijection $\phi : V \to V'$ is a graph isomorphism if and only if

- $(v_1, v_2) \in E \iff (\phi(v_1), \phi(v_2)) \in E'$ for all $v_1, v_2 \in V$,
- $\ell(v) = \ell'(\phi(v))$ for all $v \in V$.

Definition 24 For two node-labeled graphs $G(V, E, \ell)$ and $G'(V', E', \ell')$ the directed maximum common edge induced subgraph (DMECS) problem is to find some ordered pair (W, W') with $W \subseteq E$ and $W' \subseteq E'$ such that if

$$U = \{v_1 \in V \mid (v_1, v_2) \in W \text{ or } (v_2, v_1) \in W\}$$
$$U' = \{v_1 \in V' \mid (v_1, v_2) \in W' \text{ or } (v_2, v_1) \in W'\}$$

then the graphs $H = (U, W, \ell|_U)$ and $H' = (U', W', \ell'|_{U'})$ are isomorphic and the value of $|W| = |W'|$ is maximized. Here H and H' are edge-induced subgraphs of G

and G'. $|W|$ is maximized if for all $Z \subseteq E$ and $Z' \subseteq E'$ such that Z and Z' induce isomorphic subgraphs of G and G',

$$|Z| \leq |W|.$$

We define $\mathrm{DMCES}(G, G') = |W|$ where $|W|$ is maximized.

It is shown in Nerem et al. (2019) that

$$d(G, G') = 1 - \frac{\mathrm{DMCES}(G, G')}{\max(|E|, |E'|)}.$$

is a metric on the space of node-labeled, directed graphs.

Since by definition

$$DMCES(G, G') \leq \max(|E|, |E'|),$$

distance varies between 0 (most similar) and 1 (least similar).

As shown in Nerem et al. (2019), there is a polynomial-time reduction of the DMCES problem to the maximum clique problem and polynomial-time reduction of the graph isomorphism problem to the DMCES problem. Thus DMCES is no easier than the graph isomorphism problem and no harder than the maximum clique problem. This suggests that computing the DMCES problem is exponentially hard but no harder than NP-complete problems. We use an algorithm from Nerem et al. (2019) to compute the DMCES problem which leverages the special structure of the Hasse diagrams produced from datasets; namely that the partial order is built from the collection of total orders that arise from each individual time series.

5 Discussion

The method described in this paper assigns to a time series a collection of partially ordered intervals that are dependent upon a level of measurement uncertainty ε. Each interval is guaranteed to contain either a maximum or a minimum of every continuous function that is ε-close to the time series. We are particularly focused on applications in molecular and cellular biology where 'omics data can measure expression levels of thousands of genes; however, this approach is widely applicable.

Due to experimental challenges, a typical time series has 10–20 time points with time resolution of minutes to hours, and significant levels of measurement error. We do not assume that the measured variables are statistically independent; in fact this dependence carries information about the interactions between the components of the system which are of great interest. There are methods that use time series to deduce a structure of the underlying causal network (Albert 2007; Sugihara et al. 2016; Cummins et al. 2015; McGoff et al. 2016), i.e. which genes up-regulate or down-regulate other genes. These methods rely on sequencing of time points when genes achieve their peak expression, lowest expression, or time points when they pass

the half saturation point. Since this is the time where maximal rate of change occurs, we may alternatively approximate the derivative of the time series and turn the problem of finding the sequence of times with maximal rates of change to the problem of finding extrema.

Our work is closely related to work on merge trees and persistence homology (Edelsbrunner and Harer 2010; Carlsson 2009; Zomorodian and Carlsson 2005). Persistence type methods use topological data analysis to extract the most prominent extrema. Since the most persistent features have large amplitude, and we are interested in all levels where extrema appear and disappear, the 0-persistence of f gives closely related, but complementary results to ours.

Similar ideas to those presented in this paper have been used in Günther et al. (2014) in the context of 2D uncertain scalar fields. Their goal is to describe mandatory critical points for 2D data for which upper and lower bounding scalar fields f^- and f^+ are given. The mandatory critical points are characterized by regions of the plane called critical components where the critical point is guaranteed to occur for any realization of a scalar field within f^- and f^+, and by critical intervals in $\mathbb{R}$ which bound the admissible height of the critical point. These critical components correspond in our approach to ε-minimal intervals when $f^+ = f + \varepsilon$ and $f^- = f - \varepsilon$.

Our approach can be viewed as an adaptation of the technique in Günther et al. (2014) to time series data, with some important differences. Since the time series data are assumed to arise from a continuous process, we first analyze continuous functions and only then extend it to the discrete time series data. Moreover, we do not assume that the upper (f^+) and lower (f^-) bounds are given; rather we parameterize these in terms of parameter ε and analyze a range of values of ε. This aspect is very important in the applications we present in Sect. 3. In these applications, we analyze multiple time series, construct partial orders of ε-extremal intervals, and use the partial orders to compare models to time series as well as quantify differences between time series replicates.

The two applications of our technique use microarry data. In the first application, the presented analysis allows the rejection of DSGRN network models (Cummins et al. 2018, 2016) that cannot reproduce the experimentally observed sequences of minima and maxima of microarray time series. For a proposed network model, DSGRN computes all possible sequences of minima and maxima that can be produced by the network, across all parameters. This data can be then compared to a partial order computed from the experimental time series data, at different levels of assumed experimental measurement uncertainty ε. We illustrate our approach on data from the yeast cell cycle, where the regulatory network is well-described and has substantial experimental validation (Cho et al. 2019; Haase and Wittenberg 2014; Orlando et al. 2008; Pramila et al. 2006; Kovacs et al. 2012; Simon et al. 2001). We show that the network model can reproduce experimental data at a low level of assumed experimental noise, but cannot reproduce data where we artificially swap labels on the time series. Swapping labels is equivalent to making the model incorrect, and such a model is consistent with data only at very high levels of noise (28% of the total signal amplitude).

In our second application, we study again the gene regulatory network that controls the cell-cycle-transcriptional program (Bristow et al. 2014; Cho et al. 2017; Orlando et al. 2008; Kovacs et al. 2012). The network of serially-activated transcription factors

activates other transcription factors at appropriate phase of the cell cycle and thus plays a key role in establishing a cell cycle generating order in cellular transcription. There is an ongoing debate in the field on the role of this network in controlling cell-cycle-regulated transcription and in the ordering of cell-cycle events (Rahi et al. 2016; Shedden and Cooper 2002). A reproducible ordering of gene expression, which we observe in this paper, argues for precise control of the transcriptional program and provides further supporting evidence for the importance of a cell-cycle gene regulatory network.

The problem of aligning data from different experiments and evaluating the similarity of two experiments is a problem in biology that goes beyond the study of cell cycles. Circadian clock networks also control large, well-ordered programs of phase-specific gene expression (Mure et al. 2018; Zhang et al. 2014), and perturbations to those programs are likely to be found in clock-associated diseases. Ordering the expression of genes is also a fundamental mechanism for the assembly of a variety of protein complexes (Kovacs et al. 2008). Thus, the ability to accurately compare ordering in gene expression will be useful for identifying perturbations in complex formation, circadian regulation, as well as cell-cycle control. The approach presented in this paper is applicable to any experimental time series data where comparison and evaluation of similarity of ordering is desired.

Acknowledgements T. G. was partially supported by NSF Grant DMS-1361240, USDA 2015-51106-23970, DARPA Grant FA8750-17-C-0054, NIH Grant 1R01GM126555-01, and NSF TRIPODS+X Grant 1839299. B.C. was partially supported by Grants USDA 2015-51106-23970, DARPA Grant FA8750-17-C-0054, NIH 1R01GM126555-01, and NSF TRIPODS+X Grant 1839299. E. B. was partially supported by NSF Grants DMS-1508040, DMS-1664858, DMS-1557716, DMS-1812055 and DMS-1945639. R. N. was supported by Montana State University's Undergraduate Scholars Program (USP) during the Fall 2018 funding cycle. S.H. was partially supported by NIH 5 R01 GM126555-03 and DARPA FA8750-17-C-0054. L.S. was supported by NIH 5 R01 GM126555-03. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

A Yeast data analysis

Time-series transcriptomic data for one replicate of wild-type yeast *Saccharomyes cerevisiae* were previously published in Orlando et al. (2008). Microarray (Affymetrix Yeast Genome 2.0) expression data were normalized as previously described (Orlando et al. 2008), although for this study Affymetrix probe IDs were re-annotated using Affymetrix Yeast Genome 2.0 microarray annotation 35. Expression data were aligned to a common cell-cycle time line using the CLOCCS (Characterizing Loss of Cell Cycle Synchrony) (Orlando et al. 2007) population synchrony model, as previously described (Orlando et al. 2008). Briefly, the CLOCCS model allows multiple time-series experiments to be aligned to a common cell-cycle timeline, using experimentally-derived yeast budding data. The CLOCCS model converts time points in the series to life points, which indicate the progression through the cell cycle. Expression data for both replicates were interpolated to integer life points with an interval of one using a Piecewise Cubic Hermite Interpolating Polynomial (PCHIP)

spline. Life points were then trimmed so both replicate time series were of identical location and duration in the cell cycle.

References

Akutsu T, Miyano S, Kuhara S (2000) Inferring qualitative relations in genetic networks and metabolic pathways. Bioinformatics 16(8):727–734. https://doi.org/10.1093/bioinformatics/16.8.727

Albert R (2007) Network inference, analysis, and modeling in systems biology. Plant Cell 19(11):3327–3338

Albert R, Collins JJ, Glass L (2013) Introduction to focus issue: quantitative approaches to genetic networks. Chaos 23(2):025001

Barker NA, Myers CJ, Kuwahara H (2011) Learning genetic regulatory network connectivity from time series data. IEEE/ACM Trans Comput Biol Bioinform 8(1):152–165. https://doi.org/10.1109/TCBB.2009.48

Bristow SL, Leman AR, Simmons Kovacs LA, Deckard A, Harer J, Haase SB (2014) Checkpoints couple transcription network oscillator dynamics to cell-cycle progression. Genome Biol 15(9):446. https://doi.org/10.1186/s13059-014-0446-7

Brunton SL, Proctor JL, Kutz JN (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc Natl Acad Sci 113(15):3932–3937. https://doi.org/10.1073/pnas.1517384113

Bunke H, Riesen K (2011) Recent advances in graph-based pattern recognition with applications in document analysis. Pattern Recognit 44(5):1057–1067. https://doi.org/10.1016/j.patcog.2010.11.015

Carlsson G (2009) Topology and data. Bull Am Math Soc (NS) 46(2):255–308

Carré C, Mas A, Krouk G (2017) Reverse engineering highlights potential principles of large gene regulatory network design and learning. NPJ Syst Biol Appl. https://doi.org/10.1038/s41540-017-0019-y

Cho CY, Motta FC, Kelliher CM, Deckard A, Haase SB (2017) Reconciling conflicting models for global control of cell-cycle transcription. Cell Cycle. https://doi.org/10.1080/15384101.2017.1367073

Cho CY, Kelliher CM, Haase SB (2019) The cell-cycle transcriptional network generates and transmits a pulse of transcription once each cell cycle. Cell Cycle. https://doi.org/10.1080/15384101.2019.1570655

Conte D, Foggia P, Sansone C, Vento M (2004) Thirty years of graph matching in pattern recognition. Int J Pattern Recognit Artif Intell 18(03):265–298. https://doi.org/10.1142/S0218001404003228

Cummins B, Nerem R (2019) ε-minimal interval software v0.4. https://doi.org/10.5281/zenodo.3405579; https://github.com/breecummins/min_interval_posets. Accessed Sept 2019

Cummins B, Gedeon T, Spendlove K (2015) On the efficacy of state space reconstruction methods in determining causality. SIAM J Appl Dyn Syst 14(1):335–381

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

Cummins B, Gedeon T, Harker S, Mischaikow K (2018) Model rejection and parameter reduction via time series. SIAM J Appl Dyn Syst 17(2):1589–1616

Davey BA, Priestley HA (2002) Introduction to lattices and order. Cambridge University Press, Cambridge

Edelsbrunner H, Harer JL (2010) Computational topology. American Mathematical Society, Providence

Edwards R (2001) Chaos in neural and gene networks with hard switching. Differ Equ Dyn Syst 9:187–220

Fu JJ (1996) Approximate pattern matching in directed graphs. In: Hirschberg D, Myers G (eds) Combinatorial pattern matching, no. 1075 in Lecture Notes in Computer Science. Springer, Berlin, pp 373–383. https://doi.org/10.1007/3-540-61258-0_27

Glass L, Kauffman SA (1973) The logical analysis of continuous, non-linear biochemical control networks. J Theor Biol 39(1):103–29

Günther D, Salmon J, Tierny J (2014) Mandatory critical points of 2d uncertain scalar fields. In: Computer graphics forum, vol 33. Wiley Online Library, pp 31–40

Haase SB, Wittenberg C (2014) Topology and control of the cell-cycle-regulated transcriptional circuitry. Genetics 196(1):65–90. https://doi.org/10.1534/genetics.113.152595

Harker S (2018) DSGRN software. https://doi.org/10.5281/zenodo.1210003; https://github.com/shaunharker/DSGRN. Accessed June 2019

Kelliher CM, Leman AR, Sierra CS, Haase SB (2016) Investigating conservation of the cell-cycle-regulated transcriptional program in the fungal pathogen, cryptococcus neoformans. PLoS Genet 12(12):e1006453

Lähdesmäki H, Shmulevich I, Yli-Harja O (2003) On learning gene regulatory networks under the boolean network model. Mach Learn 52(1):147–167. https://doi.org/10.1023/A:1023905711304

Livi L, Rizzi A (2012) Parallel algorithms for tensor product-based inexact graph matching. In: The 2012 international joint conference on neural networks (IJCNN), pp 1–8. https://doi.org/10.1109/IJCNN.2012.6252681

Maucher M, Kracher B, Khl M, Kestler HA (2011) Inferring Boolean network structure via correlation. Bioinformatics 27(11):1529–1536. https://doi.org/10.1093/bioinformatics/btr166

McGoff K, Guo X, Deckard A, Kelliher C, Leman A, Francey L, Hogenesch J, Haase S, Harer J (2016) The Local Edge Machine: inference of dynamic models of gene regulation. Genome Biol 17(1):214

Morozov D, Weber G (2013) Distributed merge trees. In: Proceedings of the annual symposium on principles and practice of parallel programming, pp 93–102

Morozov D, Beketayev K, Weber G (2013) Interleaving distance between merge trees. Discrete Comput Geom 49:22–45

Mure LS, Le HD, Benegiamo G, Chang MW, Rios L, Jillani N, Ngotho M, Kariuki T, Dkhissi-Benyahya O, Cooper HM, Panda S (2018) Diurnal transcriptome atlas of a primate across major neural and peripheral tissues. Science. https://doi.org/10.1126/science.aao0318

Nerem R, Crawford-Kahrl P, Cummins B, Gedeon T (2019) A poset metric from the directed maximum common edge subgraph. arXiv:1910.14638

Orlando DA, Lin CY, Bernard A, Iversen ES, Hartemink AJ, Haase SB (2007) A probabilistic model for cell cycle distributions in synchrony experiments. Cell Cycle 6:478–488

Orlando DA, Lin CY, Bernard A, Wang JY, Socolar JE, Iversen ES, Hartemink AJ, Haase SB (2008) Global control of cell-cycle transcription by coupled cdk and network oscillators. Nature 453(7197):944–7

Pascucci V, Cole-Mclaughlin K, Scorzelli G (2004) Multi-resolution computation and presentation of contour trees. In: IASTED conference on visualization, imaging, and image processing

Pramila T, Wu W, Miles S, Noble WS, Breeden LL (2006) The forkhead transcription factor HCM1 regulates chromosome segregation genes and fills the S-phase gap in the transcriptional circuitry of the cell cycle. Genes Dev 20(16):2266–78

Rahi SJ, Pecani K, Ondracka A, Oikonomou C, Cross FR (2016) The CDK-APC/C oscillator predominantly entrains periodic cell-cycle transcription. Cell 165(2):475–87. https://doi.org/10.1016/j.cell.2016.02.060

Shedden K, Cooper S (2002) Analysis of cell-cycle gene expression in saccharomyces cerevisiae using microarrays and multiple synchronization methods. Nucleic Acids Res 30(13):2920–9

Simmons Kovacs LA, Orlando DA, Haase SB (2008) Transcription networks and cyclin/cdks: the yin and yang of cell cycle oscillators. Cell Cycle 7(17):2626–9

Simmons Kovacs LA, Mayhew MB, Orlando DA, Jin Y, Li Q, Huang C, Reed SI, Mukherjee S, Haase SB (2012) Cyclin-dependent kinases are regulators and effectors of oscillations driven by a transcription factor network. Mol Cell 45(5):669–79. https://doi.org/10.1016/j.molcel.2011.12.033

Simon I, Barnett J, Hannett N, Harbison CT, Rinaldi NJ, Volkert TL, Wyrick JJ, Zeitlinger J, Gifford DK, Jaakkola TS, Young RA (2001) Serial regulation of transcriptional regulators in the yeast cell cycle. Cell 106(6):697–708

Smirnov D, Morozov D (2017) Triplet merge trees. In: Topological methods in data analysis and visualization V (proceedings of TopoInVis 2017) (to appear)

Sugihara G, May R, Ye H, Hsieh C, Deyle E, Fogarty M, Munch S (2016) Detecting causality in complex ecosystems. Science 338:496

Thomas R (1991) Regulatory networks seen as asynchronous automata: a logical description. J Theor Biol 153:1–23. https://doi.org/10.1016/S0022-5193(05)80350-9

Zhang R, Lahens NF, Ballance HI, Hughes ME, Hogenesch JB (2014) A circadian gene expression atlas in mammals: implications for biology and medicine. Proc Natl Acad Sci U S A 111(45):16219–24. https://doi.org/10.1073/pnas.1408886111

Zomorodian A, Carlsson G (2005) Computing persistent homology. Discrete Comput Geom 33(2):249–274