DIRECTED GRAPH DESCRIPTORS AND DISTANCES FOR ANALYZING MULTIVARIATE TIME SERIES DATA

by

Robin Lynne Belton

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

of

Doctor of Philosophy

in

Mathematics

MONTANA STATE UNIVERSITY
Bozeman, Montana

May 2022

DEDICATION

To my parents, Karen and Keith Belton. The first time I wore doctoral regalia was in 1998 at the age of three. My dad was graduating with his Ph.D. in Public Policy from George Washington University, and I was ready to walk across the stage. My parents showed me a Ph.D. is possible. Twenty-four years later, and now I can officially walk across the stage.

ACKNOWLEDGEMENTS

To Tomáš Gedeon and Bree Cummins, thank you for making this thesis a reality - for championing it, refining it, and making me feel I was in great hands. Thank you Brittany Terese Fasy for making this work more concise and exposing me to the applied topology community. I am grateful to my committee members, Jack Dockery, Lukas Geyer, and Ryan Grady, along with David Ayala, Lisbeth Fajstrup, and David Millman for helping me immensely throughout my graduate studies. Thank you NSF DGE 1649608 for generously funding this work and three years of my Ph.D. program.

My friendships have kept much of my world warm and steady: thank you Anna, Marziah, Hannah, Katrina, Delanie, Tareen, Zoe, Katahdin, Robin, Catherine, Scotty, Nellie, Kai, Chris, Tori, Danielle, Brandon, and Dan. Thank you to the Department of Mathematical Sciences at MSU for always making me feel welcomed. To my beloved Kenyon College, thank you for showing me that math is cool.

To mom and dad, I am forever indebted to you. To my brother, thank you for your enthusiasm of this work when I needed to hear it. And to Doug Anderson, my partner-in-everything, thank you for upholding my dorkiness and building me a desk. I can't wait to marry you.

TABLE OF CONTENTS

TABLE OF CONTENTS – CONTINUED

LIST OF TABLES

Table Page

LIST OF FIGURES

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF FIGURES – CONTINUED

LIST OF FIGURES – CONTINUED

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF ALGORITHMS

ABSTRACT

Local maxima and minima, or *extremal events*, in experimental time series can be used as a coarse summary to characterize data. However, the discrete sampling in recording experimental measurements suggests uncertainty in the true timing of extrema during the experiment. This in turn gives uncertainty in the timing order of extrema within the time series. Motivated by applications in genomic time series and biological network analysis, we construct a weighted directed acyclic graph (DAG) called an *extremal event DAG* using techniques from persistent homology that is robust to measurement noise. Furthermore, we define a distance between extremal event DAGs based on the edit distance between strings. We prove several properties including local stability for the extremal event DAG distance with respect to pairwise L_∞ distances between functions in the time series data. Lastly, we provide algorithms, publicly free software, and implementations on extremal event DAG construction and comparison.

CHAPTER ONE

INTRODUCTION

Experimental time series data are ubiquitous in today's science and provide a window through which we can observe the underlying dynamics of complex systems, ranging from cells to ecosystems and climate. We study collections of time series which are also referred to as multivariate time series in the literature [82, 87]. We construct a weighted directed graph descriptor of a collection of time series data using persistent homology, a technique that belongs to a collection of approaches known as Topological Data Analysis (TDA) that uses algebraic topology [35, 54] to extract shape from data. TDA is used to study data from a wide range of applications including material science [46], cancer biology [85], and political science [26]. Some of the classic and foundational papers in TDA include [24, 29, 38, 63, 90].

Our descriptor characterizes a collection of time series by the order of their extrema in a way that also captures the robustness of this order with respect to measurement uncertainty. Our motivation comes from the desire to mathematically capture and compare collections of 'omics time series data, such as transcriptomics, proteomics, and others. In particular, the coarse information of orders of extrema have been used to assess regulatory network models of gene/protein interactions [22]. Other applications involve quantifying similarity between gene expression time series [7, 77] across repeated experiments.

Our mathematical methods are motivated by the combinatorial approaches in [7, 22, 60] that use only the approximate timing of time series extrema as the relevant features of experimental data. To take into account the uncertainty of capturing temporal orderings of extrema, [22] replaced the single time point locations of extrema with time intervals that were determined by manual inspection. If the intervals are disjoint, then the ordering of extrema is

interpreted to be robust to measurement uncertainty. In the follow-up paper [7], an approach was developed in which the intervals are algorithmically constructed using merge trees [25, 53], branch decompositions [52], and sublevel sets. These intervals are called ε-extremal intervals and have the property that they are the smallest intervals for which all continuous perturbations of a continuous function (with additional technical restrictions) that lie within an ε-band are guaranteed to attain an extremum under measurement uncertainty of size ε. Using the ε-extremal intervals, labeled directed acyclic graphs (DAGs) are constructed to represent the time series data for any fixed value of ε. We refer to these DAGs as ε-DAGs. Vertices or nodes in the ε-DAG represent extrema in the time series data. Directed edges $a \to b$ indicate that we can unambiguously discern the order (in time) of events corresponding to vertices a and b under measurement uncertainty of size ε.

Continuing this line of research, [60] defined a distance metric that compares two collections of time series by comparing the corresponding ε-DAGs. This metric involves computing the directed maximal common edge subgraph (DMCES) and was applied in [7] to quantify similarity in replicate experiments of microarray yeast cell cycle data. Additionally, the metric was used in [77] to provide quantitative evidence that an intrinsic oscillator drives the blood stage cycle of the malaria parasite *Plasmodium falciparum*. The metric for ε-DAGs using the DMCES is effective in capturing similarity between the time series, but is computationally expensive. This limits the total number of extrema across all time series that can be effectively analyzed. Another limitation is that the distance can only be measured at a single measurement uncertainty level ε, which is often unknown and thus the distance has to be computed multiple times for a collection of ε values. A better measurement would incorporate information about changes in similarity as a function of changing ε in a single value.

We significantly expand and generalize the work of [7] and [60] by constructing a *weighted* DAG that reflects robustness of the extremal ordering for all levels of measurement

error ε. We call this an *extremal event DAG*. Vertices in this graph again represent extrema in the time series data, and a directed edge $a \to b$ indicates that extremum a occurs before extremum b. The node weights measure prominence of extrema while the edge weights indicate the smallest ε level for which the relative order between the two associated extrema can no longer be guaranteed. The node weights are computed using sublevel set persistence [25]. After representing the collection of time series as an extremal event DAG, we define a distance between extremal event DAGs as a modified version of the edit distance (Chapter 15 of [21]). The edit distance quantifies similarity of two strings based on the minimum number of operations (e.g., insertion, substitution, and deletion) it takes to align the two strings. This distance is commonly used in many applications including DNA sequence alignment, see [59] for one of the first papers on the topic. The standard algorithm for the edit distance between two strings of length n can be computed via dynamic programming in $\Theta(n^2)$ (Chapter 15 of [21]).

We prove several key properties of the extremal event DAG weights. Most importantly for computability and applications, Theorem 3.2.2 gives a simple criterion to compute edge weights. Furthermore, we analyze stability properties of distances between extremal event DAGs. Chapter 5 gives stability results for distances used to compare extremal event DAGs with respect to pairwise L_∞ distances between the underlying continuous functions. These stability results show that small changes within time series data lead to small changes in the corresponding extremal event DAG distances. In Theorem 5.2.11, we show the extremal event DAG distance is stable in a local case: two paired continuous functions from the two collections of time series must lie within an ε band that allows for an unambiguous alignment of the minima and maxima between the two time series. Additionally, one of the time series can have small amplitude additional maxima and minima.

Lastly, in Chapter 6, we provide a polynomial time algorithm to compute the extremal event DAG given collections of discrete time series, and the dynamic program needed to

compute the extremal event DAG distance. Free and public software on computing extremal event DAGs and the distance between these descriptors is available at [4].

1.1 Biological Motivation

Extremal event DAGs are developed abstractly for mulitvariate time series in general, however, we focus on the application of analyzing 'omics data that measures expression levels of thousands of genes. Transcription of genes produces messenger RNA (mRNA) which are translated to proteins. Gene expression, measured by either the amount of mRNA produced (transcriptomics) or by the amount of corresponding protein (proteomics), can be used to measure the level of activity of a given gene product. There is strong evidence that the relative phases of oscillating regulators are important to controlling important cellular processes such as the cell cycle [44], circadian rhythm, or malaria parasite periodic infection of human blood cells. The assertion of [7, 22] is that the ordering of extrema is a reasonable approximation of control by phase relationship.

For example, it is hypothesized that a small transcriptional regulatory network controlling the cell cycle activates hundreds of other transcription factors in a phase-specific manner to play a vital role in maintaining the proper progression of DNA replication and cell division [10, 17, 43, 62, 73]. There are still many open questions about the precise role of the transcriptional network in the ordering of cell cycle events [68, 72]. A reproducible ordering of gene expression such as what was observed in [7] provides supporting evidence for the central role of the cell cycle gene regulatory network in orchestrating timely expression of other cell cycle events.

A question of interest in biology is evaluating the similarity of two experiments across labs or experimental conditions. For example, an experimentalist may wish to measure the similarity of expression level of genes driving the cell cycle between replicate experiments between time series collected under different growth conditions, or across organisms and

tissues. For example, circadian clock networks in different tissues that control the temporal ordering of phase specific gene expression [56,89]. Similarity and differences in timing of the same network in tissues like heart and liver can tell us about their mutual coupling as well as coupling to the master circadian clock in the brain [50,70]. In summary, mathematically modeling and comparing orders of extremal events in 'omics data is useful for identifying time series differences in multiple biological applications. In particular, extremal event DAGs and distances can be used to study the important biological questions about time dependent cellular processes, some of which we have mentioned here.

1.2 Background

We give an overview of topics from TDA that relate to this work. We refer the reader to [25, 35] for additional background in algebraic topology, and computational topology respectively.

1.2.1 Homology

Topology is a branch of mathematics that studies properties of objects and spaces that are preserved when stretched and compressed, but not torn, punctured, or glued. For example, a circle is topologically equivalent to a square since either can be stretched or bent into the other. However, a filled circle (i.e., a disc) is not topologically equivalent to a circle because one has a hole while the other does not. Reshaping a disc to a circle requires puncturing a hole. More technically speaking, topology studies properties of topological spaces that are invariant under continuous maps. Some of the classic texts in topology include [35, 54, 55].

Homology is an invariant under continuous maps and is a way to uncover n-dimensional "holes" in a topological space. The construction begins by defining a *chain complex* on a topological space X which is a sequence of abelian groups $C_0, C_1, \ldots,$ connected by

homomorphisms $\partial_n : C_n \to C_{n-1}$ called *boundary operators*, i.e.,

$$...\xrightarrow{\partial_{n+1}} C_n \xrightarrow{\partial_n} C_{n-1} \xrightarrow{\partial n-1} \dots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0$$

such that $\text{im}(\partial_{n+1}) \subset \ker(\partial_n)$. This is equivalent to the statement that the composition of two consecutive boundary maps is trivial, meaning $\partial_n \circ \partial_{n+1} = 0$ where 0 is the zero map. This property is saying that a boundary does not have a boundary. Since all groups C_n are abelian, all their subgroups are normal. Therefore $\text{im}(\partial_{n+1})$ is a normal subgroup of $\ker(\partial_n)$. The quotient group

$$H_n(X) = \ker(\partial_n)/\,\text{im}(\partial_{n+1})$$

is well-defined and is the n^{th} *homology group of* X. Each element of $H_n(X)$ is a *homology class*.

In this work, we are interested in the zeroth homology group of X that measures the connected pieces of X. Given $x, y \in X$, a *path* in X from x to y is a continuous map $f : [a, b] \to X$ of a closed interval of $\mathbb{R}$ into X such that $f(a) = x$ and $f(b) = y$. A topological space X is *path connected* if there exists a path between every pair of points in X. Given X, a natural way to partition it is by considering its connected pieces.

Definition 1.2.1 ((Path) Connected Components)**.** Let X be a topological space. Define an equivalence relation on X by declaring $x \sim y$ if there is a path connected subspace of X containing both x and y. The equivalence classes are the *(path) connected components* of X.

In this work, we refer to path connected components as connected components since path connectedness implies connectedness. A fundamental result in algebraic topology is that the rank of the zeroth homology group of X is equal to the number of connected components of X (Proposition 2.6 of [35]).

1.2.2 Simplicial Homology

Simplicial homology (See section 2.1 of [35]) is commonly used when analyzing data using topological techniques. This is possible when we are working with a vector space that is also a topological space. An *n-simplex* is the smallest convex set of $n+1$ vectors $v_0, v_1, \ldots, v_n$ where $v_1 - v_0, \ldots, v_n - v_0$ are linearly independent. We think of an n-simplex as an n-dimensional generalization of a triangle. See Figure 1.1. Additionally, the convex hull of any nonempty subset of the $n+1$ vectors is a *face* of the simplex, which is also a simplex. Attaching simplices together "nicely" gives a simplicial complex.

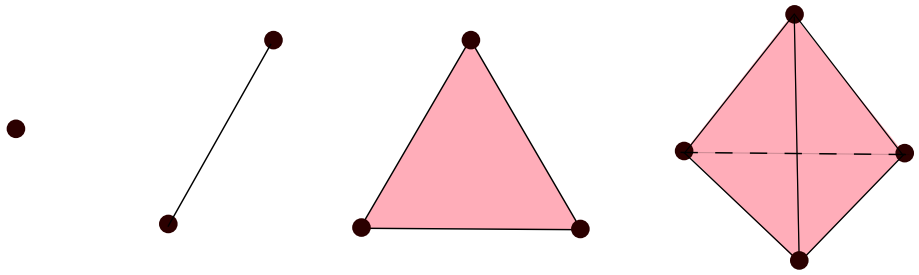


Figure 1.1: An n-simplex for $n = 0, 1, 2, 3$. A 0-simplex is a point, 1-simplex is an edge, 2-simplex is a filled triangle, and 3-simplex is a filled tetrahedron.

Definition 1.2.2 (Simplicial Complex)**.** A *simplicial complex* $\mathcal{K}$ is a set of simplices such that

- If τ is a face of a simplex $\sigma \in \mathcal{K}$, then $\tau \in \mathcal{K}$.
- If $\sigma_1, \sigma_2 \in \mathcal{K}$ and $\sigma_1 \cap \sigma_2 \neq \emptyset$, then $\sigma_1 \cap \sigma_2$ is a face of both σ_1 and σ_2.

Given a simplicial complex $\mathcal{K}$, we can define a chain complex. We define C_n to be a vector space with basis consisting of all n-simplices contained in $\mathcal{K}$. The coefficients of this vector space may be $\mathbb{R}$, $\mathbb{Q}$, or a finite field $\mathbb{Z}/p\mathbb{Z}$ where p is prime. The vector space consists

of all linear combinations $\sum_i k_i\sigma_i$ where $k_i \in \mathbb{F}$ such that $\mathbb{F}$ is a field and the sum is over all n-simplices $\sigma_i \in \mathcal{K}$.

The boundary of an n-simplex is a signed union of $(n-1)$-simplices that are faces of the n-simplex. Mathematically, the boundary map $\partial_n : C_n \to C_{n-1}$ is defined on an n-simplex $\sigma = [v_0, v_1, \ldots, v_n]$ by

$$\partial_n(\sigma) = \sum_{i=1}^{n} (-1)^i [v_0, \ldots, \hat{v_i}, \ldots, v_n]$$

where $[v_0, \ldots, \hat{v_i}, \ldots, v_n]$ is the $(n-1)$-simplex obtained by removing the vertex v_i. For example, $\partial_2([v_0, v_1, v_2]) = [v_1, v_2] - [v_0, v_2] + [v_0, v_1]$.

Having constructed a chain complex for a simplicial complex, we can compute the n^{th} *simplicial homology group* of $\mathcal{K}$ with coefficient field $\mathbb{F}$ to be

$$H_n(\mathcal{K}, \mathbb{F}) := \ker(\partial_n)/\operatorname{im}(\partial_{n+1}).$$

1.2.3 Sublevel Set Persistence

Topology is often applied to data by fist constructing some mathematical shape or structure out of the data. For example, if our data is in the form of a point cloud, we could construct line segments, triangles, tetrahedrons, and other geometric shapes i.e., simplicial complexes from the data points. Common constructions that do this include the Vietoris Rips complex [32, 36, 84] and the Čech complex [30]. We can then study the data by analyzing topological features such as the number of connected components or the homology of the associated Vietoris-Rips complex or Čech complex. However, the construction of the complexes from the data depends on a scale parameter. Studying how the homology changes over all scale parameters is the key idea behind persistent homology which is the main tool used in TDA. Some of the first papers to apply persistent homology include [27, 69].

We work with a specific case of persistent homology that encodes the changes of the connectedness of sublevel sets of a real-valued function $f : X \to \mathbb{R}$ defined on a topological

space X as the height parameter ranges from $-\infty$ to ∞. This information is encoded in a topological descriptor called a persistence diagram and encodes the prominence of the local extrema of f. Persistent homology is a general mathematical framework and we only provide the definitions necessary for our results here; see [25, 65] for more detailed introductions to persistence. Below we define sublevel set persistence using notation similar to page 181 of [25]. One assumption we need is tameness of f.

Definition 1.2.3 (Homological Critical Values)**.** Let X be a topological space, $f : X \to \mathbb{R}$ a function. We call $a \in \mathbb{R}$ a *homological critical value* if there exists a non-negative integer n and $\delta > 0$ such that for all $0 < \varepsilon < \delta$, the linear map $H_n(f^{-1}(-\infty, a-\varepsilon]) \to H_n(f^{-1}(-\infty, a+\varepsilon])$ induced by the inclusion of sublevel sets is not an isomorphism.

In other words, the homological critical values are the values at which the homology of the sublevel sets change. For Morse functions f on a smooth manifold, these points are exactly the heights of the local extrema of f [51].

Definition 1.2.4 (Tameness)**.** Let X be a topological space. A function $f : X \to \mathbb{R}$ is *tame* if it has a finite number of homological critical values and the homology groups $H_n(f^{-1}(-\infty, a])$ are finite for every $a \in \mathbb{R}$.

Next, we consider a nested sequence of sublevel sets. A *filtration* of a topological space X is a nested family of subspaces $(X_r)_{r\in T}$ starting at the empty set, where $T \subset \mathbb{R}$, such that for all $r, s \in T$ where $r \leq s$, we have $X_r \subset X_s$, and $\bigcup_{r\in T} X_r = X$. For $f : X \to \mathbb{R}$, the sequence of all *sublevel sets* $(f^{-1}(-\infty, r])_{r\in\mathbb{R}}$, ordered by inclusion and indexed by $\mathbb{R}$, is called the *sublevel set filtration*. Observe that the inclusion map between sublevel sets induces a homomorphism between corresponding homology groups. In particular, for $r \leq s$, we have $f^{-1}(-\infty, r] \subset f^{-1}(-\infty, s]$. The inclusion map $\iota : f^{-1}(-\infty, r] \to f^{-1}(-\infty, s]$ induces

a homomorphism between homology groups

$$g_n^{r,s} : H_n(f^{-1}(-\infty, r]) \to H_n(f^{-1}(-\infty, s]).$$

This homomorphism takes the homology of the sublevel set of $f^{-1}(-\infty, r]$ to the homology of the sublevel set of $f^{-1}(-\infty, s]$. The image of $g_n^{r,s}$ contains all this important information. We define the n^{th} *persistent homology groups* to be the images of all the homomorphisms $H_n^{r,s} := \text{im}(g_n^{r,s})$. The n^{th} *persistent Betti numbers* are the ranks; $\beta_n^{r,s} := \text{rank}(H_n^{r,s})$.

Next we describe how to encode the persistent homology groups into a multiset of points in the extended plane. Consider a tame function $f : X \to \mathbb{R}$. Let $(r_i)_{i=1}^N$ be the ordered sequence of homological critical values of f. Since we are working with a tame function, there are only a finite number of heights we need to consider where the sublevel sets change. To ease notation, we write $H_n^{i,j} := H_n^{r_i,r_j}$, $\beta_n^{i,j} := \beta_n^{r_i,r_j}$, and $g_n^{i,j} := g_n^{r_i,r_j}$. The persistent homology group $H_n^{i,j}$ consists of the homology classes of $f^{-1}(-\infty, r_i]$ that still persist in $f^{-1}(-\infty, r_j]$. The persistent Betti number $\beta_n^{i,j}$ counts the number of homology classes that persist between $f^{-1}(-\infty, r_i]$ and $f^{-1}(-\infty, r_j]$. The first index for which a homology class appears is the *birth* of that class. When a class in $f^{-1}(-\infty, r_{i-\varepsilon}]$ merges with another in $f^{-1}(-\infty, r_i]$, then the class *dies* at a height of r_i. When classes merge together we follow the *elder rule* (See section 7.1 of [25]) that requires the class with a greater birth height to merge with the class of the lower birth height.

Definition 1.2.5 (Persistence Diagram $D_n(f)$)**.** Let $f : X \to \mathbb{R}$ be a tame function with homological critical values $R := \{r_i\}_{i=1}^N$. Let $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$. Consider the set $S := \{r_i\}_{i=1}^N \cup \{\infty\}$. The n^{th} dimensional *persistence diagram* $D_n(f)$ is the multiset set of points

in $\overline{\mathbb{R}}^2$ such that the point $p = (r_i, r_j) \in R \times S$ where $r_i \leq r_j$ is included with multiplicity,

$$\mu_n(p) = \lim_{\varepsilon \to 0^+} (\beta_n^{i,j} - \beta_n^{i,j+\varepsilon}) - (\beta_n^{i-\varepsilon,j} - \beta_n^{i-\varepsilon,j+\varepsilon}).^1$$

We set $p \in D_n(f)$ if, and only if, $\mu_n(p) > 0$.

In regards to $\mu_n(p)$, the first difference counts the number of homology classes that are born at or before a height of r_i and die at a height of r_j. The second difference counts the number of homology classes that are born at or before a height of $r_i - \varepsilon$ and die at a height of r_j.

The persistence diagram summarizes the homology groups as the height parameter ranges from $-\infty$ to ∞. Each persistence point $p = (b, d) \in D_n(f)$ is called a *birth-death pair* since it represents a unique generator of the homology groups of the sublevel sets of f that is born at parameter b and dies going into parameter d. In this work, we are concerned with a special type of tame function.

Definition 1.2.6 (Nicely Tame Functions)**.** Let $X \subset \mathbb{R}$ be a topological space. A function $f : X \to \mathbb{R}$ is *nicely tame* if f is tame, continuous, and for each critical value y, the preimage $f^{-1}(y)$ is a finite set.

Specifically, we work with a nicely tame function $f : [a_1, a_2] \to \mathbb{R}$ where $x, y \in \mathbb{R}$. If the function values at the local extrema are unique, then there is a one-to-one correspondence between persistence points and the local minima of f. Then a persistence point (b, d) corresponds to the local minimum $(t, f(t) = b)$ since the homological critical values are the local minima and maxima of f.

In the event that the values of several minima are the same, this correspondence is not unique. However, a unique correspondence can be induced by fixing an order on the local

[1] In the case of $p = (r_i, \infty)$ we assume that the sublevel set at a height parameter of ∞ is the entire domain X and no homology classes are present at a height parameter $\infty + \varepsilon$.

minima (e.g., the domain coordinates) and using that ordering to break ties. For a multiset A, we write $|A|$ for the *total multiplicity of* A i.e., $|A| = \sum_{p \in A} \mu(p)$. Figure 1.2 gives an example of a function and its persistence diagram from a sublevel set filtration.

In this work, we are only concerned with $D_0(f)$ and we denote $D(f) := D_0(f)$. Furthermore, all our persistence diagrams have a unique point in $D(f)$ with a death coordinate of ∞. We call this the *essential connected component*. If t does not represent the essential component, then there exists a local maximum $(t', f(t'))$ such that $(f(t), f(t')) \in D(f)$. In this case, $f(t')$ is the height at which the connected component of $f^{-1}(-\infty, f(t')]$ containing t merges with another connected component of the sublevel set $f^{-1}(-\infty, f(t')]$ represented by a local minimum s where $f(s) < f(t)$.[2] We call $f(t')$ the *merge height of* t.

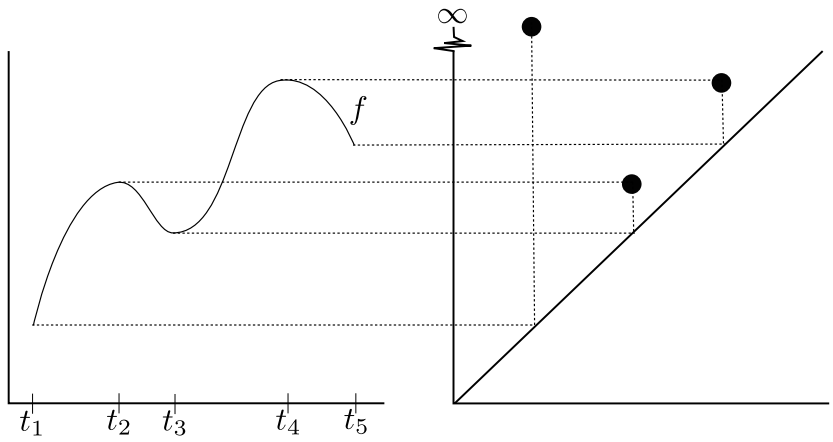


Figure 1.2: Left. A function, $f : [a_1, a_2] \to \mathbb{R}$. Right. Persistence diagram of f, $D(f)$ obtained from a sublevel set filtration of f. The set $D(f)$ is $\{(f(t_1), \infty), (f(t_3), f(t_2)), (f(t_5), f(t_4))\}$. The first coordinate of each point in $D(f)$ is the height of a local minimum, while the second coordinate is the height of a local maximum or ∞.

Sublevel set persistence has been previously studied to characterize prominent features of data. In [5], well groups that are defined using $\varepsilon > 0$ and sublevel sets of a continuous

[2]The choice of pairings in $D(f)$ follows the Elder Rule. In the case where both connected components are born at the same height, we arbitrarily choose to continue the connected component that occurs first in the domain.

function between topological spaces are used to measure the robustness of homology groups from sublevel sets of a function under any ε-perturbation. Unlike ε-extremal intervals, well groups are algebraic groups and are not specific to a local extremum. Furthermore, in [15], sublevel set persistence is used to define a topological regularizer that can then be used as a classifier for machine learning.

1.2.4 TDA in Analyzing Time Dependent Data

Our use of TDA differs significantly from other approaches that use TDA to study time series data and the dynamics of underlying biological networks. One common method studies single variable time series by using Takens' embedding theorem [80]. The data is transformed by computing a sliding window embedding of the data into a point cloud in $\mathbb{R}^n$. This point cloud is then analyzed using one dimensional persistent homology to detect and quantify periodicity. Examples that take this or a modified approach include detecting periodicity of genomic time series data [66], studying entropy and the dynamics of the underlying system [1, 75], characterizing gene regulatory networks [8], and distinguishing between audio signals of the same note from different instruments [71].

The second common method to study single variable time series using TDA is to apply sublevel set persistence to detect prominent features. Applications include signal processing [40], Fourier spectrum analysis and parameter detection [57], arrythmia detection [23], and cancer studies [45]. Furthermore, sublevel set persistence on time series can be used to determine noise that is often seen as small peaks in the time series [58].

Lastly, TDA has been used to study other types of time dependent data. One includes dynamic metric spaces that can be used to describe phenomena such as bird flocks, insect swarms, schools of fish, and aphid trajectories. TDA techniques to study these type of data include vineyards [20], CROCKER plots [81, 83, 88], spatiotemporal filtrations [42] and zig-zag persistence [12, 41]. Furthermore, time series data of fMRI images have been used to

construct functional networks, and then applying filtrations on the weights of these networks to extract topological features [67, 79]. A more extensive summary on TDA techniques to study time series can be found in [28].

1.2.5 Stability of Persistence Diagrams

A desired property of any descriptor of data is that small perturbations in the data result in small perturbations of the data descriptor. This property is referred to as stability. Fortunately, persistence diagrams are stable [2, 11, 18, 19, 74], and the stability of persistence diagrams is one of the most important results in TDA. To formulate the stability result, we need to define distances between persistence diagrams. These distances involve bijections between points in two diagrams. Since diagrams often contain a different number of points, we need a way to account for this. Hence, for the purpose of defining distances, we consider the diagonal $\Delta = \{(x, x) \in \overline{\mathbb{R}}^2\}$ to be part of the persistence diagram where each point on the diagonal has infinite multiplicity.

Definition 1.2.7 (Bottleneck Distance)**.** Let Γ denote the set of all bijections $\gamma : D_n(f) \to D_n(g)$ where $f, g : X \to \mathbb{R}$ are real-valued tame functions from a topological space X. The *bottleneck distance* between two persistence diagrams is:

$$W_\infty(D_n(f), D_n(g)) = \inf_{\gamma \in \Gamma} \sup_{x \in D_n(f)} \|x - \gamma(x)\|_\infty .$$

The bottleneck distance is computed by first recording the largest L_∞ distance between paired points over all bijections, and then picking the smallest of these values to be the bottleneck distance. Importantly, the bottleneck distance is a metric on the space of persistence diagrams [25]. Another common distance between persistence diagrams is the *interleaving distance* [13] which can be shown to be equivalent to the bottleneck distance [47].

Bottleneck stability was originally proved in [18]. One assumption needed is that X

is a *triangulable space* meaning X is homeomorphic to a finite simplicial complex. Finally, the following theorem uses $\mathbb{Z}/2\mathbb{Z}$ coefficients for homology, which are the coefficients that we adapt for the rest of the work presented in this thesis.

Theorem 1.2.8 (Bottleneck Stability Theorem [18])**.** *Let X be a triangulable space with continuous tame functions $f, g : X \to \mathbb{R}$. Then for every $n \geq 0$,*

$$W_\infty(D_n(f), D_n(g)) \leq \|f - g\|_\infty .$$

This result has also been generalized to algebraic [2] and categorical settings [11]. Another common distance between persistence diagrams also involves bijections of points.

Definition 1.2.9 (Wasserstein Distance)**.** Let $p > 0$. Let X be a topological space with tame functions $f, g : X \to \mathbb{R}$. Let Γ denote the set of all bijections $\gamma : D(f) \to D(g)$. The p^{th} *Wasserstein distance* between two persistence diagrams is:

$$W_p(D_n(f), D_n(g)) := \inf_{\gamma \in \Gamma} \left(\sum_{x \in D_n(f)} \|x - \gamma(x)\|_\infty^p \right)^{\frac{1}{p}} .$$

The bottleneck distance can be viewed as a limit of p-Wasserstein distances [14, 18]

$$W_\infty(D_n(f), D_n(g)) := \lim_{p \to \infty} W_p(D_n(f), D_n(g)).$$

Persistence diagrams with respect to the sum of Wasserstein distances over all dimensions are stable for Lipshitz functions on triangulable, compact metric spaces [19]. Wasserstein stability results are generalized in [74].

1.2.6 Edit Distance

We now introduce another distance which is applicable to finite strings of text rather than persistence diagrams. Suppose we wish to transform one string of text $\mathbf{x} = (x_1, \ldots, x_m)$ to another string $\mathbf{y} = (y_1, \ldots, y_n)$ using a combination of the following "editing" operations: insertion, deletion, and substitution. Each operation has a non-negative cost and the *Levenshtein distance* is the minimum cost of the sequence of operations needed to transform $\mathbf{x}$ to $\mathbf{y}$ [48]. Consider the following example of transforming SUNDAY to SUNNY. We assume a cost of one for each operation.

1. SUN**N**Y $\rightarrow$ SUN**D**Y (substitute "D" for "N")
2. SUNDY $\rightarrow$ SUND**A**Y (insert "A")

We performed two operations, and this is the smallest number of operations needed to transform SUNNY to SUNDAY or vice-versa, and so the Levenshtein distance is equal to two. This distance belongs to a larger family of *edit distances* that allows for any set of operations as long as there is an inverse operation with equal cost. Since insertion and deletion are inverse operations of one another and substitution is an inverse operation of itself, the Levenshtein distance is a type of *edit distance* and it is sometimes referred to as the edit distance. Each edit distance satisfies the properties of a distance metric [31].

To compute an edit distance, we could define a recursive algorithm that checks through all ways we can transform one string to another. However, this is highly inefficient with an exponential time complexity. Fortunately, we can use *dynamic programming* which uses the the edit distance between all substrings of the input to efficiently compute the edit distance between the two strings [3]. See Chapter 15 of [21] for a more detailed exposition to dynamic programming.

To compute the Levenshtein distance between two strings $\mathbf{x}$ and $\mathbf{y}$ of lengths m and n respectively via dynamic programming, we construct an $(m+1) \times (n+1)$ matrix.

Definition 1.2.10 (Edit Matrix)**.** Let $\mathbf{x} = (x_1, x_2, \ldots, x_m)$, $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ be strings. The *edit matrix*, denoted by $\mathbf{mat}$, is an $(m+1)\times(n+1)$ matrix recursively defined as follows:

$$\mathbf{mat}[i,j] = \begin{cases} 0 & i = j = 1 \\ i-1, & i > 1, j = 1 \\ j-1, & i = 1, j > 1 \\ \min \left\{ \begin{array}{l} \mathbf{mat}[i-1,j]+1 \\ \mathbf{mat}[i,j-1]+1 \\ \mathbf{mat}[i-1,j-1]+\mathbf{diff}(x_{i-1},y_{j-1}) \end{array} \right\} & \text{otherwise,} \end{cases}$$

where $\mathbf{diff} : \mathbf{x} \times \mathbf{y} \to \{0,1\}$ is defined by

$$\mathbf{diff}\,(x_k, y_l) = \begin{cases} 0, & \text{if } x_k = y_l \\ 1, & \text{otherwise} \end{cases}.$$

The Levenshtein distance is equal to $\mathbf{mat}[m+1, n+1]$. See Figure 1.4 for the edit matrix for strings SUNNY and SUNDAY.

To find a sequence of operations resulting in the Levenshtein distance, we can apply backtracking in the edit matrix. We start at $\mathbf{mat}[m+1, n+1]$ and recursively trace through the entries that led to this value until we hit $\mathbf{mat}[0,0]$. Each path from $\mathbf{mat}[m+1, n+1]$ to $\mathbf{mat}[0,0]$ indicates a sequence of operations resulting in the Levenshtein distance given by $\mathbf{mat}[m+1, n+1]$. The following moves through the matrix $\mathbf{mat}$ in where adjacent entries have a difference of one mean the following operations for transforming string $\mathbf{x}$ to string $\mathbf{y}$:

- Vertical move from $\mathbf{mat}[i,j]$ to $\mathbf{mat}[i-1,j]$ indicates deleting x_{i-1}.
- Horizontal move from $\mathbf{mat}[i,j]$ to $\mathbf{mat}[i,j-1]$ indicates inserting y_{j-1} between x_{i-1}

	j	1	2	3	4	5	6	7
i			S	U	N	D	A	Y
1		0	1	2	3	4	5	6
2	S	1	0	1	2	3	4	5
3	U	2	1	0	1	2	3	4
4	N	3	2	1	0	1	2	3
5	N	4	3	2	1	1	2	3
6	Y	5	4	3	2	2	2	2

Figure 1.3: Edit matrix for computing the Levenshtein distance between strings SUNNY and SUNDAY where the operations of insertion, deletion, and substitution have a cost of one. The Levenshtein distance is equal to $\mathbf{mat}[6, 7] = 2$.

and x_i.

- Diagonal move from $\mathbf{mat}[i, j]$ to $\mathbf{mat}[i-1, j-1]$ indicates substituting y_{j-1} for x_{i-1}.

See Figure 1.4 for an example of backtracking.

Edit distances have many applications. In computational biology, edit distances are used to find optimal alignments in DNA and protein sequences. The classic algorithms are Needleman-Wunsch [59], Smith-Waterman [78] and Hirschberg's [37]. Other applications include natural language processing and information retrieval [49, 61, 64].

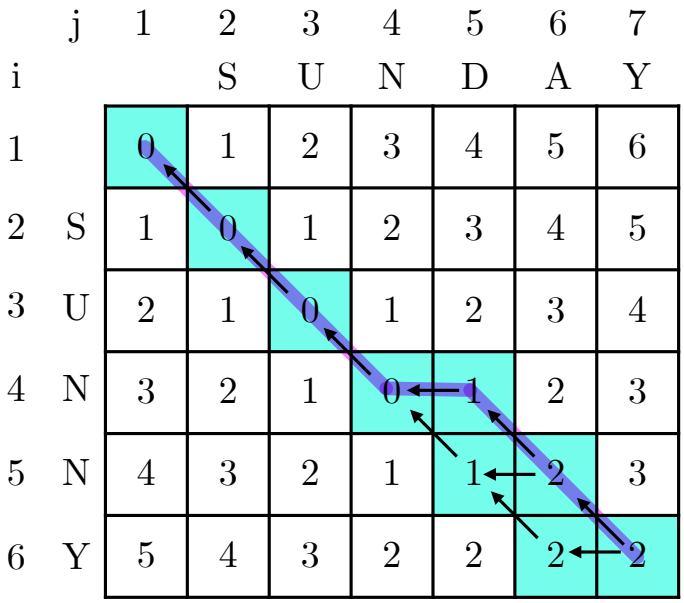


Figure 1.4: Backtracking in edit matrix for computing the Levenshtein distance between strings SUNNY and SUNDAY. The arrows between highlighted entries are from backtracking. The path highlighted in purple is one sequence of operations between SUNNY and SUNDAY that results in a Levenshtein distance equal to two. The horizontal move from $\mathbf{mat}[4,5]$ to $\mathbf{mat}[4,4]$ means we insert y_4 between x_3 and x_4. Hence, we insert "D" between "N" and "N" in SUNNY. The diagonal move from $\mathbf{mat}[5,6]$ to $\mathbf{mat}[4,5]$ means we substitute y_5 for x_4. Thus we substitute "A" for the second "N" in SUNNY. Combining these two operations we get SUNDAY.

CHAPTER TWO

PRELIMINARIES

We now summarize necessary terminology for the results that have been developed in this thesis. Many of these terms build off of ideas mentioned in Chapter 1. Throughout this section and the rest of this paper, we use the following notation. Let $X \subset \mathbb{R}$ denote an arbitrary subset of $\mathbb{R}$. Let $C := [a_1, a_2] \subset \mathbb{R}$ be a closed interval of $\mathbb{R}$.

2.1 Extrema

For a subset $X \subset \mathbb{R}$, $x \in X$ and $\varepsilon > 0$, let $B_\varepsilon(x)$ be the open neighborhood of radius ε centered at x. That is

$$B_\varepsilon(x) := \{y \in X \mid |y - x| < \varepsilon\}.$$

Definition 2.1.1 (Local Extrema)**.** Let $f : X \to \mathbb{R}$ be a function. We say f has a *local minimum* at $x \in X$ if there exists $\varepsilon > 0$ for which $f(x) < f(y)$ for all $y \in B_\varepsilon(x) \setminus \{x\}$. Similarly, f has a *local maximum* at $x \in X$ if there exists $\varepsilon > 0$ for which $f(x) > f(y)$ for all $y \in B_\varepsilon(x) \setminus \{x\}$. We refer to any local minimum or local maximum as a *local extremum* of f. If $x \in X$ with $f(x) < f(y)$ for all $y \in X \setminus \{x\}$, we say f has a *global minimum* at x. Similarly, for $x \in X$ where $f(x) > f(y)$ for all $y \in X \setminus \{x\}$, we say f has a *global maximum* at x.

We often order the extrema of a function. To ease notation, we write $[n]$ to be the set of the first n integers. That is

$$[n] := \{1, 2, \ldots, n\}.$$

2.2 Distances

We use the L_∞ metric to quantify distances between collections of points and functions.

Definition 2.2.1 (L_∞ metric)**.** For points $p = (p_1, p_2, \ldots, p_n)$, $q = (q_1, q_2, \ldots, q_n) \in \overline{\mathbb{R}}^n$, we define the L_∞ *distance between points* p and q as $\|p - q\|_\infty = \max_i |p_i - q_i|$[1]. For functions $f, g : K \to \mathbb{R}$ where $K \subset \mathbb{R}$ is compact, we define the L_∞ *distance between functions* f and g as $\|f - g\|_\infty = \sup_{x \in K} |f(x) - g(x)|$.

For a subset $X \subset \overline{\mathbb{R}}^n$, $x \in X$ and $\varepsilon > 0$, let $\square_\varepsilon(x)$ be the L_∞ open neighborhood of radius ε centered at x. That is

$$\square_\varepsilon(x) := \{y \in X \mid \|y - x\|_\infty < \varepsilon\}.$$

2.3 ε-Perturbations

We consider perturbations of a function f.

Definition 2.3.1 (ε-Neighborhood of f)**.** Let $f \colon K \to \mathbb{R}$ be a continuous function, where $K \subset \mathbb{R}$ is compact. For $\varepsilon \geq 0$, define

$$N_\varepsilon(f) := \{g : K \to \mathbb{R} \mid g \text{ is continuous and } \|f - g\|_\infty < \varepsilon\}$$

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

2.4 ε-Extremal Intervals

Let $\mathrm{INT}(C)$ be the set of relatively open intervals contained in $C := [a_1, a_2] \subset \mathbb{R}$. To enable comparability between local extrema for functions in $N_\varepsilon(f)$ for different levels of ε,

[1] We set $|\infty - \infty| := 0$.

we use the following definition.

Definition 2.4.1 (ε-Extremal Interval at t)**.** Let $f : C \to \mathbb{R}$ be a continuous function and $T = \{t_i\}_{i=1}^n$ be the domain coordinates of all local extrema of f. Let $\varepsilon > 0$. Define $\varphi_\varepsilon^f : T \to \mathrm{INT}(C)$ such that

Case 1 If $t \in T$ and $(t, f(t))$ is a local minimum, define $\varphi_\varepsilon^f(t)$ to be the connected component of $(f - \varepsilon)^{-1}(-\infty, f(t) + \varepsilon)$ that contains t.

Case 2 If $t \in T$ and $(t, f(t))$ is a local maximum, define $\varphi_\varepsilon^f(t)$ to be the connected component of $(f + \varepsilon)^{-1}(f(t) - \varepsilon, \infty)$ that contains t.

We call $\varphi_\varepsilon^f(t)$ the *ε-extremal interval at t* (see Figure 2.1).

We note that we often refer to the ε-extremal intervals as *extremal intervals*. We also sometimes omit the superscript f and simply write φ_ε when the function used to construct the ε-extremal intervals is clear.

Remark 2.4.2 (Notation for Endpoints of Extremal Intervals)*.* Let $f : C \to \mathbb{R}$ be a continuous function with a local extremum at $t \in C$. Suppose $\varphi_\varepsilon^f(t)$ is the ε-extremal interval at t. We define the left endpoint of $\varphi_\varepsilon^f(t)$ to be $\mathsf{left}(\varphi_\varepsilon^f(t)) := \inf(\varphi_\varepsilon^f(t))$. We define the right endpoint of $\varphi_\varepsilon^f(t)$ to be $\mathsf{right}(\varphi_\varepsilon^f(t)) := \sup(\varphi_\varepsilon^f(t))$. Finally, we denote the length of $\varphi_\varepsilon^f(t_i)$ by $\mathsf{len}(\varphi_\varepsilon^f(t))$.

The next Lemma shows that ε-extremal intervals are nested as a function of increasing $\varepsilon > 0$.

Lemma 2.4.3 (ε-Extremal Intervals)**.** *Let $f : C \to \mathbb{R}$ be a nicely tame function. Let t be the domain coordinate of a local extremum of f. If $0 < \varepsilon_0 < \varepsilon_1$, then,*

$$\varphi_{\varepsilon_0}(t) \subset \varphi_{\varepsilon_1}(t).$$

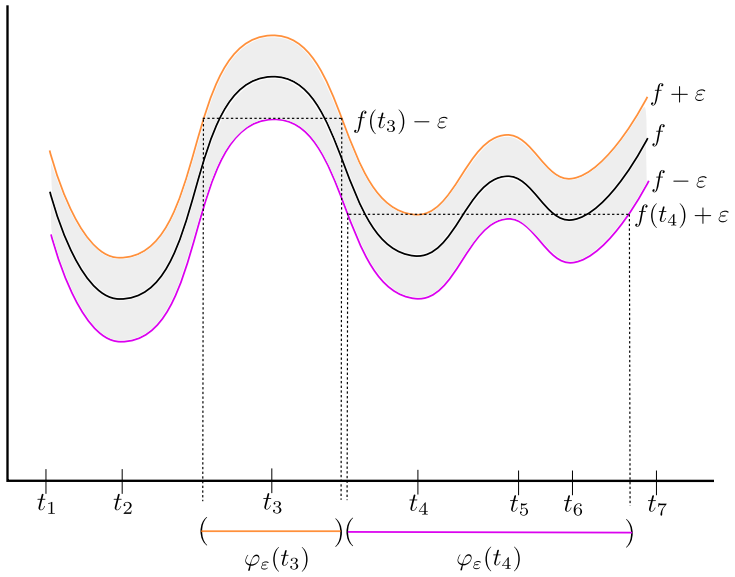


Figure 2.1: The ε-extremal intervals at t_3 and t_4. The ε-extremal interval at t_3 is the connected component of $(f+\varepsilon)^{-1}(f(t_3)-\varepsilon,\infty)$ that contains t_3 (Case 2). The ε -extremal interval at t_4 is the connected component of $(f-\varepsilon)^{-1}(-\infty,f(t_4)+\varepsilon)$ that contains t_4 (Case 1).

Proof. Since sublevel sets of a function form a filtration, we have

$$(f-\varepsilon_0)^{-1}(-\infty,f(t)+\varepsilon_0)\subset(f-\varepsilon_0)^{-1}(-\infty,f(t)+\varepsilon_1).$$

At the same time, because $\varepsilon_0<\varepsilon_1$,

$$(f-\varepsilon_0)^{-1}(-\infty,f(t)+\varepsilon_1)\subset(f-\varepsilon_1)^{-1}(-\infty,f(t)+\varepsilon_1).$$

Combining these two statements we find that

$$(f-\varepsilon_0)^{-1}(-\infty,f(t)+\varepsilon_0)\subset(f-\varepsilon_1)^{-1}(-\infty,f(t)+\varepsilon_1). \tag{2.1}$$

Note that $\varphi_{\varepsilon_0}(t)\cap\varphi_{\varepsilon_1}(t)\neq\emptyset$ because they are connected components of $(f-\varepsilon_0)^{-1}(-\infty,f(t)+\varepsilon_0)$ and $(f-\varepsilon_1)^{-1}(-\infty,f(t)+\varepsilon_1)$ respectively that contain t.

Therefore (2.1) implies $\varphi_{\varepsilon_0}(t) \subset \varphi_{\varepsilon_1}(t)$. □

2.5 Node Lives

Given a function $f\colon C \to \mathbb{R}$ and the domain coordinate t of a local extremum, any continuous function $g \in N_\varepsilon(t)$ is guaranteed to have the same type of extremum in the interval $\varphi_\varepsilon^f(t)$ for small enough ε. At some value of ε, however, this is no longer guaranteed. We use the persistence diagram $D(f)$ to assist us with understanding when this occurs.

Definition 2.5.1 (Birth-Death Pairing Map)**.** Let $f : C \to \mathbb{R}$ be a nicely tame function. Let $\{t_i\}_{i=1}^n$ be the set of domain coordinates of the local minima of f. Define the *birth-death pairing map* to be $\zeta_f : \{t_i\}_{i=1}^n \to \mathbb{R}_{>0}$ by

$$\zeta_f(t_i) = \begin{cases} \max(f) & \text{if } t_i \text{ represents the essential component} \\ f(t_j) & \text{otherwise,} \end{cases}$$

where $f(t_j)$ is the merge height of the minimum at t_i.

Observe the minima for f are the maxima of $-f$ and vice-versa. Additionally, the absolute difference in heights between extrema of f remain the same in both f and $-f$. Hence, we can study the prominence of maxima of f by studying the minima of $-f$. This follows from [6] which discusses the symmetry between persistence diagrams computed from height filtrations that are ascending versus descending.

Definition 2.5.2 (Persistence of Extrema)**.** Let $f : C \to \mathbb{R}$ be a nicely tame function, and let $(b, d) \in D(f)$. The persistence of (b, d) is the difference between the birth and death times, $d - b$. Suppose t is the domain coordinate such that $f(t) = b$ and $(t, f(t))$ is the local minimum of f representing the pair (b, d). We define the *persistence of the extremum*

$(t, f(t))$, denoted $\text{pers}_f(t)$, as

$$\text{pers}_f(t) := \begin{cases} \max(f) - f(t), & \text{if } (t, f(t)) \text{ is the global minimum of } f \\ d - b, & \text{if } (t, f(t)) \text{ is a local (and not global) minimum of } f \\ \text{pers}_{-f}(t), & \text{if } (t, f(t)) \text{ is a local maximum of } f. \end{cases}$$

See Figure 2.2 for an example of computing the persistence of local extrema.

Definition 2.5.3 (Node Life)**.** Let $f : C \to \mathbb{R}$ be a nicely tame function with a local extremum at domain coordinate t. The *node life* of t is $\text{pers}_f(t)/2$.

We sometimes omit the subscript f from $\text{pers}_f(t)$ when the function we are computing the node life from is clear. Proposition 2 and Corollary 1 from [7] states that $\varphi_\varepsilon^f(t)$ is the smallest interval for which any nicely tame ε-perturbation of f is guaranteed to have at least one local extremum of the same type as t, as long as ε is less than the node life.

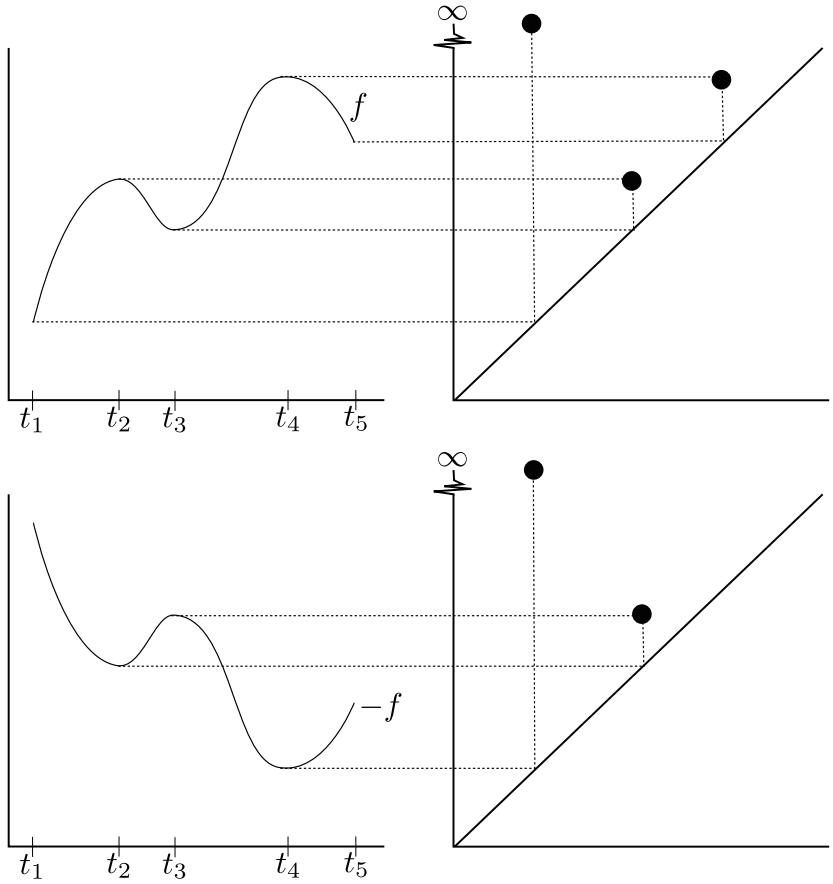


Figure 2.2: Top. A continuous function f and its persistence diagram from a sublevel set filtration. In this example, $\text{pers}_f(t_1) = \max(f) - f(t_1)$, $\text{pers}_f(t_3) = f(t_2) - f(t_3)$, and $\text{pers}_f(t_5) = f(t_4) - f(t_5)$. Bottom. $-f$ and its persistence diagram from a sublevel set filtration. Now we can compute the persistence of the local maxmima of f as $\text{pers}_f(t_4) = f(t_4) - \min(f)$ and $\text{pers}_f(t_2) = f(t_2) - f(t_3)$.

CHAPTER THREE

EXTREMAL EVENT DAG

In this section, we define the extremal event DAG that captures information about extrema of multiple time series. To do so, we first need a notion of comparability of local extrema.

Definition 3.0.1 (Comparability of Extrema)**.** Let $f, g : C \to \mathbb{R}$ be nicely tame functions. Let t_f, t_g be local extrema of f and g respectively. Let $\varepsilon > 0$. We declare $t_f \prec_\varepsilon t_g$ if for every nicely tame ε-perturbation of f and g there exists ε-perturbed extrema t'_f and t'_g such that $t'_f \in \varphi^f_\varepsilon(t_f)$, $t'_g \in \varphi^g_\varepsilon(t_g)$, and $t'_f < t'_g$. We say t_f and t_g are *comparable at* ε if all of the following hold.

1. $\text{pers}_f(t_f) > 2\varepsilon$
2. $\text{pers}_g(t_g) > 2\varepsilon$
3. $t_f \prec_\varepsilon t_g$ or $t_g \prec_\varepsilon t_f$

If at least one of these conditions does not hold, then t_f and t_g are *incomparable at* ε.

Definition 3.0.1 relates order of extrema to possible ε-perturbations of functions. Using this definition, we are ready to define the extremal event DAG.

Definition 3.0.2 (Extremal Event DAG)**.** Let $F = \{f_i : C \to \mathbb{R}\}_{i=1}^k$ be a collection of nicely tame functions. For $i \in [n]$, let $t^i_1 < t^i_2 < \cdots < t^i_{k_i}$ be the domain coordinates for the local extrema of f_i. The *extremal event DAG of* F is the directed graph, $\text{DAG}(F) := (V, E, \omega_V, \omega_E)$, where

- $V := \{v(i,j) \mid i \in [n] \text{ and } j \in [k_i]\}$. In particular $v(i,j) \in V$ corresponds to the extremum of f_i at t^i_j.

- $E := \{(v(i,j), v(r,s)) \mid t_j^i < t_s^r\}$.
- $\omega_V \colon V \to \mathbb{R}_{\geq 0}$ is defined by the node life $\omega_V(v(i,j)) := \frac{1}{2}\text{pers}_{f_i}(t_j^i)$. We call ω_V the *node weights*.
- $\omega_E \colon E \to \mathbb{R}_{\geq 0}$ is defined by $\omega_E(v(i,j), v(r,s)) := \inf\{\varepsilon \mid t_j^i \text{ and } t_s^r \text{ are incomparable}\}$. We call ω_E the *edge weights*.

For a fixed value of $\varepsilon > 0$, we can easily recover ε-DAG(F) from DAG(F) where ε-DAG(F) is defined in [7]. Specifically, ε-DAG(F) is the subgraph of the DAG(F) that consists of vertices and edges with a weight less than or equal to ε. Hence, DAG(F) is a stronger descriptor since it is not dependent on ε, but contains information allowing to recover ε-DAG(F) for all $\varepsilon > 0$.

Computing the vertices and directed edges of the extremal event DAG can be done directly from the graphs of the functions in F. Computing the weights requires more information. Expanding upon earlier observation about node lives, we note that we choose to define the node weights as the node lives for the following reason. If $(t, f(t))$ is a local extremum of a nicely tame function f, then Proposition 2 of [7] states that if $|J_t| > 2\varepsilon$, then every nicely tame ε-perturbation of f has a local extremum of the same type as t contained in $\varphi_\varepsilon^f(t)$. Noting that $|J_t| = \text{pers}_f(t)$, this means that every nicely tame $g \in N_\varepsilon(f)$ has a local extremum of the same type as t, say $t' \in \varphi_\varepsilon^f(t)$ as long as $\varepsilon < \frac{1}{2}\text{pers}_f(t)$. Furthermore, Proposition 1 of [7] states that for any two local extrema at $(s, f(s))$, and $(t, f(t))$ of f of the same type, we have $\varphi_\varepsilon^f(t) \cap \varphi_\varepsilon^f(s) = \emptyset$. Hence, when $\varepsilon < \frac{1}{2}\text{pers}_f(t)$, we guarantee a relative ordering of extrema for ε-perturbations of f.

If $\varepsilon > \frac{1}{2}\text{pers}_f(t)$, Proposition 1 of [7] does not apply and we lose the association between the extrema of the perturbed function of $g \in N_\varepsilon(f)$ and the extremum of f at t.

3.1 Properties of Extremal Intervals

Next, we prove some properties of the extremal intervals that are useful for computing the edge weights. For Lemma 3.1.1, we omit the superscript and subscript f from φ_ε^f and pers_f since f is the only function we are considering.

Lemma 3.1.1 (Properties of $\varphi_\varepsilon(t_i)$)**.** *Let $f : C \to \mathbb{R}$ be a nicely tame function. Let $t_1 < t_2 < ... < t_n$ be the domain coordinates of the local extrema of f. The following statements hold.*

1. *The length $\mathsf{len}(\varphi_\varepsilon^f(t))$ increases as a function of ε.*
2. *For $i < n$, $\varepsilon \leq \frac{1}{2}|f(t_i) - f(t_{i+1})|$ if and only if $t_{i+1} \notin \varphi_\varepsilon(t_i)$ and $t_i \notin \varphi_\varepsilon(t_{i+1})$.*
3. *For $i < n$, if $\varepsilon \leq \frac{1}{2}\min\{pers(t_i), pers(t_{i+1})\}$, then $t_{i+1} \notin \varphi_\varepsilon(t_i)$ and $t_i \notin \varphi_\varepsilon(t_{i+1})$.*

Proof. Let f and $T = \{t_i\}_{i=1}^n$ be defined as in the lemma statement. We prove the three statements for local minima first. Let $i \in [n]$. Assume that $(t_i, f(t_i))$ is a local minimum. Note that, since minimum and maximum alternate in T, we know that t_{i+1} (if it exists) is a local maximum.

Proof of Statement (1) for minima. Consider two values $0 < \varepsilon_0 < \varepsilon_1$. By Lemma 2.4.3, we find

$$\varphi_{\varepsilon_0}(t_i) \subset \varphi_{\varepsilon_1}(t_i).$$

Therefore $\mathsf{len}(\varphi_{\varepsilon_0}(t_i)) \leq \mathsf{len}(\varphi_{\varepsilon_1}(t_i))$. For an illustration of the growth of these intervals, see Figure 3.1.

Proof of Statement (2) for minima. For the forward direction, we assume $i < n$ and $\varepsilon \leq \frac{1}{2}|f(t_i) - f(t_{i+1})|$. Since t_i is a local minimum and t_{i+1} is a local maximum, we have $\varepsilon \leq$

$\frac{1}{2}(f(t_{i+1}) - f(t_i))$, which implies

$$f(t_{i+1}) - \varepsilon \geq f(t_i) + \varepsilon. \tag{3.1}$$

By definition of ε-extremal intervals (Definition 2.4.1) and since t_i is a local minimum, any point $x \in \varphi_\varepsilon(t_i)$ satisfies $f(x) - \varepsilon < f(t_i) + \varepsilon$. Since we already established that $f(t_{i+1}) - \varepsilon \geq f(t_i) + \varepsilon$ in Equation (3.1), we know that $t_{i+1} \notin \varphi_\varepsilon(t_i)$. Similarly, since t_{i+1} is a maximum, for any point $y \in \varphi_\varepsilon(t_{i+1})$, we know that $f(y) + \varepsilon > f(t_{i+1}) - \varepsilon$. Along with Equation (3.1), we conclude $t_i \notin \varphi_\varepsilon(t_{i+1})$.

Next, we prove the backward direction by contrapositive. Assume $i < n$ and $\varepsilon > \frac{1}{2}|f(t_i) - f(t_{i+1})|$. Since t_i is a local minimum and t_{i+1} is a local maximum, we have

$$f(t_{i+1}) - \varepsilon < f(t_i) + \varepsilon.$$

Therefore, $t_{i+1} \in (f-\varepsilon)^{-1}(-\infty, f(t_i)+\varepsilon)$. In order for $t_{i+1} \in \varphi_\varepsilon(t_i)$, we need to show that t_{i+1} is in the connected component of $(f-\varepsilon)^{-1}(-\infty, f(t_i)+\varepsilon)$ containing t_i. Recalling that $\mathsf{left}(*)$ denotes the left endpoint of an interval and since $t_i \in \varphi_\varepsilon(t_i)$, we have $\mathsf{left}(\varphi_\varepsilon(t_i)) < t_i < t_{i+1}$. In addition, since t_i and t_{i+1} are adjacent, $\mathsf{right}(\varphi_\varepsilon(t_i)) > t_{i+1}$. We conclude that $t_{i+1} \in \varphi_\varepsilon(t_i)$. Therefore, Statement (2) holds for minima.

Proof of Statement (3) for minima. Statement (3) follows directly from [7, Proposition 4].

For the case where $(t_i, f(t_i))$ is a local maximum, we substitute $-f$ for f and follow the proofs above. □

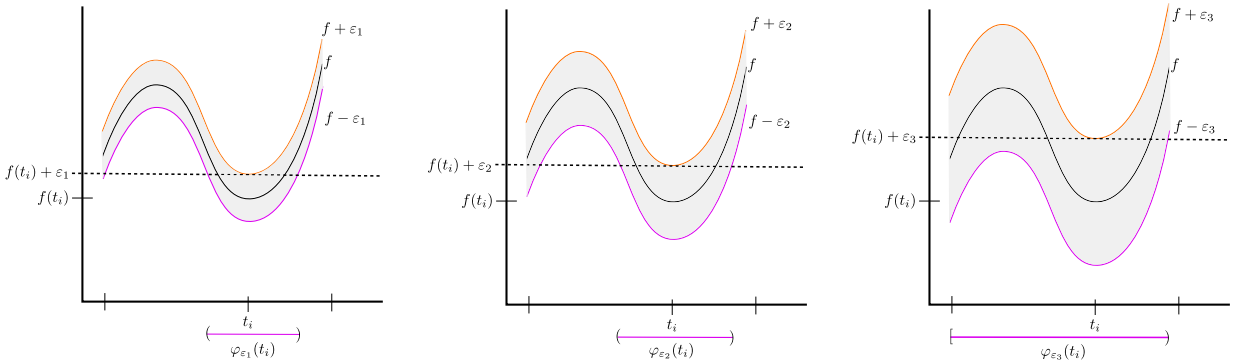


Figure 3.1: Illustrating how $\varphi_\varepsilon(t_i)$ grows with increasing ε by showing (Left) $\varphi_\varepsilon(t_i)$ with ε equal to ε_1, (Middle) ε_2, and (Right) ε_3 for $\varepsilon_1 < \varepsilon_2 < \varepsilon_3$.

3.2 Computing Edge Weights

We state a condition for checking that requirement 3 of Definition 3.0.1 is met. To do this, we need the following lemma.

Lemma 3.2.1 (Comparability of extrema from same function when ε is smaller than node weights)**.** *Let $F = \{f_i : C \to \mathbb{R}\}_{i=1}^n$ be a collection of nicely tame functions where $t_1^i < t_2^i < \cdots < t_{k_i}^i$ are all the domain coordinates of the local extrema of f_i. Let $DAG(F) := (V, E, \omega_V, \omega_E)$ be the extremal event DAG of F. Let $(v(i,j), v(c,d)) \in E$. Suppose $i = c$, $\varepsilon < \min\{\omega_V(v(i,j)), \omega_V(v(c,d))\}$, and $\varphi_\varepsilon^{f_i}(t_j^i) \cap \varphi_\varepsilon^{f_c}(t_d^c) = \emptyset$. Then t_j^i and t_d^c are comparable.*

Proof. Since $i = c$ in this case, we omit the superscripts i and c of t_j^i and t_d^c. Additionally we set $f := f_i$. Furthermore, we omit the subscript and superscript f from the functions pers_f and φ_ε^f.

By Proposition 1 and Theorem 2 of [7], one of t_j or t_d is the domain coordinate of a local maximum while the other is the domain coordinate of a local minimum, and these two extrema are adjacent, i.e. there are no extrema of f between t_j and t_d. Without loss of generality, suppose $(t_j, f(t_j))$ is a local minimum, $(t_d, f(t_d))$ is a local maximum, and $t_j < t_d$.

By way of contradiction, suppose t_j and t_d are incomparable. Since we assume that the first two requirements of Definition 3.0.1 are satisfied, it is the third condition that is violated. Therefore it is not true that $t_j \prec_\varepsilon t_d$ nor it is true that $t_d \prec_\varepsilon t_j$. Since $t_j < t_d$, there exists $g \in N_\varepsilon(f)$ such that for every $t'_j \in \varphi_\varepsilon(t_j)$ that is a domain coordinate of a local minimum of g, and $t'_d \in \varphi_\varepsilon(t_d)$ that is a domain coordinate of a local maximum of g, we have $t'_j > t'_d$. Consider such t'_j and t'_d that are adjacent.

We claim that $g(t'_j) > f(t_d) - \varepsilon$. On the contrary, suppose $g(t'_j) \leq f(t_d) - \varepsilon$. We show there exists a local maximum $(t'_g, g(t'_g))$ such that $t'_g \in \varphi_\varepsilon(t_d)$ and $t'_j < t'_g$, which contradicts the assumption t_j and t_d are incomparable.

Since $g \in N_\varepsilon(f)$, we have $f(t_d) - \varepsilon < g(t_d)$. Hence, $g(t'_j) < g(t_d)$. Suppose $C = [a, b]$ where $a, b \in \mathbb{R}$. We discuss two cases: either $t_d = b$ or $t_d \neq b$. We prove Case 1 that $t_d \neq b$. By definition of ε-extremal intervals, $f(\mathsf{right}(\varphi_\varepsilon(t_d))) + \varepsilon = f(t_d) - \varepsilon$. Since $g \in N_\varepsilon(f)$,

$$g(\mathsf{right}(\varphi_\varepsilon(t_d))) < f(\mathsf{right}(\varphi_\varepsilon(t_d))) + \varepsilon = f(t_d) - \varepsilon.$$

Recall, $f(t_d) - \varepsilon < g(t_d)$. Therefore, $g(\mathsf{right}(\varphi_\varepsilon(t_d))) < g(t_d)$. By Statement (3) of Lemma 3.1.1, we have $t_d \notin \varphi_\varepsilon(t_j)$. Hence, $t'_j < t_d$. Furthermore, by assumption $t'_d < t'_j$ where $t'_d \in \varphi_\varepsilon(t_d)$. Hence, $t'_j > t'_d > \mathsf{left}(\varphi_\varepsilon(t_d))$ and $t'_j < t_d < \mathsf{right}(\varphi_\varepsilon(t_d))$. Thus, $t'_j \in \varphi_\varepsilon(t_d)$. All together we have,

$$t'_j < t_d < \mathsf{right}(\varphi_\varepsilon(t_d)), \quad g(t'_j) < g(t_d), \quad g(t_d) > g(\mathsf{right}(\varphi_\varepsilon(t_d))).$$

This and the assumption g is nicely tame implies there exists a local maximum $(t'_g, g(t'_g))$ such that $t'_g \in (t'_j, \mathsf{right}(\varphi_\varepsilon(t_d)))$. Hence, $t'_g \in \varphi_\varepsilon(t_d)$ and $t'_j < t'_g$.

We prove Case 2 $t_d = b$. Since $g \in N_\varepsilon(f)$, $g(t_d) = g(b) \in (f(b) - \varepsilon, f(b) + \varepsilon)$. By assumption, $g(t'_j) \leq f(b) - \varepsilon < g(b)$. Using the same reasoning as in the case $(t_d \neq b)$, we find $t'_j > \mathsf{left}(\varphi_\varepsilon(b))$ and $t'_j < b$. Altogether we can conclude there exists a local maximum

$t'_g \in (t'_j, b] \subset \varphi_\varepsilon(b)$ of g.

In both cases ($t_d = b$ and $t_d \neq b$) we find there exists a local maximum $(t'_g, g(t'_g))$ such that $t'_g \in \varphi_\varepsilon(t_d)$ and $t'_j < t'_g$. This shows that $t_j \preceq_\varepsilon t_d$, which is a contradiction.

Therefore the claim

$$g(t'_j) > f(t_d) - \varepsilon$$

holds. A similar argument can be used to show

$$g(t'_d) < f(t_j) + \varepsilon.$$

By Statement (3) of Lemma 3.1.1, $t_j \notin \varphi_\varepsilon(t_d)$ since $\varepsilon < \min\{\omega_V(v(i,j)), \omega_V(v(c,d))\} = \frac{1}{2}\min\{\text{pers}(t_j), \text{pers}(t_d)\}$. Applying Statement (2) of Lemma 3.1.1, we get $\varepsilon \leq \frac{1}{2}(f(t_d) - f(t_j))$. Hence,

$$g(t'_j) > f(t_d) - \varepsilon \geq \frac{1}{2}(f(t_d) + f(t_j)).$$

$$g(t'_d) < f(t_j) + \varepsilon \leq \frac{1}{2}(f(t_d) + f(t_j)).$$

This implies that

$$g(t'_d) < g(t'_j).$$

Since $t'_d < t'_j$ and $(t'_d, g(t'_d))$ is a local maximum while $(t'_j, g(t'_j))$ is local minimum of g, there must exist domain coordinates of a local minimum and maximum of g between t'_d and t'_j. Hence, we reach a contradiction with the assumption that t'_d and t'_j are adjacent extrema. Therefore, we conclude that t_j and t_d are comparable.

□

Now we prove a condition for checking that requirement 3 of Definition 3.0.1 is met.

Theorem 3.2.2 (Computing Edge Weights)**.** *Let $F = \{f_i : C \to \mathbb{R}\}_{i=1}^n$ be a collection of nicely tame functions where $t_1^i < t_2^i < \cdots < t_{k_i}^i$ are all the domain coordinates of the*

local extrema of f_i. Let $DAG(F) := (V, E, \omega_V, \omega_E)$ be the extremal event DAG of F. For all edges $(v(i,j), v(c,d)) \in E$, the following statements hold

1. *If $i = c$, then*

$$\omega_E(v(i,j), v(c,d)) = \min\{\omega_V(v(i,j)), \omega_V(v(c,d))\}.$$

2. *If $i \neq c$, then*

$$\omega_E(v(i,j), v(c,d)) = \min\{\omega_V(v(i,j)), \omega_V(v(c,d)), \varepsilon^*(t_j^i, t_d^c)\},$$

where

$$\varepsilon^*(t_j^i, t_d^c) := \inf\{\varepsilon \mid \varphi_\varepsilon^{f_i}(t_j^i) \cap \varphi_\varepsilon^{f_c}(t_d^c) \neq \emptyset\}.$$

Proof. First, we prove Statement (1). Since $i = c$ in this case, we omit the superscripts i and c of t_j^i and t_d^c. Additionally we set $f := f_i$. Furthermore, we omit the subscript and superscript f from the functions pers_f and φ_ε^f.

First, suppose $\varepsilon < \min\{\omega_V(v(i,j)), \omega_V(v(c,d))\}$. We show that t_j and t_d are comparable. We consider two cases.

Suppose $\varphi_\varepsilon(t_j) \cap \varphi_\varepsilon(t_d) = \emptyset$. Without loss of generality, assume $t_j < t_d$. Let $g \in N_\varepsilon(f)$. Using Proposition 2 and Corollary 2 of [7], we get that $\varphi_\varepsilon(t_j)$ and $\varphi_\varepsilon(t_d)$ contain local extrema t_j' and t_d' of the same type as t_j and t_d. Since $\varphi_\varepsilon(t_j) \cap \varphi_\varepsilon(t_d) = \emptyset$, $t_j < t_d$ implies $t_j' < t_d'$. Therefore, t_j and t_d are comparable.

Next suppose $\varphi_\varepsilon(t_j) \cap \varphi_\varepsilon(t_d) \neq \emptyset$. This is the content of Lemma 3.2.1. Altogether we find that if $\varepsilon < \min\{\omega_V(v(i,j), \omega_V(v(c,d))\}$, then t_j and t_d are comparable.

Lastly, we consider the case that $\varepsilon \geq \min\{\omega_V(v(i,j), \omega_V(v(c,d))\}$. By Definition 3.0.1, t_j and t_d are incomparable.

We have shown that t_j, t_d are comparable for all $\varepsilon < \min\{\omega_V(v(i,j)), \omega_V(v(c,d))\}$ and t_j, t_d are incomparable for all $\varepsilon \geq \min\{\omega_V(v(i,j)), \omega_V(v(c,d))\}$. Therefore,

$$\min\{\omega_V(v(i,j)), \omega_V(v(c,d))\} = \inf\{\varepsilon \mid t_j \text{ and } t_d \text{ are incomparable}\}.$$

We conclude $\omega_E(v(i,j), v(c,d)) = \min\{\omega_V(v(i,j)), \omega_V(v(c,d))\}$.

Next, we prove Statement (2). Let $(t_j^i, f_i(t_j^i))$ and $(t_d^c, f_c(t_d^c))$ be local extrema. First, let

$$\varepsilon \leq \min\{\omega_V(v(i,j)), \omega_V(v(c,d)), \varepsilon^*\}.$$

Then, by definition of ε^*, the intervals $\varphi_\varepsilon^{f_i}(t_j^i)$ and $\varphi_\varepsilon^{f_c}(t_d^c)$ are disjoint. Additionally, from Proposition 2 and Corollary 2 of [7], both intervals guarantee existence of local extrema of the appropriate type under any ε-perturbation. Therefore, t_j^i and t_d^c are comparable.

Next, if $\varepsilon^* < \varepsilon \leq \min\{\omega_V(v(i,j)), \omega_V(v(c,d))\}$, then $\varphi_\varepsilon^{f_i}(t_j^i) \cap \varphi_\varepsilon^{f_c}(t_d^c) \neq \emptyset$. Since a local extremum of an ε-perturbation can happen anywhere in $\varphi_\varepsilon(t_j^i)$ and $\varphi_\varepsilon(t_d^c)$, then t_j^i and t_d^c are incomparable at ε.

Lastly, if $\varepsilon \geq \min\{\omega_V(v(i,j)), \omega_V(v(c,d))\}$, then by Definition 3.0.1, t_j^i and t_d^c are incomparable.

We conclude that $\omega_E(v(i,j), v(c,d)) = \min\{\omega_V(v(i,j), \omega_V(v(c,d)), \varepsilon^*(t_j^i, t_d^c)\}$. □

3.3 Example of Extremal Event DAG Construction

Example 3.3.1. We construct the extremal event DAG for $\sin(x) : [0, 2\pi] \to \mathbb{R}$, $\cos(x) : [0, 2\pi] \to \mathbb{R}$ as illustrated in Figure 3.2. First we compute the persistence diagram from a sublevel set filtration of $\sin(x)$, $-\sin(x)$, $\cos(x)$, and $-\cos(x)$. This computes the node lives

of all local extrema in $\sin(x)$ and $\cos(x)$. These node lives are the node weights in the extremal event DAG. Next we compute edge weights between nodes based on Theorem 3.2.2(1).

For an illustration of this, consider the local extrema at $x = 0$ and $x = \frac{\pi}{2}$ of $\sin(x)$. The node lives of these two local extrema are 0.5 and 1 respectively. The edge weight between the two corresponding vertices in the extremal event DAG is the minimum of these two node lives: 0.5.

Computing the edge weights between two vertices corresponding to different function is more involved. We need to apply Theorem 3.2.2(2). To illustrate this, consider the local extrema at $x = \frac{\pi}{2}$ of $\sin(x)$ and $x = \pi$ of $\cos(x)$. Since $\sin(x)$ and $\cos(x)$ are translations of one another, the extremal intervals grow at the same rate for both $\sin(x)$ and $\cos(x)$. We know that $\varphi_\varepsilon^{\sin}(\pi/2)$ and $\varphi_\varepsilon^{\cos}(\pi)$ first intersect at the half-way point of the domain coordinates, which is $\frac{3\pi}{4}$. Using the definition of the ε-extremal intervals we find

$$\begin{aligned} \sin(\pi/2) - \varepsilon &= \sin(3\pi/4) + \varepsilon \\ \varepsilon &= \frac{1}{4}(2 - \sqrt{2}) \approx 0.14 \end{aligned}$$

$$\begin{aligned} \cos(\pi) + \varepsilon &= \cos(3\pi/4) - \varepsilon \\ \varepsilon &= \frac{1}{4}(2 - \sqrt{2}) \approx 0.14 \end{aligned}$$

The epsilon value computed is the infimum ε for which $\varphi_\varepsilon^{\sin}(\pi/2)$ and $\varphi_\varepsilon^{\cos}(\pi)$ both contain $\frac{3\pi}{4}$. Hence this is also the infinum ε for which $\varphi_\varepsilon^{\sin}(\pi/2) \cap \varphi_\varepsilon^{\cos}(\pi) \neq \emptyset$. Since 0.14 is less than the node life of either extrema, then by Theorem 3.2.2(2), the edge weight between the vertices corresponding to $x = \pi/2$ in $\sin(x)$ and $x = \pi$ in $\cos(x)$ is 0.14. Applying a similar process to all edges, we get the extremal event DAG.

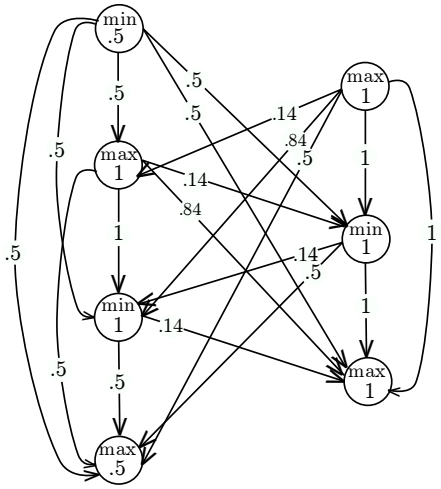


Figure 3.2: Extremal Event DAG for $\sin(x) : [0, 2\pi] \to \mathbb{R}$ and $\cos(x) : [0, 2\pi] \to \mathbb{R}$. The nodes on the left represent the local extrema of $\sin(x)$ while the nodes on the right represent the local extrema of $\cos(x)$. The node weights are the node lives of the corresponding local extrema while the edge weights are computed using Theorem 3.2.2.

CHAPTER FOUR

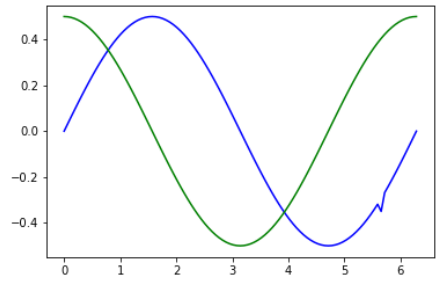
EXTREMAL EVENT DAG DISTANCE

In this section, we define distances between extremal event DAGs representing different collections of time series. We call each such collection a *dataset*. In particular, we first discuss the alignment of nodes in different datasets and then the alignment of edges. We call the result an *extremal event supergraph*. The weights on the nodes and edges of the extremal event supergraph are determined by the weights on the two extremal event DAGs. The distance between the extremal event DAGs is then computed from the weights on the extremal event supergraph.

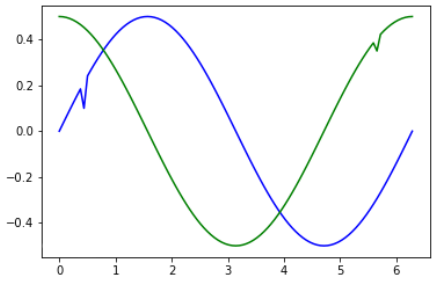
4.1 Backbone Distance

From Figure 3.2, one can see that each time series is translated into an ordered linear sequence of alternating minima and maxima. These linear sequences greatly simplify the comparison between datasets assuming that there is a one-to-one correspondence between the identifications of each of the time series in each dataset. For example, consider two gene expression datasets under different experimental conditions. In this case, each time series has a unique identity corresponding to the gene that it represents. Our primary task in this section is to perform a matching operation between the extrema of two time series with matching identities. To perform the matching, we use a modified version of the DNA alignment algorithm from [59]. Throughout this section, we refer to Figure 4.1 for illustrations of definitions.

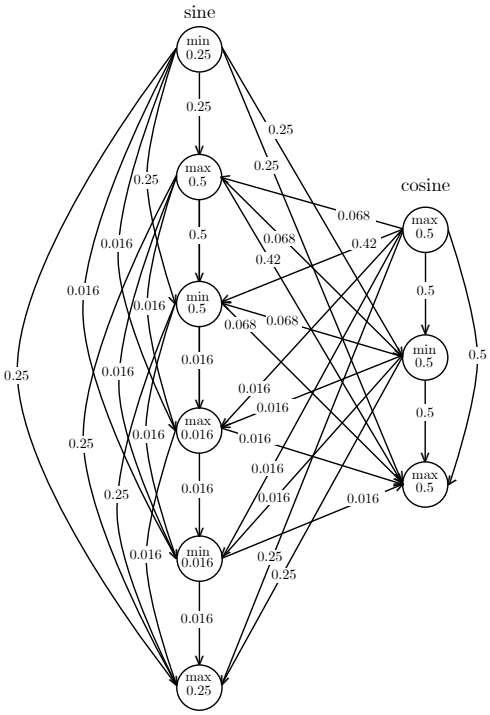
Definition 4.1.1 (Backbones)**.** A *backbone* is a finite sequence $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, where each x_i is a tuple $x_i = (s_i, w_i)$ with s_i a string, and $w_i \in \mathbb{R}_{\geq 0}$. The empty string is denoted by



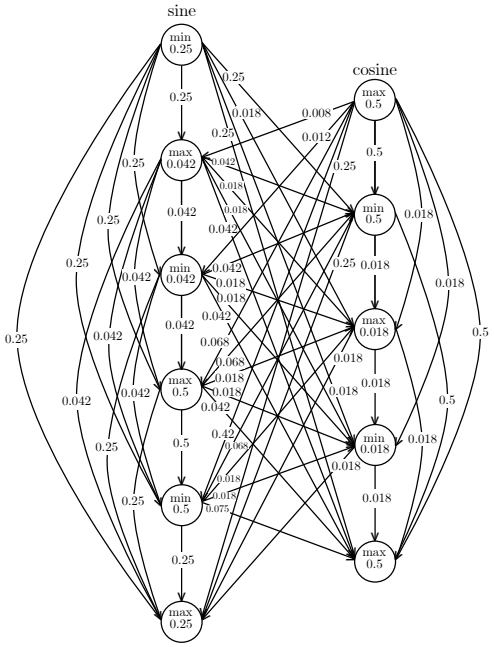
(a) Dataset 1



(b) Dataset 2



(c) DAG1

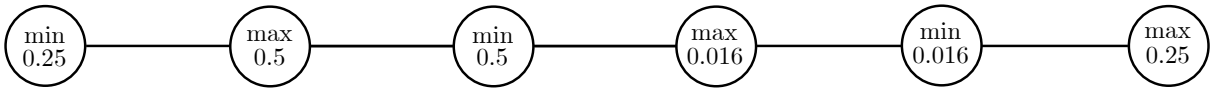


(d) DAG2

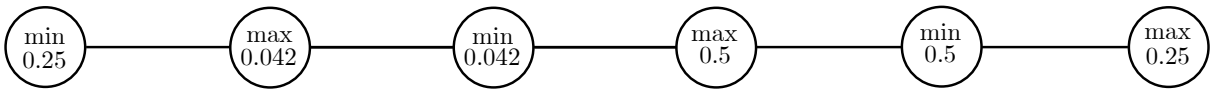
Figure 4.1: Time Series Data and Corresponding Extremal Event DAGs. We consider two datasets consisting of two functions, $\frac{1}{2}\sin(x)$ and $\frac{1}{2}\cos(x)$ over $[0, 2\pi]$ with some added noise. In Figure 4.1a and Figure 4.1b, we label the blue curve as "sine" and green curve as "cosine". Figure 4.1c is the extremal event DAG for Dataset 1 while Figure 4.1d is extremal event DAG for Dataset 2.

0. The *length* of $\mathbf{x}$ is denoted $\mathsf{len}(\mathbf{x})$, and is equal to the number of elements in the sequence (here, $\mathsf{len}(\mathbf{x}) = n$). We call each x_i a *node* and we denote the first k terms of $\mathbf{x}$ by $\mathbf{x}[1:k]$.

Remark 4.1.2 (Constructing Backbones from Nicely Tame Functions). In what follows, we construct a backbone from a nicely tame function $f\colon C \to \mathbb{R}$ by computing $\mathrm{DAG}(\{f\}) = (V, E, \omega_V, \omega_E)$ and removing all edges and edge weights; the nodes are ordered by their corresponding domain coordinates. Then, the data associated to each node $v \in V$ is a string representing which type of local maxima (min or max) along with the node weight $\omega_V(v)$. This backbone for f is denoted $B(f)$. See Figure 4.2 for an example.



(a) Sine Backbone 1



(b) Sine Backbone 2

Figure 4.2: Extracting Sine Backbones from DAG1 and DAG2. Figure 4.2a illustrates the backbone where each node corresponds to local extrema of the sine labeled curve from Dataset 1 (see Figure 4.1a). Mathematically, this backbone is the sequence $((\min, 0.25), (\max, 0.5), (\min, 0.5), (\max, 0.016), (\min, 0.016), (\max, 0.25))$. Figure 4.2b illustrates the backbone where each node corresponds to local extrema of the sine labeled curve from Dataset 2 (see Figure 4.1b). Mathematically, this backbone is the sequence $((\min, 0.25), (\max, 0.042), (\min, 0.042), (\max, 0.5), (\min, 0.5), (\max, 0.25))$.

Remark 4.1.3 (Backbones as Sets). We consider functions over backbones to other spaces. For these settings, we think of a backbone as an ordered multiset, $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, (i.e., repeated elements are allowed) equipped with an injective index function, $\iota_{\mathbf{x}} : \mathbf{x} \to \{1, 2, \ldots, n\}$ where $\iota_{\mathbf{x}}(x_i) = i$. Let $\mathbf{0} := (0, 0)$. We also define $\widetilde{\mathbf{x}} = \{\mathbf{0}\} \cup \mathbf{x}$, where $\mathbf{0}$ is the *empty node.* The function ι is not extended to $\widetilde{\mathbf{x}}$.

Next we discuss alignments and how to compute a distance between two backbones using an optimal alignment.

Definition 4.1.4 (Alignment)**.** Let $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ be backbones. Recall that $[k]$ denotes the first k natural numbers; that is, $[k] := \{1, 2, \ldots, k\} \subset \mathbb{N}$. An *alignment* is a totally ordered correspondence between $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{y}}$ that does not repeat elements of $\mathbf{x}$ or $\mathbf{y}$ and respects the labels (or strings) of the backbones. We say that the number of pairs in the correspondence is the *length* of the alignment. More precisely, we represent an alignment of length k between $\mathbf{x}$ and $\mathbf{y}$ as a function $\alpha : [k] \to \widetilde{\mathbf{x}} \times \widetilde{\mathbf{y}}$, where $\alpha(i)$ can be written as two coordinate functions $\alpha(i) := (\alpha_{\mathbf{x}}(i), \alpha_{\mathbf{y}}(i))$, such that

1. **No Null Alignments.** The pair $(\mathbf{0}, \mathbf{0})$ is not in the image of α, which we denote by $\mathrm{im}(\alpha)$.

2. **Preserves Order of Backbones.** The coordinate functions $\alpha_{\mathbf{x}} : [k] \to \widetilde{\mathbf{x}}$, $\alpha_{\mathbf{y}} : [k] \to \widetilde{\mathbf{y}}$ are *partially monotone*. The function $\alpha_{\mathbf{x}}$ is partially monotone if and only if for every $i, j \in [k]$ such that $\alpha_{\mathbf{x}}(i) \neq \mathbf{0}$ and $\alpha_{\mathbf{x}}(j) \neq \mathbf{0}$, we have

$$\iota_{\mathbf{x}}(\alpha_{\mathbf{x}}(i)) < \iota_{\mathbf{x}}(\alpha_{\mathbf{x}}(j)) \text{ if and only if } i < j.$$

   An analogous definition applies to $\alpha_{\mathbf{y}}$.

3. **No Misalignments.** For each $((s_x, w_x), (s_y, w_y)) \in \mathrm{im}(\alpha)$, we either have equality in strings $s_x = s_y$, or one of (s_x, w_x), (s_y, w_y) is equal to $\mathbf{0}$.

4. **Restriction to Matching.** Each element of $\mathbf{x}$ and $\mathbf{y}$ appears in the image of $\alpha_{\mathbf{x}}$ and $\alpha_{\mathbf{y}}$ exactly once. That is, for each $x_i \in \mathbf{x}$, there exists exactly one $j \in [k]$ for which $\alpha_{\mathbf{x}}(j) = x_i$. The analogous statement holds for each $y_i \in \mathbf{y}$.

If $\alpha(i) = (\alpha_{\mathbf{x}}(i), \mathbf{0})$, we say that $\alpha_{\mathbf{x}}(i)$ is aligned with an *insertion*; similarly for $\alpha(i) = (\mathbf{0}, \alpha_{\mathbf{y}}(i))$. We denote the restriction of α to the first h integers, $[h] = \{1, 2, \ldots, h\} \subset [k]$ as $\alpha[1:h]$.

Notation 4.1.5 (Elements in $\mathrm{im}(\alpha)$)**.** When we use the notation $(x, y) \in \mathrm{im}(\alpha)$, we always assume that $x \neq \mathbf{0}$ and $y \neq \mathbf{0}$. We also use notation $(x, \mathbf{0}) \in \mathrm{im}(\alpha)$ and $(\mathbf{0}, y) \in \mathrm{im}(\alpha)$ to denote that x or y is aligned with an insertion.

Note that the restriction of $\mathrm{im}(\alpha) \cap (\mathbf{x} \times \mathbf{y})$ is a partial matching (that is, each element in $\mathbf{x} \times \mathbf{y}$, if not aligned with an insertion, is aligned with a distinct element of the other backbone). We call any pair $(x, y) \in \mathrm{im}(\alpha) \cap (\mathbf{x} \times \mathbf{y})$ a *nontrivial match*. An example of two different alignments of the sine backbones shown in Figure 4.2 is given in Figure 4.3. Figure 4.3a is aligned without insertions, while Figure 4.3b has two insertions in each of the backbones. Notice that the insertions occur at the small noisy extrema in each of the time series.

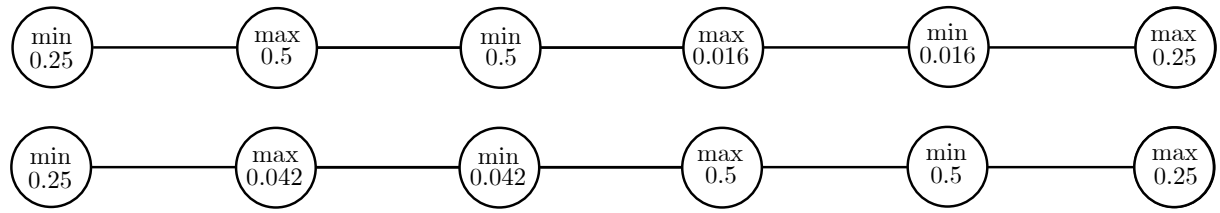
Definition 4.1.6 (Cost of Alignment)**.** Let $\mathbf{x}$ and $\mathbf{y}$ be backbones and $\alpha : [k] \to \widetilde{\mathbf{x}} \times \widetilde{\mathbf{y}}$ be an alignment of length k. The *cost* of α is defined as

$$\mathbf{cost}(\alpha) := \sum_{(x,y) \in \mathrm{im}(\alpha)} |w_x - w_y|,$$

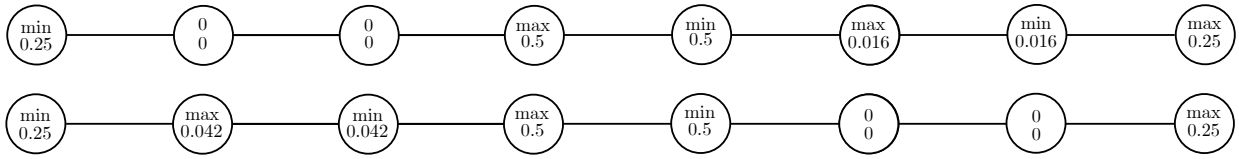
where $x = (s_x, w_x)$ and $y = (s_y, w_y)$. We define the cost of the partial alignment $c_{\mathbf{x},\mathbf{y}}(i, j)$ to be the minimum cost of aligning $\mathbf{x}[1:i]$ with $\mathbf{y}[1:j]$, that is,

$$c_{\mathbf{x},\mathbf{y}}(i, j) := \min\{\mathbf{cost}(\alpha) \mid \alpha \text{ is an alignment of } \mathbf{x}[1:i] \text{ and } \mathbf{y}[1:j]\}.$$

Referring again to Figure 4.3, it is easy to compute that the alignment in Figure 4.3b has a lower cost, 0.116, than that in Figure 4.3a, 0.932.



(a) Alignment 1



(b) Alignment 2

Figure 4.3: Two Possible Alignments of Sine Backbones. We consider the backbones shown in Figure 4.2. Call these $\mathbf{x}$ and $\mathbf{y}$ respectively. Figure 4.3a gives an alignment, $\alpha_1 : \{1, 2, \ldots, 6\} \to \widetilde{\mathbf{x}} \times \widetilde{\mathbf{y}}$ of the two backbones where $\alpha_1(i) = (x_i, y_i)$. Figure 4.3b gives an alignment $\alpha_2 : \{1, 2, \ldots, 8\} \to \widetilde{\mathbf{x}} \times \widetilde{\mathbf{y}}$ where $\alpha_2(1) = (x_1, y_1)$, $\alpha_2(2) = (\mathbf{0}, y_2)$, $\alpha_2(3) = (\mathbf{0}, y_3)$, $\alpha_2(4) = (x_2, y_4)$, $\alpha_2(5) = (x_3, y_5)$, $\alpha_2(6) = (x_4, \mathbf{0})$, $\alpha_2(7) = (x_5, \mathbf{0})$, and $\alpha_2(8) = (x_6, y_6)$.

Definition 4.1.7 (Optimal Alignment)**.** Let $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ be backbones. We call an alignment $\alpha : [k] \to \widetilde{\mathbf{x}} \times \widetilde{\mathbf{y}}$ *optimal* if $\mathbf{cost}(\alpha) = c_{\mathbf{x},\mathbf{y}}(m, n)$.

An optimal alignment minimizes cost. We note that there could be multiple alignments that minimize cost and so an optimal alignment is not necessarily unique.

We define the distance between two backbones $\mathbf{x}$ and $\mathbf{y}$ using an optimal alignment. To do this, we need to identify nontrivial matches, i.e., those alignment pairs that do not involve insertions.

Definition 4.1.8 (Backbone Distance)**.** The *backbone distance* between backbones $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$d_{\mathcal{B}}(\mathbf{x}, \mathbf{y}) = \inf_{\alpha} \left(\sum_{(x,y)\in \mathrm{im}(\alpha)} |w_x - w_y| + \sum_{(x,\mathbf{0})\in \mathrm{im}(\alpha)} w_x + \sum_{(\mathbf{0},y)\in \mathrm{im}(\alpha)} w_y \right) \tag{4.1}$$

where α ranges over all alignments between $\mathbf{x}$ and $\mathbf{y}$.

The backbone distance finds the best alignment between $\mathbf{x}$ and $\mathbf{y}$, then defines the distance to be the L_1-norm between a vector consisting of the node weights in $\widetilde{\mathbf{x}}$ and a vector consisting of the matching node weights in $\widetilde{\mathbf{y}}$. The first term of Equation 4.1 accounts for the cost of the nodes in $\mathbf{x}$ that are aligned with nodes in $\mathbf{y}$, the second term accounts for the cost of the nodes in $\mathbf{x}$ that are aligned with an insertion, and the third term accounts for the cost of the nodes in $\mathbf{y}$ that are aligned with an insertion.

We show that this distance is in fact a metric. In order to show the triangle inequality, $d_{\mathcal{B}}(\mathbf{x}, \mathbf{z}) \leq d_{\mathcal{B}}(\mathbf{x}, \mathbf{y}) + d_{\mathcal{B}}(\mathbf{y}, \mathbf{z})$, we need a way of composing an alignment between $\mathbf{x}$ and $\mathbf{y}$ with an alignment between $\mathbf{y}$ and $\mathbf{z}$. This is the content of Construction 4.1.9 and Lemma 4.1.10.

Construction 4.1.9 (Composition of Alignments)**.** Let $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ be backbones and $\alpha_1 : [k] \to \widetilde{\mathbf{x}} \times \widetilde{\mathbf{y}}$, $\alpha_2 : [l] \to \widetilde{\mathbf{y}} \times \widetilde{\mathbf{z}}$ be alignment maps. The *composition of* α_1 *and* α_2 induces an ordered

correspondence between $\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{z}}$ whose nontrivial pairs are given by

$$A := \{(x, z) \mid \exists y \in \mathbf{y} \text{ s.t. } (x, y) \in \mathrm{im}(\alpha_1) \text{ and } (y, z) \in \mathrm{im}(\alpha_2)\}.$$

The pairs in A are ordered using the order given in $\mathbf{x}$. The set A, however does not form a complete alignment of $\mathbf{x}$ and $\mathbf{z}$ because $x_i \in \mathbf{x}$ with $(x_i, \mathbf{0}) \in \mathrm{im}(\alpha_1)$ and $z_j \in \mathbf{z}$ with $(\mathbf{0}, z_j) \in \mathrm{im}(\alpha_2)$ are not accounted for in pairs in A. We include all such pairs $(x_i, \mathbf{0})$ and $(\mathbf{0}, z_j)$ to construct a function $\alpha_2 \circ \alpha_1 : [s] \to \widetilde{\mathbf{x}} \times \widetilde{\mathbf{z}}$ in such a way that each pair satisfies

(a) If $\iota_{\mathbf{x}}(x_i) < \iota_{\mathbf{x}}(x)$ ($\iota_{\mathbf{x}}(x) < \iota_{\mathbf{x}}(x_j)$) and $(x_i, z_i) \in A$ ($(x_j, z_j) \in A$) then for

$$\alpha_2 \circ \alpha_1(s_1) = (x_i, z_i), \quad \alpha_2 \circ \alpha_1(s_2) = (x, \mathbf{0}), \quad \alpha_2 \circ \alpha_1(s_3)(x_j, z_j)$$

we have the order $s_1 < s_2$ ($s_2 < s_3$).

(b) If $\iota_{\mathbf{z}}(z_i) < \iota_{\mathbf{z}}(z)$ ($\iota_{\mathbf{z}}(z) < \iota_{\mathbf{z}}(z_j)$) and $(x_i, z_i) \in A$ ($(x_j, z_j) \in A$) then for

$$\alpha_2 \circ \alpha_1(s_1) = (x_i, z_i), \quad \alpha_2 \circ \alpha_1(s_2) = (\mathbf{0}, z), \quad \alpha_2 \circ \alpha_1(s_3)(x_j, z_j)$$

we have the order $s_1 < s_2$ ($s_2 < s_3$).

An extension of A to an alignment $\alpha_2 \circ \alpha_1 : [s] \to \widetilde{\mathbf{x}} \times \widetilde{\mathbf{z}}$ exists, but also that it may not be unique since there may be multiple options on where to place consecutive insertions. We resolve this ambiguity arbitrarily in the following way. If there is a set of consecutive insertions $\{(x_{i_1}, \mathbf{0}), (x_{i_2}, \mathbf{0}), \ldots, (x_{i_k}, \mathbf{0}), (\mathbf{0}, z_{j_1}), \ldots, (\mathbf{0}, z_{j_s})\}$ between a pair (x_i, z_i) and a pair (x_j, z_j), then we order them starting with the insertions $\{(x_{i_r}, \mathbf{0})\}_{r=1}^{k}$ ordered by the order given in $\mathbf{x}$, followed by the insertions $\{(\mathbf{0}, z_{j_r})\}_{r=1}^{s}$ insertions ordered by the order given in $\mathbf{z}$. We hasted to point out that the following results (Lemma 4.1.10, Lemma 4.1.11, and Lemma 5.1.2) hold for $\alpha_2 \circ \alpha_1$ defined using any choice of order of such consecutive insertions.

The ordered correspondence $\alpha_2 \circ \alpha_1$ constructed in Construction 4.1.9 is, in fact, an alignment:

Lemma 4.1.10 (Composition is an Alignment)**.** *Let $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ be backbones and $\alpha_1 : [k] \to \widetilde{\boldsymbol{x}} \times \widetilde{\boldsymbol{y}}$, $\alpha_2 : [l] \to \widetilde{\boldsymbol{y}} \times \widetilde{\boldsymbol{z}}$ be alignment maps. Then, $\alpha_2 \circ \alpha_1$ defined in Construction 4.1.9, is an alignment between $\boldsymbol{x}$ and $\boldsymbol{z}$.*

Proof. We show that all properties of Definition 4.1.4 hold. Since α_1, α_2 are alignments, they contain no null alignments or misalignments. This implies that $\alpha_2 \circ \alpha_1$ also contains no null alignments or misalignments. Furthermore, $\mathrm{im}(\alpha_2 \circ \alpha_1)$ contains all $x \in \mathbf{x}$ that are nontrivially aligned with $y \in \mathbf{y}$ along with all $x \in \mathbf{x}$ that are aligned with an insertion. Hence all $x \in \mathbf{x}$ appear in the image of $(\alpha_2 \circ \alpha_1)_{\mathbf{x}}$ exactly once. The same can be said for all $z \in \mathbf{z}$. Therefore, Property (1), Property (3), and Property (4) of Definition 4.1.4 hold.

Lastly, we show that $\alpha_2 \circ \alpha_1$ preserves order of the backbones $\mathbf{x}$ and $\mathbf{z}$. First consider the set of nontrivial pairs A. We show the order of the (partial) backbones $\mathbf{x}$ and $\mathbf{z}$ is preserved in this set. By construction, we know the order of $\mathbf{x}$ is preserved. Replacing each pair $(x, z) \in A$ by $(x, y) \in \mathrm{im}(\alpha_1)$ where $(y, z) \in \mathrm{im}(\alpha_2)$, we see the order of $\mathbf{y}$ is preserved because α_1 is an alignment. Next, replacing each of these pairs with (y, z) where $(y, z) \in \mathrm{im}(\alpha_2)$, we see that order $\mathbf{z}$ is preserved in A because α_2 is an alignment. Hence, the order of the backbones $\mathbf{x}$ and $\mathbf{z}$ are preserved in A. By Construction 4.1.9, each remaining trivial pair is added to the set A so that the order of the backbones $\mathbf{x}$ and $\mathbf{z}$ are preserved. Thus, Property (2) of Definition 4.1.4 also holds and we have an alignment $\alpha_2 \circ \alpha_1$ between $\mathbf{x}$ and $\mathbf{z}$. □

Next, we prove that the backbone distance satisfies the triangle inequality. We consider three backbones $\mathbf{x}, \mathbf{y}$, and $\mathbf{z}$. We use optimal alignments between $\mathbf{x}$ and $\mathbf{y}$, and $\mathbf{y}$ and $\mathbf{z}$, to construct an alignment between $\mathbf{x}$ and $\mathbf{z}$. We show that the constructed alignment between $\mathbf{x}$ and $\mathbf{z}$ satisfies the triangle inequality. Since the alignment between $\mathbf{x}$ and $\mathbf{z}$ that we find

is an upper bound of an optimal alignment between $\mathbf{x}$ and $\mathbf{z}$, the backbone distance must also satisfy the triangle inequality.

Lemma 4.1.11 (Backbone Distance Satisfies Triangle Inequality)**.** *Let $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ be backbones. Then,*

$$d_{\mathcal{B}}(\boldsymbol{x}, \boldsymbol{z}) \leq d_{\mathcal{B}}(\boldsymbol{x}, \boldsymbol{y}) + d_{\mathcal{B}}(\boldsymbol{y}, \boldsymbol{z}).$$

Proof. Let $\alpha_1 : [k] \to \widetilde{\mathbf{x}} \times \widetilde{\mathbf{y}}$, $\alpha_2 : [m] \to \widetilde{\mathbf{y}} \times \widetilde{\mathbf{z}}$ be optimal alignments. Consider the composition alignment $\alpha_2 \circ \alpha_1$ from Construction 4.1.9. Define A as in Construction 4.1.9,

$$A_{\mathbf{x}} := \{(x, y) \in \mathrm{im}(\alpha_1) \mid \exists z \in \mathbf{z} \text{ s.t. } (y, z) \in \mathrm{im}(\alpha_2)\},$$

and

$$A_{\mathbf{z}} := \{(y, z) \in \mathrm{im}(\alpha_2) \mid \exists x \in \mathbf{x} \text{ s.t. } (x, y) \in \mathrm{im}(\alpha_1)\}.$$

We start by considering and justifying all the relations we need in the triangle inequality computation. Since $d_{\mathcal{B}}(\mathbf{x}, \mathbf{z})$ is computed using an optimal alignment between $\mathbf{x}$ and $\mathbf{z}$ and $\alpha_2 \circ \alpha_1$ is one alignment,

$$d_{\mathcal{B}}(\mathbf{x}, \mathbf{z}) \leq \sum_{(x,z) \in \mathrm{im}(\alpha_2 \circ \alpha_1)} |w_x - w_z| + \sum_{(x, \mathbf{0}) \in \mathrm{im}(\alpha_2 \circ \alpha_1)} w_x + \sum_{(\mathbf{0}, z) \in \mathrm{im}(\alpha_2 \circ \alpha_1)} w_z \tag{4.2}$$

We apply the L_1-norm triangle inequality to the first term in (4.2) to get

$$\sum_{(x,z) \in \mathrm{im}(\alpha_2 \circ \alpha_1)} |w_x - w_z| \leq \sum_{(x,y) \in A_{\mathbf{x}}} |w_x - w_y| + \sum_{(y,z) \in A_{\mathbf{z}}} |w_y - w_z|. \tag{4.3}$$

Now we discuss the second term in (4.2). Define

$$X_{(y, \mathbf{0})} := \{(x, \mathbf{0}) \in \mathrm{im}(\alpha_2 \circ \alpha_1) \mid \exists y \in \mathbf{y} \text{ s.t. } (x, y) \in \mathrm{im}(\alpha_1) \setminus A_{\mathbf{x}}\}.$$

Observe, if $(x, y) \in \mathrm{im}(\alpha_1) \setminus A_{\mathbf{x}}$, then for all $z \in \mathbf{z}$, $(y, z) \notin \mathrm{im}(\alpha_2)$. Hence, $(y, \mathbf{0}) \in \mathrm{im}(\alpha_2)$. This implies the set

$$(\mathbf{x} \times \{\mathbf{0}\}) \cap \mathrm{im}(\alpha_2 \circ \alpha_1) = ((\mathbf{x} \times \{\mathbf{0}\}) \cap \mathrm{im}(\alpha_1)) \cup X_{(y,\mathbf{0})}$$

and this union is disjoint. Thus,

$$\sum_{(x,\mathbf{0}) \in \mathrm{im}(\alpha_2 \circ \alpha_1)} w_x = \sum_{(x,\mathbf{0}) \in \mathrm{im}(\alpha_1)} w_x + \sum_{(x,\mathbf{0}) \in X_{(y,\mathbf{0})}} w_x. \tag{4.4}$$

By definition of $X_{(y,\mathbf{0})}$, for each $(x, \mathbf{0}) \in X_{(y,\mathbf{0})}$, there exists $y \in \mathbf{y}$ such that $(x, y) \in \mathrm{im}(\alpha_1) \setminus A_{\mathbf{x}}$ and, at the same time, $(y, \mathbf{0}) \in \mathrm{im}(\alpha_2)$. Noting this observation and applying the triangle inequality from the L_1-norm to the last term of Equation (4.4), gives

$$\begin{aligned} \sum_{(x,\mathbf{0}) \in \mathrm{im}(\alpha_2 \circ \alpha_1)} w_x \leq \sum_{(x,\mathbf{0}) \in \mathrm{im}(\alpha_1)} w_x + \sum_{\mathrm{im}(\alpha_1) \setminus A_{\mathbf{x}}} |w_x - w_y| \\ + \sum_{(y,\mathbf{0}) \in \mathrm{im}(\alpha_2) \text{ s.t. } (x,y) \in \mathrm{im}(\alpha_1)} w_y. \end{aligned} \tag{4.5}$$

Now we discuss the third term in (4.2). Define

$$Z_{(\mathbf{0},y)} := \{(\mathbf{0}, z) \in \mathrm{im}(\alpha_2 \circ \alpha_1) \mid \exists y \in \mathbf{y} \text{ s.t. } (y, z) \in \mathrm{im}(\alpha_2) \setminus A_{\mathbf{z}}\}.$$

Similarly, the set

$$(\{\mathbf{0}\} \times \mathbf{z}) \cap \mathrm{im}(\alpha_2 \circ \alpha_1) = ((\{\mathbf{0}\} \times \mathbf{z}) \cap \mathrm{im}(\alpha_2)) \cup Z_{(\mathbf{0},y)}.$$

This implies

$$\sum_{(\mathbf{0},z)\in \mathrm{im}(\alpha_2\circ\alpha_1)} w_z = \sum_{(\mathbf{0},z)\in \mathrm{im}(\alpha_2)} w_z + \sum_{(\mathbf{0},z)\in Z_{(\mathbf{0},y)}} w_z. \tag{4.6}$$

By definition, for each $(\mathbf{0}, z) \in Z_{(\mathbf{0},y)}$, there exists $y \in \mathbf{y}$ such that $(y, z) \in \mathrm{im}(\alpha_2) \setminus A_{\mathbf{z}}$ and $(\mathbf{0}, y) \in \mathrm{im}(\alpha_1)$. Applying the triangle inequality from the L_1-norm to the last term of Equation (4.6), we get

$$\begin{aligned}\sum_{(\mathbf{0},z)\in \mathrm{im}(\alpha_2\circ\alpha_1)} w_z \leq \sum_{(\mathbf{0},z)\in \mathrm{im}(\alpha_2)} w_z + \sum_{(y,z)\in \mathrm{im}(\alpha_2)\setminus A_{\mathbf{z}}} |w_z - w_y| \\ + \sum_{(\mathbf{0},y)\in \mathrm{im}(\alpha_1) \text{ s.t. } (y,z)\in \mathrm{im}(\alpha_2)} w_y.\end{aligned} \tag{4.7}$$

We derive two additional sets of relationships that will be used in the final estimate. We first note that since $\mathrm{im}(\alpha_1) \cap (\mathbf{x} \times \mathbf{y}) = A_{\mathbf{x}} \cup ((\mathrm{im}(\alpha_1) \cap (\mathbf{x} \times \mathbf{y})) \setminus A_{\mathbf{x}})$ and this union is disjoint,

$$\sum_{(x,y)\in A_{\mathbf{x}}} |w_x - w_y| + \sum_{(x,y)\in \mathrm{im}(\alpha_1)\setminus A_{\mathbf{x}}} |w_x - w_y| = \sum_{(x,y)\in \mathrm{im}(\alpha_1)} |w_x - w_y|. \tag{4.8}$$

Analogously, since $\mathrm{im}(\alpha_2) \cap (\mathbf{y} \times \mathbf{z}) = A_{\mathbf{z}} \cup ((\mathrm{im}(\alpha_2) \cap (\mathbf{y} \times \mathbf{z})) \setminus A_{\mathbf{z}})$ and this union is disjoint. Thus,

$$\sum_{(y,z)\in A_{\mathbf{z}}} |w_y - w_z| + \sum_{(y,z)\in \mathrm{im}(\alpha_2)\setminus A_{\mathbf{z}}} |w_y - w_z| = \sum_{(y,z)\in \mathrm{im}(\alpha_2)} |w_y - w_z|. \tag{4.9}$$

The second set of relationships are inequalities. First, notice if $(y, z) \in \mathrm{im}(\alpha_2) \setminus A_{\mathbf{z}}$, then y must align with an empty node in the alignment α_1. Hence, $\{(\mathbf{0}, y) \mid (y, z) \in \mathrm{im}(\alpha_2) \setminus A_{\mathbf{z}}\} \subset \{(\mathbf{0}, y) \in \mathrm{im}(\alpha_1)\}$. This implies

$$\sum_{(y,\mathbf{0}) \text{ s.t. } (y,z)\in \mathrm{im}(\alpha_2)\setminus A_{\mathbf{z}}} w_y \leq \sum_{(\mathbf{0},y)\in \mathrm{im}(\alpha_1)} w_y. \tag{4.10}$$

Finally, we note that $\{(y, \mathbf{0}) \in \mathrm{im}(\alpha_2) \mid (x, y) \in \mathrm{im}(\alpha_1)\} \subset \mathrm{im}(\alpha_2) \cap (\mathbf{y} \times \{\mathbf{0}\})$. Therefore,

$$\sum_{(y,\mathbf{0})\in\mathrm{im}(\alpha_2) \text{ s.t. } (x,y)\in\mathrm{im}(\alpha_1)} w_y \leq \sum_{(y,\mathbf{0})\in\mathrm{im}(\alpha_2)} w_y \tag{4.11}$$

Now we can put all these relations together to prove the backbone distance satisfies the triangle inequality.

$$
\begin{aligned}
d_{\mathcal{B}}(\mathbf{x},\mathbf{z}) \leq & \sum_{(x,z)\in \mathrm{im}(\alpha_2\circ\alpha_1)} |w_x - w_z| + \sum_{(x,\mathbf{0})\in \mathrm{im}(\alpha_2\circ\alpha_1)} w_x + \sum_{(\mathbf{0},z)\in \mathrm{im}(\alpha_2\circ\alpha_1)} w_z \\
& \text{by Equation (4.2)} \\
\leq & \sum_{(x,y)\in A_{\mathbf{x}}} |w_x - w_y| + \sum_{(y,z)\in A_{\mathbf{z}}} |w_y - w_z| + \sum_{(x,\mathbf{0})\in \mathrm{im}(\alpha_2\circ\alpha_1)} w_x \\
& + \sum_{(\mathbf{0},z)\in \mathrm{im}(\alpha_2\circ\alpha_1)} w_z \text{ by Equation (4.3)} \\
\leq & \sum_{(x,y)\in A_{\mathbf{x}}} |w_x - w_y| + \sum_{(x,y)\in \mathrm{im}(\alpha_1)\setminus A_{\mathbf{x}}} |w_x - w_y| \\
& + \sum_{(x,\mathbf{0})\in \mathrm{im}(\alpha_1)} w_x + \sum_{(y,\mathbf{0})\in \mathrm{im}(\alpha_2) \text{ s.t. } (x,y)\in \mathrm{im}(\alpha_1)} w_y \\
& + \sum_{(y,z)\in A_{\mathbf{z}}} |w_y - w_z| + \sum_{(y,z)\in \mathrm{im}(\alpha_2)\setminus A_{\mathbf{z}}} |w_z - w_y| \\
& + \sum_{(\mathbf{0},y)\in \mathrm{im}(\alpha_1) \text{ s.t. } (y,z)\in \mathrm{im}(\alpha_2)} w_y + \sum_{(\mathbf{0},z)\in \mathrm{im}(\alpha_2)} w_z \\
& \text{by Equations (4.5) and (4.7)} \\
\leq & \sum_{(x,y)\in \mathrm{im}(\alpha_1)} |w_x - w_y| + \sum_{(x,\mathbf{0})\in \mathrm{im}(\alpha_1)} w_x + \sum_{(\mathbf{0},y)\in \mathrm{im}(\alpha_1)} w_y \\
& + \sum_{(y,z)\in \mathrm{im}(\alpha_2)} |w_y - w_z| + \sum_{(y,\mathbf{0})\in \mathrm{im}(\alpha_2)} w_y + \sum_{(\mathbf{0},z)\in \mathrm{im}(\alpha_2)} w_z \\
& \text{by Equations (4.8), (4.10), (4.9), and (4.11)} \\
= & \, d_{\mathcal{B}}(\mathbf{x},\mathbf{y}) + d_{\mathcal{B}}(\mathbf{y},\mathbf{z}).
\end{aligned}
$$

Hence, $d_{\mathcal{B}}(\mathbf{x},\mathbf{z}) \leq d_{\mathcal{B}}(\mathbf{x},\mathbf{y}) + d_{\mathcal{B}}(\mathbf{y},\mathbf{z})$.

□

We can now prove the backbone distance is a metric.

Proposition 4.1.12 (Backbone Distance is a Metric)**.** *The backbone distance (Defini-*

tion 4.1.8) is a metric.

Proof. Let $\mathbf{x}$, $\mathbf{y}$ be backbones. Recall that for each $x \in \mathbf{x}$, we can write $x = (s_x, w_x)$; likewise for $y \in \mathbf{y}$. Let $\alpha : [k] \to \widetilde{\mathbf{x}} \times \widetilde{\mathbf{y}}$ be an optimal alignment. We verify all properties of a metric.

Non-Negativity. Since all node weights are non-negative, $d_{\mathcal{B}}(\mathbf{x}, \mathbf{y}) \geq 0$.

Symmetry. By construction, $d_{\mathcal{B}}$ is symmetric; see (4.1).

Definiteness. If $\mathbf{x} = \mathbf{y}$, then the optimal alignment aligns each node with itself, and there are no insertions. Hence, all node weights match and $d_{\mathcal{B}}(\mathbf{x}, \mathbf{y}) = 0$.

On the other hand, assume $d_{\mathcal{B}}(\mathbf{x}, \mathbf{y}) = 0$. This implies

$$0 = \sum_{(x,y)\in \mathrm{im}(\alpha)} |w_x - w_y| + \sum_{(x,\mathbf{0})\in \mathrm{im}(\alpha)} w_x + \sum_{(\mathbf{0},y)\in \mathrm{im}(\alpha)} w_y.$$

Since all weights are non-negative, the latter two summands (corresponding to nodes aligned with insertions) sum to zero. Since there are no null alignments (Definition 4.1.4 Property (1)), every node in $\mathbf{x}$ is aligned with a node in $\mathbf{y}$. We have

$$0 = \sum_{(x,y)\in \mathrm{im}(\alpha)} |w_x - w_y|$$

and so $w_x = w_y$ for all nontrivial pairs (x, y). Additionally, if $(x, y) \in \mathrm{im}(\alpha)$, we know that $s_x = s_y$. Since this is true for all nodes, we must have $\mathbf{x} = \mathbf{y}$.

Triangle Inequality. By Lemma 4.1.11, the triangle inequality holds.

Therefore, the backbone distance is a metric.

□

4.2 Extremal Event DAG Distance

Using the backbone distance, we define a distance between two extremal event DAGs, D and D', constructed from comparable datasets. Comparable here means that the number

and identity of the time series are the same between the two datasets so that the choice of which backbones to align is clear. Once the alignments have been computed, we construct a supergraph based on D and D' where there is a vertex for each node pair from each alignment. We add a directed edge between two vertices if the edge exists between the two vertices in either D or D'. After we construct the supergraph, we impose two weight functions on the vertices and nodes given by the weights of the nodes and edges in D and D' respectively. The difference between these weight vectors is the extremal event DAG distance between D and D'.

Definition 4.2.1 (Extremal Event Supergraph)**.** Let $D = (V, E, \omega_V, \omega_E)$ and $D' = (V', E', \omega'_V, \omega'_E)$ be two extremal event DAGs with n pairs of aligned backbones. Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ be the backbones of D, and $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$ be the backbones of D' and, for each $i \in [k]$, let $\alpha^{(i)} : [k_i] \to \widetilde{\mathbf{x}}_i \times \widetilde{\mathbf{y}}_i$ be the corresponding alignments where $k_i = \mathsf{len}(\alpha^i)$. Just as we expanded α to two coordinate functions in Definition 4.1.4, $\alpha^{(i)}$ can be expanded into two coordinate functions $\alpha^{(i)} := (\alpha_{\mathbf{x}_i}, \alpha_{\mathbf{y}_i})$. The *extremal event supergraph determined by the alignments* $\{\alpha^{(i)}\}_{i=1}^n$ of D and D' is a doubly weighted directed graph $(V_\alpha, E_\alpha, \omega_\alpha, \omega'_\alpha)$, where

- $V_\alpha := \{v(i,j) \mid i \in [n], j \in [k_i]\}$. That is, the vertices of V_α are in one-to-one correspondence with each element of every alignment. Note $V \cup V' \subset V_\alpha$.
- An ordered pair of vertices $(v(i,j), v(k,l)) \in V_\alpha \times V_\alpha$ is a directed edge in E_α if and only if either one or both of the following is true
  - $(\alpha_{\mathbf{x}_i}(j), \alpha_{\mathbf{x}_k}(l)) \in E$
  - $(\alpha_{\mathbf{y}_i}(j), \alpha_{\mathbf{y}_k}(l)) \in E'$.

  Note $E \cup E' \subset E_\alpha$.

- The weight function $\omega_\alpha : V_\alpha \cup E_\alpha \to \mathbb{R}_{\geq 0}$ is defined by

$$\omega_\alpha(x) = \begin{cases} \omega_V(v(i,j)), & v(i,j) \in V_\alpha \\ \omega_E(v(i,j), v(k,l)), & x = (v(i,j), v(k,l)) \in E \subseteq E_\alpha \\ 0 & \text{otherwise.} \end{cases}$$

- The weight function $\omega'_\alpha : V_\alpha \cup E_\alpha \to \mathbb{R}_{\geq 0}$ is defined by

$$\omega'_\alpha(x) = \begin{cases} \omega_V(v((i,j)), & v(i,j) \in V_\alpha \\ \omega'_E(v(i,j), v(k,l)), & x = (v(i,j), v(k,l)) \in E' \subseteq E_\alpha \\ 0 & \text{otherwise.} \end{cases}$$

We give an example of an extremal event supergraph and its weights in Figure 4.4.

We define the extremal event DAG distance to be the sum of absolute differences in node and edge weights from the extremal event supergraph determined by the best alignment we can easily compute.

Definition 4.2.2 (Extremal Event DAG Distance)**.** D and D' be two extremal event DAGs where $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n$ are the backbones of D and $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$ are the backbones of D'. The *extremal event DAG distance* is defined as:

$$d_{ED}(D, D') = \sum_{i=1}^{n} d_{\mathcal{B}}(\mathbf{x}_i, \mathbf{y}_i) + \inf_{\{\alpha_i\}_{i=1}^n} \sum_{(u,v) \in E_\alpha} |\omega_D^\alpha(u,v) - \omega_{D'}^\alpha(u,v)|.$$

where $\{\alpha_i\}_{i=1}^n$ ranges over all sets of optimal alignments between the backbones.

We define extremal event DAG distance using optimal alignments between backbones because of its computability. As we show in Chapter 6, we use modified edit distance alignment algorithms to efficiently compute backbone alignments.

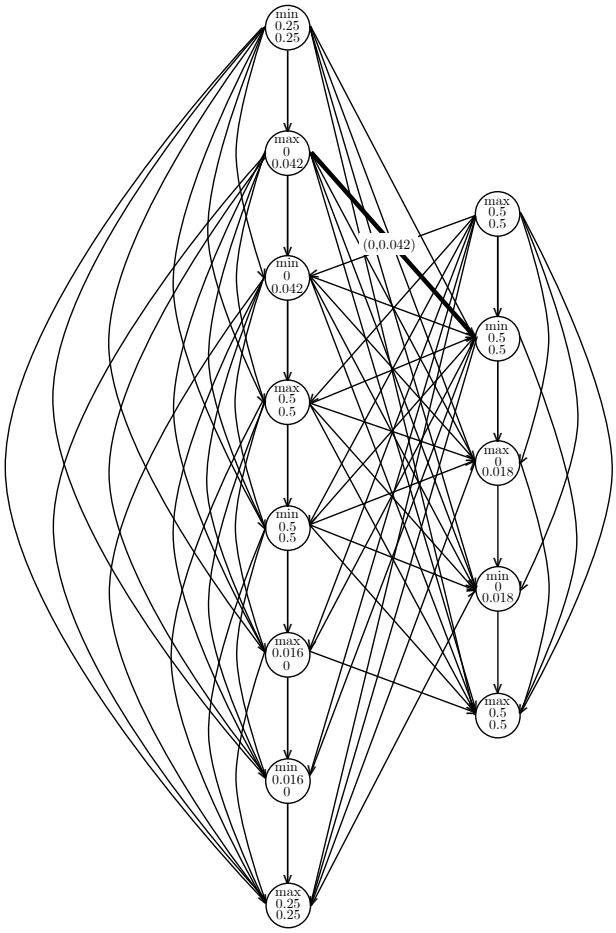


Figure 4.4: Extremal event supergraph of DAG1 and DAG2 from Figure 4.1. The nodes on the left represent to the optimal alignment between the sine backbones in DAG1 and DAG2. The nodes on the right represent to the optimal alignment between the cosine backbones in DAG1 and DAG2. The upper node weight comes from the weight function for DAG1 and the lower node weight comes from the weight function for DAG2. For readability, we present only one edge weight pair, associated to the bold edge. The edge weight on the left is from DAG1 and the edge weight on the right is from DAG2.

An open conjecture is that the sum of differences of edge weights is minimized only under an optimal alignment; that is,

$$d_{ED}(D, D') = \inf_{\{\alpha\}_{i=1}^{n}} \left(\sum_{u \in V_\alpha} |\omega_D^\alpha(u) - \omega_{D'}^\alpha(v)| + \sum_{(u,v) \in E_\alpha} |\omega_D^\alpha(u,v) - \omega_{D'}^\alpha(u,v)| \right)$$

where $\{\alpha_i\}_{i=1}^n$ ranges over all sets of alignments between the backbones.

If this conjecture is true, then we can prove the triangle inequality for the extremal event DAG distance using the same composition of alignments that we used for showing the triangle inequality holds for the backbone metric. If the conjecture is not true, then it is possible that the triangle inequality does not hold for the extremal event DAG distance. For the biological applications we have in mind, the key property that we desire is from a distance is not the triangle inequality, but rather *stability*, which is the property that small changes in two datasets does not cause a large jump in the distance between the associated extremal event DAGs. We show this property holds in Chapter 5 when the functions are "close" to one another.

CHAPTER FIVE

STABILITY RESULTS

In this section, we prove a Lipschitz stability result: that small changes in functions that are sufficiently close result in small distances between the corresponding extremal event DAGs. Our results are similar in flavor to stability for persistence diagrams [18].

5.1 Stability in Backbone Distance

We begin by proving stability results for the backbone distance. The main result of this section is Corollary 5.1.15, which states the backbone distance between backbones of two nicely tame real valued functions defined on a closed interval is bounded by a constant times the L_∞ distance between the two functions. That is,

$$d_{\mathcal{B}}(B(f), B(f')) \leq K \left\| f - f' \right\|_\infty .$$

5.1.1 Backbone Infinity Distance

To prove backbone stability, we show that the maximum difference in node weights arising from an optimal alignment is bounded by the L_∞-distance of the two corresponding functions (Theorem 5.1.14). This leads us to comparing the backbone distance to the following distance that looks at the maximum distance between aligned node weights that arises from an optimal backbone alignment.

Definition 5.1.1 (Backbone Infinity Distance)**.** Let $\mathbf{x}$, $\mathbf{y}$ be backbones. We define the *backbone infinity distance* between $\mathbf{x}$ and $\mathbf{y}$ as

$$d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{y}) = \inf_{\alpha} \max \left| \omega_{\mathbf{x}}(\alpha_{\mathbf{x}}(i)) - \omega_{\mathbf{y}}(\alpha_{\mathbf{y}}(i)) \right|$$

where α ranges over all alignments of $\mathbf{x}$ and $\mathbf{y}$.

We show the backbone infinity distance is a metric. We start by proving that the triangle inequality holds.

Lemma 5.1.2 (Backbone Infinity Distance Satisfies Triangle Inequality)**.** *Let $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ be backbones. Then,*

$$d_{\mathcal{B}_\infty}(\boldsymbol{x}, \boldsymbol{z}) \leq d_{\mathcal{B}_\infty}(\boldsymbol{x}, \boldsymbol{y}) + d_{\mathcal{B}_\infty}(\boldsymbol{y}, \boldsymbol{z})$$

Proof. Let $\alpha_1 : [k] \to \widetilde{\mathbf{x}} \times \widetilde{\mathbf{y}}$, $\alpha_2 : [m] \to \widetilde{\mathbf{y}} \times \widetilde{\mathbf{z}}$ be optimal alignments. Consider a composition alignment $\alpha_2 \circ \alpha_1$ from Construction 4.1.9. Define A as in Construction 4.1.9,

$$A_{\mathbf{x}} := \{(x, y) \in \mathrm{im}(\alpha_1) \mid \exists z \in \mathbf{z} \text{ s.t. } (y, z) \in \mathrm{im}(\alpha_2)\},$$

and

$$A_{\mathbf{z}} := \{(y, z) \in \mathrm{im}(\alpha_2) \mid \exists x \in \mathbf{x} \text{ s.t. } (x, y) \in \mathrm{im}(\alpha_1)\}.$$

Let

$$C_1 := \max_{(x,z) \in \mathrm{im}(\alpha_2 \circ \alpha_1)} \{|w_x - w_z|\}, \quad C_2 = \max_{(x,\mathbf{0}) \in \mathrm{im}(\alpha_2 \circ \alpha_1)} \{w_x\}, \quad C_3 = \max_{(\mathbf{0},z) \in \mathrm{im}(\alpha_2 \circ \alpha_1)} \{w_z\}.$$

Because $d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{z})$ is computed from an optimal alignment between $\mathbf{x}$ and $\mathbf{z}$, then

$$d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{z}) \leq \max\{C_1, C_2, C_3\}.$$

Suppose $\max\{C_1, C_2, C_3\} = C_1$. Let $(x, z) \in \mathrm{im}(\alpha_2 \circ \alpha_1)$ such that $|w_x - w_z| = C_1$. Observe since $(x, z) \in \mathrm{im}(\alpha_2 \circ \alpha_1)$ then there exists $y \in \mathbf{y}$ such that $(x, y) \in \mathrm{im}(\alpha_1)$ and

$(y, z) \in \mathrm{im}(\alpha_2)$. This observation and the triangle inequality from the L_1-norm implies

$$\begin{aligned} |w_x - w_z| &\leq |w_x - w_y| + |w_y - w_z| \\ &\leq d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{y}) + d_{\mathcal{B}_\infty}(\mathbf{y}, \mathbf{z}). \end{aligned}$$

Hence, in the case $\max\{C_1, C_2, C_3\} = C_1$, we have $d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{z}) \leq d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{y}) + d_{\mathcal{B}_\infty}(\mathbf{y}, \mathbf{z})$.

Next, suppose $\max\{C_1, C_2, C_3\} = C_2$. Let $(x, \mathbf{0}) \in \mathrm{im}(\alpha_2 \circ \alpha_1)$ such that $w_x = C_2$. Define

$$X_{(y,\mathbf{0})} := \{(x, \mathbf{0}) \in \mathrm{im}(\alpha_2 \circ \alpha_1) \mid \exists y \in \mathbf{y} \text{ s.t. } (x, y) \in \mathrm{im}(\alpha_1) \setminus A_{\mathbf{x}}\}.$$

Observe, if $(x, y) \in \mathrm{im}(\alpha_1) \setminus A_{\mathbf{x}}$, then for all $z \in \mathbf{z}$, $(y, z) \notin \mathrm{im}(\alpha_2)$. Hence, $(y, \mathbf{0}) \in \mathrm{im}(\alpha_2)$. This implies the set

$$(\mathbf{x} \times \{\mathbf{0}\}) \cap \mathrm{im}(\alpha_2 \circ \alpha_1) = ((\mathbf{x} \times \{\mathbf{0}\}) \cap \mathrm{im}(\alpha_1)) \cup X_{(y,\mathbf{0})}$$

and this union is disjoint. Either $(x, \mathbf{0}) \in (\mathbf{x} \times \{\mathbf{0}\}) \cap \mathrm{im}(\alpha_1)$ or $(x, \mathbf{0}) \in X_{(y,\mathbf{0})}$. If $(x, \mathbf{0}) \in (\mathbf{x} \times \{\mathbf{0}\}) \cap \mathrm{im}(\alpha_1)$, then

$$C_2 = w_x \leq d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{y}) \leq d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{y}) + d_{\mathcal{B}_\infty}(\mathbf{y}, \mathbf{z}).$$

Now suppose $(x, \mathbf{0}) \in X_{(y,\mathbf{0})}$. By definition of $X_{(y,\mathbf{0})}$, there exists $y \in \mathbf{y}$ such that $(x, y) \in \mathrm{im}(\alpha_1) \setminus A_{\mathbf{x}}$ and, at the same time, $(y, \mathbf{0}) \in \mathrm{im}(\alpha_2)$. Noting this observation and applying the triangle inequality from the L_1-norm gives

$$\begin{aligned} w_x &\leq |w_x - w_y| + w_y \\ &\leq d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{y}) + d_{\mathcal{B}_\infty}(\mathbf{y}, \mathbf{z}). \end{aligned}$$

We can conclude that in the case $\max\{C_1, C_2, C_3\} = C_2$ that $d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{z}) \leq d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{y}) +$

$d_{\mathcal{B}_\infty}(\mathbf{y}, \mathbf{z})$.

Lastly, suppose $\max\{C_1, C_2, C_3\} = C_3$. Let $(\mathbf{0}, z) \in \mathrm{im}(\alpha_2 \circ \alpha_1)$ such that $w_z = C_3$. Define

$$Z_{(\mathbf{0},y)} := \{(\mathbf{0}, z) \in \mathrm{im}(\alpha_2 \circ \alpha_1) \mid \exists y \in \mathbf{y} \text{ s.t. } (y, z) \in \mathrm{im}(\alpha_2) \setminus A_{\mathbf{z}}\}.$$

Similarly, the set

$$(\{\mathbf{0}\} \times \mathbf{z}) \cap \mathrm{im}(\alpha_2 \circ \alpha_1) = ((\{\mathbf{0}\} \times \mathbf{z}) \cap \mathrm{im}(\alpha_2)) \cup Z_{(\mathbf{0},y)}$$

and this union is disjoint. Either $(\mathbf{0}, z) \in (\{\mathbf{0}\} \times \mathbf{z}) \cap \mathrm{im}(\alpha_2)$ or $(\mathbf{0}, z) \in Z_{(\mathbf{0},y)}$. If $(\mathbf{0}, z) \in (\{\mathbf{0}\} \times \mathbf{z}) \cap \mathrm{im}(\alpha_2)$, then

$$C_3 = w_z \leq d_{\mathcal{B}}(\mathbf{y}, \mathbf{z}) \leq d_{\mathcal{B}}(\mathbf{x}, \mathbf{y}) + d_{\mathcal{B}}(\mathbf{y}, \mathbf{z}).$$

Now suppose $(\mathbf{0}, z) \in Z_{(\mathbf{0},y)}$. By definition of $Z_{(\mathbf{0},y)}$, there exists $y \in \mathbf{y}$ such that $(y, z) \in \mathrm{im}(\alpha_2) \setminus A_{\mathbf{z}}$ and $(\mathbf{0}, y) \in \mathrm{im}(\alpha_1)$. Noting this observation and applying the triangle inequality from the L_1-norm, we get

$$\begin{aligned} w_z &\leq |w_z - w_y| + w_y \\ &\leq d_{\mathcal{B}_\infty}(\mathbf{y}, \mathbf{z}) + d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{y}). \end{aligned}$$

We can conclude in the case $\max\{C_1, C_2, C_3\} = C_3$ that $d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{z}) \leq d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{y}) + d_{\mathcal{B}_\infty}(\mathbf{y}, \mathbf{z})$.

Therefore, the backbone infinity distance satisfies the triangle inequality. □

Proposition 5.1.3 (Backbone Infinity Distance is a Metric)**.** *Let $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ be backbones. The backbone infinity distance (Definition 5.1.1) satisfies all properties of a metric.*

Proof. Recall that for each $x \in \mathbf{x}$, we can write $x = (s_x, w_x)$; likewise for $y \in \mathbf{y}$. Let $\alpha : [k] \to \widetilde{\mathbf{x}} \times \widetilde{\mathbf{y}}$ be an optimal alignment. We verify all properties of a metric.

Non-Negativity. Since all node weights are non-negative, $d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{y}) \geq 0$.

Symmetry. By construction, $d_{\mathcal{B}_\infty}$ is symmetric; see Definition 5.1.1.

Definiteness. If $\mathbf{x} = \mathbf{y}$, then the optimal alignment aligns each node with itself, and there are no insertions. Hence, all node weights match and $d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{y}) = 0$.

On the other hand, assume $d_{\mathcal{B}_\infty}(\mathbf{x}, \mathbf{y}) = 0$. This implies

$$0 = \inf_{\alpha} \max |\omega_{\mathbf{x}}(\alpha_{\mathbf{x}}(i)) - \omega_{\mathbf{y}}(\alpha_{\mathbf{y}}(i))|.$$

Therefore, $|\omega_{\mathbf{x}}(\alpha_{\mathbf{x}}(i)) - \omega_{\mathbf{y}}(\alpha_{\mathbf{y}}(i))| \leq 0$ for all $i \in [n]$. Furthermore, $|\omega_{\mathbf{x}}(\alpha_{\mathbf{x}}(i)) - \omega_{\mathbf{y}}(\alpha_{\mathbf{y}}(i))| \geq 0$ for all $i \in [n]$. Thus, $|\omega_{\mathbf{x}}(\alpha_{\mathbf{x}}(i)) - \omega_{\mathbf{y}}(\alpha_{\mathbf{y}}(i))| = 0$ for all $i \in [n]$. Hence, each aligned pair of nodes must have the same node weight. By Definition 4.1.4, we never align two empty nodes. This implies each node in $\mathbf{x}$ is aligned with a node in $\mathbf{y}$. Furthermore, each aligned pair must have the same label by Definition 4.1.4. We can conclude $\mathbf{x} = \mathbf{y}$.

Triangle Inequality. By Lemma 5.1.2, the triangle inequality holds.

Therefore, the backbone infinity distance is a metric. □

We prove backbone distance stability by moving between concepts of local extrema of functions, points in persistence diagrams, and backbone nodes. We describe the relationship between these three concepts next.

Let $f : C \to \mathbb{R}$ be a nicely tame function and $(t, f(t))$ be a local minimum of f that does not represent the essential component in $D(f)$. Recall from Section 2.5 that at a height of $f(t)$ in the sublevel set filtration, a new connected component is born. The death of this connected component happens at the height of a local maximum denoted as $\zeta_f(t)$ (recall Definition 2.5.1). This implies existence of a point $(f(t), \zeta_f(t)) \in D(f)$ in the persistence diagram. We then compute the node life, $\frac{1}{2}\text{pers}_f(t) = \frac{1}{2}(\zeta_f(t) - f(t))$. This shows that the node $(\min, \frac{1}{2}\text{pers}_f(t)) \in B(f)$ is a backbone node. In summary, a local minimum $(t, f(t))$ corresponds to a point in the persistence diagram $(f(t), \zeta_f(t)) \in D(f)$, and a vertex in a

backbone $(\min, \frac{1}{2}\text{pers}_f(t)) \in B(f)$.

Now suppose t represents the essential component in $D(f)$, which means that t is a global minimum of f. Then, $(t, f(t))$ corresponds to $(f(t), \infty) \in D(f)$ and $(\min, \frac{1}{2}\text{pers}_f(t)) \in B(f)$ where $\frac{1}{2}\text{pers}_f(t) = \frac{1}{2}(\max(f) - f(t))$.

There is the same type of correspondence for local maxima of f by applying the same process to $-f$. All the local maxima of f become local minima of $-f$.

To make the correspondence between nodes in backbones, extrema, and points in persistence diagrams more precise, we define the following.

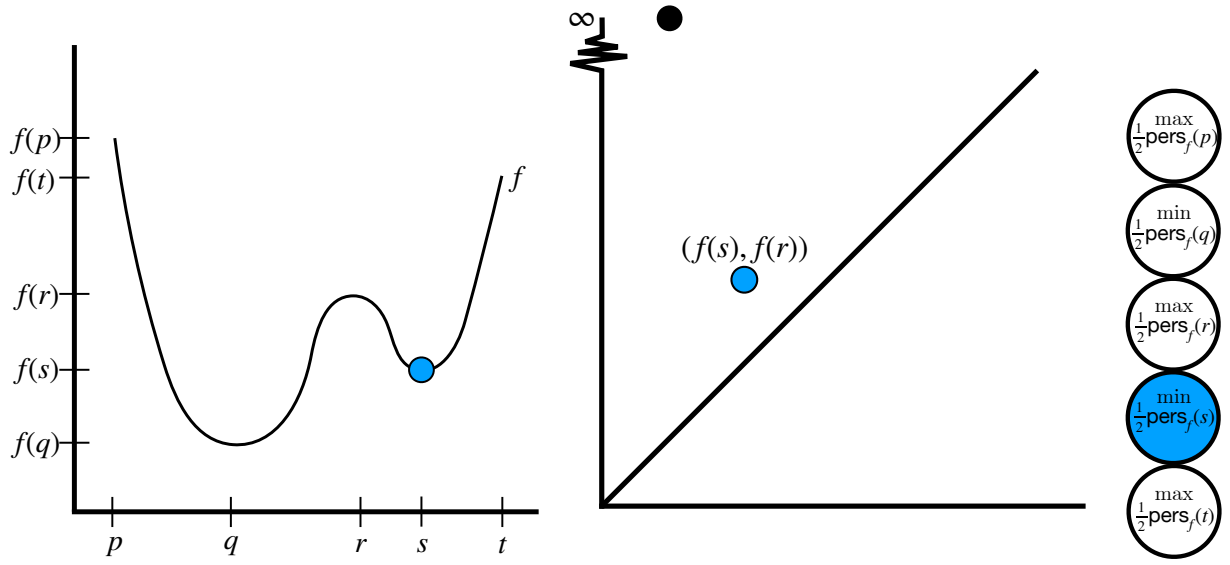
Definition 5.1.4 (Truncated Persistence Points)**.** Let $f : C \to \mathbb{R}$ be a nicely tame function. Let $(t, f(t))$ be a local minimum of f. The point $(f(t), \zeta_f(t))$ is the *truncated persistence point* of $(t, f(t))$.

We often refer to the truncated persistence points as persistence points. We note that if $(t, f(t))$ is a local maximum of f, then we declare the point $(-f(t), \zeta_{-f}(t))$ as the persistence point of $(t, f(t))$. Because of the correspondence between extrema, persistence points, and backbone nodes, we discuss pairings of extrema or persistence points to get aligned pairs in backbone alignments.

Remark 5.1.5 (Persistence Diagram Containing Diagonal). Persistence diagrams are often defined as in Definition 1.2.5 along with a union of all points on the diagonal $\Delta = \{(x, x) \in \mathbb{R}^2\}$ counted with infinite multiplicity. The addition of the diagonal is useful for defining distances between persistence diagrams. For the remainder of this section, we assume that persistence diagrams contain all points on the diagonal counted with infinite multiplicity. This representation is useful for constructing alignments between backbones.

5.1.2 Stability in Backbone Infinity Distance

A key result that we use is the *Box Lemma*, that is proved in [18] to prove stability for persistence diagrams.



(a) f (b) $D(f)$ (c) $B(f)$

Figure 5.1: Moving Between Local Minima, Persistence Diagrams, and Backbone Nodes. The local minimum, $(s, f(s))$ in light blue corresponds to the point $(f(s), f(r)) \in D(f)$ and $(\min, \frac{1}{2}\text{pers}_f(s)) \in B(f)$.

Lemma 5.1.6 (Box Lemma ([18])). *Let X be a topological space, $f, g : X \to \mathbb{R}$ be tame functions and let $\varepsilon = \|f - g\|_\infty$. For $a < b < c < d$, let $R = [a, b] \times [c, d]$ be a box in the extended plane, $\overline{\mathbb{R}}^2$, and $R_\varepsilon = [a + \varepsilon, b - \varepsilon] \times [c + \varepsilon, d - \varepsilon]$ be the box obtained by shrinking R by ε on all sides. Then,*

$$|D(f) \cap R_\varepsilon| \leq |D(g) \cap R| .$$

The next result is similar in flavor to the Easy Bijection Lemma from [18]. We first prove stability for the backbone infinity distance in a special case. The result will depend on two constants.

Definition 5.1.7 (Constants $\delta_{\min}$, $\delta_{\max}$). Let X be a topological space and $f : X \to \mathbb{R}$ be a tame function. Define $\delta_{\min}$ to be half of the smallest distance between two distinct

off-diagonal points, or a point in $D(f)$ and a point on the diagonal, that is,

$$\delta_{\min} := \frac{1}{2}\min\{\|p-q\|_\infty \mid p \in D(f) \setminus \Delta, q \in D(f), p \neq q\}.$$

The constant $\delta_{\max}$ is defined analogously using $D(-f)$.

Next, we note a relationship between Definition 5.1.7 and the minimum node life of extrema of f.

Lemma 5.1.8 (Minimum of Node Lives is Bounded Below by $\delta_{\min}$)**.** *Let $f : C \to \mathbb{R}$ be a nicely tame function. Let $\{t_i\}_{i=1}^n$ be the domain coordinates for local minima of f. Define δ to be half the minimum of the node lives of t_i, that is,*

$$\delta := \frac{1}{2}\min\{pers_f(t_i)\}_{i=1}^n.$$

Then $\delta_{\min} \leq \delta$.

Proof. Let $t \in \{t_i\}_{i=1}^n$ such that $\frac{1}{2}\text{pers}_f(t) = \delta$. Observe the point $(\frac{1}{2}\text{pers}_f(t), \frac{1}{2}\text{pers}_f(t))$ is the orthogonal projection of $(f(t), \zeta_f(t))$ onto the diagonal. In particular, $(\frac{1}{2}\text{pers}_f(t), \frac{1}{2}\text{pers}_f(t))$ is the closest point on the diagonal to $(f(t), \zeta_f(t))$. Notice,

$$\left\|(f(t), \zeta_f(t)) - (\frac{1}{2}\text{pers}_f(t), \frac{1}{2}\text{pers}_f(t))\right\|_\infty = \frac{1}{2}\text{pers}_f(t) = \delta.$$

Additionally, since for all $t_i \in \{t_i\}_{i=1}^n$ where $t_i \neq t$, we have $\frac{1}{2}\text{pers}_f(t_i) \geq \delta$, it must be the case that

$$\left\|(f(t_i), \zeta_f(t_i)) - (\frac{1}{2}\text{pers}_f(t_i), \frac{1}{2}\text{pers}_f(t_i))\right\|_\infty \geq \delta.$$

This implies that half the minimum distance between a point $p \in D(f) \setminus \Delta$ and a point on

the diagonal is equal to δ, that is,

$$\frac{1}{2}\min\{\|p-q\|_\infty \mid p \in D(f) \setminus \Delta, q \in \Delta\} = \delta.$$

Lastly, since

$$\{\|p-q\|_\infty \mid p \in D(f) \setminus \Delta, q \in \Delta\} \subset \{\|p-q\|_\infty \mid p \in D(f) \setminus \Delta, q \in D(f), p \neq q\}$$

we conclude $\delta_{\min} \leq \delta$. □

Using the same proof but with $-f$ and $D(-f)$, we find $\delta_{\max} \leq \delta$. We use the two constants $\delta_{\min}$ and $\delta_{\max}$ to determine when functions are "close".

Definition 5.1.9 (Very Close)**.** Let $f : C \to \mathbb{R}$ be a nicely tame function. Let $\delta_f = \min\{\delta_{\min}, \delta_{\max}\}$. A nicely tame function $f' : C \to \mathbb{R}$ is *very close* to f if $\|f - f'\|_\infty < \delta_f$.

Next we prove an analogue of the Easy Bijection Lemma [18] for backbones. We start by constructing an alignment between two backbones arising from nicely tame functions f and f' where f' is very close to f. Figure 5.2 shows an example on how to construct the direct alignment between very close functions.

Construction 5.1.10 (Direct Alignment)**.** Let $f, f' : C \to \mathbb{R}$ be nicely tame functions such that f' is very close to f. Let $\varepsilon = \|f - f'\|_\infty$. Note, that since f, f' are very close, we have $\varepsilon < \delta_f$. The *direct alignment* construction consists of two steps:

1. Pairing nodes in $B(f)$ with $B(f')$. Recall that each node in $B(f)$ and $B(f')$ corresponds to a local extremum of f and f', respectively. We begin by pairing local minima of f with local minima of f'. By definition of persistence diagrams, there is a one-to-one correspondence between the local minima of f and the points in $D(f)$. Thus, we

can pair local minima of f and f' by pairing off diagonal points in $D(f)$ and $D(f')$, respectively.

Let $p = (p_1, p_2) \in D(f) \setminus \Delta$. We describe what point in $D(f')$ is paired with p. Since $p \in \square_\varepsilon(p)$, the Box Lemma tells us that the multiplicity $\mu(p)$ of p satisfies

$$\mu(p) \leq |D(f') \cap \square_\varepsilon(p)| \leq |D(f) \cap \square_{2\varepsilon}(p)| .$$

By definition of δ_f and the assumption $\varepsilon < \delta_f$, we know that p is the only point contained in the set $D(f) \cap \square_{2\varepsilon}(p)$. Therefore, $|D(f') \cap \square_\varepsilon(p)| = \mu(p)$. Furthermore, since $p \in D(f) \cap \square_\varepsilon(p)$, there is the same number of points, with multiplicity, in $D(f') \cap \square_\varepsilon(p)$ and in $D(f) \cap \square_\varepsilon(p)$.

We explain how to define a bijection by pairing the points in the squares $D(f) \cap \square_\varepsilon(p)$ and $D(f') \cap \square_\varepsilon(p)$. Let $n = \mu(p)$ and let $\{t_i\}_{i=1}^n$ be the set of the domain coordinates of the local minima of f for which $f(t_i) = p_1$. Let $q = (q_1, q_2) \in D(f') \cap \square_\varepsilon(p)$. Observe $q_1 = f'(t)$ for some local minimum $(t, f'(t))$ of f'. Because $p, q \in \square_\varepsilon(p)$, we have $\|p - q\|_\infty \leq \varepsilon < \delta_f$. In particular,

$$|p_1 - q_1| = |f(t_i) - f'(t)| < \delta_f, \text{ for all } i \in [n].$$

This implies that

$$f(t_i) - \delta_f < f'(t) < f(t_i) + \delta_f, \text{ for all } i \in [n].$$

This inequality, the fact that $\delta_f \leq \delta_2$, and $f(t_i) = p_1$ for all $i \in [n]$ implies

$$t \in A := (f - \delta_f)^{-1}(-\infty, p_1 + \delta_f).$$

By Lemma 5.1.8, $\delta_f \leq \frac{1}{2}\min\{\text{pers}_f(t_i)\}_{i=1}^n$. Applying Proposition 1 of [7], we find A is a disjoint union of intervals and each contains exactly one t_i, i.e., $A = \bigcup_{i=1}^n \varphi_{\delta_f}(t_i)$. Let $t_i^* \in \{t_i\}_{i=1}^n$ such that $t \in \varphi_{\delta_f}(t_i^*)$. For our alignment, we pair the local minima $(t_i^*, f(t_i^*))$ with $(t, f'(t))$. Iterating this process for all points $q \in D(f') \cap \square_\varepsilon(p)$ results in a bijection between points in the squares $D(f) \cap \square_\varepsilon(p)$ and $D(f') \cap \square_\varepsilon(p)$.

Iterating the above procedure for all points $p \in D(f) \setminus \Delta$, we pair all local minima of f with local minima of f'. All remaining local minima of f' are paired with an empty node. The order of the alignment is given by the domain coordinates of f'.

What remains are the local maxima of f and f'. To pair these extrema, we apply the same exact process to minima of $-f$ and $-f'$ since they are local maxima of f and f'.

2. Indexing the pairs so that order of the backbones for $B(f)$ and $B(f')$ are preserved. Let $\mathbf{x} = B(f)$ and $\mathbf{x}' = B(f')$. Since extrema in f (and f') are in one-to-one correspondence with nodes in $\mathbf{x}$ (and $\mathbf{x}'$, respectively), we know that each pair of aligned extrema in Step (1) correspond to a pair of nodes in $\widetilde{\mathbf{x}} \times \widetilde{\mathbf{x}}'$. To construct the *direct alignment* $\alpha : [k] \to \widetilde{\mathbf{x}} \times \widetilde{\mathbf{x}}'$, we order the pairs found in Step (1) based on the order of the domain coordinates of the local extrema of f'. It follows that k is the number of local extrema of f'. Let $(t_i', f'(t_i'))$ be the i^{th} extremum based on order of domain coordinates of f', and assume, without loss of generality, that this extremum is a local minimum. Then, $\alpha(i) = (\alpha_{\mathbf{x}}(i), \alpha_{\mathbf{x}'}(i))$ is given by $\alpha_{\mathbf{x}'}(i) = (\min, \frac{1}{2}\text{pers}_{f'}(t_i'))$ and $\alpha_{\mathbf{x}}(i)$ is the paired node pair from Step (1).

Lemma 5.1.11 (Direct Alignment is an Alignment)**.** *Let $f, f' : C \to \mathbb{R}$ be nicely tame functions such that f' is very close to f. Let $\boldsymbol{x} = B(f)$ and $\boldsymbol{x}' = B(f')$. The direct alignment, $\alpha : [k] \to \widetilde{\boldsymbol{x}} \times \widetilde{\boldsymbol{x}}'$ constructed in Construction 5.1.10 is an alignment.*

Proof. We show Definition 4.1.4 holds. By Construction 5.1.10, we immediately see we have no null alignments, misalignments, and have a restriction to matching. Hence, Property (1),

Property (3), and Property (4) hold. What remains is showing the alignment preserves order of backbones.

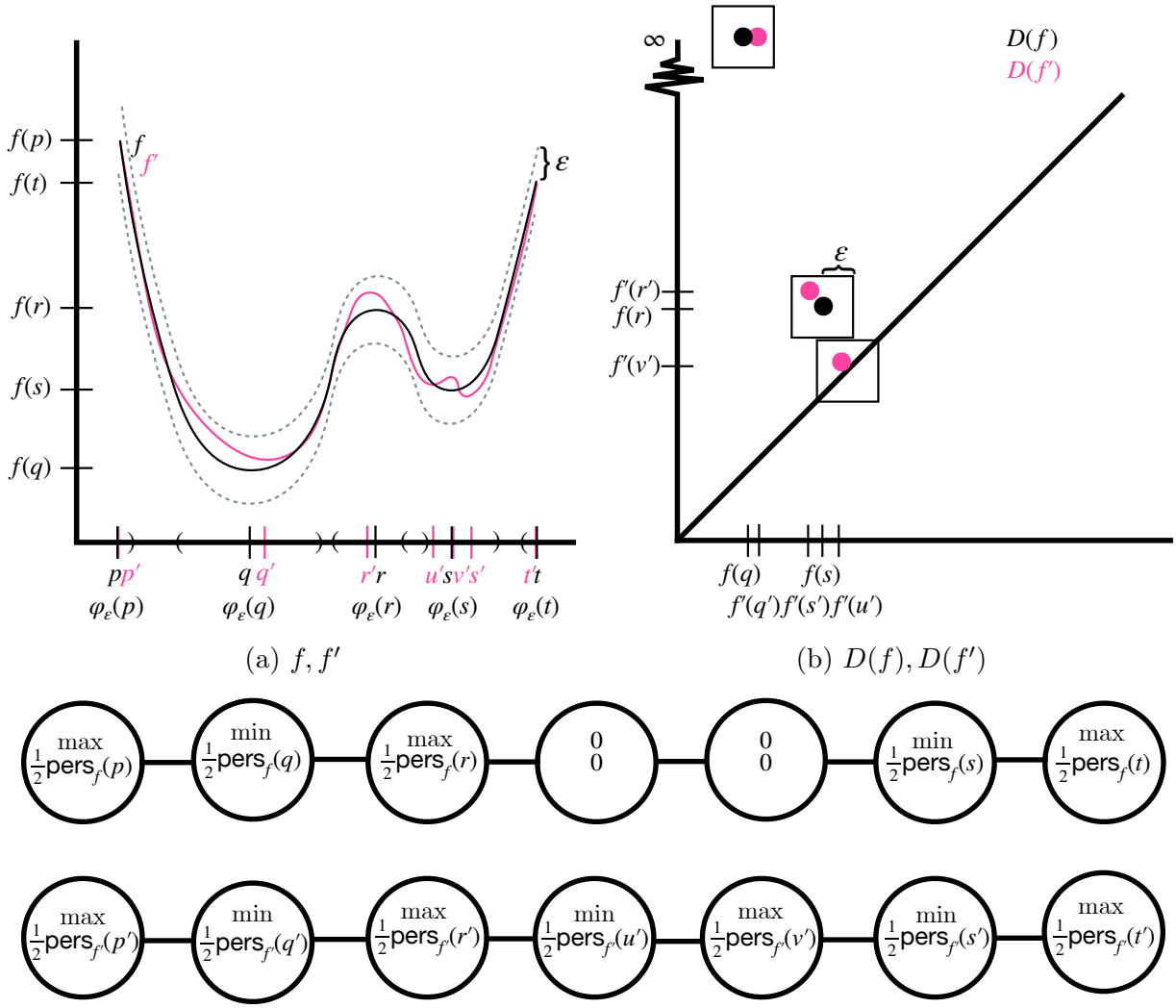
Since the nodes of $\mathbf{x}'$ are ordered by domain coordinates of local extrema of f', the alignment α preserves the order of nodes of $\mathbf{x}'$. We now show that the alignment α also preserves the order of nodes in $B(f)$. Consider nodes $\alpha_{\mathbf{x}}(i)$, $\alpha_{\mathbf{x}}(j) \in \mathbf{x} = B(f)$ such that $i < j$ and $\alpha_{\mathbf{x}}(i)$, $\alpha_{\mathbf{x}}(j)$ map to local extrema $(t, f(t))$ and $(s, f(s))$, respectively. Trivially, t, s are contained in $\varphi_{\delta_f}(t)$ and $\varphi_{\delta_f}(s)$, respectively. Since $\|f - f'\|_\infty < \delta_f \leq \min\{\frac{1}{2}\text{pers}_f(t_i)\}_{i=1}^n$ by Lemma 5.1.8, then if t and s are not adjacent, $\varphi_{\delta_f}(t) \cap \varphi_{\delta_f}(s) = \emptyset$ by Proposition 1 of [7]. Otherwise t and s are adjacent and by Lemma 3.1.1, Statement (3), we have $t \notin \varphi_{\delta_f}(s)$ and $s \notin \varphi_{\delta_f}(t)$. Either way, we find $\iota(\alpha_{\mathbf{x}'}(i)) < \iota(\alpha_{\mathbf{x}'}(j))$ implies $\iota(\alpha_{\mathbf{x}}(i)) < \iota(\alpha_{\mathbf{x}}(j))$. The same argument in reverse can be used to show that if $i < j$, then $\iota(\alpha_{\mathbf{x}}(i)) < \iota(\alpha_{\mathbf{x}}(j))$. Therefore, the order of the backbones $B(f)$ and $B(f')$ is preserved and we have constructed an alignment between $B(f)$ and $B(f')$. □

Next, we prove a bound on the absolute difference between aligned weights in the Direct alignment.

Lemma 5.1.12 (Bound in Difference in Node Weights in Direct Alignment)**.** *Let $f, f' : C \to \mathbb{R}$ be nicely tame functions such that f' is very close to f. Let $\varepsilon := \|f - f'\|_\infty$. Let $\boldsymbol{x} = B(f)$, $\boldsymbol{x}' = B(f')$, and $\alpha : [k] \to \widetilde{\boldsymbol{x}} \times \widetilde{\boldsymbol{x}}'$ be the direct alignment as defined in Construction 5.1.10. Then, the absolute difference in weights between aligned nodes is bounded by ε; that is, for all $(x, x') \in \text{im}(\alpha)$, $|w_x - w_{x'}| \leq \varepsilon$.*

Proof. Let $(x, x') \in \text{im}(\alpha)$. Either both represent extrema from f and f', or the node x is the empty node.

First, assume that both nodes represent extrema. For this proof, we assume they are minima and note an analogous argument holds for maxima. Let $(t, f(t))$ and $(t', f'(t'))$ be the local minima corresponding to nodes x and x', respectively. Either both of these



(a) f, f'

(b) $D(f), D(f')$

(c) Direct Alignment between $B(f)$ and $B(f')$

Figure 5.2: Construction of Direct Alignment. In Figure 5.2a, f is the black function while f' is the pink function. The black labeled ticks denote the domain coordinates of the local extrema of f and the pink labeled ticks denote the domain coordinates of the local extrema of f'. The ε-extremal intervals for the local extrema of f are illustrated. Since any two points in $D(f)$ where one is not a diagonal point, have a distance of at least ε, and $f' \in N_\varepsilon(f)$, we have f' is very close to f. Applying the Direct Alignment Lemma, we get pairings of points in $D(f)$ and $D(f')$ as shown in Figure 5.2b. From the pairings in $D(f)$ and $D(f')$, we get pairings of nodes with the label "min" that preserve order. The preservation of order comes from how the extremal intervals for minima of f are disjoint. We apply an analogous proccess to pair nodes with the label "max". The alignment that is constructed in the Direct Alignment Lemma for f and f' is shown in Figure 5.2c.

extrema do not represent the essential component in $D(f)$ and $D(f')$ or at least one of them does. Suppose neither represents the essential component. Consider the persistence points $p = (f(t), \zeta_f(t))$, $q = (f'(t'), \zeta_{f'}(t'))$ in $D(f) \setminus \Delta$ and $D(f') \setminus \Delta$ respectively. From Construction 5.1.10, we know that both $p, q \in \square_\varepsilon(p)$. This implies $\|p - q\|_\infty \leq \varepsilon$. Hence,

$$\begin{aligned}
|w_x - w_{x'}| &= \frac{1}{2}|\text{pers}_f(t) - \text{pers}_{f'}(t')| \\
&= \frac{1}{2}|(\zeta_f(t) - f(t)) - (\zeta_{f'}(t') - f'(t'))| \\
&= \frac{1}{2}|(\zeta_f(t) - \zeta_{f'}(t')) + (f'(t') - f(t))| \\
&\leq \frac{1}{2}|\zeta_f(t) - \zeta_{f'}(t')| + \frac{1}{2}|f'(t') - f(t)| \\
&\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.
\end{aligned}$$

Next, consider the case at least one of $(t, f'(t))$ or $(t', f'(t'))$ represents the essential component. By the Box Lemma, we can infer that both points have to represent the essential component. Then we know these points are global minima of f and f'. Since $(f(t), \infty)$ and $(f'(t'), \infty)$ are both contained in the square of radius ε centered at $(f(t), \infty)$ in the extended plane, $|f(t) - f'(t')| \leq \varepsilon$. Additionally, by Construction 5.1.10 we know that $t_{\max}$, a global maximum of f, is paired with $t'_{\max}$, a global maximum of f' such that $t'_{\max} \in \varphi_{\delta_f}(t_{\max})$. This implies $|f(t_{\max}) - f'(t'_{\max})| \leq \varepsilon$. Applying the same computation as above, we see $|w_x - w_{x'}| \leq \varepsilon$.

Lastly, consider the case $(t', f'(t'))$ is paired with an empty node. Consider the point $(f'(t'), \zeta_{f'}(t')) \in D(f')$. The Box Lemma implies $\square_\varepsilon((f'(t'), \zeta_{f'}(t')))$ must contain at least one point from $D(f)$. By assumption $(t', f'(t'))$ is paired with an empty node, and so $\square_\varepsilon((f'(t'), \zeta_{f'}(t'))$ must contain a point on the diagonal. Therefore, $(f'(t'), \zeta_{f'}(t'))$ is within an L_∞ distance of ε from a point on the diagonal. Because the point $(\frac{1}{2}\text{pers}_{f'}(t'), \frac{1}{2}\text{pers}_{f'}(t'))$ is the orthogonal projection of $(f'(t'), \zeta_{f'}(t'))$ onto the diagonal, it is the closest point on the

diagonal in $\mathbb{R}^2$ to $(f'(t'), \zeta_{f'}(t'))$. Therefore

$$\left\| (f'(t'), \zeta_{f'}(t')) - (\frac{1}{2}\text{pers}_{f'}(t'), \frac{1}{2}\text{pers}_{f'}(t')) \right\|_\infty \leq \varepsilon. \tag{5.1}$$

Comparing the x-coordinates of pair of points in (5.1), we find $\frac{1}{2}\text{pers}_{f'}(t') - f'(t') \leq \varepsilon$ and comparing the y-coordinates we find $\zeta_{f'}(t') - \frac{1}{2}\text{pers}_{f'}(t') \leq \varepsilon$. Therefore,

$$\zeta_{f'}(t') \leq \frac{1}{2}\text{pers}_{f'}(t') + \varepsilon, \quad f'(t') \geq \frac{1}{2}\text{pers}_{f'}(t') - \varepsilon.$$

Altogether we find

$$\text{pers}_{f'}(t') = \zeta_{f'}(t') - f'(t') \leq (\frac{1}{2}\text{pers}_{f'}(t') + \varepsilon) - (\frac{1}{2}\text{pers}_{f'}(t') - \varepsilon) = 2\varepsilon,$$

and, therefore, $\frac{1}{2}\text{pers}_{f'}(t') \leq \varepsilon$.

We conclude that for all paired extrema,

$$|w_x - w_{x'}| \leq \varepsilon.$$

□

We can now prove local stability for the backbone infinity distance.

Lemma 5.1.13 (Local Backbone Infinity Stability)**.** *Let $f, f' : C \to \mathbb{R}$ be nicely tame functions such that f' is very close to f. Then,*

$$d_{\mathcal{B}_\infty}(B(f), B(f')) \leq \|f - f'\|_\infty .$$

Proof. In Lemma 5.1.12, we showed that using the direct alignment between $B(f)$ and $B(f')$, the absolute difference in aligned node weights is bounded by $\|f - f'\|_\infty$. Since the backbone infinity distance is defined by using an optimal alignment, we get

$$d_{\mathcal{B}_\infty}(B(f), B(f')) \leq \|f - f'\|_\infty .$$

□

To remove the assumption that f and f' are very close and thus to globalize the backbone infinity stability result, we construct a straight-line homotopy between f and f' and consider a finite number of functions within this homotopy for which every two successive functions are very close. For each such pair of functions, Lemma 5.1.13 applies and we are able to apply almost the same argument as the proof of the Interpolation Lemma in [18]. Because we sample functions between f and f' in the homotopy, we are able to conclude that $d_{\mathcal{B}_\infty}(B(f), B(f')) \leq \|f - f'\|_\infty$.

Theorem 5.1.14 (Backbone Infinity Stability)**.** *Let $f, f' : C \to \mathbb{R}$ be nicely tame functions. Then,*

$$d_{\mathcal{B}_\infty}(B(f), B(f')) \leq \|f - f'\|_\infty .$$

Proof. Let $c := \|f - f'\|_\infty$. Define $h_\lambda := (1 - \lambda)f + \lambda f'$ where $\lambda \in [0, 1]$. This is the family of convex combinations of f and f' forms a linear interpolation between the two functions, starting at $h_0 = f$ and ending at $h_1 = f'$. Furthermore, we define $\delta(\lambda) := \delta_{h_\lambda}$ as in Definition 5.1.9. Consider the open cover U of $[0, 1]$ by open intervals $J_\lambda = (\lambda - \delta(\lambda)/2c, \lambda + \delta(\lambda)/2c)$ for all $\lambda \in [0, 1]$. The compactness of $[0, 1]$ implies the existence of a finite subcover U' of U. Let $\lambda_1 < \lambda_2 < \cdots < \lambda_n$ be the midpoints of the open intervals in U'. Observe, that half the length of J_λ is equal to $\delta(\lambda)/2c$. Since any two consecutive intervals

J_{λ_i} and $J_{\lambda_{i+1}}$ have a non-empty intersection,

$$\begin{aligned}\lambda_{i+1} - \lambda_i &\leq \delta(\lambda_i)/2c + \delta(\lambda_{i+1})/2c \\ &\leq 2\max\{\delta(\lambda_i)/2c, \delta(\lambda_{i+1})/2c\} \\ &= \max\{\delta(\lambda_i), \delta(\lambda_{i+1})\}/c\end{aligned}$$

Furthermore, note

$$\begin{aligned}|h_{\lambda_i} - h_{\lambda_{i+1}}| &= |((1-\lambda_i)f + \lambda_i f') - ((1-\lambda_{i+1})f + \lambda_{i+1}f')| \\ &= |f(\lambda_{i+1} - \lambda_i) - f'(\lambda_{i+1} - \lambda_i)| \\ &= \|f - f'\|_\infty (\lambda_{i+1} - \lambda_i).\end{aligned}$$

This implies

$$\left\|h_{\lambda_i} - h_{\lambda_{i+1}}\right\|_\infty = c(\lambda_{i+1} - \lambda_i) \leq \max\{\delta(\lambda_i), \delta(\lambda_{i+1})\}.$$

Therefore, h_{λ_i} is very close to $h_{\lambda_{i+1}}$ or vice-versa. Either way, Lemma 5.1.13 applies and we have

$$d_{\mathcal{B}_\infty}(h_{\lambda_i}, h_{\lambda_{i+1}}) \leq \left\|h_{\lambda_i} - h_{\lambda_{i+1}}\right\|_\infty$$

for all $1 \leq i \leq n-1$. Setting $\lambda_0 = 0$ and $\lambda_{n+1} = 1$, we see the inequality also holds for $i = 0$ and $i = n$ because h_{λ_1} is very close to h_{λ_0}, and $h_{\lambda_{n+1}}$ is very close to h_{λ_n} Therefore,

$$\begin{aligned}d_{\mathcal{B}_\infty}(B(f), B(f')) &\leq \sum_{i=0}^{n} d_{\mathcal{B}_\infty}(B(h_{\lambda_i}), B(h_{\lambda_{i+1}})) \\ &\leq \sum_{i=0}^{n} \left\|h_{\lambda_i} - h_{\lambda_{i+1}}\right\|_\infty \\ &= \|f - g\|_\infty .\end{aligned}$$

The first inequality follows from the triangle inequality of the backbone infinity distance (Lemma 5.1.2). The last equality follows from how the collection h_{λ_i} samples the straight

line homotopy from f to f'. Thus $d_{\mathcal{B}_\infty}(B(f), B(f')) \leq \|f - f'\|_\infty$. □

5.1.3 Backbone Stability

Using Theorem 5.1.14, we also get stability results for the backbone distance. If the alignment that realizes the backbone infinity distance between two backbones, $B(f)$ and $B(f')$ is of length K, then the sum of absolute differences of node weights is bounded by $K\|f - f'\|_\infty$. This is because the backbone distance is bounded by the sum of absolute differences in node weights from the alignment realizing the backbone infinity distance.

Corollary 5.1.15 (Backbone Stability)**.** *Let $f, f' : C \to \mathbb{R}$ be nicely tame functions. Let K be the length of the alignment realizing the backbone infinity distance between $B(f)$ and $B(f')$. Then,*

$$d_{\mathcal{B}}(B(f), B(f')) \leq K\|f - f'\|_\infty.$$

If we are unable to compute K, note that we can bound K by the number of extrema of f plus the number of extrema of f' because that is the longest possible length of an alignment between $B(f)$ and $B(f')$.

5.2 Local Stability in Extremal Event DAG Distance

We showed stability between backbones. In this section, we extend those results to the entire extremal event DAG in a local case (when f' is extremely close to f). We start by proving that the direct alignment for functions we call *extremely close* is the optimal backbone alignment for the backbones of those two functions.

Definition 5.2.1 (Extremely Close)**.** Let $f : C \to \mathbb{R}$ be a nicely tame function. Let δ_f be as defined in Definition 5.1.7. A nicely tame function $f' : C \to \mathbb{R}$ is *extremely close* to f if $\|f - f'\|_\infty < \delta_f/2$.

The difference between functions that are very close and extremely close is that the constant is δ_f is divided by two for functions that are extremely close. This is needed in order for the following lemma to hold.

Lemma 5.2.2 (Extremal Pairs With Node Life Differences Greater Than ε)**.** *Let $f, f' : C \to \mathbb{R}$ be nicely tame functions such that f' is extremely close to f. Let $\varepsilon := \|f - f'\|_\infty$. Suppose $p := (f(t), \zeta_f(t)) \in D(f) \setminus \Delta$ and $q := (f'(t'), \zeta_{f'}(t')) \in D(f') \cap \square_\varepsilon(p)$. Then,*

1. *$\frac{1}{2} pers_{f'}(t') > \varepsilon$.*
2. *$|\frac{1}{2} pers_{f'}(t') - \frac{1}{2} pers_f(s)| > \varepsilon$ for all $(f(s), \zeta_f(s)) \in D(f) \setminus \square_\varepsilon(p)$.*

Proof. Suppose $p := (f(t), \zeta_f(t)) \in D(f) \setminus \Delta$ and $q := (f'(t'), \zeta_{f'}(t')) \in D(f') \cap \square_\varepsilon(p)$.

1. We first show $\frac{1}{2}\text{pers}_{f'}(t') > \varepsilon$. Consider $\square_{4\varepsilon}(p)$. Since f' is extremely close to f, $\varepsilon < \delta_f/2$. Hence, $4\varepsilon < 2\delta_f$. Recall $2\delta_f$ is at most the smallest distance between any two points in $D(f)$, provided that at least one point is not on the diagonal. Therefore, $\square_{4\varepsilon}(p)$ does not intersect the diagonal. Consider a square of radius 4ε where the right bottom corner is the point $(f(t) + 4\varepsilon, f(t) + 4\varepsilon)$, that is, $\square_{4\varepsilon}((f(t), f(t) + 8\varepsilon))$. Since $q \in D(f') \cap \square_\varepsilon(p)$, the difference of the y- and x- coordinates of q (which is the persistence of q) is bounded below by the difference of y- and x- coordinates of the point $(f(t) + \varepsilon, f(t) + 7\varepsilon)$, see Figure 5.3a. Hence,

$$\frac{1}{2}\text{pers}_{f'}(t') > \frac{1}{2}((f(t) + 7\varepsilon) - (f(t) + \varepsilon)) = 3\varepsilon > \varepsilon.$$

2. Let $(f(s), \zeta_f(s)) \in D(f) \setminus \square_\varepsilon(p)$. Since f' is extremely close to f, $\varepsilon < \delta_f/2$ and hence $2\varepsilon < \delta_f$. Since δ_f is at most half the smallest distance between two points in $D(f)$ where at least one point is not on the diagonal, the squares of radius 2ε centered at

different points in $D(f) \setminus \Delta$ are all disjoint. This implies

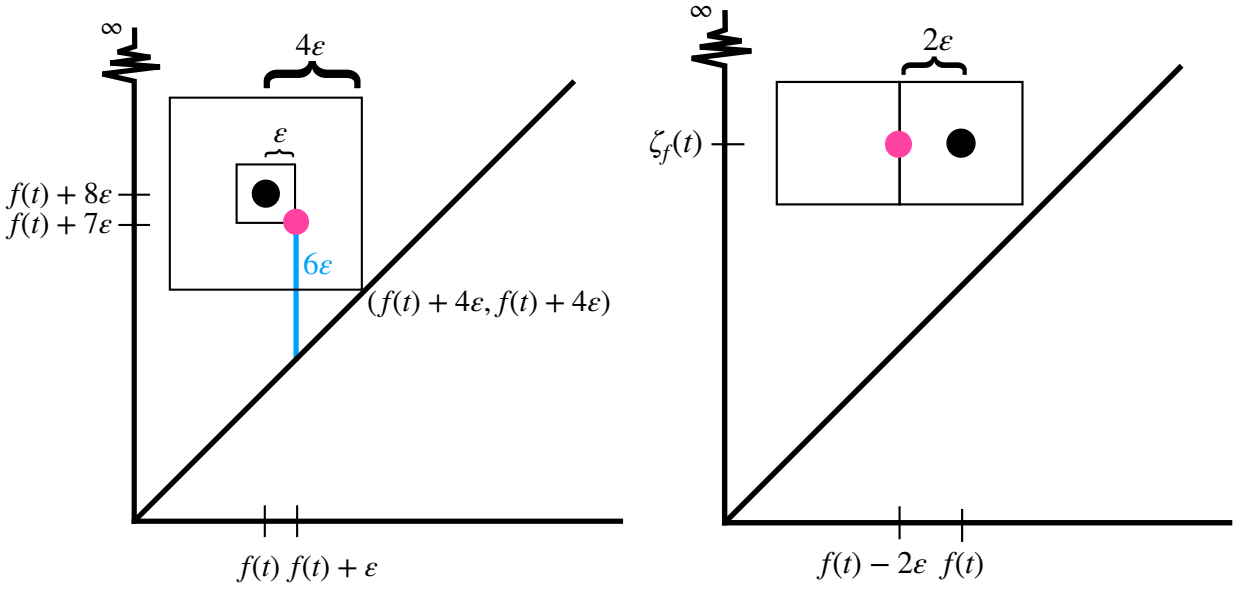
$$\|(f(t), \zeta_f(t)) - (f(s), \zeta_f(s))\|_\infty > 2\varepsilon.$$

Furthermore, by the direct alignment, we know $q \in \square_\varepsilon(p)$. This implies $q \notin \square_{2\varepsilon}((f(s), \zeta_f(s)))$. Consider the point $((f(s) - 2\varepsilon), \zeta_f(s))$. From planar geometry (See Figure 5.3b),

$$|\text{pers}_{f'}(t') - \text{pers}_f(s)| > |(\zeta_f(s) - f(s)) - (\zeta_f(s) - (f(s) - 2\varepsilon))| = 2\varepsilon.$$

Therefore, $\frac{1}{2}|\text{pers}_{f'}(t') - \text{pers}_f(s)| > \varepsilon$.

□



(a) Case 1 (b) Case 2

Figure 5.3: Geometric Arguments for Lemma 5.2.2. In Figure 5.3a, $\frac{1}{2}\text{pers}_{f'}(t') > \frac{1}{2}((f(t) + 7\varepsilon) - (f(t) + \varepsilon) = 3\varepsilon > \varepsilon$. In Figure 5.3b, $\frac{1}{2}|\text{pers}_f(t) - \text{pers}_{f'}(t')| > \frac{1}{2}|(\zeta_f(t) - f(t)) - (\zeta_f(t) - (f(t) - 2\varepsilon))| = \varepsilon$.

Next, we prove the uniqueness and optimality of the direct alignment for extremely close and nicely tame functions. We first need a technical lemma to prove this.

Lemma 5.2.3 (Bijections Within Boxes)**.** *Let $f, f' : C \to \mathbb{R}$ be nicely tame functions such that f' is extremely close to f. Let α be the direct alignment defined in Construction 5.1.10 between $\boldsymbol{x} := B(f)$ and $\boldsymbol{x}' := B(f')$. Suppose η is a different alignment between $\boldsymbol{x}$ and $\boldsymbol{x}'$ such that $\boldsymbol{cost}(\eta) \leq \boldsymbol{cost}(\alpha)$. For each $x' \in \boldsymbol{x}'$, let $x_\alpha \in \widetilde{\boldsymbol{x}}$ be the unique element such that $(x_\alpha, x') \in \mathrm{im}(\alpha)$. For each $x' \in \boldsymbol{x}'$, define x_η similarly. Suppose there exists $x' \in \boldsymbol{x}'$ such that*

$$|w_{x'} - w_{x_\eta}| \leq |w_{x'} - w_{x_\alpha}|$$

and $x_\eta \neq x_\alpha$. Then

1. *$|w_{x'} - w_{x_\eta}| = |w_{x'} - w_{x_\alpha}|$.*
2. *There exists $z' \in \boldsymbol{x}'$ for which $|w_{z'} - w_{x_\eta}| > \varepsilon$ where $\varepsilon := \|f - f'\|_\infty$.*

Proof. We prove Statement (1).

Let $(t', f'(t'))$ be the local extemum of f' associated with node x', and let $(t, f(t))$ be the local extremum of f associated with x_α. Without loss of generality, we asume $(t', f'(t'))$ is a local minimum. (Note that if $(t', f'(t'))$ is a local maximum, then we apply the same argument with $-f'$ and $-f$). Since $(x_\alpha, x') \in \mathrm{im}(\alpha)$, we know that $(t, f(t))$ is also a local minimum. We first prove, by the way of contradiction, that neither x_η nor x_α is the empty node.

If x_η is the empty node, then, by the assumption $x_\eta \neq x_\alpha$, it follows that x_α is not the empty node. From Construction 5.1.10, we know that $(f'(t'), \zeta_{f'}(t')) \in D(f') \cap \square_\varepsilon(f(t), \zeta_f(t))$.

Applying Lemma 5.2.2, we find

$$w_{x'} > \varepsilon. \tag{5.2}$$

On the other hand, by assumption,

$$|w_{x'} - w_{x_\eta}| \leq |w_{x'} - w_{x_\alpha}| \leq \varepsilon,$$

where the last inequality follows from Lemma 5.1.12. Finally, using (5.2) and that $w_{x_\eta} = 0$, we have

$$|w_{x'} - w_{x_\eta}| = w_{x'} > \varepsilon,$$

giving a contradiction. Therefore, we conclude that x_η is not the empty node.

If x_α is the empty node, then by the same argument as above, we arrive at contradiction. Therefore, x_α is also not the empty node.

Therefore, neither x_η nor x_α is an empty node. Let $(s_\eta, f(s_\eta))$ be the local extremum of f associated with node x_η. By Lemma 5.1.12, $|w_{x'} - w_{x_\alpha}| \leq \varepsilon$. Additionally, consider the point $p = (f(s_\eta), \zeta_f(s_\eta)) \in D(f)$ and $\square_\varepsilon(p)$. In Construction 5.1.10, we established a bijection between the multiplicity of p, denoted $\mu(p)$, and the number of points contained in $D(f') \cap \square_\varepsilon(p)$. Additionally, in Lemma 5.1.12, we showed for all points $(f'(x), \zeta_{f'}(x)) \in D(f') \cap \square_\varepsilon(p)$, we have $\frac{1}{2}|\text{pers}_{f'}(x) - \text{pers}_f(s)| \leq \varepsilon$. Furthermore, by Lemma 5.2.2, $\frac{1}{2}|\text{pers}_{f'}(y) - \text{pers}_f(s)| > \varepsilon$ for all $(f'(y), \zeta_{f'}(y)) \in D(f') \setminus \square_\varepsilon(p)$. By assumption,

$$|w_{x'} - w_{x_\eta}| \leq |w_{x'} - w_{x_\alpha}| \leq \varepsilon. \tag{5.3}$$

Let $(s_\alpha, f(s_\alpha))$ be the local extremum of f associated with node x_α. Since the only point of $D(f)$ contained in the square $\square_\varepsilon(p)$ is the point p with multiplicity $\mu(p)$, we must have $(f(s_\eta), \zeta_f(s_\eta)) = (f(s_\alpha), \zeta_f(s_\alpha))$ in order for (5.3) to hold. Therefore, the two extrema of f have the same node lives and thus $w_{x_\eta} = w_{x_\alpha}$. We can conclude

$$|w_{x'} - w_{x_\eta}| = |w_{x'} - w_{x_\alpha}|,$$

as was to be shown to prove Statement (1).

Next we prove Statement (2). We note that the ε-extremal intervals discussed in this proof are all constructed from f. We claim there exists a local extremum $(s', f'(s'))$ of f' such that $(f'(s'), \zeta_{f'}(s')) \in D(f') \cap \square_\varepsilon(p)$ and is aligned via η with an extremum $(s, f(s))$ such that $(f(s), \zeta_f(s)) \in D(f) \backslash \square_\varepsilon(p)$. By way of contradiction, suppose that is not the case. Hence, all persistence points in $D(f') \cap \square_\varepsilon(p)$ are paired with persistence points in $D(f) \cap \square_\varepsilon(p)$. Since $\eta \neq \alpha$, η restricted to $D(f) \cap \square_\varepsilon(p)$ is a bijection onto $D(f') \cap \square_\varepsilon(p)$ that is different from the bijection α. In other words, if we denote $\gamma := \alpha_{|D(f) \cap \square_\varepsilon(p)}$ and $\gamma' := \eta_{|D(f) \cap \square_\varepsilon(p)}$, we have $\gamma \neq \gamma'$. By Construction 5.1.10 of α and noting $\eta \neq \alpha$, there exists an extremum $(s', f'(s'))$ that aligns via η with an extremum $(s, f(s))$ such that s' does not belong to interval of size ε around s; that is, $s' \notin \varphi_\varepsilon(s)$. Without loss of generality, suppose $s < s'$. Since f' is extremely close to f, the number of extrema with persistence points contained in $(D(f) \cup D(f')) \cap \square_\varepsilon(p)$ where the domain coordinates are greater than s is the same for both f and f'. Let

$$A := \{(t, f(t)) \mid (f(t), \zeta_f(t)) \in D(f) \cap \square_\varepsilon(p) \text{ and } t > s\}$$

be the set of extrema of f whose persistence points are contained in $D(f) \cap \square_\varepsilon(p)$ such that the domain coordinates of these extrema are greater than s. Additionally, let

$$B := \{(t, f'(t)) \mid (f'(t), \zeta_{f'}(t)) \in D(f') \cap \square_\varepsilon(p) \text{ and } t > s'\}$$

To preserve order in η, elements of A must be aligned with the elements of B. Since $|A| > |B|$, by the pigeonhole principle, at least two extrema of f are aligned by η with the same extremum of f'. This contradicts the fact that η is a bijection. This contradiction shows that there exists a $z \in \mathbf{x}'$ for which $|w_{z'} - w_{z_\eta}| > \varepsilon$. By construction of the direct alignment, $|w_{z'} - w_{z_\eta}| \leq \varepsilon$. Therefore, $|w_{z'} - w_{z_\eta}| > |w_{z'} - w_{z_\eta}|$.

$\square$

Now we prove the direct alignment gives an optimal backbone alignment when two functions are extremely close.

Lemma 5.2.4 (Direct Alignment Gives Optimal Backbone Alignment)**.** *Let $f, f' : C \to \mathbb{R}$ be nicely tame functions such that f' is extremely close to f. Then, the direct alignment defined in Construction 5.1.10 is the unique optimal alignment that realizes $d_{\mathcal{B}}(B(f), B(f'))$.*

Proof. Let α be the direct alignment between $\mathbf{x} := B(f)$ and $\mathbf{x}' := B(f')$. Recall in Definition 4.1.1 that each $x \in \mathbf{x}$ can be written as a tuple $x = (s_x, w_x)$; likewise, we can write $x' \in \mathbf{x}'$ as $x' = (s_{x'}, w_{x'})$. By way of contradiction, suppose η is a different alignment between $\mathbf{x}$ and $\mathbf{x}'$ such that $\mathbf{cost}(\eta) \leq \mathbf{cost}(\alpha)$. Recall from the construction of the direct alignment (Construction 5.1.10) that the length of the direct alignment is the number of nodes in $\mathbf{x}'$. By Construction 5.1.10, for each $x' \in \mathbf{x}'$, there exists a unique $x_\alpha \in \widetilde{\mathbf{x}}$ such that $(x_\alpha, x') \in \mathrm{im}(\alpha)$. Hence, we can write

$$\mathbf{cost}(\alpha) = \sum_{(x,x') \in \mathrm{im}(\alpha)} |w_x - w_{x'}| = \sum_{x' \in \mathbf{x}'} |w_{x'} - w_{x_\alpha}|.$$

Since η is an alignment, it must align all nodes of $\mathbf{x}'$. Hence, we have $\mathsf{len}\,\eta \geq \mathsf{len}\,\mathbf{x}' = \mathsf{len}\,\alpha$. Let x_η denote the unique element of $\widetilde{\mathbf{x}}$ such that $(x_\eta, x') \in \mathrm{im}(\eta)$. We now discuss the following logical dichotomy: either

Case 1: for all $x' \in \mathbf{x}'$, $|w_{x'} - w_{x_\eta}| > |w_{x'} - w_{x_\alpha}|$, or

Case 2: there exists at least one $x' \in \mathbf{x}'$ for which $|w_{x'} - w_{x_\eta}| \leq |w_{x'} - w_{x_\alpha}|$.

We first consider Case 1. Suppose, for all $x' \in \mathbf{x}'$, we have $|w_{x'} - w_{x_\eta}| > |w_{x'} - w_{x_\alpha}|$.

Since η aligns all nodes of $\mathbf{x}'$, we have the first inequality below

$$\begin{aligned}\mathbf{cost}(\eta) &\geq \sum_{x' \in \mathbf{x}'} |w_{x'} - w_{x_\eta}| \\ &> \sum_{x' \in \mathbf{x}'} |w_{x'} - w_{x_\alpha}| \\ &= \mathbf{cost}(\alpha).\end{aligned}$$

This is a contradiction with $\mathbf{cost}(\eta) \leq \mathbf{cost}(\alpha)$.

Now, we consider Case 2. Suppose there exists $x' \in \mathbf{x}'$ for which

$$|w_{x'} - w_{x_\eta}| \leq |w_{x'} - w_{x_\alpha}|. \tag{5.4}$$

If $x_\eta = x_\alpha$ for all such $x' \in \mathbf{x}'$, then we get that either η is the same alignment as α or there must exist $y' \in \mathbf{x}'$ for which (5.4) is not true and hence $|w_{y'} - w_{y_\eta}| > |w_{y'} - w_{y_\alpha}|$. In particular, since for all instances where $|w_{x'} - w_{x_\eta}| \leq |w_{x'} - w_{x_\alpha}|$, $x_\eta = x_\alpha$, this implies $|w_{x'} - w_{x_\eta}| = |w_{x'} - w_{x_\alpha}|$. For all other $y' \in \mathbf{x}'$, we must have $|w_{y'} - w_{x_\eta}| > |w_{y'} - w_{x_\alpha}|$. Therefore in this case, $\mathbf{cost}(\eta) > \mathbf{cost}(\alpha)$. In either case, we have a contradiction with $\mathbf{cost}(\eta) \leq \mathbf{cost}(\alpha)$.

Hence, there must exist some $x' \in \mathbf{x}'$ for which (5.4) holds and $x_\eta \neq x_\alpha$.

In Lemma 5.2.3, we showed that both of the following statements are true:

(a) $|w_{x'} - w_{x_\eta}| = |w_{x'} - w_{x_\alpha}|$.

(b) There exists a $z' \in \mathbf{x}'$ for which $|w_{z'} - w_{x_\eta}| > \varepsilon$ where $\varepsilon := \|f - f'\|_\infty$.

From (a) and (b), we find that for all $x' \in \mathbf{x}'$ for which $|w_{x'} - w_{x_\eta}| \leq |w_{x'} - w_{x_\alpha}|$ and $x_\eta \neq x_\alpha$, the equality $|w_{x'} - w_{x_\eta}| = |w_{x'} - w_{x_\alpha}|$ must hold. Furthermore, there exists $z' \in \mathbf{x}'$ for which

$$|w_{z'} - w_{z_\eta}| > |w_{z'} - w_{z_\alpha}|.$$

For all remaining $y' \in \mathbf{x}'$, we have

$$|w_{y'} - w_{y_\eta}| \geq |w_{y'} - w_{y_\alpha}|.$$

Putting this all together, we obtain

$$\begin{aligned}
\mathbf{cost}(\eta) &\geq \sum_{x' \in \mathbf{x}'} |w_{x'} - w_{x_\eta}| \\
&> \sum_{x' \in \mathbf{x}'} |w_{x'} - w_{x_\alpha}| \\
&= \mathbf{cost}(\alpha),
\end{aligned}$$

which again contradicts the assumption that $\mathbf{cost}(\eta) \leq \mathbf{cost}(\alpha)$.

We conclude that any alignment that is different from the direct alignment has a higher cost. Therefore, the direct alignment is the unique optimal alignment that realizes $d_{\mathcal{B}}(\mathbf{x}, \mathbf{x}') = d_{\mathcal{B}}(B(f), B(f'))$.

□

We next prove a few lemmas that bound differences in node weights of aligned extrema.

For the rest of this subsection, we use the following assumptions and notation:

Assumptions 5.2.5 (Local Stability Assumptions). Let $F = \{f_i\}_{i=1}^n$ and $F' = \{f_i'\}_{i=1}^n$ be collections of nicely tame functions from C to $\mathbb{R}$. Furthermore, suppose f_i' is extremely close to f_i for each $i \in [n]$. Let $D = (V, E, \omega_V, \omega_E)$ and $D' = (V', E', \omega_V', \omega_E')$ be the extremal event DAGs of F and F', respectively. Let $S_\alpha = (V_\alpha, E_\alpha, \omega_\alpha, \omega_\alpha')$ be the extremal event supergraph arising from the set of alignments $\alpha = \{\alpha_i\}_{i=1}^n$ that is used to compute the extremal event DAG distance between D and D'.

Lemma 5.2.6 (Bound on Difference in Node Lives)**.** *Assume Assumptions 5.2.5. Let*

$v(i,j) \in V_\alpha$. *Then,*

$$|\omega_\alpha(v(i,j)) - \omega'_\alpha(v(i,j))| \leq \|f_i - f'_i\|_\infty .$$

Proof. By Lemma 5.2.4, the alignment α_i between $B(f_i)$ and $B(f'_i)$ is the direct alignment. In Lemma 5.1.12, we showed the absolute difference in node weights between aligned nodes is bounded by $\|f_i - f'_i\|_\infty$. Therefore,

$$|\omega_\alpha(v(i,j)) - \omega'_\alpha(v(i,j))| \leq \|f_i - f'_i\|_\infty .$$

□

From Lemma 5.2.6, we can conclude that if f' is extremely close to f, then the backbone distance between $B(f)$ and $B(f')$ is bounded by the number of extrema in f' multiplied by $\|f - f'\|_\infty$. This is because the direct alignment has a length of $B(f')$ and the absolute difference in aligned node weights for each pair is bounded by $\|f - f'\|_\infty$.

Corollary 5.2.7 (Bound on Backbone Distance for Extremely Close Functions)**.** *Let $f, f' : C \to \mathbb{R}$ be nicely tame functions such that f' is extremely close to f. Let k be the number of extrema of f'. Then,*

$$d_{\mathcal{B}}(B(f), B(f')) \leq k \|f - f'\|_\infty .$$

Next we give a bound on the absolute difference in heights of aligned extrema when every pair of functions is extremely close. In the following lemma we simplify notation as follows:

- $u_j := \alpha_{B(f_i)}(j)$
- $u'_j := \alpha_{B(f'_i)}(j)$.

Lemma 5.2.8 (Bound on Difference in Heights of Aligned Extrema)**.** *Assume Assumptions 5.2.5. Let $v(i,j)$ be a vertex in S_α. Let $\{\alpha_i\}_{i=1}^n$ be the set of backbone alignments between $B(f_i)$ and $B(f_i')$ that determines S_α. Let $u_j \in B(f_i)$ and $u_j' \in B(f_i')$. Let $(t, f_i(t))$ and $(t', f_i(t'))$ be the local extrema corresponding to u_j and u_j', respectively. Then,*

$$|f_i(t) - f_i'(t')| \leq \|f_i - f_i'\|_\infty .$$

Proof. By Lemma 5.2.4, α_i is the direct alignment for all $i \in [n]$. Recall from Construction 5.1.10, that both $(f_i(t), \zeta_{f_i}(t))$ and $(f_i'(t'), \zeta_{f_i'}(t'))$ are contained in the square centered at $(f_i(t), \zeta_{f_i}(t))$ of radius $\|f_i - f_i'\|_\infty$. Hence, $|f_i(t) - f_i'(t')| \leq \|f_i - f_i'\|_\infty$. □

We now have a bound on the maximum difference between node weights in extremal event DAGs. What remains is bounding the difference in edge weights between extremal event DAGs when each pair of functions is extremely close. Let $(v(i,k), v(j,m))$ be an edge in the extremal event supergraph. We show

$$|\omega_\alpha(v(i,k), v(j,m)) - \omega_\alpha'(v(i,k), v(j,m))| \leq \max\{\|f_i - f_i'\|_\infty , \|f_j - f_j'\|_\infty\}.$$

For Lemma 5.2.9, recall from Theorem 3.2.2 that $\varepsilon^*(t,s)$ is the infimum ε for which $\varphi_\varepsilon(t) \cap \varphi_\varepsilon(s) \neq \emptyset$. Furthermore, we simplify notation as follows:

- $u_k := \alpha_{B(f_i)}(k)$
- $u_k' := \alpha_{B(f_i')}(k)$
- $s_m := \alpha_{B(f_j)}(m)$
- $s_m' := \alpha_{B(f_j')}(m)$.

Lemma 5.2.9 (Bound on Difference of Extremal Interval Intersection Values)**.** *Assume Assumptions 5.2.5. Let $(v(i,k), v(j,m)) \in E_\alpha$ such that $i \neq j$, and all four nodes defining*

these two edges, u_k, u'_k, s_m, s'_m are not empty nodes. Suppose the extrema these nodes represent are $(t, f_i(t))$, $(s, f_j(s))$, $(t', f'_i(t'))$, and $(s', f'_j(s'))$, respectively. Then,

$$|\varepsilon^*(t,s) - \varepsilon^*(t',s')| \leq \varepsilon_{i,j}$$

where $\varepsilon_{i,j} := \max\{\|f_i - f'_i\|_\infty, \|f_j - f'_j\|_\infty\}$.

Proof. Consider the case that both $(t, f_i(t))$ and $(s, f_j(s))$ are local minima. In the case that one or both are local maxima, we replace one, or both f_i, f_j by the corresponding negative function and convert the problem to a problem about two minima. Hence, only considering the case that both are local minima is sufficient. Additionally, we omit superscripts on ε-extremal intervals to avoid notational clutter. An input of t or s indicates the ε-extremal interval is computed from f_i or f_j, respectively. An input of t' or s' indicates the ε-extremal interval is computed from f'_i or f'_j, respectively. For convenience of exposition, let $\varepsilon_i = \|f_i - f'_i\|_\infty$, $\varepsilon_j = \|f_j - f'_j\|_\infty$, and $\varepsilon_{ij} := \max\{\varepsilon_i, \varepsilon_j\}$.

Suppose $\varepsilon^*(t,s) < \varepsilon^*(t',s')$. Let $\varepsilon > \varepsilon^*(t,s)$. Then $\varphi_\varepsilon(t) \cap \varphi_\varepsilon(s) \neq \emptyset$. By Lemma 5.2.8,

$$|f_i(t) - f'_i(t')| \leq \varepsilon_i \leq \varepsilon_{i,j}.$$

Hence, $f_i(t) \leq f'_i(t') + \varepsilon_{i,j}$. Additionally, since $\|f_i - f'_i\|_\infty \leq \varepsilon_{i,j}$,

$$(f'_i - \varepsilon_{i,j})(x) \leq f_i(x) \text{ for all } x \in C.$$

These two inequalities imply $f_i(t) + \varepsilon \leq f'_i(t') + \varepsilon + \varepsilon_{i,j}$ and $(f'_i - \varepsilon - \varepsilon_{i,j})(x) \leq f_i(x) - \varepsilon$ for

all $x \in C$. Recall

$\varphi_\varepsilon(t)$ is the connected component of $(f_i - \varepsilon)^{-1}(f_i(t) + \varepsilon)$ containing t,

$\varphi_{\varepsilon+\varepsilon_{i,j}}(t')$ is the connected component of $(f_i' - \varepsilon - \varepsilon_{i,j})^{-1}(f_i'(t') + \varepsilon + \varepsilon_{i,j})$ containing t'.

Therefore, $\mathsf{left}(\varphi_{\varepsilon+\varepsilon_{i,j}}(t')) < \mathsf{left}(\varphi_\varepsilon(t))$ and $\mathsf{right}(\varphi_{\varepsilon+\varepsilon_{i,j}}(t')) > \mathsf{right}(\varphi_\varepsilon(t))$. We get $\varphi_\varepsilon(t) \subset \varphi_{\varepsilon+\varepsilon_{i,j}}(t')$. Similarly, we get $\varphi_\varepsilon(s) \subset \varphi_{\varepsilon+\varepsilon_{i,j}}(s')$. The non-empty intersection of $\varphi_\varepsilon(t) \cap \varphi_\varepsilon(s)$ implies $\varphi_{\varepsilon+\varepsilon_{i,j}}(t') \cap \varphi_{\varepsilon+\varepsilon_{i,j}}(s') \neq \emptyset$. This non-empty intersection holds true for all $\varepsilon > \varepsilon^*(t,s)$. Since $\varepsilon^*(t,s) < \varepsilon^*(t',s')$, we get

$$\varepsilon^*(t',s') \leq \varepsilon_{i,j} + \varepsilon^*(t,s).$$

Therefore,

$$\varepsilon^*(t',s') - \varepsilon^*(t,s) \leq \varepsilon_{i,j}.$$

In the case $\varepsilon^*(t',s') < \varepsilon^*(t,s)$, we get $\varphi_\varepsilon(t') \subset \varphi_{\varepsilon+\varepsilon_{i,j}}(t)$ and $\varphi_\varepsilon(s') \subset \varphi_{\varepsilon+\varepsilon_{i,j}}(s)$ by symmetry. Therefore,

$$\varepsilon^*(t,s) - \varepsilon^*(t',s') \leq \varepsilon_{i,j}.$$

Combining these two cases, we conclude

$$|\varepsilon^*(t,s) - \varepsilon^*(t',s')| \leq \varepsilon_{i,j}.$$

□

Next, we can bound the absolute difference in aligned edge weights.

Lemma 5.2.10 (Bound on Differences in Edge Weights)**.** *Assume Assumptions 5.2.5. Then,*

$$|\omega_\alpha(v(i,k),v(j,m)) - \omega_\alpha'(v(i,k),v(j,m))| \leq \max\{\|f_i - g_i\|_\infty, \|f_j - g_j\|_\infty\}.$$

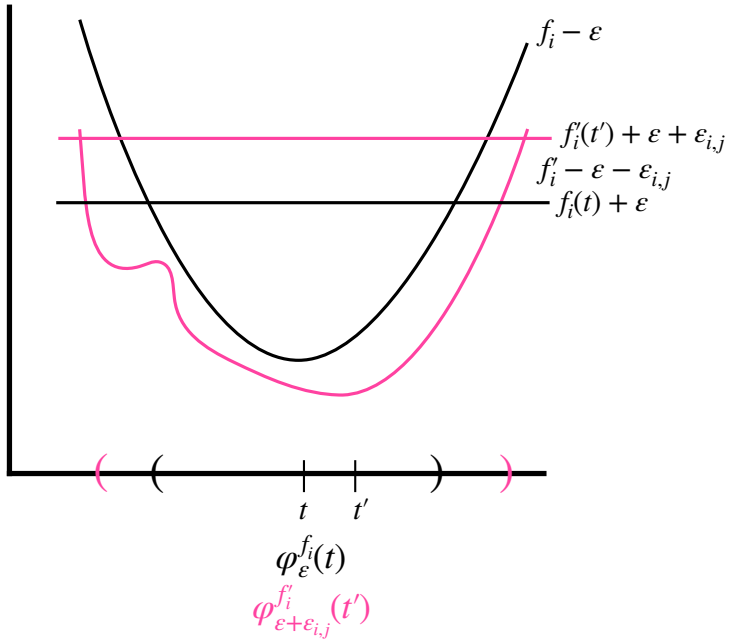


Figure 5.4: Nested ε-extremal intervals. We see that $\varphi_{\varepsilon}^{f_i}(t) \subset \varphi_{\varepsilon+\varepsilon_{i,j}}^{f_i'}(t')$.

Proof. Let $\varepsilon_{i,j} := \max\{\|f_i - g_i\|_\infty, \|f_j - g_j\|_\infty$. Let $(t, f_i(t))$, $(s, f_j(s))$, $(t', f_i'(t'))$, and $(s', f_j'(s'))$ be the local extrema corresponding to nodes $\alpha_{B(f_i)}(k)$, $\alpha_{B(f_j)}(m)$, $\alpha_{B(f_i')}(k)$, and $\alpha_{B(f_j')}(m)$, respectively. Additionally, we omit superscripts on ε-extremal intervals to avoid notational clutter. An input of t or s indicates the ε-extremal interval is computed from f_i or f_j, respectively. An input of t' or s' indicates the ε-extremal interval is computed from f_i' or f_j', respectively.

We prove this lemma by discussing several cases. First, we assume that $\omega_\alpha(v(i,k), v(j,m))$ and $\omega_\alpha'(v(i,k), v(j,m))$ are non-zero. Then, by definition,

$$\begin{aligned} E_{\textbf{diff}} &:= |\omega_\alpha(v(i,k), v(j,m)) - \omega_\alpha'(v(i,k), v(j,m))| \\ &= |\min\{\frac{1}{2}\text{pers}(t), \frac{1}{2}\text{pers}(s), \varepsilon^*(t,s)\} - \min\{\frac{1}{2}\text{pers}(t'), \frac{1}{2}\text{pers}(s'), \varepsilon^*(t',s')\}|. \end{aligned}$$

Let $\varepsilon_i = \|f_i - g_i\|_\infty$, $\varepsilon_j = \|f_j - g_j\|_\infty$, and $\varepsilon_{i,j} = \max\{\varepsilon_i, \varepsilon_j\}$. Now we begin to go

through the cases. Note that $E_{\mathbf{diff}}$ can be one of nine absolute differences depending on which value the minimum is achieved. In the cases where the difference comes from node weights of the same node or the extremal intersection values, we can apply either Lemma 5.2.6 or Lemma 5.2.9. In all other cases we split the equality $E_{\mathbf{diff}} = |U_1 - U_2|$ into two cases $E_{\mathbf{diff}} = U_1 - U_2$ or $E_{\mathbf{diff}} = U_2 - U_1$. We replace the larger term with one of the possible values from $E_{\mathbf{diff}}$ so that we can apply Lemma 5.2.6 or Lemma 5.2.9.

1. $E_{\mathbf{diff}} = \frac{1}{2}|\text{pers}(t) - \text{pers}(t')|$. Applying Lemma 5.2.6, we find

$$E_{\mathbf{diff}} = \frac{1}{2}|\text{pers}(t) - \text{pers}(t')| \leq \varepsilon_i \leq \varepsilon_{i,j}.$$

2. $E_{\mathbf{diff}} = \frac{1}{2}|\text{pers}(s) - \text{pers}(s')|$. Applying Lemma 5.2.6, we find

$$E_{\mathbf{diff}} = \frac{1}{2}|\text{pers}(s) - \text{pers}(s')| \leq \varepsilon_j \leq \varepsilon_{i,j}.$$

3. $E_{\mathbf{diff}} = |\varepsilon^*(t,s) - \varepsilon^*(t',s')|$. Applying Lemma 5.2.9, we find

$$E_{\mathbf{diff}} = |\varepsilon^*(t,s) - \varepsilon^*(t',s')| \leq \varepsilon_{i,j}.$$

4. $E_{\mathbf{diff}} = \frac{1}{2}(\text{pers}(t) - \text{pers}(s'))$. Then, $\text{pers}(t) \leq \text{pers}(s)$. Applying Lemma 5.2.6, we find

$$E_{\mathbf{diff}} = \frac{1}{2}(\text{pers}(t) - \text{pers}(s')) \leq \frac{1}{2}(\text{pers}(s) - \text{pers}(s')) \leq \varepsilon_j \leq \varepsilon_{i,j}.$$

5. $E_{\mathbf{diff}} = \frac{1}{2}(\text{pers}(s') - \text{pers}(t))$. Then, $\text{pers}(s') \leq \text{pers}(t')$. Applying Lemma 5.2.6, we find

$$E_{\mathbf{diff}} = \frac{1}{2}(\text{pers}(s') - \text{pers}(t)) \leq \frac{1}{2}(\text{pers}(t') - \text{pers}(t)) \leq \varepsilon_i \leq \varepsilon_{i,j}.$$

6. $E_{\textbf{diff}} = \frac{1}{2}(\text{pers}(s) - \text{pers}(t'))$. Then, $\text{pers}(s) \leq \text{pers}(t)$. Applying Lemma 5.2.6, we find

$$E_{\textbf{diff}} = \frac{1}{2}(\text{pers}(s) - \text{pers}(t')) \leq \frac{1}{2}(\text{pers}(t) - \text{pers}(t')) \leq \varepsilon_i \leq \varepsilon_{i,j}.$$

7. $E_{\textbf{diff}} = \frac{1}{2}(\text{pers}(t') - \text{pers}(s))$.Then, $\text{pers}(t') \leq \text{pers}(s')$. Applying Lemma 5.2.6, we find

$$E_{\textbf{diff}} = \frac{1}{2}(\text{pers}(t') - \text{pers}(s)) \leq \frac{1}{2}(\text{pers}(s') - \text{pers}(s)) \leq \varepsilon_j \leq \varepsilon_{i,j}.$$

8. $E_{\textbf{diff}} = \varepsilon^*(t,s) - \frac{1}{2}\text{pers}(t')$. Then, $\varepsilon^*(t,s) \leq \frac{1}{2}\text{pers}(t)$. Applying Lemma 5.2.6, we find

$$E_{\textbf{diff}} = \varepsilon^*(t,s) - \frac{1}{2}\text{pers}(t') \leq \frac{1}{2}(\text{pers}(t) - \text{pers}(t')) \leq \varepsilon_i \leq \varepsilon_{i,j}.$$

9. $E_{\textbf{diff}} = \frac{1}{2}\text{pers}(t') - \varepsilon^*(t,s)$. Then, $\frac{1}{2}\text{pers}(t') \leq \varepsilon^*(t',s')$. Applying Lemma 5.2.9, we find

$$E_{\textbf{diff}} = \frac{1}{2}\text{pers}(t') - \varepsilon^*(t,s) \leq \varepsilon^*(t',s') - \varepsilon^*(t,s) \leq \varepsilon_{i,j}.$$

10. $E_{\textbf{diff}} = \varepsilon^*(t,s) - \frac{1}{2}\text{pers}(s')$. Then, $\varepsilon^*(t,s) \leq \frac{1}{2}\text{pers}(s)$. Applying Lemma 5.2.6, we find

$$E_{\textbf{diff}} = \varepsilon^*(t,s) - \frac{1}{2}\text{pers}(s') \leq \frac{1}{2}(\text{pers}(s) - \text{pers}(s')) \leq \varepsilon_j \leq \varepsilon_{i,j}.$$

11. $E_{\textbf{diff}} = \frac{1}{2}\text{pers}(s') - \varepsilon^*(t,s)$. Then, $\frac{1}{2}\text{pers}(s') \leq \varepsilon^*(t',s')$. Applying Lemma 5.2.9, we find

$$E_{\textbf{diff}} = \frac{1}{2}\text{pers}(s') - \varepsilon^*(t,s) \leq \varepsilon^*(t',s') - \varepsilon^*(t,s) \leq \varepsilon_{i,j}.$$

12. $E_{\textbf{diff}} = \varepsilon^*(t', s') - \frac{1}{2}\text{pers}(t)$. Then, $\varepsilon^*(t', s') \leq \frac{1}{2}\text{pers}(t')$. Applying Lemma 5.2.6, we find

$$E_{\textbf{diff}} = \varepsilon^*(t', s') - \frac{1}{2}\text{pers}(t) \leq \frac{1}{2}(\text{pers}(t') - \text{pers}(t)) \leq \varepsilon_i \leq \varepsilon_{i,j}.$$

13. $E_{\textbf{diff}} = \frac{1}{2}\text{pers}(t) - \varepsilon^*(t', s')$. Then $\frac{1}{2}\text{pers}(t) \leq \varepsilon^*(t, s)$. Applying Lemma 5.2.9, we find

$$E_{\textbf{diff}} = \frac{1}{2}\text{pers}(t) - \varepsilon^*(t', s') \leq \varepsilon^*(t, s) - \varepsilon^*(t', s') \leq \varepsilon_{i,j}.$$

14. $E_{\textbf{diff}} = \varepsilon^*(t', s') - \frac{1}{2}\text{pers}(s)$. Then, $\varepsilon^*(t', s') \leq \frac{1}{2}\text{pers}(s')$. Applying Lemma 5.2.6, we find

$$E_{\textbf{diff}} = \varepsilon^*(t', s') - \frac{1}{2}\text{pers}(s) \leq \frac{1}{2}(\text{pers}(s') - \text{pers}(s)) \leq \varepsilon_j \leq \varepsilon_{i,j}.$$

15. $E_{\textbf{diff}} = \frac{1}{2}\text{pers}(s) - \varepsilon^*(t', s')$. Then $\frac{1}{2}\text{pers}(s) \leq \varepsilon^*(t, s)$. Applying Lemma 5.2.9, we find

$$E_{\textbf{diff}} = \frac{1}{2}\text{pers}(s) - \varepsilon^*(t', s') \leq \varepsilon^*(t, s) - \varepsilon^*(t', s') \leq \varepsilon_{i,j}.$$

Now assume one of $\omega_\alpha(v(i,k), v(j,m))$ or $\omega'_\alpha(v(i,k), v(j,m))$ is equal to zero. Without loss of generality, suppose $\omega'_\alpha(v(i,k), v(j,m)) = 0$. Then,

$$E_{\textbf{diff}} = \min\{\frac{1}{2}\text{pers}(t), \frac{1}{2}\text{pers}(s), \varepsilon^*(t, s)\}.$$

Applying Lemma 5.2.6, we find

$$\frac{1}{2}\text{pers}(t) \leq \varepsilon_i \leq \varepsilon_{i,j}, \quad \frac{1}{2}\text{pers}(s) \leq \varepsilon_j \leq \varepsilon_{i,j}.$$

If $E_{\textbf{diff}} = \varepsilon^*(t, s)$, then $\varepsilon^*(t, s) \leq \frac{1}{2}\text{pers}(t) \leq \varepsilon_i \leq \varepsilon_{i,j}$. Hence, $E_{\textbf{diff}} \leq \varepsilon_{i,j}$.

Combining all the cases, we can conclude

$$|\omega_\alpha(v(i,k),v(j,m)) - \omega'_\alpha(v(i,k),v(j,m))| \leq \varepsilon_{i,j}.$$

□

Using the bounds we established between aligned node and edge weights in the extremal event supergraph arising from extremely close functions, we can bound the extremal event DAG distance.

Theorem 5.2.11 (Extremal Event DAG Stability)**.** *Assume Assumptions 5.2.5. Let k_i be the number of extrema in f'_i. Let $\varepsilon_i := \|f_i - f'_i\|_\infty$ and $\varepsilon_{i,j} := \max\{\varepsilon_i, \varepsilon_j\}$. Let P be the set of unordered pairs between the first n positive integers. Let $S_\alpha|_{\alpha_i}$ be the restricted subgraph of S_α that is induced by α_i. Furthermore, let $i \neq j$ and denote $E_{i,j}$ to be the set of cross edges in S_α, that is, $(u,v) \in E_{i,j}$ if $u \in S_\alpha|_{\alpha_i}$ and $v \in S_\alpha|_{\alpha_j}$. Then,*

$$d_{ED}(D,D') \leq \sum_{i=1}^{n} \varepsilon_i \left(k_i + \binom{k_i}{2}\right) + \sum_{(i,j)\in P} |E_{i,j}|\varepsilon_{i,j}.$$

Proof. Let E_i be the set of edges in the extremal event supergraph restricted to the subgraph induced by α_i. Hence, for each edge $(u,v) \in E_i$, we have u, $v \in S_\alpha|_{\alpha_i}$. The extremal event DAG distance between D and D' can be expressed as the sum of three terms.

$$\begin{aligned} d_{ED}(D,D') = \sum_{i=1}^{n} d_{\mathcal{B}}(B(f_i),B(f'_i)) + \sum_{(u,v)\in E_i} |\omega_\alpha(u,v) - \omega'_\alpha(u,v)| \\ + \sum_{(u,v)\in E_{i,j}} |\omega_\alpha(u,v) - \omega'_\alpha(u,v)|. \end{aligned}$$

The first term is the sum of backbone distances between $B(f_i)$ and $B(f_i')$ for each $i \in [n]$. The second term is the sum of the absolute difference in edge weights where the nodes defining each edge are from the same backbone alignment. The third term is the sum of the absolute difference in edge weights where the nodes defining each edge are contained in different backbones.

Applying Corollary 5.2.7, we know that $d_{\mathcal{B}}(B(f_i), B(f_i')) \leq k_i\varepsilon_i$. Hence, we can bound the first term

$$\sum_{i=1}^{n} d_{\mathcal{B}}(B(f_i), B(f_i')) \leq \sum_{i=1}^{n} k_i\varepsilon_i.$$

Applying Lemma 5.2.10, we know that if $(u, v) \in E_i$, then $|\omega_\alpha(u, v) - \omega_\alpha'(u, v)| \leq \varepsilon_i$. There are $\binom{k_i}{2}$ edges in E_i. Hence, we can bound the second term by

$$\sum_{(u,v)\in E_i} |\omega_\alpha(u, v) - \omega_\alpha'(u, v)| \leq \sum_{i=1}^{n} \binom{k_i}{2}\varepsilon_i.$$

Let $|E_{i,j}|$ be the cardinality of the set $E_{i,j}$. Applying Lemma 5.2.10, we can bound the third term

$$\sum_{(u,v)\in E_{i,j}} |\omega_\alpha(u, v) - \omega_\alpha'(u, v)| \leq \sum_{(i,j)\in P} |E_{i,j}|\varepsilon_{i,j}.$$

Combining the three bounds we find that

$$d_{ED}(D, D') \leq \sum_{i=1}^{n} \left(k_i + \binom{k_i}{2}\right)\varepsilon_i + \sum_{(i,j)\in P} |E_{i,j}|\varepsilon_{i,j}.$$

□

Note that Theorem 5.2.11 requires that in collections $\{f_i\}_{i=1}^n$ and $\{f_i'\}_{i=1}^n$ each pair (f_j, f_j') is extremely close. Therefore this a local stability result. Theorem Theorem 5.1.14 offers an approach to use a local result (Lemma 5.1.13) to prove a global result. This globalization approach uses a homotopy that is sampled sufficiently densely so that each

consecutive pair satisfies the assumptions of the local result. However, the key ingredient used to aggregate the local results to a global estimate is a triangle inequality. It remains an important open question whether extremal event DAG distance satisfies the triangle inequality. If so, than a similar globalization process would yield a version of Theorem 5.2.11 without restrictions on closeness of f_j and f'_j.

CHAPTER SIX

COMPUTATIONS

We describe how to compute the extremal event DAG and extremal event DAG distance. Since experimental time series data collects discrete time points as opposed to continuous functions, we first take a detour to discuss ε-extremal intervals and their associated properties in the discrete setting.

6.1 Discrete ε-Extremal Intervals

Definition 6.1.1 (Collection of Time Series)**.** A set $\mathcal{D} = \{D_j\}_{j=1}^{K}$ is a *dataset* composed of *time series* D_j on the closed interval $C := [a_1, a_2]$ where $D_j = \{(z_i, h_i^j)\}_{j=1}^{N}$ with

$$Z := \{z_1 = a_1, z_2, \ldots, z_{N-1}, z_N = a_2\},$$

is an ordered set with $z_j < z_{j+1}$ and the heights h_j^i are the heights of the j^{th} points at z_j of time series i.

For our purposes, we assume that Z denotes the progression of time. However, the following results hold even when Z denotes some ordered quantity other than time, for example, distance.

Definition 6.1.2 (Discrete ε-Extremal Intervals)**.** Let $f_i : [a_1, a_2] \to \mathbb{R}$ be the linear interpolation of the time series D_i. Let $\varepsilon > 0$, and suppose f_i has a local extremum at t. Define the *discrete ε-extremal interval* to be a relatively open interval $d_\varepsilon^{f_i}(t) \subset [a_1, a_2]$ with endpoints in Z such that

1. $d_\varepsilon^{D_i}(t) \supset \varphi_\varepsilon^{f_i}(t)$

2. $d_\varepsilon^{D_i}(t)$ is the minimal such interval, meaning there does not exist an interval I with endpoints in Z such that $d_\varepsilon^{D_i}(t) \supsetneq I \supset \varphi_\varepsilon^{f_i}(t)$.

We note that we omit the superscript D_i from $d_\varepsilon^{D_i}$ if the identity of the time series is clear.

A few properties of ε-extremal intervals still hold in the discrete case. Namely, Propositions 1 and 2 of [7]. Proposition 1 implies that as long as ε does not exceed the node life of two local minima $(t_j, f_i(t_j))$, $(t_k, f_i(t_k))$ of D_i, then $d_\varepsilon(t_j) \cap d_\varepsilon(t_k) = \emptyset$. Proposition 2 implies that as long as ε does not exceed the node of life of the extremum at t_j, then any ε-perturbation of f_i has a local minimum contained in $d_\varepsilon(t_j)$. The ε-extremal interval property that is lost is minimality, which is Proposition 3 of [7]. This proposition states that the ε-extremal intervals are the smallest intervals to guarantee extrema of ε-perturbations of f_i, given the node life of extrema is less than ε. Since discrete ε-extremal intervals contain the ones we would get from the linear interpolations, we cannot guarantee minimality. A more thorough discussion of these properties in the discrete case can be found in Section 4.2 of [7].

In regards to properties mentioned in this paper, Statement (1) of Lemma 3.1.1 states ε-extremal intervals grow as ε increases also holds in the discrete setting. This is because the computation of the discrete ε-extremal intervals is the same as the continuous case except that the intervals are widened so that the endpoints are contained in the domain of the time series. This does not affect the monotonicity of growth in the ε-extremal intervals.

Lemma 6.1.3 (Monotonicity of $d_\varepsilon(t_i)$)**.** *Let D_j be a time series with $t_1 < t_2 < ... < t_n$ the domain coordinates of the local extrema. Then, for each $i \in [n]$, the length $\mathsf{len}(d_\varepsilon(t_i))$ is increasing with respect to ε.*

Additionally, Statement (2) of Lemma 3.1.1 holds in the discrete case. We prove that here since we apply it in Section 6.2.

Lemma 6.1.4 (Containment Property of Discrete ε-Extremal Intervals)**.** *Let D_j be a time series with domain coordinates $\{t_i\}_{i=1}^n$ for $i \in [n]$ where $t_1 < t_2 < ... < t_n$. Suppose D_j has a local minimum at t_i where $i \leq n-1$. Then,*

$$\varepsilon \leq \frac{1}{2}|f_j(t_i) - f_j(t_{i+1})| \quad \textit{if and only if} \quad t_{i+1} \notin d_\varepsilon(t_i).$$

Proof. Let t_i be the domain coordinate of a local minimum of D_j. For this proof we omit the subscript j from D_j and its linear interpolation f_j.

First assume $\varepsilon \leq \frac{1}{2}|f(t_i) - f(t_{i+1})|$. We show $t_{i+1} \notin d_\varepsilon(t_i)$. Since t_i is the domain coordinate of a local minimum, $f(t_i) < f(t_{i+1})$. Thus,

$$\varepsilon \leq \frac{1}{2}(f(t_{i+1}) - f(t_i))$$

$$f(t_i) + \varepsilon \leq f(t_{i+1}) - \varepsilon.$$

This implies $t_{i+1} \notin (f-\varepsilon)^{-1}(-\infty, f(t_i)+\varepsilon)$. Hence, $t_{i+1} \notin \varphi_\varepsilon(t_i)$. By Definition 6.1.2, the right endpoint of $d_\varepsilon(t_i)$ is equal to t_{i+1}, and the right half of this interval is open. Hence, $t_{i+1} \notin d_\varepsilon(t_i)$.

Next we prove that $t_{i+1} \notin d_\varepsilon(t_i)$ implies $\varepsilon \leq \frac{1}{2}|f(t_i) - f(t_{i+1})|$. We do this by proving the contrapositive. Assume $\varepsilon > \frac{1}{2}|f(t_i) - f(t_{i+1})|$. Since t_i is the domain coordinate of a local minimum, $f(t_i) < f(t_{i+1})$. Thus,

$$\varepsilon > \frac{1}{2}(f(t_{i+1}) - f(t_i))$$

$$f(t_i) + \varepsilon > f(t_{i+1}) - \varepsilon.$$

This implies $t_{i+1} \in (f-\varepsilon)^{-1}(-\infty, f(t_i)+\varepsilon)$. Furthermore, t_{i+1} must be in the same connected

component as t_i in $(f-\varepsilon)^{-1}(-\infty, f(t_i)+\varepsilon)$ since t_{i+1} is adjacent to t_i. Therefore, $t_{i+1} \in \varphi_\varepsilon(t_i)$. By Definition 6.1.2, $d_\varepsilon(t_i) \supset \varphi_\varepsilon(t_i)$. Therefore, $t_{i+1} \in d_\varepsilon(t_i)$. □

Applying a symmetric argument as in Lemma 6.1.4, we see that $\varepsilon \leq \frac{1}{2}|f_j(t_i) - f_j(t_{i-1})|$ if and only if $t_{i-1} \notin d_\varepsilon(t_i)$ and $i \geq 2$.

In discrete time series, the idea of incomparability is reduced to intersections between a finite number of intervals. Generally speaking, any continuous, monotone interpolation of a time series giving rise to ε-extremal intervals φ_ε results in greater values of ε^* than the discrete intervals d_ε. The ε^* determined by d_ε is the infinimum of values of ε^* over all smooth, monotone interpolations of the discrete time series. Therefore ε^* determined by d_ε is the most conservative estimate of incomparability available from the time series information.

Lastly, we remark that local stability of the extremal event DAG distance extends to the discrete case. To see this, note the node lives of extrema in discrete functions can be computed from the sublevel set persistence diagram obtained through linearly interpolating the values of the discrete function. Therefore, Lemma 5.2.2, Lemma 5.2.4, Lemma 5.2.6, Corollary 5.2.7, Lemma 5.2.8 that are all statements about node lives and differences in aligned node lives all extend to the discrete case. Furthermore, the proof of Lemma 5.2.9 (that bounds the difference between ε-extremal intersection values between aligned edge weights) relies on the nesting property of ε-extremal intervals. This property still holds in the discrete case and so Lemma 5.2.9 also extends in the discrete case. The proofs of Lemma 5.2.10 and Theorem 5.2.11 use the aforementioned lemmas. Therefore local stability for extremal event DAGs holds for discrete time series.

6.2 Computing the Extremal Event DAG

In this section, we describe Algorithm 6.5, which computes the extremal event DAG from a collection of time series over a closed interval. This algorithm is based on two key

insights. First, the edge weight between two local extrema from the same function is the minimum of the node lives of the two local extrema (Statement (1) of Theorem 3.2.2). Second, the edge weight between two local extrema from different functions is the minimum of the two node lives and the infimum ε for when the two ε-extremal intervals intersect (Statement (2) of Theorem 3.2.2). We first describe the computation of merge trees that are used to compute the node lives of local extrema.

6.2.1 Merge Trees and Node Lives.

The merge tree captures the connectivity of sublevel sets of a function. The information we get from the merge tree of a function is very similar to information we capture from the zeroth-dimensional persistence diagram from a sublevel set filtration of f.

Definition 6.2.1 (Merge Tree)**.** Let f be a real-valued function. Let $\Gamma(f)$ be the graph of f. We declare $x \sim y$ if there exists an $h \in \mathbb{R}$ for which $x, y \in f^{-1}(h)$ and x, y are in the same connected component of $f^{-1}(-\infty, h]$. The *merge tree of* f, denoted M_f, is defined to be the quotient space

$$M_f := \Gamma(f)/\sim .$$

Given a nicely tame function, $f : C \to \mathbb{R}$, we construct the structure of the merge tree, $G = (V, E)$, by [76]. This structure consists of a list of *merge triplets* where each triplet is a three-tuple of real numbers (u, s, v) such that u represents the connected component containing itself for $a \in [f(u), f(s))$ and v becomes the representative of the connected component at a height of $f(s)$.

In relation to zeroth-dimensional persistence diagram from a sublevel set filtration, $(f(u), f(s))$, is a birth-death pair and the two connected components represented by u and v merge into the connected component represented by v at a height of $f(s)$. If the merge triplet has identical components denoted as (u, u, u), then u is the global minimum of its connected component in G. The time complexity of computing the Merge Tree using

Kruskal's algorithm for a function represented as a graph, $G = (V, E)$ is $O(m \log n)$ where $n = |V|$ and $m = |E|$ [76]. In our setting, the number of edges is bounded by n and so the time complexity is $O(n \log n)$.

We describe how we compute the node lives and node labels of the extremal event DAG, using the triplets computed from the merge trees. Algorithm 6.1 (GetMinLives) takes input a merge tree M for a time series $D = \{(z_i, h_i)\}_{i=1}^N$. Algorithm 6.1 outputs the node lives of the local minima of D. For each merge triplet with distinct components, (z_i, z_j, z_k) in M, (h_i, h_j) is a point in the zeroth-dimensional persistence diagram from the sublevel set filtration. By Definition 2.5.3, we get that $\frac{1}{2}\text{pers}_D(z_i) = (h_j - h_i)/2$. If the merge triplet has all identical components, (z_j, z_j, z_j), then $\frac{1}{2}\text{pers}_D(z_j) = \frac{1}{2}(\max(\{h_i\}_{i=1}^N) - \min(\{h_i\}_{i=1}^N))$. Applying one of these two computations to all merge triplets computes node lives for all local minima in D. To compute the node lives of the local maxima, we apply the same process to $-D$, where we take the negative of all heights h_i.

Algorithm 6.1: GetMinLives(D, M)

Input: Array of merge tree triplets and time series $D = \{(z_i, h_i)\}_{i=1}^N$.

Output: Dictionary of node lives for each point in curve.

```
minlives ← Initialize dictionary keyed by locations of extrema
for (z_i, z_j, z_k) ∈ M do
    if z_j = z_k then
        minlives(z_i) ← (max({h_i}_{i=1}^N) − min({h_i}_{i=1}^N))/2
    else
        minlives(z_i) ← |h_j − h_i|/2
    end if
end for
return minlives
```

The algorithm GetNodeLives($D, M_{\min}, M_{\max}$) applies GetMinLives twice for both D and $-D$ with merge trees $M_{\min}$ and $M_{\max}$ respectively. Suppose D has n local extrema. Each line in Algorithm 6.1 takes constant time. Since the loop has at most n iterations, then the total time complexity of Algorithm 6.1 is $O(n)$. Hence, the time complexity of GetNodeLives is also $O(n)$.

6.2.2 Computing Edge Weights.

We explain how to compute the edge weights of the extremal event DAG. First consider the edge weight between two nodes in $B(f)$, a single backbone. By Statement (1) of Theorem 3.2.2, we know the edge weight is the minimum node life between the two extrema. Computing the minimum between two values takes constant time.

Next, we describe how we compute the edge weight between two nodes from different backbones. In order to do this, we must first compute the infimum ε for which two ε-extremal intervals intersect. In the discrete setting, the growth of the ε-extremal intervals only change at a finite number of ε. We refer to the ε values where discontinuous changes in length occur as *jumps*. Algorithm 6.2 (GetEpsJumpsRight) computes the ε jumps for the right endpoint of an ε-extremal interval. We recall in the discrete setting, connected components are determined by the linear interpolation of points in the image of f. In Lemma 6.2.2, we use $\sim$ to denote the equivalence relation given by connected components i.e., for a time series $D_i = \{(z_j, h_j^i)\}_{j=1}^N$, and $z_j, z_k \in Z$, we declare $z_j \sim z_k$ at $\varepsilon > 0$ if both z_j, z_k are contained in the same connected component of $(f_i - \varepsilon)^{-1}(-\infty, f_i(x) + \varepsilon)$ where f_i is the linear interpolation of D_i and $x \in Z$.

Lemma 6.2.2 (Correctness of Algorithm 6.2)**.** *Let $D = \{(z_i, h_i)\}_{i=1}^N$ be a time series. Let z_i be the domain coordinate of a local extremum of D. Let $r_i(\varepsilon) : \mathbb{R}_{>0} \to \mathbb{R}$ denote the right endpoint of $d_\varepsilon(z_i)$. Then,* GetEpsJumpsRight(D, z_i) *Algorithm 6.2 returns all ε values for which r_i has a jump discontinuity.*

Algorithm 6.2: GetEpsJumpsRight(D, z_i)

Input: Time series $D = \{(z_i, h_i)\}_{i=1}^{N}$ and domain point $z_i \in Z$.

Output: Vector of real numbers indicating values for which the right endpoint of $d_\varepsilon(z_i)$ jumps.

Initialize epsilon array and jump height
if $i \neq N$ **then**
    epsilons $\leftarrow \langle |h_{i+1} - h_i|/2 \rangle$
    levelheight $\leftarrow h_{i+1}$
    for $j \leftarrow (i+2) \ldots N$ **do**
        if (extremum$(z_i) = \min \wedge h_j \geq$ levelheight)
        $\bigvee$ (extremum$(z_i) = \max \wedge h_j \leq$ levelheight) **then**
            epsilons **append** $|h_j - h_i|/2$
            levelheight $\leftarrow h_j$
        end if
    end for
end if
return epsilons

Proof. We note that if $i = N$, then $r_i(\varepsilon) = z_i$ for all $\varepsilon \geq 0$. Hence, there are no jumps. GetEpsJumpsRight returns the empty array, which is correct. For the rest of this proof, we assume $i \neq N$. To prove correctness, we show that we have the following loop invariant. At the start of each iteration of the for loop, the array epsilons consists of jumps of $r_i(\varepsilon)$ in sorted order. We first remark that jump discontinuities of $r_i(\varepsilon)$ occur at the infimum ε for which a point $z_j \in Z$ is contained in $d_\varepsilon(z_i)$ with $j > i$. This is because in the discrete setting, the ε-extremal intervals only grow when a new point in Z is contained in $d_\varepsilon(z_i)$. In particular, we show the loop invariant that $z_{i+j} \in d_{\varepsilon'}(z_i)$, where ε' is the maximum of the array named epsilons after the j^{th} iteration.

Initialization First, consider $\varepsilon_1 = |h_i - h_{i+1}|/2$. By Lemma 6.1.4, $z_{i+1} \notin d_{\varepsilon_1}(z_i)$ and for all $\varepsilon > \varepsilon_1$, $z_{i+1} \in d_\varepsilon(z_i)$. Hence, ε_1 is the infimum ε for which $z_{i+1} \in d_\varepsilon(z_i)$. This implies that ε_1 is a jump discontinuity of r_i. From Lemma 6.1.3, we know that $d_\varepsilon(z_i)$ increases monotonically. Hence no other point in Z is contained in $d_\varepsilon(z_i)$ at a smaller value of ε. Therefore, ε_1 is the smallest jump discontinuity of $r_i(\varepsilon)$ and so the loop invariant holds before the first iteration.

Maintenance Assume the loop invariant holds after the j^{th} iteration. We show it also holds after the $j + 1^{st}$ iteration. First assume z_i is a local minimum. Then,

$$\text{levelheight} = \max\{h_k \mid k \in [i+1, i+j]\}.$$

Denote $z_* := z_{i+(j+1)}$. We want to find the infimum $\varepsilon > 0$ for which $z_* \in d_\varepsilon(z_i)$.

Suppose $h_* <$ levelheight. We claim that $z_* \in d_{\varepsilon'}(z_i)$ where $\varepsilon' := \max\{\text{epsilons}\}$. This means that no new ε value needs to be added to the epsilons vector in Algorithm 6.3. Let

$$z' := \text{argmax}\{h_k \mid k \in [i+1, i+j]\}.$$

Thus $h' =$ levelheight. Since $h_* <$ levelheight $= h'$ and $\varepsilon' = (h' - h_i)/2$, then

$$h_* - h_i < h' - h_i = 2\varepsilon'$$

$$h_* - \varepsilon' < h_i + \varepsilon'$$

This implies that $z_* \in (f - \varepsilon')^{-1}(-\infty, f(z_i) + \varepsilon')$ (recall f is the linear interpolation of D). Observe, $z_{i+j} \in d_{\varepsilon'}(z_i)$ by the assumption that the loop invariant holds at the j^{th} iteration. Since z_* is adjacent to z_{i+j}, z_* must be in the same connected component of $(f - \varepsilon')^{-1}(-\infty, f(z_i) + \varepsilon')$ as z_i, i.e., $z_* \sim z_i$. Therefore, $z_* \in d_{\varepsilon'}(z_i)$ and the loop invariant holds.

Next suppose $h_* \geq$ levelheight. Let ε' be as before. Applying a similar computation as above we find that $z_* \notin (f - \varepsilon')^{-1}(-\infty, f(z_i) + \varepsilon')$. Observe that at $\varepsilon^* := (h_* - h_i)/2$ we have

$$f(z_*) - \varepsilon^* = f(z_i) + \varepsilon^*.$$

This leads to the observation that $z_* \in (f - \varepsilon)^{-1}(-\infty, h_i + \varepsilon)$ for any $\varepsilon > \varepsilon^*$ by Lemma 6.1.3. Since $\varepsilon^* > \varepsilon'$, $z_{i+j} \in (f - \varepsilon)^{-1}(-\infty, f(z_i) + \varepsilon)$ as well. Since $z_i \sim z_{i+j}$ by the loop invariant and $z_{i+j} \sim z_*$ by adjacency, $z_i \sim z_*$ as desired for any $\varepsilon > \varepsilon^*$. These two observations tell us that ε^* is the infimum ε for which $z_* \in d_\varepsilon(z_i)$. Therefore, ε^* should indeed be added to the epsilons array and is larger than all other values in the array. By assumption, the epsilons array is sorted. Hence, the loop invariant holds.

In the case that (z_i, h_i) is a local maximum, we apply a symmetric argument by noting that $d_\varepsilon(z_i)$ is the (expanded out) connected component of $(f + \varepsilon)^{-1}(f(z_i) - \varepsilon, \infty)$ containing z_i and a new value is added to the ε vector if $h_j \leq$ levelheight.

End Note that the for loop terminates since there are only a finite number of iterations.

Since the number of jumps of $r_i(\varepsilon)$ is bounded by $N - i + 1$, and epsilons consists of all

infimum ε for which a point in N and greater than z_i is contained in $d_\varepsilon(z_i)$, then Algorithm 6.2 is correct. □

Next, we analyze the time complexity of Algorithm 6.2. Every line takes constant time. Since the for loop (Line 5-Line 11) has at most $N-1$ iterations, then the total time complexity of Algorithm 6.2 is $O(N)$.

Furthermore, we can apply the same algorithm but with going through points on the left of z_i to find all the points for which the left endpoint of $d_\varepsilon(z_i)$ changes. Note that there are $N-1$ points to the left and right of z_i combined, and so finding all ε-jumps of $d_\varepsilon(z_i)$ takes $O(N)$. We call this combined function, GetEpsJumps (Algorithm 6.3).

Algorithm 6.3: GetEpsJumps(D, z_i)

Input: Time series $D = \{(z_i, h_i)\}_{i=1}^{N}$ and domain point $z_i \in Z$.

Output: Vector of real numbers indicating values for which the left or right endpoint of $d_\varepsilon(z_i)$ jumps.

1: Initialize epsilon array and jump height
2: epsilons $\leftarrow$ {GetEpsJumpsLeft(D, z_i)}
3: epsilons **append** GetEpsJumpsRight(D, z_i)
4: **return** epsilons

Lastly, to find the infimum ε for which two ε-extremal intervals intersect, we apply Algorithm 6.4. Algorithm 6.4 takes input of two time series D_j, D_k, two domain coordinates of local extrema of D_j, D_k, and merge trees of $D_j, -D_j, D_k, -D_k$. Algorithm 6.3 (GetEpsJumps) is applied to both functions and extrema. Then, Algorithm 6.4 goes through all jumps in order to find the smallest one for which the two ε-extremal intervals intersect.

Algorithm 6.4: GetEpsIntersection(D_j, D_k, z_j, z_k, M_{D_j}, M_{-D_j}, M_{D_k}, M_{-D_k})

Input: Time series $D_j = \{(z_i, h_i^j)\}_{i=1}^N$, $D_k = \{(z_i, h_i^k)\}_{i=1}^N$ and domain points of extrema in D_j and D_k, denoted $z_j,\ z_k \in Z$. M_{D_j}, M_{-D_j}, M_{D_k}, M_{-D_k} are merge trees of $D_j, -D_j, D_k, -D_k$ respectively.

Output: $\inf_\varepsilon d_\varepsilon^{D_j}(z_j) \cap d_\varepsilon^{D_k}(z_k) \neq \emptyset$

Initialize epsilon array
epsilons $\leftarrow \{$GetEpsJumps$(D_j, z_j)\}$
epsilons **append** GetEpsJumps(D_k, z_k)
sort epsilons
for $\varepsilon \in$ epsilons **do**
    $d_\varepsilon^{D_j}(z_j) \leftarrow$ GetExtremalInterval$(D_j, z_j, M_{D_j}, M_{-D_j}, \varepsilon)$
    $d_\varepsilon^{D_k}(z_k) \leftarrow$ GetExtremalInterval$(D_k, z_k, M_{D_k}, M_{-D_k}, \varepsilon)$
    if $d_\varepsilon^{D_j}(z_j) \cap d_\varepsilon^{D_k}(z_k) \neq \emptyset$ **then**
        return ε
    end if
end for

Next, we discuss the time complexity of Algorithm 6.4. The number of jumps is bounded by the number of points in the domains of the two functions. Additionally, at each jump, we compute GetExtremalInterval for the two extrema, where the computation of the discrete extremal intervals are discussed in [7] and implemented in [4]. In particular, the time complexity for computing GetExtremalInterval(D, z_i, M_D, M_{-D}) is $O(N)$. This is because computing the ε-extremal interval requires evaluating f at points in Z near z_i. In summary

- Line 2 and Line 3 each take $O(N)$.
- Line 4 takes $O(N \log N)$.
- Line 6 and Line 7 take $O(N)$.
- All other lines take constant time.
- The number of iterations of the for loop in Line 5-Line 11 is bounded by $2N$.

All together we compute the time complexity of GetEpsIntersection$(D_j, D_k, z_j, z_k, M_{D_j}, M_{-D_j}, M_{D_k}, M_{-D_k})$ as

$$2O(N) + O(N \log N) + 2N(O(N)) = O(N \log N) + O(N^2).$$

6.2.3 Computing Extremal Event DAG

Algorithm 6.5 computes the extremal event DAG from a collection of time series $\mathcal{D} = \{D_j\}_{j=1}^K$. Algorithm 6.5 uses previously defined algorithms and functions from this section along with InitializeGraph. This algorithm is designed and implemented in [4]. InitializeGraph takes a collection of time series $\mathcal{D}$ as input and outputs (T, H, V, E) where V, E are vertices and directed edges of the extremal event DAG of $\mathcal{D}$, T is the domain coordinates of the local extrema, and H is the heights of local extrema. This function checks

through all points in Z for extrema to record as vertices and then goes through all vertex pairs to check for edges. Let $N = |Z|$. The number of vertices is bounded by NK and the number of edges is bounded by $\binom{NK}{2}$. Hence, the time complexity of InitializeGraph is

$$O(NK + \frac{(NK)(NK-1)}{2}) = O((NK)^2).$$

The correctness of Algorithm 6.5 follows from the correctness of all our other previously defined algorithms. Next we analyze the time complexity.

- Line 1 and Line 2 each take $KO(N \log N)$.
- Line 3 takes $KO(N)$.
- Initializing the extremal event DAG in Line 4 takes $O((NK)^2)$.
- Computing all the node weights in Line 7-Line 12 has a time complexity of $O(NK)$.
- Each iteration in the for loop between Line 13 - Line 21 takes at most $O(N \log N) + O(N^2)$. Since the number of vertices is bounded above by NK, then the number of edges is bounded above by $\binom{NK}{2} = \frac{(NK)(NK-1)}{2}$. Thus, the number of iterations of this for loop is bounded by $\frac{(NK)(NK-1)}{2}$.

In total, we get the time complexity to be

$$KO(N \log N) + KO(N) + O((NK)^2) + O(NK) + \frac{(NK)(NK-1)}{2}(O(N \log N) + O(N^2))$$
$$= O(N^3K^2 \log N) + O(N^4K^2).$$

6.3 Computing Optimal Backbone Alignments

To compute a distance between extremal event DAGs, we align backbones in the extremal event DAGs in an optimal manner. Here, we describe how the alignment is

Algorithm 6.5: GetExtremalEventDAG($\mathcal{D}$)

Input: A collection of time series $\mathcal{D} = \{D_j\}_{j=1}^{K}$.

Output: The extremal event DAG of $\mathcal{D}$.

$M_{D_j} \leftarrow$ merge tree for D_j

$M_{-D_j} \leftarrow$ merge tree for $-D_j$

$\text{NodeLives}_j \leftarrow \text{GetNodeLives}(D_j, M_{D_j}, M_{-D_j})$

$(T, V, E) \leftarrow \text{InitializeGraph}(\mathcal{D})$ Initialize unweighted extremal event DAG.

Initialize function $\omega_V \colon V \to \mathbb{R}$ with all values set to zero.

Initialize function $\omega_E \colon E \to \mathbb{R}$ with all values set to zero.

for $v \in V$ **do**

    if $(T(v), H(v)) \in D_j$ **then**

        $\omega_V(v) \leftarrow \text{NodeLives}_j(v)$

    end if

end for

for $e = (u, v) \in E$ **do**

    if $(T(u), H(u)), (T(v), H(v)) \in D_j$ **then**

        $\omega_E(e) \leftarrow \min(\text{NodeLives}_j(u), \text{NodeLives}_j(v))$

    else if $(T(u), H(u)) \in D_j, (T(v), H(v)) \in D_k$ **then**

        $\varepsilon \leftarrow \text{GetEpsIntersection}(D_j, D_k, T(u), T(v), M_{D_j}, M_{-D_j}, M_{D_k}, M_{-D_k})$

        $\omega_E(e) \leftarrow \min(\text{NodeLives}_j(u), \text{NodeLives}_k(v), \varepsilon)$

    end if

end for

return $(V, E, \omega_V, \omega_E)$

computed and prove that the alignment is optimal. Recall, that α denotes an alignment between two backbones (Definition 4.1.4).

Definition 6.3.1 (Alignment Matrix)**.** Let $\mathbf{x} = (x_1, x_2, \ldots, x_m)$, $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ be backbones. Note that each x_i can be written as the pair $x_i = (s_{\mathbf{x},i}, w_{\mathbf{x},i})$; likewise each y_i can be written as the pair $y_i = (s_{\mathbf{y},i}, w_{\mathbf{y},i})$. The *alignment matrix*, denoted **mat**, is an $(m+1) \times (n+1)$ matrix recursively defined as follows:

$$\mathbf{mat}[i,j] = \begin{cases} 0 & i = j = 1 \\ \sum_{k=1}^{i-1} w_{\mathbf{x},k}, & i > 1, j = 1 \\ \sum_{k=1}^{j-1} w_{\mathbf{y},k}, & i = 1, j > 1 \\ \min \left\{ \begin{array}{l} \mathbf{mat}[i-1,j] + w_{\mathbf{x},i-1} \\ \mathbf{mat}[i,j-1] + w_{\mathbf{y},j-1} \\ \mathbf{mat}[i-1,j-1] + \mathbf{diff}(x_{i-1}, y_{j-1}) \end{array} \right\} & \text{otherwise,} \end{cases}$$

where $\mathbf{diff} : \mathbf{x} \times \mathbf{y} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ is defined by

$$\mathbf{diff}\left((s_x, w_x), (s_y, w_y)\right) = \begin{cases} |w_x - w_y|, & \text{if } s_x = s_y \\ \infty, & \text{otherwise} \end{cases}.$$

Next we note that the bottom right entry in the alignment matrix is the minimum cost of aligning two backbones $\mathbf{x}$ and $\mathbf{y}$. This follows from [86]. Recall the definition of the cost function in Definition 4.1.6.

Proposition 6.3.2 (Alignment Matrix Finds Minimum Cost)**.** *Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ be backbones. Let **mat** be the $(m+1) \times (n+1)$ alignment matrix. Then, $\boldsymbol{mat}[m+1, n+1] = c_{\boldsymbol{x},\boldsymbol{y}}(m, n)$.*

Proof. For $i \in [n]$, let $x_i = (s_{\mathbf{x},i}, w_{\mathbf{x},i})$ and $y_i = (s_{\mathbf{y},i}, w_{\mathbf{y},i})$.

We proceed by induction. For the base case, first observe that $c_{\mathbf{x},\mathbf{y}}(1,0) = w_{\mathbf{x},1}$ and $c_{\mathbf{x},\mathbf{y}}(0,1) = w_{\mathbf{y},1}$. By construction, $\mathbf{mat}[2,1] = w_{\mathbf{x},1} = c_{\mathbf{x},\mathbf{y}}(1,0)$ and $\mathbf{mat}[1,2] = w_{\mathbf{y},1} = c_{\mathbf{x},\mathbf{y}}(0,1)$. Next consider $c_{\mathbf{x},\mathbf{y}}(1,1)$. The possible alignments (see Definition 4.1.4) of $\mathbf{x}_1 := \mathbf{x}[1:1]$ and $\mathbf{y}_1 := \mathbf{y}[1:1]$ are

1. $\alpha_1 : \{1,2\} \to \widetilde{\mathbf{x}}_1 \times \widetilde{\mathbf{y}}_1$, where $\alpha_1(1) = (x_1, \mathbf{0})$ and $\alpha_1(2) = (\mathbf{0}, y_1)$.
2. $\alpha_2 : \{1,2\} \to \widetilde{\mathbf{x}}_1 \times \widetilde{\mathbf{y}}_1$, where $\alpha_2(1) = (\mathbf{0}, y_1)$ and $\alpha_2(2) = (x_1, \mathbf{0})$.
3. $\alpha_3 : \{1\} \to \widetilde{\mathbf{x}}_1 \times \widetilde{\mathbf{y}}_1$, where $\alpha_3(1) = (x_1, y_1)$ is a possible alignment.

Observe $\mathbf{cost}(\alpha_1) = \mathbf{cost}(\alpha_2) = w_{\mathbf{x},1} + w_{\mathbf{y},1}$ and $\mathbf{cost}(\alpha_3) = \mathbf{diff}(x_1, y_1) = |w_{\mathbf{x},1} - w_{\mathbf{y},1}|$. Therefore,

$$c_{\mathbf{x},\mathbf{y}}(1,1) = \min \begin{cases} w_{\mathbf{x},1} + w_{\mathbf{y},1} \\ \mathbf{diff}(x_1, y_1) \end{cases}$$

By construction,

$$\mathbf{mat}[2,2] = \min \begin{cases} \mathbf{mat}[2,1] + w_{\mathbf{y},1} \\ \mathbf{mat}[1,2] + w_{\mathbf{x},1} \\ \mathbf{mat}[1,1] + \mathbf{diff}(x_1, y_1) \end{cases}$$

Substituting $w_{\mathbf{x},1}$ for $\mathbf{mat}[2,1]$, $w_{\mathbf{y},1}$ for $\mathbf{mat}[1,2]$ and zero for $\mathbf{mat}[1,1]$, we find $c_{\mathbf{x},\mathbf{y}}(1,1) = \mathbf{mat}[2,2]$. This shows the base case holds.

For the induction hypothesis we assume that $\mathbf{mat}[h,k] = c_{\mathbf{x},\mathbf{y}}(h-1, k-1)$ for all $h \leq i$ and $k \leq j$ where $i \leq m$ and $j \leq n$.

In the induction step, we show $\mathbf{mat}[i+1,j] = c_{\mathbf{x},\mathbf{y}}(i, j-1)$, $\mathbf{mat}[i,j+1] = c_{\mathbf{x},\mathbf{y}}(i-1, j)$, and $\mathbf{mat}[i+1, j+1] = c_{\mathbf{x},\mathbf{y}}(i,j)$. First consider $c_{\mathbf{x},\mathbf{y}}(i, j-1)$. Let $\alpha : [k] \to \widetilde{\mathbf{x}}[1:i] \times \widetilde{\mathbf{y}}[1 : j-1]$ be an alignment of the first i nodes of $\mathbf{x}$ with the first $j-1$ nodes of $\mathbf{y}$ with cost $c_{\mathbf{x},\mathbf{y}}(i, j-1)$. Consider the last pair of nodes aligned via $\alpha(k)$. The cost of these two nodes

is either (a) the cost of x_i aligned with an insertion, (b) the cost of y_{j-1} aligned with an insertion, or (c) the cost of x_i aligned with y_{j-1}. Note, by Definition 4.1.4, we never have an insertion aligned with an insertion. Since the cost is the minimum across these three possibilities, the cost is

$$c_{\mathbf{x},\mathbf{y}}(i, j-1) = \min \begin{cases} c_{\mathbf{x},\mathbf{y}}(i-1, j-1) + w_{\mathbf{x},i} \\ c_{\mathbf{x},\mathbf{y}}(i, j-2) + w_{\mathbf{y},j-1} \\ c_{\mathbf{x},\mathbf{y}}(i-1, j-2) + \mathbf{diff}(x_i, y_{j-1}) \end{cases}$$

Applying the induction hypothesis, we find

$$c_{\mathbf{x},\mathbf{y}}(i, j-1) = \min \begin{cases} \mathbf{mat}[i, j] + w_{\mathbf{x},i} \\ \mathbf{mat}[i+1, j-1] + w_{\mathbf{y},j-1} \\ \mathbf{mat}[i, j-1] + \mathbf{diff}(x_i, y_{j-1}) \end{cases}$$

By construction of **mat** (Definition 6.3.1), we see that $c_{\mathbf{x},\mathbf{y}}(i, j-1) = \mathbf{mat}[i+1, j]$. Using a similar approach, we find $\mathbf{mat}[i, j+1] = c_{\mathbf{x},\mathbf{y}}(i-1, j)$, and $\mathbf{mat}[i+1, j+1] = c_{\mathbf{x},\mathbf{y}}(i, j)$.

This concludes the induction argument. Thus, $\mathbf{mat}[i, j] = c_{\mathbf{x},\mathbf{y}}(i-1, j-1)$ for all $i \leq m+1$ and $j \leq n+1$. In particular, $\mathbf{mat}[m+1, n+1] = c_{\mathbf{x},\mathbf{y}}(m, n)$. □

6.3.1 Finding Optimal Alignment from Alignment Matrix

Definition 6.3.3 (Path)**.** Let M be an $m \times n$ matrix with real valued entries. A *path in M* is an injective function $p : [k] \to M$ such that $p(i)$ and $p(i+1)$ are adjacent values in a row, column, or diagonal for all $i \in \{1, 2, \ldots, k-1\}$.

To find an optimal alignment from the alignment matrix we construct a path via *backtracking*.

Path via Backtracking. Let $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ be backbones. Let **mat** be the corresponding alignment matrix. We construct a path p in **mat** recursively as follows:

- $p(1) = \mathbf{mat}[m+1, n+1]$
- If $p(h) = \mathbf{mat}[i, j]$ for $h \geq 1$ and $i, j > 1$, then,

$$p(h+1) = \begin{cases} \mathbf{mat}[i-1, j] \text{ if } \mathbf{mat}[i, j] = \mathbf{mat}[i-1, j] + w_{\mathbf{x}, i-1} \\ \mathbf{mat}[i, j-1] \text{ if } \mathbf{mat}[i, j] = \mathbf{mat}[i, j-1] + w_{\mathbf{y}, j-1} \\ \mathbf{mat}[i-1, j-1] \text{ if } \mathbf{mat}[i, j] = \mathbf{mat}[i-1, j-1] + \mathbf{diff}(x_{i-1}, y_{j-1}) \end{cases}$$

  If multiple of the conditions hold, then define $p(h+1)$ to be any *one* of them. We call p a *backtracking* path.

In summary, we are undoing the matrix construction to figure out which matrix entries lead to the cost in $\mathbf{mat}[m+1, n+1]$. Once we apply backtracking, we have at least one path from $\mathbf{mat}[m+1, n+1]$ to $\mathbf{mat}[1, 1]$. We remark that backtracking is well-defined. For any entry $p(h) = \mathbf{mat}[i, j]$, one of the three upper left entries ($\mathbf{mat}[i-1, j]$, $\mathbf{mat}[i, j-1]$, $\mathbf{mat}[i-1, j-1]$) equals $p(h+1)$ by construction of the alignment matrix (Definition 6.3.1). Since we have a finite matrix, we eventually end at $\mathbf{mat}[1, 1]$. We note that a path constructed from backtracking is not necessarily unique. For describing the alignment from a backtracking path $p : [k] \to \mathbf{mat}$, we consider the *reverse path* $p' : [k] \to \mathbf{mat}$ where $p'(i) = p(k - (i-1))$.

Alignment from Backtracking. Let $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ be backbones. Let $\mathbf{mat}$ be the corresponding alignment matrix. Let $p : [k] \to \mathbf{mat}$ be a path computed from backtracking and $p' : [k] \to \mathbf{mat}$ be the reverse path. We construct an alignment $\alpha : [k-1] \to \widetilde{\mathbf{x}} \times \widetilde{\mathbf{y}}$ such that

$$\alpha(h) = \begin{cases} (x_i, \mathbf{0}) & \text{if } p'(h) = \mathbf{mat}[i, j] \text{ and } p'(h+1) = \mathbf{mat}[i+1, j] \\ (\mathbf{0}, y_j) & \text{if } p'(h) = \mathbf{mat}[i, j] \text{ and } p'(h+1) = \mathbf{mat}[i, j+1] \\ (x_i, y_j) & \text{if } p'(h) = \mathbf{mat}[i, j] \text{ and } p'(h+1) = \mathbf{mat}[i+1, j+1] \end{cases} .$$

In other words, the following moves of p' through the matrix $\mathbf{mat}$ mean:

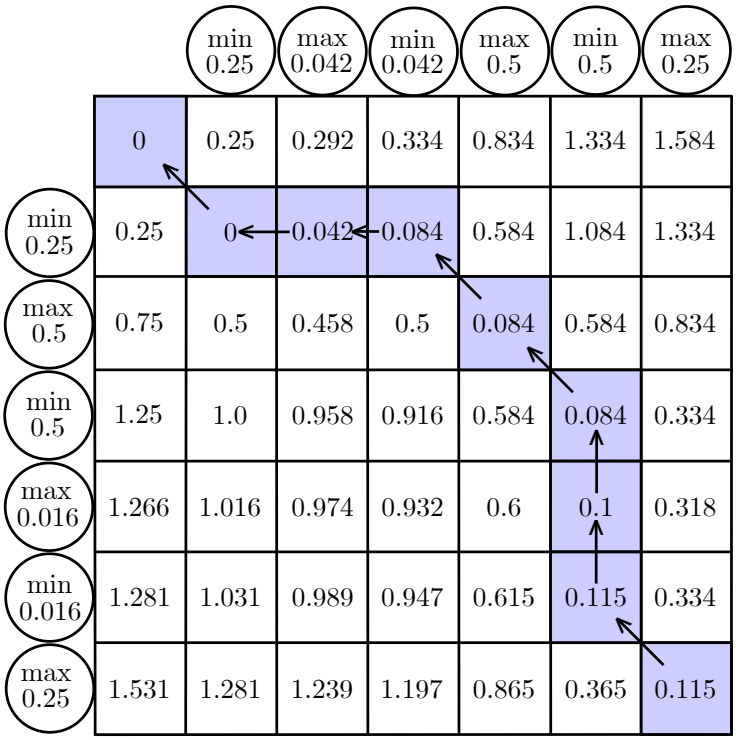


Figure 6.1: Backtracking in alignment matrix of sine backbones. Consider the backbones $\mathbf{x}$ and $\mathbf{y}$ from Figure 4.2. The path $p : \{1, 2, \ldots, 7\} \to \widetilde{\mathbf{x}} \times \widetilde{\mathbf{y}}$, constructed from backtracking is illustrated through the black arrows and purple highlighted entries. In particular, $p(1) = \mathbf{mat}[7, 7] = 0.115$ and $p(9) = \mathbf{mat}[1, 1] = 0$.

- Vertical move from $\mathbf{mat}[i, j]$ to $\mathbf{mat}[i + 1, j]$ indicates an alignment of x_i with an insertion.
- Horizontal move from $\mathbf{mat}[i, j]$ to $\mathbf{mat}[i, j + 1]$ indicates an alignment of y_j with an insertion.
- Diagonal move from $\mathbf{mat}[i, j]$ to $\mathbf{mat}[i+1, j+1]$ indicates an alignment of x_i with y_j.

Next we verify that an alignment found from backtracking is indeed an alignment.

Proposition 6.3.4 (Backtracking Finds an Alignment)**.** *Let* $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ *and* $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ *be backbones. Let* $\boldsymbol{mat}$ *be the* $(m + 1) \times (n + 1)$ *alignment matrix. Let* $\alpha : [k] \to \widetilde{\boldsymbol{x}} \times \widetilde{\boldsymbol{y}}$ *be an alignment found from backtracking. Then,* α *is an alignment.*

Proof. We verify that α is well-defined, and α satisfies all four properties of being an alignment (Definition 4.1.4). Let $p' : [k + 1] \to \mathbf{mat}$ be the path used to construct α.

First we show α is well-defined. Let $h \in [k - 1]$ and consider $\alpha(h)$. Observe exactly one of the three conditions (vertical move, horizontal move, diagonal move) that define $\alpha(h)$ holds by the construction of p', and $\alpha(h) \in \widetilde{\mathbf{x}} \times \widetilde{\mathbf{y}}$. Hence, α is well-defined.

Observe by construction, α has no null alignments. Hence, Property (1) of Definition 4.1.4 holds. Next we show the remaining properties.

Property (2) (Preserves Order of Backbones). We use the index function $\iota_{\mathbf{x}}$ and the coordinate function $\alpha_{\mathbf{x}}$ from Definition 4.1.4. Suppose $\alpha_{\mathbf{x}}(i), \alpha_{\mathbf{x}}(j) \in \mathbf{x}$ where $i < j$. Suppose further that $p'(i) = \mathbf{mat}[q, r]$ and $p'(j) = \mathbf{mat}[s, t]$, indicating that $\alpha_{\mathbf{x}}(i) = x_q$, $\alpha_{\mathbf{x}}(j) = x_s$. By construction of p' and the fact that $i < j$ either (1) $q = s$ and $r < t$ or (2) $q < s$. We claim that only (2) holds in our setting.

Assume for a contradiction that $q = s$ and $r < t$. By construction of p', this means that there are only horizontal moves between $p'(i)$ and $p'(j)$. Hence, $\alpha_{\mathbf{x}}(i)$ or $\alpha_{\mathbf{x}}(j)$ or both must be the empty node. This contradicts the assumption that both $\alpha_{\mathbf{x}}(i)$ and $\alpha_{\mathbf{x}}(j)$ are in $\mathbf{x}$. Therefore $q < s$, and in particular, $q < s$ if and only if $i < j$.

Since $\alpha_{\mathbf{x}}(i) = x_q$ and $\alpha_{\mathbf{x}}(j) = x_s$, we have the index function $\iota_{\mathbf{x}}(\alpha_{\mathbf{x}}(i)) = q$ and $\iota_{\mathbf{x}}(\alpha_{\mathbf{x}}(j)) = s$ if and only if $\iota_{\mathbf{x}}(\alpha_{\mathbf{x}}(i)) < \iota_{\mathbf{x}}(\alpha_{\mathbf{x}}(j))$. Therefore $\iota_{\mathbf{x}}(\alpha_{\mathbf{x}}(i)) < \iota_{\mathbf{x}}(\alpha_{\mathbf{x}}(j))$ if and only if $i < j$, so that α preserves the order of nodes in the backbone $\mathbf{x}$. The same argument substituting $\mathbf{y}$ for $\mathbf{x}$ also shows that α preserves the order of nodes in the backbone $\mathbf{y}$.

Property (3) (No Misalignments). By design of **mat**, a misalignment has an infinite cost. Since each entry in the alignment matrix is a minimum of three values where at least two values are finite, then **mat** does not contain any infinite entries. This implies that when applying backtracking, we never have a diagonal move corresponding to a misalignment.

Property (4) (Restriction to Matching). Let $x_i \in \mathbf{x}$. By definition of backtracking, there exists exactly one $j \in [k+1]$ such that $p'(j) = \mathbf{mat}[i+1, h]$ where $h \in [n+1]$. This implies that x_i appears in $\text{im}(\alpha_{\mathbf{x}})$ exactly once. Similarly, for $y_i \in \mathbf{y}$, there exists exactly one $j \in [k+1]$ such that $p'(j) = \mathbf{mat}[h, j+1]$ where $h \in [m+1]$. Hence y_i appears in $\text{im}(\alpha_{\mathbf{y}})$ exactly once. Therefore we have a restriction to matching.

□

We now prove that an alignment found using this backtracking has a cost equal to $\mathbf{mat}[m+1, n+1]$.

Proposition 6.3.5 (Backtracking Finds Alignment with Cost Computed from Alignment Matrix)**.** *Let* $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ *and* $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ *be backbones. Let* ***mat*** *be the* $(m+1) \times (n+1)$ *alignment matrix. Let* $\alpha : [k] \to \widetilde{\boldsymbol{x}} \times \widetilde{\boldsymbol{y}}$ *be an alignment found from backtracking. Then,* $\boldsymbol{cost}(\alpha) = \boldsymbol{mat}[m+1, n+1]$.

Proof. We show $\mathbf{cost}(\alpha) = \mathbf{mat}[m+1, n+1]$. To do this, we prove $\mathbf{cost}(\alpha[1:h]) = p'(h+1)$ for all $h \leq k$ by induction. For the base case, consider $\alpha[1:1] = \alpha(1)$. There are three possibilities for $\alpha(1)$. Either:

1. $\alpha(1) = (x_1, \mathbf{0})$

2. $\alpha(1) = (\mathbf{0}, y_1)$

3. $\alpha(1) = (x_1, y_1)$.

Recall that $x_i \in \mathbf{x}$ can be expanded as $x_i = (s_{\mathbf{x},i}, w_{\mathbf{x},i})$; likewise, for $y_i \in \mathbf{y}$, we can write $y_i = (s_{\mathbf{y},i}, w_{\mathbf{y},i})$. If (1), then $\mathbf{cost}(\alpha(1)) = w_{\mathbf{x},1} = \mathbf{mat}[2,1] = p'(2)$. If (2), then $\mathbf{cost}(\alpha(1)) = w_{\mathbf{y},1} = \mathbf{mat}[1,2] = p'(2)$. If (3), then $\mathbf{cost}(\alpha(1)) = \mathbf{diff}(x_1, y_1) = \mathbf{mat}[2,2] = p'(2)$. In all three cases we find $\mathbf{cost}(\alpha(1)) = p'(2)$.

Next, we assume the induction hypothesis that $\mathbf{cost}(\alpha[1:h]) = p'(h+1)$ for some $h < k$.

Suppose $p'(h+1) = \mathbf{mat}[i,j]$. There are three possibilities for $\alpha(h+1)$. Either

1. $\alpha(h+1) = (x_i, \mathbf{0})$

2. $\alpha(h+1) = (\mathbf{0}, y_j)$

3. $\alpha(h+1) = (x_i, y_j)$.

If (1), then

$$\mathbf{cost}(\alpha[1:h+1]) = \mathbf{cost}(\alpha[1:h]) + w_{\mathbf{x},i} = p'(h+1) + w_{\mathbf{x},i} = \mathbf{mat}[i,j] + w_{\mathbf{x},i} = p'(h+2).$$

If (2), then

$$\mathbf{cost}(\alpha[1:h+1]) = \mathbf{cost}(\alpha[1:h]) + w_{\mathbf{y},j} = p'(h+1) + w_{\mathbf{y},j} = \mathbf{mat}[i,j] + w_{\mathbf{y},j} = p'(h+2).$$

If (3), then

$$\begin{aligned}\mathbf{cost}(\alpha[1:h+1]) &= \mathbf{cost}(\alpha[1:h]) + \mathbf{diff}(x_i, y_j) = p'(h+1) + \mathbf{diff}(x_i, y_j) \\ &= \mathbf{mat}[i,j] + \mathbf{diff}(x_i, y_j) = p'(h+2).\end{aligned}$$

All equalities follow from either the induction hypothesis or construction of p'

By induction, $\mathbf{cost}(\alpha[1:h]) = p'(h+1)$ for all $h \leq k$. In particular,

$$\mathbf{cost}(\alpha[1:k]) = \mathbf{cost}(\alpha) = p'(k+1) = \mathbf{mat}[m+1, n+1].$$

□

Observe that Proposition 6.3.2 and Proposition 6.3.5 give the following corollary.

Corollary 6.3.6 (Backtracking Finds Optimal Alignment)**.** *Let* $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ *and* $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ *be backbones. Let* $\boldsymbol{mat}$ *be the* $(m+1) \times (n+1)$ *alignment matrix. Let* $\alpha : [k] \to \widetilde{\boldsymbol{x}} \times \widetilde{\boldsymbol{y}}$ *be an alignment found from backtracking. Then* $\boldsymbol{cost}(\alpha) = c_{\boldsymbol{x},\boldsymbol{y}}(m, n)$.

6.3.2 Time Complexity of Computing Backbone and Extremal Event DAG Distance

Using the dynamic program above, we can compute the backbone distance in $O(mn)$ time where m and n are the lengths of the two backbones. However, since backbone alignments are not always unique, then computing all optimal backbone alignments can become costly.

When computing the extremal event DAG distance, we must compute the optimal backbone alignments that minimize the difference in weights over all possible aligned edges. Since we could have multiple optimal backbone alignments, then computing the extremal event DAG distance in the worst case is expensive. However, we have found empirically for the applications below that almost always there is a unique optimal alignment. This then results in a polynomial time complexity for computing the extremal event DAG distance.

CHAPTER SEVEN

APPLICATIONS

We apply the extremal event DAG construction and distance to two applications: (1) quantifying similarity in replicate experiments of microarray yeast cell cycle data and (2) providing quantitative evidence that an intrinsic oscillator drives the blood stage cycle of the malaria parasite *Plasmodium falciparum*. These two datasets were analyzed in [7] and [77], respectively, using a directed maximal common edge subgraph (DMCES) metric that compared ε-DAGs (recall ε-DAGs described after Definition 3.0.2) with a sequence of fixed ε. Because of the computational complexity of computing the DMCES metric, these calculations were done on a limited number of time series with a limited number of extrema per time series, i.e., on less noisy data. Additionally, since the ε-DAGs specify a value of a parameter ε, the experiments were performed over a range of ε between 0 and 0.15. The construction of extremal event DAGs does not need the value of ε to be specified. Additionally, both the extremal event DAG construction and distance are computed in polynomial time as opposed to the exponential time complexity of the DMCES metric. This all means we can compute distances over much larger sets of genes in a significantly shorter amount of time.

7.1 Yeast Cell Cycle Data

The first dataset consists of microarray time series transcriptomics from the yeast *Saccharomyces cerevisiae*, published in [62]. The yeast cell cycle is well studied and has experimental validation [16, 33, 34, 73]. The amplitude of the data has been normalized between -0.5 and 0.5 and its phase has been shifted by alignment using CLOCCS analysis, see "Appendix A: Yeast Data Analysis" in [7]. Using the CLOCCS analysis, the replicate

experiments were aligned so that the time series start at the same point in the yeast cell cycle. Furthermore, the data were truncated to one period so that the data analysis focuses on the extrema from a synchronized cell population, since the production of daughter cells causes increasing levels of cell division asynchrony that reduces the periodic signal. We analyze two collections of time series data $\mathcal{D}_1$ and $\mathcal{D}_2$ that each consists of 16 genes and 265 time points.

We perform three different comparison computations:

1. We focus on a subset of $\mathcal{D}_1$ and $\mathcal{D}_2$ that consists of the time series for four genes: SWI4, YOX1, NDD1, and HCM1. We denote these sub-datasets as $\mathcal{D}_1'$ and $\mathcal{D}_2'$ respectively. We then compute the extremal event DAG distance between the extremal event DAGs of $\mathcal{D}_1'$ and $\mathcal{D}_2'$,

$$d_{ED}(\mathrm{DAG}(\mathcal{D}_1'), \mathrm{DAG}(\mathcal{D}_2')).$$

2. We consider dataset $\mathcal{D}_2'$ but switch labels between time series for $CLB2$ and $YOX1$. We call this mislabeled dataset $\mathcal{D}_3'$. Then we compute

$$d_{ED}(\mathrm{DAG}(\mathcal{D}_1'), \mathrm{DAG}(\mathcal{D}_3')).$$

   The comparison between $d_{ED}(\mathrm{DAG}(\mathcal{D}_1'), \mathrm{DAG}(\mathcal{D}_2'))$ to $d_{ED}(\mathrm{DAG}(\mathcal{D}_1'), \mathrm{DAG}(\mathcal{D}_3'))$ indicates the impact of the replacement of one time series by another on the extremal event DAG distance.

3. Lastly, we assess the distance between the full datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ by constructing a baseline distribution for the expected distance. We do this by first by scrambling the gene names in $\mathcal{D}_2$ to create a dataset $\hat{\mathcal{D}}_2$. We then compute

$$d_{ED}(\mathrm{DAG}(\mathcal{D}_1), \mathrm{DAG}(\hat{\mathcal{D}}_2)).$$

We repeat this computation 100 times for 100 random name assignments. This experiment gives us an idea on the range of possible distances between $\mathcal{D}_1$ and $\mathcal{D}_2$. We then compare this distribution to the actual distance

$$d_{ED}(\mathrm{DAG}(\mathcal{D}_1), \mathrm{DAG}(\mathcal{D}_2)).$$

Since the extremal event DAG distance can be any non-negative number, it can be difficult to discern how similar $\mathcal{D}_1$ and $\mathcal{D}_2$ are solely based on computing $d_{ED}(\mathrm{DAG}(\mathcal{D}_1), \mathrm{DAG}(\mathcal{D}_2))$. To gain a better understanding of how similar $\mathcal{D}_1$ and $\mathcal{D}_2$ are, we perform computation (3) to get a baseline distribution of distances between the time series in $\mathcal{D}_1$ and time series in $\mathcal{D}_2$. This distribution can then be used as a null hypothesis H_0 for testing H_1 that $\mathcal{D}_1$ and $\mathcal{D}_2$ measure gene expression in the identically behaving cell in the same environmental condition.

Computations 1 & 2. We computed

$$d_{ED}(\mathrm{DAG}(\mathcal{D}_1'), \mathrm{DAG}(\mathcal{D}_2'))) = 10.34$$
$$d_{ED}(\mathrm{DAG}(\mathcal{D}_1'), \mathrm{DAG}(\mathcal{D}_3'))) = 15.48.$$

The mismatched gene dataset causes a 50% increase in distance even though only 25% of the dataset was perturbed, a substantial change. This result is consistent with the result from numerical experiment 3 in [7] where the same data were analyzed using ε-DAGs and the DMCES metric. Specifically, DMCES similarity was computed between the ε-DAGs at ε values ranging between 0 and 0.15. A similarity score of one indicated that the ε-DAGs are equal whereas a similarity score of 0 indicates the ε-DAGs are very dissimilar. The similarity ranged between 0.7 and 1 for $\mathcal{D}_1'$ and $\mathcal{D}_2'$, whereas the similarity ranged between 0.4 and 0.6 for $\mathcal{D}_1'$ and $\mathcal{D}_3'$. The same qualitative conclusion can be drawn from our results

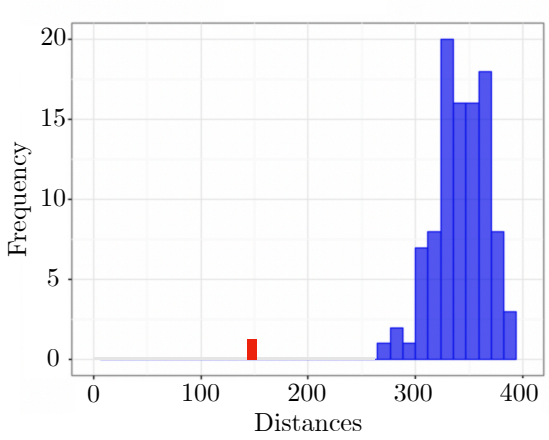


Figure 7.1: Extremal Event DAG Distances in Experiment 3. The red bar is $d_{ED}(\mathrm{DAG}(\mathcal{D}_1), \mathrm{DAG}(\mathcal{D}_2))$, the distance between the two yeast datasets without any scrambling of genes.

and the earlier work; namely that replacing one time series with another decreases similarity, or increases distance, between datasets.

Computation 3. After computing $d_{ED}(\mathrm{DAG}(\mathcal{D}_1)), \mathrm{DAG}(\hat{\mathcal{D}}_2))$ 100 times we get the distribution of distances shown in Figure 7.1 with the following statistics

- Maximum Distance = 384.30
- Mean Distance = 341.90
- Minimum Distance = 273.95
- Standard Deviation = 23.14

We found $d_{ED}(\mathrm{DAG}(\mathcal{D}_1), \mathrm{DAG}(\mathcal{D}_2)) = 150.44$.

Therefore $d_{ED}(\mathrm{DAG}(\mathcal{D}_1), \mathrm{DAG}(\mathcal{D}_2))$ is roughly eight standard deviations (see Figure 7.1) below the mean of the estimated null distribution, which suggests that there is a significant amount of similarity between $\mathcal{D}_1$ and $\mathcal{D}_2$.

In numerical experiment 4 in [7], the same goal of measuring replicate similarity was approached using a different technique. Subsets of 4 and 8 genes out of the 16 total were used to construct ε-DAGs over the range of ε 0-0.15. No baseline was calculated, but the computations showed a mean similarity that was usually over the relatively high value of 80% and frequently over 90%.

Overall, the results of these three computations are consistent with the numerical experiments in [7]. We offer the significant improvement that we did not need to run each of these computations over a range of ε because the extremal event DAG construction does not depend on ε. Because of the vast increase in computational efficiency, we were able to perform computation 3 using all 16 genes in the two datasets instead of randomly sampled subsets. It was shown in [7] that using subsets of increasing size decreased variance in the subsampled computations rather than substantially changing mean performance, which was taken to be evidence that subsampling was a good proxy for the distance between $\mathcal{D}_1$ and $\mathcal{D}_2$. In other words, if the DMCES method could have computed the distance between $\mathcal{D}_1$ and $\mathcal{D}_2$, then increasing the size of the subsamples would show convergence to that value. We offer a method that does not require a convergence argument, but rather can compute the value directly.

7.2 Malaria Parasite Data

In this application, we seek to show that oscillatory genes in strains of *Plasmodium falciparum* maintain as much of a phase ordering as well known circadian genes across various mouse tissues. This provides circumstantial evidence that malaria parasites have an internal clock that is at least as conserved as that of the circadian oscillator, as shown in [77]. The mouse data comes from [89] that contains the circadian transcriptomes of 12 mouse organ tissues every two hours for 48 hours. In [77], similarity between datasets within malaria or mouse was determined by choosing "reference datasets" to which the other strains and

tissues were compared, as opposed to computing all pairwise comparisons. The 3D7 strain and the liver tissue were chosen as the reference datasets in *Plasmodium falciparum* and the mouse tissues respectively.

In both collections of time series, subsets of periodic genes were selected that peak at similar times across parasite strains or mouse tissues. These genes are called "in-phase" subsets. After this subset of genes was found, all datasets were interpolated with piecewise cubic Hermite interpolating polynomial spline to one hour intervals. The data were wrapped so that there could be a common starting point. Furthermore, *Plasmodium falciparum* was down-sampled by removing every odd datapoint. More details on the experiments and data preprocessing can be found in the supplementary materials of [77]. After the pre-processing steps, our time series contain 119 parasite genes and 107 mouse genes.

To gain a better understanding of how similar our datasets are, we created a baseline distribution for each strain or tissue by randomly interchanging gene names and shifting each time series by a random amount, using the same random shifts as in [77]. Specifically, if we view the values of a time series as an ordered list $h_1, h_2, \ldots, h_n$ of length n, we randomly select $1 \leq m \leq n$ to create a new shifted time series

$$h_m, h_{m+1}, \ldots, h_n, h_1, h_2, \ldots, h_{m-1}.$$

The phase shift operation preserves characteristics of the dataset except for the ordering of extrema. For each strain and tissue, the baseline distance was computed between the unpermuted and unshifted reference dataset and the permuted and shifted datasets.

We use the following notation for the parasite datasets:

- Let $\mathcal{D}_1$ be the collection of time series from strain 3D7. This is the reference dataset.
- Let $\mathcal{D}_2$ be the collection of time series from strain FVO-NIH.

- Let $\mathcal{D}_3$ be the collection of time series from strain SA250.
- Let $\mathcal{D}_4$ be the collection of time series from strain D6.
- Let $\mathcal{D}'_2$ be the collection of shifted and permuted time series of $\mathcal{D}_2$.
- Let $\mathcal{D}'_3$ be the collection of shifted and permuted time series of $\mathcal{D}_3$.
- Let $\mathcal{D}'_4$ be the collection of shifted and permuted time series of $\mathcal{D}_4$.

We perform the following computations to study the parasite data.

1. Pick 1500 random subsets of 15 genes. With the arbitrary choice of 15 genes fixed, let $\hat{\mathcal{D}}_i$ denote the subset of the corresponding time series from $\mathcal{D}_i$ for $i \in \{1, 2, 3, 4\}$. Compute
$$d_{ED}(\mathrm{DAG}(\hat{\mathcal{D}}_1), \mathrm{DAG}(\hat{\mathcal{D}}_j))$$
for each random subset and each $j \in \{2, 3, 4\}$.
2. Pick a new set of 1500 random subsets of 15 genes, where $\hat{\mathcal{D}}_i$ for $i \in \{1, 2, 3, 4\}$ is defined as above for each subset. Let $\hat{\mathcal{D}}'_j$ denote the subset of the shifted and permuted time series $\mathcal{D}'_j$ for $j \in \{2, 3, 4\}$. Compute $d_{ED}(\mathrm{DAG}(\hat{\mathcal{D}}_1), \mathrm{DAG}(\hat{\mathcal{D}}'_j))$ for each random subset and $j \in \{2, 3, 4\}$.

We do analogous experiments to study the mouse gene data. We use the following notation.

- Let $\mathcal{D}_a$ be the collection of time series recorded from liver tissue. This is the reference dataset.
- Let $\mathcal{D}_b$ be the collection of time series recorded from kidney tissue.
- Let $\mathcal{D}_c$ be the collection of time series recorded from lung tissue.

- Let $\mathcal{D}'_b$ be the collection of shifted and permuted time series of $\mathcal{D}_b$.
- Let $\mathcal{D}'_c$ be the collection of shifted and permuted time series of $\mathcal{D}_c$.

We perform the following experiments to study the mouse tissue data.

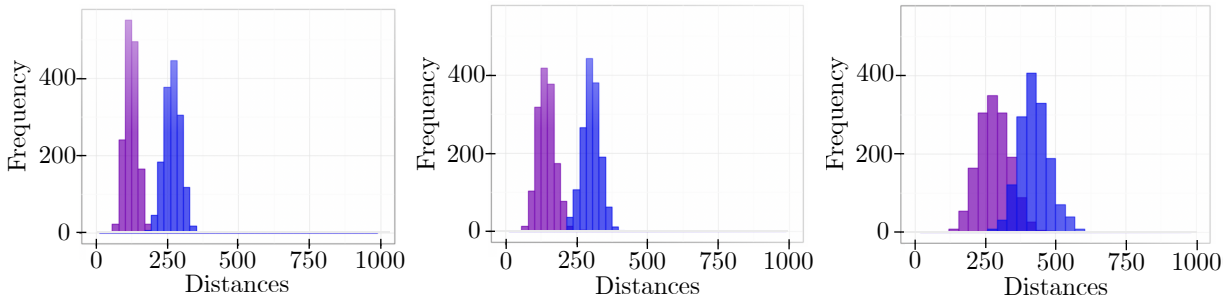
1. Pick 1500 random subsets of 15 genes. Let $\hat{\mathcal{D}}_i$ denote the subset of the corresponding time series from $\mathcal{D}_i$ for $i \in \{a, b, c\}$. Compute $d_{ED}(\text{DAG}(\hat{\mathcal{D}}_a), \text{DAG}(\hat{\mathcal{D}}_j))$ for each random subset and $j \in \{b, c\}$.
2. Pick a new set of 1500 random subsets of 15 genes, where $\hat{\mathcal{D}}_i$ for $i \in \{1, 2, 3, 4\}$ is defined as above for each subset. Let $\hat{\mathcal{D}}'_j$ denote the subset of the shifted and permuted time series $\mathcal{D}'_j$ for $j \in \{b, c\}$. Compute $d_{ED}(\text{DAG}(\hat{\mathcal{D}}_a), \text{DAG}(\hat{\mathcal{D}}'_j))$ for each random subset and $j \in \{b, c\}$.

We summarize the results of the experiments below. In Figure 7.2 and Figure 7.3, we see in general that the baseline distances are larger than the distances between cell lines and reference cell line.

We provide tables summarizing the results of the computations.

Distance	Mean	Median	Standard Deviation
$d_{ED}(\text{DAG}(\hat{\mathcal{D}}_1), \text{DAG}(\hat{\mathcal{D}}_2))$	123.13	122.68	22.12
$d_{ED}(\text{DAG}(\hat{\mathcal{D}}_1), \text{DAG}(\hat{\mathcal{D}}'_2))$	267.97	268.18	28.68
$d_{ED}(\text{DAG}(\hat{\mathcal{D}}_1), \text{DAG}(\hat{\mathcal{D}}_3))$	143.44	142.20	30.37
$d_{ED}(\text{DAG}(\hat{\mathcal{D}}_1), \text{DAG}(\hat{\mathcal{D}}'_3))$	301.70	302.38	30.55
$d_{ED}(\text{DAG}(\hat{\mathcal{D}}_1), \text{DAG}(\hat{\mathcal{D}}_4))$	282.00	279.72	56.59
$d_{ED}(\text{DAG}(\hat{\mathcal{D}}_1), \text{DAG}(\hat{\mathcal{D}}'_4))$	424.06	421.20	53.24

Table 7.1: Summary of Results from Parasite Data.



(a) D6 (b) FVO-NIH (c) SA250

Figure 7.2: Histograms from *Plasmodium falciparum* experiments. The reference strain was 3D7 for all experiments. The distribution of baseline extremal event DAG distances is shown in blue for each graph and the distribution of extremal event DAG distances is shown in purple. In all three plots, the extremal event DAG distances are smaller than the corresponding baseline distances. Performing a paired t-test to the blue and purple distributions with a null hypothesis that the distributions are the same in all three plots resulted in a p-value below machine precision.

(a) Lung (b) Kidney

Figure 7.3: Histograms from mouse experiments. The reference cell line was liver for all experiments. The distribution of baseline extremal event DAG distances is shown in blue for each graph and the distribution of extremal event DAG distances is shown in purple. In all three plots, we see the extremal event DAG distances are smaller than the corresponding baseline distances. Performing a paired t-test to the blue and purple distributions with a null hypothesis that the distributions are the same in all three plots resulted in a p-value below machine precision.

Distance	Mean	Median	Standard Deviation
$d_{ED}(\mathrm{DAG}(\hat{\mathcal{D}}_a), \mathrm{DAG}(\hat{\mathcal{D}}_b))$	493.40	490.88	71.38
$d_{ED}(\mathrm{DAG}(\hat{\mathcal{D}}_a), \mathrm{DAG}(\hat{\mathcal{D}}'_b))$	642.51	640.78	66.70
$d_{ED}(\mathrm{DAG}(\hat{\mathcal{D}}_a), \mathrm{DAG}(\hat{\mathcal{D}}_c))$	436.17	432.42	59.62
$d_{ED}(\mathrm{DAG}(\hat{\mathcal{D}}_a), \mathrm{DAG}(\hat{\mathcal{D}}'_c))$	607.97	607.06	60.33

Table 7.2: Summary of Results from Mouse Data.

Overall, we find the extremal event DAG distances from the parasite experiments to be smaller than the extremal event DAG distances from the mouse experiments, both in absolute value and in terms of distribution overlap, as found in [77], indicating support for a malarial clock. Additionally, Figure 7.2 shows another pattern seen in [77], which is that the D6 data have the smallest distance to the 3D7 reference data and that SA250 has the largest distance to 3D7 compared both in absolute value and to their respective baselines.

In this computation, we used subsampling to computationally handle the size of dataset, as in [77], and computed 1500 samples instead of 5000. However, we used more than double the genes in our samples, 15 versus 6, and incorporated information about all levels of ε instead of only a small fixed subset of ε. The consistency of results found between the computations validates the methodology of using extremal event DAGs in place of ε-DAGs.

CHAPTER EIGHT

CONCLUSION

We constructed a weighted directed graph descriptor of collections of time series data that keeps track of the order and prominence of extrema, and is robust to experimental noise that arises from taking discrete time samples of a continuous process. Furthermore, we define a distance between these extremal event DAGs that constructs an extremal event supergraph using a modified version of the edit distance and then computes the distance by taking the L_1 distance between aligned node and edge weights in the extremal event supergraph. The benefit of this distance is that it can be computed via dynamic programming and therefore can be computed efficiently. We used this distance to compare the similarity of experimental replicates in yeast cell cycle data, the similarity of circadian gene expression in different mouse tissues, and the similarity of gene expression across malaria parasite strains. Our results are consistent with results from other literature [7, 77] that used directed maximal common edge subgraphs of ε-DAGs [60]. The benefit to using the extremal event DAG methodology is that the savings in memory and computation speed facilitates the analysis of significantly larger datasets.

Furthermore, we prove several stability results. In particular, the backbone distance arising from two functions is bounded by the L_∞ distance of the two functions multiplied by the number of nodes in the backbone infinity alignment. Using backbone stability, we prove the extremal event DAG distance is stable in a local case. Local here means that the individual time series in one collection differs from the corresponding time series in the other collection by the amount that allows direct alignment of the minima and maxima between the two time series. Additionally, one of the time series can have additional small amplitude maxima and minima. Extension of the local stability result to a stability between arbitrary

multivariate time series is challenging. Theoretically, one needs a way to globally compare distances between multivariate time series. If the triangle inequality for the extremal event DAG distance holds, then the local stability of the extremal event DAG can be used to prove a global stability result using the same technique as in the proof of backbone stability (Theorem 5.1.14). While in this paper we define the extremal event DAG distance using the L_1-norm, we suspect that stability also holds if we use any L_p-norm. We leave this generalization as future work.

We focus on a descriptor that is robust to measurement error that arises from taking discrete time samples of a continuous process. However, there can be other types of uncertainty present in the data. One type is related to signal processing and is seen as small peaks in the data. If we want to remove this type of measurement errors from our analysis, we can apply a preprocessing step using techniques from [58]. This technique applies sublevel set persistence to a time series to determine a node life threshold. Nodes with a node life below the threshold are classified as results of signal processing errors, or as noise. Eliminating nodes classified as noise and then computing extremal event DAGs gives a smaller descriptor of a collection of time series that further increases the size of computationally feasible datasets. Similar preprocessing steps to remove small peaks can be made using Fourier transforms [9, 39].

In summary, extremal event DAGs are a new computational tool that can be used alone or in combination with noise reduction algorithms to summarize and compare collections of time series data.

REFERENCES CITED

[1] Christoph Bandt and Bernd Pompe. Permutation entropy: A natural complexity measure for time series. *Phys. Rev. Lett.*, 88:174102, Apr 2002.

[2] Ulrich Bauer and Michael Lesnick. Induced matchings of barcodes and the algebraic stability of persistence. In *SoCG '14: Proceedings of the 30th Annual Symposium on Computational Geometry*, 2014.

[3] Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[4] Robin Belton, Bree Cummins, and Robert R. Narem. Computing and comparing extremal event dags. `https://github.com/breecummins/min_interval_posets`, 2021.

[5] Paul Bendich, Herbert Edelsbrunner, Dmitriy Morozov, and Amit Patel. Homology and Robustness of Level and Interlevel sets. *Homology, Homotopy and Applications*, 15(1):51 – 72, 2013.

[6] Paul Bendich and John Harer. Persistent intersection homology. *Foundations of Computational Mathematics*, 11:305–336, December 2010.

[7] Eric Berry, Bree Cummins, Robert R. Narem, Lauren M. Smith, Steven B. Haase, and Tomáš Gedeon. Using extremal events to characterize noisy time series. *Journal of Mathematical Biology*, 80:1523–1557, February 2020.

[8] Jesse Berwald and Marian Gidea. Critical transitions in a model of a genetic regulatory system. *Mathematical Biosciences & Engineering*, 11(4):723–740, 2014.

[9] Steven Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(2):113–120, 1979.

[10] Sara L. Bristow, Adam R. Leman, Laura A. Simmons Kovacs, Anastasia Deckard, John Harer, and Steven B. Haase. Checkpoints couple transcription network oscillator dynamics to cell-cycle progression. *Genome Biology*, 15(446), 2014.

[11] Peter Bubenik, Vin de Silva, and Johnathan Scott. Metrics for generalized persistence modules. *Foundations of Computational Mathematics*, 15(6):1501–1531, 2015.

[12] Gunnar Carlsson and Vin de Silva. Zigzag persistence. *Foundations of Computational Mathematics*, 10:367–405, 2010.

[13] Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Y. Oudot. Proximity of persistence modules and their diagrams. In *SoCG '09: Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry*, pages 237–246, 2009.

[14] Frédéric Chazal, Vin de Silva, Marc Glisse, and Steve Oudot. *The Structure and Stability of Persistence Modules*. Springer, 1 edition, 2016.

[15] Chao Chen, Xiuyan Ni, Qinxun Bai, and Yusu Wang. A topological regularizer for classifiers via persistent homology. In *AISTATS*, 2019.

[16] Chun-Yi Cho, Christina M. Kelliher, and Steven B. Haase. The cell-cycle transcriptional network generates and transmits a pulse of transcription once each cell cycle. *Cell Cycle*, 2019.

[17] Chun-Yi Cho, Francis C. Motta, Christina M. Kelliher, and Steven B. Haase. Reconciling conflicting models for global control of cell-cycle transcription. *Cell Cycle*, 2017.

[18] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete and Computational Geometry*, 37:103–120, 2007.

[19] David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Yuriy Mileyko. Lipschitz functions have l_p-stable persistence. *Foundations of Computational Mathematics*, 10:127–139, 2010.

[20] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *SoCG 2006: Proceedings of the Twenty-Second Annual Symposium on Computational Geometry*, pages 119–126, 2006.

[21] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.

[22] Bree Cummins, Tomáš Gedeon, Shaun Harker, and Konstantin Mischaikow. Model rejection and parameter reduction via time series. *SIAM Journal of Applied Dynamical Systems*, 17(2):1589–1616, 2018.

[23] Meryll Dindin, Yuhei Umeda, and Frédéric Chazal. Topological Data Analysis for Arrhythmia Detection through Modular Neural Networks. In *CanadianAI 2020 - 33rd Canadian Conference on Artificial Intelligence*, Proc. 33rd Canadian Conference on Artificial Intelligence, May 2020., Ottawa, Canada, May 2020. 7 pages, 4 figures.

[24] Herbert Edelsbrunner and John Harer. Persistent homology - a survey. *Surveys on Discrete and Computational Geometry: Twenty Years Later*, pages 257–282, 2008.

[25] Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. Applied Mathematics. American Mathematical Society, 2010.

[26] Michelle Feng and Mason A. Porter. Persistent homology of geospatial data: A case study with voting. *SIAM Review*, 63:67–99, 2021.

[27] Patrizio Frosini and Claudia Landi. Size theory as a topological tool for computer vision. *Pattern Recognition and Image Analysis*, 9:596–603, 1999.

[28] Shafie Gholizadeh and Wlodek Zadrozny. A short survey of topological data analysis in time series and system analysis. *arXiv:1809.10745*, 2018.

[29] Robert Ghrist. The persistent topology of data. *American Mathematical Society. Bulletin. New Series.*, 45:61–75, 2008.

[30] Robert Ghrist. *Elementary Applied Topology*. Createspace, 1 edition, 2014.

[31] Navarro Gonzalo. A guided tour to approximate string matching. *ACM Computing Surveys*, 33, 2001.

[32] Mikhail Gromov. *Hyperbolic Groups*, pages 75–263. Springer New York, New York, NY, 1987.

[33] David Günther, Joseph Salmon, and Julien Tierny. Mandatory critical points of 2d uncertain scalar fields. *Computer Graphics Forum*, 33:31–40, 2014.

[34] Steven B. Haase and Curt Wittenberg. Topology and control of the cell-cycle-regulated transcriptional circuitry. *Genetics*, 196:65–90, 2014.

[35] Allen Hatcher. *Algebraic Topology*. Algebraic Topology. Cambridge University Press, 2002.

[36] Jean-Claude Hausmann. On the Vietoris-Rips Complexes and a Cohomology Theory for Metric Spaces. In *Prospects in Topology: Proceedings of a Conference in honour of William Browder, Annals of Mathematics Studies*, pages 175–188. Princeton University Press, 1995.

[37] Daniel S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18:341 – 343, 1975.

[38] Tomasz Kaczynski, Konstantin Mischaikow, and Marian Mrozek. *Computational Homology*, volume 157. Springer-Verlag, 2004.

[39] Sunil D. Kamath and Philipos C. Loizou. A multi-band spectral subtraction method for enhancing speech corrupted by colored noise. *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 4, 2002.

[40] Firas A. Khasawneh and Elizabeth Munch. Topological data analysis for true step detection in periodic piecewise constant signals. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 474, 2018.

[41] Woojin Kim and Facundo Mémoli. Stable signatures for dynamic metric spaces via zigzag persistent homology, 2017. arXiv:1712.04064.

[42] Woojin Kim and Facundo Mémoli. Spatiotemporal persistent homology for dynamic metric spaces. *Discrete Computational Geometry*, 66:831–875, 2021.

[43] Laura A. Simmons Kovacs, Michael B. Mayhew, David A. Orlando, Yuanjie Jin, Qingyun Li, Chenchen Huang, Steven I. Reed, Sayan Mukherjee, and Steven B. Haase. Cyclin-dependent kinases are regulators and effectors of oscillations driven by transcription factor network. *Molecular Cell*, 45(5):669–679, 2012.

[44] Laura A. Simmons Kovacs, David A. Orlando, and Steven B. Haase. Transcription networks and cyclin/cdks: The yin and yang of cell cycle oscillators. *Foundations of Computational Mathematics*, 7:2626–2629, 2008.

[45] Peter Lawson, Andrew B. Sholl, J. Quincy Brown, Brittany Terese Fasy, and Carola Wenk. Persistent homology for the quantitative evaluation of architectural features in prostate cancer. *Scientific Reports*, 9(1139), 2019.

[46] Yongjin Lee, Senja D. Barthel, Paweł Dłotko, Seyed Mohamad Moosavi, Kathryn Hess, and Berend Smit. High-throughput screening approach for nanoporous materials genome using topological data analysis: Application to zeolites. *Journal of Chemical Theory and Computation*, 14(8):4427–4437, 2018.

[47] Michael Lesnick. The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics*, 15:613–650, 2015.

[48] Vladimir Iosifovich Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[49] Roy Lowrance and Robert A. Wagner. An extension of the string-to-string correction problem. *Journal of the ACM*, 22:177–183, 1975.

[50] Di Ma, Tongyu Liu, Lin Chang, Crystal Rui, Yuanyuan Xiao, Siming Li, John B. Hogenesch, Y. Eugene Chen, and Jiandie D. Lin. The liver clock controls cholesterol homeostasis through trib1 protein-mediated regulation of pcsk9/low density lipoprotein receptor (ldlr) axis*. *Journal of Biological Chemistry*, 290(52):31003–31012, 2015.

[51] John Milnor. *Morse Theory*. Princeton University Press, New Jersey, 1963.

[52] Dimitriy Morozov and Gunther H. Weber. Distributed merge trees. In *Proceedings of the Annual Symposium on Principles and Practice of Parallel Programming*, pages 93–102. ACM, February 2013.

[53] Dmitriy Morozov, Kenes Beketayev, and Gunther H. Weber. Interleaving distance between merge trees. *Discrete Computational Geometry*, 49:22–45, 2013.

[54] James R. Munkres. *Elements of Algebraic Topology*. Addison-Wesley, 1984.

[55] James R. Munkres. *Topology: Second Edition*. Upper Saddle River: Prentice Hall, 2000.

[56] Ludovic S. Mure, Hiep D. Le, Giorgia Benegiamo, Max W. Chang, Luis Rios, Ngalla Jillani, Maina Ngotho, Thomas Kariuki, Ouria Dkhissi-Benyahya, Howard M. Cooper, and Satchidananda Panda. Diurnal transcriptome atalas of primate across major neural and peripheral tissues. *Science*, 359, 2018.

[57] Audun Myers and Firas A. Khasawneh. On the automatic parameter selection for permutation entropy. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(033130), 2020.

[58] Audun Myers, Firas A. Khasawneh, and Brittany Terese Fasy. Separating persistent homology of noise from time series data using topological signal processing. *arXiv:2012.04039*, 2020.

[59] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, March 1970.

[60] Riley Nerem, Peter Crawford-Kahrl, Bree Cummins, and Tomáš Gedeon. A poset metric from the directed maximum common edge subgraph, 2019. arXiv:1910.14638.

[61] John C. Nesbit. The accuracy of approximate string matching algorithms. *Journal of Computer Based Instruction*, 13, 1986.

[62] David A. Orlando, Charles Y. Lin, Allister Bernard, Jean Y. Wang, Joshua E. S. Socolar, Edwin S. Iversen, Alexander J. Hartemink, and Steven B. Haase. Global control of cell cycle transcription by coupled cdk and network oscillators. *Nature*, 453(7197):944–947, 2008.

[63] Nina Otter, Mason A. Porter, Ulrike Tillmann, Peter Grindrod, and Heather A. Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6, 2017.

[64] Olumide Owolabi and Douglas R. McGregor. Fast approximate string matching. *Software: Practice and Experience*, 18, 1988.

[65] Jose A. Perea. A brief history of persistence. *Morfismos*, 23:1–16, 2019.

[66] Jose A. Perea, Anastasia Deckard, Steve B. Haase, and John Harer. Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. *BMC Bioinformatics*, 16, August 2015.

[67] G. Petri, P. Expert, F. Turkheimer, R. Carhart-Harris, D. Nutt, P.J. Hellyer, and F. Vaccarino. Homological scaffolds of brain functional networks. *Journal of the Royal Society Interface*, 11(20140873), 2014.

[68] Sahand Jamal Rahi, Kresti Pecani, Andrej Ondracka, and Catherine Oikonomou. The cdk-apc/c oscillator predominantly entrains periodic cell-cycle transcription. *Cell*, 165:475–487, 2016.

[69] Vanessa Robins. Toward computing homology from finite approximations. *Topology Proceedings*, 24:503–532, 1999.

[70] Marc D. Ruben, Gang Wu, David F. Smith, Robert E. Schmidt, Lauren J. Francey, Yin Yeng Lee, Ron C. Anafi, and John B. Hogenesch. A database of tissue-specific rhythmically expressed human genes has potential applications in circadian medicine. *Science Translational Medicine*, 10, September 2018.

[71] Nicole F. Sanderson, Elliott Shugerman, Samantha Molnar, James D. Meiss, and Elizabeth Bradley. Computational topology techniques for characterizing time-series data. In Niall Adams, Allan Tucker, and David Weston, editors, *Advances in Intelligent Data Analysis XVI*. Springer, 2017.

[72] Kerby Shedden and Stephen Cooper. Analysis of cell-cycle gene expression in saccharomyces cerevisiae using microarrays and multiple synchronization methods. *Nucleic Acids Research*, 30:2920–2929, 2002.

[73] I. Simon, J. Barnett, N. Hannett, C.T. Harbison, N.J. Rinaldi, T.L. Volkert, J.J. Wyrick, J. Zeitlinger, Gifford D.K., T.S. Jaakkola, and R.A. Young. Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell*, 106:697–708, 2001.

[74] Primoz Skraba and Katharine Turner. Wasserstein stability for persistence diagrams. *arXiv: Algebraic Topology*, 2020.

[75] Michael Small. Complex networks from time series: Capturing dynamics. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2509–2512, 2013.

[76] Dmitriy Smirnov and Dmitriy Morozov. Triplet merge trees. *Topological Methods in Data Analysis and Visualization V (TopoInVis'17)*, 2020.

[77] Lauren M. Smith, Francis C. Motta, Garima Chopra, J. Kathleen Moch, Robert R. Nerem, Bree Cummins, Kimberly E. Roche, Christina M. Kelliher, Adam R. Leman, John Harer, Tomáš Gedeon, Norman C. Waters, and Steven B. Haase. An intrinsic oscillator drives the blood stage cycle of the malaria parasite plasmodium falciparum. *Science*, 368(6492):754–759, 2020.

[78] Temple F. Smith and Michael S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[79] Bernadette J. Stolz, Heather A. Harrington, and Mason A. Porter. Persistent homology of time-dependent functional networks constructed from coupled time series. *Chaos*, 27, 2017.

[80] Floris Takens. Detecting strange attractors in turbulence. *Dynamical Systems and Turbulence, Warwick 1980, Lecture Notes in Mathematics*, 898:366 – 381, 1981.

[81] Chad M. Topaz, Lori Ziegelmeier, and Tom Halverson. Topological data analysis of biological aggregation models. *PLoS ONE*, 2015.

[82] Ruey S. Tsay. *Multivariate Time Series Analysis: With R and Financial Applications*. Wiley, 1st edition, 2013.

[83] M. Ulmer, Lori Ziegelmeier, and Chad M. Topaz. A topological approach to selecting models of biological experiments. *PLoS ONE*, 2019.

[84] Leopold Vietoris. Über den höheren zusammenhang kompakter räume und eine klasse von zusammenhangstreuen abbildungen. *Mathematische Annalen*, pages 454–472, 1927.

[85] Oliver Vipond, Joshua A. Bull, Philip S. Macklin, Ulrike Tillmann, Christopher W. Pugh, Helen M. Byrne, and Heather A. Harrington. Multiparameter persistent homology landscapes identify immune cell spatial patterns in tumors. *Proceedings of the National Academy of Sciences*, 118(41), 2021.

[86] Robert A. Wagner and Michael J. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21:168–173, 1974.

[87] William W. S. Wei. *Multivariate Time Series Analysis and Applications*. Wiley, 1st edition, 2019.

[88] Lu Xian, Henry Adams, Chad M. Topaz, and Lori Ziegelmeier. Capturing dynamics of time-varying data via topology. *Foundations of Data Science*, 2021.

[89] Ray Zhang, Nicholas F. Lahens, Heather I. Ballance, Michael E. Hughes, and John B. Hogenesch. A circadian gene expression atlas in mammals: Implications for biology and medicine. *Proceedings of the National Academy of Sciences of the United States of America*, 111:16219–16224, 2014.

[90] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete Computational Geometry*, 33:249–274, 2005.