DANTE: Determining Adaptation trajectories in biological Networks Through Evolutionary mapping

Tamim Khatib
CISE Department
University of Florida
Gainesville, FL, USA
0009-0006-7236-2221

Oscar Diaz de la Rua
CISE Department
University of Florida
Gainesville, FL, USA
0009-0000-9516-5593

Kawthar Moria
Computer Science Department
King Abdulaziz University
Jeddah, SAU
0000-0001-6241-2658

Tamer Kahveci
CISE Department
University of Florida
Gainesville, FL, USA
0000-0002-4403-8612

***Abstract*—Biological networks are dynamic structures. They continuously evolve by rewiring their interactions. These rewirings happen at different rates for different cells, and the rates can change over time, yet we can only observe the cell at a limited number of stages of their evolution. In this paper, we consider the problem of determining evolutionary trajectories of dynamic biological networks. We develop a novel algorithm DANTE (Determining Adaptation trajectories in biological Networks Through Evolutionary mapping), which maps multiple cellular network evolution patterns in accordance with the greatest possible similarity. We evaluate our method on protein-protein interaction (PPI) networks using a mouse model with evolutionary patterns caused by chronic myeloid leukemia (CML) and compare it to four alternative strategies. Our experimental results demonstrate that DANTE outperforms competing methods in terms of trajectory similarity, and the advantages of DANTE over competing methods grow when network trajectories are incomplete.**

***Index Terms*—dynamic networks, rewiring interactions, network evolution trajectory**

I. Introduction

Cellular systems governed by complex networks continuously evolve by rewiring the interactions among molecules. Genetic and epigenetic mutations [20], response to external stimulants [24], and variations in DNA replication timing [7], [21] are among the reasons behind this change. The average human undergoes roughly 6.7 trillion genetic mutations per day, with each of their 37 trillion cells undergoing about 1700 mutations after 25 years [16]. In addtıon to thıs, circadian rhythm, particularly disruptions to it, can cause periodic changes to cellular structure [34], aging leads to a slow, steady change in cellular structure [26], and epigenetic mutations cause temporary changes that impact cellular structure during the time it remains in effect [19].

The evolution rates of cellular systems are complex and variable, both across individuals as well as across tissues within the same individual. There are many reasons behind such variability. For instance, while the genetic makeup of every human body is 99.9 percent identical, the 0.1 percent variation leads to hundreds of millions of variant sequences. The average human genome has more than 2000 structural variations [5]. Such variation can result in a substantial difference in the progression of a disease and the response to a drug treatment [22], [27], [28], [42]. A patient's response to a drug can vary wildly depending on their genetic makeup, with genetic factors accounting for up to 95 percent of patient variability [6]. External factors, such as tobacco, sun light, and drugs, further contribute to the variation in the rewiring rates of interactions per organism, per cell, and even over time [33]. Thus, how fast the network topology evolves varies from person to person, cell to cell, and time period to time period.

One major gap in personalized medicine and cell engineering is that most studies focus on the target state of the cell and ignore the trajectory of changes the underlying topology of the cellular interactions go through to achieve the final cell state. For instance, when the same drug is administered on different patients, the response to that drug can be extremely diverse to the extent that a dose to one patient may be lethal, while that same dose may be too little to elicit even the desired effect on another patient [36]. The reasons for this huge variation can be explained by observing the trajectory through which the cells' interaction network evolves in response to the external stimulant, as the intermediate states of the cell may contribute to the outcome. Even if the sequence of changes for two patients in response to the same drug are identical, the variations in the speed at which these changes take place may affect the response time to that drug [35]. Therefore, finding out how the cellular networks evolve as well as the speed at which they continue to evolve is of utmost importance for understanding and manipulating cellular functions.

Monitoring the evolution of a cell state continuously is not feasible, as it requires obtaining tissue samples for each time point at which a cell is observed in wet-lab experiments. The availability of tissue samples is limited due to often requiring invasive procedures to collect samples from patients [18]. Additionally, the cost of processing each tissue sample is high, since it necessitates human expertise, wet-lab resources, and time to process the samples [14]. For instance, the cost of RNAseq experiments for one sample exceeds $500-$2,000 using 30-50 million reads per sample. Given these practical limitations in data collection, the current way to monitor the evolutionary trajectory of the cell state after it is manipulated is to measure gene characteristics at specific time points, leading to a time series of observed interaction patterns. For instance, weekly monitoring of cancer cells after administering apoptosis-inducing drugs can partially reveal alterations in cancer and normal cells [25]. Yet our knowledge about the evolutionary trajectory of individual patients remains incomplete, as different patients may respond to the same drug at different rates, and thus, the time lapse between two consecutively measured time points may yield significantly more rewiring of interactions for one patient than another [35].

Our contributions. In this paper, we consider the problem

of determining evolutionary trajectories of dynamic biological networks. Mathematically, we formulate this as a new variant of the dynamic network alignment problem where given biological networks may have missing data. Here, the missing data refers to the time points where different dynamic networks exhibit different trajectories in their evolution. We develop the novel algorithm DANTE (Determining Adaptation trajectories in biological Networks Through Evolutionary mapping), which maps multiple cellular network evolution patterns in accordance with the greatest possible similarity. DANTE is capable of both interpolating between two observed network topologies and extrapolating before the first and after the last observed network topology. DANTE considers each sample's genetic mapping as a three-dimensional structure, where the first and second dimensions correspond to specific genes and the third dimension corresponds to time. A filled cell corresponds to an interaction between two different genes at a specific time point. We evaluate our method on protein-protein interaction (PPI) networks using a mouse model with evolutionary patterns caused by chronic myeloid leukemia (CML) and compare it to four alternative strategies. Our experimental results demonstrate that DANTE outperforms competing methods in terms of trajectory similarity, and the gap between DANTE and competing methods grows in favor of DANTE when the network trajectories are incomplete. We also observe up to 77.7% success in locating and positioning the missing network positions when we randomly remove networks in the given network sequences. These results suggest that DANTE has great potential to advance our understanding of evolving systems and build more accurate generative machine learning models to construct the evolution of the wiring of dynamic networks.

The rest of the paper is organized as follows. Section II presents the key studies and gaps in the literature. Section III presents our DANTE algorithm. Section IV experimentally evaluates our methods. We conclude with a brief summary in Section V.

II. Related Works and Gaps In Literature

The problem of identifying the evolutionary trajectories of biological network topologies is associated with two orthogonal challenges, namely data imputation and network alignment. Below, we summarize the previous literature on these two fields and why they alone are not sufficient to address the problem considered in this paper.

A. Data imputation

The purpose of data imputation is to train on the already-existing data to predict any missing data. This problem has been considered for estimating interactions among molecules in the literature. Wang et al. developed a method using similarity-regularized matrix factorization to predict anticancer drug responses on cell lines [23]. Suphavilai et al. utilized a recommender system to predict cancer drug responses for unseen cell-lines and patients [40]. More recently developed methods to impute drug-interaction use manifold learning [1], 3D-Fiber-based Tensors [8], and the distribution of missing drug-drug interactions [9]. Chen et al provides a comprehensive survey on data imputation for drug response prediction [10].

Prediction of edges is another subset of data imputation. We categorize edge prediction methods based on the information used. These methods may rely on local similarity, global similarity, node attributes, correlation information, identifying the most influential node, and machine learning/AI. Shalforoushan and Jalali developed an edge prediction method that utilizes Bayesian Networks [39]. Aouay et al. introduced a feature-based prediction method that used supervised learning [3]. Yan and Gregory developed a method that relied on a mixture of local and global similarity measures, which they referred to as a node's "community", to find missing edges [45]. A survey of these edge prediction models can be found in [30].

Although data imputation methods may be effective for estimating data characteristics at certain states, it requires prior knowledge of the state of both the training data and the data to be estimated. However, as we explain later in detail, this prior knowledge is missing for the problem we consider in this paper, as the topology of each network may be evolving at different and varying rates over time. This makes using the existing data imputation methods ineffective for estimating evolutionary trajectories, for the trajectory itself is needed to estimate the network topology.

B. Network alignment

The network alignment problem aims to find a mapping between the nodes of given input networks. We consider the existing studies on this problem in three categories: pairwise alignment, multiple alignment, and dynamic alignment. Pairwise alignment assumes that only two networks are aligned, and these two networks are static (i.e., their topologies do not change over time). GRAAL [29], GHOST [31], SPINAL [2], and PINALOG [32] are a few examples to the algorithms in this category. A comparative survey of these methods are available at Clark et al. [13]. Multiple alignment generalizes this to more than two networks, while still maintaining the assumption that the input networks are static. MultiMAGNA++, which is a multiple alignment extension of the pairwise MAGNA, [44] and CrossMNA, which can operate without any additional attribute information, [11] are two examples of such. The problem of aligning dynamic networks considers the case when the given networks are not static; they evolve over time. Thus, each system considered is actually an ordered sequence of networks. There are two variants of these methods. The first one updates the alignment of the network as they evolve. The second one finds an aggregated best alignment, which is invariant, although the input networks change. DynaMAGNA++ [43], GoT-WAVE [4], Tempo++ [17], and Twadn [46] are just a few examples of dynamic alignment algorithms. We refer the readers to the survey by Cinaglia and Cannataro for a detailed overview of the methods in this category [12].

All the literature discussed above focuses on the mapping of the molecules of the input networks under varying mapping models. While each of the listed dynamic network alignment algorithms have their benefits and drawbacks, none of them are tailored towards finding evolutionary trajectories. They assume either that the input networks are static or, even if they evolve, that the evolutionary trajectories are already known and focus on aligning network topologies under this assumption. This assumption, however, is very strong, as different cellular

systems react to changes at varying speeds and thus, it is not possible to have this information a priori.

III. DANTE ALGORITHM

A. Terminology & problem definition

We model each PPI network as a graph denoted with $G = (V, E)$, where each protein corresponds to a node in set V and each interaction between a pair of proteins is an edge in set E. Consider a dynamic cellular system with an evolving network topology, which is observed at t time points with the ith time point being before the $(i+1)$th time point for all $0 < i < t$. We denote the topology of the observed network at the ith time point with $G_i = (V_i, E_i)$. Without losing generality, we assume that the set of nodes remain unchanged over time, and thus simplify our notation to $G_i = (V, E_i)$. This assumption holds by setting $V = \bigcup_i V_i$. Thus, we represent an evolving network observed with t time points as $S = [G_1, G_2, \cdots, G_t]$. Consider a collection of k evolving cellular systems, each leading to a sequence of graph topologies as described above, with the jth system observed at t_j time points (t_j is a positive integer). We represent networks of these k systems as

$$\begin{aligned} S_1 &= [G_{1,1}, G_{1,2}, \cdots, G_{1,t_1}], \\ S_2 &= [G_{2,1}, G_{2,2}, \cdots, G_{2,t_2}], \\ &\cdots \\ S_k &= [G_{k,1}, G_{k,2}, \cdots, G_{k,t_k}] \end{aligned} \tag{1}$$

with $G_{j,i} = (V, E_{j,i})$. Notice that the notation we use above is powerful, as it does not impose any limit to the number of time points. Additionally, it does not make assumptions on the time lapse between consecutive time points. That is, for a given evolving network, the time lapse between the ith and $(i+1)$th time points may be different than that between the i'th and $(i'+1)$th time points for $i \neq i'$. Furthermore, our notation does not assume that the time point of the observed network of two different evolving networks (i.e., $G_{j,i}$ and $G_{j',i}$ for $j \neq j'$) corresponds to the same evolutionary time for the two systems. It merely denotes that they are the ith observed time point in chronological order in their corresponding systems.

The alignment of the trajectories of a collection of evolving networks aims to find a mapping of the observed time points of these network sequences. Let us denote a mapping of the time points in sequences S_{j_1} and S_{j_2} with a partial bijective function $\psi_{j_1,j_2}() : \{1, 2, \cdots, t_{j_1}\} \rightarrow \{1, 2, \cdots, t_{j_2}\} \cup \{\perp\}$. Here, the symbol $\perp$ represents NULL, and we use it when the observed time point in the sequence of networks S_{j_1} cannot be found in the sequence S_{j_2}. We say that any two observed graphs in the given collection of evolving networks can be aligned with one another so long as they satisfy all of the following three conditions:

1) The two graphs belong to different sequences. Meaning, we cannot align a network instance from an evolving system from another instance from the same system.
2) A graph is not aligned to more than one graph from the same sequence. That is, $\forall i \neq i'$, $\psi_{j_1,j_2}(i) \neq \psi_{j_1,j_2}(i')$ other than the NULL mapping $\psi_{j_1,j_2}(i) = \perp$.

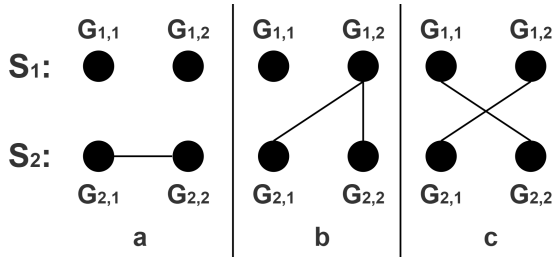


Fig. 1: Illustration of the violations of the three alignment conditions. Each black circle represents a network instance of an evolving system observed at a specific time point. The time progresses from left to right (i.e., the black circle on the right denotes a network instance which is observed later in time). The line connecting two networks indicates that those two networks are aligned with each other. (a) Aligned graphs belong to the same evolving network at different time points. (b) A graph in the second sequence is aligned to more than one graph in the first sequence. (c) The second graph in the second sequence is aligned with a graph in the first sequence at a time point before that of the first graph in the same sequence, and thus the mappings cross each other.

3) The mapping between graph pairs do not cross over each other. That is, $\forall i < i'$, $\psi_{j_1,j_2}(i) > \psi_{j_1,j_2}(i')$ is not a legal alignment.

Figure 1 illustrates these three conditions.

Given a collection of k evolving networks $S_1, S_2, \cdots, S_k$, we say that a collection of mappings $\psi_{j_1,j_2}()$, $1 \leq j_1 \neq j_2 \leq k$ is an alignment of the trajectories of these k evolving networks if they satisfy the following two properties.

- **Reflexivity:** If $\psi_{j_1,j_2}(i) \neq \perp$, then $\psi_{j_2,j_1}(\psi_{j_1,j_2}(i)) = i$.
- **Transitivity:** If $\psi_{j_1,j_2}(i) = i' \neq \perp$, we have $\forall j_3 \in \{1, 2, \cdots, k\} - \{j_1, j_2\}$, $\psi_{j_1,j_3}(i) = \psi_{j_2,j_3}(i')$.

Now we are ready to define the trajectory alignment problem considered in this paper.

Definition: Trajectory Alignment Problem (TAP). Assume that we are given a collection of k evolving networks $S_1, S_2, \cdots, S_k$. Let us denote an arbitrary graph in any of these evolving networks with G. Given a similarity function $\varphi() : (G \cup \{\perp\}) \times (G \cup \{\perp\}) \rightarrow [0 : 1]$ between two graphs (observed or not observed), TAP seeks to find the alignment $\psi_{j_1,j_2}()$, $1 \leq j_1 \neq j_2 \leq k$, which maximizes the total pairwise similarity score

$$\frac{1}{\text{Alignment length}} \sum_{1 \leq j_1 < j_2 \leq k} \sum_{i} \varphi(G_{j_1,i}, G_{j_2,\psi_{j_1,j_2}(i)}).$$

In the definition of TAP above, we use a similarity function notation $\varphi()$. We elaborate on the computation of this function in Section III-B as it is a part of the algorithm which computes the alignment. Notice that TAP is different than the classic network alignment problem in the literature, which aims to map the nodes of two or more networks. This is because TAP assumes that the aligned networks model the interactions among the same molecules, and thus the set of nodes in the corresponding graphs are identical throughout different time points as well as dynamic systems. Therefore, identical nodes map to each other regardless of the time point. What is challenging here is to map the time points at which their wiring of interactions are most similar.

We develop a dynamic programming (DP) solution to TAP. We call our algorithm DANTE (Determining Adaptation trajectories in biological Networks Through Evolutionary mapping). In the following, we first explain DANTE when the number of input network sequences is two ($k = 2$). We then generalize it to arbitrary number of dynamic networks.

B. Alignment of a pair of network trajectories

Assume that we are given two sequences of graphs S_1 and S_2 as input. Also assume that these sequences consist of t_1 and t_2 networks, respectively. Consider the mapping functions $\psi_{1,2}() : \{1, 2, \cdots, t_1\} \rightarrow \{1, 2, \cdots, t_2\} \cup \{\perp\}$ and $\psi_{2,1}() : \{1, 2, \cdots, t_2\} \rightarrow \{1, 2, \cdots, t_1\} \cup \{\perp\}$, which satisfy the three conditions for legal alignment and the reflexivity property described in Section III-A.

DANTE computes a DP matrix D, which has t_1+1 rows and t_2+1 columns. We number the rows and columns of this matrix with indices in $\{0, 1, \cdots, t_1\}$ and $\{0, 1, \cdots, t_2\}$, respectively. The entry of this matrix at row i_1 and column i_2, denoted with $D[i_1, i_2]$, contains the similarity score of the alignment of the first i_1 networks of S_1 with the first i_2 networks of S_2 when they are optimally aligned.

Algorithm 1 presents the pseudo-code for DANTE. We start by initializing the first column (lines 1-3) and the first row (lines 4-6) of the matrix D. An entry $D[0, i_2]$ corresponds to the case when the first network in S_1 is aligned with the $(i_2 + 1)$th network in S_2. In other words, the evolution trajectory of the dynamic system observed in S_1 is similar to that in S_2 only after the i_2 observed time points of S_2. In that case, since we do not have any observed network topologies from S_1 with a trajectory similar to S_2 during the first i_2 observed time points of S_2, we compute how similar the two systems are at i_2 unobserved time points of S_1 prior to the first observed time point of S_1 as the expected similarity between them. We call this process *extrapolation*. We detail the extrapolation procedure later in this section. Similarly, an entry $D[i_1, 0]$ corresponds to the case when the first network in S_2 is aligned with the (i_1+1)th network in S_1, and thus the trajectory of S_1 is late as compared to S_2. In this case, we extrapolate the i_1 unobserved points of S_2 prior to the first observed time point of S_2.

In order to find the value of any given cell $D[i_1, i_2]$, DANTE considers three options and selects the one with the highest score (lines 9-12).

- **Option 1.** Align G_{1,i_1} with G_{2,i_2}. This option computes $D[i_1, i_2]$ as the sum of the diagonal entry $D[i_1 - 1, i_2 - 1]$ and the similarity between graphs G_{1,i_1}, G_{2,i_2}.
- **Option 2.** Align G_{1,i_1} with an unobserved network topology in the other evolving system between time points $i_2 - 1$ and i_2. We label this unobserved network topology an *interpolation* between G_{2,i_2-1} and G_{2,i_2}. We elaborate on how DANTE computes the similarity between an observed network topology and an interpolated network later in this section. It is possible that more than one network in S_1 (i.e., $a \in \{1, 2, \cdots, i_1\}$ networks; $G_{1,i_1-(a-1)}$, $G_{1,i_1-(a-1)}$, $\cdots$, G_{1,i_1}) is aligned with the interpolated networks between G_{2,i_2-1} and G_{2,i_2}. To account for all of these possibilities, for each $a \in \{1, 2, \cdots, i_1\}$, we compute the sum of $D[i_1-a, i_2-1]$ and the expected similarity between networks $G_{1,i_1-(a-1)}$, $G_{1,i_1-(a-2)}$, $\cdots$, G_{1,i_1} and the a interpolated networks between G_{2,i_2-1} and G_{2,i_2}. We then choose the largest score among the a cases.
- **Option 3.** Align G_{2,i_2} with an unobserved network topology in the other evolving system between time points $i_1 - 1$ and i_1. This is the dual scenario to Option 2 above. Here, we do interpolation on S_1 instead of S_2. Thus, for each $a \in \{1, 2, \cdots, i_2\}$, we compute the sum of $D[i_1 - 1, i_2 - a]$ and the expected similarity between networks $G_{2,i_2-(a-1)}$, $G_{2,i_2-(a-2)}$, $\cdots$, G_{2,i_2} and the a interpolated networks between G_{1,i_1-1} and G_{1,i_1}. We then pick the largest score among the a cases.

Once we reach the final entry of the DP matrix, $D[t_1, t_2]$, we encounter two more cases, which are the dual scenarios of the extrapolation we perform for the first row and column of the DP matrix. This time, we align a suffix of graphs in one evolving network with extrapolated graphs at the end of the other evolving network (lines 15-20).

- **Case 1.** Extrapolate tail of S_2. For each $a \in \{1, 2, \cdots, t_1\}$, we compute the sum of $D[t_1 - a, t_2]$ and the expected similarity between the last a observed networks $G_{1,t_1-(a-1)}$, $G_{1,t_1-(a-2)}$, $\cdots$, G_{1,t_1} and a extrapolated networks after G_{2,t_2}. We then pick the largest score among the a cases.
- **Case 2.** Extrapolate tail of S_1. For each $a \in \{1, 2, \cdots, t_1\}$, we compute the sum of $D[t1, t_2 - a]$ and the expected similarity between networks $G_{2,t_2-(a-1)}$, $G_{2,t_2-(a-2)}$, $\cdots$, G_{2,t_2} and a extrapolated networks after G_{1,t_1}. We then pick the largest score among the a cases.

We report the largest value of the two cases above and $D[t_1, t_2]$ as the final result (line 21). Figure 2 illustrates the concept of interpolation and extrapolation.

Similarity between two observed networks. Next, we explain how DANTE computes the similarity between two observed

Algorithm 1: Dynamic Alignment Algorithm

```
   Input: S_1, S_2;
1  for i_2 ← 1 to t_2 do
2  |  D[0, i_2] ← start extrapolation for S_2;
3  end
4  for i_1 ← 1 to t_1 do
5  |  D[i_1, 0] ← start extrapolation for S_1;
6  end
7  for i_1 ← 1 to t_1 do
8  |  for i_2 ← 1 to t_2 do
9  |  |  R ← max of all interpolations for i_1 − 1;
10 |  |  C ← max of all interpolations for i_2 − 1;
11 |  |  A ← φ(G_{1,i_1−1}, G_{2,i_2−1});
12 |  |  D[i_1, i_2] ← φ(G_{1,i_1}, G_{2,i_2}) + max{A, R, C};
13 |  end
14 end
15 for i_1 ← 0 to t_1 do
16 |  maxR ← end extrapolation for first sequence;
17 end
18 for i_2 ← 0 to t_2 do
19 |  maxC ← end extrapolation for second sequence;
20 end
21 Return ← max{R, C, D[t_1, t_2]}
```

network topologies $G_{1,i_1} = (V, E_{1,i_1})$ and $G_{2,i_2} = (V, E_{2,i_2})$, namely $\varphi(G_{1,i_1}, G_{2,i_2})$. Let us denote the number of edges shared by both graphs, $|E_{1,i_1} \cap E_{2,i_2}|$ as the number of true positives (TP), the number of edges missing from both graphs $\binom{|V|}{2} - |E_{1,i_1} \cap E_{2,i_2}|$ as the number of true negatives (TN), the number of edges listed only in the first graph $|E_{1,i_1} - E_{2,i_2}|$ as the number of false positives (FP), and the number of edges missing only in the first graph $|E_{2,i_2} - E_{1,i_1}|$ as the number of false negatives (FN). Most biological networks are sparse. As a consequence, edges are not distributed evenly to these four categories; we expect the value of TN to be significantly larger than the other three. To calculate the similarity between two graphs while avoiding the bias introduced from the skewed distribution arising from the sparseness of networks, we compute the similarity between the two networks as follows,

$$\varphi(G_{1,i_1}, G_{2,i_2}) = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right). \quad (2)$$

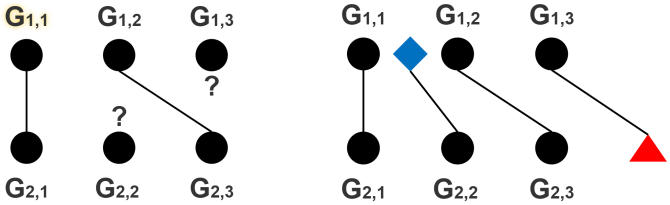
The first term computes the fraction of the edges in G_{1,i_1}, which are also in G_{2,i_2}. The second term computes the fraction of the edges not in G_{1,i_1}, which are also not in G_{2,i_2}. These two terms are also known as positive and negative accuracy, respectively. Our similarity function thus computes the average of the positive and negative similarities between the two networks. This formula is also known as the balanced accuracy (BA).

Next we discuss the two crucial steps of DANTE, namely interpolation and extrapolation. These two steps compute the *expected similarity* between a subsequence of a observed network states in one evolving system, with a subsequence of a unobserved network topologies in the other evolving system. If the unobserved network instances are between two observed networks, we call it interpolation. If they are before the first or after the last observed network, we call it extrapolation. Before we explain how DANTE computes interpolation and extrapolation, we present two important notes:

1) For unobserved network topologies, we do not predict a specific network topology. We instead compute the expected similarity as the average similarity for all possible network topologies which can be observed at a particular time point during the evolution of the given network.
2) We compute the set of possible network topologies based on the evolution trend of that evolving system at the observed time points before/after the estimated unobserved time points, depending on whether it is interpolation or extrapolation.

Interpolation. Without losing generality, assume that, for some j and i_1, the subsequence of j observed networks from S_1, namely G_{1,i_1+1}, G_{1,i_1+2}, $\cdots$, G_{1,i_1+j}, are aligned with j interpolated (i.e., unobserved) networks in S_2. Again, without losing generality, assume that this interpolation is performed between the two observed networks G_{2,i_2} and G_{2,i_2+1}.

As the network topology evolves from G_{2,i_2} to G_{2,i_2+1}, with j interpolated networks in between, some edges must be inserted and others removed to obtain each interpolated network. For each $r \in \{1, 2, \cdots, j\}$, let us denote the set of edges in the rth interpolated network among the j interpolated networks with E_r. To simplify our notation, let us also define $E_0 = E_{2,i_2}$. We would like to compute the expected similarity between G_{1,i_1+r} and the rth interpolated network using the formulation in Equation 2, that is $\text{Exp}[\varphi(G_{1,i_1+r}, (V, E_r))]$.



(a) Dilemma (b) Dilemma Resolution

Fig. 2: Illustration of the dilemma that arises from aligning networks and the dilemma's resolution. Black circles represent network instances of an evolving system observed at specific time points, with time progressing from left to right. Two networks aligned with each other is illustrated with a line connecting the two. (a) This alignment possibility creates a dilemma where network $G_{2,2}$ has no possible network alignment without violating one of the three conditions. Network $G_{1,3}$ is similarly isolated. (b) Our resolution to the dilemma, we create an interpolated network (the blue diamond/square) between $G_{1,1}$ and $G_{1,2}$, then align it with the formerly-isolated $G_{2,2}$. Similarly, we create an extrapolated network (the red triangle) after $G_{2,3}$ and align it with $G_{1,3}$.

During the evolution of the network from G_{2,i_2} to G_{2,i_2+1}, the set of edges in the network change from E_0 to E_{2,i_2+1}. The edges in $E_0 \cap E_{2,i_2+1}$ and those in $(V \times V) - \{(u,u)|u \in V\} - (E_0 \cap E_{2,i_2+1})$ correspond to the edges which are existing and absent in both the initial and final network topologies respectively. Under the mild assumption that the evolution of the wiring of a network happens with the smallest possible set of mutations in the network topology, we do not change the present/absent status of these edges during the network's trajectory, as the initial and the final network topologies are already identical for these edges. On the other hand, the edges in $E_0 - E_{2,i_2+1}$ are removed from, and the edges in $E_{2,i_2+1} - E_0$ are inserted into, the initial network G_{2,i_2} where the evolution begins. At any intermediate stage of the interpolation, say E_r (the rth step out of j interpolation steps), a fraction of the edges in $E_{2,i_2+1} - E_0$ are included and a fraction of the edges in $E_0 - E_{2,i_2+1}$ are excluded. The expected value of the total number of edge insertions and deletions from E_0 to E_r is thus

$$\frac{r}{j}(|E_{2,i_2+1} - E_0| + |E_0 - E_{2,i_2+1}|) \quad (3)$$

Next, we formulate the distribution of these rewirings to obtain E_r into two sets: those removed from and those inserted into E_0.

- We denote the number of edges removed from E_0, but not E_{1,i_1+r} with random variable $x_1 = |(E_0 - E_{2,r}) - E_{1,i_1+r}|$.
- We denote the number of edges removed from both E_0 and E_{1,i_1+r} with random variable $x_2 = |(E_0 - E_{2,r}) \cap E_{1,i_1+r}|$.
- We denote the number of edges inserted from both E_{2,i_2+1} and E_{1,i_1+r} with random variable $x_3 = |(E_{2,r} - E_0) \cap E_{1,i_1+r}|$
- We denote the number of edges inserted from E_{1,i_1+r}, but not from E_{2,i_2+1} and with random variable $x_4 = |(E_{2,r} - E_0) - E_{1,i_1+r}|$.

Figure 3a illustrates these random variables and their association with the three edge sets.

Let us return to Equation 2. In order to compute $\text{Exp}[\varphi(G_{1,i_1+r}, (V, E_r))]$, the four relevant terms are TP, TN, FP, and FN. We first focus on the first term: $\frac{TP}{TP+FN}$. TP is the number of edges shared by the interpolated graph and G_{1,i_1+r}. As Figure 3a illustrates, there are three edge subsets where we observe these edges. Thus $TP = (|E_0 \cap E_{2,i_2+1} \cap E_{1,i_1+r}|) + (|(E_0 \cap e_{1,i_1+r}) - E_{2,i_2+1}| - x_2) + (x_3)$. Here, each parenthesis shows the number of edges in one of the regions. Note that $|E_0 \cap E_{2,i_2+1} \cap G_{1,i_1+r}| + |(E_0 \cap G_{1,i_1+r})|$ is constant, allowing us to replace it with a simple constant denoted with a. This reduces the equation to $TP = a - x_2 + x_3$. The denominator $TP + FN$ is simply the number of edges in E_{1,i_1+r}, and thus $TP + FN = |E_{1,i_1+r}|$ is a constant as well. We denote this constant with b. Therefore, the first term of $\varphi(G_{1,i_1+r}, (V, E_r))$ reduces to

$$\frac{TP}{TP+FN} = \frac{a - x_2 + x_3}{b} \tag{4}$$

Next, we focus on the second term in Equation 2, $\frac{TF}{TF+FP}$. TF is the number of edges missing from both the interpolated graph and G_{1,i_1+r}. Once again, Figure 3a illustrates that there are three regions where this occurs. This leads to the rather lengthy expression $TF = \binom{|V|}{2} - |E_0 \cup E_{2,i_2+1} \cup E_{1,i_1+r}| + |E_{2,i_2+1} - E_0 - E_{1,i_1+r}| - x_4 + x_1$. Since $\binom{|V|}{2} - |E_0 \cup E_{2,i_2+1} \cup G_{1,i_1+r}| + |E_{2,i_2+1} - E_0 - E_{1,i_1+r}|$ is a constant, we denote this constant with c, reducing the expression to $c - x_4 + x_1$. Because $TF + FP$ is the number of edges missing from E_{1,i_1+r}, we compute it as $\binom{|V|}{2} - |E_{1,i_1+r}$—. Once again, this is a constant, so denote it with d.

$$\frac{TN}{TN+FP} = \frac{c - x_4 + x_1}{d} \tag{5}$$

With both fractions in Equation 2 addressed, we compute the similarity between G_{1,i_1+r} and the interpolated network (V, E_r) as

$$\begin{aligned} \varphi(G_{1,i_1}, (V, E_r)) &= \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) \\ &= \frac{1}{2}\left(\frac{a - x_2 + x_3}{b} + \frac{c - x_4 + x_1}{d}\right) \end{aligned} \tag{6}$$

Equation 6 consists of four constants and four variables. Notice that the sum of the random variables $x_1 + x_2 + x_3 + x_4$ denotes the total number of edges inserted or deleted up to the rth interpolated network. While the specific edges to be added or removed is randomly determined, the number of edges chosen is predetermined. In other words, $x_1 + x_2 + x_3 + x_4 = k$, where k is a constant (the value of k is provided in Equation 3). By substituting x_4 with $k - x_1 - x_2 - x_3$, we eliminate x_4 from Equation 6.

To improve the accuracy of the prediction, rather than using one random interpolation, we instead calculate the expected interpolated graph by finding the average of all possible interpolated graphs. This can be done by calculating the expected values of $\text{Exp}[x_1] = \frac{TN \times r}{j}$, $\text{Exp}[x_2] = \frac{FN \times r}{j}$, $\text{Exp}[x_3] = \frac{TP \times r}{j}$, and $\text{Exp}[x_4] = \frac{FP \times r}{j}$. From this, we are able to reach our final similarity score equation, which is derived in Equation 7 as $\text{Exp}[\varphi(G_{1,i_1+r}, (V, E_r))] =$

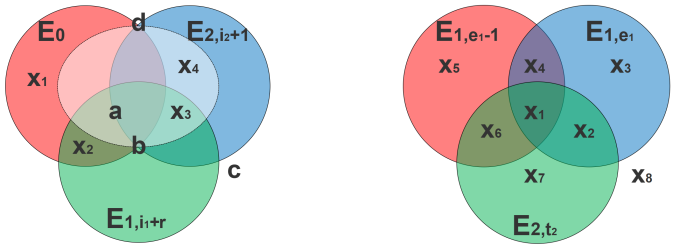
$$\begin{aligned} &= \text{Exp}[\frac{1}{2}(\frac{TP}{TP+FN} + \frac{TN}{TN+FP})] \\ &= \frac{1}{2}\text{Exp}[(\frac{a - x_2 + x_3}{b} + \frac{c - x_4 + x_1}{d})] \\ &= \frac{1}{2}\text{Exp}[(\frac{a - x_2 + x_3}{b} + \frac{c - k + 2x_1 + x_3 + x_2}{d})] \\ &= \frac{a + \text{Exp}[x_2 - x_3]}{2b} + \frac{c - k + 2\text{Exp}[x_1 + x_2 + x_3]}{2d} \end{aligned} \tag{7}$$

Extrapolation. We describe extrapolation at the end of a sequence of networks below. Extrapolation at the beginning of the sequence is symmetric to this description. Without losing generality, assume that for some j and e_1, the subsequence of j observed networks from S_1, namely G_{1,e_1+1}, G_{1,e_1+2}, $\cdots$, G_{1,e_1+j}, are aligned with j extrapolated (i.e., unobserved) networks in S_2, with $e_1 + j = t_1$. Again, without losing generality, assume that this extrapolation is performed after the final network G_{2,t_2}. Calculation of all extrapolated graphs begins with G_{1,e_1+1}, then proceeds in an iterative manner until we reach G_{1,e_1+j}. As such, this paper will discuss in detail only G_{1,e_1+1}, which relies on three graphs for its derivation: G_{1,e_1-1}, G_{1,e_1}, and G_{2,t_2}. Figure 3b shows a visual representation of this as a Venn diagram. All edges fall into one of the eight regions, which are labeled $x_1, x_2, ..., x_8$.

Consider the separate Venn diagram for graphs G_{1,e_1}, G_{1,e_1+1}, and the extrapolated G_{2,t_2+1}. Such a Venn diagram also has eight regions, which can be labelled $y_1, y_2, ..., y_8$ in the same manner as its x counterpart. Since the number of potential edges is constant, $\sum_{i=1}^{8} x_i = \sum_{i=1}^{8} y_i$. Unlike with x_i, we cannot directly determine the values of y_i due to the extrapolated graph G_{2,t_2+1}. However, note that x_1, x_2, x_6, and x_7 all contain edges within E_{2,t_2}, while x_3, x_4, x_5, and x_8 all contain edges excluding it (within $\binom{|V|}{2} - E_{2,t_2}$). Additionally, note that, excluding E_{2,t_2}, x_6 and x_5 consist only of edges in G_{1,e_1-1}, x_2 and x_3 consist only of edges in G_{1,e_1}, x_1 and x_4 consist only of edges in both, and x_7 and x_8 consist only of edges in neither. From there, we assume that the proportions $\frac{x_1}{x_4} = \frac{y_1}{y_4}$, $\frac{x_2}{x_3} = \frac{y_2}{y_3}$, $\frac{x_6}{x_5} = \frac{y_6}{y_5}$, and $\frac{x_7}{x_8} = \frac{y_7}{y_8}$ hold. In other words, the rate of change in each region of the possible distribution of edges inserted and removed during the evolution remains proportional, allowing for an average approximation of the extrapolated graph G_{2,t_2+1} using the expression, $\frac{1}{2}(\frac{y1+y2}{y1+y2+y6+y7} + \frac{y5+y8}{y3+y4+y5+y8})$, which is a variation of Equation 2. This method can be applied recursively for additional extrapolated graphs after the last graph in a sequence, and can be reversed for extrapolated graphs before the first graph in a sequence.

C. Alignment of more than two network trajectories

The process described in the above section is sufficient for handling two sequences of graphs. To handle cases with more than two sequences of graphs in polynomial time, we turn to a strategy called star alignment. Under star alignment, we pick one of the input network sequences as the "center". Then, we align the center sequence independently with all other sequences of graphs using the algorithm described in Section III-B. We identify the mapping among all other pairs of sequences from the transitivity rule described in Section III-A. Figure 4 illustrates this on a small example. We repeat this



(a) Interpolation Venn Diagram (b) Extrapolation Venn Diagram

Fig. 3: Venn diagrams for interpolation and extrapolation. Each circle in the Venn diagrams represents the set of edges belonging to an observed network. (a) The interpolation Venn diagram contains a white oval, which represents the edges in the interpolated (i.e., unobserved) graph. This set encompasses the entire region shared by E_0 and E_{2,i_2+1} and does not touch any area not encompassed by at least one of the two networks/circles. The variables x_1, x_2, x_3, and x_4 each represent the one region they are within. The constants a, b, c, and d represent one or more regions, with each constant touching all regions they encompass. (b) The extrapolation Venn diagram consists of eight regions, labeled in the form of x_i, where i is an integer. Four of these regions are within the E_{2,t_2} circle and correlate with four corresponding regions outside the circle. The proportions between the corresponding regions (i.e. $\frac{x_1}{x_4}$) are used to generate the edges of the extrapolated graphs.

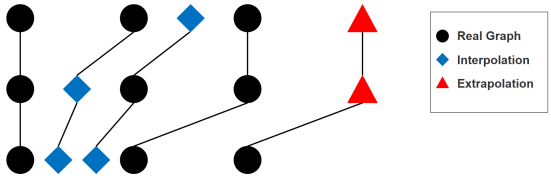


Fig. 4: Illustration of the alignment of three network sequences. The black circles denote the observed (i.e., input) networks. The network sequence in the middle is the center of the star alignment. The sequences on the top and bottom are independently aligned with the middle sequence. The blue diamonds and the red triangles denote interpolated and extrapolated networks, respectively.

process by choosing each input sequence as the center, and pick the alignment with the highest similarity score.

IV. Experimental Evaluation

Transcription data. In order to model an evolving sequence of networks, we use transcription data obtained from mice from the Gene Expression Omnibus (GEO) Series GSE244990 [15]. This dataset contains transcription values for a total of 20 mice, all of which have chronic myeloid leukemia (CML). We consider these mice in five categories as follows:

1) *(Control)* 3 mice receive tetracycline (Tet-on/Control) for 18 weeks. We call this the *control* group.
2) *(CML)* 6 mice do not receive tetracycline (Tet-off/CML). We name this the *CML* group.
3) *(Tet-Off-On)* 4 mice only receive tetracycline after six weeks (Tet-Off-On) to simulate the scenario for successful treatment after six weeks. We call this group *Tet-Off-On*.
4) *(TKI)* 7 mice receive nilotinib, a tyrosine kinase inhibitor (TKI), after six weeks for four weeks to simulate a treatment window. We name this group *TKI*.
5) *(All)* We refer to all 20 mice combined as *All*.

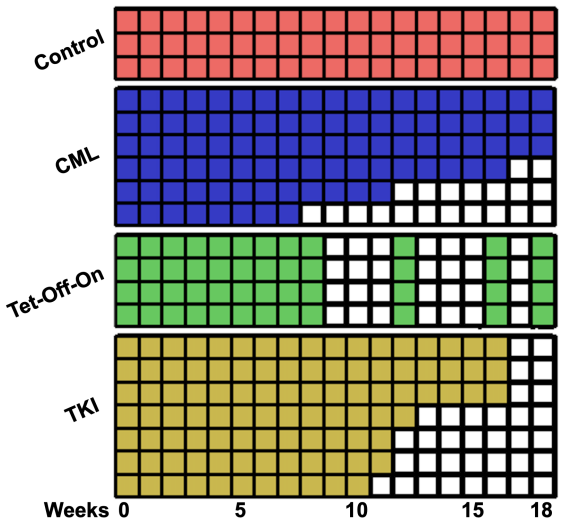


Fig. 5: Summary of the network data. Each row corresponds to a mouse in one of the four groups. Each column is a time point measured in weeks, from week 0 to 18. Highlighted entries show the time point at which an observation is available for the corresponding mice. The block of white cells at the tail end of a mouse indicates that the mouse has perished.

The dataset contains the transcription values for each mouse obtained weekly for up to 18 weeks, assuming they did not perish first, creating a maximum of 19 graphs per mouse (week 0 + 18 weeks of observation). Figure 5 summarizes the dataset.

Network data. We use the PPI network of mice from the STRING database [41]. This database contains interactions between proteins based on several evidences, including co-expression, gene fusion, and text mining. Depending on the evidences supporting each interaction, it reports a confidence value in the $[0 : 1]$ interval with higher values indicating stronger evidence. We use two orthogonal criteria to filter the interactions in this network. The first factor is the confidence level as reported in the STRING database. We use 0.7 and 0.9 as minimum confidence cutoffs to obtain PPI networks at two different stringency levels. The second factor is the minimum transcription level, which quantifies the current level of activity for a specific gene. For each mouse and time point combination, we filter all PPI interactions where the transcription levels of the two genes encoding those two interacting proteins are below the minimum transcription cutoff value. We use two transcription cutoff values in our experiments, namely 30 and 50, to test two levels of stringency for gene activity levels. In total, we have 20 evolving network sequences, with each network observed at up to 19 time points, for four stringency levels of interactions and gene activity levels. With a minimum confidence cutoff of 0.9 and a minimum transcription level of 50, there are an average of 595 nodes and 8278 edges per network sequence. After decreasing the confidence cutoff to 0.7, the average number of edges increases to 11339. With a confidence of 0.9 and a transcription level of 30, there are an average average of 970 nodes and 10772 edges per network sequence. With a confidence of 0.7, this increases to 16601 edges.

Competing methods. We compare our algorithm to four other methods. The first method aligns the first observed networks of all sequences with one another, then the second networks,

in this order without any interpolation. If a sequence runs out of networks before the others, it pads it using extrapolation. We call this method *Extrapol End*. The second method is dual to the first one. It starts aligning the sequences of networks starting from the last observed time points and pads each sequence via extrapolation before the first observed time point if needed. We call this strategy *Extrapol Start*. The third method, called *Extrapol Both* allows for extrapolation from either end of the sequences (i.e., before the first observed network as well as after the last observed network) to occur with equal chances. The fourth method uses the *Dynamic Time Warping (DTW)* algorithm [37], [38]. DTW stretches the sequences of networks to match the time points by duplicating observed networks if needed. By doing this, DTW imitates the interpolation/extrapolation strategy of DANTE. The fundamental difference is DTW duplicates observed networks, while DANTE computes the expected similarity across all possible intermediate graph topologies based on the trajectory of the network it is aligned with. Note that the DTW algorithm works for only pairs of sequences. In order to use it for multiple sequences, we use the same strategy we developed for DANTE in Section III-C.

A. Evaluation of the similarity of the evolutionary trajectory

In our first experiment, we evaluate how well each algorithm aligns the evolutionary trajectories of dynamic networks. We use DANTE and each of the four other competing methods to align the entire sequences of networks for the mice in five groups (Control, CML, Tet-Off-On, TKI, and All). Here, the first four groups represent homogeneous sets of mice with same treatment. The final group models the case when we have a heterogeneous set of evolving patterns. For each group, we build the dynamic network for the two transcription cutoffs 30 and 50 and the two interaction confidence cutoff values 0.7 and 0.9, leading to a total of 80 evolving networks (20 mice $\times$ 4 cutoff combinations), with each network observed at up to 19 time points.

For each mouse group and cutoff combination, we run DANTE and the four competing methods and report the similarity score for the aligned trajectory. Thus, we have a total 100 alignment results (5 mice groups $\times$ 4 cutoff combinations $\times$ 5 methods). Figure 6 presents the results. Our results demonstrate that in all test cases, DANTE consistently outperforms the three methods Extrapol End, Extrapol Start, and Extrapol Both. DANTE performs better than DTW in 60% of cases and performs equally well in 30% of cases. DTW was marginally better than DANTE in only two experiments. Particularly for the two homogeneous groups Control and TKI, and the heterogeneous group All, the gap in accuracy was noticeably high in favor of DANTE. This suggests that our method is highly preferable when the evolution trajectories are highly variable. We observe that the accuracy of all methods increase as we increase the stringency of the interactions (i.e., for higher interaction confidence cutoff). Similarly, the accuracy is also higher for higher stringency for transcription values.

B. Impact of missing trajectories

This experiment evaluates the effects of missing networks on the alignment of network trajectories. To do that, we remove

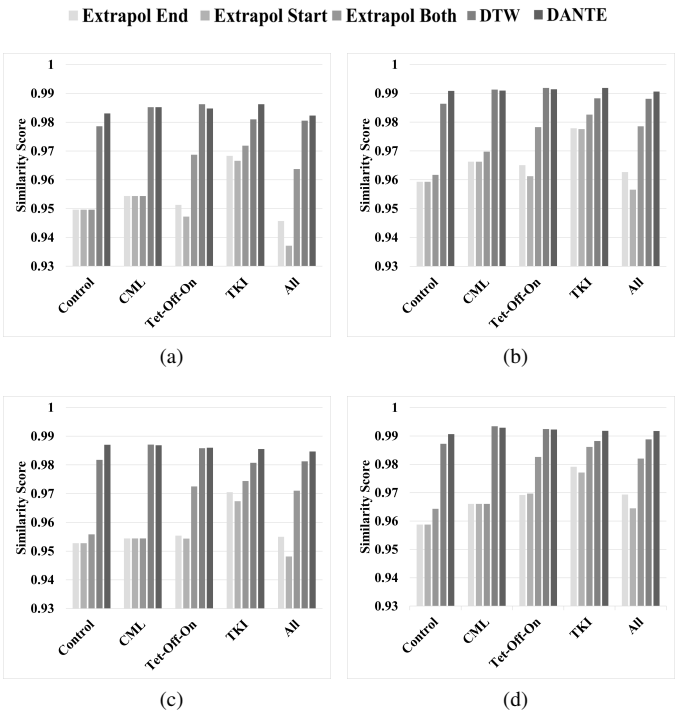


Fig. 6: Similarity of the trajectory mapping obtained by DANTE and four competing methods for five mice groups. (a) Transcription cutoff = 30, interaction confidence cutoff = 0.7. (b) Transcription cutoff = 30, interaction confidence cutoff = 0.9. (a) Transcription cutoff = 50, interaction confidence cutoff = 0.7. (a) Transcription cutoff = 50, interaction confidence cutoff = 0.9.

a randomly selected subset of λ observed networks from the input dataset. We test this for each value of $\lambda \in \{1, 2, 3, 4, 5\}$ as follows. We randomly select a dynamic network $S_j = [G_{j,1}, G_{j,2}, \cdots, G_{j,t_j}]$. We then randomly select a network $G_{j,i}$ from this list and remove it from S_j. We repeat this λ times. Thus the discarded networks can be from any sequence and at any time point. In order to eliminate any bias introduced, for each λ, we repeat this process five times to create five such datasets. We set the minimum transcription cutoff as 50 and minimum interaction confidence to 0.9 (i.e., stringent thresholds) in favor of the competing methods, as their performances were improving for stringent thresholds. We have a total of 125 datasets (5 mice groups $\times$ 5 λ values $\times$ 5 repetitions). We test the top three methods from the previous experiment, Extrapol Both, DTW, and DANTE, as Extrapol End and Extrapol Start yield significantly inferior similarity values in all experiments. We report the average value for each method and λ combination. In total, we perform 375 alignments (3 methods $\times$ 125 datasets).

Similarity of trajectories. Figure 7 shows the similarity value obtained by aligning the trajectories using DANTE and the top two competing methods. Our results demonstrate that DANTE finds the trajectory alignment with the highest similarity score in the vast majority of test cases. Extrapol Both consistently has the least similarity in all the experiments. DANTE outperforms DTW in 72% of cases and performs equally well in 20% of cases. Particularly, for the two homogeneous datasets (Control and CML groups) and the heterogeneous All group, DANTE

yields significantly better similarity scores for all values of λ. Another important observation following from our results is that DANTE is robust to missing data. For all datasets, as the number of artificially removed networks from the trajectory (i.e., λ) increases, DANTE yields the same similarity, whereas Extrapol Both reports fluctuating values and DTW suffers slowly decaying similarity values in three datasets (see All, TKI, and CML datasets).

Accuracy of identifying missing networks. The previous experiment demonstrates that even after removing a subset of network instances from the input network sequences, DANTE can still align the sequences with high similarity. However, that alone is insufficient to determine if DANTE is capable of revealing the networks which were removed. In this experiment, we evaluate how well DANTE can locate the positions of the missing weeks, which is the ultimate goal of this study.

Assume that for two sequences S_{j_1} and S_{j_2}, their alignment maps the network G_{j_1,i_1} to the network G_{j_2,i_2} (i.e., $\varphi_{j_1,j_2}(i_1) = i_2$). Assume that G_{j_2,i_2} is removed along with some other networks. Let us denote the mapping function we compute after removal of these networks with $\varphi'()$. We say that DANTE correctly identifies the position of the artificially removed network G_{j_2,i_2}, if the alignment interpolates a network between the immediately before and immediately after the available time points to the deleted network and maps that interpolated network to G_{j_1,i_1}. Mathematically, the mapping function $\varphi'()$ for the alignment of the input sequences after removal of networks satisfy both of the following two conditions:

i) Let G_{j_2,i_2-a} be the rightmost network in S_2 which is not deleted for $a > 0$. Similarly, let G_{j_2,i_2+b} be the leftmost network in S_2 which is not deleted for $b > 0$. That is G_{j_2,i_2} is between G_{j_2,i_2-a} and G_{j_2,i_2+b}. Then for $\varphi'_{j_2,j_1}(i_2-a) < i_1$ and $\varphi'_{j_2,j_1}(i_2+b) > i_1$.

ii) $\varphi'_{j_1,j_2}(i_1) = \perp$.

If DANTE maps G_{j_1,i_1} to the time point directly before or after the missing time point in S_2 (i.e., off by one time point), we consider this to be a partial success. We treat any other mapping as a failure. To eliminate any bias by random network deletions, we repeat the experiment 30 times for each dataset and each λ value, each time removing λ randomly selected networks from the dataset and report the average. Figure 8 presents the results.

Our results demonstrate that DANTE can locate the missing network locations and also map them to the correct time points in other evolving networks in up to 37% of the cases. DANTE achieves a partial success rate of up to 61.1%, and a combined success rate of up to 77.7%. On average, DANTE perfectly predicts the missing week more than 20% of the time and partially predicts the missing week more than 30% of the time, giving us a combined success rate average of nearly 51%. We observe the highest success in DANTE results for the four homogeneous mice groups, particularly for the Control and CML groups. This indicates that common stress conditions are more likely to help in recovering missing interaction patterns. The number of sequences (mice) in the group does not appear to affect the accuracy of our results. This, however, needs to be tested further as the number of mice in total is not large. As the number of missing networks increases, the combined accuracy tends to drop. This is not surprising, as larger pieces of the trajectory are missing as we remove more networks from the dataset.

V. Conclusion

Cellular systems continuously evolve, and as a result, the topology of interactions among molecules changes through rewiring. However, we only observe a limited number of these networks, usually at arbitrary points in their evolution. In this paper, we developed our novel algorithm DANTE, which aligns the evolution trajectories of multiple sequences of biological networks. In doing so, DANTE identifies the similarities among these evolution patterns. We evaluated our method on protein-protein interaction (PPI) networks using a mouse model with evolutionary patterns caused by chronic myeloid leukemia (CML). Our experimental results demonstrated that DANTE outperformed four competing methods in terms of the trajectory similarity when the network trajectories are incomplete, and acihieved high success in locating and positioning the missing network positions. These results suggest that DANTE has great potential to advance our understanding of evolving systems and build more accurate generative machine learning models to construct the evolution of the wiring of dynamic networks.

References

[1] F. Ahmadi Moughari and C. Eslahchi. Adrml: anticancer drug response prediction using manifold learning. *Scientific Reports*, 10, 08 2020.

[2] Ahmet E. Aladağ and Cesim Erten. Spinal: scalable protein interaction network alignment. *Bioinformatics*, 29:917–924, April 2013.

[3] Saoussen Aouay, Salma Jamoussi, and Faiez Gargouri. Feature based link prediction. *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, pages 523–527, 2014.

[4] David Aparício, Pedro Ribeiro, Tijana Milenković, and Fernando Silva. Temporal network alignment via got-wave. *Bioinformatics*, 35(18):3527–3529, 2019.

[5] Adam Auton, Lisa D Brooks, Richard M Durbin, Erik P Garrison, Hyun Min Kang, Jan O Korbel, Jonathan L Marchini, Shane McCarthy, Gilean A McVean, Goncalo R Abecasis, et al. A global reference for human genetic variation. *Nature*, 526(7571):68–74, 2015.

[6] Donna J Belle and Harleen Singh. Genetic factors in drug metabolism. *American family physician*, 77(11):1553–1560, 2008.

[7] Alexa N. Bracci, Anissa Dallmann, Qiliang Ding, Melissa J. Hubisz, Madison Caballero, and Amnon Koren. The evolution of the human dna replication timing program. *Proceedings of the National Academy of Sciences of the United States of America*, 120, 2023.

[8] A. Bumin, K. Huang, and T. Kahveci. Partialfibers: An efficient method for predicting drug-drug interactions. *International Conference on Computational Advances in Bio and Medical Sciences*, 2023.

[9] A. Bumin, A. Ritz, D. Slonim, T. Kahveci, and K. Huang. Fit: fiber-based tensor completion for drug repurposing. *ACM International Conference on Bioinformatics*, 2022.

[10] Jinyu Chen and Louxin Zhang. A survey and systematic assessment of computational methods for drug response prediction. *Briefings in Bioinformatics*, 22(1):232–246, 01 2020.

[11] Xiaokai Chu, Xinxin Fan, Di Yao, Zhihua Zhu, Jianhui Huang, and Jingping Bi. Cross-network embedding for multi-network alignment. *The World Wide Web Conference*, page 273–284, 2019.

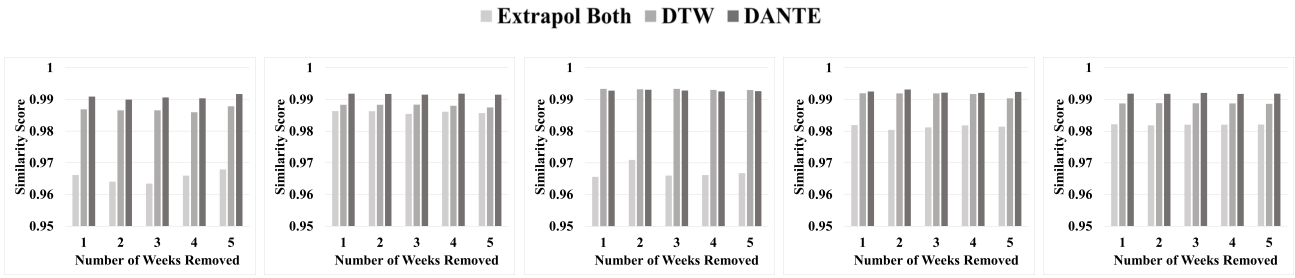
[12] P. Cinaglia and M. Cannataro. Network alignment and motif discovery in dynamic networks. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 11(38), October 2022.

[13] Connor Clark and Jugal Kalita. A comparison of algorithms for the pairwise alignment of biological networks. *Bioinformatics*, 30(16):2351–2359, 05 2014.

[14] Ana Conesa, Pablo Madrigal, Sonia Tarazona, David Gomez-Cabrero, Aurelio Cervera, Andrew McPherson, Michał W Szcześniak, Daniel J Gaffney, Laura L Elo, Xiang Zhang, and Ali Mortazavi. A survey of best practices for rna-seq data analysis. *Genome Biology*, 17(1):13, 2016.

[15] Frankhouser DE, Rockne RC, Uechi L, and Zhao D. State-transition modeling of blood transcriptome predicts disease evolution and treatment response in chronic myeloid leukemia. *bioRxiv*, Dec 2023.

[16] Piet C de Groen. Muons, mutations, and planetary shielding. *Frontiers in astronomy and space sciences*, 9, 2022.



(a) Control (b) CML (c) Tet-Off-On (d) TKI Treatment (e) All

Fig. 7: Similarity of the trajectory mapping obtained by DANTE, DTW, and Extrapol Both for five mice groups for increasing number of artificially removed networks (λ).

(a) Control (b) CML (c) Tet-Off-On (d) TKI Treatment (e) All

Fig. 8: Prediction accuracy over number of weeks removed. Different graphs correspond to mouse groups. Partial success is stacked on top of perfect success, with the combined bar forming the combined success rate.

[17] Rasha Elhesha, Aisharjya Sarkar, Pietro Cinaglia, Christina Boucher, and Tamer Kahveci. Co-evolving patterns in temporal networks of varying evolution. *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, page 494–503, 2019.

[18] Elliot S Gershon, Ney Alliey-Rodriguez, and Kay Grennan. Ethical and public policy challenges for pharmacogenomics. *Dialogues in Clinical Neuroscience*, 16(4):567–574, 2014.

[19] James P. Hamilton. Epigenetics: principles and practice. *Digestive diseases*, 29, 2011.

[20] Antony M Jose. Heritable epigenetic changes are constrained by the dynamics of regulatory architectures. *eLife*, 12, May 2024.

[21] Amnon Koren, Paz Polak, James Nemesh, Jacob J. Michaelson, Jonathan Sebat, Shamil R. Sunyaev, and Steven A. McCarroll. Differential relationship of dna replication timing to different forms of human mutation and variation. *American journal of human genetics*, 91:1033–40, 2012.

[22] Leonid Kruglyak and Deborah A Nickerson. Variation is the spice of life. *Nature genetics*, 27(3):234–236, 2001.

[23] Wang L, Li X, Zhang L, and Gao Q. Improved anticancer drug response prediction in cell lines using matrix factorization with similarity regularization. *BMC Cancer*, 17, Aug 2017.

[24] Guanyu Li, Ryan LeFebre, Alia Starman, Patrick Chappell, Andrew Mugler, and Bo Sun. Temporal signals drive the emergence of multicellular information networks. *Proceedings of the National Academy of Sciences of the United States of America*, 119, 2022.

[25] Bora Lim, Yoshimi Greer, Stanley Lipkowitz, and Naoko Takebe. Novel apoptosis-inducing agents for the treatment of cancer, a new arsenal in the toolbox. *Cancers*, 11, 2019.

[26] Ling Liu and Thomas A. Rando. Chapter 6 - aging of stem cells: Intrinsic changes and environmental influences. *Handbook of the Biology of Aging (Seventh Edition)*, pages 141–161, 2011.

[27] Margaret Mroziewicz and Rachel F Tyndale. Pharmacogenetics: a tool for identifying genetic factors in drug dependence and response to treatment. *Addiction science & clinical practice*, 5(2):17, 2010.

[28] National Institutes of Health (US) and Biological Sciences Curriculum Study. Understanding human genetic variation. *NIH Curriculum Supplement Series [Internet]*, 2007.

[29] Kuchaiev Oleksii, Milenković Tijana, Hayes Wayne Memišević Vesna, and Pržulj Nataša. Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface*, 7:1341–1354, 2010.

[30] Babita Pandey, Praveen Kumar Bhanodia, Aditya Khamparia, and Devendra Kumar Pandey. A comprehensive survey of edge prediction in social networks: Techniques, parameters and challenges. *Expert Systems with Applications*, 124:164–181, 2019.

[31] Rob Patro and Carl Kingsford. Global network alignment using multiscale spectral signatures. *Bioinformatics*, 28:3105–3114, December 2012.

[32] Hang T. T. Phan and Michael J. E. Sternberg. Pinalog: a novel approach to align protein interaction networks—implications for complex detection and function prediction. *Bioinformatics*, 28:1239–1245, May 2012.

[33] Scott W. Piraino, Valentina Thomas, Peter O'Donovan, and Simon J. Furney. Mutations: Driver versus passenger. *Encyclopedia of Cancer (Third Edition)*, pages 551–562, 2019.

[34] F. Rijo-Ferreira and J.S Takahashi. Genomics of circadian rhythms in health and disease. *Genome Medicine*, 11, 2019.

[35] Dan M. Roden, Russell A. Wilke, 2 Heyo K. Kroemer, PhD, and C. Michael Stein. Pharmacogenomics: the genetics of variable drug responses. *Circulation*, 123:1661–70, 2011.

[36] Ahmed S, Zhou Z, Zhou J, and Chen SQ. Pharmacogenomics of drug metabolizing enzymes and transporters: Relevance to precision medicine. *Genomics Proteomics Bioinformatics*, Apr 2018.

[37] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[38] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. In *Proceedings of the 3rd international conference on Knowledge Discovery and Data Mining*, pages 70–80. ACM, 2007.

[39] Seyedeh Hamideh Shalforoushan and Mehrdad Jalali. Link prediction in social networks using bayesian networks. *2015 The International Symposium on Artificial Intelligence and Signal Processing (AISP)*, pages 24 6 –250, 20 1 5.

[40] Chayaporn Suphavilai, Denis Bertrand, and Niranjan Nagarajan. Predicting cancer drug response using a recommender system. *Bioinformatics*, 34(22):3907–3914, 06 2018.

[41] Damian Szklarczyk and et al. The string database in 2023: protein-protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic Acids Research*, pages D638–D646, 2023.

[42] W Kalow BK Tang and L Endrenyi. Hypothesis: comparisons of inter-and intra-individual variations can substitute for twin studies in drug research. *Pharmacogenetics and Genomics*, 8(4):283–289, 1998.

[43] Vipin Vijayan, Dominic Critchlow, and Tijana Milenković. Alignment of dynamic networks. *Bioinformatics*, 33(14):i180–i189, 2017.

[44] Vipin Vijayan and Tijana Milenković. Multiple network alignment via multimagna++. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(5):1669–1682, 2018.

[45] Bowen Yan and Steve Gregory. Finding missing edges in networks based on their community structure. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 85(5):056112, 2012.

[46] Jinxiong Zhang, Cheng Zhong, Hai Xiang Lin, and Mian Wang. Identifying protein complexes from dynamic temporal interval protein-protein interaction networks. *BioMed research international*, 2019(1):3726721, 2019.