

## Subject Section

# Identification of co-evolving temporal networks

Rasha Elhesha, Aisharjya Sarkar, Christina Boucher, and Tamer Kahveci

Department of Computer & Information Science & Engineering, University of Florida, Gainesville, 32611, US.

\*To whom correspondence should be addressed.

Associate Editor: XXXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

## Abstract

**Motivation:** Biological networks describes the mechanisms which govern cellular functions. Temporal networks show how these networks evolve over time. Studying the temporal progression of network topologies is of utmost importance since it uncovers how a network evolves and how it resists to external stimuli and internal variations. Two temporal networks have co-evolving subnetworks if the topologies of these subnetworks remain similar to each other as the network topology evolves over a period of time. In this paper, we consider the problem of identifying co-evolving pair of temporal networks, which aim to capture the evolution of molecules and their interactions over time. Although this problem shares some characteristics of the well-known network alignment problems, it differs from existing network alignment formulations as it seeks a mapping of the two network topologies that is invariant to temporal evolution of the given networks. This is a computationally challenging problem as it requires capturing not only similar topologies between two networks but also their similar evolution patterns.

**Results:** We present an efficient algorithm, *Tempo*, for solving identifying coevolving subnetworks with two given temporal networks. We formally prove the correctness of our method. We experimentally demonstrate that *Tempo* scales efficiently with the size of network as well as the number of time points, and generates statistically significant alignments—even when evolution rates of given networks are high. Our results on a human aging dataset demonstrate that *Tempo* identifies novel genes contributing to the progression of Alzheimer's, Huntington's and Type II diabetes, while existing methods fail to do so.

**Availability:** Software is available online (<https://www.cise.ufl.edu/~relhesha/temporal.zip>).

**Contact:** relhesha@ufl.edu

**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

## 1 Introduction

Biological networks describe the interaction between molecules. They are frequently represented as graphs, where the nodes correspond to the molecules (e.g., proteins or genes) and the edges correspond to their interactions (Zhu *et al.*, 2007). Formally, we denote a biological network as  $G = (V, E)$  where  $V$  and  $E$  represent the set of nodes and the set of edges, respectively. Analysis of these networks enable the elucidation of cellular functions (Freyre-González *et al.*, 2008), the identification of variations in cancer networks (Leiserson *et al.*, 2015), and the characterization of variations in drug resistance (Charlebois *et al.*, 2014). Studying biological networks led to numerous computational challenges as well as methods which address these challenges. Network alignment is one of the most important of these challenges (Flannick *et al.*, 2006) as it has a profound set of applications ranging from the detection of conserved motifs to the

prediction of protein functions (Clemente *et al.*, 2006). This problem aims to find a mapping of the nodes of two given networks in which nodes that are similar in terms of content (i.e. homology) and interaction structure (i.e. topology) are mapped to each other. Hence, we represent the alignment between two given networks  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$  as a bijection function  $\psi : V_1 \rightarrow V_2$ , and the score resulting from alignment  $\psi$  as  $score(G_1, G_2 | \psi)$ . The network alignment problem seeks the function  $\psi$  that maximizes this score. We note that there are various ways to calculate the scoring function.

There are two categories of network alignment problem: local and global alignment. The former problem aims to find pairs of highly-conserved sub-networks in two given networks in which a sub-network of the query network is mapped to multiple sub-networks in the target network. Global network alignment aims to maximize the similarity in the networks in which all nodes in the query network are mapped to a set of nodes in the target network. Network alignment is a challenging task as the graph

and subgraph isomorphism problems which are known to be GI and NP-hard (Cook, 1971), reduce to them. In Section 2, we give a brief review of the methods addressing the global network alignment problem as the problem we consider in this paper is associated with that problem.

Biological networks have dynamic topologies (Przytycka et al., 2010). There are various reasons behind this dynamic behavior. For example, genetic and epigenetic mutations can alter molecular interactions (Sadi-kovic et al., 2008), and variation in gene copy number can affect the existence of interactions (De Smith et al., 2008). Due to this dynamic behavior, the topology of the network that models the molecular interaction evolve over time (Holme and Saramäki, 2012). Majority of the previous work on alignment of biological networks assume the network topology is static (Singh et al., 2007)—an assumption that ignores the history of network evolution, and may lead to biased or incorrect analysis. For example, identifying causes and consequences of the influence of external stimuli is impossible when analyzing static topologies. To address this oversight, we define a biological network using a model that accounts for the evolution of the underlying network at consecutive time points. We refer to this model as a *temporal network* (Hulovatyy et al., 2015). We view this model as containing a single snapshot of the network at each time point in a sequence of time points and thus, as a time series network. More formally, we denote a temporal network with  $t$  consecutive time points as  $\mathcal{G} = [G_1, G_2, \dots, G_t]$ , where  $G_i = (V, E_i)$  represents the topology of the network at the  $i$ th time point.

In this paper, we consider the problem of identifying *coevolving* subnetworks in a given pair of temporal networks. We say that two subnetworks are coevolving if their topologies remain similar even though their topologies evolve over time. We define this more formally as follows. We consider two input temporal networks  $\mathcal{G}^1 = [G_1^1, G_2^1, \dots, G_t^1]$  and  $\mathcal{G}^2 = [G_1^2, G_2^2, \dots, G_t^2]$ , where  $\forall i \in \{1, 2, \dots, t\}$ ,  $G_i^1 = (V^1, E_i^1)$  and  $G_i^2 = (V^2, E_i^2)$  represent  $\mathcal{G}^1$  and  $\mathcal{G}^2$  respectively at the time point  $i$ . Without losing generality, let  $\mathcal{G}^1$  be the query (smaller) network and  $\mathcal{G}^2$  be the target network, i.e.,  $|V^1| \leq |V^2|$ . An alignment of  $\mathcal{G}^1$  and  $\mathcal{G}^2$  maps  $\mathcal{G}_i^1$  to  $\mathcal{G}_i^2$  across all time points  $i$ . Thus, we represent the alignment of the two temporal networks  $\mathcal{G}^1$  and  $\mathcal{G}^2$  as a bijection of their nodes and denote it as a function  $\psi : V^1 \rightarrow V^2$ . We compute the score of the alignment  $\psi$  of  $\mathcal{G}^1$  and  $\mathcal{G}^2$ , denoted with  $score(\mathcal{G}^1, \mathcal{G}^2 | \psi)$ , as the sum of the scores of the alignment at all time points. Hence,  $score(\mathcal{G}^1, \mathcal{G}^2 | \psi) = \sum_{i=1}^t score(G_i^1, G_i^2 | \psi)$ . We assume  $\mathcal{G}^1$  is connected at all time points, but it may be impossible to find an alignment that is connected in the target network at all time points.

It is worth emphasizing that the temporal network alignment problem described above is dramatically different than existing network alignment problems, which can be categorized as follows: (i) pairwise alignment, (ii) multiple network alignment, and (iii) dynamic network alignment. We illustrate these problems as well as the temporal one in Figure 1. The pairwise network alignment problem (Figure 1(a)) ignores that the network topology evolves. Although the multiple alignment problem (Figure 1(b)) can consider more than two networks at once, it lacks the ability to capture the temporal changes since it treats all networks as having static topologies. The dynamic network alignment problem (Figure 1(c)) considers topological changes over time. It however, it seeks a different solution to the alignment problem at each time point. Thus, it can not identify coevolving subnetwork. A new algorithm is needed to capture such evolving characteristics. Unlike these alignment problems, temporal network alignment (Figure 1(d)) captures that network topologies coevolve over time.

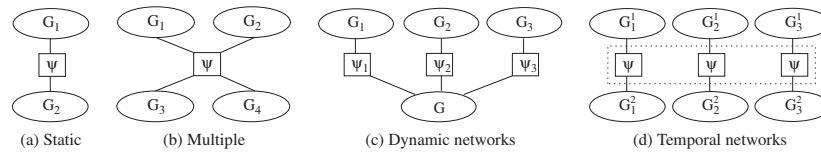
**Contributions in this paper.** We develop an efficient algorithm, *Tempo*, to identify coevolving subnetworks in a given pair of the temporal networks. Briefly, our algorithm first finds an initial alignment between the input networks  $\mathcal{G}^1$  and  $\mathcal{G}^2$  using the similarity score between pairs of aligned nodes across all time points. It then performs a dynamic programming strategy that maximizes the alignment quality (i.e. score) by

repeatedly altering the aligned nodes in the target network. We demonstrate the efficiency and accuracy of *Tempo* using both real and synthetic data. We compare the running time and the quality of the alignments found by *Tempo* against those of three existing alignment algorithms, IsoRank (Singh et al., 2007), MAGNA++ (Vijayan et al., 2015) and GHOST (Patro and Kingsford, 2012). Note that all these networks are tailored towards optimizing alignment at a single time point. To have a fair comparison, we allow each of these methods to consider each time point independently then apply the resulting alignments to all other time points and took the average. We show *Tempo* has competitive running time and generates significantly better alignments. We use a human brain aging (Berchtold et al., 2008) dataset, and integrate this dataset to analyze three phenotypes—two age related diseases (Alzheimer’s and Huntington’s) and one disease that is less prone to aging (Type II diabetes). We perform gene ontology analysis on the aligned genes reported by our algorithm and compared algorithms. Our algorithm could successfully align genes of the phenotype query (i.e. the underlying disease) to strongly related genes in the target network despite their evolving topologies unlike other algorithms. Consequently, we could predict disease-related genes based on the generated alignment using *tempo* which suggests that *Tempo* generates alignments that reflect the evolution of nodes topologies through time as well as their homological similarities while other methods only focuses on static and independent topologies. Lastly, we observe that alignments of age related phenotype is significantly higher than alignment of non age phenotype which reflects their high evolution rates and shows that *Tempo* could identify between different queries.

## 2 Related Work

One of the key studies on pairwise global network alignment is IsoRank (Singh et al., 2007), which is based on the conjecture that two nodes should be matched if their respective neighbors can also be matched. It formulates the alignment as an eigenvalue problem and computes the similarity between pairs of nodes from two given networks as a combination of their homological and topological similarities. It obtains the global alignment of the two given networks using their maximum weight bipartite match with the scores as the weights. The GRAAL (GRAPH ALigner) family (Kuchaiev and Pržulj, 2011) of global network alignment methods use the graphlet degree similarity to align two networks. Briefly, the graphlet-degree of a node counts the number of graphlets (i.e. induced subgraph) that this node touches, for all graphlets of 2 to 5 nodes. GRAAL (Kuchaiev et al., 2010) first selects a pair of nodes (one from each of the two given networks) with high graphlet degree signature similarity as the seed of the alignment, and greedily expands the alignment by iteratively including new pairs of similar nodes. H-GRAAL (Milenković et al., 2010), MI-GRAAL, and L-GRAAL algorithms also belong to the same family. The SPINAL algorithm (Aladağ and Erten, 2013) iteratively grows the alignment based on a priori computed node similarity score. MAGNA (Saraph and Milenković, 2014) optimizes the edge conservation between two networks using a genetic algorithm. There are several other methods for pairwise network alignment (Kelley et al., 2004; Phan and Sternberg, 2012; Gülsoy et al., 2012; Hasan and Kahveci, 2015; Patro and Kingsford, 2012; Neyshabur et al., 2013; Sun et al., 2015; Hu et al., 2013). Although the underlying algorithms of these methods vary, the end goal is similar to those discussed above.

Several algorithms address the multiple network alignment (Alkan and Erten, 2013; Ibragimov et al., 2014; Sahraeian and Yoon, 2013). IsoRankN (Liao et al., 2009) extends IsoRank. It adopts spectral clustering on the induced graph of pairwise alignment scores. The algorithm developed by Shih et al. (Shih and Parthasarathy, 2012) is a seed-expansion heuristic that first selects a set of node pairs with high similarity scores using a clustering algorithm, and then expands these pairs by aligning nodes that maximizes the number of the total conserved edges of aligned nodes.



**Fig. 1.** This figure represents different network alignment problems in different types of biological networks. (a) This represents the alignment between two input static networks. (b) This represents the alignment between multiple time points where each network represent a different organism. (c) This represents the alignment between two input networks where one of them is static and one of them is dynamic. Here, there exist different alignment between the static network and each version of the dynamic network. (d) This represents the alignment between two input temporal networks where each have time specific snapshots that was taken at three specific time points. Here, the alignment is persist across all time points.

INQ (Hasan and Kahveci, 2014) aligns a dynamically evolving query network with one static target network. It uses ColT (Hasan and Kahveci, 2013) to find an initial alignment of the initial query, then it observes the differences between the topologies of the already aligned query network and the new query network, and finally, uses these differences to refine the alignment found for the previous query and generate alignment of the current query network. DynaMAGNA++ (Vijayan *et al.*, 2017) aligns two dynamic networks. It assigns a value to each node based on how the incident edges and graphlets change through dynamic events. It assigns each node a value based on dynamic graphlet degree vector (DGDV) of graphlets up to size four. It considers a pair of nodes from two networks similar if their DGDVs are similar.

### 3 Problem Formulation

In this section, we develop a new scoring function,  $score(G_1^1, G_2^2 | \psi)$ , that integrates the similarities of the aligned nodes and their evolving topologies, and includes a penalty for each disconnected component in the aligned subnetworks of the target network at each time point. Next, we introduce the terminology and discuss how we drive our scoring function.

Given a network  $G = (V, E)$  and a subset of nodes  $\bar{V}$ , we define the induced subnetwork of  $\bar{V}$  in  $G$  as the nodes in  $\bar{V}$  and all incident edges (i.e.,  $\bar{E} = \{\bar{V} \times \bar{V}\} \cap E$ ). We denote this induced network as  $\bar{G} = (\bar{V} | G)$ . We say two nodes  $u$  and  $v$  in  $G$  are connected if there exists a path between  $u$  and  $v$  in  $G$ . We say a subset of nodes in  $G$  form a *connected component* if all pairs of nodes in that subset are connected in  $G$ . We define a subset of nodes  $\bar{V}$  in  $G$  as a *maximum connected component* if the following conditions hold: (i)  $\bar{V}$  is a connected component in  $G$ , and (ii) there is no node in  $V - \bar{V}$  which is connected to a node in  $\bar{V}$ . In the rest of the paper, we use the term “connected component” instead of “maximum connected component”. We denote the number of connected components of a given network  $G$  with  $NCC(G)$ .

Given two temporal networks with  $t$  time points,  $\mathcal{G}^1 = [G_1^1, G_2^1, \dots, G_t^1]$  and  $\mathcal{G}^2 = [G_1^2, G_2^2, \dots, G_t^2]$ , we denote the similarity between a pair of nodes  $u \in V^1$  and  $v \in V^2$  at time point  $i$  ( $1 \leq i \leq t$ ) with  $S_i(u, v)$ . We use an existing pairwise alignment method to calculate  $S_i(u, v)$ . The alignment function  $\psi$  maps all nodes in  $V^1$  to a subset of the nodes in  $V^2$ . We denote this subset with  $\Psi(V^1)$  (i.e.  $\Psi(V^1) = \{\psi(u) | u \in V^1\}$ ). We note that  $\psi$  yields an induced subnetwork  $(\Psi(V^1) | G_i^2)$  of  $G_i^2$  for each time point  $i$ , and each induced subnetwork  $(\Psi(V^1) | G_i^2)$  forms one or more connected components. See Figure 2(a) for an illustration of this latter point. We denote the number of connected components of the induced subnetwork  $(\Psi(V^1) | G_i^2)$  at time point  $i$  as  $NCC(\Psi(V^1) | G_i^2)$ . If the number of connected components at time point  $i$  is greater than one then the corresponding induced subnetwork is disconnected. We incur a penalty to account for the missing edges which would connect the disconnected components, and apply this penalty to each disconnected component.

The minimum number of edges needed to join  $NCC(\Psi(V^1) | G_i^2)$  connected components is  $NCC(\Psi(V^1) | G_i^2) - 1$ . We penalize each edge insertion with a constant value denoted with  $\delta$ , where  $\delta \geq S_i(u,$

$v)$ ,  $\forall u \in V^1, v \in V^2$  and  $i \in \{1, 2, \dots, t\}$ . We define the score of the alignment  $\psi$  at time point  $i$  as:  $score(G_i^1, G_i^2 | \psi) = \sum_{u \in V^1} S_i(u, \psi(u)) - \delta(NCC(\Psi(V^1) | G_i^2) - 1)$ . We define the temporal network alignment as

$$\operatorname{argmax}_{\psi} \left\{ \sum_{i=1}^t \left( \sum_{u \in V^1} S_i(u, \psi(u)) - \delta(NCC(\Psi(V^1) | G_i^2) - 1) \right) \right\}. \quad (1)$$

### 4 Methods

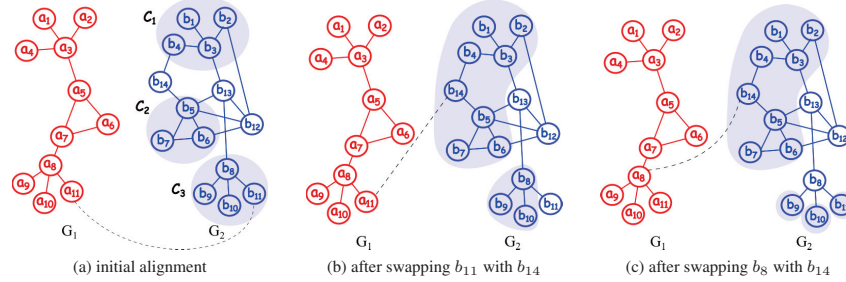
**Overview.** Our algorithm for solving the temporal network alignment problem has two phases. The first phase finds an initial alignment between the input networks  $\mathcal{G}^1$  and  $\mathcal{G}^2$  using the similarity score between pairs of aligned nodes across all time points. The induced subnetwork of  $\mathcal{G}^2$  obtained by this alignment may be disconnected since this phase ignores the penalty incurred by edge insertions. The second phase reduces the number of connected components, improving the alignment score. In the second phase, we improve the alignment between the input networks by *swapping* a subset of the nodes in  $\mathcal{G}^2$  that are aligned with nodes in  $\mathcal{G}^1$  with other nodes in  $\mathcal{G}^2$ . In order to swap a node  $v_i \in \Psi(V^1)$  with  $v_j \in V^2 - \Psi(V^1)$ , we update the alignment function  $\psi$  to  $\psi'$  such that  $\forall u \in \bar{V}$  one of the two conditions is satisfied: (i)  $\psi'(u) = v_j$  if  $\psi(u) = v_i$ ; and (ii)  $\psi'(u) = \psi(u)$  if  $\psi(u) \neq v_i$ . Figure 2 illustrates this. Here, initially  $b_{11}$  is aligned to  $a_{11}$  (Figure 2(a)). Swapping  $b_{11}$  with  $b_{14}$  updates the alignment function so that  $b_{14}$  is aligned to  $a_{11}$  (Figure 2(b)). We observe that this swapping reduces the number of connected components in the induced subnetwork of  $\mathcal{G}^2$  by one. Notice that if we swap  $b_8$  with  $b_{14}$  (instead of  $b_{11}$  with  $b_{14}$ ) then the number of connected components increases (Figure 2(c)).

We note that the number of connected components may simultaneously decrease at one time point and increase at other time points when we swap two nodes. We prove that the problem of finding the subset of node swaps that minimizes the number of connected components across all time points is NP-hard. We give a reduction from the Maximum Coverage problem (Feige, 1998) to this problem later in this section.

**Algorithm details.** Tempo takes two networks ( $\mathcal{G}^1$  and  $\mathcal{G}^2$ ) and the maximum number of allowed swaps (denoted as  $k$ ) as input. In the following, we explain the two phases of our method in detail.

**Phase I (Initialization).** Here, we construct an initial alignment of  $\mathcal{G}^1$  and  $\mathcal{G}^2$ . There exists several algorithms to perform pairwise alignment of two static networks at a single time point. Each of these methods assign similarity scores to all node pairs (one from the first network and one from the second) and then choose the alignment that maximizes the total score of all aligned node pairs. We adopt one of these methods to obtain the similarity scores of each network pairs ( $G_i^1, G_i^2$ ) at each time point  $i$ , and use the outputted scores to calculate an initial alignment. We denote the similarity of the node pair  $(u, v)$ ,  $u \in V^1$  and  $v \in V^2$  generated by such method at the  $i$ th time point with  $S_i(u, v)$ .

We generate an initial alignment  $\psi_0$  as follows. We first construct a weighted bipartite network  $G_{bp} = (V^1, V^2, \mathcal{E})$  as follows: we insert an edge in  $G_{bp}$  between each pair of nodes  $(u, v)$  such that  $u \in V^1$  and  $v \in$



**Fig. 2.** This figure represents an alignment between two networks  $G_1$  and  $G_2$ . Each node in the query network  $G_1$  has a one-to-one mapping with a node in the network  $G_2$ . The dashed line between two nodes emphasizes that they are mapped to each other. (a) This represents a hypothetical alignment where  $a_i$  is aligned with  $b_i$  for all  $1 \leq i \leq 11$ . The induced subnetwork of the aligned nodes in  $G_2$  forms three connected components;  $C_1 = \{b_1, b_2, b_3, b_4\}$ ,  $C_2 = \{b_5, b_6, b_7\}$ , and  $C_3 = \{b_8, b_9, b_{10}, b_{11}\}$ . Gap nodes are  $\{b_{12}, b_{13}, b_{14}\}$ . (b) After swapping  $b_{11}$  with  $b_{14}$ . This swapping results in two connected components in  $G_2$ . (c) After swapping  $b_8$  with  $b_{14}$ . The aligned nodes in  $G_2$  form four connected components.

$V^2$ . We set the weight of the edge  $(u, v)$  as the similarity between nodes  $u$  and  $v$  aggregated over all time points. We denote the similarity as  $S(u, v) = \sum_1^t S_i(u, v)$ . The maximum-weight bipartite matching algorithm maps each node in  $V^1$  to a node in  $V^2$  (Papadimitriou and Steiglitz, 1998). This mapping represents the initial alignment,  $\psi_0$ . We call the nodes in  $V^2$  that are not mapped to any node in  $V^1$  as *gap nodes* and denote with  $F = V^2 - \Psi(V^1)$ .

Phase II (select  $k$  swapping pairs). Here, we describe our dynamic programming algorithm that selects a set of  $k$  swaps which maximize the alignment score by reducing the number of connected components in the induced alignment across all time points of  $\mathcal{G}^2$  (see Equation 1).

We denote a set of  $r$  swaps with  $\Delta = \{(u_1, v_1), (u_2, v_2), \dots, (u_r, v_r)\}$  with  $\forall i \neq j, u_i \neq u_j$  and  $v_i \neq v_j$ . We denote the alignment after applying the swaps in a given set  $\Delta$  as  $\psi_\Delta$ . Let us denote the optimal set of  $r$  swaps for the alignment  $\psi$  with  $\text{solution}(r, \psi, \mathcal{G}^1, \mathcal{G}^2)$ . Also, for a given  $u_i \in \Psi(V^1)$ , we denote the optimal set of  $r$  swaps for the alignment  $\psi$  which contains the swap pair  $(u_i, v_i)$ ,  $\exists v_i \in F$ , with  $\text{solution}(r, u_i, \psi, \mathcal{G}^1, \mathcal{G}^2)$ .

Our algorithm works iteratively. In the first iteration, our algorithm selects one swapping pair for each aligned node  $u_i \in \Psi(V^1)$  as

$$\text{solution}(1, u_i, \psi, \mathcal{G}^1, \mathcal{G}^2) = \underset{\Delta = \{(u_i, v_i), v_i \in F\}}{\text{argmax}} \{ \text{score}(\mathcal{G}^1, \mathcal{G}^2 | \psi_\Delta) \}.$$

At each subsequent iteration  $r$  where  $2 \leq r \leq k$ , for each aligned node  $u_i \in \Psi(V^1)$ , our algorithm selects a set of  $r$  swapping pairs denoted with  $\text{solution}(r, u_i, \psi, \mathcal{G}^1, \mathcal{G}^2)$  by adding one swapping pair  $(u_i, v_i)$ ,  $\exists v_i \in F$ , to the previously selected  $r-1$  pairs as follows.

$$\underset{\Delta = \{(u_i, v_i)\} \cup \text{solution}(r-1, u_j, \psi, \mathcal{G}^1, \mathcal{G}^2), \Theta}{\text{argmax}} \{ \text{score}(\mathcal{G}^1, \mathcal{G}^2 | \psi_\Delta) \}. \quad (2)$$

Here  $\Theta$  represents the necessary conditions to include the  $(u_i, v_i)$  swap pair with a set of  $r-1$  swap pairs as

$$\Theta = (v_i \in F) \text{ AND} \\ (u_j \in \Psi(V^1)) \text{ AND} \\ (\nexists v \in F, \text{ such that } (u_i, v) \in \text{solution}(r-1, u_j, \psi, \mathcal{G}^1, \mathcal{G}^2)) \\ \text{AND } (\nexists u \in \Psi(V^1), \text{ such that} \\ (u, v_i) \in \text{solution}(r-1, u_j, \psi, \mathcal{G}^1, \mathcal{G}^2)).$$

The first condition above ensures that node  $u_i$  is swapped with a gap node and the second ensures the dynamic programming iterates over all size  $r-1$  swap sets for all aligned nodes of  $\mathcal{G}^2$ . The third condition ensures that the aligned node  $u_i$  has not already been swapped in the  $r-1$  sized swap set. The final condition is the dual of the previous one, as it ensures that the gap node  $v_i$  has not already been swapped in the  $r-1$  sized swap set. When these conditions hold, the two nodes  $u_i$  and  $v_i$  can be swapped

and included into the existing set of  $r-1$  swaps without conflicting with any of the existing swaps.

We report the output of the algorithm at end of the  $k$ th iteration as set of  $k$  swaps with the highest alignment score using equation

$$\Delta = \text{solution}(k, \psi, \mathcal{G}^1, \mathcal{G}^2) = \underset{\text{argmax}_{u_i \in \Psi(V^1), \Delta_i = \text{solution}(k, u_i, \psi, \mathcal{G}^1, \mathcal{G}^2)} \{ \text{score}(\mathcal{G}^1, \mathcal{G}^2 | \psi_{\Delta_i}) \}}{\text{argmax}} \quad (3)$$

We represent the set cardinalities  $|V^1|$ ,  $|V^2|$ , and  $|F|$  with  $m$ ,  $n$ ,  $l$ , respectively. The complexity of our algorithm is  $\mathcal{O}(m^2 n^2) + \mathcal{O}(mn \log m) + ml \sum_{i=1}^t |E_i^2| + \mathcal{O}(k^2 l^2 m)$ . We provide the derivation of this complexity in Section A.1 of the appendix. We note that  $k \leq NCC(\psi(V^1) | \mathcal{G}^2) - 1$ . This value is either given as input or we set it to  $NCC(\psi(V^1) | \mathcal{G}^2) - 1$ .

**Proof of correctness.** Here, we formally proof the correctness of our algorithm. We say that swapping the pair of nodes  $(u_i, v_i)$  is *proper* if that the swapping does not increase the number of connected components of the aligned nodes. We first prove that our algorithm will always find a proper swapping node  $u_i$  from the set of aligned node for each gap node  $v_i$ . We first present a lemma which is necessary for the proof of our first theorem. Let us denote the degree of a node  $v$  (i.e. number of edges connected to this node) within a component  $C_i = (V_c, E_c)$  of the induced subnetwork  $\bar{G}_i^2 = (\Psi(V^1) | \mathcal{G}_i^2)$  at time point  $i$  with the function  $\text{deg}(v | C_i)$ .

**Lemma 1.** Given an undirected subnetwork of  $\bar{G}_i^2$ ,  $\bar{G}_i^2 = (\Psi(V^1) | \mathcal{G}_i^2)$  where  $|V_c| = z$  and  $\bar{G}_i^2$  is acyclic network (has no cycle) within its topology, then  $\sum_{v \in C_i} \text{deg}(v | C_i) = 2(z-1)$ .

*Proof.* Since  $C_i$  is a connected subnetwork with no cycles, the number of edges in  $C_i$  equals  $z-1$  edges. Each edge belongs to an undirected network increases the sum of the network nodes degrees by two. Thus,  $\sum_{v \in C_i} \text{deg}(v | C_i) = 2(z-1)$ . ■

**Lemma 2.** Given a gap node  $v_i$  that connects at least two connected components, there exist at least one aligned node  $u_i$  which we can swap with  $v_i$  without increasing the number of connected component.

*Proof.* We formally prove this by induction on the size of connected components that  $u_i$  belongs to.

**Base case.** We consider a component  $C_i = (V_c, E_c)$  where  $|V_c| = 2$  and  $v_i$  is connected to  $C_i$  through  $u_j$ , and assume  $u_i$  belongs  $C_i$ . If we swap  $v_i$  with  $u_i$ , then  $C_i$  will contain  $u_j$  and  $v_i$  which corresponds to one component. Thus, the number of connected components of  $C_i$  is still one after swapping.

**Induction hypothesis.** We assume there exists a node  $u_i$  for all components of size  $q$  nodes that can be swapped without disconnecting its component. We consider two cases of one component  $C_i$  where  $v_i$  is connected to through  $u_j$ . The first case is when  $C_i$  contains at least one cycle

with the set of nodes,  $V_{c1} = \{v_1, v_2, \dots, v_n\}$ . It follows that for each node  $u_i \in V_{c1}$  and  $u_i \neq u_j$ ,  $u_i$  can be swapped with  $v_i$  without disconnecting  $C_i$ . In the second case,  $C_i$  represents acyclic network with no cycles. Next, we prove our theorem in this case by contradiction. First, we assume that the number of nodes in  $C_i$  with degree equal to 1 is less than 2. Consequently,  $\sum_{v \in C_i} \deg(v|C_i) \geq 2(z-1) + 1$ , which contradicts Lemma 1. Thus, the number of nodes in  $C_i$  with degree equal to 1 is at least 2 nodes and thus,  $\exists v, w \in C$  st.  $\deg(v|C) = 1$  and  $\deg(w|C) = 1$  and  $v \neq w$ . Therefore, we can swap  $v_i$  with either  $v$  or  $w$ . ■

Next, we prove that swapping a gap node  $v_i$  with an aligned node  $u_i$  at each iteration will increase the alignment score  $score(\mathcal{G}^1, \mathcal{G}^2|\psi)$ , showing that the alignment score will always improve by our dynamic programming algorithm.

**Theorem 1.** *Given a value of  $\delta$  where  $\delta$  is greater than or equal to  $S(\psi(u_i), u_i)$  for all  $u_i \in V^2$ . At each iteration of our algorithm,  $score(\mathcal{G}^1, \mathcal{G}^2|\psi)$  monotonically increases.*

*Proof.* We assume that our algorithm chooses one pair of nodes to swap; a gap node  $v_i$  and aligned node  $u_i$  which will connect  $x$  number of components. We note that the condition  $x \geq 2$  must be satisfied for  $v_i$  to be considered for swapping. Also, it follows from Lemma 2 that if we swap  $v_i$  and  $u_i$  then the number of connected components will not increase. Thus, the difference in the score equals  $D = \delta(x-1) - p_{uv}$  where  $p_{uv}$  is the difference in pairwise score from swapping (i.e.  $p_{uv} = S(\psi(u_i), u_i) - S(\psi(u_i), v_i)$ ). Since  $\delta$  is greater than or equal to  $S(u, v) \forall u \in V^1$  and  $v \in V^2$ , then  $\delta(x-1) \geq p_{uv}$ . Consequently,  $D \geq 0$  and  $score(\mathcal{G}^1, \mathcal{G}^2|\psi)$  will not decrease. ■

**Proof of NP-hardness.** Here, we prove that our problem is NP-hard. To do that, we reduce the *Maximum Coverage Problem (MCP)*, which is known to be NP-hard (Karp, 1972), to our problem. Given a positive integer  $k$  and a collection of sets,  $S = \{S_1, S_2, \dots, S_m\}$ , MCP seeks the subset  $\hat{S} \subseteq S$  such that  $|\hat{S}| \leq k$  and the number of covered elements  $|\bigcup_{S_i \in \hat{S}} S_i|$  is maximized.

We reduce MCP to an instance of our problem. Let  $U = \{x_1, x_2, \dots, x_n\}$  be the union of elements in  $S$  (i.e.  $U = \bigcup_{S_i \in S} S_i$ ). We construct a target temporal network  $\mathcal{G}^2$  with one time point  $G^2 = (V^2, E^2)$  as follows. We initialize  $G^2$  as  $V^2 = \emptyset$  and  $E^2 = \emptyset$ . Next, we add a node  $a_j$  in  $G^2$  for each element  $x_j \in U$ . Also, for each set  $S_i \in S$ , we add two nodes  $f_i$  and  $b_i$  in  $V^2$ . Formally,  $V^2 = \{a_1, a_2, \dots, a_n\} \cup \{b_1, b_2, \dots, b_m\} \cup \{f_1, f_2, \dots, f_m\}$ . Next, we populate the set of edges  $E^2$ . To do that, for all  $S_i \in S$  and  $x_j \in S_i$ , we insert the edge  $(f_i, a_j)$  in  $E^2$ . In addition, for all pair of sets  $S_i, S_j \in S$ , where  $i < j$ , we insert the edge  $(f_i, f_j)$  in  $E^2$ . Finally, for a given query network  $G^1 = (V^1, E^1)$ , we construct the set of nodes in  $G^2$  aligned to those in  $G^1$  as  $\Psi(V^1) = \{a_1, a_2, \dots, a_n\} \cup \{b_1, b_2, \dots, b_m\}$ . Thus, the set of gap nodes is  $\{f_1, f_2, \dots, f_m\}$ . Notice that, the subnetwork of  $G^2$  induced by  $\Psi(V^1)$  has  $m+n$  nodes but it contains no edges as all the edges in  $G^2$  are connected to a gap node by our construction. Thus, the alignment yields  $n+m$  connected components as each node in  $\Psi(V^1)$  represents a component.

Recall that each swapping operation swaps an aligned node with a gap node. Also, recall that the optimization problem we solve for aligning temporal networks aims to find at most  $k$  swaps, such that after applying those swaps, the number of connected components  $NCC(\Psi(V^1) | G^2)$  is minimized (see Section 3). We call this optimization problem *minimum Connected Component Problem (mCCP)* in the rest of this proof. Next, we prove that MCP is maximized if and only if mCCP is minimized.

First, we prove that if there exists a solution to mCCP, then there exists a solution to MCP. In other words, we prove that minimizing mCCP maximizes MCP. Let us denote the nodes corresponding to the elements in a set  $S_i$  with  $A_i = \bigcup_{x_j \in S_i} \{a_j\}$ . In our problem instance, a swap operation swaps  $f_i$  with a node in the set  $V^2 - A_i - \{f_i\}$ . This is because all

nodes in  $A_i$  are connected to  $f_i$ , and thus swapping  $f_i$  with a node not in  $A_i$  ensures that all nodes in  $S_i \cup \{f_i\}$  form one connected component. Therefore, to minimize the number of connected components, we swap  $f_i$  with one of the nodes which is not a part of this connected component. To ensure that, we swap  $f_i$  with a node in the set  $\{b_1, b_2, \dots, b_m\}$ . Since all nodes in this set are disconnected, swapping  $f_i$  with any node in this set will yield the same number of connected components. Let us assume that the solution to mCCP performs  $k$  swaps. Following from the discussion above, without losing generality, we assume that these swaps are  $\{(f_1, b_1), (f_2, b_2), \dots, (f_k, b_k)\}$ . Notice that after these swaps, the nodes in  $(\bigcup_{i=1}^k A_i) \cup \{f_1, f_2, \dots, f_k\}$  forms one connected component, and all remaining nodes are isolated. Let us denote the number of connected components after these swaps with  $\beta$ . Let us denote the number of nodes in  $(\bigcup_{i=1}^k A_i)$  with  $\tau$ . Notice that  $\tau$  also reflects the number of elements covered in  $(\bigcup_{i=1}^k S_i)$ . We have  $\beta = (m-k) + (n-\tau) + 1$ .

In the formulation above, the first term  $(m-k)$  is the number of nodes  $b_j$  which are not swapped with a gap node. Since all those nodes are isolated, each one forms a connected component by itself. The second term  $(n-\tau)$  is the number of nodes  $a_j$  which are not included in the set  $(\bigcup_{i=1}^k A_i)$ . These nodes remain isolated even after swapping of nodes. The last term (i.e., 1) is the connected component containing the nodes in  $(\bigcup_{i=1}^k A_i) \cup \{f_1, f_2, \dots, f_k\}$ . After minor algebraic manipulation, we rewrite the equation above as  $\beta = (m+n-k+1) - \tau$ . In this equation, the parameters  $m$ ,  $n$ , and  $k$  are input to the given mCCP problem, and thus we denote the first term above with the constant  $c = m+n-k+1$ . Therefore, we have  $\beta = c - \tau$ . In this equality the smaller the value of  $\beta$  is, the larger  $\tau$  gets. Thus, minimizing the number of connected components  $\beta$  in mCCP maximizes the number of elements covered in MCP.

Second, we prove that if there exists a solution to MCP, then there exists a solution to mCCP. In other words, we prove that maximizing MCP minimizes mCCP. Let us assume that the solution to MCP is  $\hat{S} = \{S_1, S_2, \dots, S_k\}$ . The number of elements covered by this solution is  $\tau = |\bigcup_{S_i \in \hat{S}} S_i|$ . By constructing an instance of mCCP as described above, we have  $k$  swaps denoted with the set  $\{(f_1, b_1), (f_2, b_2), \dots, (f_k, b_k)\}$ . Consequently, after performing these swaps, the nodes in  $(\bigcup_{i=1}^k S_i) \cup \{f_1, f_2, \dots, f_k\}$  forms one connected component, and all the remaining nodes are isolated. Let us denote the number of connected components with  $\hat{\beta}$ . We have  $\hat{\beta} = (m-k) + (n-\tau) + 1$ .

After minor algebraic manipulation, we rewrite the equation above as  $\tau = (m+n-k+1) - \hat{\beta}$ . Since  $m$ ,  $n$ , and  $k$  are input parameters, we have  $\tau = c - \hat{\beta}$ , where  $c$  is a constant ( $c = (m+n-k+1)$ ). In this equality, the larger the value of  $\tau$  is, the smaller  $\hat{\beta}$  gets. Thus, maximizing  $\tau$  in MCP results in maximizing  $\hat{\beta}$  in mCCP.

Lastly, the proof we describe above reduces an instance of MCP to an instance of mCCP in polynomial time and space as it requires only building a network with  $\mathcal{O}(n+m)$  nodes and edges. Thus, we conclude that the mCCP problem is NP-hard.

## 5 Results and Discussion

We evaluate the performance of our algorithm on synthetic and real data. Next, we describe both datasets in detail.

**Real Dataset.** We obtain our real dataset from two sources. The first one is the human brain aging dataset (Berchtold *et al.*, 2008). Recall that this dataset contains gene expressions of 173 samples obtained from 55 individuals spanning 37 ages from 20 to 99 years. The ages in this dataset are not uniformly spaced. In order to bring consecutive time gaps to a more uniform values, we remove two data points which have an age gap of more than 5 years from their successive age values, leading to 35 ages. We select five temporal networks each having seven time points. Next, we explain how we do that for the first temporal network. We start with the first (i.e., youngest) time point in the aging data. We then skip the next four time points and take the sixth time point in aging data iteratively until we have



seven time points. Similarly, for  $1 < j \leq 5$ , we select the  $j$ th temporal network starting from the  $j$ th time point. In this manner, we form five non-overlapping and interleaved temporal networks. In order to integrate static PPI network with gene expression data to form age-specific PPI networks, we set a cut-off on the gene-expression value. All the interactions that have a lower transcription value for either or both the proteins are removed from the corresponding age-specific network. We use the protein-protein interaction (PPI) network data from BioGRID (Breitkreutz et al., 2007). For the second source, we select phenotype specific query temporal networks from this dataset. We use two neurodegenerative disorders which are conjectured to be age-related (Alzheimer's and Huntington's) and a third one which we expect to be less prone to aging (Type II diabetes). We retrieve the gene sets specific to these three diseases from KEGG database (Ogata et al., 1999). We form three query PPI temporal networks by keeping only the interactions where both the interactors are from each of the three phenotype-specific (Alzheimer's, Huntington's or Type II Diabetes) gene set.

**Synthetic dataset.** We generate synthetic networks to observe the performance of our method under a wide spectrum of parameters classified under two categories: (i) network size and (ii) temporal model parameters, namely number of time points, temporal rate, and cold rate. We vary the target network size to take values from  $\{100, 250, 500, 750, 1000\}$ . We fix the network density to two edges per node on the average (i.e., mean node degree is set to four). We randomly select  $G_1^1$  as a connected subnetwork of  $G_1^2$ . We set the size of the query network to 50 nodes. We generate target network  $G_1^2$  using Barabási-Albert (BA) (Barabási and Albert, 1999) model as this model produces scale-free networks. In order to explain the parameters in the second category, we describe how we generate the query and target networks  $G_1^1$  and  $G_1^2$  at the first time point. We then explain how we use the parameters in this category to build the query and target networks at the remaining time points.

We generate the subsequent networks for the remaining time points using the three parameters in the second category above as follows. The first parameter is the number of time points  $t$  in  $\mathcal{G}^1$  and  $\mathcal{G}^2$ . We use 5, 10, 15, and 20 time points in our experiments. Recall that we select a subnetwork of the target network  $G_1^2$  as the first query network  $G_1^1$ . We mark all nodes and edges in  $G_1^1$  within this subnetwork as *cold* nodes and edges respectively. We mark all other nodes and edges in  $G_1^2$  as *hot*. Next, we iteratively generate the networks  $G_i^1$  and  $G_i^2$  at the  $i$ th time point ( $i > 1$ ) from  $G_{i-1}^1$  and  $G_{i-1}^2$  respectively as follows. Let us denote temporal and cold rates (two real numbers) with  $\epsilon$  and  $\epsilon^c$  respectively such that  $0 \leq \epsilon^c \leq \epsilon \leq 1$ . Let us denote the ratio of cold edges to the total number of edges in the target network  $G_i^2$  with  $\gamma$ . We calculate the hot rate, denoted with  $e^h$ , from temporal rate and cold rate as  $e^h = (\epsilon - \epsilon^c \gamma) / (1 - \gamma)$ . Conceptually, hot and cold rates model the rate of evolution of hot and cold edges between two consecutive time points respectively. More specifically, for each subsequent time point  $i$ , we generate  $G_i^2$  by randomizing  $G_{i-1}^2$  as follows. We iterate over all edges in  $G_{i-1}^2$ . For each edge  $e$ , if it is a cold edge we remove it with probability  $\epsilon^c$  and insert a new edge between two randomly chosen cold nodes. If  $e$  is a hot edge, we remove it with probability  $e^h$  and insert a new hot edge between two random nodes (with at least one being a hot node). We generate query networks at subsequent time points using almost the same procedure with the only difference being that all edges are cold. We generate datasets by varying  $\epsilon$  and  $\epsilon^c$  to take the values  $\{0.05, 0.1, 0.2, 0.4, 0.8\}$  and  $\{0.05, 0.1, 0.2\}$  respectively. For each parameter setting we generate 10 target and query temporal networks.

Recall that, we generate the scoring matrix based on both homology and topology similarities. We generate the homology score between two pair of nodes  $u \in V^1$  and  $v \in V^2$  as follows. If  $v$  was originally selected as cold node and  $u$  is the same as  $v$ , then we generate a homology score between  $u$  and  $v$  from log-normal distribution (Johnson et al., 1994) with mean  $2\mu$  and standard deviation  $\sigma$ . Otherwise, we randomly generate

the homology score between  $u$  and  $v$  from log-normal distribution with mean  $\mu$  and standard deviation  $\sigma$ . In this way, we allow nodes in query network to be likely to align to nodes in the target network that were originally extracted from. In this paper, we set  $\mu$  and  $\sigma$  to be 2 and 0.25 respectively. Notice that the homology scores do not change through time points, although topology scores do. Thus, evolution through time points of query and target networks may affect how the query is aligned to the cold region in the target network. We set the edge insertion penalty  $\delta$  to be

$$\max_{u \in V^1, v \in V^2} S(u, v).$$

We compare the accuracy and running time of our algorithm against IsoRank, MAGNA++ and GHOST (see Appendix, Section A.3 for extended results). IsoRank, MAGNA++ and GHOST are designed to align two networks at a single time point. We therefore find the alignment using each of these methods at each time point, impose the alignment to all the time points and report the average. We analyze the biological significance of our results on real data by performing gene ontology analysis and exploring publication evidence. We implemented Tempo in C++, performed all experiments on a computer equipped with AMD FX(tm)-8320 Eight-core Processor 1.4 GHz CPU, 32 GB of RAM running Linux operating system, and used  $\alpha = 0.7$  unless otherwise stated.

### 5.1 Evaluation of recovered region

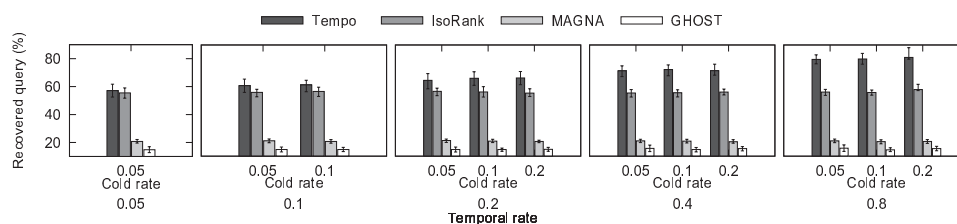
In this experiment, we compare the accuracy of the alignment generated by Tempo against that of IsoRank, MAGNA++, and GHOST. We recall that we select the original query network from a subset of nodes and their edges from the target network, and then evolve the query through time points. Here, we evaluate the accuracy by calculating the percentage of the aligned nodes from query network that are paired with the same nodes of the target network that they were originally selected from. We refer to this percentage as *recovered region*. We illustrate the results in Figure 3, which demonstrate that Tempo recovers high percentage of the query networks compared to other methods. As the temporal rate increases, the accuracy of Tempo improves dramatically while that of IsoRank remains nearly stagnant and while MAGNA++ and GHOST continue to generate alignments with low recovery rates. Growing the temporal rate while keeping the cold rate unchanged means that the topology of the query network (i.e., cold edges) is evolving slower than the rest of the temporal network (i.e., hot edges). This implies that Tempo can capture the variation in such evolutionary rate while competing alignment strategies which fail to do so.

### 5.2 Evaluation of induced conserved structure

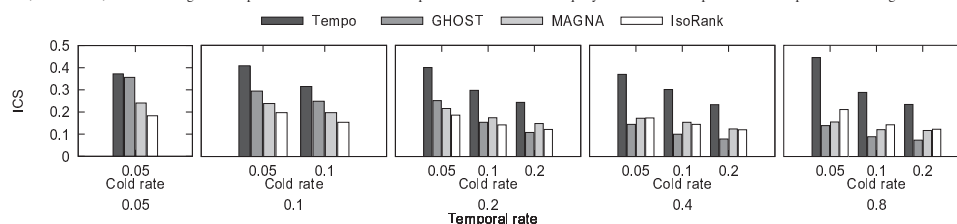
Next, we evaluate the topological quality of the alignment generated by Tempo through comparison with IsoRank, MAGNA++, and GHOST. For this purpose, we measure the shared topological structure between  $G_i^1$  and  $G_i^2$  which is preserved under the alignment function  $\psi$  through all time points  $i$ . Induced conserved structure (ICS) measures the percentage of edges from  $G_i^1$  that are aligned to edges in  $G_i^2$  to the total edges of the induced subnetwork  $\Psi(V^1|G_i^2)$ , and is one of the most common measures of topological quality (Vijayan et al., 2015). Formally,  $ICS(G^1, G^2, \psi) = \sum_{i=1}^t \frac{|E_i^1 \cap E_i^2[\Psi(V^1|G_i^2)]|}{|E_i^2[\Psi(V^1|G_i^2)]|}$ . Figure 4 presents the results, which demonstrate that Tempo generates alignments with high quality based on ICS compared to other algorithms. We note that GHOST was created to optimize ICS, however, Tempo outperforms GHOST on this measure—especially when the temporal rate is high since the performance of GHOST degrades. We also evaluate our algorithm against other algorithms using the edge correctness (EC) measure (see Section A.3).

### 5.3 Evaluation of statistical significance of the alignment

We compare the statistical significance of the alignments generated by Tempo against that of existing methods. In order to ensure that our experiments do not give any advantage to our algorithm, we use IsoRank to



**Fig. 3.** The percentage of recovered query in the resulting alignment varying  $\epsilon$  and  $\epsilon^c$  to take the values {0.05, 0.1, 0.2, 0.4, 0.8} and {0.05, 0.1, 0.2} respectively. The x-axis shows temporal rate,  $\epsilon$  and cold rate,  $\epsilon^c$  (these are the parameters used for constructing synthetic temporal network, with varying evolution rates). The y-axis shows the percentage of recovered query of IsoRank, MAGNA++, and GHOST against Tempo. The error bars show the 80-percentile of the recovered query based on the 10 repetitions of each parameters setting.

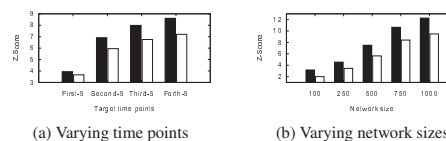


**Fig. 4.** The induced conserved structure (ICS) score of the resulting alignment varying  $\epsilon$  and  $\epsilon^c$  to take the values {0.05, 0.1, 0.2, 0.4, 0.8} and {0.05, 0.1, 0.2} respectively. The x-axis shows temporal rate,  $\epsilon$  and cold rate,  $\epsilon^c$ . The y-axis shows the ICS score of GHOST, MAGNA++, and IsoRank against our method (Tempo).

generate initial alignments for Tempo and thus, compare the statistical significance against IsoRank only.

**Varying evolution rate.** In this experiment, we evaluate the effect of varying the temporal rate ( $\epsilon$ ) and cold rate ( $\epsilon^c$ ) on the significance of the score of the alignments produced by Tempo and that of IsoRank. We generate synthetic networks of sizes {100, 250, 500, 750, 1000} and 20 time points. We fix the network density to two edges per node on the average, and vary  $\epsilon$  and  $\epsilon^c$  ( $\epsilon^c \leq \epsilon$ ) to take the values {0.05, 0.1, 0.2, 0.4, 0.8} and {0.05, 0.1, 0.2}, respectively. Next, we randomly selected 50 nodes from target network 1,000 times, and calculate the alignment score of each, i.e., each random selection corresponds to an alignment. We calculate the mean and standard deviation of these 1,000 scores and generate the z-score of the alignment generated by Tempo using this mean and standard deviation. Hence, we denote the score generated from our method by  $S^*$ , and denote the mean and standard deviation of 1,000 scores generated from the random selections with  $S_\mu$  and  $\sigma$ , respectively. We calculate the z-score of our method as  $(S^* - S_\mu)/\sigma$ . We calculate the z-score of the IsoRank method in a similar manner. Figure 5 presents the average z-score values across all target network sizes. The results show that as we increase the temporal rate, the z-score of Tempo significantly increases while the z-score of IsoRank increases by small amount. As the evolution rate increases, the topology of the alignment found by Tempo differs significantly from the topology of rest of the network, and thus, it becomes more challenging to find the correct alignment. However, Tempo continues to generate accurate and significant results especially for large evolution rates unlike IsoRank which considers each single time point independently. We observe the same pattern as we increase cold rate.

**Varying time points.** In this experiment, we evaluate how the z-scores of Tempo and IsoRank differ as the input networks evolve and deviate from each other. More specifically, we consider aligning the query network with each of the four target sets we have which have evolving time points (i.e. older ages) as we move to later target sets. First, we measure the z-score of aligning the query to the first target set (i.e., containing time points 2, 7, 12, ...) then we measure the z-score of aligning the query to the second target set (i.e., containing time points 3, 8, 13, ...) and so on. We present the average z-score across all temporal and cold rates. Figure 6(a) presents the



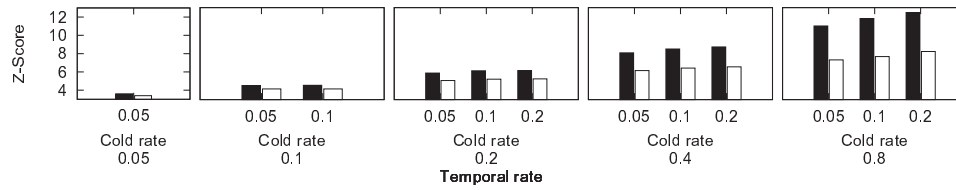
**Fig. 6.** The average z-score of Tempo (black) against IsoRank (white) (a) varying target time points, the x-axis shows time point selected, and (b) varying network size, the x-axis shows network sizes in terms of number of nodes.

results. The results show that Tempo continues to generate alignment with high score significance as we evolve the network. We observe the same pattern for IsoRank, however, Tempo outperforms IsoRank—especially when the time points are distant. This confirms the fact that as the target and query networks evolve and deviate from each other, Tempo is able to take into account the evolution through consecutive time points and generate accurate alignments that persist.

**Varying network size.** In this experiment, we compare the significance of the alignment generated by Tempo against IsoRank as the target network size increases and the query becomes small with respect to the target. We average the z-score across all evolution rates and vary target network size to take values {100, 250, 500, 750, 1000}. Figure 6(b) presents the results, which show that the significance of the alignment (best alignment) increases as we increase the size of the underlying target network. We expect this behavior since we compare the aligned nodes (50 nodes) to a random selection of 50 nodes from the underlying target network. Thus, the chance of selecting the best alignment decreases. That said, Tempo was able to identify the accurate alignment which results in high significant values.

## 5.4 Evaluation on real data

Next, we evaluate Tempo on the real data. We first evaluate the significance of alignment score using Tempo. We calculate the z-score by comparing the score of aligned nodes to the score of 1,000 randomly selected alignments of the same number of nodes. We compare our results to those of IsoRank.



**Fig. 5.** The average z-score of Tempo across network sizes (100, 250, 500, 750, 1000) varying  $\epsilon$  and  $\epsilon^c$  to take the values {0.05, 0.1, 0.2, 0.4, 0.8} and {0.05, 0.1, 0.2} respectively. The x-axis shows temporal rate,  $\epsilon$  and cold rate,  $\epsilon^c$ . The y-axis shows the z-score of IsoRank (white) against Tempo (black).

We repeat this experiment for three different disease network queries: Alzheimer's, Huntington's and Type-II diabetes. Figure 7 shows the results. Our results demonstrate that Tempo yields highly significant alignments, and outperforms IsoRank in terms of z-score. We also observe that z-scores of non-age related disease (diabetes) is lower than those of age-related diseases (i.e. Alzheimer and Huntington's). Although there are some fluctuations in the z-score with growing time gap between query and target networks, we observe that the z-score tends to increase for Alzheimer's and Huntington's disease unlike the Type-II diabetes. This suggests that age-related pathways have higher evolution rate than other pathways. Thus, we conjecture that Tempo, which takes all time points into consideration, is suitable for capturing evolving topologies.

Next, we consider the biological significance of our results by identifying aligned gene pairs in which the aligned genes are different, and determining prior evidence that these gene pairs are biologically relevant. We use Tempo to identify 4, 4 and 6 such pairs for Alzheimer's, Huntington's and Type-II diabetes, respectively. We note that Alzheimer's, Huntington's and Type-II diabetes query sizes are 39, 36, and 23. Thus, the percentages of the different genes found to all the genes in the alignment are 10% to 26%. IsoRank only mapped genes to themselves, suggesting that IsoRank only considers static topologies while our algorithm could map genes based on homological similarities as well as evolving topologies. MAGNA++ and GHOST could only map few genes to themselves while other mapped genes were poorly related (see appendix, Section A.3).

For each combination of disease and differently mapped gene pairs identified by Tempo, we first search PubMed for publication evidence specific to that disease. For instance, in case of Alzheimer's disease, the gene DAB1 that was selected by Tempo and was identified as a potential gene that encode proteins related to functions in biological pathways relevant to the disease (Gao et al., 2015). Genes found by Tempo for type II diabetes, for example gene ACTA1, has remarkable change in gene expression value that was observed for the in diabetic samples compared to non-diabetic samples (Nashida et al., 2013). Moreover, significant up-regulation of GRB2 is observed in transgenic samples compared to controls (Burdon et al., 2015).

In order to determine the biological processes of the aligned genes found by Tempo in gene aging dataset, we perform the gene ontology analysis of the aligned genes in target network using Gene Ontology Consortium (Consortium et al., 2004). We identify the biological processes or signaling pathways that play significant roles in the disorder. We calculate how many related pathways found by our method (Tempo) against MAGNA and GHOST and their significance. We also counted the frequency of those pathways when used different range of time points. Table 1 present the results. We find references of certain pathways that are related to specific neurodegenerative disorders (Alzheimer's and Huntington's diseases). For genes we identify when we use Alzheimer's disease as a query network, we find two pathways, namely *Alzheimer disease-amyloid secretase* and *Alzheimer disease-presenilin* are related to Alzheimer's disease (Mattson, 2004). Various growth factors alter the brain development process at younger age, that manifest as a variety of risk factors at an older

Disease	Tempo	MAGNA++	GHOST
Alzheimer	2 / 4 / 2.29E-14	1 / 2 / 2.14E-03	1 / 2 / 3.32E-04
Huntington's	1 / 4 / 1.15E-22	0	0
Diabetes	2 / 4 / 2.29E-09	1 / 1 / 2.2E-01	0

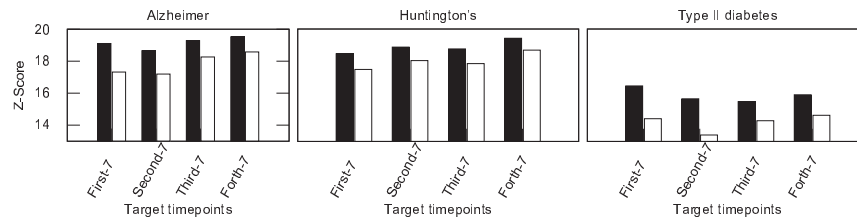
**Table 1.** Number and significance of functional pathways associated with the underlying disease observed among the aligned genes of target network. Each cell lists the results in the form  $x/y/z$ . Here,  $x$  represents number of pathways identified,  $y$  denotes the number of time points at which these pathways are observed, and  $z$  is the statistical significance (p-value) of the least significant of these pathways. The cell with the value 0 implies that no pathways were found.

age and eventually results in aging-related diseases such as Alzheimer's and Huntington's diseases (Bartzokis, 2004). For the genes we identify when we use type II diabetes phenotype as a query, we find two pathways that they are commonly associated with type II diabetes (Liu et al., 2007) namely *Insulin/IGF pathway-protein kinase B signaling cascade* and *Insulin/IGF pathway-mitogen activated protein kinase kinase/MAP kinase cascade*. On the other hand, MAGNA or GHOST found at most one pathway with very low significance and did not appear through all tested target networks (see Table 1). In conclusion, studying temporal networks in general and human aging specifically using Tempo enables us to identify age related genes from non age related genes successfully. More importantly, Tempo takes the network alignment problem one huge step forward by moving beyond the classical static network models.

## References

- Aladağ, A. E. and Erten, C. (2013). SPINAL: scalable protein interaction network alignment. *Bioinformatics*, **29**(7), 917–924.
- Alkan, F. and Erten, C. (2013). BEAMS: backbone extraction and merge strategy for the global many-to-many alignment of multiple PPI networks. *Bioinformatics*, **30**(4), 531–539.
- Barabási, A. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, **286**(5439), 509–512.
- Bartzokis, G. (2004). Age-related myelin breakdown: a developmental model of cognitive decline and Alzheimer's disease. *Neurobiology of Aging*, **25**(1), 5–18.
- Berchtold, N. C., Cribbs, D. H., Coleman, P. D., Rogers, J., Head, E., Kim, R., Beach, T., Miller, C., Troncoso, J., Trojanowski, J. Q., et al. (2008). Gene expression changes in the course of normal brain aging are sexually dimorphic. *Proceedings of the National Academy of Sciences*, **105**(40), 15605–15610.
- Breitkreutz, B., Stark, C., Reguly, T., Boucher, L., Breitkreutz, A., Livstone, M., Oughtred, R., Lackner, D. H., Bähler, J., Wood, V., et al. (2007). The BioGRID interaction database: 2008 update. *Nucleic Acids Research*, **36**(suppl\_1), D637–D640.
- Burdon, K. P., Fogarty, R. D., Shen, W., Abhary, S., Kaidonis, G., Appukuttan, B., Hewitt, A. W., Sharma, S., Daniell, M., Essex, R. W., et al. (2015). Genome-wide association study for sight-threatening diabetic retinopathy reveals association with genetic variation near the GRB2 gene. *Diabetologia*, **58**(10), 2288–2297.
- Charlebois, D. A., Balázs, G., and Kærn, M. (2014). Coherent feedforward transcriptional regulatory motifs enhance drug resistance. *Physical Review E*, **89**(5), 052708.
- Clemente, J., Satou, K., and Valiente, G. (2006). Finding conserved and non-conserved reactions using a metabolic pathway alignment algorithm. *Genome*





**Fig. 7.** The average z-score of our method using real data of three different diseases; Alzheimer's, Huntington's and Type-II diabetes. The x-axis shows which time points were selected to represent the target network. The y-axis shows the z-score of IsoRank (white bars) against our method (black bars).

*Informatics*, **17**(2), 46–56.

Consortium, G. O. *et al.* (2004). The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Research*, **32**(suppl 1), D258–D261.

Cook, S. A. (1971). The complexity of theorem-proving procedures. In *Proceedings of ACM Symposium on Theory of Computing*, pages 151–158.

De Smith, A., Walters, R., Froguel, P., and Blakemore, A. (2008). Human genes involved in copy number variation: mechanisms of origin, functional effects and implications for disease. *Cytogenetic and Genome Research*, **123**(1-4), 17–26.

Feige, U. (1998). A threshold of  $\ln n$  for approximating set cover. *Journal of the ACM (JACM)*, **45**(4), 634–652.

Flannick, J., Novak, A., Srinivasan, B. S., McAdams, H. H., and Batzoglou, S. (2006). Graemlin: general and robust alignment of multiple large interaction networks. *Genome Research*, **16**(9), 1169–1181.

Freyre-González, J. A., Alonso-Pavón, J. A., Treviño-Quintanilla, L. G., and Collado-Vides, J. (2008). Functional architecture of *Escherichia coli*: new insights provided by a natural decomposition approach. *Genome Biology*, **9**(10), R154.

Gao, H., Tao, Y., He, Q., Song, F., and Saffen, D. (2015). Functional enrichment analysis of three Alzheimer's disease genome-wide association studies identifies DAB1 as a novel candidate liability/protective gene. *Biochemical and Biophysical Research Communications*, **463**(4), 490–495.

Gülsöy, G., Gandhi, B., and Kahveci, T. (2012). TOPAC: alignment of gene regulatory networks using topology-aware coloring. *Journal of Bioinformatics and Computational Biology (JBCB)*, **10**(01), 1240001.

Hasan, M. and Kahveci, T. (2013). Color distribution can accelerate network alignment. In *International Conference On Bioinformatics and Computational Biology*, page 52. ACM-BCB.

Hasan, M. and Kahveci, T. (2014). Incremental Network Querying in Biological Networks. In *Proceedings of the ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, BCB, pages 752–759.

Hasan, M. and Kahveci, T. (2015). Indexing a protein-protein interaction network expedites network alignment. *BMC Bioinformatics*, **16**(1), 326.

Holme, P. and Saramäki, J. (2012). Temporal networks. *Physics Reports*, **519**(3), 97–125.

Hu, J., Kehr, B., and Reinert, K. (2013). NetCofree: a fast and accurate global alignment approach to identify functionally conserved proteins in multiple networks. *Bioinformatics*, **30**(4), 540–548.

Hulovatyy, Y., Chen, H., and Milenković, T. (2015). Exploring the structure and function of temporal networks with dynamic graphlets. *Bioinformatics*, **31**(12), i171–i180.

Ibragimov, R., Malek, M., Baumbach, J., and Guo, J. (2014). Multiple graph edit distance: simultaneous topological alignment of multiple protein-protein interaction networks with an evolutionary algorithm. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, pages 277–284. ACM.

Johnson, N., Kotz, S., and Balakrishnan, N. (1994). *Continuous Univariate Probability Distributions*, (Vol. 1).

Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer.

Kelley, B. P., Yuan, B., Lewitter, F., Sharan, R., Stockwell, B. R., and Ideker, T. (2004). PathBLAST: a tool for alignment of protein interaction networks. *Nucleic Acids Research*, **32**(suppl\_2), W83–W88.

Kuchaiev, O. and Pržulj, N. (2011). Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, **27**(10), 1390–1396.

Kuchaiev, O., Milenković, T., Memišević, V., Hayes, W., and Pržulj, N. (2010). Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface*.

Leiserson, M., Vandin, F., Wu, H., Dobson, J. R., Eldridge, J. V., Thomas, J. L., Papoutsaki, A., Kim, Y., Niu, B., McLellan, M., *et al.* (2015). Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nature Genetics*, **47**(2), 106–114.

Liao, C., Lu, K., Baym, M., Singh, R., and Berger, B. (2009). IsoRankN: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, **25**(12), i253–i258.

Liu, M., Liberzon, A., Kong, S., Lai, W. R., Park, P. J., Kohane, I. S., and Kasif, S. (2007). Network-based analysis of affected biological processes in type 2 diabetes models. *PLoS Genetics*, **3**(6), e96.

Mattson, M. P. (2004). Pathways towards and away from Alzheimer's disease. *Nature*, **430**(7000), 631–639.

Milenković, T., Ng, W., Hayes, W., and Pržulj, N. (2010). Optimal network alignment with graphlet degree vectors. *Cancer Informatics*, **9**, 121.

Nashida, T., Yoshie, S., Haga-Tsujimura, M., Imai, A., and Shimomura, H. (2013). Atrophy of myoepithelial cells in parotid glands of diabetic mice; detection using skeletal muscle actin, a novel marker. *FEBS Open Bio*, **3**(1), 130–134.

Neyshabur, B., Khadem, A., Hashemifar, S., and Arab, S. (2013). NETAL: a new graph-based method for global alignment of protein-protein interaction networks. *Bioinformatics*, **29**(13), 1654–1662.

Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., and Kanehisa, M. (1999). KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, **27**(1), 29–34.

Papadimitriou, C. H. and Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity*.

Patro, R. and Kingsford, C. (2012). Global network alignment using multiscale spectral signatures. *Bioinformatics*, **28**(23), 3105–3114.

Phan, H. T. and Sternberg, M. J. (2012). PINALOG: a novel approach to align protein interaction networks—implications for complex detection and function prediction. *Bioinformatics*, **28**(9), 1239–1245.

Przytycka, T. M., Singh, M., and Slonim, D. K. (2010). Toward the dynamic interactome: it's about time. *Briefings in Bioinformatics*, page bbp057.

Sadikovic, B., Al-Romaih, K., Squire, J., and Zielenska, M. (2008). Cause and consequences of genetic and epigenetic alterations in human cancer. *Current Genomics*, **9**(6), 394–408.

Sahraeian, S. and Yoon, B. (2013). SMETANA: accurate and scalable algorithm for probabilistic alignment of large-scale biological networks. *PLoS One*, **8**(7), e67995.

Saraph, V. and Milenković, T. (2014). MAGNA: maximizing accuracy in global network alignment. *Bioinformatics*, **30**(20), 2931–2940.

Shih, Y. and Parthasarathy, S. (2012). Scalable global alignment for multiple biological networks. *BMC Bioinformatics*, **13**(3), S11.

Singh, R., Xu, J., and Berger, B. (2007). Pairwise global alignment of protein interaction networks by matching neighborhood topology. In *International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 16–31.

Sun, Y., Crawford, J., Tang, J., and Milenković, T. (2015). Simultaneous optimization of both node and edge conservation in network alignment via WAVE. In *International Workshop on Algorithms in Bioinformatics*, pages 16–39. Springer.

Vijayan, V., Saraph, V., and Milenković, T. (2015). MAGNA++: Maximizing Accuracy in Global Network Alignment via both node and edge conservation. *Bioinformatics*, **31**(14), 2409–2411.

Vijayan, V., Critchlow, D., and Milenkovic, T. (2017). Alignment of dynamic networks. *International Conference on Intelligent Systems for Molecular Biology (ISMB)*.

Zhang, Z., Schwartz, S., Wagner, L., and Miller, W. (2000). A greedy algorithm for aligning DNA sequences. *Journal of Computational Biology*, **7**(1-2), 203–214.

Zhu, X., Gerstein, M., and Snyder, M. (2007). Getting connected: analysis and principles of biological networks. *Genes & Development*, **21**(9), 1010–1024.