PLATO: Predicting Longitudinally-Aligned Time Observations of biological networks

Tamim Khatib
University of Florida
Gainesville, Florida, USA
tkhatib@ufl.edu

Shahaddin Gafarov
University of Florida
Gainesville, Florida, USA
shahaddingafarov@ufl.edu

Justin D. Strikowski
University of Florida
Gainesville, Florida, USA
jstrikowski@ufl.edu

Mehmet Turan
Bogazici University
Istanbul, Turkiye
mehmet.turan@bogazici.edu.tr

Tamer Kahveci
University of Florida
Gainesville, Florida, USA
tamer@cise.ufl.edu

ABSTRACT

Biological networks are dynamic structures, continuously evolving 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, limiting the number of possible observed gene networks. In this paper, we consider the problem of predicting entire gene networks of dynamic biological networks. We develop a novel algorithm PLATO (Predicting Longitudinally-Aligned Time Observations), which utilizes dynamic network alignment that maps multiple systems of networks to improve the prediction accuracy of the matrix factorization model. We evaluate our method on gene-gene interaction networks using a mouse model with evolutionary patterns caused by chronic myeloid leukemia (CML) and compare it to four existing state of the art methods, including two deep learning and one matrix factorization techniques. Our experimental results demonstrate that PLATO outperforms both traditional matrix factorization and other competing methods in terms of gene-gene interaction prediction accuracy.

CCS CONCEPTS

• **Applied computing** → *Biological networks*; *Bioinformatics*; • **Computing methodologies** → *Non-negative matrix factorization.*

ACM Reference Format:
Tamim Khatib, Shahaddin Gafarov, Justin D. Strikowski, Mehmet Turan, and Tamer Kahveci. 2025. PLATO: Predicting Longitudinally-Aligned Time Observations of biological networks. In *Proceedings of the 16th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB '25), October 11–15, 2025, Philadelphia, PA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3765612.3767242

This work is partially supported by NSF under award number 2111679. Mehmet Turan was supported by the Turkish Directorate of Strategy and Budget, Turkey under TAM project number 2007K12-873.

BCB '25, October 11–15, 2025, Philadelphia, PA, USA

ACM ISBN 979-8-4007-2200-4/2025/10
https://doi.org/10.1145/3765612.3767242

1 INTRODUCTION

Biological networks are inherently dynamic, as the topology of the interactions among molecules continuously change over time. Many factors contribute to this temporal variability, including genetic and epigenetic mutations [11], cellular responses to external stimuli [16], and alterations in DNA replication timing [2, 14]. Consequently, the gene regulatory network within each cell exhibits substantial structural plasticity. In humans, a single somatic cell accumulates approximately 1,700 mutations over a 25-year period [6]. Given that the human body has roughly 37 trillion cells, this equates to an estimated 6.7 trillion mutations occurring daily.

This dynamic nature makes it unrealistic to denote the interaction network topology with a single static graph model. Capturing the interaction patterns of cells across multiple time points is thus sorely needed for accurate modeling. The most direct approach is to measure gene activity levels in wet-lab experiments at each time point. However, this is impractical at large scale due to economic and logistical constraints [19]. Beyond cost, these procedures demand substantial human labor and time [20]. Additionally, tissue extraction is often invasive, and patient availability for repeated sampling is limited [9]. Furthermore, ethical debates persist regarding the long-term ownership of biological samples [1].

There are methods to extract insights from a minimal number of wet lab samples. The prevailing approach involves recording a cell's gene network at regular time intervals. However, this strategy often fails to capture gene network states at time points between samples. Another limitation arises from inter-patient variability in response to stimuli. Two individuals may receive the same treatment for the same condition within the same time frame, yet exhibit remarkably different gene network alterations [21].

Existing studies explored data imputation techniques to infer missing interactions among molecules. A range of drug response prediction models have emerged, including similarity-regularized matrix factorization [15] and 3D-fiber-based tensor models [3]. In parallel, numerous edge/link prediction methods have been developed, using Bayesian networks [22] and graph convolutional networks (GCNs) [8]. These models rely on temporal information derived from the same cell across multiple time points and cannot use gene network data from entirely different organisms to enhance their model accuracy. New generative models that holistically study populations of evolving networks are needed to fill the gaps in at time points where wet-lab data is missing.

Our contributions. In this study, we investigate the problem of inferring missing gene interaction networks along a cell's evolutionary trajectory. We propose a novel algorithm, PLATO (Predicting Longitudinal Aligned Transcriptional Interaction Observations), designed to reconstruct protein-protein interaction (PPI) networks from sparse and temporally misaligned data. PLATO distinguishes itself from prior approaches in two key respects. First, it integrates observed interaction topologies from multiple biological samples collected at arbitrary and irregular time points to construct individualized predictive models. Second, it accounts for heterogeneity in both the evolutionary rates of cellular processes and the timing of wet-lab observations.

PLATO leverages shared structural patterns across a population of gene regulatory networks to infer missing PPIs through alignment of observed network topologies. It represents the networks as a three-dimensional tensor, with the first two dimensions indexing gene pairs and the third indexing temporal observations. We evaluate PLATO on a dataset derived from a mouse model of chronic myeloid leukemia (CML) and benchmark its performance against three baselines, including EvolveGCN[18] and DGCN[8]. Our results demonstrate that PLATO outperforms competing methods in both deep learning and classical machine learning category in terms of prediction accuracy. These results suggest that PLATO has great potential to advance our understanding of evolving biological networks and brings us closer to achieving near-perfect generated biological networks in place of wet-lab experimentation.

The rest of the paper is organized as follows. Section 2 presents related key studies and gaps in the current literature. Section 3 introduces PLATO and the mathematical explanation of the algorithm. Section 4 discusses the datasets, explains the experiment setup, and evaluates the results our algorithm along with the competing methods. We conclude with a brief summary in Section 5.

2 BACKGROUND AND GAPS IN LITERATURE

Predicting the structure of biological networks poses significant challenges due to the continuous evolution of their topologies and incompleteness of the data needed for constructing their topologies throughout their evolution. Reliable prediction methods are needed to infer their topologies at time points when wet-lab observations are missing. We categorize existing approaches into two classes: node prediction and edge (i.e., interaction, link) prediction.

Node prediction methods are capable of predicting missing edges and generating new nodes along associated edges. Many popular node prediction algorithms employ an auto-regressive model for graph generation. For example, GraphRNN [24], based on the RNN structure, is divided into graph-level and edge-level RNNs. These RNN models generate the new nodes and the edges that build the graph in a recurrent manner. GRAN[17] is a type of auto-regressive model that adds the nodes to the graph variable-sized blocks of nodes, instead of sequentially like GraphRNN. This greatly improves the time complexity and scalability of the graph generation process at the cost of sample quality. NNP-GNN [26] introduces a novel approach to predict all potential links for previously unobserved or isolated nodes. Lastly, DMNP[7] is a deep-learning-based node prediction approach that does single-node prediction, named DMNP-S, in multiple steps, where it first connects the missing node with all possible connections. This network gets passed into a 2-layer GCN, and for each connection multilayer perceptron is used to assess the reliability of an edge connection.

While these models outperform traditional link prediction methods in scenarios where new nodes must be inferred, they suffer from a critical limitation: they do not leverage homologous genes from related organisms. Instead, they attempt to generate entirely new nodes, which can degrade prediction accuracy, especially for sparse networks and small amount of data is available.

Edge prediction methods, focus on inferring missing links between already observed nodes. Several prominent models fall into this category. EvolveGCN [18], an extension of the Graph Convolutional Network (GCN) [13], employs a recurrent neural network to evolve GCN parameters over time, enabling link prediction across multiple time points. DGCN [8] improves on this by incorporating a modified Dice similarity metric to more accurately evaluate node similarity. DyGFormer [25], based on a transformer architecture, leverages the interaction history of a node's immediate neighbors and performs particularly well when such history is extensive.

Despite their strengths, edge prediction models do not incorporate information from similar organisms or exploit cross-species homology, which constrains their ability to generalize. Node prediction methods, though more flexible in generating new structures, also ignore cross-organism data and typically operate on static network snapshots without modeling temporal evolution.

3 OUR ALGORITHM

We develop PLATO for constructing the topologies of systems of evolving networks. In the following section, we first present the essential terminologies that are necessary to understand the basics of our algorithm. We then describe our algorithm in greater detail.

3.1 Terminology and graph representation

We model each network as a graph $G = (V, E)$, where each gene corresponds to a node in the set V, and each interaction between a pair of genes corresponds to an edge in the set E. For a dynamic system of graphs observed at t time points, where the ith time point precedes the $(i+1)$th time point and $0 < i \leq t$, we represent the graph at time i as $G_i = (V_i, E_i)$. Without loss of generality, we assume the set of nodes is unchanged over time, i.e., $V = \bigcup_i V_i$, which allows us to simplify the representation to $G_i = (V, E_i)$.

We define a system of graphs as a collection of graphs, where each graph captures the network structure at a specific time point. Changes in edges between consecutive graphs reflect the network's dynamic nature. We represent a system of t time points as $S = [G_1, G_2, \ldots, G_t]$. Suppose we observe k such evolving systems of networks, where the jth system is recorded at t_j time points and $0 < j \leq k$. We represent these k systems as

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

where each graph $G_{j,i} = (V, E_{j,i})$ shares the same node set V across all systems.

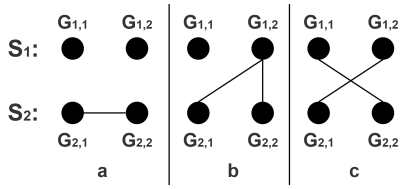

Figure 1: Illustration of the violations of the three alignment conditions. Each black circle represents a network instance of a specific system and time point. Time progresses from left to right. The line connecting two networks indicates an alignment. (a) Aligned graphs belong to the same evolving network. (b) A graph in the second sequence is aligned to more than one graph in the first sequence. (c) Two alignments between two networks cross.

We define network alignment within a system of graphs as a symmetric and transitive mapping of time points between two graph sequences. Given two sequences S_{j_1} and S_{j_2}, we define the alignment using a partial bijective function $\psi_{j_1,j_2}() : \{1, 2, \ldots, t_{j_1}\} \rightarrow \{1, 2, \ldots, t_{j_2}\} \cup \{\perp\}$, where $\perp$ denotes a null mapping. A null mapping indicates that a time point in S_{j_1} has no corresponding match in S_{j_2}, or vice versa. The symmetry condition ensures that $\forall j_1 \neq j_2$ and $\forall i, 1 \leq i \leq t_{j_1}$, we have $i = \psi_{j_2,j_1}(\psi_{j_1,j_2}(i))$. Transitivity requires that $\forall j_1, j_2, j_3$ with $j_1 \neq j_2$, $j_1 \neq j_3$, $j_2 \neq j_3$ and $\forall i, 1 \leq i \leq t_{j_1}$, we have $\psi_{j_1,j_3}(i) = \psi_{j_2,j_3}(\psi_{j_1,j_2}(i))$.

Two graphs from different evolving systems can be aligned if:

(1) The graphs belong to different sequences, i.e., $\forall \psi_{j_1,j_2}(), j_1 \neq j_2$.
(2) No graph may align with more than one graph from the same sequence. Formally, for all $i \neq i'$, $\psi_{j_1,j_2}(i) \neq \psi_{j_1,j_2}(i')$, except in the case where $\psi_{j_1,j_2}(i) = \perp$.
(3) The alignment must preserve temporal order. That is, for all $i < i'$, we disallow alignments where $\psi_{j_1,j_2}(i) > \psi_{j_1,j_2}(i')$.

Figure 1 illustrates violations of these conditions.

3.2 PLATO algorithm

Initial tensor formulation. We model each network $G_{j,i} = (V, E_{j,i})$ as a two-dimensional adjacency matrix with genes along both the x- and y-axes, as shown by Figure 2(a). By including all time points for a given system, we obtain a three-dimensional tensor, where the z-axis represents time. We then construct a unified tensor by stacking graphs across systems at each time point (i.e., $G_{1,i}, G_{2,i}, \ldots, G_{k,i}$) so that the y-axis expands to include genes repeated across k systems, as illustrated by 2(b).

Assume there are γ dynamic systems. Let us denote the number of genes with I, the number of genes across all systems with $J = \gamma I$, and the number of time points in the aligned system with K. This yields a three-dimensional tensor $\boldsymbol{X} \in \mathbb{R}^{I \times J \times K}$, We represent an interaction between gene i and gene j at time k in the sth system as $x_{i,(s-1)I+j,k}$. Figure 2(c) shows a three-dimensional tensor.

Tensor completion aims to predict missing entries in $\boldsymbol{X}$ by learning from known values. We apply rank decomposition via matrix factorization, which approximates the tensor using three low-rank factor matrices. Given rank R (a small positive integer denoting the number of latent features), we introduce three matrices:

- $\boldsymbol{A} \in \mathbb{R}^{I \times R}$: encodes latent features for all unique gene,
- $\boldsymbol{B} \in \mathbb{R}^{J \times R}$: encodes latent features for all genes in all γ systems,
- $\boldsymbol{C} \in \mathbb{R}^{K \times R}$: encodes latent features for temporal dynamics.

For each factor matrix, let the latent vector for index i in $\boldsymbol{A}$ be $\boldsymbol{a}_i$, index j in $\boldsymbol{B}$ be $\boldsymbol{b}_j$, and index k in $\boldsymbol{C}$ be $\boldsymbol{c}_k$. Figure 2(d) illustrates the tensor model of a collection of evolving network systems. Let Ω denote the set of observed entries in $\boldsymbol{X}$. The objective function for rank-R tensor completion minimizes the squared error over Ω:

$$\underset{\boldsymbol{a}_i, \boldsymbol{b}_j, \boldsymbol{c}_k}{\text{minimize}} \sum_{(i,j,k) \in \Omega} \left\| x_{ijk} - \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} \right\|^2. \tag{2}$$

The model predicts the value of each observed tensor cell x_{ijk} as the sum of the Hadamart product of the corresponding latent dimensions across all ranks r. It then compares this predicted value with the observed value and computes the squared error.

To minimize loss, we update the factor matrices using stochastic gradient descent. For an observed entry (i, j, k), the gradients are:

$$\begin{aligned} \nabla \boldsymbol{a}_i &= -2 \left(x_{ijk} - \sum_r a_{ir} b_{jr} c_{kr} \right) (\boldsymbol{b}_j \cdot \boldsymbol{c}_k), \\ \nabla \boldsymbol{b}_j &= -2 \left(x_{ijk} - \sum_r a_{ir} b_{jr} c_{kr} \right) (\boldsymbol{a}_i \cdot \boldsymbol{c}_k), \\ \nabla \boldsymbol{c}_k &= -2 \left(x_{ijk} - \sum_r a_{ir} b_{jr} c_{kr} \right) (\boldsymbol{a}_i \cdot \boldsymbol{b}_j). \end{aligned} \tag{3}$$

We iterate over all entries in Ω during each training epoch. After computing the gradients, we update each factor matrix. This repeats over multiple epochs to gradually minimize the reconstruction error and improve predictions of missing interactions.

Incorporating evolution of interactions into formulation. We conjecture that network alignment improves prediction of network topology due to the alignment bringing similar topological patterns together, assisting in building more accurate latent features in matrix decomposition. In Figure 2(c), consider the first unknown network topology in Network 2. The alignment maps this topology with Network 1's second observed time point and Network 3's first observed time point. Network 2 is expected to share substantial similar interactions with the two observed topologies of Networks 1 and 3. We refer to this strategy as *vertical alignment* and integrate it into our tensor completion framework.

We design a novel regularization strategy to integrate network alignment information into the tensor completion objective. Specifically, given a set of evolving networks, we use the DANTE algorithm [12] to compute optimal alignments across all network systems and time points. It is worth noting that our algorithm does not depend on the particular alignment algorithm; this step can be replaced by any algorithm which align a collection of evolving networks. The resulting alignment guides the factorization process by enforcing consistency across structurally aligned networks.

Let λ denote a regularization coefficient and f be the alignment-based regularization term. We revise the standard rank-factorization-based tensor completion objective as follows:

$$\underset{\boldsymbol{a}_i, \boldsymbol{b}_j, \boldsymbol{c}_k}{\text{minimize}} \sum_{(i,j,k) \in \Omega} \left\| x_{ijk} - \sum_r \boldsymbol{a_{ir}} \boldsymbol{b_{jr}} \boldsymbol{c_{kr}} \right\|^2 + \lambda f. \tag{4}$$

To define f, we observe that in vertical alignment, the genes i and time points k remain fixed, while only the system index

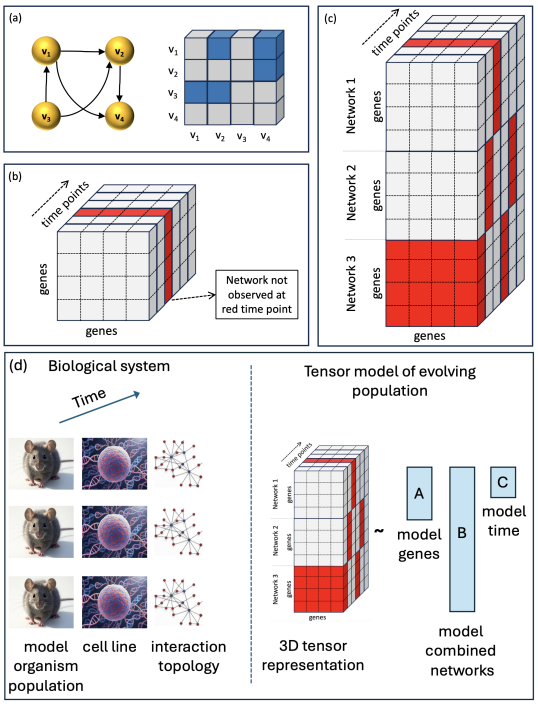

Figure 2: (a) Adjacency matrix representation of network with four nodes; v_1, v_2, v_3, and v_4. Blue highlighted entries indicate existing interactions. (b) 3-D adjacency tensor representation of an evolving biological network. The red highlighted slice (matrix) within the tensor illustrates a time point at which the network topology is not observed and needs to be generated. (c) 3-D combined adjacency tensor representation of three evolving biological networks, each observed at some time points and to be generated at red highlighted time points. (d) Modeling of evolving population as matrices

j changes. Let j^* be a system index such that cell x_{ij^*k} is vertically aligned with x_{ijk}. The regularization factor penalizes the discrepancy between the predicted interaction at (i, j, k) and the corresponding observed value at (i, j^*, k). Let the set of systems aligned with system j at time k be $\mathcal{A}(j, k)$ and the small nonnegative constant to model the contribution of the vertical alignment to the prediction be α (e.g., $\alpha = 0.2$) as a soft penalty to non-integral deviations. We write the alignment penalty as:

$$f = \sum_{(i,j,k)\in\Omega'} \sum_{j^*\in\mathcal{A}(j,k)} \left[\left\lfloor x_{ij^*k} \right\rfloor - \left\lfloor \sum_r \boldsymbol{a_{ir} b_{jr} c_{kr}} \right\rfloor + \alpha \left(x_{ij^*k} - \sum_r \boldsymbol{a_{ir} b_{jr} c_{kr}} \right) \right]^2, \tag{5}$$

Model training time complexity of PLATO. We update the gradients of the rank matrices per-cell. This update occurs along the column of each rank matrix of size R. Therefore, updating each gradient has a time-complexity of $O(R)$. Additionally, because the regularization factor encompasses all systems γ aligned with the current cell's system, it has a time complexity of $O(\gamma)$. We repeat this process for the entire three-dimensional tensor, which is of size $I \times J \times K$, bringing the time complexity of updating each gradient, and therefore the model training, to $O(IJK(R + \gamma))$.

Prediction time complexity of PLATO. We calculate the predicted value for a cell in constant time. Comparing the predicted cell value with the observed cell value is also done in constant time. Each per-cell process must be repeated for the entire three-dimensional tensor, which is of size $I \times J \times K$. Therefore, prediction time complexity is $O(IJK)$.

	Control	CML	Tet-Off-On	TKI	All
min	94882	73120	133917	92315	73120
max	107838	181435	163733	158203	181435
μ	101887	125204	146939	123838	123323
σ	5706	38400	15262	26098	28615
ρ	0.00414	0.00397	0.00377	0.00426	0.00407

Table 1: Aggregate statistics on number of interactions (minimum, maximum, average (μ), standard deviation (σ), and matrix density (ρ) in five datasets used in our experiments.

4 EXPERIMENTAL EVALUATION

4.1 Experiment Setup

Transcription data. We use transcription data from mice found in the Gene Expression Omnibus (GEO) Series GSE244990 to model an evolving sequence of networks [5]. This dataset contains transcription values for 20 mice, all of which have chronic myeloid leukemia (CML), which are divided into five categories:

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

The dataset contains transcription values for each mice measured once each week for up to 19 weeks, or until the mice perish. The most amount of data per mouse is 19 weeks, while the least amount of data is 8 weeks. Some mice have less than 19 weeks due to premature death, while others were only sampled every few weeks. On average, each mice has about 15 weeks of data. Additionally, there are a total of 1432 genes whose interactions with one another are sampled at each mouse/week combination.

Network Data. We obtain PPI network data from the STRING database. STRING provides a confidence value in the [0,1] interval for each interaction, with larger values indicating stronger evidences [23]. The first criteria used to filter interactions is minimum confidence level. The second criteria is minimum transcription level of the two interacting genes. The transcription value measures how gene activity to determine if the gene has sufficient products to carry out the interactions. We use a minimum confidence level cutoff of 0.9 and a minimum transcription value of 50 in our experiments. The result is a total 1432 genes per mouse/week combination that satisfy the above parameters. Table 1 summarizes the interaction data for each group of mice.

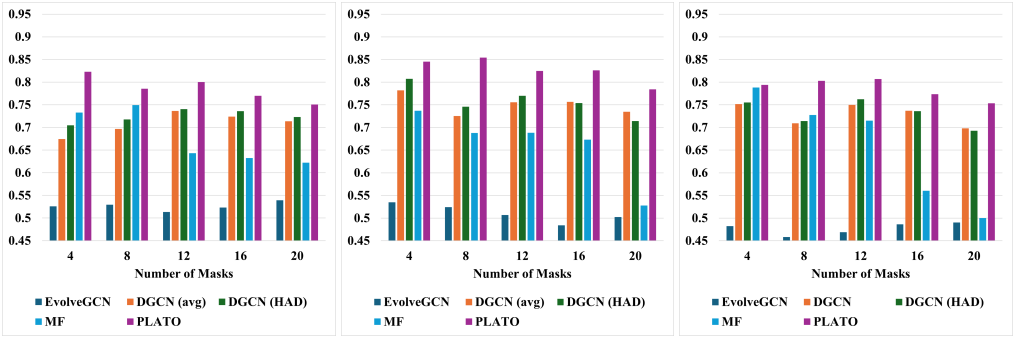

Figure 3: Balanced accuracy of PLATO and the other models for increasing number genes and varying mask sizes. (a) 100 genes (b) 200 genes. (c) 400 genes.

Competing Methods. We compare our algorithm to four competing methods; among them, three are deep learning methods. The first is EvolveGCN, which uses the data of previous networks to predict a missing PPI network [18]. The second and third are variations of DGCN, namely DGCN Average (DGCN avg) and DGCN Hadamard (DGCN HAD) [8]. DGCN is a GCN-based graph representation learning method, with DGCN avg and DGCN HAD variations affecting the classification of the nodes within the graph. The fourth and final competing method is the state-of-the-art matrix factorization (MF) method [4, 10].

Overview of our experiments. We design two experiments to evaluate how accurately each method predicts the topology of PPI networks at missing time points. In each experiment, we mask k randomly selected PPIs (remove the networks of k mice–time point pairs). The remaining unmasked networks are training data. After training, each method predicts the topologies of the masked networks. We then compare the predicted network topology with the actual one to measure the prediction performance. We refer to the number of PPIs masked (k) as the *mask size*.

PPI networks are sparse, so we assess prediction performance using *balanced accuracy* to mitigate class imbalance bias. Suppose we have a predicted network G_{j_p,i_p} and an observed network G_{j_o,i_o} with edge sets E_{j_p,i_p} and E_{j_o,i_o}. We represent the number of edges shared by both networks, true positive (TP), as $|E_{j_p,i_p} \cap E_{j_o,i_o}|$. The number of edges missing from both networks, true negative (TN), is $\binom{|V|}{2} - |E_{j_p,i_p} \cap E_{j_o,i_o}|$. The number of edges listed only in graph G_{j_p,i_p}, false positive (FP), is $|E_{j_p,i_p} - E_{j_o,i_o}|$. The number of edges listed only in graph G_{j_o,i_o}, false negative (FN), is $|E_{j_o,i_o} - E_{j_p,i_p}|$.

Let the balanced accuracy score between the predicted and observed networks be represented as the expression $\varphi(G_{j_p,i_p}, G_{j_o,i_o})$. The formula for calculating balanced accuracy (ρ) is as follows:

$$\varphi(G_{1,i_1}, G_{2,i_2}) = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right). \tag{6}$$

We test with 4, 8, 12, 16, and 20 masked mice–time point pairs and carry out two experiments to evaluate how each method reacts to different dataset characteristics. Each experiment is run three times, with the median balanced accuracy score taken. **Experiment 1** evaluates the impact of increasing the number of genes and uses the *Tet-Off-On* dataset. **Experiment 2** explores the effect of increasing heterogeneity and runs on all datasets except *Control*, which includes too few mice for proper evaluation.

4.2 Effect of gene count on prediction accuracy

In this experiment, we assess how the number of genes impacts the prediction accuracy of each method. We conduct tests using the *Tet-Off-On* dataset and evaluate performance under varying mask sizes (4, 8, 12, 16, and 20) across three gene counts: 100, 200, and 400. Figure 3 presents the balanced accuracy scores of PLATO and four competing methods under these settings.

We observe that **PLATO consistently outperforms all competing methods**, maintaining robust accuracy even as the number of masks increases. MF performs competitively when the mask count is low, but its performance degrades sharply with decreasing density. Both DGCN variants exhibit moderate resilience to mask count, with DGCN HAD outperforming its counterpart across all conditions. Finally, EvolveGCN consistently underperforms.

There is an important trend with increasing gene count: **higher gene counts reduce performance variance across mask sizes**, particularly for PLATO. At 100 genes, limited information causes all methods except PLATO to struggle. At 200 genes, both DGCN variants briefly approach PLATO's performance with few masks, but their accuracy degrades with additional masks. PLATO, in contrast, sustains high accuracy up to 20 masks. At 400 genes, PLATO remains stable, while MF shows a marked drop beyond 16 masks despite strong early performance.

4.3 Effect of system heterogeneity on prediction accuracy

In this experiment, we evaluate how the homogeneity of evolving systems affects PLATO's predictive accuracy. In order to eliminate the bias due to variance in the number of mice across different mice groups, for all tests, we use 200 genes and 4 mice per group. To achieve this, for mice groups with more than 4 mice, we randomly select 4 mice in that group. Figure 4 plots the results.

We measure homogeneity based on the number of distinct time points produced by the DANTE alignment algorithm. Systems with fewer aligned time points are considered to be more homogeneous, as DANTE interpolates and extrapolates less frequently. Based

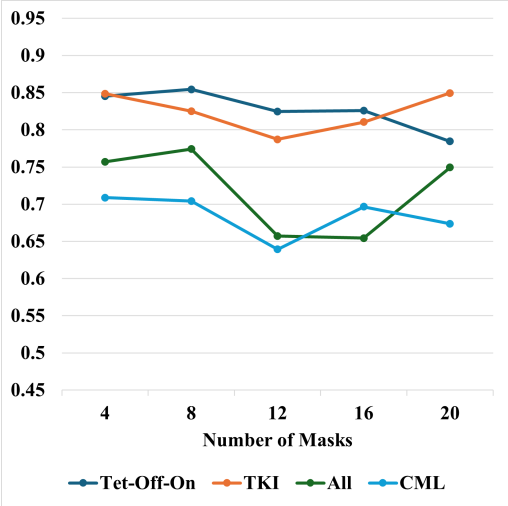

Figure 4: A line graph of the balanced accuracy scores for increasing the heterogeneity of evolving biological systems (i.e. mice). The mice are selected from each dataset (except for Control, due to a lack of mice)

on this criterion, the groups rank as follows: *Tet-Off-On* (26 time points), *TKI* (30), *All* (48), and *CML* (66), with *Tet-Off-On* being the most homogeneous and *CML* being the most heterogeneous.

We observe that **PLATO performs best on more homogeneous systems**, with accuracy decreasing as system heterogeneity increases. In low-mask scenarios, PLATO achieves strong balanced accuracy across all groups—even for the highly heterogeneous *CML* and *All* datasets. However, as the number of masks increases, the model's performance declines more sharply on heterogeneous datasets, while remaining stable on homogeneous ones. These results suggest that **system homogeneity plays a key role in sustaining high predictive accuracy, particularly in sparse settings**. Nevertheless, PLATO handles heterogeneity reasonably well when the number of missing networks is modest.

5 CONCLUSION

In this paper, we considered the problem of predicting missing gene interaction networks in a cell's evolutionary trajectory, where we have limited number of observations at a sequence of possibly unevenly distributed time points. We developed our novel algorithm PLATO to solve this problem. Unlike existing methods which consider one network at a time, PLATO integrates the observations from a population of cells. It does that via optimal network alignment of multiple sequences of biological networks; and models the alignment as a matrix factorization problem.

REFERENCES

[1] N Fazulunnisa Begum, Karthikeyan Ramalingam, Pratibha Ramani, and N Fazulunnisa Begum III. 2024. Storage, retention, and use of leftover pathology specimens: the underestimated treasures. *Cureus* 16, 1 (2024).

[2] Alexa N. Bracci, Anissa Dallmann, Qiliang Ding, Melissa J. Hubisz, Madison Caballero, and Amnon Koren. 2023. The evolution of the human DNA replication timing program. *Proceedings of the National Academy of Sciences of the United States of America* 120 (2023). Issue 10.

[3] A. Bumin, K. Huang, and T. Kahveci. 2023. PartialFibers: An efficient method for predicting Drug-Drug Interactions. *International Conference on Computational Advances in Bio and Medical Sciences* (2023).

[4] J. Douglas Carroll and Jih-Jie Chang. 1970. Analysis of Individual Differences in Multidimensional Scaling Via an N-way Generalization of "Eckart-Young" Decomposition. *Psychometrika* 35, 3 (1970), 283–319. https://doi.org/10.1007/BF02310791

[5] Frankhouser DE, Rockne RC, Uechi L, and Zhao D. 2023. State-transition Modeling of Blood Transcriptome Predicts Disease Evolution and Treatment Response in Chronic Myeloid Leukemia. *bioRxiv* (Dec 2023).

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

[7] Faezeh Faez, Ali Akhoondian Amiri, Mahdieh Soleymani Baghshah, and Hamid R. Rabiee. 2022. DMNP: A Deep Learning Approach for Missing Node Prediction in Partially Observed Graphs. *2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2022), 76–79. https://api.semanticscholar.org/CorpusID:257721209

[8] Chao Gao, Junyou Zhu, Fan Zhang, Zhen Wang, and Xuelong Li. 2023. A Novel Representation Learning for Dynamic Graphs Based on Graph Convolutional Networks. *IEEE Transactions on Cybernetics* 53, 6 (2023), 3599–3612.

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

[10] R. A. Harshman. 1970. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics* 16 (1970), 1–84.

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

[12] Tamim Khatib, Oscar Diaz de la Rua, Kawthar Moria, and Tamer Kahveci. 2024. DANTE: Determining Adaptation trajectories in biological Networks Through Evolutionary mapping. In *ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. 1–6.

[13] Thomas Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *ArXiv* abs/1609.02907 (2016).

[14] Amnon Koren, Paz Polak, James Nemesh, Jacob J. Michaelson, Jonathan Sebat, Shamil R. Sunyaev, and Steven A. McCarroll. 2012. Differential relationship of DNA replication timing to different forms of human mutation and variation. *American journal of human genetics* 91 (2012), 1033–40. Issue 6.

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

[16] Guanyu Li, Ryan LeFebre, Alia Starman, Patrick Chappell, Andrew Mugler, and Bo Sun. 2022. Temporal signals drive the emergence of multicellular information networks. *Proceedings of the National Academy of Sciences of the United States of America* 119 (2022). Issue 37.

[17] Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, William L. Hamilton, David Duvenaud, Raquel Urtasun, and Richard Zemel. 2019. *Efficient graph generation with graph recurrent attention networks.* Curran Associates Inc., Red Hook, NY, USA.

[18] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, and Charles E. Leisersen. 2019. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. *ArXiv* abs/1902.10191 (2019).

[19] Kathryn A. Phillips, Michael P. Douglas, and Deborah A. Marshall. 2020. Expanding Use of Clinical Genome Sequencing and the Need for More Data on Implementation. *JAMA* 324, 20 (11 2020), 2029–2030.

[20] Rafał Płoski. 2016. Chapter 1 - Next Generation Sequencing—General Information about the Technology, Possibilities, and Limitations. In *Clinical Applications for Next-Generation Sequencing*, Urszula Demkow and Rafał Płoski (Eds.). Academic Press, Boston, 1–18.

[21] Dan M. Roden, Russell A. Wilke, 2 Heyo K. Kroemer, PhD, and C. Michael Stein. 2011. Pharmacogenomics: the genetics of variable drug responses. *Circulation* 123 (2011), 1661–70. Issue 15.

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

[23] Damian Szklarczyk and et al. 2023. The STRING database in 2023: protein-protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic Acids Research* 51 (2023), D638–D646. Issue D1.

[24] Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. 2018. GraphRNN: A Deep Generative Model for Graphs. *CoRR* abs/1802.08773 (2018). arXiv:1802.08773 http://arxiv.org/abs/1802.08773

[25] Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. 2023. Towards Better Dynamic Graph Learning: New Architecture and Unified Library. *ArXiv* abs/2303.13047 (2023).

[26] Damiano Zanardini and Emilio Serrano. 2024. Introducing New Node Prediction in Graph Mining: Predicting All Links from Isolated Nodes with Graph Neural Networks. *Neurocomputing* 609 (2024), 128474.