Discovering Protein Interactions and Repurposing Drugs in SARS-CoV-2 (COVID-19) via Learning on Robust Multipartite Graphs

Xiangyu Li[1], Armand Ovanessians[2] and Hua Wang[1]
[1]*Department of Computer Science*
[2]*Department of Quantitative Biosciences and Engineering*
Colorado School of Mines, Golden, Colorado, USA
Email: {lixiangyu, ovanessians}@mines.edu, huawangcs@gmail.com

2023 IEEE International Conference on Data Mining (ICDM) | 979-8-3503-0788-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDM58522.2023.00038

***Abstract*—The COVID-19 pandemic caused by SARS-CoV-2 has emphasized the importance of studying virus-host protein-protein interactions (PPIs) and drug-target interactions (DTIs) to discover effective antiviral drugs. While several computational algorithms have been developed for this purpose, most of them overlook the interplay pathways during infection along PPIs and DTIs. In this paper, we present a novel multipartite graph learning approach to uncover hidden binding affinities in PPIs and DTIs. Our method leverages a comprehensive biomolecular mechanism network that integrates protein-protein, genetic, and virus-host interactions, enabling us to learn a new graph that accurately captures the underlying connected components. Notably, our method identifies clustering structures directly from the new graph, eliminating the need for post-processing steps. To mitigate the detrimental effects of noisy or outlier data in sparse networks, we propose a robust objective function that incorporates the $\ell_{2,p}$-norm and a constraint based on the pth-order Ky-Fan norm applied to the graph Laplacian matrix. Additionally, we present an efficient optimization method tailored to our framework. Experimental results demonstrate the superiority of our approach over existing state-of-the-art techniques, as it successfully identifies potential repurposable drugs for SARS-CoV-2, offering promising therapeutic options for COVID-19 treatment.**

***Index Terms*—Multipartite Graph Learning, Robust Learning, Biomolecular Mechanism Network.**

I. Introduction

The Coronavirus Disease 2019 (COVID-19) is caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), a member of the coronavirus family. Understanding the mechanisms of SARS-CoV-2 infection and developing effective therapeutic interventions against COVID-19 are of paramount importance. However, experimental approaches for identifying virus-host protein-protein interactions and repurposing drugs can be time-consuming, expensive, and yield unreliable results due to false negatives and positives. Therefore, there is a significant need to develop reliable computational methods that can provide testable hypotheses for preliminary investigations into hidden virus-host protein-protein interactions and drug-target interactions.

Most existing computational methods for predicting protein-protein or drug-target interactions rely on clustering relations between host proteins and viral proteins or drug compounds and targeted genes [37]. Unfortunately, few prediction methods simultaneously cluster protein-protein interactions (PPI) and drug-target interactions (DTI), and most existing computational methods fail to provide systematic investigations on the entire pathways of infection interplay. Moreover, viruses and drugs often indirectly affect proteins, leading to cascading effects in the human interactome, such as the cytokine storm observed in severe inflammatory responses to SARS-CoV-2 infection [3]. Therefore, gaining a more comprehensive understanding of the virus-host-drug mechanism is crucial.

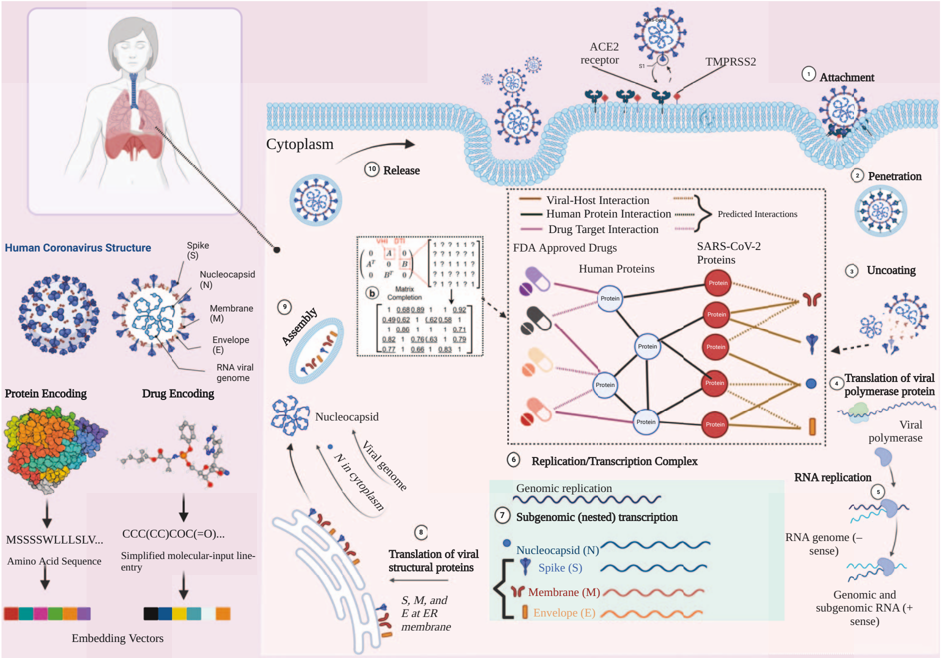
In this paper, we propose an innovative computational approach that integrates protein-protein, drug-target, and virus-host interactions to uncover hidden interactions within the host interactome. Our approach, depicted in Figure 1, provides in-depth knowledge of pathological and pharmacological interactions by considering both direct and indirect effects in the host interactome. Importantly, recent insights into developing computational methods for discovering new protein-protein interactions (PPIs) and drug-target interactions (DTIs) have predominantly focused on deep neural networks, which often require large amounts of training data, resulting in intensive computations and limited interpretability.

To address these challenges, we develop a reliable, efficient, and powerful computational method for modeling and predicting PPIs and DTIs, serving as an auxiliary tool for proteomics or pharmacologic research. We study an integrative dataset that combines molecular mechanisms across protein-protein and drug-target interactions. The dataset comprises three interaction networks: protein-protein interactions (PPIs), genetic interactions (GIs), and virus-host interactions (VHIs), sourced from the BioGRID database [2]. To generate an informative molecular interaction network, we apply the Vanunu smoothening method [34] to integrate these networks. Additionally, we extend the drug-target interaction (DTI) network, originally collected from the DrugBank database [11], to a generalized drug-target interaction network using the Tanimoto similarity coefficient [1], [29]. Further details on the preprocessing and construction of the dataset are provided in Section II.

To uncover hidden interactions within the integrated net-

979-8-3503-0788-7/23/$31.00 ©2023 IEEE

DOI 10.1109/ICDM58522.2023.00038



$$\begin{pmatrix} 0 & A & 0 \\ A^T & 0 & B \\ 0 & B^T & 0 \end{pmatrix} \quad \begin{bmatrix} 1 & ? & ? & 1 & 1 & ? \\ ? & ? & 1 & ? & ? & 1 \\ 1 & ? & 1 & 1 & 1 & ? \\ ? & 1 & ? & ? & 1 & ? \\ ? & 1 & ? & 1 & ? & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & \underline{0.68} & \underline{0.89} & 1 & 1 & \underline{0.92} \\ \underline{0.49} & \underline{0.62} & 1 & \underline{0.62} & \underline{0.58} & 1 \\ 1 & \underline{0.86} & 1 & 1 & 1 & \underline{0.71} \\ \underline{0.82} & 1 & \underline{0.76} & \underline{0.63} & 1 & \underline{0.79} \\ \underline{0.77} & 1 & \underline{0.66} & 1 & \underline{0.83} & 1 \end{bmatrix}$$

Fig. 1. The life cycle of SARS-CoV-2 and our proposed modeling approach for predicting the virus-host-drug interactome. (1) The virus enters the host cell by binding the ACE2 receptor and cleavage by TMPRSS2. (2) The viral RNA genome is released into the host cell cytoplasm. (3) The virus hijacks the host machinery to replicate and translate its RNA genome and structural proteins. (4) The newly synthesized viral proteins and RNA are assembled into a new virion in the Golgi vesicles. (5) The new virion is released from the host cell into the environment to repeat the infection cycle. Our modeling approach consists of two main parts: (a) the encoder framework that converts the drug compound and protein sequence into vector embeddings, and (b) our novel matrix-completion framework that takes the embeddings and constructs a tripartite graph to simultaneously predict putative virus-host protein-protein interactions (VHIs) and drug-target interactions (DTIs). Created with BioRender.com

work, we develop an innovative k-partite graph-based co-clustering method. This method allows us to learn a new graph with exact connected components, presenting an explicit clustering structure and indicators without the need for additional sorting or processing work. We propose a regularized multipartite (k-partite) graph with constraints based on the pth-order Ky-Fan and Schatten norms, effectively determining the rank of the normalized Laplacian matrix. To enhance the robustness of the objective function, we employ a loss function based on the $\ell_{2,p}$-norm, considering the sparsity and potential outliers in the collected dataset. Addressing the optimization challenges posed by the proposed objective function, we develop an efficient and reliable optimization algorithm that integrates the alternating direction method of multipliers (ADMM) with an iteratively reweighted method (IRM). Our method outperforms current state-of-the-art methods in interaction prediction and demonstrates superior robustness to noise.

Through extensive experimental evaluations and literature validation, we identify undiscovered binding affinities for PPIs and DTIs, leading to the discovery of potential drug candidates for treating COVID-19. Importantly, our proposed method is not limited to the studied dataset but is also applicable to any situation that can be modeled as a multipartite graph. By providing accurate predictions and uncovering hidden interactions, our computational approach contributes to the understanding of the virus-host-drug mechanism and holds promise for advancing proteomics and pharmacologic research in the context of COVID-19 and beyond.

II. Materials and Data Sources

For our study, we utilized an integrative dataset comprising protein-protein interactions (PPIs), genetic interactions (GIs), virus-host interactions (VHIs), and drug-target interactions (DTIs) [37]. The PPIs, GIs, and VHIs were sourced from the BioGRID database [2], providing a comprehensive network consisting of $16,431$ human proteins and 332 SARS-CoV-2 proteins. The genetic interaction network encompassed $3,302$ genes, while the metabolic interaction network incorporated $1,530$ genes from the KEGG database [21]. By combining these networks, we constructed a molecular interaction network consisting of $16,872$ genes.

The VHIs affinity matrix, denoted as A_{VH}, exhibited high sparsity due to the low virus-to-host ratio of approximately 0.0197 $(332/16,872)$. To address this sparsity issue, we employed the protein network propagation method proposed by Vanunu et al. [34]. This method iteratively spreads prior information on causal genes to their network neighbors, resulting in the smoothing of protein information across the network. We set the smoothing parameter α to 0.7 and used a convergence threshold of $|A_{VH}^{t} - A_{VH}^{t-1}| < 10^{-5}$ to obtain a smoothed affinity matrix A_{VH}.

To incorporate the DTIs into our analysis, we transformed them into an affinity matrix denoted as A_{DT}. We employed the Tanimoto similarity coefficient [1], [29] to calculate the similarity between the fingerprints of each pair of drugs. The SMILES string of each of the $8,279$ drugs in the DrugBank database was numerically encoded into a vector fingerprint. If

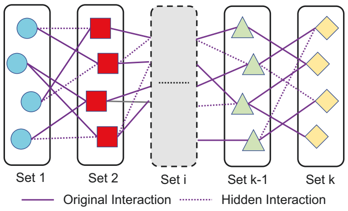


Fig. 2. Illustration of the k partite graph with k disjoint sets. The pairwise co-clustering aims to simultaneously discover all hidden interactions given original interactions.

the similarity score for a pair of drugs exceeded 95% of all other drug pairs, we considered it to have a strong potential to interact with targeted genes, and we added the corresponding connection to the smoothed affinity matrix for DTIs. This process resulted in a DTI matrix containing $1,727,436$ active connections.

By combining the smoothed VHIs affinity matrix A_{VH} and the DTI matrix A_{DT}, we obtained an integrative dataset that encompasses the interactions among proteins, genes, viruses, and drugs. This dataset serves as the foundation for our computational approach, enabling the prediction of virus-host protein-protein interactions (VHIs) and drug-target interactions (DTIs) in a comprehensive manner.

III. Methods

Throughout this paper, matrices are denoted by bold uppercase letters, while vectors are represented by bold lowercase letters. The ℓ_p-norm ($0 < p \leq 2$) of $\mathbf{v}$ is defined as $|\mathbf{v}|_p = (\sum_i v_i^p)^{\frac{1}{p}}$. For a matrix $\mathbf{M} = [m_{ij}]$, the trace of $\mathbf{M}$ is defined as $\text{tr}(\mathbf{M}) = \sum_i m_{ii}$.

A common method for describing multiple distinct independent interactions involves using an undirected weighted k-partite graph. In this representation, data points (vertices) can be partitioned into k disjoint sets, with every pair in the k sets being adjacent, as illustrated in Figure 2. When $k = 2$, these graphs are referred to as bipartite, and when $k = 3$, they are called tripartite. It is important to note that pairwise interactions can propagate through the relations, creating connected influences in a k-partite graph, as shown in Figure 1. Often, the goal is to identify pairwise clustering relations among these data points. However, extracting explicit clustering structures from a k-partite graph can be challenging, especially with large, high-dimensional datasets. Although some graph-based learning methods, such as spectral learning or isoperimetric clustering, have been developed to handle this type of data, their dependencies can introduce additional challenges in revealing the clustering structures within the data. This is because these approaches require the graph to possess specific properties. Furthermore, another notable issue with these methods is the need for additional processing approaches to determine the final clustering results, leading to a significant increase in computational cost as the number of data points grows.

To address these challenges, we propose a novel method that aims to learn a high-level representation matrix from the given k-partite data graph. In this case, the desired pairwise co-clustering results can be directly represented in the learned matrix, eliminating the need for additional clustering approaches on the data graph. Before initiating the learning process, we must provide an end-to-end description of the original interactions within the given data points. A common approach in such situations involves storing the data points in an affinity matrix, where an entry of this matrix denotes the mutual similarity between the data in two disjoint sets represented by the corresponding row and column of the matrix. Given two data points x_i, x_j in set i and j, the similarity distance $a_{i,j}$ can be constructed using the radial basis function (RBF) kernel: $w_{i,j} = \exp\left(-\frac{|x_i - x_j|_2^2}{2\sigma^2}\right)$, where σ is the Gaussian parameter.

From this perspective, we can construct an undirected weighted k-partite graph using a squared affinity matrix A as follows:

$$A = \begin{bmatrix} 0 & A_{1,2} & 0 & \cdots & 0 & 0 \\ A_{1,2}^T & 0 & A_{2,3} & \cdots & 0 & 0 \\ 0 & A_{2,3}^T & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & A_{k-1,k} \\ 0 & 0 & 0 & \cdots & A_{k-1,k}^T & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}, \tag{1}$$

where $A \in \mathbb{R}^{n \times n}$ is the global affinity matrix, $A_{i,j} \in \mathbb{R}^{n_i \times n_j}$ is a local affinity matrix measuring the similarity distance between data points in sets i and j, n_i and n_j represent the number of data points in sets i and j, respectively, and $n = n_1 + n_2 + \cdots + n_k$ denotes the total number of data points.

The global affinity matrix A and associated local matrices $A_{i,j}$ are not block-diagonal matrices, meaning clustering relations cannot be explicitly involved in them. Much of the research into finding underlying clustering relations has focused on the Spectral Graph Partitioning method [10], which aims to find an optimal cutting solution using the edge-cut-based (or normalized cut) criterion. As mentioned earlier, this method and its variants require implementing post-processing approaches to the results. In contrast, our model seeks to learn a new general similarity matrix S from the global affinity matrix A, while the learned matrix S is connected with clustering components. Consequently, our model has a significant advantage in that the optimal clustering structures can be clearly represented in the learned matrix S, providing an efficient and straightforward way to interpret underlying relations within the data.

Our learning process aims not only to obtain clustering relations but also to ensure that the similarity matrix S is as close to the given affinity matrix A as possible. To achieve this, we minimize the reconstruction error between S and A. Additionally, we impose the $\ell_{2,p}$-norm on the distance function to prevent the learning process from being corrupted by noise potentially present in real-world datasets. Moreover, to prevent the learned matrix S from becoming extremely sparse, we also impose a constraint on S where the sum of each row of S is one. Thus, we can formulate our objective function as:

$$\begin{aligned} &\min ||S-A||_{2,p}^{p}, \\ \text{s.t. } &\sum_{j} s_{ij}=1, \quad s_{ij} \geq 0, S \in \Theta, S_{I,I+1} \in \Theta, \end{aligned} \tag{2}$$

where s_{ij} represents the element at position i, j of matrix S, and $S \in \Theta$ is a constraint that restricts the set of matrices in S to be connected with exactly c clustering components. The co-clustering structures in a k-partite graph can be viewed as a connection passing through consecutive bipartite graphs. To avoid ill-clustering structures (the clustering inconsistency between consecutive bipartite graphs), it is beneficial to add a constraint for each sub-bipartite graph such that $S_{I,I+1} = \begin{pmatrix} 0 & S_{I,I+1} \\ S_{I,I+1}^T & 0 \end{pmatrix} \in \Theta$, where $S_{I,I+1}$ is a sub-bipartite graph between set I and $I+1$ $(1 \leq I < k)$.

Since $S \in \Theta$ and $S_{I,I+1} \in \Theta$ are non-linear constraints, optimizing the objective in Eq. (2) can be an extremely challenging task. To simplify the optimization problem, we can replace the non-linear constraints with low-rank constraints on the Laplacian matrices. According to Theorem 1-(5) [10], the number of zero eigenvalues of the Laplacian L_S (i.e., the multiplicity of the 0 eigenvalue) equals the number of connected components of the graph $\mathcal{G}$. Therefore, we can efficiently handle the non-linear constraints by replacing them with $rank(L_S) = n - c$ and $rank(L_{S_{I,I+1}}) = n_I + n_{I+1} - c$ (c is the number of connected components). This leads to the following simplified objective function:

$$\begin{aligned} &\min \|S-A\|_{2,p}^{p}, \quad s.t. \sum_{j} s_{ij}=1, s_{ij} \geq 0, \\ &rank(L_S)=n-c, rank(L_{S_{I,I+1}})=n_I+n_{I+1}-c. \end{aligned} \tag{3}$$

However, optimizing the objective function in Eq. (3) with the low-rank constraints can still be challenging. Therefore, we propose to treat the low-rank constraints as penalty terms added to the cost function. This allows us to transform the constrained minimization problem into an unconstrained one. Specifically, we introduce non-convex penalties involving the p-th order of the Laplacian matrices L_S and $L_{S_{I,I+1}}$. The objective function becomes:

$$\begin{aligned} \min ||S-A||_{2,p}^{p} + \lambda_0 \sum_{i=1}^{c} \sigma_i^p(L_S) + \sum_{I=1}^{k-1} \lambda_I \sum_{i=1}^{c} \sigma_i^p(L_{S_{I,I+1}}), \\ s.t. \sum_{j} s_{ij}=1, s_{ij} \geq 0, \end{aligned} \tag{4}$$

where $\sigma_i(L_S)$ and $\sigma_i(L_{S_{I,I+1}})$ represent the i-th smallest eigenvalues of L_S and $L_{S_{I,I+1}}$, respectively. The parameters λ_0 and λ_I control the strength of the penalties.

To further simplify the objective function, we can express the penalties using the trace operator and Ky-Fan norm. According to the Ky-Fan norm theorem [13], we have:

$$\begin{aligned} &\sum_{i=1}^{c} \sigma_i^p(L_S) = \min \text{Tr}(F_0^T L_S^p F_0), \text{ s.t. } F_0^T F_0 = I, \\ &\sum_{i=1}^{c} \sigma_i^p(L_{S_{I,I+1}}) = \min \text{Tr}(F_I^T L_{S_{I,I+1}}^p F_I), \text{ s.t. } F_I^T F_I = I. \end{aligned}$$

Thus, the objective function is updated as:

$$\begin{aligned} \min ||S-A||_{2,p}^{p} + \lambda_0 Tr(F_0^T L_S^p F_0) + \sum_{I=1}^{k-1} \lambda_I Tr(F_I^T L_{S_{I,I+1}}^p F_I), \\ s.t. \sum_{j} s_{ij}=1, s_{ij} \geq 0, F_0^T F_0 = I, F_I^T F_I = I. \end{aligned} \tag{5}$$

In the next section, we will present the optimization algorithm for solving the objective function in Eq. (5).

A. Our Algorithm

Solving the optimization problem associated with the proposed objective function in Eq. (5) can be challenging due to its non-smooth and non-convex nature. To address this challenge, we propose a generalized version of the Iteratively Reweighted Method (IRM) that allows for optimization of a surrogate loss function. This generalized IRM provides smoothness and convexity advantages, overcoming the limitations of prior studies that focused only on sparsity-inducing penalties.

1) Generalized Iteratively Reweighted Method: First, we introduce a more general problem as below:

$$\min_{x \in \mathbb{C}} f(x) + \sum_{i} \|g_i(x)\|_*^p, \tag{6}$$

where $g_i(x)$ is a scalar, vector, or matrix output function. Thus, we have:

1) If $g_i(x)$ is a scalar output function, (6) changes to:

$$\min_{x \in \mathbb{C}} f(x) + \sum_{i} |g_i(x)|^p. \tag{7}$$

2) If $g_i(x)$ is a vector output function, (6) changes to:

$$\min_{x \in \mathbb{C}} f(x) + \sum_{i} \|g_i(x)\|_2^p. \tag{8}$$

3) If $g_i(x)$ is a matrix output function, (6) changes to:

$$\min_{x \in \mathbb{C}} f(x) + \sum_{i} \|g_i(x)\|_{S_p}^p. \tag{9}$$

It can be seen that the problem in (6) is equivalent to:

$$\min_{x \in \mathbb{C}} f(x) + \sum_{i} \text{Tr}\left[\left(g_i^T(x) g_i(x)\right)^{\frac{p}{2}}\right]. \tag{10}$$

Since the problem in (10) is not smooth and too difficult to solve, we can transform it to an approximation problem that is smooth, which is formulated as:

$$\min_{x \in \mathbb{C}} f(x) + \lambda \sum_{i} \text{Tr}\left[\left(g_i^T(x) g_i(x) + \delta I\right)^{\frac{p}{2}}\right], \tag{11}$$

where δ is the smooth parameter that depends on x. If δ approximates to zero, (11) changes to the problem in (6). We thus propose an iterative algorithm for finding the solution to this problem as described in the below algorithm, and provide a theoretical analysis to prove the convergence of the proposed algorithm.

Algorithm 1 Algorithm to solve the problem in Eq. (6)

```
Initialization: $x \in \mathbb{C}$
while not converge do
    Calculate $D_i = \frac{p}{2}\left(\mathrm{Tr}[g_i^T(x)g_i(x)] + \delta I\right)^{\frac{p-2}{2}}$;
    Update $x$ by solving $\min_{x \in \mathbb{C}} f(x) + \sum_i \mathrm{Tr}\left[g_i^T(x)g_i(x)D_i\right]$;
end while
Output: $x$.
```

2) Optimization analysis: To begin with, we need to calculate the variable derivation by using the following lemma.

Lemma 1. ***Chain rule:*** *Given $g(x)$ is a matrix output function, $h(x)$ is a scalar output function, x is a scalar, vector, or matrix variable, we have:*

$$\begin{aligned}\nabla_x h\left(g(x)\right) &= \mathrm{Tr}\left[\left(\sum_{i,j} \nabla_{g_{ij}(x)} h\left(g(x)\right)\right) \nabla g_{ij}(x)\right] \\ &= \mathrm{Tr}\left[\left(\nabla_{g(x)} h\left(g(x)\right)\right)^T \nabla g(x)\right].\end{aligned}$$

Based on the chain rule in Lemma 1, we can further have:

Lemma 2. *Given $g(x)$ is a scalar, vector or matrix output function, x is a scalar, vector or matrix variable, we can have:*

$$\begin{aligned}&\nabla_x \mathrm{Tr}\left[\left(g^T(x)g(x) + \delta I\right)^{\frac{p}{2}}\right] \\ &= \mathrm{Tr}\left[p\left(g^T(x)g(x) + \delta I\right)^{\frac{p-2}{2}} g^T(x)\nabla g(x)\right].\end{aligned}$$

Proof. Suppose that $h(x) = \mathrm{Tr}\left[x^T x + \delta I\right]^{\frac{p}{2}}$, we have:

$$\nabla_x h(x) = 2\frac{p}{2} x(x^T x + \delta I)^{\frac{p-2}{p}}, \tag{12}$$

which can lead to:

$$\nabla_{g(x)} h\left(g(x)\right) = 2\frac{p}{2} g(x)\left(g^T(x)g(x) + \delta I\right)^{\frac{p-2}{2}}. \tag{13}$$

□

Furthermore, we can have the following lemma.

Lemma 3. *Given $g(x)$ is a scalar, vector or matrix output function, x is a scalar, vector or matrix variable, D is a constant and D is symmetrical if D is a matrix, we can have:*

$$\nabla_x \mathrm{Tr}\left[g^T(x)g(x)D\right] = \mathrm{Tr}\left[2Dg^T(x)\nabla g(x)\right]. \tag{14}$$

Proof. Suppose that $h(x) = \mathrm{Tr}[x^T x D]$, we can thus have $\nabla_x h\left(g(x)\right) = 2g(x)D$. As a result of the chain rule, we can achieve Lemma 3. □

The constrained minimization problem in Eq. (11) can be solved by constructing a Lagrange function, which is given by:

$$\mathcal{L}(x, \lambda) = f(x) + \mu \sum_i \mathrm{Tr}\left[g_i^T(x)g_i(x) + \delta I\right]^{\frac{p}{2}} - r(x, \lambda), \tag{15}$$

where $r(x, \lambda)$ represents a Lagrangian term for the constraint $x \in \mathbb{C}$. Taking the derivative of the Lagrange function with respect to x and setting the derivative to zero, we can have:

$$\begin{aligned}\nabla_x \mathcal{L}(x, \lambda) = &\nabla f(x) + \mu \sum_i \nabla \mathrm{Tr}\left[g_i^T(x)g_i(x) + \delta I\right]^{\frac{p}{2}} \\ &- \nabla r(x, \lambda) = 0.\end{aligned}$$

According to Lemma 2, the problem is equivalent to:

$$\begin{aligned}&\nabla f(x) + \mu \sum_i \mathrm{Tr}\left[p\left(g_i^T(x)g_i(x) + \delta I\right)^{\frac{p-2}{2}} g_i^T(x)\nabla g_i(x)\right] \\ &- \nabla r(x, \lambda) = 0.\end{aligned}$$

According to the Karush-Kuhn-Tucker conditions, solving the problem in Eq. (11) is equivalent to solving the Lagrange problem. However, solving the Lagrange problem directly is challenging. Therefore, we propose an iterative algorithm that allows us to find a stationary point or optimal solution instead. Denote that $D = \frac{p}{2}\left(g^T(x)g(x) + \delta I\right)^{\frac{p-2}{2}}$ is a given constant, we can thus rewrite the problem to:

$$\nabla f(x) + \mu \sum_i \mathrm{Tr}[2D_i g_i^T(x)\nabla g_i(x)] - \nabla r(x, \lambda) = 0. \tag{16}$$

This problem is equivalent to:

$$\min_{x \in \mathbb{C}} f(x) + \mu \sum_i \mathrm{Tr}[g_i^T(x)g_i(x)D_i]. \tag{17}$$

In summary, we first denote the initialization of x and then calculate D_i with respect to x. The iterative algorithm repeats the cycling through x and D_i until reaching convergence.

3) Convergence analysis of the generalized iteratively reweighted algorithm: To begin with, we introduce the following lemma.

Lemma 4. *For any $\sigma > 0$, the following inequality holds when $0 < p \leq 2$:*

$$p(\sigma - 1) \geq 2(\sigma^{\frac{p}{2}} - 1). \tag{18}$$

Proof. We denote $f(\sigma) = p(\sigma - 1) - 2(\sigma^{\frac{p}{2}} - 1)$ and calculate its first-order and second-order derivatives with respect to σ:

$$\nabla f(\sigma) = p - p\sigma^{\frac{p}{2}-1}, \nabla^2 f(\sigma) = p\left(1 - \frac{p}{2}\right)\sigma^{\frac{p}{2}-2}. \tag{19}$$

For $0 < p \leq 2$ and $\sigma > 0$, we have $\nabla^2 f(\sigma) \geq 0$. The condition $\nabla f(\sigma) = 0$ is satisfied when $\sigma = 1$. Moreover, we have $f(\sigma) = 0$ when $\sigma = 1$. Therefore, if $0 < p \leq 2$ and $\sigma > 0$, $f(\sigma)$ is non-negative, which proves Lemma 4. □

Lemma 5. *Given $\tilde{M} = U\Sigma U^T$ and $M = V\Lambda V^T$ as eigendecompositions for positive definite matrices $\tilde{M}$ and M of the same size, where Σ and Λ are diagonal matrices* *with eigenvalues arranged in increasing and decreasing order, respectively, the following inequality holds:*

$$Tr[\tilde{M}M] \geq Tr[\Sigma\Lambda]. \tag{20}$$

Based on Lemma 5, we can derive the following lemma.

Lemma 6. *Given positive definite matrices $\tilde{M}$ and M of the same size, the following inequality holds for $0 < p \leq 2$:*

$$2\left(Tr[\tilde{M}^{\frac{p}{2}}] - Tr[M^{\frac{p}{2}}]\right) \leq p\left(Tr\left[\tilde{M}M^{\frac{p}{2}-1}\right] - Tr\left[MM^{\frac{p}{2}-1}\right]\right) \tag{21}$$

Proof. For $0 < p \leq 2$ and positive values λ and σ, applying Lemma 4 yields:

$$p\left(\sigma\lambda^{\frac{p}{2}-1} - \lambda^{\frac{p}{2}}\right) \geq 2\left(\sigma^{\frac{p}{2}} - 1\right). \tag{22}$$

By taking eigendecompositions of $\tilde{M}$ and M, $\tilde{M} = U\Sigma U^T$ and $M = V\Lambda V^T$, we obtain the following inequalities:

$$\begin{aligned} Tr\left[\Sigma\Lambda^{\frac{p}{2}-1}\right] &\leq Tr\left[\tilde{M}M^{\frac{p}{2}-1}\right] \qquad (23) \\ p\left(Tr\left[\tilde{M}M^{\frac{p}{2}-1}\right] - Tr\left[\Lambda^{\frac{p}{2}}\right]\right) &\leq 2\left(Tr\left[\Sigma^{\frac{p}{2}}\right] - Tr\left[\Lambda^{\frac{p}{2}}\right]\right). \end{aligned}$$

Since $Tr\left[\tilde{M}^{\frac{p}{2}}\right] - Tr\left[\Sigma\right] = 0$ and $Tr\left[M^{\frac{p}{2}}\right] - Tr\left[\Lambda^{\frac{p}{2}}\right] = 0$, the above inequalities yield Lemma 6. □

Furthermore, we can extend Lemma 6 to any two matrices $\tilde{A}$ and A of the same size, as shown below.

Lemma 7. *Given any matrices $\tilde{A}$ and A of the same size, and $\delta > 0$, the following inequality holds for $0 < p \leq 2$:*

$$\begin{aligned} &2\left(Tr\left[\left(\tilde{A}^T\tilde{A} + \delta I\right)^{\frac{p}{2}}\right] - Tr\left[\left(A^TA + \delta I\right)^{\frac{p}{2}}\right]\right) \leq \\ &p\left(Tr\left[\tilde{A}^T\tilde{A}\left(A^TA + \delta I\right)^{\frac{p}{2}-1}\right] - Tr\left[A^TA\left(A^TA + \delta I\right)^{\frac{p}{2}-1}\right]\right). \end{aligned}$$

Proof. Since $\tilde{A}^T\tilde{A} + \delta I(\delta > 0)$ and $A^TA + \delta I(\delta > 0)$ are positive definite matrices, Lemma 6 can be applied to yield Lemma 7. □

Finally, we propose the following theorem to establish the convergence of our algorithm.

Theorem 1. *The problem in Eq.* (6) *monotonically decreases with the proposed algorithm and reaches convergence over iterations.*

Proof. Let's denote the updated x as x'. The inequality below holds if and only if the algorithm updates the objective to convergence.

$$\begin{aligned} &f(x') + \sum_i Tr\left[g_i^T(x')g_i(x')D_i\right] \\ &\leq f(x) + \sum_i Tr\left[g_i(x)^Tg_i(x)D_i\right]. \end{aligned}$$

Applying Lemma 7, we have:

$$\begin{aligned} &2\left(Tr\left[\left(g_i^T(x')g_i(x') + \delta I\right)^{\frac{p}{2}}\right] - Tr\left[\left(g_i^T(x)g_i(x) + \delta I\right)^{\frac{p}{2}}\right]\right) \\ &\leq p\Bigg(Tr\left[g_i^T(x')g_i(x')\left(g_i^T(x)g_i(x) + \delta I\right)^{\frac{p}{2}-1}\right] \\ &- Tr\left[g_i^T(x)g_i(x)\left(g_i^T(x)g_i(x) + \delta I\right)^{\frac{p}{2}-1}\right]\Bigg). \end{aligned}$$

Since $D_i = \frac{p}{2}\left(g_i^T(x)g_i(x) + \delta I\right)^{\frac{p}{2}-1}$, we have:

$$\begin{aligned} &Tr\left[\left(g_i^T(x')g_i(x') + \delta I\right)^{\frac{p}{2}}\right] - Tr\left[g_i^T(x')g_i(x')D_i\right] \qquad (24) \\ &\leq Tr\left[\left(g_i^T(x)g_i(x) + \delta I\right)^{\frac{p}{2}}\right] - Tr\left[g_i^T(x)g_i(x)D_i\right], \qquad (25) \end{aligned}$$

which implies that, when summed over iterations, we have:

$$\begin{aligned} &\sum_i Tr\left[\left(g_i^T(x')g_i(x') + \delta I\right)^{\frac{p}{2}}\right] - \sum_i Tr\left[g_i^T(x')g_i(x')D_i\right] \\ &\leq \sum_i Tr\left[\left(g_i^T(x)g_i(x) + \delta I\right)^{\frac{p}{2}}\right] - \sum_i Tr\left[g_i^T(x)g_i(x)D_i\right], \end{aligned} \tag{26}$$

Finally, by adding Equations (26) and the inequality together, we obtain:

$$\begin{aligned} &\sum_i Tr\left[\left(g_i^T(x')g_i(x') + \delta I\right)^{\frac{p}{2}}\right] + f(x') \\ &\leq \sum_i Tr\left[\left(g_i^T(x)g_i(x) + \delta I\right)^{\frac{p}{2}}\right] + f(x). \end{aligned} \tag{27}$$

It is clear that Equation (27) holds if and only if the objective converges using the proposed algorithm. Therefore, the theorem is proved. □

Additionally, it is important to note that our proposed algorithm can find a locally optimal solution for the non-convex objective, but when the objective is convex, the proposed algorithm can find a globally optimal solution.

B. Alternating Direction Method of Multipliers (ADMM)

To efficiently obtain a closed-form solution, we employ the Alternating Direction Method of Multipliers (ADMM) method. The orthogonal constraints $F_0^TF_0 = I$ and $F_I^TF_I = I$ play a crucial role in preventing degenerate solutions and ensuring numerical stability. We leverage these constraints by solving the Orthogonal Procrustes problem during the update process.

Using our new optimization framework, we rewrite the objective as:

$$\begin{aligned} \min \mathrm{Tr}&\left((S-A)^TD_S(S-A)\right) + \lambda_0\mathrm{Tr}(F_0^TL_S^pF_0) \\ &+ \sum_{I=1}^{k-1}\lambda_I\mathrm{Tr}(F_I^TL_{S_{I,I+1}}^pF_I), \\ \text{s.t.} &\sum_j s_{ij} = 1, \quad s_{ij} \geq 0, \quad F_0^TF_0 = I, \quad F_I^TF_I = I, \end{aligned} \tag{28}$$

where D_S is a diagonal matrix with the i-th diagonal element given by $\frac{p}{2}\left(|s_i - a_i|2^2 + \delta\right)^{\frac{p-2}{2}}$, and si and a_i are the i-th column vectors of matrices S and A, respectively.

Following the ADMM framework, we can rewrite the objective in Eq. (28) as:

$$\begin{aligned}
&\min \operatorname{Tr}\left((P-A)^T D_S (S-A)\right) + \lambda_0 \operatorname{Tr}(E_0^T L_S^p F_0) \\
&+ \sum_{I=1}^{k-1} \lambda_I \operatorname{Tr}(E_I^T L^p_{S_{I,I+1}} F_I) + \frac{\mu}{2}\|P - S + \frac{1}{\mu}\Lambda_S\|_F^2 \\
&+ \frac{\mu}{2}\|E_0 - F_0 + \frac{1}{\mu}\Lambda_0\|_F^2 + \sum_{I=1}^{k-1}\frac{\mu}{2}\|E_I - F_I + \frac{\mu}{2}\Lambda_I\|_F^2, \\
&\text{s.t.} \sum_j s_{ij} = 1, \quad s_{ij} \geq 0, \quad F_0^T F_0 = I, \quad F_I^T F_I = I,
\end{aligned} \tag{29}$$

where Λ_S, Λ_0, and Λ_I are the Lagrangian multipliers for the constraints $P = S$, $E_0 = F_0$, and $E_I = F_I$, respectively. We have also introduced the variables P, E_0, and E_I as auxiliary variables that facilitate the optimization process. Note that we have added the ℓ_2 penalty terms in the objective function to account for the constraint violations, and μ is a parameter that controls the trade-off between the data fidelity term and the penalty terms.

The detailed procedures to solve Eq.(29) using the ADMM method are provided in the following steps:

Step 1: Initialization.

Step 2: Solving for P. Fix all other variables except for P and solve the following optimization problem:

$$\min_P \operatorname{Tr}\left((P-A)^T D_S (S-A)\right) + \frac{\mu}{2}\|P - S + \frac{1}{\mu}\Lambda_S\|_F^2. \tag{30}$$

Taking the derivative of Eq. (30) with respect to P and setting it to 0, we have:

$$P = \frac{D_S}{\mu}(A - S) + S - \frac{\Lambda_S}{\mu}. \tag{31}$$

Step 3: Solving for S. Fix all other variables except for S and solve the following optimization problem:

$$\min_S \operatorname{Tr}\left((P-A)^T D_S (S-A)\right) + \frac{\mu}{2}\|P - S + \frac{1}{\mu}\Lambda_S\|_F^2. \tag{32}$$

Taking the derivative of Eq. (32) with respect to S and setting it to 0, we have:

$$S = \frac{D_S}{\mu}(A - P) + P + \frac{\Lambda_S}{\mu}. \tag{33}$$

Step 4: Solving for E_0. Fix all other variables except for E_0 and solve the following optimization problem:

$$\min_{E_0} \lambda_0 \operatorname{Tr}(E_0^T L_S^p F_0) + \frac{\mu}{2}\|E_0 - F_0 + \frac{1}{\mu}\Lambda_0\|_F^2. \tag{34}$$

We can simplify Eq. (34) as follows:

$$\begin{aligned}
&\lambda_0 \operatorname{Tr}(E_0^T L_S^p F_0) + \frac{\mu}{2}\|E_0 - F_0 + \frac{1}{\mu}\Lambda_0\|_F^2 \\
&= \frac{\mu}{2}\operatorname{Tr}\left(E_0 E_0^T - 2(F_0 - \frac{1}{\mu}\Lambda_0 - \frac{\lambda_0}{\mu} L_S^p F_0) E_0^T\right).
\end{aligned} \tag{35}$$

Let $M_0 = F_0 - \frac{1}{\mu}\Lambda_0 - \frac{\lambda_0}{\mu} L_S^p F_0$, then we can rewrite the optimization problem in Eq. (34) as follows:

$$\min_{E_0} \|E_0 - M_0\|_F^2, \quad \text{s.t. } e_{0_{ij}} \geq 0. \tag{36}$$

This can be decoupled to solve every element of E_0:

$$\min_{e_{0ij}} (e_{0ij} - m_{0ij})^2, \quad \text{s.t. } e_{0ij} \geq 0. \tag{37}$$

The solution can be easily obtained as $e_{0ij} = \max(m_{0ij}, 0)$.

Step 5: Solving F_0. Fix all other variables except for F_0 and solve the following optimization problem:

$$\min_{F_0} \lambda_0 \operatorname{Tr}(E_0^T L_S^p F_0) + \frac{\mu}{2}\|E_0 - F_0 + \frac{1}{\mu}\Lambda_0\|_F^2, \quad \text{s.t. } F_0^T F_0 = I. \tag{38}$$

We can simplify this problem as follows:

$$\begin{aligned}
&\lambda_0 \operatorname{Tr}(E_0^T L_S^p F_0) + \frac{\mu}{2}\|E_0 - F_0 + \frac{1}{\mu}\Lambda_0\|_F^2 \\
&= \frac{\mu}{2}\operatorname{Tr}\left(F_0 F_0^T - 2F_0\left(\frac{1}{\mu}E_0^T + \frac{1}{\mu}\Lambda_0 - \frac{\lambda_0}{\mu}E_0^T L_S^p\right)\right).
\end{aligned} \tag{39}$$

Denoting $M_0^T = \frac{1}{\mu}E_0^T + \frac{1}{\mu}\Lambda_0 - \frac{\lambda_0}{\mu}E_0^T L_S^p$, we can rewrite this problem as an Orthogonal Procrustes problem:

$$\min_{F_0} \|F_0 - M_0\|_F^2 - \|M_0\|_F^2, \quad \text{s.t. } F_0^T F_0 = I. \tag{40}$$

According to Theorem 1 in [36], this problem is equivalent to:

$$\max_{F_0} \operatorname{Tr}(F_0^T M_0), \quad \text{s.t. } F_0^T F_0 = I, \tag{41}$$

and we obtain the solution $F_0 = U_{F_0} V_{F_0}^T$ by computing the SVD of M_0: $\text{SVD}(M_0) = U_{F_0}\Sigma_{F_0}V_{F_0}^T$.

Step 6: Solving E_I is similar to **Step 4**.

Step 7: Solving F_I is similar to **Step 5**.

Step 8: Update Λ_S by $\Lambda_P = \Lambda_P + \mu(P - S)$.

Step 9: Update Λ_0 by $\Lambda_0 = \Lambda_0 + \mu(E_0 - F_0)$.

Step 10: Update Λ_I by $\Lambda_I = \Lambda_I + \mu(E_I - F_I)$.

Step 11: Update μ by $\mu = \rho\mu$.

Algorithm 2 Solve the optimization problem

```
Require: Affinity matrix $A \in \mathbb{R}^{n\times n}$
Initialization: $\lambda_0, c, \mu, \lambda_I, P, S, E_0, F_0, E_I, F_I, \Lambda_S, \Lambda_0, \Lambda_I$
repeat
    Step 1: Update P
    Step 2: Update S
    Step 3: Update $E_0$
    Step 4: Update $F_0$
    Step 5: Update $E_I$
    Step 6: Update $F_I$
    Step 7: Update Lagrangian multipliers
    Step 9: Update $\mu$
until convergence
```

IV. Experiment

A. Interaction Prediction Evaluation

In this section, we evaluate our method's performance in predicting interacting pairs. We use four standard evaluation metrics: precision, recall, AUC (area under the ROC curve), and AP (average precision). These metrics provide a comprehensive assessment of the prediction performance and enable meaningful comparisons with other methods.

To assess the performance of our method, we compare it with five competitive prediction methods: ENN [18], SFCN

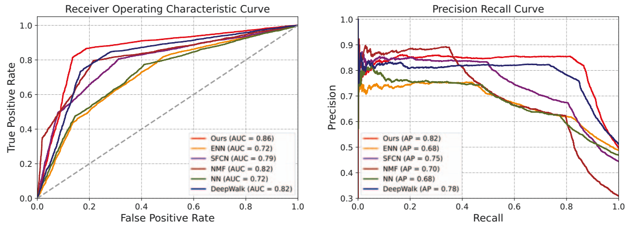


Fig. 3. ROC and PR curves for evaluating the capability of the proposed method and its counterparts to recover the hidden protein-protein interactions using 10-fold cross-validation.

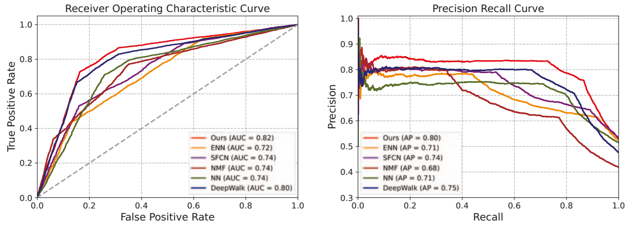


Fig. 4. ROC and PR curves for evaluating the capability of the proposed method and its counterparts to recover the hidden drug-target interactions using 10-fold cross-validation.

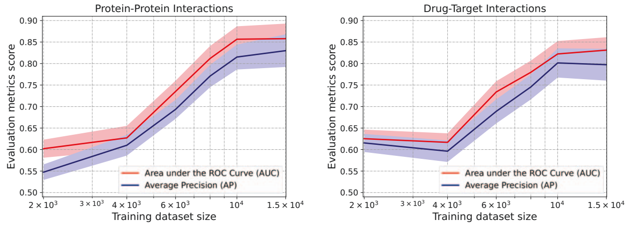


Fig. 5. Mean AUC and AP values (solid lines) and their standard deviations (shadow areas) of our method on the PPI and DTI test data with respect to different training sizes.

[24], NMF [35], NN [19], and DeepWalk [32]. These methods were selected as they represent state-of-the-art approaches for interaction prediction.

To conduct a comprehensive evaluation, we perform a 10-fold cross-validation, repeated ten times. In each trial, we randomly remove 10% of known links from the input graph and attempt to recover them using the remaining graph. The reported results of our 10-fold cross-validation represent the average values across all folds and repeats.

Figures 3 and 4 present the ROC-AUC and PR-AP curves for protein-protein interaction (PPI) and drug-target interaction (DTI) predictions, respectively. It is evident from the curves that our method consistently outperforms the compared methods. The superiority of our approach is further supported by the higher average AUC values (0.86 for PPI and 0.82 for DTI) and AP values (0.82 for PPI and 0.80 for DTI).

To assess the robustness of our method, we conduct different-fold cross-validations, as shown in Figures 6 and 7. The ROC-AUC and PR-AP curves demonstrate the stability

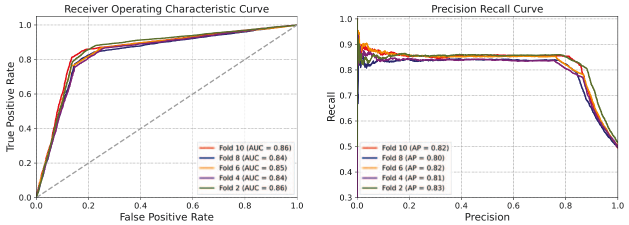


Fig. 6. ROC and PR curves for Protein-Protein interactions using different fold cross-validations.

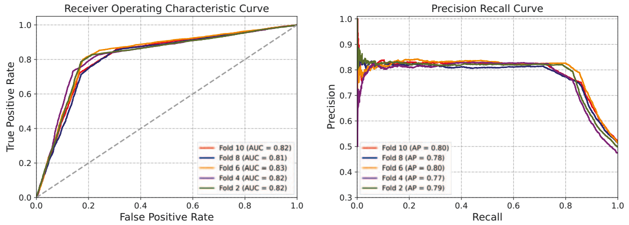


Fig. 7. ROC and PR curves for Drug-Target interactions using different fold cross-validations.

and robustness of our method, as indicated by the insignificant standard deviation observed across different-fold cross-validation on the unbalanced dataset. These results reinforce the reliability of our model for predicting hidden PPIs and DTIs.

Furthermore, we evaluate the performance of our method on datasets of different sizes. Figure 5 illustrates the mean AUC and AP values (solid lines) along with their standard deviations (shadow areas) on the PPI and DTI test data with respect to different training sizes. The trend line clearly shows a rapid increase in performance until the training size reaches 10,000. This finding suggests that our method benefits significantly from the addition of more training samples in this range, indicating the effectiveness of our approach in learning from larger training datasets.

The above evaluations and analyses confirm the superior performance, robustness, and scalability of our method in predicting hidden PPIs and DTIs.

B. Discovery of Repurposable Drugs for SARS-CoV-2

In this subsection, we present a subset of candidate drug-target interactions with the highest-ranking scores derived from our proposed method. These putative interactions are supported by published antiviral research, indicating their potential effectiveness in combating SARS-CoV-2. Table I provides the DrugBank ID, drug name, known indication, gene target, and associated validation documents with their PubMed IDs.

We highlight some of the significant candidate interactions and their supporting evidence below:

Lidocaine - IL1B: IL1B is a protein-coding gene that plays a crucial role in proinflammatory cytokines. The lung damage observed in patients with SARS-CoV-2 is largely

TABLE I

The literature-derived antiviral evidence for the repurposable drugs with the highest-ranking scores predicted by our method.

DrugBank ID	Drug Name	Known Indication	Gene Target	PubMed ID
DB00281	Lidocaine	sore throat, arrhythmia	IL1B	11876744, 32171193
DB01394	Colchicine	anti-inflammatory	TUBA1A	22814904
DB01234	Dexamethasone	anti-inflammatory	RASD1	16489124
DB01244	Bepridil	chronic stable angina	FOXO3	33597253, 27283899
DB06151	Acetylcysteine	mucolytic therapy, acetaminophen overdose	EGFR	31891230
DB06287	Sirolimus	immunosuppressant	APOE	14697997
DB08901	Ponatinib	blast phase chronic myeloid leukemia	BRAF	32873792
DB00539	Toremifene	metastatic breast cancer	CLU	18508997
DB00615	Rifabutin	mycobacterium avium complex	SLCO1B1	26482301
DB00541	Vincristine	acute lymphocytic leukemia	AKT1	19968493

related to the inflammatory response induced by cytokines such as IL1B. Lidocaine and related drugs have been found to induce immunoregulatory effects and regulate the secretion of the cytokine IL1B [6], [22].

Colchicine - TUBA1A: TUBA1A plays an important role in microtubule formation and morphologically differentiated neurologic cells. Colchicine, known for its anti-inflammatory effects, exerts its mechanism of action by suppressing microtubule formation. The interaction between colchicine and TUBA1A has been extensively studied for its therapeutic potential in various conditions [16], [25], [27].

Dexamethasone - RASD1: RASD1 is a gene expressed in multiple tissues and involved in receptor-independent signal transduction pathways. Dexamethasone, a corticosteroid, possesses anti-inflammatory properties and can affect RASD1 expression and related cellular processes [9], [14], [28].

Bepridil - FOXO3: FOXO3 plays a role in protein turnover regulation and apoptosis during the cell cycle. Bepridil, used in the treatment of chronic stable angina, has been shown to upregulate FOXO3 expression and impact signaling pathways associated with this gene [15], [30], [33].

Acetylcysteine - EGFR: EGFR is a receptor tyrosine kinase involved in cell growth, proliferation, and survival. Acetylcysteine, also known as N-acetylcysteine (NAC), acts as an antioxidant and inducer of glutathione. Combined treatment of N-acetylcysteine and gefitinib, an EGFR inhibitor, has been shown to reverse EGFR-TKI resistance in lung cancer cells [23], [26].

Sirolimus - APOE: APOE is associated with lipid particle clearance and various chronic conditions. Sirolimus, an immunosuppressant, has been investigated for its impact on APOE expression and its potential benefits in reducing atherosclerotic lesion size [8], [12].

Ponatinib - BRAF: BRAF is a protein kinase involved in cell signaling pathways. Ponatinib, a kinase inhibitor, has been extensively used for the treatment of chronic myeloid leukemia (CML). The interaction between Ponatinib and BRAF has been studied for its potential therapeutic effects [5], [7].

Toremifene - CLU: CLU is a gene associated with cellular debris clearance and apoptosis. Toremifene, used in the treatment of metastatic breast cancer, has been shown to affect CLU expression and play a role in the response to anti-estrogen treatment [4], [20].

Rifabutin - SLCO1B1: SLCO1B1 encodes a liver-specific transporter involved in drug uptake. Rifabutin, an antibiotic, has been studied for its impact on SLCO1B1 and its influence on drug exposure in tuberculosis patients [17], [31].

Vincristine - AKT1: AKT1 is a protein kinase involved in various cellular functions, including cell proliferation and survival. Vincristine, a cytotoxic compound used in the treatment of leukemia, has been shown to interact with AKT1 and modulate cell apoptosis and drug sensitivity [27], [38].

The significant overlaps between our predictions and the findings in existing literature provide strong evidence for the potential effectiveness of the suggested repurposable drugs. However, it is crucial to validate all predicted drug-target interactions in clinical and experimental trials specific to SARS-CoV-2 before considering their application in patient treatment.

Table I presents a subset of the predicted drug-target interactions along with their associated literature evidence, supporting the potential repurposing of these drugs for SARS-CoV-2.

V. Conclusion

In this study, we proposed a novel k-partite graph-clustering framework for predicting protein-protein interactions (PPIs) and drug-target interactions (DTIs). Our approach stands out from existing methods by learning a new k-partite graph with an explicit cluster structure and ensuring the desired connected components through a rank constraint on the Laplacian matrix. To enhance model robustness, we introduced the $\ell_{2,p}$-norm on the distance function and developed an efficient algorithm that integrates the alternating direction method of multipliers (ADMM) with the iteratively reweighted method.

Through extensive evaluations and experiments, we demonstrated the reliability, efficiency, and power of our method in computational approaches for PPI and DTI prediction. We compared our method with state-of-the-art approaches and consistently outperformed them in terms of precision, recall, AUC, and AP. The evaluation results, including ROC and PR curves, average AUC and AP values, and the analysis of different-fold cross-validation and scalability, all provided strong evidence of the superior performance and robustness of our approach.

Furthermore, we applied our method to discover repurposable drugs for the treatment of SARS-CoV-2. By predicting undiscovered binding affinities, we identified a subset of candidate drug-target interactions supported by existing antiviral research. These findings hold promise for potential therapeutic interventions against COVID-19. However, it is crucial to validate these predicted interactions through clinical and experimental trials specific to SARS-CoV-2 before considering their application in patient treatment.

In summary, our proposed k-partite graph-clustering framework offers a reliable and efficient computational approach for predicting PPIs and DTIs. It serves as a valuable auxiliary tool for proteomics and pharmacologic research, providing insights into potential interactions and facilitating the discovery of new therapeutic strategies.

Acknowledgment

Corresponding author: Hua Wang (huawangcs@gmail.com).

This work was supported in part by the National Science Foundation (NSF) under the grants of IIS 1652943, IIS 1849359, CNS 1932482 and CCF 2029543.

References

[1] Denis M Bayada, Hans Hamersma, and Vincent J van Geerestein. Molecular diversity and representativity in chemical databases. *Journal of chemical information and computer sciences*, 39(1):1–10, 1999.

[2] BioGRID. BioGRID database of protein, genetic and chemical interactions, 2021.

[3] Daniel Blanco-Melo, Benjamin E Nilsson-Payant, Wen-Chun Liu, Skyler Uhl, Daisy Hoagland, Rasmus Møller, Tristan X Jordan, Kohei Oishi, Maryline Panis, David Sachs, et al. Imbalanced host response to sars-cov-2 drives development of covid-19. *Cell*, 181(5):1036–1045, 2020.

[4] Vera Cappelletti, Manuela Gariboldi, Loris De Cecco, Sara Toffanin, James F Reid, Lara Lusa, Emilio Bajetta, Luigi Celio, Marco Greco, Alessandra Fabbri, et al. Patterns and changes in gene expression following neo-adjuvant anti-estrogen treatment in estrogen receptor-positive breast cancer. *Endocrine-related cancer*, 15(2):439–449, 2008.

[5] Raffaele Ciampi and Yuri E Nikiforov. Alterations of the braf gene in thyroid tumors. *Endocrine pathology*, 16(3):163–171, 2005.

[6] P Conti, G Ronconi, AL Caraffa, CE Gallenga, R Ross, I Frydas, and SK Kritas. Induction of pro-inflammatory cytokines (il-1 and il-6) and lung inflammation by coronavirus-19 (covi-19 or sars-cov-2): anti-inflammatory strategies. *J Biol Regul Homeost Agents*, 34(2):327–331, 2020.

[7] Xiomaris M Cotto-Rios, Bogos Agianian, Nadege Gitego, Emmanouil Zacharioudakis, Orsi Giricz, Yang Wu, Yiyu Zou, Amit Verma, Poulikos I Poulikakos, and Evripidis Gavathiotis. Inhibitors of braf dimers using an allosteric site. *Nature communications*, 11(1):1–16, 2020.

[8] National Research Council, Committee on Population, et al. Cells and surveys: Should biological measures be included in social science research? 2001.

[9] FDA Approved Drug Products: Dexamethasone Injection. Dexamethasone, 2022.

[10] Inderjit S Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274, 2001.

[11] DrugBank. DrugBank Database, 2021.

[12] M Merle Elloso, Neal Azrolan, Suren N Sehgal, Pa-Lang Hsu, Kristen L Phiel, Caroline A Kopec, Michael D Basso, and Steven J Adelman. Protective effect of the immunosuppressant sirolimus against aortic atherosclerosis in apo e-deficient mice. *American Journal of Transplantation*, 3(5):562–569, 2003.

[13] Ky Fan. Maximum properties and inequalities for the eigenvalues of completely continuous operators. *Proceedings of the National Academy of Sciences of the United States of America*, 37(11):760, 1951.

[14] Ming Fang, Samie R Jaffrey, Akira Sawa, Keqiang Ye, Xiaojiang Luo, and Solomon H Snyder. Dexras1: Ag protein specifically coupled to neuronal nitric oxide synthase via capon. *Neuron*, 28(1):183–193, 2000.

[15] GeneCards Summary. FOXO3, 2022.

[16] Susan Bane Hastie. Interactions of colchicine with tubulin. *Pharmacology & therapeutics*, 51(3):377–401, 1991.

[17] Stefanie Hennig, Suhashni Naiker, Tarylee Reddy, Deirdre Egan, Tracy Kellerman, Lubbe Wiesner, Andrew Owen, Helen McIlleron, and Alexander Pym. Effect of slco1b1 polymorphisms on rifabutin pharmacokinetics in african hiv-infected patients with tuberculosis. *Antimicrobial agents and chemotherapy*, 60(1):617–620, 2016.

[18] Lei Huang, Li Liao, and Cathy H Wu. Completing sparse and disconnected protein-protein network by deep learning. *BMC bioinformatics*, 19(1):1–12, 2018.

[19] Martial Hue, Michael Riffle, Jean-Philippe Vert, and William S Noble. Large-scale prediction of protein-protein interactions from structures. *BMC bioinformatics*, 11(1):1–9, 2010.

[20] Steve E Jones and Catherine Jomary. Clusterin. *The international journal of biochemistry & cell biology*, 34(5):427–431, 2002.

[21] KEGG. KEGG kyoto encyclopedia of genes and genomes, 2021.

[22] M Lahav, M Levite, L Bassani, A Lang, H Fidder, R Tal, S Bar-Meir, L Mayer, and Y Chowers. Lidocaine inhibits secretion of il-8 and il-1 β and stimulates secretion of il-1 receptor antagonist by epithelial cells. *Clinical & Experimental Immunology*, 127(2):226–233, 2002.

[23] Jun Li, Xiao-Hui Wang, Jing Hu, Meng Shi, Lu Zhang, and Hong Chen. Combined treatment with n-acetylcysteine and gefitinib overcomes drug resistance to gefitinib in nsclc cell line. *Cancer medicine*, 9(4):1495–1502, 2020.

[24] Shibao Li, Junwei Huang, Zhigang Zhang, Jianhang Liu, Tingpei Huang, and Haihua Chen. Similarity-based future common neighbors model for link prediction in complex networks. *Scientific reports*, 8(1):1–11, 2018.

[25] Yan Lu, Jianjun Chen, Min Xiao, Wei Li, and Duane D Miller. An overview of tubulin inhibitors that interact with the colchicine binding site. *Pharmaceutical research*, 29(11):2943–2971, 2012.

[26] Tetsuya Mitsudomi and Yasushi Yatabe. Epidermal growth factor receptor in relation to tumor development: Egfr gene and cancer. *The FEBS journal*, 277(2):301–308, 2010.

[27] National Library of Medicine (US), National Center for Biotechnology Information. National Library of Medicine (US), 2022.

[28] Chau H Nguyen and Val J Watts. Dexamethasone-induced ras protein 1 negatively regulates protein kinase c δ: implications for adenylyl cyclase 2 signaling. *Molecular pharmacology*, 69(5):1763–1771, 2006.

[29] Nina Nikolova and Joanna Jaworska. Approaches to measure chemical similarity–a review. *QSAR & Combinatorial Science*, 22(9-10):1006–1026, 2003.

[30] See-Hyoung Park, Young Min Chung, Jessica Ma, Qin Yang, Jonathan S Berek, and Mickey CT Hu. Pharmacological activation of foxo3 suppresses triple-negative breast cancer in vitro and in vivo. *Oncotarget*, 7(27):42110, 2016.

[31] Marja K Pasanen, Pertti J Neuvonen, and Mikko Niemi. Global analysis of genetic variation in slco1b1. 2008.

[32] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[33] Renae J Stefanetti, Sarah Voisin, Aaron Russell, and Séverine Lamon. Recent advances in understanding the role of foxo3. *F1000Research*, 7, 2018.

[34] Oron Vanunu, Oded Magger, Eytan Ruppin, Tomer Shlomi, and Roded Sharan. Associating genes and protein complexes with disease via network propagation. *PLoS computational biology*, 6(1):e1000641, 2010.

[35] Hua Wang, Heng Huang, Chris Ding, and Feiping Nie. Predicting protein–protein interactions from multimodal biological data sources via nonnegative matrix tri-factorization. *Journal of Computational Biology*, 20(4):344–358, 2013.

[36] Hua Wang, Feiping Nie, and Heng Huang. Multi-view clustering and feature learning via structured sparsity. In *International conference on machine learning*, pages 352–360. PMLR, 2013.

[37] Carme Zambrana, Alexandros Xenos, René Böttcher, Noël Malod-Dognin, and Nataša Pržulj. Network neighbors of viral targets and differentially expressed genes in covid-19 are drug target candidates. *Scientific reports*, 11(1):1–15, 2021.

[38] Hao Zhang, Bin Li, Shu-wei Bai, and Hui-juan Wang. Constitutively active akt contributes to vincristine resistance in human retinoblastoma cells. *Cancer Investigation*, 28(2):156–165, 2009.