Reconstruction of Gene Regulatory Networks by integrating biological model and a recommendation system

Yijie Wang[1†*], Justin M Fear[2†], Isabelle Berger[2], Hangnoh Lee[2], Brian Oliver[2*], and Teresa M Przytycka[3*]

[1] Computer Science Department, Indiana University, Bloomington, IN 47408, USA
[2] Laboratory of Cellular and Developmental Biology, National Institute of Diabetes and Digestive and Kidney Diseases, 50 South Drive, Bethesda, MD, 20892, USA
[3] National Center of Biotechnology Information, National Library of Medicine, NIH, Bethesda MD 20894, USA
[*] correspondence to yijwang@iu.edu, briano@niddk.nih.gov, or przytyck@ncbi.nlm.nih.gov
[†] co-first author.

Abstract. Gene Regulatory Networks (GRNs) control many aspects of cellular processes including cell differentiation, maintenance of cell type specific states, signal transduction, and response to stress. Since GRNs provide information that is essential for understanding cell function, the inference of these networks is one of the key challenges in systems biology. Leading algorithms to reconstruct GRN utilize, in addition to gene expression data, prior knowledge such as Transcription Factor (TF) DNA binding motifs or results of DNA binding experiments. However, such prior knowledge is typically incomplete hence resulting in missing values. Therefore, the integration of such incomplete prior knowledge with gene expression to elucidate the underlying GRNs remains difficult.

To address this challenge we introduce NetREX-CF – Regulatory **Net**work **R**econstruction using **EX**pression and **C**ollaborative **F**iltering – a GRN reconstruction approach that brings together a modern machine learning strategy (Collaborative Filtering model) and a biologically justified model of gene expression (sparse Network Component Analysis based model). The Collaborative Filtering (CF) model is able to overcome the incompleteness of the prior knowledge and make edge recommends for building the GRN. Complementing CF, the sparse Network Component Analysis (NCA) model can use gene expression data to validate the recommended edges. Here we combine these two approaches using a novel data integration method and show that the new approach outperforms the currently leading GRN reconstruction methods.

Furthermore, our mathematical formalization of the model has lead to a complex optimization problem of a type that has not been attempted before. Specifically, the formulation contains ℓ_0 norm that can not be separated from other variables. To fill this gap, we introduce here a new method Generalized PALM (GPALM) that allows us to solve a broad class of non-convex optimization problems and prove its convergence.

1 Introduction

Regulation of gene expression is central to cellular function. The regulatory relationships between transcription factors (TFs) and the genes they target (TGs) are captured by the Gene Regulatory Network (GRN). Inference of these cell type specific GRNs is a current challenge in systems biology. Earlier work focused on predicting regulatory networks using gene expression data alone, but these methods tend to have poor predictive power [1, 2]. Indeed, inference of network edges based solely on gene expression data is challenging; network reconstruction uses an enormous search space, and the underlying biology is multilayered with many factors including post-transcriptional and post-translation regulation contributing to TF's activity. We and others have found that network accuracy is drastically improved by including additional biological data such as chromatin structure (i.e., ATAC-Seq and ChIP-Seq), TF DNA binding motifs, and DNA sequence conservation scores [3, 4, 5, 2, 6, 7].

Additional biological data have been used as *a prior* to inform network model selection in a variety of contexts [7, 8, 9]. MerlinP [5] uses network priors to influence the objective function for model selection. While, Inferelator [3], a method built on network component analysis (NCA), uses given gene expression data and a network prior to estimate TF activity. Furthermore, Inferelator predicts the GRN by uncovering the relationship between TF activity and their target genes' expression. We recently developed NetREX [2], which is also based on the NCA model, but NetREX simultaneously estimates TF activity while modifying the prior network by adding and removing edges.

Because of the NCA model's simplicity yet biological relevance, this approach becomes the foundation of the current state-of-the-art methods for GRN reconstruction [10, 11, 12, 3, 4, 2, 13, 14]. NCA uses the prior network's structure to inform the decomposition of gene expression into TF activities [10]. Specifically, TF activities are modelled as a hidden variable accounting for the complex and often unknown relationships between TF expression and TF regulatory activity. TF activity is more robust and has been proved to be superior to TF gene expression in the task of GRN reconstruction [3]. However, NCA-based methods heavily rely on the quality of the prior network. If a prior network is very noisy, NCA-based methods cannot reliably predict TF activity, and in such circumstances the GRNs predicted by those methods are not trustworthy [2]. Therefore, building a reliable prior network becomes the key factor to employ the NCA-based methods.

A GRN prior is typically built by integrating various types of biological data, but construction of a quality prior is challenging due to the incompleteness of available data. For example, we can build a prior network by using TF-DNA binding data (e.g. ChIP-seq). However, we often only have access to ChIP-seq data for a fraction of TFs. Therefore, all interactions with TFs that do not have ChIP-seq data are considered as missing values. Similarly, computational mapping of TF-DNA binding motifs may miss true physical binding sites due to the problem of multiple testing, leading to incompleteness in the TF-DNA motif prior. Current methods for building GRNs by integrating multiple sources of prior knowledge do not directly account for the fact that there is missing data [7]. However, in the last decade we have witnessed a rapid development of machine learning methods capable dealing with large amounts of missing data. One particularly successful approach is Collaborative Filtering (CF), the method used by NETFLIX's movie recommendation system [15, 16]. Given incomplete information about a user's preferences, CF infers informative features and then applies them to provide movie recommendation for other users in the absence of complete information.

In this work, we present NetREX-CF – Regulatory **Net**work **R**construction using **EX**pression and **C**ollaborative **F**iltering – a GRN reconstruction approach that uses the idea of CF in a completely novel way, namely by combining such recommendation system with expression-based model optimization. Similar to its precursor, NetREX, NetREX-CF selects a network model by simultaneously optimizing network topology and its NCA-based fit of gene expression data. However, rather than arriving to a final network by reprogramming the edges in the prior network, NetREX-CF uses a joint optimization function to directly integrate expression data with other types of prior knowledge using CF. We demonstrate that CF takes the fullest advantage of the prior data, and when combined with the biologically relevant NCA-based model, provided a remarkable improvement over existing approaches.

Mathematically, the simultaneous optimization of network topology, fit of the NCA model, and feature selection for the CF yielded a complex optimization problem of a type that has not been attempted before. Specifically, the optimization is non-convex and non-smooth due to the binary nature of presence/absence of network edges. More importantly, the optimization contains ℓ_0 norm that can not be separated from other variables that need to be optimized. While the recently introduced PALM method [17] can solve a certain class of such non-convex optimization problems, where the ℓ_0 norm is separable (in particular the one used in NetREX), a simultaneous optimization of all three sets of parameters yields a problem that cannot be solved by PALM. To fill this gap, we introduce GPALM (Generalized PALM), a new provably convergent method for solving a broad class of non-convex optimization problems with an inseparable ℓ_0 norm. Therefore, in addition to introducing a new method to reconstruct GRNs that outcompetes previous methods, this work also provides a solution to an important class of optimization problems.

2 NetREX-CF - Method Overview

The NetREX-CF model is a novel data integration framework for reconstructing GRNs by organically utilizing both gene expression E and a set of prior networks $P = \{P^1, ...P^d\}$. The main idea behind the NetREX-CF model is an integration of two complementary optimization strategies: (i) a machine learning component designed based on Collaborative Filtering that is able to identify hidden features from the current observed prior networks P and utilize these features to recommend an improved GRN and (ii) a sparse NCA-based network remodelling component that can refine the topology of a GRN based on given gene expression E. These two computational components operate alternatively. The CF component recommends new edges to the current GRN and the sparse NCA-based network remodelling component screens the recommended edges and keeps the edges that are essential to explain the given gene expression. Once the sparse NCA-based network remodelling component confirms some of the recommended edges, the CF component further utilizes those retained recommended edges to make new edge recommendations for the sparse NCA-based network remodelling component to further examine (illustrated in Fig. 1).

Computationally, this is achieved by a simultaneous optimization of the following sets of variables: (i) the activities of TFs (matrix A), (ii) a weighted GRN (matrix S), and (iii) two feature matrices: the hidden features for target genes (X where the ith row x_i represents the hidden feature vector for gene i) and the hidden features for TFs (Y where the jth row y_j represents the hidden feature vector for TF j). The matrix A is optimized by the sparse NCA-based network remodelling component while the matrices X and Y are optimized by the Collaborative Filtering component. Notably, matrix S is the connection between the two components and should be optimized by considering both components.

Formally, $E \in \mathbb{R}^{n\times l}$ is the matrix of expression data of n genes in l experiments and prior network $P^k \in \mathbb{R}^{n\times m}, \forall k$ is a weighted adjacency matrix of the bipartite graph that records the prior knowledge of regulations between m TFs and n genes. Matrix $A \in \mathbb{R}^{m\times l}$ is the TF activity for m TFs in l samples and $S \in \mathbb{R}^{n\times m}$ is a weighted GRN. We further define penalty matrix C and observation matrix B based on the set of prior networks P. Each element in C can be computed by

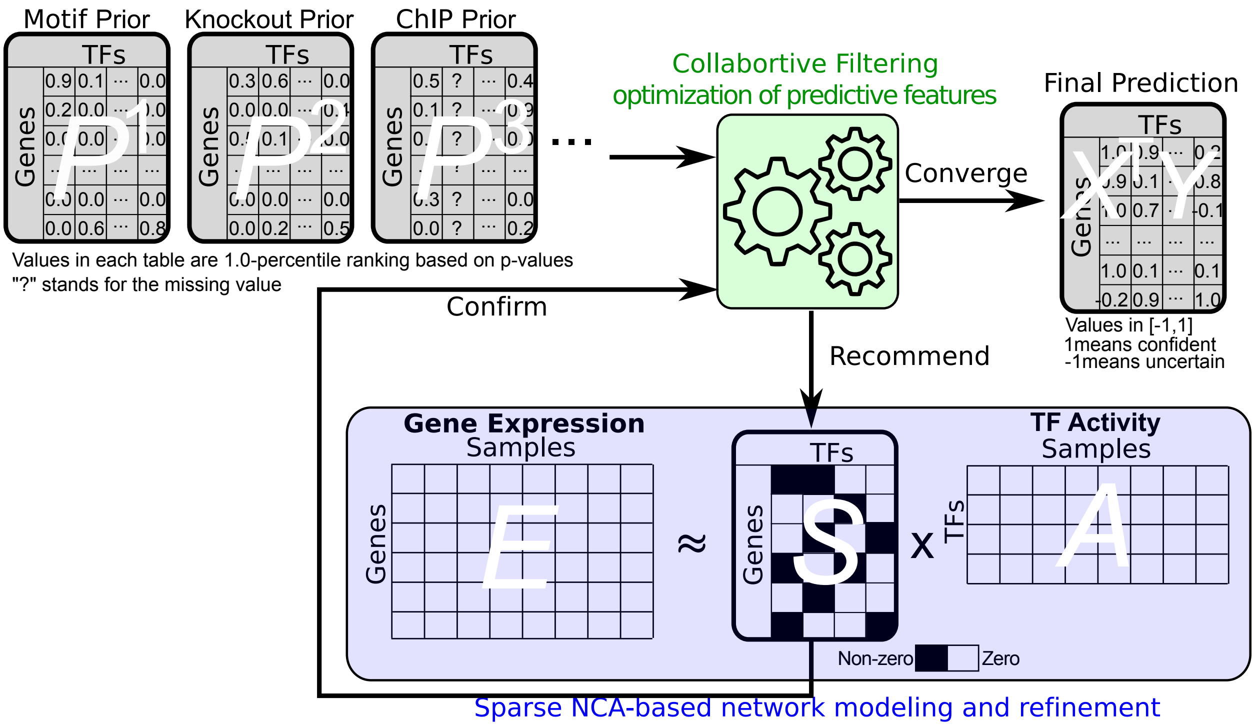


Fig. 1: Method Overview. Collaborative Filtering (CF) and NCA-based gene expression modelling alternatively inform each other during a joint optimization process: CF recommends edges to be confirmed by the NCA model and used to improve CF.

$C_{ij} = 1 + a\sum_k P^k_{ij}$ ($a = 60$ suggested by [16]) and each element in B is binary and can be computed by $B_{ij} = 1$ if $\sum_k P^k_{ij} \neq 0$ and $B_{ij} = 0$ otherwise. $X \in \mathbb{R}^{n\times h}$ contains feature vector x_i for gene i and $Y \in \mathbb{R}^{m\times h}$ contains feature vector y_j for TF j. Then, our optimization problem is formalized as following:

$$\begin{aligned} \min_{S,A,X,Y} \; & \mathcal{H}(S,A) + \lambda\mathcal{F}(S,X,Y) \\ s.t.\; & \|x_i\|^2 \leq 1,\ \forall i \\ & \|y_j\|^2 \leq 1,\ \forall j. \end{aligned} \tag{1}$$

where:

- $\mathcal{H}(S,A) := \|E - SA\|^2_F + \lambda_A\|A\|^2_F + \lambda_S\|S\|^2_F + \sum_{ij}\eta_{ij}\|S_{ij}\|_0$ is the sparse NCA-based network remodelling component; $\lambda_A\|A\|^2_F + \lambda_S\|S\|^2_F$ are standard regularization terms and $\sum_{ij}\eta_{ij}\|S_{ij}\|_0$ induces sparsity of a given prior GRN and therefore only essential edges that help to minimize $\mathcal{H}(S,A)$ are retained. $\|S_{ij}\|_0$ is the ℓ_0 norm that is 1 if $S_{ij} \neq 0$ and 0 otherwise.
- $\mathcal{F}(S,X,Y) := \sum_{i,j}\Omega_{ij}(\Theta_{ij} - x_i^T y_j)^2$ optimizes the hidden features X and Y of the Collaborative Filtering component; Θ_{ij} is a binary matrix of edges to be predicted by the hidden features in the given iteration and Ω_{ij} encodes penalties that guide the predictions. Both $\Theta_{ij} := \|S_{ij}\|_0 \oplus B_{ij}$ and $\Omega_{ij} := \bar{C}_{ij}\|S_{ij}\|_0 + C_{ij}(1 - \|S_{ij}\|_0)$ are defined based on $\|S_{ij}\|_0$ and the prior information (C_{ij}). Detailed explanation of Θ_{ij} and Ω_{ij} are provided in Method Details section. For the initialization step, both Θ_{ij} and Ω_{ij} are defined based on the prior networks only while in the subsequent steps they also take into account the results of the sparse NCA-based network remodelling component (illustrated in Fig. 1 and Fig. 3).

To solve problem (1), we first put all continuous terms together and define $H(S,A) := \|E - SA\|^2_F + \lambda_A\|A\|^2_F + \lambda_S\|S\|^2_F$ and non-continuous terms together and define $F(S,X,Y) := \sum_{i,j}\Omega_{ij}(\|S_{ij}\|_0 \oplus$

$B_{ij} - x_i^T y_j)^2 + \sum_{ij} \eta_{ij} \|S_{ij}\|_0$. Then the optimization problem has a general format of an objective function as $\Phi(S, A, X, Y) = H(S, A) + F(S, X, Y)$, where $H(S, A)$ is continuous but non-convex and $F(S, X, Y)$ is a composite function of ℓ_0 norm of elements of S and other variables so it is neither continuous nor convex. More importantly, $\|S_{ij}\|_0$ is coupled with x_i and y_j, so that $\|S_{ij}\|_0$ can not be separated from $F(S, X, Y)$ as a separated term. To the best of our knowledge, there has been no known method that can optimize such a complex and non-convex function involving inseparable ℓ_0 norm. To fill this gap, we propose here a new algorithm, Generalized PALM (GPALM) that generalizes the so called PALM method [17] and solves a class of problems of the format above, under the assumption that $F(S, X, Y)$ is lower semi-continuous (see Supplementary Material A). In the Supplementary Material B, we propose the new GPALM method and prove its convergence.

3 Experimental Results

To demonstrate the capability of our proposed GRN reconstruction method, we collect multiple datasets that measure different perspectives of the gene regulation in yeast. These datasets include TF ChIP [5, 18, 19], TF DNA binding motif [5, 20], genetic knockout [5, 21, 22], and yeast gene expression [5, 23, 24, 25]. TF ChIP, motif, and genetic knockout datasets serve as prior knowledge for TF-gene interactions in the yeast GRN. The details of these priors are summarized in Table 1 and the overlap among priors is illustrated in Table 1. We further utilize TF-gene interactions extracted from YEASTRACT database [26] as a gold standard to validate the performance of GRN reconstruction. These gold standard TF-gene interactions are supported by both DNA binding and expression evidence. The details of the gold standard TF-gene interactions and their overlap with the prior datasets are shown in Table 1. Results generated by NetREX-CF are benchmarked against the results obtained from the published sequential methods. In the following, we detail the comparison between NetREX-CF, MerlinP [5], NetREX [2], LassoStARS [4], the original CF [16], and the summation of all prior knowledge (PriorSum). For a detailed description of parameter selection for competing methods, we refer the reader to the Supplementary Material D.

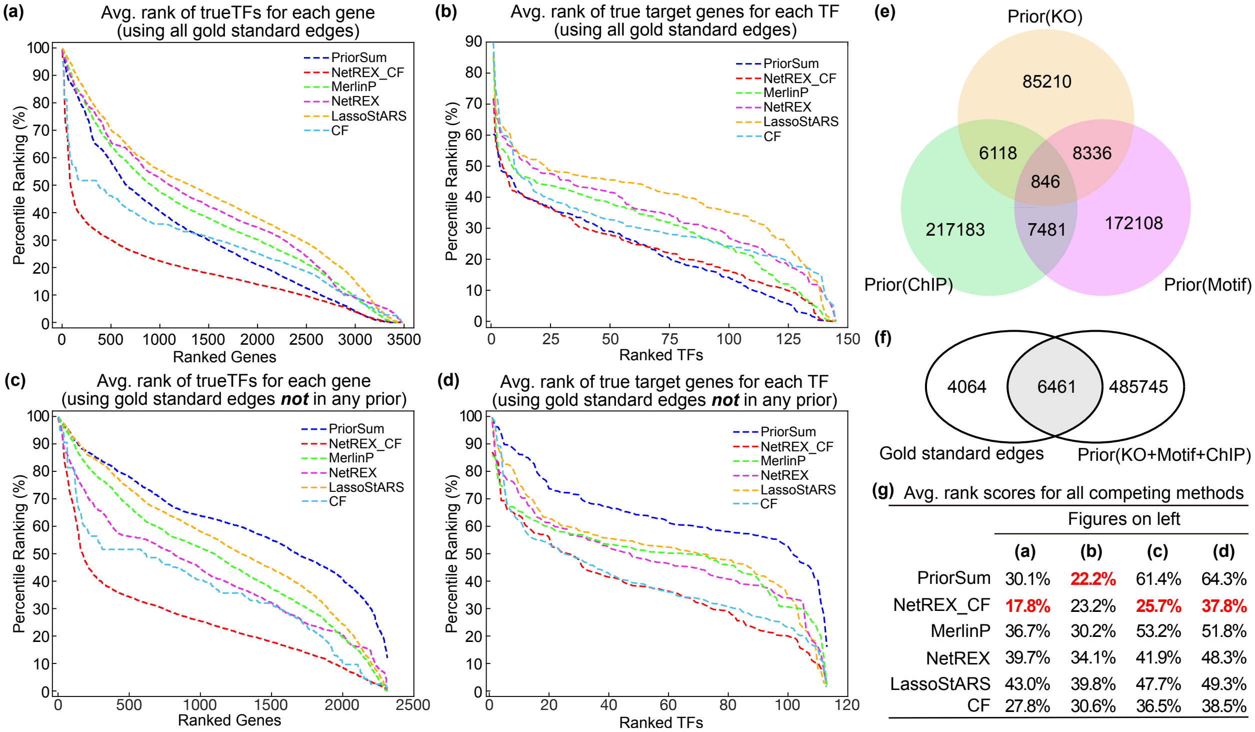
To ensure an impartial comparison, we use average percentile ranking scores. For each method and for each gene i, we can obtain a list of TFs that are predicted to regulate gene i and sort these TFs by the confidence of the prediction (most confident at the top). We use r^g_{ij} to denote the percentile-ranking of TF j within the ordered list of all TFs for gene i. Thus, $r^g_{ij} = 0\%$ means that TF j is predicted with the highest confidence to regulate gene i, preceding all other TFs in the list. Based on the gold standard TF-gene interaction dataset I, we set $I_{ij} = 1$ if TF j regulates gene i in the gold standard dataset and $I_{ij} = 0$ otherwise. For any gene i, we use the average rank of the gold standard edges in the list of TF predicted to regulate gene i as the measure quality of the prediction:

$$\overline{rank}^g_i = \frac{\sum_j r^g_{ij} I_{ij}}{\sum_j I_{ij}} \tag{2}$$

Table 1: Overlap between prior networks and the gold standard network.

Network	# Genes	# TFs	# Edges	#Overlap with Motif	#Overlap with Knockout	#Overlap with ChIP	#Overlap with YEASTRACT
Motif	5,506	197	187,079	187,079 (100%)	9,236 (8.4%)	8,717 (3.5%)	3,497 (31.0%)
Knockout	5,543	262	96,809	9,236 (4.6%)	96, 809 (100%)	7,027 (2.9%)	3,050 (27.0%)
ChIP	5,557	318	229,936	8,717 (4.3%)	7,027 (6.4%)	229,936 (100%)	2,656 (23.5%)
YEASTRACT	3,731	148	10,525	3,497 (1.7%)	3050 (2.8%)	2,656 (1.1%)	10,525 (100%)

The last four columns show overlap between different networks and parentheses shows the corresponding percentage. YEASTRACT is the gold standard network we used for performance comparison.



	Figures on left			
	(a)	**(b)**	**(c)**	**(d)**
PriorSum	30.1%	**22.2%**	61.4%	64.3%
NetREX_CF	**17.8%**	23.2%	**25.7%**	**37.8%**
MerlinP	36.7%	30.2%	53.2%	51.8%
NetREX	39.7%	34.1%	41.9%	48.3%
LassoStARS	43.0%	39.8%	47.7%	49.3%
CF	27.8%	30.6%	36.5%	38.5%

Fig. 2: Performance comparison for all competing algorithms on the yeast dataset. **(a)** The performance of the methods on the task of predicting regulating TFs: for each algorithm, we compute for each gene the rankings of gold standard edges $\overline{rank}_i^g$ adjacent to it, sort them in descending order, and plot the sorted average rankings. **(b)** The performance of the methods on the task of predicting regulated genes using a measure similar as in (a) but focusing on genes regulated by TFs. **(c)** The performance of the methods on the task of predicting regulating TFs that are not observed in prior data. The procedure is the same as in (a) but only the gold standard edges that are not included in the prior knowledge are used for the evaluation. **(d)** The performance of the methods on the task of predicting regulated genes not observed in prior data (similar to (c) but focusing on genes regulated by TFs). **(e)** Overlap between priors. **(f)** Venn diagram for gold standard dataset and the union of the three prior datasets. **(g)** Summary of the average rankings of each algorithm for tasks reported in panels (a), (b), (c), and (d).

Lower values of $\overline{rank}_i^g$ are more preferable, as they indicate gold standard TFs for gene i have lower rank than others. Furthermore, the over all ranking considering all genes can be computed by

$$\overline{rank}^g = \frac{\sum_i \overline{rank}_i^g}{\# \text{ genes in } I} \tag{3}$$

The denominator is the number of genes that have gold standard TFs in dataset I.

Similarly, for each TF we can measure the quality of the sorted list of genes predicted to be regulated by it:

$$\overline{rank}_j^t = \frac{\sum_i r_{ij}^t I_{ij}}{\sum_i I_{ij}}, \quad \overline{rank}^t = \frac{\sum_j \overline{rank}_j^t}{\# \text{ TFs in } I}, \tag{4}$$

where r_{ij}^t denotes the percentile-ranking of gene i with in the ordered list of all genes for TF j and $\overline{rank}_j^t$ is the average rankings for the gold standard genes for TF j. $\overline{rank}^t$ is the overall average rankings considering all TFs.

Fig. 2 (a) illustrates the comparison between the competing algorithms in terms of average rankings of gold standard TFs for each target gene. As shown, the sorted average ranking curve for NetREX-CF is below all other methods, indicating that the average rankings of gold standard TFs predicted by NetREX-CF for each gene are much lower than the rankings predicted by other methods. In the average rankings of gold standard genes among the genes predicted to be regulated by each TF, surprisingly, PriorSum (the weighted edge summation of three priors) outperforms all previous computational methods by a large margin. In contrast, NetREX-CF is competitive with PriorSum (Fig. 2 (b)) indicating that it takes the best advantage of the prior data. Notably, NetREX-CF outperforms the original CF, which demonstrates that the integration of CF model and sparse NCA-based model is beneficial.

Next, in order to demonstrate the advantages of NetREX-CF in predicting ranks for missing data (edges that does not appear in the prior knowledge datasets), we identified all edges that are in gold standard dataset but are not supported by any prior dataset. Indeed, as shown in Fig. 2 (f), a large portion of gold standard dataset (4,064 out of 10,525 gold standard TF-gene interactions) are not covered by any prior dataset. Therefore, we can use these gold standard interactions with missing prior data to compare the ability of the competing methods in recovering rankings under the assumption of missing data. As shown in Fig. 2 (c) and (d), NetREX-CF achieves much lower rankings for those missing data. The curves of NetREX-CF in Fig. 2 (c) and (d) are below curves of other methods by large margins except for Fig. 2 (d), where NetREX-CF is marginally better than the original CF demonstrating the benefits of integrating the CF method for predicting target genes. As shown in Fig. 2 (g), NetREX-CF achieves the lowest overall average ranking scores for all but one task where its performance is competitive with the winning method.

4 Method Details

We now describe our method in more detail. We first elucidate the NetREX-CF model presented in (1). Then, we illuminate the specified GPALM algorithm we developed to solve the NetREX-CF model.

4.1 NetREX-CF Model

Before describing the mathematical foundation of the NetREX-CF model, we provide a brief overview of Collaborative Filtering model and the sparse NCA-based network remodelling model, respectively. Next we formally introduce the integration of these two models.

Collaborative Filtering Model As illustrated in Fig. 3, to reconstruct GRNs we might have access of several prior networks, each of which reflects different perspective of the gene regulation process. Here we illustrate three prior networks: the Motif prior network, the Knockout prior network, and the ChIP prior network. In general, the prior networks are partial observation of the gene regulation process and therefore incomplete. The incompleteness of prior networks can be further demonstrated by Table 1, where there are only a small number of overlaps between the yeast prior networks and the gold standard GRN. Previous prior-based GRN reconstruction methods [7] typically make efforts to preserve those edges in the prior networks into the final GRN reconstruction but are unable to predict new edges to resolve the incompleteness of the prior networks.

Collaborative filtering, a machine learning technique, is an approach to mitigate the incompleteness of the prior networks. Collaborative filtering is able to make prediction based on partial observation. Given a set of prior networks $P = \{P^1, ...P^d\}$, the mathematical formulation of collaborative filtering can be presented by

$$\begin{aligned} \min_{x_i, y_j} : & \sum_{i,j} C_{ij} \left(B_{ij} - x_i^T y_j \right)^2 \\ s.t. & \left\| x_i \right\|^2 \leq 1, \ \forall i \\ & \left\| y_j \right\|^2 \leq 1, \ \forall j. \end{aligned} \tag{5}$$

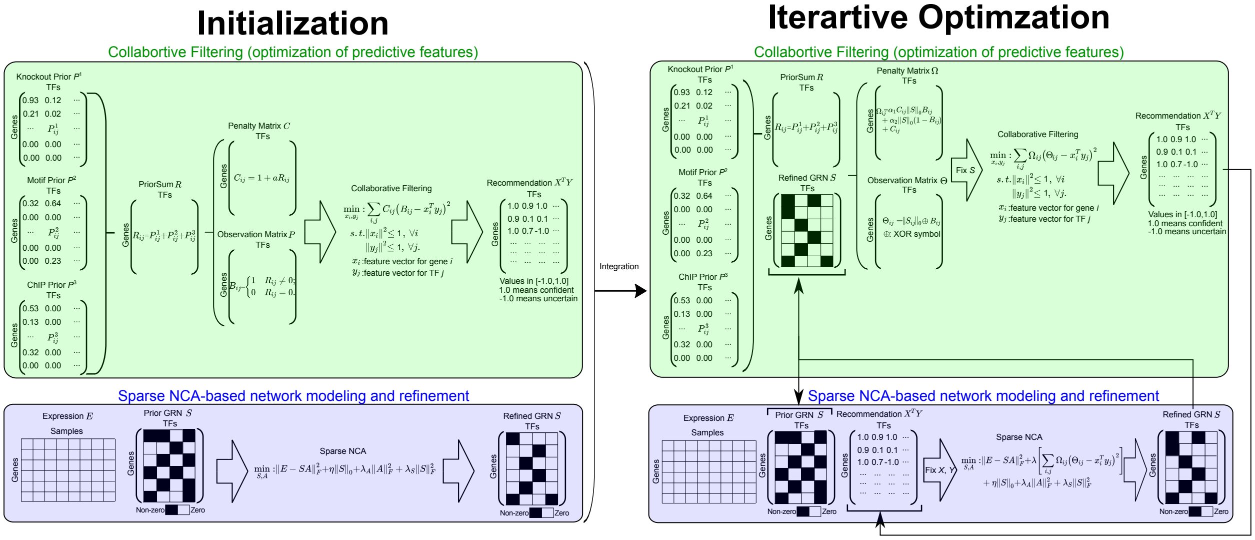

Initialization
Collabortive Filtering (optimization of predictive features)
Knockout Prior P^1
Motif Prior P^2
ChIP Prior P^3
TFs
Genes
P_{ij}^1
P_{ij}^2
P_{ij}^3
PriorSum R
$R_{ij}=P_{ij}^1+P_{ij}^2+P_{ij}^3$
Penalty Matrix C
$C_{ij}=1+aR_{ij}$
Observation Matrix P
$B_{ij}=\begin{cases}1 & R_{ij}\neq 0;\\ 0 & R_{ij}=0.\end{cases}$
Collaborative Filtering
$\min_{x_i,y_j}:\sum_{i,j}C_{ij}(B_{ij}-x_i^Ty_j)^2$
$s.t.\|x_i\|^2\leq 1,\ \forall i$
$\|y_j\|^2\leq 1,\ \forall j.$
x_i :feature vector for gene i
y_j :feature vector for TF j
Recommendation X^TY
Values in [-1.0,1.0]
1.0 means confident
-1.0 means uncertain
Sparse NCA-based network modeling and refinement
Expression E
Samples
Prior GRN S
Non-zero
Zero
Sparse NCA
$\min_{S,A}:\|E-SA\|_F^2+\eta\|S\|_0+\lambda_A\|A\|_F^2+\lambda_S\|S\|_F^2$
Refined GRN S
Integration
Iterartive Optimzation
Penalty Matrix Ω
$\Omega_{ij}=\alpha_1C_{ij}\|S\|_0B_{ij}+\alpha_2\|S\|_0(1-B_{ij})+C_{ij}$
Observation Matrix Θ
$\Theta_{ij}=\|S_{ij}\|_0\oplus B_{ij}$
$\oplus$: XOR symbol
Fix S
$\min_{x_i,y_j}:\sum_{i,j}\Omega_{ij}(\Theta_{ij}-x_i^Ty_j)^2$
Fix X, Y
$\min_{S,A}:\|E-SA\|_F^2+\lambda\left[\sum_{i,j}\Omega_{ij}(\Theta_{ij}-x_i^Ty_j)^2\right]+\eta\|S\|_0+\lambda_A\|A\|_F^2+\lambda_S\|S\|_F^2$

Fig. 3: The overview of the information flow in the NetREX-CF optimization .

We recall that x_i and y_j are hidden feature vectors for gene i and TF j, respectively, and B_{ij} is a binary number that equals to 1 when we observe the edge between gene i and TF j in any prior and equals to 0 otherwise. B_{ij} encodes that predictions that feature vectors need to make. $C_{ij} = 1 + a\sum_k P_{ij}^k$ is the penalty for learning the edge between gene i and TF j. Larger C_{ij} implies $B_{ij} = 1$ and also encourages the dot product $x_i^T y_j$ between gene feature vector x_i and TF feature vector y_j to be $x_i^T y_j = 1$. Details of the CF model is illustrated in Fig. 3 top left panel.

After solving the optimization problem (5), we can use $x_i^T y_j$, $\forall i, j$ to predict edges that are not in the prior networks. Because of the constraints in (5), we know $x_i^T y_j \in [-1, 1]$ based on Cauchy–Schwarz inequality. $x_i^T y_j$ is close to 1 implies that the collaborative filtering method recommends the edge between gene i and TF j. However, to obtain reliable predictions, it is beneficial that the correctness of the edge recommendation is further confirmed by other methods.

Sparse NCA-based Network Remodelling Model Other than utilizing prior information, such as binding properties, we can use gene expression to help build reliable GRNs. Currently, the state-of-art methods to use gene expression for reconstructing GRNs are NCA-based approaches [10, 11, 12, 3, 4, 2, 13, 14]. However, in order to use the NCA model, we need a prior network in addition to gene expression data. Given gene expression $E \in \mathbb{R}^{n\times l}$ for n genes in l samples and a prior network $S \in \mathbb{R}^{n\times m}$, the sparse NCA-based network remodelling model can be presented as

$$\min_{S,A} : \|E - SA\|_F^2 + \lambda_A\|A\|_F^2 + \lambda_S\|S\|_F^2 + \sum_{i,j} \eta_{ij}\, \|S_{ij}\|_0 \,, \tag{6}$$

where the first term is the basic NCA model [10] ($A \in \mathbb{R}^{m\times l}$ is the TF activity for m TFs in l samples) and the second and third terms are standard regularization terms and the last term involving ℓ_0 norm that is able to induce sparsity of the given prior network. Therefore, solving (6) would yield a refined GRN that only retains key edges from the prior network. The details of the sparse NCA-based network remodelling model is illustrated in Fig. 3 bottom left panel. Since for most of cases, we do not have a prior network, we need to build a reliable prior based on multiple sources of prior information.

Formulation of the NetREX-CF Model Here we propose to integrate both CF model and Sparse NCA-based Network Remodelling model. As we mentioned that the CF model needs a way to confirm the recommended edges and the sparse NCA-based network remodelling model needs a prior network to work with. Therefore, it is very natural to combine theses two models together. The CF model can recommend a prior network for the sparse NCA-based network remodelling model, and as a reward, the sparse NCA-based network remodelling model is able to confirm the recommend edges and thus allow the CF model to predict new edges. The mathematical formulation of the NetREX-CF model is

$$\begin{aligned} \min_{S,A,x_i,y_j} : & \left[\|E - SA\|_F^2 + \lambda_A \|A\|_F^2 + \lambda_S \|S\|_F^2 + \sum_{i,j} \eta_{ij} \left\|S_{ij}\right\|_0 \right] + \lambda \left[\sum_{i,j} \varOmega_{ij} \left(\varTheta_{ij} - x_i^T y_j \right)^2 \right] \\ s.t.\ & \|x_i\|^2 \leq 1,\ \forall i \\ & \|y_j\|^2 \leq 1,\ \forall j. \end{aligned} \tag{7}$$

The first square bracket is the sparse NCA-based model and the second square bracket is the CF model. λ is the balance between these two models. In the NetREX-CF model, we define $\varTheta_{ij} = \|S_{ij}\|_0 \oplus B_{ij} = \|S_{ij}\|_0 + (1 - \|S_{ij}\|_0) B_{ij}$ ($\oplus$ is XOR operation) to let the CF model not only predict edges in the prior networks B_{ij}, but also take into account the edges confirmed by the sparse NCA-based model S_{ij}. Furthermore, $\varOmega_{ij}$ is defined as $\varOmega_{ij} = \bar{C}_{ij} \|S_{ij}\|_0 + C_{ij}(1 - \|S_{ij}\|_0)$, where $\bar{C}_{ij}$ is the user defined penalty for edges confirmed by the sparse NCA-based model $S_{ij} \neq 0$ and C_{ij} is the penalty for edges not in S ($S_{ij} = 0$). The details of the NetREX-CF model is illustrated in Fig. 3 right panel. The details of how to select all the user defined parameters of the NetREX-CF model are elaborated in the Supplementary Material D.

Once we put the definition of $\varOmega_{ij}$ and $\varTheta_{ij}$ into (7) and we put $\sum_{ij} \eta_{ij} \|S_{ij}\|_0$ into the second square bracket, we have

$$\begin{aligned} \min_{S,A,x_i,y_j} : & \left[\|E - SA\|_F^2 + \lambda_A \|A\|_F^2 + \lambda_S \|S\|_F^2 \right] \\ & + \lambda \left[\sum_{i,j} \bar{C}_{ij} \|S_{ij}\|_0 + C_{ij}(1 - \|S_{ij}\|_0) \left(\|S_{ij}\|_0 + (1 - \|S_{ij}\|_0) B_{ij} - x_i^T y_j \right)^2 + \sum_{i,j} \eta_{ij} \left\|S_{ij}\right\|_0 \right] \\ s.t.\ & \|x_i\|^2 \leq 1,\ \forall i \\ & \|y_j\|^2 \leq 1,\ \forall j. \end{aligned} \tag{8}$$

Then the function in the first square bracket is continuous and we define it as $H(S, A)$. The function in the second square bracket is lower semi-continuous (Supplementary Material D.6) and we define it as $F(S, X, Y)$. Clearly, we cannot separate $\|S_{ij}\|_0$ from x_i and y_i and put every term involving $\|S_{ij}\|_0$ together as a separated term. To the best of our knowledge, there is no known method that is able to solve the optimization problem (8). In the following, we elaborate the algorithm we developed to solve the NetREX-CF model.

4.2 The NetREX-CF Algorithm

Because current methods can not solve problem (8), we propose a Generalized PALM (GPALM) algorithm that is an extension of the PALM algorithm [17]. GPALM can be used to solve this class of optimization problem involving inseparable ℓ_0 norm, which is when ℓ_0 norm cannot be separated from other optimized variables as a separated term. The format of the problem that GPALM can solve is provide in the Supplementary Material A. The GPALM algorithm and its convergence proof

Algorithm 1: The algorithm for problem (M).

```
Initialization: $A^0$, $S^0$, $X^0$, $Y^0$, $\mu_A^0$, $\mu_S^0$, $\mu_X^0$ and $\mu_Y^0$.
for $k = 0,\ 1,\ ...,\ K$ do
    $A^{k+1} \in \mathrm{prox}_{\mu_A^k}^{H(\cdot,S^k)}\left(A^k\right)$    (9)
    for $i = 0,\ 1,\ ...,\ m$ do
        $S_i^{k+1} \in \mathrm{prox}_{\mu_{S_i}^k}^{F(\cdot,x^k,y^k)}\left(S_i^k - \frac{1}{\mu_{S_i}^k}\nabla_{S_i} H\left(A^{k+1}, S_i^k\right)\right)$    (10)
    end
    $X^{k+1} \in \mathrm{prox}_{\mu_X^k}^{F(S^{k+1},\cdot,y^k)}\left(X^k\right)$    (11)
    $Y^{k+1} \in \mathrm{prox}_{\mu_Y^k}^{F(S^{k+1},x^k,\cdot)}\left(Y^k\right)$    (12)
end
```

are provided in the Supplementary Material B. Here we directly applied the GPALM algorithm to solve our NetREX-CF model. The algorithm is listed as follows. The proximal operator used in the algorithm is defined as:

$$\mathrm{prox}_\lambda^\sigma(x) := \arg\min\left\{\sigma(u) + \frac{\lambda}{2}\|u - x\|, u \in \mathbb{R}^d\right\} \tag{13}$$

The proximal operator and proximal gradient methods are often applied to replace conventional smooth optimization techniques for functions that are not continuous but can be approximated by well behaving functions (or have other nice bounding properties).

We show in the following that, for all proximal operators used in the above algorithm, we can compute the corresponding update steps by either using a closed form that we are able to derive or by reducing the computation to a convex optimization problem.

Update A The proximal operator (9) has a closed form solution.

$$A \in \mathrm{prox}_{\mu_A^k}^{H(\cdot,S^k)}\left(A^k\right) = \left((S^k)^T S^k + \frac{2\lambda_A + \mu_A^k}{2} I\right)^{-1}\left(E^T S^k + \frac{\mu_A^k}{2}(A^k)^T\right)^T, \tag{14}$$

where μ_A^k is the Lipschitz constant that can be computed by $\mu_A^k = \left\|(S^k)^T S^k + \lambda_A I\right\|_F$. The details of the derivation related to update A can be found in Supplementary Material C.1.

Update S Similarly, the proximal operator (10) also has a closed form solution.

$$S_{ij}^{k+1} \in \mathrm{prox}_{\mu_{S_i}^k}^{F(\cdot,x^k,y^k)}\left(S_i^k - \frac{1}{\mu_{S_i}^k}\nabla_{S_i} H\left(A^{k+1}, S_i^k\right)\right) = \arg\min\left\{\left(S_{ij} - U_{ij}^k\right)^2 + c_{ij}^2\|S_{ij}\|_0\right\}, \tag{15}$$

where $c_{ij} = \sqrt{\frac{2}{\mu_{S_{ij}}^k}\left\{\lambda\left[\bar{C}_{ij}(1 - B_{ij})(1 + 2(B_{ij} - x_i^T y_j)) + (\bar{C}_{ij} - C_{ij})(B_{ij} - x_i^T y_j)^2\right] + \ \eta_{ij}\right\}}$ and μ_S^k is the Lipschitz constant that can be computed by $\mu_S^k = \left\|A^{k+1}(A^{k+1})^T + \lambda_S I\right\|_F$. Therefore, the closed solution of the above problem is

$$S_{ij}^{k+1} = \begin{cases} U_{ij}^k, & \text{if } \left|U_{ij}^k\right| > c_{ij}; \\ \{0, c_{ij}\}, & \text{if } \left|U_{ij}^k\right| = c_{ij}; \\ 0, & o.w.. \end{cases} \tag{16}$$

The details of the derivation related to update S can be found in the Supplementary Material C.2.

Update X Each row x_i of X needs to be updated by solving the following proximal operator.

$$x_i^{k+1} \in \text{prox}_{\mu_x^k}^{F\left(S^{k+1},\cdot,y^k\right)}\left(x_i^k\right) = \arg\min_{\|x_i\|^2 \le 1} \left\{x_i^T \phi x_i - \varphi x_i\right\}, \tag{17}$$

where $\phi = \frac{\mu_x^k}{2} I_{h\times h} + Y^k \widetilde{A} Y^{k^T}$ and $\varphi = 2\bar{S}_i \widetilde{A} Y^{k^T} + \mu_x^k x_i^{k^T}$. $\widetilde{A}$ be the diagonal matrix with the values $\bar{A}_{i1}, \bar{A}_{i2}, ..\bar{A}_{im}$ on the diagonal, where $\bar{A}_{ij} = \lambda\left(\bar{C}_{ij} + (\bar{C}_{ij} - C_{ij})\left\|S_{ij}^{k+1}\right\|_0\right)$. And $\bar{S}_i$ is defined as $\bar{S}_i = \left[\left\|S_{i1}^{k+1}\right\|_0 \oplus B_{i1}, .., \left\|S_{in}^{k+1}\right\|_0 \oplus B_{im}\right]$. Since the problem becomes a Quadratically Constrained Quadratic Program (QCQP), we leave the rest to the CVXPY python package [27, 28]. The details of the derivation related to update X can be found in the Supplementary Material C.3.

Update Y Each row y_i of Y needs to be updated by solving the following proximal operator.

$$y_j^{k+1} \in \text{prox}_{\mu_Y^k}^{F\left(S^{k+1},x^k,\cdot\right)}\left(y_j^k\right) = \arg\min_{\|y_j\|^2 \le 1} \left\{y_j^T \phi y_j - \varphi y_j\right\}, \tag{18}$$

where $\phi = X^{k+1} \widetilde{A} X^{k+1^T} + \frac{\mu_y^k}{2} I_{p\times p}$ and $\varphi = 2\bar{S}_j^T \widetilde{A} X^{k+1^T} + \mu_y^k y_j^{k^T}$. $\widetilde{A}$ that is also a diagonal matrix with the values $\bar{A}_{1j}, \bar{A}_{2j}, ..\bar{A}_{mj}$ on the diagonal and $\bar{S}_j = \left[\left\|S_{1j}^{k+1}\right\|_0 \oplus B_{ij}, .., \left\|S_{nj}^{k+1}\right\|_0 \oplus B_{nj}\right]^T$. Since the problem also becomes a QCQP, we leave the rest to the CVXPY python package. The details of the derivation related to update Y can be found in the Supplementary Material C.4.

5 Conclusions

Data integration and predictive modelling are the two key tasks of Computational Biology. However, these two tasks are rarely considered together. GRN reconstruction is an example of an important and challenging computational biological problem that can benefit from both approaches. Here we propose a method that combines machine learning based data integration strategy and a gene expression modelling approach into one global iterative optimization strategy where machine learning component informs the expression based modeling component and vice versa. Our new integrative GRN reconstruction method outperforms previous computational methods for this task demonstrating the power of our integrative approach.

We believe that the general approach presented in this study provides not only an important step towards reconstructing better GRNs, but it has also a potential to become a paradigm for addressing other optimization problems in computational biology.

Acknowledgement

This research was supported by the Intramural Research Program of the National Library of Medicine and the National Institute of Diabetes and Digestive and Kidney Diseases at the National Institutes of Health, USA.

References

[1] Daniel Marbach et al. “Revealing strengths and weaknesses of methods for gene network inference.” In: *Proceedings of the National Academy of Sciences of the United States of America* 107.14 (2010), pp. 6286–91.

[2] Y. Wang et al. “Reprogramming of regulatory network using expression uncovers sex-specific gene regulation in Drosophila”. In: *Nat Commun* 9.1 (Oct. 2018), p. 4061.

[3] Mario L. Arrieta-Ortiz et al. “An experimentally supported model of the Bacillus subtilis global transcriptional regulatory network.” In: *Molecular systems biology* 11.11 (2015), p. 839.

[4] E. R. Miraldi et al. “Leveraging chromatin accessibility for transcriptional regulatory network inference in T Helper 17 Cells”. In: *Genome Res.* 29.3 (Mar. 2019), pp. 449–463.

[5] Alireza F Siahpirani and Sushmita Roy. “A prior-based integrative framework for functional transcriptional regulatory network inference.” In: *Nucleic acids research* 45.4 (2016), gkw963.

[6] K. Y. Lam et al. “Fused Regression for Multi-source Gene Regulatory Network Inference”. In: *PLoS Comput. Biol.* 12.12 (Dec. 2016), e1005157.

[7] Daniel Marbach et al. “Predictive regulatory models in Drosophila melanogaster by integrative inference of transcriptional networks”. In: *Genome Research* 22.7 (2012), pp. 1334–1349.

[8] Adriano V Werhli and Dirk Husmeier. “Gene regulatory network reconstruction by Bayesian integration of prior knowledge and/or different experimental conditions”. en. In: *J. Bioinform. Comput. Biol.* 6.3 (June 2008), pp. 543–572.

[9] Sach Mukherjee and Terence P Speed. “Network inference using informative priors”. en. In: *Proc. Natl. Acad. Sci. U. S. A.* 105.38 (Sept. 2008), pp. 14313–14318.

[10] J. C. Liao et al. “Network component analysis: reconstruction of regulatory signals in biological systems”. In: *Proc. Natl. Acad. Sci. U.S.A.* 100.26 (Dec. 2003), pp. 15522–15527.

[11] J. Wang et al. “A transcriptional dynamic network during Arabidopsis thaliana pollen development”. In: *BMC Syst Biol* 5 Suppl 3 (2011), S8.

[12] A. Misra and G. Sriram. “Network component analysis provides quantitative insights on an Arabidopsis transcription factor-gene regulatory network”. In: *BMC Syst Biol* 7 (Nov. 2013), p. 126.

[13] L. M. Tran, D. R. Hyduke, and J. C. Liao. “Trimming of mammalian transcriptional networks using network component analysis”. In: *BMC Bioinformatics* 11 (Oct. 2010), p. 511.

[14] J. M. Buescher et al. “Global network reorganization during dynamic adaptations of Bacillus subtilis metabolism”. In: *Science* 335.6072 (Mar. 2012), pp. 1099–1103.

[15] Yehuda Koren, Robert Bell, and Chris Volinsky. “Matrix Factorization Techniques for Recommender Systems”. In: *Computer* 42.8 (Aug. 2009), pp. 30–37.

[16] Y. Hu, Y. Koren, and C. Volinsky. “Collaborative filtering for implicit feedback datasets”. In: *Data Mining, 2008. ICDM’08. Eighth IEEE International Conference on.* IEEE. 2008, pp. 263–272.

[17] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. “Proximal alternating linearized minimization for nonconvex and nonsmooth problems”. In: *Mathematical Programming* 146.1-2 (2014), pp. 459–494.

[18] C. T. Harbison et al. “Transcriptional regulatory code of a eukaryotic genome”. In: *Nature* 431.7004 (Sept. 2004), pp. 99–104.

[19] B. J. Venters et al. “A comprehensive genomic binding map of gene and chromatin regulatory proteins in Saccharomyces”. In: *Mol. Cell* 41.4 (Feb. 2011), pp. 480–492.

[20] R. Gordan et al. “Curated collection of yeast transcription factor DNA binding specificity data reveals novel structural and gene regulatory insights”. In: *Genome Biol.* 12.12 (Dec. 2011), R125.

[21] Z. Hu, P. J. Killion, and V. R. Iyer. “Genetic reconstruction of a functional transcriptional regulatory network”. In: *Nat. Genet.* 39.5 (May 2007), pp. 683–687.

[22] J. Reimand et al. “Comprehensive reanalysis of transcription factor knockout expression data in Saccharomyces cerevisiae reveals many new targets”. In: *Nucleic Acids Res.* 38.14 (Aug. 2010), pp. 4768–4777.

[23] R. B. Brem and L. Kruglyak. “The landscape of genetic complexity across 5,700 gene expression traits in yeast”. In: *Proc. Natl. Acad. Sci. U.S.A.* 102.5 (Feb. 2005), pp. 1572–1577.

[24] E. N. Smith and L. Kruglyak. “Gene-environment interaction in yeast gene expression”. In: *PLoS Biol.* 6.4 (Apr. 2008), e83.

[25] J. Zhu et al. “Stitching together multiple data dimensions reveals interacting metabolomic and transcriptomic networks that modulate cell regulation”. In: *PLoS Biol.* 10.4 (2012), e1001301.

[26] M. C. Teixeira et al. “YEASTRACT: an upgraded database for the analysis of transcription regulatory networks in Saccharomyces cerevisiae”. In: *Nucleic Acids Res.* 46.D1 (Jan. 2018), pp. D348–D353.

[27] Akshay Agrawal et al. “A Rewriting System for Convex Optimization Problems”. In: *Journal of Control and Decision* 5.1 (2018), pp. 42–60.

[28] Steven Diamond and Stephen Boyd. “CVXPY: A Python-Embedded Modeling Language for Convex Optimization”. In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.

Supplementary Materials

We extend the original PALM algorithm [17] and propose the GPALM algorithm that can solve more general problems. The format of the problem that GPALM can solve is explained in section A. The actual algorithm and the its convergence proof are provided in section B.

A GPALM Preliminary

A.1 The problem and basic assumptions

We are interested in solving the non-convex and non-smooth minimization problem with the following structure

$$(\mathrm{M}) \qquad \min : \Psi\left(X, Y, Z\right) := H\left(X, Y\right) + F\left(Y, Z\right), \tag{19}$$

where we have the following assumption:

Assumption 1. *The assumptions for problem (M) is as follow:*

1. $H : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ *is a* C^1 *function.*
2. $F : \mathbb{R}^m \times \mathbb{R}^l \to (-\infty + \infty]$ *is a proper and lower semicontinuous (PLS) function. And* $F(Y, Z)$ *has the following structure* $F(Y, Z) := \sum_{i=1}^{m} p_i(Z) g_i(Y_i) + Q(Z)$, *where* $p_i : \mathbb{R}^l \to \mathbb{R}$ *is Lipschitz continuous with moduli* $L_i(Z)$ *and* $p_i(Z) > 0$, $\forall i$ *and* $g_i : \mathbb{R} \to \mathbb{R}$ *is lower semicontinuous and* $\sup g_i(Y_i) < \lambda_i$, $\forall i$ *and* $Q : \mathbb{R}^l \to \mathbb{R}$ *is Lipschitz continuous with moduli* $L_Q(Z)$. $Y = [Y_1,\ ...,\ Y_i,\ ...,\ Y_m]$.
3. $\inf_{\mathbb{R}^n \times \mathbb{R}^m} H > -\infty$ *and* $\inf_{\mathbb{R}^m \times \mathbb{R}^l} F > -\infty$.
4. *For any* Y *the function* $X \to H(X, Y)$ *is* $C^{1,1}_{L_X(Y)}$, *namely the partial gradient* $\nabla_X H(X, Y)$ *is globally Lipschitz with moduli* $L_1(Y)$, *that is*

   $$\left\| \nabla_X H(X_1, Y) - \nabla_X H(X_2, Y) \right\| \leq L_1(Y) \left\| X_1 - X_2 \right\|. \tag{20}$$

   Likewise, for any fixed X *the function* $Y_i \to H(X, Y_i)$ *is assumed to be* $C^{1,1}_{L_{Y_i}(X)}$.
5. *For any fixed* Y *the function* $Z \to F(Y, Z)$ *is assumed to be* $C^{1,1}_{L_Z(Y)}$.
6. ∇H *is Lipschitz continuous on bounded subsets of* $\mathbb{R}^n \times \mathbb{R}^m$. *In other words, for each bounded subsets* $T_1 \times T_2$ *of* $\mathbb{R}^n \times \mathbb{R}^m$ *there exist* $M > 0$ *such that any* (X_1, Y_1) *and* (X_2, Y_2):

   $$\left\| \left(\nabla_X H\left(X_1, Y_1\right) - \nabla_X H\left(X_2, Y_2\right), \nabla_Y H\left(X_1, Y_1\right) - \nabla_Y H\left(X_2, Y_2\right) \right) \right\| \leq M \left\| \left(X_1 - X_2, Y_1 - Y_2 \right) \right\|. \tag{21}$$

A.2 Subdifferentials of nonconvex and nonsmooth functions

Definition 1. *Let* $\sigma : \mathbb{R}^d \to (-\infty, +\infty]$ *be a PLS function. For a given* $x \in \operatorname{dom} \sigma$, *the Frechet subdifferential of* σ *at* x, *written* $\hat{\partial}\sigma(x)$, *is the set of all vectors* $u \in \mathbb{R}^d$ *which satisfy*

$$\lim_{y \neq x} \inf_{y \to x} \frac{\sigma(y) - \sigma(x) - < u, y - x >}{\|y - x\|} \geq 0. \tag{22}$$

When $x \in \operatorname{dom} \sigma$, *we set* $\hat{\partial}\sigma(x) = \emptyset$.

Proposition 1. $\partial(\lambda f(x)) = \lambda \partial f(x)$ *for any* $\lambda > 0$.

The proposition can be proved based Definition 1.

A.3 Proximal map

Let $\sigma : \mathbb{R}^d \to (-\infty, +\infty]$ be a PLS function. Given $x \in \mathbb{R}^d$ and $t > 0$, the proximal map associate to σ id defined by:

$$\operatorname{prox}_\lambda^\sigma(x) := \arg\min \left\{ \sigma(u) + \frac{\lambda}{2} \|u - x\|, u \in \mathbb{R}^d \right\} \tag{23}$$

The proximal map has the following important property (Lemma 3.2 in []).

Lemma 1. *Let $h : \mathbb{R}^d \to \mathbb{R}$ be a continuously differentiable function with gradient ∇h assumed L_h Lipschitz continuous and let $\sigma : \mathbb{R}^d \to (-\infty, +\infty]$ be a proper and lower semicontinuous function with $\inf_{\mathbb{R}^d} \sigma > -\infty$. Fix any $t > L_h$, then for any $u \in \operatorname{dom}\sigma$ and any $u^+ \in \mathbb{R}^d$ defined by*

$$u^+ \in \operatorname{prox}_t^\sigma \left(u - \frac{1}{t} \nabla h(u) \right), \tag{24}$$

we have

$$h(u^+) + \sigma(u^+) \leq h(u) + \sigma(u) - \frac{1}{2}(t - L_h) \left\| u^+ - u \right\|^2. \tag{25}$$

B GPALM Algorithm and its Convergence Analysis

B.1 The Algorithm

Here we first write out the algorithm that is able to solve problem (M) with convergence guarantee.

Algorithm 2: The algorithm for problem (M).

```
Initialization: $X^0$, $Y^0$, and $Z^0$.
for $k = 0,\ 1,\ ...,\ K$ do
    $X^{k+1} \in \operatorname{prox}_{\mu_X^k}^{H(\cdot, Y^k)} \left( X^k \right)$    (26)
    for $i = 0,\ 1,\ ...,\ m$ do
        $Y_i^{k+1} \in \operatorname{prox}_{\mu_{Y_i}^k}^{F(\cdot, Z^k)} \left( Y_i^k - \frac{1}{\mu_{Y_i}^k} \nabla_{Y_i} H \left( X^{k+1}, Y_i^k \right) \right)$    (27)
    end
    $Z^{k+1} \in \operatorname{prox}_{\mu_Z^k}^{F(Y^{k+1}, \cdot)} \left( Z^k \right)$    (28)
end
```

B.2 Convergence analysis

The proof procedure is followed the proofs introduced in the original PALM algorithm [17].

Theorem 1. *Assume $\Psi(B)$ is a PLS function with $\inf \Psi > -\infty$, the sequence $\left\{ B^k \right\}_{k \in \mathbb{N}}$ is a Cauchy sequence and converges to a critical point of $\Psi(B)$, if the following four conditions hold []:*

(i) Sufficiently decreasing: there exist some positive constant $\rho_1 > 0$, such that

$$\Psi(B^k) - \Psi(B^{k+1}) \geq \rho_1 \left\| B^{k+1} - B^k \right\|^2, \forall k. \tag{29}$$

(ii) Relative error: there exist some positive constant $\rho_2 > 0$, such that for any $w^k \in \partial\Psi(B^k)$,

$$\left\| w^k \right\| \leq \rho_2 \left\| B^{k+1} - B^k \right\|, \forall k. \tag{30}$$

(iii) Continuity: there exist a subsequence $\left\{B^{k_j}\right\}_{j\in\mathbb{N}}$ and B^*, such that

$$B^{k_j} \to B^*,\ \Psi\left(B^{k_j}\right) \to \Psi\left(B^*\right),\ \text{as } j \to +\infty. \tag{31}$$

(iv) KL property: Ψ satisfies KL property in its effective domain.

By the theorem above, we only need to check that the sequence generated by Algorithm 2 satisfy the conditions (i) - (iv).

Proposition 2. *Algorithm 2 is a global convergence algorithm.*

Proof. Follow Theorem 1, we prove Algorithm 2 satisfies conditions (i)- (iv).

Condition (i). Based on (26), we know

$$X^{k+1} \in \operatorname{prox}_{\mu_X^k}^{H(\cdot,Y^k)}\left(X^k\right) = \arg\min\left\{H\left(X, Y^k\right) + \frac{1}{\mu_X^k}\left\|X - X^k\right\|, X \in \mathbb{R}^n\right\}, \tag{32}$$

which implies

$$H\left(X^{k+1}, Y^k\right) + F\left(Y^k, Z^k\right) \leq H\left(X^k, Y^k\right) + F\left(Y^k, Z^k\right) - \frac{\mu_X^k}{2}\left\|X^{k+1} - X^k\right\| \tag{33}$$

We then apply Lemma 1 to (27),

$$H\left(X^{k+1}, Y_i^{k+1}\right) + F\left(Y_i^{k+1}, Z^k\right) \leq H\left(X^{k+1}, Y_i^k\right) + F\left(Y_i^k, Z^k\right) - \frac{1}{2}\left(\mu_{Y_i}^k - L_Y(X^{k+1})\right)\left\|Y_i^{k+1} - Y_i^k\right\| \tag{34}$$

Similar to the derivation related to X, for Z we get

$$H\left(X^{k+1}, Y^{k+1}\right) + F\left(Y^{k+1}, Z^{k+1}\right) \leq H\left(X^{k+1}, Y^{k+1}\right) + F\left(Y^{k+1}, Z^k\right) - \frac{\mu_Z^k}{2}\left\|Z^{k+1} - Z^k\right\| \tag{35}$$

Let $B^k = \left(X^k, Y^k, Z^k\right)$ and sum over equations from (33) to (35). We have

$$\Psi\left(B^{k+1}\right) \leq \Psi\left(B^k\right) - \frac{\mu_X^k}{2}\left\|X^{k+1} - X^k\right\| - \sum_i \frac{1}{2}\left(\mu_{Y_i}^k - L_{Y_i}(X^{k+1})\right)\left\|Y_i^{k+1} - Y_i^k\right\| - \frac{\mu_Z^k}{2}\left\|Z^{k+1} - Z^k\right\|. \tag{36}$$

We know that μ_X^k, μ_Y^k, and μ_Z^k have their lower bound and $\mu_{Y_i}^k > L_Y(X^{k+1})$. Therefore, we can get $\rho_1 = \frac{\mu_Y^k}{2} + \sum_i \frac{1}{2}\left(\mu_{Y_i}^k - L_{Y_i}(X^{k+1})\right) + \frac{\mu_Z^k}{2}$. Then for $B^k = \left(X^k, Y^k, Z^k\right)$ we have

$$\Psi(B^k) - \Psi(B^{k+1}) \geq \rho_1 \left\| B^{k+1} - B^k \right\|^2, \forall k \tag{37}$$

tha proves condition (i).

Condition (ii). Writing down the optimality condition for (26), we have

$$\nabla_X H\left(X^{k-1}, Y^{k-1}\right) + \mu_X^{k-1}\left(X^k - X^{k-1}\right) = 0. \tag{38}$$

Let $w_X^k := -\mu_X^{k-1}\left(X^k - X^{k-1}\right) - \nabla_X H\left(X^{k-1}, Y^{k-1}\right) + \nabla_X H\left(X^k, Y^k\right)$. It is easy to prove that $w_X^k \in \partial_X \varPsi\left(X^k, Y^k, Z^k\right)$. Then

$$\begin{aligned}\left\|w_X^k\right\| &\le \mu_X^{k-1}\left\|X^k - X^{k-1}\right\| + \left\|\nabla_X H\left(X^k, Y^k\right) - \nabla_X H\left(X^{k-1}, Y^{k-1}\right)\right\| \\ &\le \mu_X^{k-1}\left\|X^k - X^{k-1}\right\| + M\left(\left\|X^k - X^{k-1}\right\| + \left\|Y^k - Y^{k-1}\right\|\right) \\ &\le \left(\mu_X^{k-1} + 2M\right)\left\|B^k - B^{k-1}\right\|.\end{aligned} \tag{39}$$

The first inequality comes from the fact that ∇H is Lipschitz continuous on bounded subset $\mathbb{R}^n \times \mathbb{R}^m$ as assumed in Assumption 1 (6).
The optimality condition for (27), we have

$$\nabla_{Y_i} H\left(X^k, Y_i^{k-1}\right) + \mu_{Y_i}^{k-1}\left(Y_i^k - Y_i^{k-1}\right) + \partial_{Y_i} F\left(Y_i^k, Z^{k-1}\right) = 0. \tag{40}$$

Let $w_{Y_i}^k := -\mu_{Y_i}^k\left(Y_i^{k+1} - Y_i^k\right) - \nabla_{Y_i} H\left(X^k, Y_i^{k-1}\right) + \nabla_{Y_i} H\left(X^k, Y_i^k\right) - \partial_{Y_i} F\left(Y_i^k, Z^{k-1}\right) + \partial_{Y_i} F\left(Y_i^k, Z^k\right)$. Clearly, $w_{Y_i}^k \in \partial_Y \varPsi\left(X^k,\ \cdots,\ Y_{i-1}^k,\ Y_i^k,\ Y_{i+1}^k,\ \cdots,\ Z^k\right)$, then we have

$$\begin{aligned}\left\|w_{Y_i}^k\right\| &\le \mu_{Y_i}^k\left\|Y_i^{k+1} - Y_i^k\right\| + \left\|\nabla_{Y_i} H\left(X^k, Y_i^k\right) - \nabla_{Y_i} H\left(X^k, Y_i^{k-1}\right)\right\| + \left\|\partial_{Y_i} F\left(Y_i^k, Z^k\right) - \partial_{Y_i} F\left(Y_i^k, Z^{k-1}\right)\right\| \\ &\le \mu_{Y_i}^k\left\|Y_i^{k+1} - Y_i^k\right\| + M_{Y_i}\left\|Y_i^{k+1} - Y_i^k\right\| + \left\|\partial_{Y_i}\left(p_i(Z^k)g_i(Y_i)\right) - \partial_{Y_i}\left(p_i(Z^{k-1})g_i(Y_i)\right)\right\| \\ &\le (\mu_{Y_i}^k + M_{Y_i})\left\|Y_i^{k+1} - Y_i^k\right\| + \left\|\partial g_i(Y_i)\left(p_i(Z^k) - p_i(Z^{k-1})\right)\right\| \\ &\le (\mu_{Y_i}^k + M_{Y_i})\left\|Y_i^{k+1} - Y_i^k\right\| + M_i^Z\left\|\partial g_i(Y_i)\right\|\left\|\left(Z^k - Z^{k-1}\right)\right\| \\ &\le \left(\mu_{Y_i}^k + M_{Y_i} + M_i^Z U_{Y_i}\right)\left\|B^k - B^{k-1}\right\|.\end{aligned} \tag{41}$$

The second inequality utilizes the structure of $F(Y, X)$ introduced in Assumption 1 (2). The third inequality uses Proposition 1. We set $M_{Y_i} > L_{Y_i}(X)$, $M_i^Z > L_i(Z)$, and $U_{Y_i} > U_i$.
Similar to things related to X, writing down the optimality condition for (28),

$$\nabla_Z F\left(Y^k, Z^{k-1}\right) + \mu_Z^{k-1}\left(Z^k - Z^{k-1}\right) = 0. \tag{42}$$

Let $w_Z^k := -\mu_Z^{k-1}\left(Z^k - Z^{k-1}\right) - \nabla_Z F\left(Y^k, Z^{k-1}\right) + \nabla_Z F\left(Y^k, Z^k\right)$. We find that $w_Z^k \in \partial_Z \varPsi\left(X^k, Y^k, Z^k\right)$ and we have

$$\left\|w_Z^k\right\| \le \left(\mu_Z^{k-1} + M_Z\right)\left\|B^{k+1} - B^k\right\|, \tag{43}$$

where $M_Z > L_Z(Y)$.
Let $\rho_2 = \max\left\{\mu_X^{k-1} + 2M, \mu_{Y_i}^k + M_{Y_i} + M_i^Z U_{Y_i}, \mu_Z^{k-1} + M_Z\right\}$ and sum (39), (41), (43), we have

$$\left\|w^k\right\| \le \rho_2\left\|B^{k+1} - B^k\right\|, \tag{44}$$

where $w^k = \left(w^k_X, ..., w^k_Y, ..., w^k_Z\right) = \left(\partial^k_X \Psi, ..., \partial_{Y_1^k} \Psi, ..., \partial_{Z^k} \Psi\right) = \partial \Psi(X^k, Y^k, X^k) \in \partial \Psi(B^k)$.

Condition (iii). Summing (37) from $k = 0$ to $N - 1$ we have

$$\rho_1 \sum_{k}^{N-1} \left\| B^{k+1} - B^k \right\|^2 \leq \Psi(B^0) - \Psi(B^N) \tag{45}$$

Since $\left\{\Psi(B^N)\right\}$ is decreasing and $\inf \Psi > -\infty$, there exist some $\bar{\Psi}$ such that $\Psi(B^N) \to \bar{\Psi}$ as $N \to +\infty$. Therefore, let $N \to +\infty$ in (45), we have

$$\rho_1 \sum_{k}^{+\infty} \left\| B^{k+1} - B^k \right\|^2 \leq \Psi(B^0) - \bar{\Psi}, \tag{46}$$

which implies that $\lim \left\| B^k - B^{k-1} \right\| = 0$. Let $B^* = (X^*, Y^*, Z^*)$ be a limit point of $\left\{B^k\right\}_{k \in \mathbb{N}} = \left\{(X^k, Y^k, Z^k)\right\}_{k \in \mathbb{N}}$. Then (46) indicates that there is a subsequence $\left\{(X^{k_j}, Y^{k_j}, Z^{k_j})\right\}_{j \in \mathbb{N}}$ such that $(X^{k_j}, Y^{k_j}, Z^{k_j}) \to (X^*, Y^*, Z^*)$ as $j \to +\infty$.
From (27), we know

$$Y_i^{k+1} \in \arg\min \left\{ < Y - Y_i^k, \nabla_{Y_i} H\left(X^k, Y_i^k\right) > + \frac{\mu^k_{Y_i}}{2} \left\| Y - Y_i^k \right\|^2 + F(Y, Z^k) \right\} \tag{47}$$

Let $Y = Y_i^*$ the limiting point of $\left\{Y_i^k\right\}_{k \in \mathbb{N}}$, we have

$$\begin{aligned} &< Y_i^{k+1} - Y_i^k, \nabla_{Y_i} H\left(X^k, Y_i^k\right) > + \frac{\mu^k_{Y_i}}{2} \left\| Y_i^{k+1} - Y_i^k \right\|^2 + F(Y_i^{k+1}, Z^k) \\ &\leq < Y_i^* - Y_i^k, \nabla_{Y_i} H\left(X^k, Y_i^k\right) > + \frac{\mu^k_{Y_i}}{2} \left\| Y_i^* - Y_i^k \right\|^2 + F(Y_i^*, Z^k) \end{aligned} \tag{48}$$

Set $k = k_j - 1$, we obtain

$$\begin{aligned} &< Y_i^{k_j} - Y_i^{k_j - 1}, \nabla_{Y_i} H\left(X^{k_j - 1}, Y_i^{k_j - 1}\right) > + \frac{\mu^{k_j - 1}_{Y_i}}{2} \left\| Y_i^{k_j} - Y_i^{k_j - 1} \right\|^2 + F(Y_i^{k_j}, Z^{k_j - 1}) \\ &\leq < Y_i^* - Y_i^{k_j - 1}, \nabla_{Y_i} H\left(X^{k_j - 1}, Y_i^{k_j - 1}\right) > + \frac{\mu^{k_j - 1}_{Y_i}}{2} \left\| Y_i^* - Y_i^{k_j - 1} \right\|^2 + F(Y_i^*, Z^{k_j - 1}) \end{aligned} \tag{49}$$

Let $j \to +\infty$, we get

$$\lim_{j \to +\infty} \sup F(Y_i^{k_j}, Z^{k_j - 1}) \leq F(Y_i^*, Z^*) \tag{50}$$

From the fact that F is a PLS function, we also have

$$\lim_{j \to +\infty} \sup F(Y_i^{k_j}, Z^{k_j - 1}) \geq F(Y_i^*, Z^*) \tag{51}$$

Based on (50) and (51), we know $\lim_{j \to +\infty} = F(Y_i^*, Z^*)$. Arguing similarly with X, we finally have

$$\lim_{j \to +\infty} \Psi(X^{k_j}, Y^{k_j}, Z^{k_j}) = \lim_{j \to +\infty} H(X^{k_j}, Y^{k_j}) + F(Y^{k_j}, Z^{k_j}) = \Psi(X^*, Y^*, Z^*). \tag{52}$$

Condition (iv). The function Ψ is a semi-algebraic function, which automatically satisfies the Kurdyka-Lojasiewicz property [].

asdfasd

□

C GPALM Algorithm for NetREX-CF

In this section, we provide the detailed derivation for updating A, S, x_i, and y_i used in the GPALM algorithm for NetREX-CF.

C.1 Update A

$$A \in \arg\min \left\{ \left\| E - S^k A \right\|_F^2 + \lambda_A \|A\|_F^2 + \frac{\mu_A^k}{2} \left\| A - A^k \right\|_F^2 \right\} \tag{53}$$

$$= \arg\min \left\{ < E - S^k A, E - S^k A > + \lambda_A < A, A > + \frac{\mu_A^k}{2} < A - A^k, A - A^k > \right\} \tag{54}$$

$$= \arg\min \left\{ \mathrm{tr}(EE^T - 2E^T S^k A + S^k A A^T (S^k)^T) + \lambda_A \mathrm{tr}(A^T A) + \frac{\mu_A^k}{2} \mathrm{tr}(AA^T - 2A(A^k)^T + A^k (A^k)^T) \right\} \tag{55}$$

$$= \arg\min \left\{ [(S^k)^T S^k + \frac{2\lambda_A + \mu_A^k}{2} I] A A^T - 2[E^T S^k + \frac{\mu_A^k}{2} (A^k)^T] A \right\} \tag{56}$$

$$= \left((S^k)^T S^k + \frac{2\lambda_A + \mu_A^k}{2} I \right)^{-1} \left(E^T S^k + \frac{\mu_A^k}{2} (A^k)^T \right)^T \tag{57}$$

If we complete the square and disregard the constant we get

$$= \arg\min \left\{ \frac{\mu_A^k}{2} \mathrm{tr} \left[\left(A - (\frac{2}{\mu_A^k} E^T S^k + (A^k)^T)(\frac{2}{\mu_A^k} (S^k)^T S^k + I)^{-1} \right)^2 \right] \right\} \tag{58}$$

$$= \arg\min \left\{ \frac{\mu_A^k}{2} \left\| A - (\frac{2}{\mu_A^k} E^T S^k + (A^k)^T)(\frac{2}{\mu_A^k} (S^k)^T S^k + I)^{-1} \right\|_F^2 \right\} \tag{59}$$

$$= \left(\frac{2}{\mu_A^k} (S^k)^T S^k + I \right)^{-1} \left(\frac{2}{\mu_A^k} E^T S^k + (A^k)^T \right)^T \tag{60}$$

The derivative $\nabla_A H \left(A^k, S^k \right)$ can be computed by $2((S^k)^T S^k A^k + \lambda_A A^k - (S^k)^T E)$, which is Lipschitz continuous with Lipschitz constant $\mu_A^k = \left\| (S^k)^T S^k + \lambda_A I \right\|_F$.

C.2 Update S

$$S_{ij}^{k+1} \in \arg\min \left\{ < S_{ij} - S_{ij}^k, \nabla_{S_{ij}} H \left(A^{k+1}, S_{ij}^k \right) > + \frac{\mu_{S_{ij}}^k}{2} \left\| S_{ij} - S_{ij}^k \right\|^2 + F(S_{ij}, X^k, Y^k) \right\} \tag{61}$$

$$= \arg\min \left\{ (S_{ij} - S_{ij}^k) \nabla_{S_{ij}} H \left(A^{k+1}, S_{ij}^k \right) + \frac{\mu_{S_{ij}}^k}{2} (S_{ij} - S_{ij}^k)^2 + F(S_{ij}, X^k, Y^k) \right\} \tag{62}$$

$$= \arg\min \left\{ \frac{\mu_{S_{ij}}^k}{2} \left(S_{ij} - \left(S_{ij}^k - \frac{1}{\mu_{S_{ij}}^k} \nabla_{S_{ij}} H \left(A^{k+1}, S_{ij}^k \right) \right) \right)^2 + F(S_{ij}, X^k, Y^k) \right\} \tag{63}$$

The last equation comes from completing the square and disregarding the constant terms. Now set $U_{ij}^k = S_{ij}^k - \frac{1}{\mu_{S_{ij}}^k} \nabla_{S_{ij}} H\left(A^{k+1}, S_{ij}^k\right)$ and we have:

$$\arg\min \left\{ \frac{\mu_{S_{ij}}^k}{2} \left(S_{ij} - U_{ij}^k\right)^2 + F(S_{ij}, X^k, Y^k) \right\} \tag{64}$$

Put $F(S, X^k, Y^k)$ in and after some algebra.

$$\arg\min \left\{ \left(S_{ij} - U_{ij}^k\right)^2 + \frac{2}{\mu_{S_{ij}}^k} \left\{\lambda \left[\bar{C}_{ij}(1 - B_{ij})(1 + 2(B_{ij} - x_i^T y_j)) + (\bar{C}_{ij} - C_{ij})(B_{ij} - x_i^T y_j)^2\right] + \ \eta_{ij}\right\} \|S_{ij}\|_0 \right\} \tag{65}$$

By setting $c_{ij} = \sqrt{\frac{2}{\mu_{S_{ij}}^k} \left\{\lambda \left[\bar{C}_{ij}(1 - B_{ij})(1 + 2(P_{ij} - x_i^T y_j)) + (\bar{C}_{ij} - C_{ij})(B_{ij} - x_i^T y_j)^2\right] + \ \eta_{ij}\right\}}$, we obtain the hard thresholding problem:

$$\arg\min \left\{ \left(S_{ij} - U_{ij}^k\right)^2 + c_{ij}^2 \|S_{ij}\|_0 \right\} \tag{66}$$

which has solution

$$S_{ij}^{k+1} = \begin{cases} U_{ij}^k, & \text{if } \left|U_{ij}^k\right| > c_{ij}; \\ \{0, c_{ij}\}, & \text{if } \left|U_{ij}^k\right| = c_{ij}; \\ 0, & o.w.. \end{cases} \tag{67}$$

The derivative $\nabla_S H\left(A^{k+1}, S^k\right)$ can be computed by $2(S^k A^{k+1}(A^{k+1})^T + \lambda_S S^k - E(A^{k+1})^T)$, which is Lipschitz continuous with Lipschitz constant $\mu_S^k = \left\|A^{k+1}(A^{k+1})^T + \lambda_S I\right\|_F$.

C.3 Update X

$$x_i \in \arg\min \left\{ \sum_{i,j} \lambda(\bar{C}_{ij} + (\bar{C}_{ij} - C_{ij}) \|S_{ij}\|_0) \left(\|S_{ij}\|_0 + (1 - \|S_{ij}\|_0)B_{ij} - x_i^T y_j\right)^2 + \frac{\mu_x^k}{2} \left\|x_i - x_i^k\right\|^2 \right\} \tag{68}$$

Let $\bar{A}_{ij} = \lambda\left(\bar{C}_{ij} + (\bar{C}_{ij} - C_{ij}) \|S_{ij}\|_0\right)$ and $\widetilde{A}$ be the diagonal matrix with the values $\bar{A}_{i1}, \bar{A}_{i2}, ..\bar{A}_{im}$ on the diagonal. Let $\bar{S}_i = \left[\left\|S_{i1}^{k+1}\right\|_0 + (1 - \left\|S_{i1}^{k+1}\right\|_0)B_{i1}, .., \left\|S_{in}^{k+1}\right\|_0 + (1 - \left\|S_{im}^{k+1}\right\|_0)B_{im}\right]$. We will show that

$$\sum_{i,j} \bar{A}_{ij} \left(\left\|S_{ij}^{k+1}\right\|_0 - x_i^T y_j^k\right)^2 = \left(\bar{S}_i - x_i^T Y^k\right) \widetilde{A} \left(\bar{S}_i - x_i^T Y^k\right)^T \tag{69}$$

We have

$$\left(\bar{S}_i - x_i^T Y^k\right) \widetilde{A} \left(\bar{S}_i - x_i^T Y^k\right)^T \tag{70}$$

$$= \left(\left[\left\| S_{i1}^{k+1} \right\|_0 + (1 - \left\| S_{i1}^{k+1} \right\|_0) B_{i1}, .., \left\| S_{in}^{k+1} \right\|_0 + (1 - \left\| S_{im}^{k+1} \right\|_0) B_{im} \right] - {x_i}^T \begin{bmatrix} | & & | \\ y_1^k & \dots & y_n^k \\ | & & | \end{bmatrix} \right) \begin{bmatrix} \bar{A}_{i1} & & \\ & \ddots & \\ & & \bar{A}_{im} \end{bmatrix} \tag{71}$$

$$\left(\left[\left\| S_{i1}^{k+1} \right\|_0 + (1 - \left\| S_{i1}^{k+1} \right\|_0) B_{i1}, .., \left\| S_{in}^{k+1} \right\|_0 + (1 - \left\| S_{im}^{k+1} \right\|_0) B_{im} \right] - {x_i}^T \begin{bmatrix} | & & | \\ y_1^k & \dots & y_n^k \\ | & & | \end{bmatrix} \right)^T \tag{72}$$

$$= \bar{A}_{i1} \left(\left\| S_{i1}^{k+1} \right\|_0 - {x_i}^T y_1^k \right)^2 + ... + \bar{A}_{in} \left(\left\| S_{in}^{k+1} \right\|_0 - {x_i}^T y_n^k \right)^2 \tag{73}$$

$$= \sum_{i,j} \bar{A}_{ij} \left(\left\| S_{ij}^{k+1} \right\|_0 - x_i^T y_j^k \right)^2 \tag{74}$$

We can now rewrite the problem in terms of the matrix formulation:

$$x_i \in \arg\min \left\{ \left(\bar{S}_i - x_i^T Y^k\right) \widetilde{A} \left(\bar{S}_i - x_i^T Y^k\right)^T + \eta_i \bar{S}_i + \frac{\mu_x^k}{2} \left(x_i - x_i^k\right) \left(x_i - x_i^k\right)^T \right\} \tag{75}$$

$$= \arg\min \left\{ \left(\bar{S}_i - x_i^T Y^k\right) \widetilde{A} \left(\bar{S}_i - x_i^T Y^k\right)^T + \eta_i \bar{S}_i + \frac{\mu_x^k}{2} x_i^T x_i - \mu_x^k x_i^T x_i^k + \frac{\mu_x^k}{2} {x_i^k}^T x_i^k \right\} \tag{76}$$

From here we simplify the problem by expanding the first term and removing the constant terms.

$$= \arg\min \left\{ \left(\bar{S}_i \widetilde{A} - x_i^T Y^k \widetilde{A}\right) \left(\bar{S}_i^T - {Y^k}^T x_i\right) + \frac{\mu_x^k}{2} x_i^T x_i - \mu_x^k x_i^T x_i^k \right\} \tag{77}$$

$$= \arg\min \left\{ \bar{S}_i \widetilde{A} \bar{S}_i^T - \bar{S}_i \widetilde{A} {Y^k}^T x_i - x_i^T Y^k \widetilde{A} \bar{S}_i^T + x_i^T Y^k \widetilde{A} {Y^k}^T x_i + \frac{\mu_x^k}{2} x_i^T x_i - \mu_x^k x_i^T x_i^k \right\} \tag{78}$$

The first term is a constant so we ignore it. Because the third term is a number it is equivalent to its transpose, so the second and third term are the same. We also note that $[Y^k \widetilde{A} {Y^k}^T]$ has dimension $h \times h$, and the second to last term may be rewritten $x_i^T [\frac{\mu_x^k}{2} I_{h \times h}] x_i$.

$$= \arg\min \left\{ -2 \bar{S}_i \widetilde{A} {Y^k}^T x_i + x_i^T [Y^k \widetilde{A} {Y^k}^T] x_i + x_i^T [\frac{\mu_x^k}{2} I_{h \times h}] x_i - \mu_x^k x_i^T x_i^k \right\} \tag{79}$$

$$= \arg\min \left\{ -2 \bar{S}_i \widetilde{A} {Y^k}^T x_i - \mu_x^k x_i^T x_i^k + x_i^T [\frac{\mu_x^k}{2} I_{h \times h} + Y^k \widetilde{A} {Y^k}^T] x_i \right\} \tag{80}$$

We now use the fact that $\mu_x^k x_i^T x_i^k$ is a number and it is equivalent to its transpose,

$$= \arg\min \left\{ -(2 \bar{S}_i \widetilde{A} {Y^k}^T + \mu_x^k {x_i^k}^T) x_i + x_i^T [\frac{\mu_x^k}{2} I_{h \times h} + Y^k \widetilde{A} {Y^k}^T] x_i \right\} \tag{81}$$

Finally, we define $\phi = \frac{\mu_x^k}{2} I_{h\times h} + Y^k \widetilde{A} {Y^k}^T$ and $\varphi = 2\bar{S}_i \widetilde{A} {Y^k}^T + \mu_x^k {x_i^k}^T$, and the problem becomes a Quadratically Constrained Quadratic Program (QCQP) of the form

$$\arg\min \left\{ x_i^T \phi x_i - \varphi x_i \right\} \tag{82}$$

$$s.t. \left\| x_i \right\|^2 \leq a \tag{83}$$

Since we know this problem can be solved, we leave the rest to the CVXPY python package. Lastly, for $Q = (\bar{S}_i - x_i^T Y^k)\widetilde{A}(\bar{S}_i - x_i^T Y^k)^T$ we find the partial gradient

$$\nabla_{x_i} Q = (2\bar{S}_i \widetilde{A}(Y^k)^T)^T + Y^k \widetilde{A}(Y^k)^T x_i + (Y^k \widetilde{A}(Y^k)^T)^T x_i \tag{84}$$

$$= (2\bar{S}_i \widetilde{A}(Y^k)^T)^T + 2Y^k \widetilde{A}(Y^k)^T x_i \tag{85}$$

which is Lipschitz continuous with Lipschitz constant $\mu_x^k = \left\| 2Y^k \widetilde{A}(Y^k)^T \right\|_F$.

C.4 Solution for Y

Now, for Y, let $\bar{A}_{ij}$ and $\widetilde{A}$ be the same, but $\widetilde{A}$ will have the values $\bar{A}_{1j}, \bar{A}_{2j}, ..\bar{A}_{nj}$ on the diagonal. Let X^{k+1} be a $h \times n$ dimensional matrix where the columns are composed of the vectors $x_1^{k+1}, x_2^{k+1}, .., x_n^{k+1}$ and $\bar{S}_j = \left[\left\| S_{1j}^{k+1} \right\|_0 + (1 - \left\| S_{1j}^{k+1} \right\|_0) B_{1j}, .., \left\| S_{nj}^{k+1} \right\|_0 + (1 - \left\| S_{nj}^{k+1} \right\|_0) B_{nj} \right]^T$. We show that the matrix formulation for this problem by proving the following:

$$\sum_{i,j} \bar{A}_{ij} \left(\left\| S_{ij}^{k+1} \right\|_0 - {x_i^{k+1}}^T y_j \right)^2 = \left(\bar{S}_j - {X^{k+1}}^T y_j \right)^T \widetilde{A} \left(\bar{S}_j - {X^{k+1}}^T y_j \right) \tag{86}$$

We have

$$\left(\bar{S}_j - {X^{k+1}}^T y_j \right)^T \widetilde{A} \left(\bar{S}_j - {X^{k+1}}^T y_j \right) \tag{87}$$

$$= \left(\begin{bmatrix} \left\| S_{1j}^{k+1} \right\|_0 + (1 - \left\| S_{1j}^{k+1} \right\|_0) B_{1j} \\ \vdots \\ \left\| S_{nj}^{k+1} \right\|_0 + (1 - \left\| S_{nj}^{k+1} \right\|_0) B_{nj} \end{bmatrix} - \begin{bmatrix} | & & | \\ x_1^{k+1} & \dots & x_m^{k+1} \\ | & & | \end{bmatrix}^T y_i \right)^T \begin{bmatrix} \bar{A}_{1j} & & \\ & \ddots & \\ & & \bar{A}_{nj} \end{bmatrix} \tag{88}$$

$$\left(\begin{bmatrix} \left\| S_{1j}^{k+1} \right\|_0 + (1 - \left\| S_{1j}^{k+1} \right\|_0) B_{1j} \\ \vdots \\ \left\| S_{nj}^{k+1} \right\|_0 + (1 - \left\| S_{nj}^{k+1} \right\|_0) B_{nj} \end{bmatrix} - \begin{bmatrix} | & & | \\ x_1^{k+1} & \dots & x_m^{k+1} \\ | & & | \end{bmatrix}^T y_i \right) \tag{89}$$

$$= \sum_{i,j} \bar{A}_{ij} \left(\left\| S_{ij}^{k+1} \right\|_0 - {x_i^{k+1}}^T y_j \right)^2 \tag{90}$$

We follow the same procedure to solve this problem

$$y_j \in \arg\min \left\{ \left(\bar{S}_j - X^{k+1^T} y_j\right)^T \widetilde{A} \left(\bar{S}_j - X^{k+1^T} y_j\right) + \eta_j \bar{S}_j + \frac{\mu_y^k}{2}\left(y_j - y_j^k\right)\left(y_j - y_j^k\right)^T \right\} \tag{91}$$

$$= \arg\min \left\{ \left(\bar{S}_j - X^{k+1^T} y_j\right)^T \widetilde{A} \left(\bar{S}_j - X^{k+1^T} y_j\right) + \eta_j \bar{S}_j + \frac{\mu_y^k}{2} {y_j}^T y_j - \mu_y^k {y_j}^T y_j^k \right\} \tag{92}$$

$$= \arg\min \left\{ \left(\bar{S}_j^T - {y_j}^T X^{k+1}\right)\left(\widetilde{A}\bar{S}_j - \widetilde{A} X^{k+1^T} y_j\right) + \eta_j \bar{S}_j + \frac{\mu_y^k}{2} {y_j}^T y_j - \mu_y^k {y_j}^T y_j^k \right\} \tag{93}$$

$$= \arg\min \left\{ \bar{S}_j^T \widetilde{A} \bar{S}_j - \bar{S}_j^T \widetilde{A} X^{k+1^T} y_j - {y_j}^T X^{k+1} \widetilde{A} \bar{S}_j + {y_j}^T X^{k+1} \widetilde{A} X^{k+1^T} y_j + \eta_j \bar{S}_j + \frac{\mu_y^k}{2} {y_j}^T y_j - \mu_y^k {y_j}^T y_j^k \right\} \tag{94}$$

Disregard constants

$$= \arg\min \left\{ -\bar{S}_j^T \widetilde{A} X^{k+1^T} y_j - {y_j}^T X^{k+1} \widetilde{A} \bar{S}_j + {y_j}^T X^{k+1} \widetilde{A} X^{k+1^T} y_j + \frac{\mu_y^k}{2} {y_j}^T y_j - \mu_y^k {y_j}^T y_j^k \right\} \tag{95}$$

Once again, we can take the transpose of the second term and last term,

$$= \arg\min \left\{ -2\bar{S}_j^T \widetilde{A} X^{k+1^T} y_j + {y_j}^T X^{k+1} \widetilde{A} X^{k+1^T} y_j + \frac{\mu_y^k}{2} {y_j}^T y_j - \mu_y^k {y_j}^T y_j^k \right\} \tag{96}$$

$$= \arg\min \left\{ -(2\bar{S}_j^T \widetilde{A} X^{k+1^T} + \mu_y^k {y_j^k}^T) y_j + {y_j}^T [X^{k+1} \widetilde{A} X^{k+1^T}] y_j + {y_j}^T [\frac{\mu_y^k}{2} I_{h \times h}] y_j \right\} \tag{97}$$

$$= \arg\min \left\{ -(2\bar{S}_j^T \widetilde{A} X^{k+1^T} + \mu_y^k {y_j^k}^T) y_j + {y_j}^T [X^{k+1} \widetilde{A} X^{k+1^T} + \frac{\mu_y^k}{2} I_{h \times h}] y_j \right\} \tag{98}$$

Letting $\phi = X^{k+1} \widetilde{A} X^{k+1^T} + \frac{\mu_y^k}{2} I_{h \times h}$ and $\varphi = 2\bar{S}_j^T \widetilde{A} X^{k+1^T} + \mu_y^k {y_j^k}^T$ gives us the QCQP

$$\arg\min \left\{ y_j^T \phi y_j - \varphi y_j \right\} \tag{99}$$

$$s.t. \left\|y_j\right\|^2 \leq b \tag{100}$$

which we solve in CVXPY using the same function. To finish the problem we take $Q = (\bar{S}_j - X^{k+1^T} y_j)^T \widetilde{A} (\bar{S}_j - X^{k+1^T} y_j)$ and find the partial gradient

$$\nabla_{y_j} Q = (-2\bar{S}_j \widetilde{A} (X^{k+1})^T)^T + X^{k+1} \widetilde{A} (X^{k+1})^T y_j + (X^{k+1} \widetilde{A} (X^{k+1})^T)^T y_j \tag{101}$$

$$= (-2\bar{S}_j \widetilde{A} (X^{k+1})^T)^T + 2X^{k+1} \widetilde{A} (X^{k+1})^T y_j \tag{102}$$

which is Lipschitz continuous with Lipschitz constant $\mu_y^k = \left\| 2X^{k+1} \widetilde{A} (X^{k+1})^T \right\|_F$.

D Parameter Selection

In this section, we introduce how we select parameters for the competing algorithms.

D.1 Paramter Selection for PriroSum

PriorSum constructs a predicted GRN by summing over weights from all prior networks $P = \{P^1, ..., P^d\}$. Therefore, PriorSum builds a GRN $\mathfrak{P}_{ij} = \sum_k P^k_{ij}$ and does not need to select any parameters.

D.2 Parameter Selection for LassoStARS

LassoStARS [4] is the latest version of Inferelator, it takes an unweighted prior and gene expression data as input. Because LassoStARS needs an unweighted prior network and the prior networks we have are weighted prior networks, we choose different cutoffs to construct prior networks for LassoStARS. We generate prior networks by assigning each gene the top N TFs based on the $\mathfrak{P}_{ij}$. For N, we set $N = \{10, 20, 30, 40\}$ and we find that $N = 10$ performs the best and report the results in Fig. 2. For other parameters used in LassoStARS, LassoStARS proposed a way to select the optimal parameters, therefore, we do not need to select other parameters.

D.3 Parameter Selection for MerlinP

For reconstructing the GRN for yeast, MerlinP [5] use the same prior networks and gene expression to build a GRN and reported in the repository `https://github.com/Roy-lab/merlin-p`. We directly download the GRN they build and compared it with other methods.

D.4 Parameter Selection for NetREX

NetREX [2] is similar to LassoStARS, taking an unweighted prior and gene expression as input. So similarly, we generate prior networks for NetREX by assigning each gene the top N TFs based on the $\mathfrak{P}_{ij}$. We set $N = \{10, 20, 30, 40\}$ and we find that $N = 20$ performs the best and report the results in Fig. 2. For the other parameters, we selected based on the suggestion provided in `https://github.com/ncbi/NetREX`.

D.5 Parameter Selection for CF

We input CF [16] with $\mathfrak{P}_{ij} = \sum_k P^k_{ij}$. The dimension of the hidden feature vector we set it to be 100, 200, and 300. The regulation term used by CF is set to be 0.1, 1, 10, 100. We try all those combination and report the result with the best performance.

D.6 Parameter Selection for NetREX-CF

Based on the formulaiton of NetREX-CF (8), we know that we need to select h, λ_A, λ_S, η_{ij}, λ, and $\bar{C}_{ij}$. h is the dimension of the hidden feature vector. We find that $h = \{100, 200, 300\}$ does not change the performance much. For computational consideration, we set $h = 100$. Because λ_A and λ_S are used as standard regulation to avoid over-fitting, we set $\lambda_A = 1.0$ and $\lambda_S = 1.0$ by default. We introduce the selection of η_{ij} and $\bar{C}_{ij}$ in the following subsection.

Selection of $\boldsymbol{\eta_{ij}}$ We need to make sure $F(S,X,Y)$ is lower semi-continuous. We can first simplify the equation into

$$\begin{aligned} F(S,X,Y) &= \lambda \left[\sum_{i,j} \Omega_{ij} \left(\|S_{ij}\|_0 + (1-\|S_{ij}\|_0) B_{ij} - x_i^T y_j \right)^2 \right] + \sum_{i,j} \eta_{ij} \|S_{ij}\|_0 \\ &= \lambda \left[\sum_{i,j} (C_{ij} + (\bar{C}_{ij} - C_{ij}) \|S_{ij}\|_0) \left(\|S_{ij}\|_0 + (1-\|S_{ij}\|_0) B_{ij} - x_i^T y_j \right)^2 \right] + \sum_{i,j} \eta_{ij} \|S_{ij}\|_0 \\ &= \sum_{i,j} \left\{ \lambda \left[\bar{C}_{ij}(1-B_{ij})(1+2(B_{ij}-x_i^T y_j)) + (\bar{C}_{ij} - C_{ij})(B_{ij} - x_i^T y_j)^2 \right] + \ \eta_{ij} \right\} \|S_{ij}\|_0 \\ &\quad + \sum_{ij} C_{ij} (B_{ij} - x_i^T y_j)^2 \end{aligned} \tag{103}$$

$F(S,X,Y)$ is lower semi-continuous when the parameter before $\|S_{ij}\|_0$ in the above equation is larger than 0. After several manipulation, we find out we need to set η_{ij} as following to make $F(S,X,Y$ lower semi-continuous.

$$\eta_{ij} = \begin{cases} \geq 0, & B_{ij} = 1, \\ \geq \lambda \frac{C_{ij}\bar{C}_{ij}}{\bar{C}_{ij} - C_{ij}}, & B_{ij} = 0. \end{cases} \tag{104}$$

Selection of $\boldsymbol{\bar{C}_{ij}}$ C_{ij} is the penalty when we want to use $x_i^T y_j$ to learn $B_{ij} = 1$. Similarly, $\bar{C}_{ij}$ is the penalty when we want to use $x_i^T y_j$ to learn $\|S_{ij}\|_0 = 1$. There are two siutations. First, when $\|S_{ij}\|_0 = 1$ and $B_{ij} = 1$, meaning the sparse NCA-based method confirms the edge in the prior, then intuitively, we need to set $\bar{C}_{ij} = \alpha C_{ij}, \alpha \geq 1$. Another situation is that $\|S_{ij}\|_0 = 1$ and $B_{ij} = 0$, meaning the sparse NCA-based model confirms an edges recommended by the CF model but not appeared in the prior networks. For this case, we set $\bar{C}_{ij} \in [C_{ij}, \max(C)]$, where $\max(C)$ is the largest element in penalty matrix C. In sum, $\bar{C}_{ij} = \alpha C_{ij}\|S_{ij}\|_0 B_{ij} + \beta \|S_{ij}\|_0 (1-B_{ij})$, where $\alpha \geq 1$ and $\beta \in [C_{ij}, \max(C)]$.

Consensus of Different Parameter Selections As explained in the previous, for η_{ij} and $\bar{C}_{ij}$, we know the range of these parameters but do not know the exact optimal values. For reconstructing GRN for the yeast experiment, we set

$$\eta_{ij} = \begin{cases} \geq \theta, & B_{ij} = 1, \\ \geq \lambda \frac{C_{ij}\bar{C}_{ij}}{\bar{C}_{ij} - C_{ij}} + \theta, & B_{ij} = 0, \end{cases} \tag{105}$$

where $\theta = \{0.1, 0.5, 1, 2\}$. And $\bar{C}_{ij} = \alpha C_{ij}\|S_{ij}\|_0 B_{ij} + \beta \|S_{ij}\|_0 (1-B_{ij})$, where $\alpha = \{1, 2, 3, 10\}$ and $\beta = 10, 20, 30, 40$. For different set of parameters, we get a GRN and we get a set of GRNs $\mathfrak{G} = \{G^1, \ldots\}$, where $G^i = X^T Y$ after applying all theses parameters. The final perdition is the average overall predictions $G^* = \dfrac{\sum_i G^i}{|\mathfrak{G}|}$.