FAST CONVEX OPTIMIZATION FOR CONNECTIVITY ENFORCEMENT IN GENE REGULATORY NETWORK INFERENCE

Aurélie Pirayre[1,2]*, Camille Couprie*[1]*, Laurent Duval*[1]*, and Jean-Christophe Pesquet*[2]

[1]IFP Energies nouvelles, Mechatronics, Computer Science and Applied Mathematics Division
[2]Univ. Paris-Est, LIGM, UMR CNRS 8049

ABSTRACT

With the advent of microarrays, arose the need to analyze gene expression data. Tools for building gene regulation networks are indeed of high interest for regulatory relationship sketching and gene interaction prediction. Given all pairwise gene regulation information available, we propose to determine the presence of edges in the final gene regulatory network by adopting a convex optimization formulation. Our energy minimization strategy includes a regularization term accounting for the difference of connectivity of particular genes (i.e. transcription factors), and we employ proximal methods to compute the optimal solution. The resulting algorithm, called "Brane relax", outperforms state-of-the-art methods while keeping a reduced computational cost.

***Index Terms*—** Bioinformatics, Genetic expression, Graphs, Optimization, Proximity operator

1. INTRODUCTION

The ability to extract plausible gene regulatory relationships is of paramount importance for improving the knowledge of living organism mechanisms. It benefits medical applications (identification of genes involved in diseases such as cancer) as well as biotechnologies (study of micro-organisms involved in biofuel production), among others. For this purpose, biologists acquire high-throughput data: for each gene, short signals related to the gene expression are obtained. Unfortunately, recovering insightful information from this set of signals becomes particularly challenging given their variability and their often incomplete and noisy nature. An additional difficulty arises when the number of observation is critically lower than the number of genes, often exceeding several thousands. Therefore, the construction of Gene Regulation Networks (GRNs) constitutes a fundamental problem that can be addressed from a graph signal processing perspective. A recent survey of state-of-the-art approaches is provided in [1]. Standard GRN inference methods define a score of similarity expression between pairs of genes to weight the edges of the network. The large majority of existing approaches reformulate the GRN inference reconstruction as a detection problem where thresholds serve to select edge subsets: only edges scoring larger than the threshold are kept in the final graph. This is typically the case for the Context Likelihood of Relatedness (CLR) approach [2], which computes a combination of Z-scores on the mutual information between gene expression profiles. CLR is widely used in biostatistics thanks to its interesting trade-off between a satisfactory prediction accuracy and a relatively low complexity. Recently, performance improvements were attained by the random-forest-based GENIE3 method [3], computing a ranking of the plausible edge presence in the final graph.

One of the limitations of the aforementioned methods is however that they weakly account for prior knowledge about the expected graph structure. Indeed, biological systems raise tremendous challenges in knowledge inference. Their representations as complex networks [4] entail an infinite number of potential structures and solutions, given the available approximate models and the reduced number of observations. Consequently, complementing networks with putative structural properties [5] is pervading biological system analysis. By incorporating such a knowledge in the GRN inference process, one can expect it to become more reliable, more robust to uncertain data, and more meaningful from a biological standpoint. Co-expression transitivity is proposed in [6] for genes with dissimilar expression profiles. Biological structures are probed with perturbations in [7]. Such approaches, constraining forms of sparsity in the inferred networks [8], often imply well-chosen measures such as a thresholded Hamming similarity [9] or a Frobenius norm between transition matrices [10].

The proposed approach also resorts to judicious choices of penalty functions, linking different sets of gene functions. Actually, genes that are *not* known to encode a transcription factor (TF), i.e. which are not identified as gene expression regulators, typically interact with less genes than others. In order to enforce a low connectivity degree to these particular genes, we introduce a novel convex formulation for GRN construction, encompassing this structural information.

The paper is organized as follows: In Section 2, we propose a novel variational approach for building GRNs, and we design an efficient algorithm to solve the related optimization problem in Section 3. Section 4 presents comparative results on a standard dataset coming from DREAM4 challenge [1], demonstrating substantial improvements over conventional approaches.

2. VARIATIONAL MODEL

2.1. Original problem formulation

Inferring a gene regulatory network $\mathcal{G}^*$ aims at selecting, among all possible edge combinations, a set of edges $\mathcal{E}^*$ reflecting the regulatory process between genes. Let g denote the number of genes, and let $x_{i,j}$ define the binary

label of the edge $e_{i,j}$. Specifically, $x_{i,j} = 1$ if an interaction exists between gene i and gene j, where $(i,j) \in \mathbb{E}$ with $\mathbb{E} = \{(i,j) \in \{1,\dots,g\}^2 \mid i \neq j\}$, and $x_{i,j} = 0$ otherwise. Our goal is to design a cost function whose minimum value is reached for the desired value of vector $\mathbf{x} = (x_{i,j})_{(i,j)\in\mathbb{E}} \in \{0,1\}^{|\mathbb{E}|}$ with $|\mathbb{E}| = g(g-1)$. Such an optimal solution is expected to lead to the desired regulatory network.

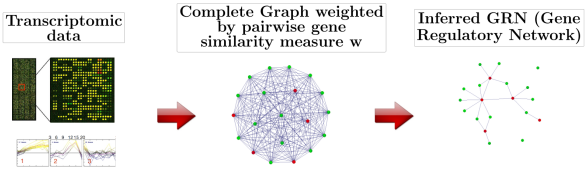

Fig. 1. Our goal is to infer a GRN from transcriptomic data. In this illustration, transcription factors are represented in red, other genes in green.

Our data thus correspond to a fully connected undirected non-reflexive graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ consisting of a set of nodes $\mathcal{V} = (v_1,\cdots,v_g)$, a set of $g(g-1)$ edges $\mathcal{E} = (e_{i,j})_{(i,j)\in\mathbb{E}}$ weighted by real values stored in a vector $\mathbf{w} = (w_{i,j})_{(i,j)\in\mathbb{E}}$ of dimension $g(g-1)$. Typically, the weight value $w_{i,j}$ of an edge $e_{i,j}$ with $(i,j) \in \mathbb{E}$ is a function of a correlation or mutual information measure between expressions of genes i and j. Fig. 1 illustrates the aforementioned graph processing task. Our model takes the following actions:

- Favor the selection of real regulatory relations based on the gene expression similarity $(w_{i,j})_{(i,j)\in\mathbb{E}}$.
- Promote the selection of edges linked to a putative transcription factor, by defining weights $(\lambda_{i,j})_{(i,j)\in\mathbb{E}}$ depending on the nature of nodes i and j.
- Assuming that a target gene is regulated by a small number of transcription factors, constrain the connectivity of the 'nonTFs' to be close to a given small number d.

It can be translated into the following optimization problem:

$$\underset{\mathbf{x}\in\mathbb{S}}{\text{minimize}} \underbrace{\sum_{(i,j)\in\mathbb{E}} \frac{w_{i,j}}{2}(1-x_{i,j})}_{\text{favors the presence of edges of strong weight}} + \underbrace{\sum_{(i,j)\in\mathbb{E}} \frac{\lambda_{i,j}}{2} x_{i,j}}_{\text{favors the presence of TF-nonTF edges}} + \mu \underbrace{\sum_{i\in\mathcal{V}\setminus\mathcal{T}} \phi\left(\sum_{j=1}^{g} x_{i,j} - d\right)}_{\text{enforces the degree of a subset of nodes being close to } d}, \quad (1)$$

where $\mu \in [0,+\infty[$ is a regularization constant, $\phi\colon \mathbb{R} \to \mathbb{R}$ is a convex function, $\mathcal{T} \subseteq \mathcal{V}$ denotes the set of putative transcription factors and

$$\mathbb{S} = \big\{(x_{i,j})_{(i,j)\in\mathbb{E}} \in \{0,1\}^{|\mathbb{E}|} \mid \big(\forall (i,j) \in \mathbb{E}\big)\ \ x_{i,j} = x_{j,i}\big\}. \quad (2)$$

The latter constraint set serves to express both the Boolean constraint and the fact that the graph is undirected. For this reason too, the following symmetry properties may be assumed to be met by vectors $\mathbf{w}$ and $\boldsymbol{\lambda} = (\lambda_{i,j})_{(i,j)\in\mathbb{E}}$:

$$\big(\forall (i,j) \in \mathbb{E}\big) \qquad w_{i,j} = w_{j,i},\ \ \lambda_{i,j} = \lambda_{j,i}. \quad (3)$$

2.2. Problem relaxation and vectorial formulation

Since the cost function of Problem (1) is not necessarily submodular, it is not amenable to optimization *via* efficient combinatorial optimization methods, e.g. GraphCut-based methods [11]. To overcome this difficulty, we relax the integrality constraint on $\mathbf{x}$, by replacing $\mathbb{S}$ by its convex hull:

$$\widehat{\mathbb{S}} = \big\{(x_{i,j})_{(i,j)\in\mathbb{E}} \in [0,1]^{|\mathbb{E}|} \mid \big(\forall (i,j) \in \mathbb{E}\big)\ \ x_{i,j} = x_{j,i}\big\}. \quad (4)$$

The relaxed optimization problem then becomes solvable in an efficient manner by using convex optimization methods. It can be reexpressed more concisely by reindexing the variables on the edges with a single index l, and taking into account explicitly the symmetry constraint, so yielding

$$\underset{\boldsymbol{x}\in[0,1]^L}{\text{minimize}} \sum_{l=1}^{L} \Big(w_l(1-x_l) + \lambda_l x_l\Big) + \mu \sum_{i=1}^{P} \phi\Big(\sum_{k=1}^{L} \Omega_{i,k} x_k - d\Big), \quad (5)$$

where $L = g(g-1)/2$ and P is the cardinality of $\mathcal{V} \setminus \mathcal{T}$. In an equivalent vector form, the optimization problem can be reexpressed as

$$\underset{\boldsymbol{x}\in\mathbb{R}^L}{\text{minimize}}\ \ f_1(\boldsymbol{x}) + f_2(\boldsymbol{x}), \quad (6)$$

where f_1 is the indicator function of the unit hypercube:

$$(\forall \boldsymbol{x} \in \mathbb{R}^L)\ f_1(\boldsymbol{x}) = \iota_{[0,1]^L}(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \boldsymbol{x} \in [0,1]^L \\ +\infty & \text{otherwise,} \end{cases}$$

and

$$(\forall \boldsymbol{x} \in \mathbb{R}^L)\ f_2(\boldsymbol{x}) = \mathbf{w}^\top(\mathbb{1}_L - \boldsymbol{x}) + \boldsymbol{\lambda}^\top \boldsymbol{x} + \mu\Phi(\boldsymbol{\Omega}\boldsymbol{x} - \mathbf{d}),$$

with $\mathbb{1}_L = [1,\dots,1]^\top \in \mathbb{R}^L$. Hereabove, Φ is the separable function defined as $\Phi\colon \mathbb{R}^P \to \mathbb{R}\colon (y_i)_{1\le i\le P} \mapsto \sum_{i=1}^{P} \phi(y_i)$, $\mathbf{d} = d\,\mathbb{1}_P$, and $\boldsymbol{\Omega} = (\Omega_{i,k})_{1\le i\le P, 1\le k\le L}$ is a binary matrix of size $P \times L$ such that, for every $i \in \{1,\dots,P\}$ and $k \in \{1,\dots,L\}$, $\Omega_{i,k} = 1$ if k corresponds to the index of an edge linking i to some vertex of the graph.

3. OPTIMIZATION STRATEGY

In this part, we show that it is possible to derive a fast algorithm for solving the optimization problem formulated in the previous section (when $\mu \neq 0$). Our developments will be grounded on recent results concerning the Forward-Backward algorithm, and more specifically the use of variable metrics derived from the Majorize-Minimize principle [12], combined with a block coordinate descent strategy [13]. Tutorial introductions to proximal optimization may be found in [14, 15].

3.1. Majorant construction

Assuming that the function ϕ is differentiable and has a β-Lipschitzian gradient with a Lipschitz constant $\beta \in]0, +\infty[$, Problem (6) can be solved with the help of a Forward-Backward algorithm. This first-order method is however known to be pretty slow and our aim will be to provide an accelerated version of it. For this purpose, we will apply the Majorization-Minimization principle to f_2 by building a quadratic majorant of this smooth function. From the descent lemma [16, Theorem 18.15(iii)], for every $(\boldsymbol{x}, \boldsymbol{x}') \in (\mathbb{R}^L)^2$,

$$f_2(\boldsymbol{x}) \leq f_2(\boldsymbol{x}') + (\boldsymbol{x} - \boldsymbol{x}')^\top \nabla f_2(\boldsymbol{x}') + \frac{\mu\beta}{2}(\boldsymbol{x} - \boldsymbol{x}')^\top \boldsymbol{\Omega}^\top \boldsymbol{\Omega}(\boldsymbol{x} - \boldsymbol{x}') \quad (7)$$

So, a quadratic majorant function of f_2 at $\boldsymbol{x}'$ is

$$Q(\boldsymbol{x}, \boldsymbol{x}') = f_2(\boldsymbol{x}') + (\boldsymbol{x} - \boldsymbol{x}')^\top \nabla f_2(\boldsymbol{x}') + \frac{\mu\beta}{2}(\boldsymbol{x} - \boldsymbol{x}')^\top \mathbf{A}(\boldsymbol{x} - \boldsymbol{x}'), \quad (8)$$

where $\mathbf{A}$ is a symmetric positive definite matrix majorizing $\boldsymbol{\Omega}^\top \boldsymbol{\Omega}$ (i.e. $\mathbf{A} - \boldsymbol{\Omega}^\top \boldsymbol{\Omega}$ is semi-definite positive). Instead of directly minimizing $f_1 + f_2$, we design our optimization algorithm to minimize the surrogate function $f_1 + Q(\cdot, \boldsymbol{x}_n)$ at iteration n, where $\boldsymbol{x}_n$ is the previous iterate. This leads to the iteration

$$\boldsymbol{x}_{n+1} = \mathrm{prox}_{\gamma_n^{-1}\mathbf{A}, f_1}\big(\boldsymbol{x}_n - \gamma_n \mathbf{A}^{-1} \nabla f_2(\boldsymbol{x}_n)\big), \quad (9)$$

where, for more flexibility, we have substituted a parameter $\gamma_n \in]0, +\infty[$ for the factor $(\mu\beta)^{-1}$. Recall that the proximity operator of function $\gamma_n f_1$ relative to the metric induced by $\mathbf{A}$ is given by

$$(\forall \boldsymbol{x} \in \mathbb{R}^L) \quad \mathrm{prox}_{\gamma_n^{-1}\mathbf{A}, f_1}(\boldsymbol{x}) = \underset{\boldsymbol{z} \in \mathbb{R}^L}{\mathrm{argmin}} \ \gamma_n f_1(\boldsymbol{z}) + \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{x}\|_{\mathbf{A}}^2, \quad (10)$$

where $\|\cdot\|_{\mathbf{A}}$ is the weighted norm of $\mathbb{R}^L$ defined as

$$(\forall \boldsymbol{z} \in \mathbb{R}^L) \quad \|\boldsymbol{z}\|_{\mathbf{A}} = (\boldsymbol{z}^\top \mathbf{A} \boldsymbol{z})^{1/2}. \quad (11)$$

It must be emphasized that working with $\mathbf{A}$ instead of $\boldsymbol{\Omega}^\top \boldsymbol{\Omega}$, or a scaled version of the Hessian of f_2 at $\boldsymbol{x}_n$, allows us to avoid the cumbersome inversion of a large-size matrix, provided that $\mathbf{A}$ has a simple structure. In our case, we propose to choose a simple diagonal form for this matrix. According to [12, Lemma 5.1], such a diagonal preconditioning matrix can be obtained as follows:

$$\mathbf{A} = \mathrm{Diag}(\boldsymbol{R}^\top \mathbb{1}_P) \quad (12)$$

where $\boldsymbol{R} = (R_{i,l'})_{1 \leq i \leq P, 1 \leq l' \leq L}$ with, for every $i \in \{1, \ldots, P\}$ and $l' \in \{1, \ldots, L\}$,

$$R_{i,l'} = \Omega_{i,l'} \sum_{l=1}^{L} \Omega_{i,l}. \quad (13)$$

Because of the specific form of matrix $\boldsymbol{\Omega}$, the l'-th diagonal element of $\mathbf{A}$ with $l' \in \{1, \ldots, P\}$ is thus equal to $g - 1$ if l' is the index of an edge between a TF and a nonTF, and it is equal to $2(g - 1)$ otherwise.

3.2. Block coordinate speedup

As our objective function has been decomposed into the sum of a differentiable function f_2 and an additively block separable function f_1, an improvement of the convergence speed can be expected by resorting to a block coordinate approach [13].

Let $(p, Q) \in \mathbb{N}^*$ be such that $L = pQ$, and let $(\mathcal{P}_k)_{1 \leq k \leq p}$ be a partition of $\{1, \ldots, L\}$ in subsets of cardinality Q. The k-th element of the partition corresponds to a set of indices defining a block of edge weights $\boldsymbol{x}_n^{(k)} \in \mathbb{R}^Q$ which may be activated at iteration n of the algorithm, the other $L - Q$ variables being unchanged. For example, one can choose $\mathcal{P}_k = \{Q(k-1) + 1, \ldots, kQ\}$. The benefit of this approach is not only to reduce the number of variables updated at each iteration, but also to perform the gradient computation only with respect to the reduced-size vector $\boldsymbol{x}_n^{(k)}$ by making use of the submatrix $\boldsymbol{\Omega}_k$ of $\boldsymbol{\Omega}$ of dimension $P \times Q$ corresponding to the activated edges. In addition, a more adapted preconditioning matrix $\mathbf{A}_k \in \mathbb{R}^{Q \times Q}$ may be employed, which is computed in a similar way as in Section 3.1 as a diagonal majorizer of $\boldsymbol{\Omega}_k^\top \boldsymbol{\Omega}_k$. The resulting algorithm is summarized below.

Algorithm 1: Block Coordinate Preconditioned Forward-Backward (BC-P-FB) algorithm

Fix $x_0 \in \mathbb{R}^N$
for $n = 0, 1, \ldots$ **do**
  Select the index $k_n \in \{1, \ldots, p\}$ of a block of variables
  $\boldsymbol{z}_n^{(k_n)} = \boldsymbol{x}_n^{(k_n)} - \gamma_n \mathbf{A}_{k_n}^{-1} \boldsymbol{\Omega}_{k_n}^\top \nabla \Phi(\boldsymbol{\Omega}\boldsymbol{x}_n - \mathbf{d})$
  $\boldsymbol{x}_{n+1}^{(k_n)} = \mathrm{prox}_{\gamma_n^{-1}\mathbf{A}_{k_n}, f_1^{(k_n)}}(\boldsymbol{z}_n^{(k_n)})$
  $\boldsymbol{x}_{n+1}^{(k)} = \boldsymbol{x}_n^{(k)}, \quad k \in \{1, \ldots, p\} \setminus \{k_n\}$

Note that last step of the above algorithm involves the computation of the proximity operator $\mathrm{prox}_{\gamma_n^{-1}\mathbf{A}_{k_n}, f_1^{(k_n)}}$, which reduces here to a simple projection onto the convex set $[0, 1]^Q$.

Assuming that the sequence of step-sizes $(\gamma_n)_{n \in \mathbb{N}}$ is such that $\inf_{n \in \mathbb{N}} \gamma_n > 0$ and $\sup_{n \in \mathbb{N}} \gamma_n < 2(\mu\beta)^{-1}$, and the block sweeping strategy follows a quasi-cyclic rule, which means that every block of variables is called in a finite number of iterations,[1] Algorithm 1 is guaranteed to converge to a (global) minimizer of Problem (6) [13, Theorem 4.1]. A last thresholding at 0.5 of the so-obtained minimizer gives us the list of edges present in the inferred graph.

4. EXPERIMENTAL RESULTS

After a short description of the validation dataset, we detail the different parameter settings for our method. We then compare our results with the current state-of-the-art methods in gene network inference, namely GENIE3 [3] and CLR [2].

[1] More precisely, there exists $K \in \mathbb{N}^*$ such that, for every $n \in \mathbb{N}$, $\{1, \ldots, p\} \subset \{k_n, \ldots, k_{n+K-1}\}$.

4.1. Dataset description and parameter setting

The Dialogue for Reverse Engineering Assessments and Methods (DREAM) [1] fourth multifactorial challenge provides five simulated datasets with real network topologies from *E. coli* and *S. cerevisia*, and simulated expression data. The networks are composed of 100 genes, with a total of 100 expression levels per gene. We use the list of TFs available in the dataset for GENIE3 and Brane relax as well.

We employ as input weights $\mathbf{w}$ the normalized Z-scores computed over mutual information values given by the CLR method. The components of the vector $\boldsymbol{\lambda}$ are chosen as $(\forall (i,j) \in \mathbb{E})\ \lambda_{i,j} = \alpha_i + \alpha_j$, where $\alpha_i = \lambda_{\mathrm{TF}}$ if $i \in \mathcal{T}$, and $\alpha_i = \lambda_{\mathrm{nonTF}}$ otherwise. These parameters may be interpreted as two thresholds: λ_{TF} acting on edges linked with at least a regulatory gene and λ_{nonTF} acting on the other edges. Given the low proportion of TF-TF relationships in the biological reality, we chose to promote the other interactions, thus to impose $\lambda_{\mathrm{nonTF}} < \lambda_{\mathrm{TF}}$. Finally, the desired mean degree d of our model is set to 3 based on biological common knowledge.

4.2. Comparative evaluation

To evaluate the obtained networks, we compare them with the true network using two measures: the Precision, defined as $\frac{\mathrm{TP}}{\mathrm{TP+FP}}$ and the Recall, defined as $\frac{\mathrm{TP}}{\mathrm{TP+FN}}$, where TP is the number of true positive, FP is the number of false positive and FN is the number of false negative. The Precision value indicates the proportion of correctly inferred edges (TP) compared to the total number of inferred edges (TP + FP). The Recall value reveals the proportion of correctly inferred edges (TP) compared to the total number of expected edges given by the gold standard (TP + FN). Results in terms of Area Under the Precision-Recall curves (AUPR) are reported in Table 1. This measure reflects the accuracy of the graph construction using different strategies in comparison with the ground truth. We first emphasize the efficiency of our term involving different $\lambda_{i,j}$ parameter values depending on the gene nature. By turning off the action of the degree prior (by choosing $\mu = 0$), we show that this term allows us to outperform the CLR method by 4.9%. Then, setting μ to 0.005 allows us to reach a 5.9% improvement over CLR, and to outperform GENIE3 as well.

We evaluate the impact of the choice of function Φ on the results. The squared ℓ_2 norm ($\Phi(\cdot) = \|\cdot\|^2$) and the Huber function have been tested. The latter is defined as: $(\forall y \in \mathbb{R})$ $\phi(y) = y^2$ if $|y| < \delta$, and $2\delta(|y| - \frac{1}{2}\delta)$ otherwise, with $\delta \in]0, +\infty[$. Although the performance of quadratic or Huber functions appear similar, the latter may benefit from more robustness to outliers when dealing with real data. Preliminary results on a large *E. Coli* compendium [2] yield an AUPR of 0.0639, which constitute relative improvements of 1.3% and 4.6% over GENIE3 and CLR respectively.

4.3. Computation times

Fig. 2 illustrates the speed gain using the preconditioning and block coordinate tricks presented in this paper in comparison with the standard Forward-Backward algorithm, and FISTA [17]. A measure of convergence of our solution is provided by the variations of $(\frac{\|\boldsymbol{x}_n - \widehat{\boldsymbol{x}}\|}{\|\widehat{\boldsymbol{x}}\|})_{n \in \mathbb{N}}$, where $\boldsymbol{x}_n$ is the labeling obtained at the n-th iteration, while $\widehat{\boldsymbol{x}}$ is the optimal labeling pre-computed over a large number of iterations. In comparison, it took one second to compute the weights $\mathbf{w}$ using CLR, and one minute for the GENIE3 approach. The speedups are important since the developed GRN inference method aims to be usable on larger datasets (with $g \simeq 4000$ for the *E. coli* compendium [2]).

Network index	1	2	3	4	5
CLR	0.251	0.255	0.306	0.304	0.310
GENIE3	0.239	0.260	0.323	0.311	0.306
Brane relax[1]	0.260	0.263	0.324	0.320	0.331
Brane relax[2]	**0.261**	**0.264**	0.327	0.323	0.331
Brane relax[3]	0.256	**0.264**	**0.332**	**0.327**	**0.334**

Table 1. AUPR ($\times 10^{-2}$) for the different methods. The Precision-Recall curves are obtained by varying the λ_{nonTF} parameter between 0 and 0.1, in a linearly, equally-spaced way. For each λ_{nonTF}, λ_{TF} are chosen to be linearly equally spaced between λ_{nonTF} and 0.9. Brane relax[1]: $\mu = 0$. Brane relax[2]: Φ Huber function ($\delta = 2.2$). Brane relax[3]: Φ squared ℓ_2 norm.

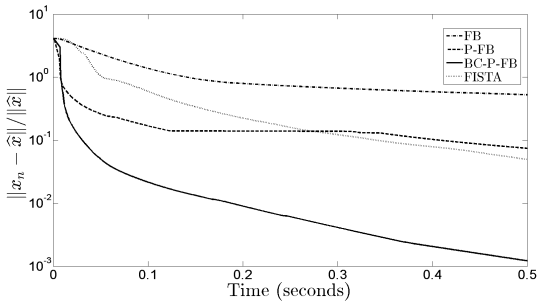

Fig. 2. Comparison of the convergence speed for various algorithms, when $\Phi = \|\cdot\|^2$ in the objective function (5). FB: Forward-Backward, P-FB: Preconditioned FB, BC-P-FB: Block Coordinate Preconditioned FB.

5. CONCLUSION

In contrast to most existing works on gene network inference which often ignore the availability of transcription factor lists, we have proposed a novel optimization approach for building such networks encompassing a connectivity prior on particular genes. Our problem reformulation and the thoughtful usage of recent advances in accelerated strategies allow us to compute an optimal network according to our model in less than one second using pre-computed weights on biological datasets. Our experiments on the DREAM4 dataset show that Brane relax compares favorably to state-of-the-art methods. Forthcoming investigations include the evaluation of the interplay of these more accurate graph predictions with clustering methods [18]. The recent developments in graph signal processing techniques [19] are also likely to provide insights in a deeper understanding of gene regulatory dependencies.

6. REFERENCES

[1] D. Marbach, R. J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, and G. Stolovitzky, "Revealing strengths and weaknesses of methods for gene network inference," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 107, no. 14, pp. 6286–6291, Apr. 2010.

[2] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins, and T. S. Gardner, "Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles," *PLoS Biol.*, vol. 5, pp. 54–66, 2007.

[3] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts, "Inferring regulatory networks from expression data using tree-based methods," *PLoS One*, vol. 5, no. 9, pp. 1–10, Sep. 2010.

[4] M. E. J. Newman, "The structure and function of complex networks," *SIAM Rev.*, vol. 45, no. 2, pp. 167–256, Jan. 2003.

[5] M. G. Rabbat, M. A. T. Figueiredo, and R. D. Nowak, "Network inference from co-occurrences," *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 4053–4068, Sep. 2008.

[6] D. Zhu and A. O. Hero, "Network constrained clustering for gene microarray data," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Philadelphia, PA, USA, Mar. 18-23, 2005, vol. 5, pp. 765–768.

[7] X. Qian, B.-J. Yoon, and E. R. Dougherty, "Structural intervention of gene regulatory networks by general rank-k matrix perturbation," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Kyoto, Japan, Mar. 25-30, 2012, pp. 729–732.

[8] A. Noor, E. Serpedin, M. Nounou, and H. Nounou, "Inferring gene regulatory networks with nonlinear models via exploiting sparsity," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Kyoto, Japan, Mar. 25-30, 2012, pp. 725–728.

[9] H. Wang, N. Bouaynaya, R. Shterenberg, and D. Schonfeld, "Sparse biologically-constrained optimal perturbation of gene regulatory networks," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Vancouver, BC, Canada, May 26-31, 2013, pp. 1167–1171.

[10] N. Bouaynaya, R. Shterenberg, and D. Schonfeld, "Optimal perturbation control of general topology molecular networks," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1733–1742, Apr. 2013.

[11] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.

[12] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, "Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function," *J. Optim. Theory Appl.*, vol. 162, no. 1, pp. 107–132, Jul. 2014.

[13] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, "A block coordinate variable metric forward-backward algorithm," Tech. Rep., 2013, `http://www.optimization-online.org/DB_HTML/2013/12/4178.html`.

[14] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*, H. H. Bauschke, R. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, Eds., pp. 185–212. Springer Verlag, 2011.

[15] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 123–231, 2013.

[16] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, CMS books in mathematics. Springer, 2011.

[17] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009.

[18] B. Abu-Jamous, R. Fa, D. J. Roberts, and A. K. Nandi, "Identification of genes consistently co-expressed in multiple microarray datasets by a genome-wide Bi-CoPaM approach," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Vancouver, BC, Canada, May 26-31, 2013, pp. 1172–1176.

[19] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.