ABSTRACT

Motivation: Network component analysis (NCA) is an efficient method of reconstructing the transcription factor activity (TFA), which makes use of the gene expression data and prior information available about transcription factor (TF)-gene regulations. Most of the contemporary algorithms either exhibit the drawback of inconsistency and poor reliability, or suffer from prohibitive computational complexity. In addition, the existing algorithms do not possess the ability to counteract the presence of outliers in the microarray data. Hence, robust and computationally efficient algorithms are needed to enable practical applications.

Results: We propose ROBust Network Component Analysis (ROBNCA), a novel iterative algorithm that explicitly models the possible outliers in the microarray data. An attractive feature of the ROBNCA algorithm is the derivation of a closed form solution for estimating the connectivity matrix, which was not available in prior contributions. The ROBNCA algorithm is compared to FastNCA and the Non-iterative NCA (NI-NCA). ROBNCA estimates the TF activity profiles as well as the TF-gene control strength matrix with a much higher degree of accuracy than FastNCA and NI-NCA, irrespective of varying noise, correlation and/or amount of outliers in case of synthetic data. The ROBNCA algorithm is also tested on Saccharomyces Cerevisiae data and Eschericia coli data and it is observed to outperform the existing algorithms. The run time of the ROBNCA algorithm is comparable to that of FastNCA, and is hundreds of times faster than NI-NCA.

Availability: The ROBNCA software is available at http://people.tamu.edu/~amina/ROBNCA.

Contact: serpedin@ece.tamu.edu

1 INTRODUCTION

Recent advances in technology have enabled monitoring of cellular activities using more sophisticated techniques, and have provided a deluge of biological data. Using this data to unravel the underlying phenomena that regulate various activities in a living organism offers the potential to reap numerous benefits. One of the key biological processes is transcriptional regulation, that controls the gene expression and amount of RNA produced. This process is regulated by transcription factors (TFs) which are specialized proteins causing the genes to express by binding onto the gene promoters. A thorough understanding of this complex transcriptional regulation and TF-gene interaction will potentially aid in predicting the biological processes and designing control strategies to cure and/or avoid the diseased conditions (Lähdesmäki et al., 2008). Microarray technologies are able to measure the level of gene expressions and quantify them in the form of gene expression data. Such data are widely used in the inference of gene-gene interactions. Transcription factor activity (TFA), which is defined as the concentration of its subpopulation with DNA binding ability, controls the transcriptional regulation (Jajamovich et al., 2011). The correlation between TFAs and TF expression level is modified at the post-transcriptional and post-translational stage. It is, therefore, much harder to measure TFA profiles experimentally, and scientists have resorted to computational methods for their estimation (Yang et al., 2005).

Several statistical techniques including principal component analysis (PCA) (Jolliffe, 1986) and independent component analysis (ICA) (Comon, 1992) have been used to deduce useful information from sets of biological data. However, the successful application of these algorithms hinges on the assumptions of orthogonality and independence between the signals, which do not hold for biological signals in practice (Chang et al., 2008). In fact, some prior information is usually available for many systems, and it should be incorporated in the system model, e.g., ChIP-chip data indicates which TFs and genes are known to interact. The gene regulatory network can be modeled linearly as follows (Liao et al., 2003)

\[ Y = AS + \Gamma, \]

where \( Y \) is the \( N \times K \) gene expression data matrix, \( A \) is the \( N \times M \) control strength or connectivity matrix, and \( S \) is the \( M \times K \) matrix denoting the TFs. The uncertainties in the observation data are assumed to be Gaussian (Chang et al., 2008; Jacklin et al., 2012), and are represented by the entries of the noise matrix \( \Gamma \). Genes and TFs are known to interact in a dynamic and non-linear manner however, a log-linear relationship provides a good approximation. Since a particular TF regulates only a few other genes, the connectivity matrix \( A \) is expected to be sparse. The problem then boils down
to estimating $S$ and $A$, where $Y$ is available and some a-priori information about the matrix $A$ is known.

Network component analysis (NCA), proposed by (Liao et al., 2003), provides a more accurate model for TF-gene regulation and makes use of the related prior information available. It was shown that provided certain conditions are met, the NCA algorithm produces a unique solution of the aforementioned estimation problem in the absence of noise. The NCA criteria require that: (i) the matrix $A$ is full column-rank; (ii) if a row is removed from $S$ as well as the output elements connected to it, the updated control strength matrix should still be of full column-rank; (iii) the TFA matrix $S$ should have a full row-rank. These criteria guarantee that the solution obtained is unique up to a scale ambiguity and there is no sign ambiguity (Liao et al., 2003; Jacklin et al., 2012). When the NCA criteria are satisfied, the optimization problem reduces to:

$$\min_{A,S} ||Y - AS||_F^2 \text{ s.t. } A(I) = 0,$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $I$ is the set of all indices where the entries of matrix $A$ are zero. The algorithm in (Galbraith et al., 2006) allows the recovery of source signals when the microarray data consists of fewer data points and (Tran et al., 2005) formulates the incorporation of regulatory knockout constraints as well.

The NCA problem in (2) was first solved by employing alternate least squares (ALS) for both $A$ and $S$ (Liao et al., 2003). However, since the ALS solution requires solving a high dimensional matrix optimization problem at each iteration, it entails prohibitive computational complexity for large data sets, which often need to be handled in gene networks. FastNCA provides a closed form solution for $A$ which employs singular value decomposition (SVD) (Chang et al., 2008), and is several tens of times faster than the ALS algorithm. The authors in (Jacklin et al., 2012) propose a non-iterative version of NCA, herein referred to as NI-NCA, which offers greater consistency in terms of TFA estimation at the cost of much higher computational complexity than FastNCA. However, since the decomposition techniques used to derive these algorithms are susceptible even to the presence of small amount of outliers (Mateos and Giannakis, 2012), their performance is expected to deteriorate significantly when data points are corrupted by outliers. It is commonly known that the microarray data are very noisy and are corrupted with outliers because of erroneous measurements and/or abnormal response of genes, and robust algorithms are required for gene network inference (Finegold and Drton, 2011). Therefore, it is imperative to develop an NCA algorithm which has an inherent ability to mitigate the effect of outliers, and also entails low computational costs and provides good consistency and accuracy. It is precisely this avenue which is the focus of our current work. The main contributions of this paper can be summarized as follows:

1. A novel algorithm, ROBust Network Component Analysis (ROBNCA), is proposed which has the inherent ability to counteract the presence of outliers in the data $Y$ by explicitly modeling the outliers as an additional sparse matrix. The iterative algorithm estimates each of the parameters efficiently at each iteration, and delivers superior consistency and greater accuracy for TFA estimation.

2. A particularly attractive feature of the ROBNCA algorithm is the derivation of a closed form solution for the estimation of the connectivity matrix $A$, a major source of high computational complexity in contemporary algorithms. In order to further lower the computational burden, a still faster closed form solution is derived that requires matrix inversion of much smaller size. The resulting algorithm is comparable to FastNCA in terms of computational complexity, and is hundreds of times faster than NI-NCA.

3. The performance of ROBNCA is tested on Hemoglobin test data from (Jacklin et al., 2012) for both low and highly correlated source signals. ROBNCA is seen to outperform the state-of-the-art algorithms for estimating both $A$ and $S$ in terms of mean square error (MSE). In addition, ROBNCA is applied to yeast cell cycle data (Lee et al., 2002) and E.coli data (Kao et al., 2004) and by plotting the standard deviation of estimates, it is observed that ROBNCA offers better consistency than FastNCA and NI-NCA.

2 METHODS

2.1 NCA with Outliers

Most of the contemporary algorithms have studied the gene network construction problem using NCA with Gaussian noise models. However, inaccuracies in measurement procedures and abnormal gene responses often render heavier tails to the gene expression data, and Gaussian noise models may no longer be a natural fit in these cases. The decomposition techniques employed in the available algorithms are highly sensitive to the presence of outliers i.e., the samples that do not conform to the Gaussian noise model, and their estimation capabilities are extremely susceptible to outliers. As a consequence, the gene network inference becomes unreliable for practical purposes. Therefore, we focus on deriving computationally efficient NCA algorithms which are robust to the presence of outliers.

Towards that end, we take the approach of explicitly modeling the outliers as an additional matrix that corrupts the data points. From (1), it follows that the complete system model that accounts for the presence of outliers as well as noise can be expressed as

$$Y = AS + O + \Gamma,$$

where the matrix $O$ denotes the outliers. The outlier matrix $O$ is a column sparse matrix since there are typically a few outliers. The joint optimization problem for the estimation of the three parameters, that also allows for controlling outlier sparsity, can be formulated as

$$\{A,S,O\} = \arg \min_{A,S,O} ||Y - AS - O||_F^2 + \lambda_0||O||_0$$

such that $A(I) = 0,$

$$\text{min}_{A,S,O} \{A,S,O\} \text{ s.t. } ||Y - AS - O||_F^2 + \lambda_0||O||_0$$

where the non-convex $l_0$ norm $||O||_0$ denotes the number of nonzero columns in $O$, and the extent of sparsity in the columns of $O$ is controlled by the tuning parameter $\lambda_0$. The optimization problem in (4) is reminiscent of compressive sampling techniques based on the $l_0$ norm, and are known to be NP-hard (Tropp, 2006). Therefore, some relaxation is needed in order to solve the joint optimization problem without incurring exponentially increasing computational complexity. A viable alternative is the column-wise $l_2$ sum i.e., $||O||_{2,e} = \sum_{k=1}^K ||O_k||_{2,e}$, which is the closest convex approximation of $||O||_0$ (Tropp, 2006). With this relaxation, the resulting joint optimization problem can be expressed as

$$\{A,S,O\} = \arg \min_{A,S,O} ||Y - AS - O||_F^2 + \lambda_2||O||_{2,e}$$

such that $A(I) = 0.$

Our goal is to estimate the three parameters $A$, $S$ and $O$ by solving the optimization problem (5). However, it can be noticed that the optimization problem is not jointly convex with respect to (w.r.t) $\{A,S,O\}$. Therefore,
we resort to an iterative algorithm that alternately optimizes (5) w.r.t one parameter at a time.

2.2 The ROBNCNA Algorithm

The update of each of the parameters, \(S(j), A(j)\) and \(O(j)\), at an iteration \(j\) is discussed as follows.

2.2.1 Update of the TFA Matrix At iteration \(j\), the value of the parameter \(S(j)\) is updated by minimizing the objective function (5) w.r.t \(S\), while fixing the parameters \(A\) and \(O\) to their respective values at iteration \((j - 1)\). By defining the matrix \(X(j) = Y - O(j - 1)\), the optimization problem can be written as

\[
S(j) = \arg \min_S \|X(j) - A(j - 1)S\|_F^2. \tag{6}
\]

Since the connectivity matrix \(A(j - 1)\) has full column rank (by virtue of NCA criterion 1), the matrix \(A^T(j - 1)A(j - 1)\) is invertible. Therefore, an estimate of the TFA matrix \(S\) at the \(j^{th}\) iteration can be readily obtained as

\[
S(j) = (A^T(j - 1)A(j - 1))^{-1} A^T(j - 1)X(j). \tag{7}
\]

The estimate \(S(j)\), so obtained, is used in the upcoming steps to determine \(A\) and \(O\).

2.2.2 Update of the Connectivity Matrix The next step in the iterative algorithm is to solve the optimization problem (5) w.r.t the matrix \(A\), while fixing the values of the parameters \(S\) and \(O\) to \(S(j)\) and \((O(j - 1)\), respectively. The result of optimizing \(A(j)\) can be written as

\[
A(j) = \arg \min_A \|X(j) - AS(j)\|_F^2,
\]

such that \(A(I) = 0\).

Remark 1. The optimization problem (8) was also considered in the original work on NCA by Liao et al. (Liao et al., 2003). However, a closed form solution was not provided and the proposed algorithm relied on costly optimization techniques to update the matrix \(A\). Since this minimization needs to be performed at each iteration until convergence, the ALS algorithm is known to be extremely slow for large networks, and computational resources required may be prohibitive (Jacklin et al., 2012). Hence, it is imperative that a closed form solution is obtained for the optimization problem in (8), so that the algorithm is faster and efficient.

Without loss of generality, we can consider the transposed system

\[
\mathbf{X} = \mathbf{S} \mathbf{A} + \mathbf{\Gamma}, \tag{9}
\]

where \(\mathbf{X}, \mathbf{S}, \mathbf{A},\) and \(\mathbf{\Gamma}\) denote the transposed of the original matrices, respectively. The resulting equivalent optimization problem can now be stated as

\[
A(j) = \arg \min_A \|X(j) - S(j)A\|_F^2,
\]

such that \(A(I) = 0\), \(10\)

where \(I\) is the set of all indices where the entries of the matrix \(A\) are known to be zero. The following theorem presents a closed form solution of the optimization problem (10), herein referred to as ROBNCNA 1.

Theorem 1. The solution of (10) at the \(j^{th}\) iteration is given by

\[
\hat{a}_n(j) = Q^{-1}(j) \left[ \tilde{w}_n(j) - C_n^{-1} \Psi^{-1}(j) C_n Q^{-1}(j) \tilde{w}_n(j) \right], \tag{11}
\]

where \(\Psi(j) = C_n Q^{-1}(j) C_n\), \(\tilde{w}_n(j) = \tilde{S}(j)\hat{a}_n(j)\), the symmetric matrix \(Q(j) = S(j)S(j)\trans\), and \(\hat{a}_n\) and \(\tilde{a}_n\) represent the \(n^{th}\) columns of matrices \(A\) and \(X\), respectively. The \(L_n \times M\) matrix \(C_n\) is a matrix of zeroes except \(C_{nn}(I_n) = 1\), where \(I_n\) is the set of indices where the entries of \(\hat{a}_n\) are zero, and \(L_n\) denotes the number of zero entries in \(\hat{a}_n\).

Proof. The \(n^{th}\) column of (9) can be written as

\[
\hat{a}_n = \hat{S} \tilde{a}_n + \gamma_n. \tag{12}
\]

The objective function in (10) can be equivalently expressed as

\[
\|X(j) - \hat{S}(j)A\|_F^2 = \sum_{n=1}^{N} \|\tilde{w}_n(j) - \hat{S}(j)\hat{a}_n\|^2. \tag{13}
\]

The constraint \(A(I) = 0\) can be written as a set of \(n\) constraints \(C_{nn}\hat{a}_n = 0\) for \(n = 1, \ldots, N\). The \(L_n \times M\) matrix \(C_n\) is constructed such that it consists of all zeroes except \(C_{nn}(I_n) = 1\). For instance, if \(M = 6\), and \(\hat{a}_n = [\alpha_n 1; \alpha_n 2; 0; \alpha_n 4; 0; \alpha_n 6]^T\), the \(2 \times 6\) matrix \(C_n\) consists of all zeroes except \(C_{nn}(1,3) = C_{nn}(2,5) = 1\). It can be easily verified that the matrix \(C_n\) so constructed has full row rank.

The optimization problem in (10) can now be written as

\[
\hat{A}(j) = \arg \min_{\hat{A}} \sum_{n=1}^{N} \|\tilde{w}_n(j) - \hat{S}(j)\hat{a}_n\|^2
\]

such that \(C_{nn}\hat{a}_n = 0\), for \(n = 1, \ldots, N\). \(14\)

The optimization problem is, therefore, separable in terms of columns of \(\hat{A}\), and can be equivalently solved by considering one column at a time. This also reduces the computational complexity of estimating the connectivity matrix \(\hat{A}\). Henceforth, we will employ convex optimization techniques to derive a closed form solution of the separable optimization problem. For the \(n^{th}\) column, we have

\[
\hat{a}_n(j) = \arg \min_{\hat{a}_n} \frac{1}{2} \tilde{w}_n(j)^T Q(j) \hat{a}_n - \tilde{w}_n(j)^T \tilde{a}_n + \mu^T C_{nn}\hat{a}_n.
\]

The Karush-Kuhn-Tucker (KKT) conditions can be written as (Boyd and Vandenberghe, 2004)

\[
Q(j)\hat{a}_n - \tilde{w}_n(j) = C_{nn}^T \mu = 0 \tag{16}
\]

\[
C_{nn}\hat{a}_n = 0. \tag{17}
\]

Lemma 1. The KKT conditions are necessary and sufficient for the optimization problem (15).

Proof. Since the optimization problem (15) contains linear equality constraints, the KKT conditions are necessary for optimality (Boyd and Vandenberghe, 2004). Let any \(\hat{a}_n^*\) be a local minimum. Then, since the KKT conditions are necessary, there exists a Lagrange multiplier \(\mu^*\) such that \(\hat{a}_n(\mu^*)\) is the solution to the system of equations in (16) and (17). Now, since the objective function is convex, it follows that \(\hat{a}_n^*\) is also a global minimum (Boyd and Vandenberghe, 2004). This implies that the KKT conditions are also sufficient for optimality.

Hence, a solution to (15) can be obtained by solving the KKT system of equations. Using (16), it follows that

\[
\hat{a}_n = Q^{-1}(j) \left( \tilde{w}(j) - C_n^T \mu \right), \tag{18}
\]

where the matrix \(Q(j)\) is indeed invertible by virtue of the linear independence of the rows of \(S\) (NCA criterion 3). Substituting (18) in (17), we have

\[
C_n Q^{-1}(j) C_n^T \mu = C_n Q^{-1}(j) \tilde{w} .
\]

Since the matrix \(C_n\) has full row rank, the matrix \(\Psi(j) = C_n Q^{-1}(j) C_n^T\) is invertible. The Lagrange multiplier can, therefore, be expressed as

\[
\mu = \Psi^{-1}(j) C_n Q^{-1}(j) \tilde{w} . \tag{19}
\]

Upon substituting (19) in (18), the solution \(\hat{a}_n\) in Theorem 1 readily follows.
Therefore, using Theorem 1, an estimate of $\hat{A}(j)$ can be efficiently obtained and this approach results in substantial reduction in computational complexity compared to the ALS algorithm.

**Remark 2.** While the aforementioned closed form solution provides a significant advantage in terms of computational complexity over the ALS algorithm, we note that the solution requires inverting the matrix $Q$. For large networks, this can potentially be a large matrix, whose inverse incurs computational load, and may lead to inaccuracies as well. In the following discussion, we derive a still faster algorithm, ROBNCA 2, that takes advantage of the special structure of the column vector $\hat{a}_n$ and provides added savings over the closed form solution derived in Theorem 1.

We begin by noting that the rows of $X$ and $\hat{A}$ can always be reordered in (9). Hence, without loss of generality, the vector $\hat{a}_n$ can be partitioned as

$$\hat{a}_n = \begin{bmatrix} \hat{a}_n^T \\ 0_{n_m \times 1} \end{bmatrix},$$

where $\hat{a}_n \in R^{(M - L_n) \times 1}$ is a vector consisting of the non-zero entries in $\hat{a}_n$. Construct an $L_n \times M$ matrix $U_n$ such that

$$U_n = \begin{bmatrix} I_{(M - L_n)} \\ 0_{L_n \times (M - L_n)} \end{bmatrix}.$$  

With the above definition, the optimization problem (15) can be equivalently represented as

$$\hat{a}_n(j) = \arg \min_{\hat{a}_n} \frac{1}{2} \hat{a}_n^T Q(j) \hat{a}_n - \hat{w}_n^T(j) \hat{a}_n$$

such that $U_n \hat{a}_n = 0$.

(22)

Define a substitution

$$\hat{a}_n = V_n \hat{a},$$

where the $M \times L_n$ matrix $V_n$ is constructed such that it lies in the null space of the matrix $U_n$, i.e., $U_n V_n = 0$. The matrix $V_n$ is, therefore, given by

$$V_n = \begin{bmatrix} I_{(M - L_n)} \\ 0_{L_n \times (M - L_n)} \end{bmatrix}.$$  

(24)

By substituting $\hat{a}_n$ from (23) into (22), and noting that the constraint is always satisfied due to the construction of $V_n$, we have an unconstrained optimization problem in the variable $\hat{a}$ given by

$$\hat{a}_n(j) = \arg \min_{\hat{a}_n} \frac{1}{2} \hat{a}_n^T V_n^T Q(j) V_n \hat{a} - \hat{w}_n^T(j) V_n \hat{a}.$$  

(25)

The solution of the aforementioned unconstrained quadratic optimization problem can be easily obtained as

$$\hat{a}_n(j) = \left( V_n^T Q(j) V_n \right)^{-1} V_n^T \hat{w}_n(j),$$

where the matrix $V_n^T Q(j) V_n$ is invertible since $V_n$ has full column rank.

The symmetric invertible matrix $Q(j)$ can be partitioned as

$$Q(j) = \begin{bmatrix} Q_{11}(j) & Q_{12}(j) \\ Q_{21}(j) & Q_{22}(j) \end{bmatrix},$$

where the invertible matrix $Q_{11}(j)$ is the upper $(M - L_n) \times (M - L_n)$ submatrix of $Q(j)$. From the structure of $V_n$, the matrix $V_n^T Q(j) V_n$ can be reduced as

$$V_n^T Q(j) V_n = \begin{bmatrix} I_{(M - L_n)} \\ 0_{(M - L_n) \times L_n} \end{bmatrix} \begin{bmatrix} Q_{11}(j) & Q_{12}(j) \\ Q_{21}(j) & Q_{22}(j) \end{bmatrix} \begin{bmatrix} I_{(M - L_n)} \\ 0_{L_n \times (M - L_n)} \end{bmatrix} = Q_{11}(j).$$  

(27)

Similarly, by partitioning $\hat{w}_n(j)$ as

$$\hat{w}_n(j) = \begin{bmatrix} \hat{w}_n^T(j) \\ \hat{w}_n^T(j) \end{bmatrix},$$

it follows that

$$V_n^T \hat{w}_n(j) = \hat{w}_n(j),$$

(28)

where $\hat{w}_n(j)$ is the upper $(M - L_n) \times 1$ vector of $\hat{w}_n(j)$. Collecting all the terms, the solution $\hat{a}_n$ can now be compactly represented as

$$\hat{a}_n(j) = Q_{11}^{-1}(j) \hat{w}_n(j).$$

(29)

Once all columns $\hat{a}_n(j)$ are determined, the connectivity matrix $A(j)$ can be easily updated.

**Remark 3.** By comparing the closed form solution derived in (11) with (29), it is clear that the latter only requires inverting a submatrix $Q_{11}(j)$ of $Q(j)$. Since the connectivity matrix is usually sparse and the number of non-zero entries $(M - L_n)$ in the $j^{th}$ column is usually very small, inverting the $(M - L_n) \times (M - L_n)$ matrix $Q_{11}(j)$ results in a considerable reduction in computational complexity and ensures a much faster implementation of the iterative algorithm.

The respective computational times incurred in calculating (11) and (29) will be quantified in Section 3 to emphasize the usefulness of deriving (29).

2.2.3 Update of the Outlier Matrix
The last step in the iterative algorithm pertains to the estimation of the outlier matrix $O$ by using the values $S(j)$ and $A(j)$ obtained in the preceding steps. It is straightforward to notice that the optimization problem (5) w.r.t $O$ decouples across the columns and results in $K$ subproblems, each of which being expressed as follows:

$$\alpha_k(j) = \arg \min_{\alpha_k} \| b_k(j) - \alpha_k \|_2^2 + \lambda_2 \| \alpha_k \|_2, \quad k = 1, \ldots, K$$

(30)

where $b_k(j) = y_j - A(j) \alpha_k(j)$. The solution to (30) is given by (Kekatos and Giannakis, 2011)

$$\alpha_k(j) = \frac{b_k(j) \left( \| b_k(j) \|_2^2 - \lambda_2^2 \right) - \lambda_2 b_k(j)}{\| b_k(j) \|_2^2}, \quad k = 1, \ldots, K$$

(31)

where $\lambda_2 \geq \max(0, g)$. The solution (31) is intuitively satisfying since it sets the outlier $\alpha_k(j)$ to zero whenever $\| b_k(j) \|_2$ fails to exceed the threshold $\lambda_2/2$, where $\lambda_2$ is the sparsity-controlling parameter. Several approaches have been identified in the literature for selecting $\lambda_2$ which depend on any a-priori information available about the extent of sparsity (Giannakis et al., 2011). If the concentration of outliers is unknown, a typical rule of thumb is to take $\lambda_2 = 0.7$ where this value has been determined to provide 95% asymptotic efficiency of the estimator (Kekatos and Giannakis, 2011). If a rough estimate of the concentration of outliers is available, (30) can be solved for a grid of values and selecting the $\lambda_2$ giving the expected number of outliers which can be performed efficiently using the Group-LARS algorithm (Yuan and Lin, 2005). It is noted, that the performance of the algorithm is insensitive to minor variations in the value of the parameter. Since the subproblems at each iteration have unique minimizers, and the non-differentiable regularization affects only the outlier matrix $O$, the convergence of the ROBNCA algorithm is established using the results in (Tseng, 2001).

**Proposition 2.** As $j \to \infty$, the iterates generated by the ROBNCA algorithm converge to a stationary point of (5).

It is important to point out that ROBNCA is significantly different from NI-NCA algorithm. NI-NCA, as the name suggests, is a non-iterative algorithm which uses a subspace based method for the estimation of the connectivity matrix $A$ using eigen-decomposition and relies on solving a constrained quadratic optimization problem which has high computational cost. On the other hand, in ROBNCA, we propose two closed form solutions for the estimation of the connectivity matrix $A$ which result in considerable reduction in computational complexity.

3 RESULTS AND DISCUSSION
This section investigates the observed performance of ROBNCA, in comparison with the state-of-the-art algorithms including FastNCA,
NI-NCA, and ALS in terms of MSE using both synthetic and real data. The efficiency and consistency of ROBNCA in estimating the TFAs under various scenarios is also illustrated. The data sets for all of the experiments as well as the MATLAB implementation of FastNCA and NI-NCA are downloaded from http://www.seas.ucla.edu/~liaoj/download.htm and http://www.ece.ucdavis.edu/~jacklin/NCA, respectively.

3.1 Synthetic and Hemoglobin Test Data
First, in order to evaluate the performance of various algorithms, test data from (Liao et al., 2003) is used. The spectroscopy data consists of $M = 7$ hemoglobin solutions formed by mixing up $N = 3$ pure hemoglobin components. The connectivity matrix in this case represents the concentration and presence or absence of each component in the mixture. In addition, the structure of this matrix is validated to comply with the NCA criteria. The absorbance spectra for the range of wavelengths, $C$ denotes the connectivity matrix and $\epsilon$ gives the spectra of the pure components.

The importance of using this data is that this experiment mimics the gene regulatory network very closely and contains all of its key properties. The knowledge of the pure spectra helps us to effectively evaluate the performance of various NCA algorithms. In addition, using the data from (Liao et al., 2003) and (Jacklin et al., 2012) ensures a fair comparison.

The proposed algorithm is tested against varying noise for two very important scenarios: (a) when the source signals are correlated, and (b) the observed data is corrupted with outliers. Using the same connectivity matrix, source signals were generated which had low, moderate and high correlation (Jacklin et al., 2012). The outliers are artificially added to the data by modeling them as a Bernoulli process. The success probability indicates the concentration of outliers present and is assumed to be the same for all the genes. Since only a few points are expected to be corrupted in the real data, the outliers are assumed to be sparse and therefore the success probability of presence of outliers is kept small. The performance of ROBNCA, FastNCA, and NI-NCA is evaluated in the aforementioned scenarios. ROBNCA algorithm is implemented in MATLAB. Since the observed data matrix $Y$ is expected to contain outlying points, the algorithms are assessed by computing the MSE incurred in estimating the matrices $A$ and $S$, instead of fitting error for $Y$.  

Fig. 1. Impact of Correlation: Normalized mean square error (NMSE) (dB) for different algorithms and different data sets with (a) low correlated TFAs (b) highly correlated TFAs against varying signal-to-noise (SNR) ratio(db) with the level of outliers = 0.05.

Fig. 2. Impact of Outliers: Normalized mean square error (NMSE) (dB) for different algorithms and different data sets with (a) level of outliers = 0.01 (b) level of outliers = 0.1 against varying signal-to-noise (SNR) ratio(db) for a highly correlated data set.

NI-NCA
3.1.1 Impact of Correlation  The algorithms are first tested for low and highly correlated source signals by varying the signal-to-noise ratio (SNR). The noise is modeled as Gaussian in all the experiments. The results are averaged over 100 iterations and are depicted in Figure 1. It is observed that the performance of a small amount of outliers makes the estimation using FastNCA very unreliable and inconsistent for both low and highly correlated signals. On the other hand, NI-NCA is able to estimate $S$ better than FastNCA, and the estimation of $A$ is quite accurate and consistent as well. It can be observed that the overall estimation performance for $A$ is much better and more consistent than that of $S$. The reason for this could be attributed to the availability of some prior information for the former. Since ROBNCA takes into account the presence of outliers in the observed data, it outperforms the other two algorithms for estimating both $A$ and $S$ and its consistent performance should be contrasted with the unboundedness and unpredictability exhibited by the other two algorithms. In general, the performance of all the algorithms improves with the increase in SNR and degrades with the increase in correlation of the source signals.

3.1.2 Impact of Outliers  As noted earlier, the presence of outliers can severely affect the performance of algorithms. It is therefore, important to investigate the impact of the presence of outlying points in the observation matrix $Y$. Comparison performed for low and high concentration of outliers is depicted in Figure 2. It is observed from Figure 2(a) that in the case of low concentration of outliers, NI-NCA provides good accuracy for $A$ and estimates it quite consistently. The estimation of $S$ gives a small MSE as well and generally performs consistently. FastNCA, however, is not able to estimate both the matrices even for high SNRs. This indicates its high vulnerability to the presence of even a small number of outliers. In case of a higher concentration of outliers, the performance of NI-NCA degrades a little bit as depicted in Figure 2(b). It is observed that ROBNCA is able to estimate the two matrices for both low and high outliers, and outperforms the other two algorithms.

The estimation of $O$ matrix is shown in the supplementary material where Figure 1 depicts the outliers present in the synthetic data and their estimates using ROBNCA algorithm. It is noted that ROBNCA is able to identify the outliers very well. Figure 2 shows the recovered signal $AS$ after subtracting the outlier matrix $O$ from the data matrix $X$. It can be observed that the recovered signal is a good match with the original signal.

These experiments indicate that ROBNCA solves the estimation problem with much more accuracy than NI-NCA and FastNCA. It is important to emphasize here that the MSE for NI-NCA is always higher than ROBNCA and its computational complexity is many times greater than the latter which can prove to be a bottle-neck in case of large data sets.

3.2 Results for Real Data  We now turn our attention to the comparison of these algorithms on real data. Two datasets are considered for this purpose which are the S.cerevisiae cell cycle data (Lee et al., 2002) and E.coli data (Kao et al., 2004). The transcription factor activity estimates are estimated for the TFs of interest in each experiment and the results are compared for different algorithms. In addition, the variability of the estimates is evaluated using the sub-network analysis (Yang et al., 2005) which will be explained in the following subsections.

3.2.1 S.cerevisiae Cell Cycle Data  The algorithms discussed in this paper are applied to the yeast cell cycle data from (Lee et al., 2002) and (Spellman et al., 1998). In order to assess the performance and variability of the various NCA algorithms, sub-network analysis is performed which has also been used previously in (Chang et al., 2008), (Yang et al., 2005) and (Jacklin et al., 2012). The core idea behind this analysis is to divide the set of transcription factors into a number of smaller subsets, which are not mutually disjoint, where the intersection of these subsets contain the TFs of interest. The subnetworks were constructed to satisfy the gNCA criteria (Tran et al., 2005) which requires that the number of TFAs $M$ should be less than the number of sample points $K$. These sub-networks are used to estimate the transcription factor activities independent of each other. These TFA estimates are then compared and a smaller disagreement between these estimates is a measure of consistency of the algorithm. This indicates that the results obtained are reliable despite of the presence or absence of certain genes or TFs from the experiment. The disagreement can be quantified as: disagreement$(i) = \frac{1}{n} \sum_{s} [\max s_{n,i}(k) - \min s_{n,i}(k)]$ where $s$ indicates the rows of matrix $S$, $i$ is the TF index and $n$ is the sub-network index. The Yeast cell-cycle data set consists of results from three different synchronization experiments. The first experiment is the synchronization by elutriation which is composed of one cell cycle from 0 to 390 mins. The data consists of 14 points
sampled at 30 min intervals. The second experiment performs the synchronization by $\alpha$-factor arrest and cell cycles from 0 to 119 mins. A total of 18 samples are taken every 7 mins. The synchronization in the third set is the result of cdc15 temperature sensitive mutant with samples taken every 20 min from 0 to 300 mins. The data from the three experiments are concatenated to form one large dataset. The Yeast cell cycle study has eleven TFs.

3.2.2 E.coli Data

The performance of NCA algorithms is now tested for Escherichia Coli data. This dataset contains the gene expression profiles obtained during transition of the sole carbon source from glucose to acetate (Kao et al., 2004). Out of 296 genes found to be relevant during the carbon source transition, 100 genes were separated so that the resulting network satisfies the NCA criteria. A total of 16 TFs were identified to be related to this experiment which are ArcA, CRP, CysB, FabR, FruR, GatR, IclR, LeuO, Lrp, NarL, PhoB, PurR, RpoE, RpoS, TrpR, TyrR. We perform subnetwork analysis to this data in order to estimate the transcription factor activities for the 16 TFs of interest. The downloaded network is divided into four subnetworks containing 81, 82, 85 and 88 genes, respectively. The number of TFs in each subnetwork is fixed to 20, where the aforementioned 16 TFs are included in all of them. The samples are taken at 5, 15, 30, 60 mins and then every hour until 6 hours. Multiple samples are taken at these instances which make a total of 25 time points. The advantage of using this data is that the ALS algorithm can be added to the performance evaluation because of its smaller subnetworks. ALS is known to have prohibitive computational complexity (Jacklin et al., 2012) and is included here only for the comparison of estimation accuracy. The reconstruction of TFAs is performed using the four algorithms and the average of the TFA estimates from four subnetworks is depicted in Figure 5. The results from ROBNCA, NI-NCA and ALS are in agreement for almost all of the TFAs. In addition, these estimates are also similar to those found in (Kao et al., 2004) except for a few TFAs. The reason for this small dissimilarity could be that, in this paper the estimates are obtained using the subnetworks whereas (Kao et al., 2004) use the complete network of 100 genes. For 5 out of the 16 TFs, namely GatR, Lrp, NarL, TrpR and TyrR, FastNCA is not able to recover the TFAs. Moreover, the TFAs predicted by ROBNCA are similar to those predicted by ALS which is the original solution as shown in Figure 5. It can therefore be inferred that ROBNCA estimates the TFAs more accurately than FastNCA.

The consistency of the algorithms is assessed for this experiment as well and the respective disagreement for each of the four algorithms is shown in Figure 4(b). FastNCA is again seen to incur the maximum disagreement. NI-NCA and ALS perform better than FastNCA; however, ROBNCA gives the least disagreement for the four estimates of TFAs and performs the most consistently out of all the algorithms.

3.2.2 E.coli Data

The performance of NCA algorithms is now tested for Escherichia Coli data. This dataset contains the gene expression profiles obtained during transition of the sole carbon source from glucose to acetate (Kao et al., 2004). Out of 296 genes found to be relevant during the carbon source transition, 100 genes were separated so that the resulting network satisfies the NCA criteria. A total of 16 TFs were identified to be related to this experiment which are ArcA, CRP, CysB, FabR, FruR, GatR, IclR, LeuO, Lrp, NarL, PhoB, PurR, RpoE, RpoS, TrpR, TyrR. We perform subnetwork analysis to this data in order to estimate the transcription factor activities for the 16 TFs of interest. The downloaded network is divided into four subnetworks containing 81, 82, 85 and 88 genes, respectively. The number of TFs in each subnetwork is fixed to 20, where the aforementioned 16 TFs are included in all of them. The samples are taken at 5, 15, 30, 60 mins and then every hour until 6 hours. Multiple samples are taken at these instances which make a total of 25 time points. The advantage of using this data is that the ALS algorithm can be added to the performance evaluation because of its smaller subnetworks. ALS is known to have prohibitive computational complexity (Jacklin et al., 2012) and is included here only for the comparison of estimation accuracy. The reconstruction of TFAs is performed using the four algorithms and the average of the TFA estimates from four subnetworks is depicted in Figure 5. The results from ROBNCA, NI-NCA and ALS are in agreement for almost all of the TFAs. In addition, these estimates are also similar to those found in (Kao et al., 2004) except for a few TFAs. The reason for this small dissimilarity could be that, in this paper the estimates are obtained using the subnetworks whereas (Kao et al., 2004) use the complete network of 100 genes. For 5 out of the 16 TFs, namely GatR, Lrp, NarL, TrpR and TyrR, FastNCA is not able to recover the TFAs. Moreover, the TFAs predicted by ROBNCA are similar to those predicted by ALS which is the original solution as shown in Figure 5. It can therefore be inferred that ROBNCA estimates the TFAs more accurately than FastNCA.

The consistency of the algorithms is assessed for this experiment as well and the respective disagreement for each of the four algorithms is shown in Figure 4(b). FastNCA is again seen to incur the maximum disagreement. NI-NCA and ALS perform better than FastNCA; however, ROBNCA gives the least disagreement for the four estimates of TFAs and performs the most consistently out of all the algorithms.
3.2.3 Computational Complexity Comparison

An important feature of all gene network reconstruction algorithms is the computational complexity incurred in their implementation. The computational complexity of estimating \( A \) in (29) at a particular iteration is approximately \( O(\sum_{n=1}^{N} (M - L_{n})^3 + (M - L_{n})^2) \), where \( (M - L_{n}) \) is the number of non-zero unknowns in the \( n^{th} \) column, which is usually very small. We now compare the computational complexity of the four algorithms using the subnetwork data from Yeast and E.coli. Average runtime calculated in seconds is summarized for four subnetworks of each data in Table 1. These experiments were performed on a Windows 7 system with a 1.90 GHz Intel Core i7 processor on a Matlab 7.10.0. It is noted that the run time of ROBNCA is comparable to that of FastNCA and is hundreds of times faster than NI-NCA algorithms for both of its implementations, i.e., involving linear programming and quadratic programming. Moreover, the run time for ROBNCA is far superior to that of the ALS, a direct consequence of the closed form solution derived for estimating the connectivity matrix. It can also be observed that the faster closed form solution for estimating \( A \) (29) provides additional savings over its predecessor (11).

Therefore, it can be inferred from these experiments on synthetic and real data sets that ROBNCA renders superior performance than the contemporary algorithms not only on the yardsticks of accuracy and reliability, but also in terms of computational complexity. The high computational complexity of NI-NCA far outweighs the benefits it offers in terms of consistency. FastNCA has the smallest run time out of all the algorithms but has poor reliability and is the least robust to the presence of outliers in the data.

4 CONCLUSION

In this work, we present ROBNCA, an algorithm for robust network component analysis for estimating the TFAs. The ROBNCA algorithm accounts for the presence of outliers by modeling them as an additional sparse matrix. A closed form solution available at each step of the iterative ROBNCA algorithm ensures faster and reliable performance. The performance of the proposed ROBNCA algorithm is compared with NI-NCA and FastNCA for synthetic as well real data sets by varying SNR, degrees of correlation and outlier concentration. It is observed that while FastNCA is computationally simpler, yet the TFA recovery is inaccurate and unreliable, a direct consequence of the sensitivity of its decomposition approach to the presence of outliers. The NI-NCA algorithm offers performance somewhat comparable to the ROBNCA algorithm, however, the ROBNCA algorithm is much more computationally efficient and does not require solving costly optimization problems. Therefore, the cumulative benefits of robustness to the presence of outliers, higher consistency and accuracy compared to the existing state-of-the-art algorithms, and much lower computational complexity make ROBNCA well-suited to the analysis of gene regulatory networks which invariably requires working with large data sets.

Funding: This work was supported by US National Science Foundation (NSF) grant 0915444 and QNRF-NPRP grant 09-874-3-235.

Conflict of Interest: none declared.

REFERENCES


