Microarrays **2015**, *4*, 596-617; doi:10.3390/microarrays4040596

OPEN ACCESS

microarrays

ISSN 2076-3905

www.mdpi.com/journal/microarrays

Review

An Overview of NCA-Based Algorithms for Transcriptional Regulatory Network Inference

Xu Wang [1], Mustafa Alshawaqfeh [1], Xuan Dang [1], Bilal Wajid [1], Amina Noor [2], Marwa Qaraqe [1] and Erchin Serpedin [1,*]

[1] Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA; E-Mails: xu.wang@tamu.edu (X.W.); mustafa.shawaqfeh@tamu.edu (M.A.); xuandt89@tamu.edu (X.D.); bilalwajidabbas@tamu.edu (B.W.); marwa@tamu.edu (M.Q.)

[2] Institute of Genomic Medicine, University of California San Diego, La Jolla, CA 92093, USA; E-Mail: amnoor@ucsd.edu

* Author to whom correspondence should be addressed; E-Mail: eserpedin@tamu.edu; Tel.: +1-979-458-2287; Fax: +1-979-862-4630.

Academic Editor: Heather J. Ruskin

Received: 1 September 2015 / Accepted: 11 November 2015 / Published: 16 November 2015

Abstract: In systems biology, the regulation of gene expressions involves a complex network of regulators. Transcription factors (TFs) represent an important component of this network: they are proteins that control which genes are turned on or off in the genome by binding to specific DNA sequences. Transcription regulatory networks (TRNs) describe gene expressions as a function of regulatory inputs specified by interactions between proteins and DNA. A complete understanding of TRNs helps to predict a variety of biological processes and to diagnose, characterize and eventually develop more efficient therapies. Recent advances in biological high-throughput technologies, such as DNA microarray data and next-generation sequence (NGS) data, have made the inference of transcription factor activities (TFAs) and TF-gene regulations possible. Network component analysis (NCA) represents an efficient computational framework for TRN inference from the information provided by microarrays, ChIP-on-chip and the prior information about TF-gene regulation. However, NCA suffers from several shortcomings. Recently, several algorithms based on the NCA framework have been proposed to overcome these shortcomings. This paper first overviews the computational principles behind NCA, and then, it surveys the state-of-the-art NCA-based algorithms proposed in the literature for TRN reconstruction.

Keywords: gene; transcription factor; transcriptional regulatory network; network component analysis

1. Introduction

For every soccer team, the coach is responsible for directing the team to victory. The primary aim is to score as many goals as possible and, at the same time, thwart the other team from doing the same. The coach may choose some players over others. Even among team members, some players attack, others defend, whereas some are good as half-back players. Moreover, the core players that form the playing team do not remain the same throughout the game. Keeping in mind the dynamics of the game, the coach may direct some players to replace others, ensuring the primary aim of the game, to win, remains intact. In close comparison with this framework, the cell does not operate very differently. The coach of the cell, the DNA within the nucleus, directs different team members, transcription factors (TFs) and genes, to execute cellular functions and complex biological processes, which help the cell to adapt to varying dynamics, including external stimuli, as well as internal changes. The team members, TFs and genes, work together to express or suppress different metabolic pathways at different instances of the cell's life. Particularly, these TFs contain DNA binding domains that allow them to bind to specific regions of DNA, called promoters [1]. By binding to these promoters, TFs initiate the process of converting genes into proteins. Transcription factor activities (TFAs) refer not only to the connectivity of any particular TF, but also to its level of activity. The connectivity of a particular TF informs its team members to collaborate in order to regulate RNA polymerase, which in its turn controls in terms of expressing or suppressing genes. TFAs cannot be measured directly; rather, they can be inferred from gene expression data. Furthermore, transcriptional regulatory networks (TRNs) represent interactions between genes and TFAs within a cell and offer a global perspective in the cellular behavior. Understanding the structure of TRNs and estimating TFAs provide insight into the cellular dynamics present in healthy and diseased tissues and organs and hold the potential to help in diagnosing, characterizing and determining cures for various diseases [2].

In the literature, several computational frameworks have been proposed to analyze regulatory interactions, which are briefly summarized below. The first class models the TRN as a dynamic system. Particularly, [3] and [4] describe gene expression as a linear and continuous time first-order differential equation. On the other hand, Boolean network models [5,6] quantize gene expressions by only two discrete levels: ON and OFF. The expression level of each gene is the Boolean function of the expression levels of other genes. These methods are generally performed using a small number of time series data and, thus, lead to an under-determined problem [7]. Another approach for TRN reconstruction is referred to as the co-expression (or relevance) networks, in which two genes are connected if the similarity between them exceeds a predefined threshold. Examples of similarity measures used in constructing relevance networks include correlation [8] and mutual information [9,10]. Relevance networks are helpful to understand the fundamental topological features of biological networks, but they do not infer causal relations among genes. The algorithms falling into the third category are commonly described as probabilistic graphical models [11–13], which include Gaussian graphical models (GGMs) and Bayesian

networks (BNs). In GGMs, the network or graph is constructed based on the notion of conditional independence, and two genes are connected if and only if they are independent given the expression levels of all other genes. GGMs are formulated using undirected graphs and represent an example of full conditional models, since the conditional dependency is considered with respect to all other genes. On the other hand, BNs entail directed acyclic graphs, and the conditional dependency is measured with respect to all subsets of the other genes [11,14]. One limitation of probabilistic graphic models is that they have strong assumptions on the joint distribution that prevent representing or interpreting some biological relationships. For example, cyclic graphs are not allowed in the BN framework. In this way, it ignores self-feedback loops among genes that are natural features in genetic networks. Additionally, the applications of probabilistic graphic models are generally limited to the network with the number of experimental measurements significantly larger than the number of genes, since analyzing the structure of large-scaled genetic networks using probabilistic graphic models is highly complex. Besides dynamic models, co-expression networks and probabilistic graphical models, structural equation modeling (SEM) also represents a widely-used technique for TRN inference [15,16]. Generally, an SEM consists of a structural model and a measurement model. The structural model describes the causal relations between the latent variables, while the measurement model depicts the relations between latent variables and observed measurements.

Recently, studies dedicated to TRN inference using the network component analysis (NCA) technique have begun to emerge in the literature [17]. NCA establishes a parameter estimation problem and reconstructs TRNs following a statistical signal processing viewpoint. Since NCA-based algorithms do not require time series data, they can collect the experimental data from different time intervals and combine them to increase the samples size and prevent the under-determination problem. Even with a limited number of experiments, NCA-based algorithms are still able to reconstruct TRNs with a large number of TFs and genes (See Section 3.3 for more details). Moreover, NCA-based algorithms take advantage of some prior knowledge about the connectivity patterns of the genetic network, which is becoming available via high-throughput experiments [18] or data mining of interaction information [19–21]. The assumed mathematical model for NCA is represented by the following system of linear equations [17]:

$$\mathbf{X} = \mathbf{AS} + \mathbf{\Gamma} \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{N \times K}$ represents the log ratios of expression values of N genes at K time points of the microarray dataset, $\mathbf{A} \in \mathbb{R}^{N \times M}$ denotes the connectivity strength between N genes and M TFs, $\mathbf{S} \in \mathbb{R}^{M \times K}$ stands for the activities of M TFs at K time points and $\mathbf{\Gamma} \in \mathbb{R}^{N \times K}$ represents the measurement noise. Examples of two TRNs with six genes and four TFs, but different connectivity topologies, are shown in Figure 1.

Generally, in Equation (1), $\mathbf{X}$ cannot be uniquely decomposed as the product of two matrices $\mathbf{A}$ and $\mathbf{S}$, unless further constraints are imposed. Principal component analysis (PCA) [22] and independent component analysis (ICA) [23] represent two conventional statistical algorithms that can provide valid solutions provided that the input signals present in $\mathbf{S}$ are independent and/or orthogonal. However, such an assumption generally does not hold for biological signals in practice. Accounting for this fact, Liao *et al.* [17] proposes NCA, which incorporates the prior information about TF-gene regulation, to infer TRNs. As will be discussed in detail in Section 2, NCA is an iterative computational

algorithm that ensures the uniqueness of decomposition solutions. Due to some drawbacks of NCA, such as the stringent conditions required to apply NCA, several alternative NCA-based algorithms have been proposed in the literature to improve NCA from different perspectives, such as less restrictive assumptions, lower computational complexity and higher robustness against noise, outliers and modeling errors.

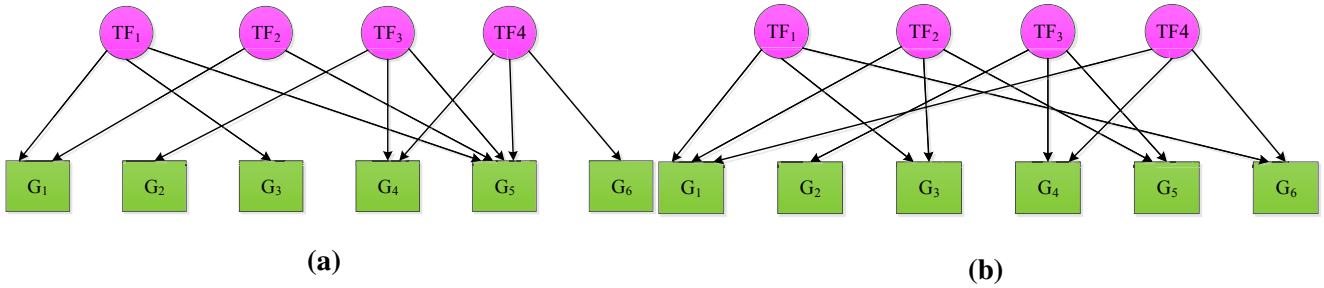


Figure 1. Examples of two transcription regulatory networks (TRNs) with six genes and four transcription factors (TFs), but different connectivity topologies.

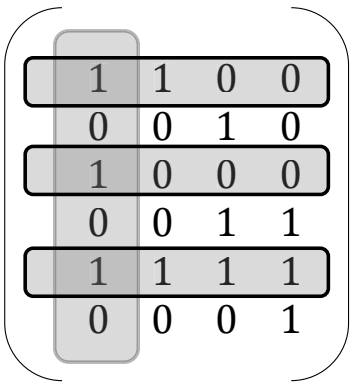
The rest of the paper, which proposes to provide a review of the major algorithms reported for NCA, is organized as follows. Section 2 introduces the NCA framework and the mathematical details of the NCA algorithm. Extensions of NCA are presented in Section 3. These extensions still rely on the NCA algorithm, but improve the applicability range of NCA by requiring less stringent assumptions. In Section 4, alternative NCA-based algorithms proposed in the literature for TRN inference are surveyed. A few illustrative computer simulation results highlighting the performance of major NCA algorithms are presented in Section 5. In addition, the comparison of these algorithms and some recommendations on how to choose the appropriate algorithm are discussed in Section 6 based on the simulation results in Section 5. Finally, Section 7 summarizes the content of this paper.

2. NCA

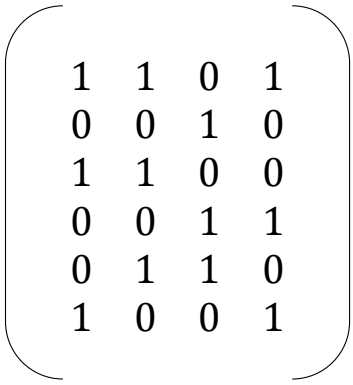
In the case when both matrices $\mathbf{A}$ and $\mathbf{S}$ are unknown, the decomposition problem in Equation (1) admits an infinite number of solutions. Fortunately, prior information is becoming available for many biological systems, e.g., ChIP-on-chip (ChIP-on-chip (also known as ChIP-chip) represents a technology that combines chromatin immunoprecipitation ("ChIP") with a DNA microarray ("chip")) data indicate whether a certain gene interacts with a certain TF. This prior information is incorporated within NCA mathematically via the constraint $A(I) = 0$, where I presents the indices of zero elements in the connectivity matrix $\mathbf{A}$, indicating a certain level of connectivity information. NCA requires three identification criteria to ensure a unique solution up to a scalar ambiguity:

1 The connectivity matrix $\mathbf{A}$ must be full-column rank.
2 If a column of $\mathbf{A}$ is removed along with all of the rows corresponding to the nonzero entries of the removed column, the remaining matrix must still be full-column rank.
3 The TFA matrix $\mathbf{S}$ must have full row rank.

To test whether the system meets the above-mentioned first two criteria, matrix $\mathbf{A}$ must be first initialized based on the prior knowledge available about connectivity. Specifically, a_{ij} is assigned to zero if $(i, j) \in I$, and it assumes any arbitrary nonzero value otherwise. Once $\mathbf{A}$ is initialized, matrix $\mathbf{A}$ is tested to see if it presents a full-column rank. Then, we sequentially remove each column of $\mathbf{A}$, as well as the genes connected to the removed TF and test whether the remaining reduced matrix still presents full-column rank. Consider TRNs in Figure 1 as an example. The initialized connectivity matrices for Figure 1a,b are illustrated in Figure 2a,b, respectively. The initialized connectivity matrix in Figure 2a is not identifiable, since the reduced matrix obtained by removing the first column along with the first, third and fifth rows is not full-column rank. This condition violates the second criterion of NCA. The initialized connectivity matrix in Figure 2b, on the other hand, satisfies all three identification criteria. In terms of the third criterion, it cannot be tested *a priori*, but it implies that the number of TFs must be less than or equal to the number of time points, *i.e.*, $M \leq K$. This rank criterion is verified after $\mathbf{S}$ is simulated using NCA [17].



(a) Initialized $\mathbf{A}$ for Figure 1a



(b) Initialized $\mathbf{A}$ for Figure 1b

Figure 2. An example of (**a**) a non-identifiable pattern and (**b**) an identifiable pattern.

NCA aims to solve the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{S}} ||\mathbf{X} - \mathbf{AS}||_F^2, \\ &\text{s.t. } \mathbf{A}(I) = 0, \end{aligned} \tag{2}$$

where $||\cdot||_F$ denotes the Frobenius norm. NCA employs an alternate least-squares (ALS) approach to iteratively update $\mathbf{A}$ and $\mathbf{S}$. At iteration j, given $\mathbf{A}(j-1)$, *i.e.*, the value of $\mathbf{A}$ at iteration $(j-1)$, the estimate of $\mathbf{S}(j)$ is obtained by solving the following least-squares (LS) problem:

$$\begin{aligned} &\mathbf{S}(j) = \arg\min_{\mathbf{S}} ||\mathbf{X} - \mathbf{A}(j-1)\mathbf{S}||_F^2 \\ &\text{s.t. } s_{i,j}^{(l)} \leq s_{i,j} \leq s_{i,j}^{(u)}, \end{aligned} \tag{3}$$

where the constraint is included to ensure that the elements of $\mathbf{S}$ remain in the domain of biologically-sensitive values [17]. The optimization problem Equation (3) can be solved by standard

convex optimization tools, such as the interior point method [24]. Once $\mathbf{S}(j)$ is obtained, the next step is to update $\mathbf{A}(j)$ via the following optimization problem:

$$\begin{aligned} \mathbf{A}(j) &= \arg\min_{\mathbf{A}} ||\mathbf{X} - \mathbf{A}\mathbf{S}(j)||_F^2 \\ \text{s.t. } & \mathbf{A}(I) = 0,\ a_{i,j}^{(l)} \leq a_{i,j} \leq a_{i,j}^{(u)}, \end{aligned} \tag{4}$$

where the constraint $a_{i,j}^{(l)} \leq a_{i,j} \leq a_{i,j}^{(u)}$ is also used to constrain the domain of $\mathbf{A}$. Particularly, eliminating the zero elements in $\mathbf{A}$ removes the connectivity constraint $\mathbf{A}(I) = 0$. This leads to a new least-squares problem with a lesser number of variables, which can also be solved using the same method employed to solve Equation (3). If the decrease in the total least-squares error after updating $\mathbf{A}$ is above a preset value e, the algorithm keeps running. Otherwise, it stops. A diagram illustrating the operation of the NCA is shown in Figure 3. Simulation results in [17] demonstrate that NCA was successfully applied to the microarray data generated from yeast *Saccharomyces cerevisiae*, and the activities of various TFs during the cell cycle were reconstructed.

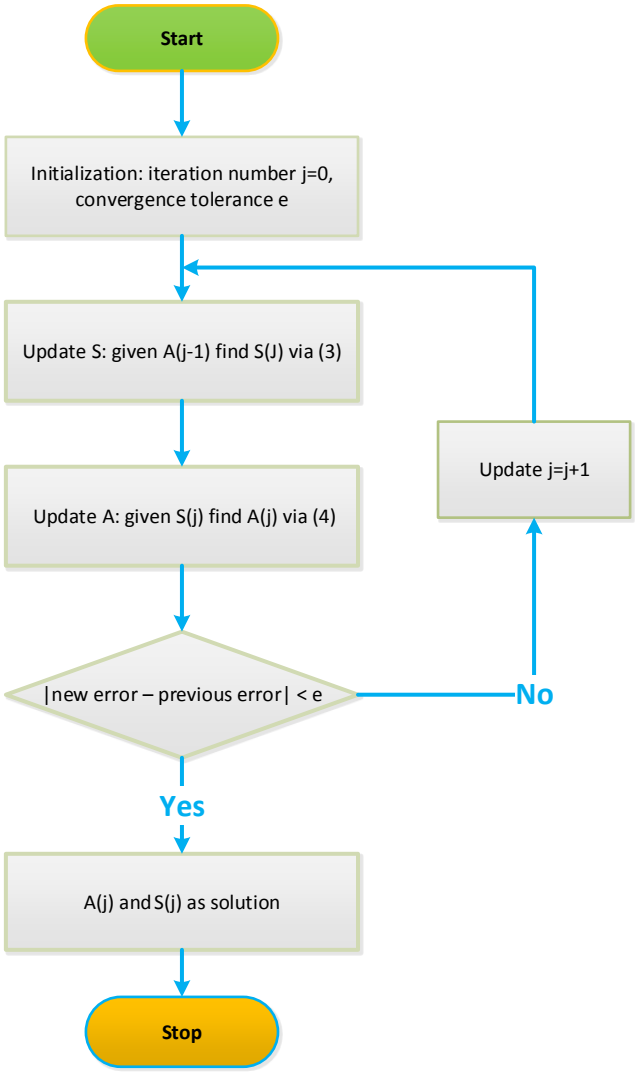


Figure 3. Network component analysis (NCA) algorithm.

3. Extensions of NCA

Despite its successful implementation in yeast data, NCA exhibits several shortcomings, which prevent its application to a wide class of regulatory network inference problems. In the literature, several papers have been proposed to tackle these issues. In this section, we focus on several improvements for NCA proposed recently in the literature. In these works, the core estimation methods are identical to NCA, but some enhancements have been implemented to make the NCA algorithm more applicable to various setups.

3.1. Motif-Directed NCA

In the original NCA work [17], the prior information about the connectivity matrix, *i.e.*, $\mathbf{A}(I)$, is provided by high-throughput experiments. However, the high-throughput ChIP-on-chip data are not available for some common species, such as rodents and humans [25]. With respect to this fact, Wang *et al*. [25] proposed a motif-directed NCA (mNCA) algorithm, which incorporates the motif information to obtain the prior network structure information and to infer TRNs. Due to the fact that the regulation between TFs and genes occurs only after TFs bind to the DNA sequence motifs in the gene's promoter region [25], the authors incorporate the motif information to recover the interaction between TFs and genes. Moreover, since the prior topology information, either from ChIP-on-chip data or motif analysis, comes from biological experiments, it may contain many false positives/negatives. Thus, a stability analysis is further proposed in [25] to extract stable TFAs from the NCA algorithm. Specifically, the authors of [25] intentionally perturb the connectivity information and use the Pearson correlation coefficient as a stability measurement to determine whether the estimated TFAs are stable or not. Experimental results on muscle regeneration microarray data demonstrate that mNCA is able to reveal important TFAs, as well as their connectivity strength to corresponding genes.

3.2. Generalized NCA

The work in [26] proposed the generalized NCA (gNCA) in an attempt to improve the NCA criteria. gNCA extends the system identification criteria required by NCA by additionally incorporating the prior information about regulatory matrix $\mathbf{S}$, such as the regulatory information obtained from regulatory gene knockouts (a gene knockout (KO) refers to a genetic technique through which one or more genes from an organism are made inoperative ("knocked out")) [26]. Thus, for the gNCA criteria to guarantee a unique decomposition solution, they require a full column rank condition for $\mathbf{A}$, a full row rank condition for $\mathbf{S}$ and an additional condition that preserves the essential features of $\mathbf{A}$ and $\mathbf{S}$. In this way, given the topology information about $\mathbf{S}$, the uniqueness of the decomposition problem might still be ensured by alternatively checking the gNCA criteria, even if the connectivity structure of $\mathbf{A}$ does not satisfy the NCA criteria. Even when the connectivity topology satisfies the NCA criteria, gNCA reduces the number of parameters to be estimated by combining the prior information about $\mathbf{S}$.

3.3. Revised NCA

The work in [27] also focuses on enhancing the NCA criteria. The work in [27] proposed revised NCA (NCAr), where the third criterion of NCA is revised to improve the applicability of NCA. As discussed earlier, to ensure a unique solution for the matrix factorization problem, the third criterion of NCA requires the matrix $\mathbf{S}$ to have full row rank, which implies that the number of TFs must be less than or equal to the number of experiments. This requirement significantly limits the sample size of TFs. The work in [27] revises the third criterion of NCA based on the observation that most of the genes are only regulated by a smaller number of TFs than the total number of TFs (*i.e.*, the connectivity matrix $\mathbf{A}$ is row-wise sparse). In particular, this condition, instead of being associated with the rank properties of matrix $\mathbf{S}$, is related to the rank properties of reduced-size matrices. Particularly, it requires that the number of experiments for each gene be greater than or equal to the number of TFs regulating that gene. The revised criterion enables NCA to be applicable to a wider class of TRN inference problems, since the number of TFs regulating a gene is generally less than five or six [27]. In this way, a large dimensional regulatory network can be uniquely inferred, even in the presence of a limited number of experiments.

3.4. Generalized-Framework NCA

The original NCA work requires the biological system to satisfy all three criteria to ensure a unique decomposition up to a scaling factor. However, NCA only checks the compliance for the initialized matrix $\mathbf{A}$. It may occur that the derived matrix $\mathbf{A}$ at certain iterations violates the NCA criteria. The work in [28] generalizes the NCA criteria, such that the system identification can be determined directly from the connectivity (topology) information, rather than checking the rank properties of the unknown connectivity matrix $\mathbf{A}$. In other words, if a certain connectivity topology, *i.e.*, $\mathbf{A}(I_0)$, meets the newly-derived conditions, then all matrices $\mathbf{A} \in \mathbf{A}(I_0)$ satisfy the first and second criterion of NCA, and thus, they guarantee the feasibility of $\mathbf{A}$ during each iteration of the NCA algorithm. To deal with the issue that the connectivity topology does not satisfy the newly-derived conditions or the TF matrix does not satisfy the third criterion of NCA (for example, when $M > K$, the linear independence of TFs is violated), the authors in [28] alternatively seek to infer subnetworks by removing the selected TF node together with all of its associated genes until all of the system identification criteria of the reduced subnetwork are verified. The resulting algorithm is referred to as generalized-framework NCA (gfNCA) [28].

4. Alternative NCA-Based Algorithms

In this section, we review some alternative NCA-based algorithms that were also recently reported in the literature. Different from the algorithms discussed in Section 3, where mNCA, gNCA, NCAr and gfNCA utilize the NCA algorithm to infer TRNs, the algorithms discussed in this section focus on designing more efficient algorithms to estimate the matrices $\mathbf{A}$ and $\mathbf{S}$ in the NCA system model Equation (1). These algorithms can be roughly classified into two classes, namely the iterative and the non-iterative class.

4.1. Iterative NCA Algorithms

As described in Section 2, NCA adopts the ALS approach to iteratively update matrices $\mathbf{A}$ and $\mathbf{S}$. Therefore, NCA, along with all of the algorithms that employ ALS, such as mNCA, gNCA, NCAr and gfNCA, is an iterative method. Another example of iterative methods, referred to as robust NCA (ROBNCA), is reviewed next.

4.1.1. Robust NCA

ROBNCA [29] is a robust NCA-based approach that tries to cope with the possible noise and outliers present in the microarray data due to erroneous measurements and/or the abnormal response of genes [30]. To counteract the presence of outliers, the system model of TRNs is formulated as:

$$\mathbf{X} = \mathbf{AS} + \mathbf{O} + \mathbf{\Gamma} \tag{5}$$

where matrix $\mathbf{O}$ models explicitly the presence of outliers. Since typically, only a few outliers exist, the outlier matrix $\mathbf{O}$ represents a column-sparse matrix. Accounting for the sparsity of matrix $\mathbf{O}$, ROBNCA aims to solve the following optimization problem:

$$\begin{aligned} \{\hat{\mathbf{A}}, \hat{\mathbf{S}}, \hat{\mathbf{O}}\} &= \arg \min_{\mathbf{A},\mathbf{S},\mathbf{O}} ||\mathbf{X} - \mathbf{AS} - \mathbf{O}||_F^2 + \lambda_0 ||\mathbf{O}||_0 \\ &\text{s.t.} \quad \mathbf{A}(I) = 0, \end{aligned} \tag{6}$$

where $||\mathbf{O}||_0$ denotes the number of nonzero columns in $\mathbf{O}$ and λ_0 is a penalization parameter used to control the extent of sparsity of $\mathbf{O}$. Due to the intractability and high complexity of computing the l_0-norm-based optimization problem, the problem Equation (6) is relaxed to:

$$\begin{aligned} \{\hat{\mathbf{A}}, \hat{\mathbf{S}}, \hat{\mathbf{O}}\} &= \arg \min_{\mathbf{A},\mathbf{S},\mathbf{O}} ||\mathbf{X} - \mathbf{AS} - \mathbf{O}||_F^2 + \lambda_2 ||\mathbf{O}||_{2,c} \\ &\text{s.t.} \quad \mathbf{A}(I) = 0 \end{aligned} \tag{7}$$

where $||\mathbf{O}||_{2,c}$ stands for the column-wise l_2-norm sum of $\mathbf{O}$, *i.e.*, $||\mathbf{O}||_{2,c} = \sum_{k=1}^{K} ||\mathbf{o}_k||_2$, where $\mathbf{o}_k$ denotes the k-th column of $\mathbf{O}$. Since the optimization problem Equation (7) is not jointly convex with respect to $\{\mathbf{A}, \mathbf{S}, \mathbf{O}\}$, an iterative algorithm is performed in [29] to optimize Equation (7) with respect to one parameter at a time.

Towards this end, the ROBNCA algorithm at iteration j assumes that the values of $\mathbf{A}$ and $\mathbf{O}$ from iteration $(j-1)$, *i.e.*, $\mathbf{A}(j-1)$ and $\mathbf{O}(j-1)$, are known. Defining $\mathbf{Y}(j) = \mathbf{X} - \mathbf{O}(j-1)$, the update of $\mathbf{S}(j)$ can be calculated by carrying out the optimization problem:

$$\mathbf{S}(j) = \arg \min_{\mathbf{S}} ||\mathbf{Y}(j) - \mathbf{A}(j-1)\mathbf{S}||_F^2$$

which admits a closed-form solution. The next step of ROBNCA at iteration j is to update $\mathbf{A}(j)$ while fixing $\mathbf{O}$ and $\mathbf{S}$ to $\mathbf{O}(j-1)$ and $\mathbf{S}(j)$, respectively. This can be performed via the following optimization problem:

$$\begin{aligned} \mathbf{A}(j) &= \arg \min_{\mathbf{A}} ||\mathbf{Y}(j) - \mathbf{AS}(j)||_F^2 \, . \\ &\text{s.t.} \quad \mathbf{A}(I) = 0 \end{aligned} \tag{8}$$

The problem Equation (8) was also considered in the original NCA paper [17] in which a closed-form solution was not provided. Since this optimization problem has to be conducted at each iteration, a closed-form solution is derived in ROBNCA using the re-parameterization of variables and the Karush–Kuhn–Tucker (KKT) conditions to reduce the computational complexity and improve the convergence speed of the original NCA algorithm. In the last step, the iterative algorithm estimates the outlier matrix $\mathbf{O}$ by using the iterates $\mathbf{A}(j)$ and $\mathbf{S}(j)$ obtained in the previous steps, *i.e.*,

$$\mathbf{O}(j) = \arg\min_{\mathbf{o}_k} ||\mathbf{C}(j) - \mathbf{O}||_2^2 + \lambda_2 ||\mathbf{O}||_{2,c} \tag{9}$$

where $\mathbf{C}(j) = \mathbf{X} - \mathbf{A}(j)\mathbf{S}(j)$. The solution to Equation (9) is obtained by using standard convex optimization techniques, and it can be expressed in a closed form.

It can be observed that at each iteration, the updates of matrices $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{O}$ all assume a closed-form expression, and it is this aspect that significantly reduces the computational complexity of ROBNCA when compared to the original NCA algorithm. In addition, the term $\lambda_2||\mathbf{O}||_{2,c}$ guarantees the robustness of the ROBNCA algorithm against outliers. Simulation results in [29] also show that ROBNCA estimates TFAs and the TF-gene connectivity matrix with a much higher accuracy in terms of normalized mean square error than FastNCA [31] and non-iterative NCA (NINCA) [32], irrespective of varying noise, the level of correlation and outliers.

4.2. Non-Iterative NCA Algorithms

This section presents four fundamental non-iterative methods, namely, fast NCA (FastNCA) [31], positive NCA (PosNCA) [33], non-negative NCA (nnNCA) [34] and non-iterative NCA (NINCA) [32]. These algorithms employ the subspace separation principle (SSP) and overcome some drawbacks of the existing iterative NCA algorithms. FastNCA utilizes SSP to preprocess the noise in gene expression data and to estimate the required orthogonal projection matrices. On the other hand, in PosNCA, nnNCA and NINCA, the subspace separation principle is adopted to reformulate the estimation of the connectivity matrix as a convex optimization problem. This convex formulation provides the following benefits: (i) it ensures a global solution; (ii) it allows usage of efficient convex programming techniques, like the interior point method [24]; and (iii) it offers the flexibility of adding additional convex constraints. Since SSP represents the core technique of these non-iterative NCA-based algorithms, this important concept is first explained in the next subsection.

4.2.1. Subspace Separation Principle

Assume matrix $\mathbf{X}$ is decomposed into the sum of two other matrices $\mathbf{X} = \mathbf{B} + \mathbf{\Gamma}$, where $\mathbf{X} \in \mathbb{R}^{N\times K}(K < N)$ stands for the observed data, $\mathbf{B} \in \mathbb{R}^{N\times K}$ represents the true signal and $\mathbf{\Gamma} \in \mathbb{R}^{N\times K}$ denotes the noise matrix. SSP attempts to partition the range space of $\mathbf{X}$ into two subspaces, where one subspace is spanned by the source signal and the other subspace is spanned by noise. One possible way to do this is via singular value decomposition (SVD). Specifically, the SVD of $\mathbf{X}$ takes the form:

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T = \sum_{i=1}^{K} \sigma_k \mathbf{u}_k \mathbf{v}_k^T \tag{10}$$

where the singular values are arranged in a descending order $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_K \geq 0$. In the situation where the noise level is low and the signal matrix is not ill-conditioned, the significant singular values (singular values with larger values) correspond to the signal subspace, and the remaining negligible singular values correspond to the noise subspace. Under the assumption of keeping ($L < K$) singular values as the signal singular values, the SVD of Equation (10) can be decomposed into two components, corresponding to the signal ($\mathbf{X}_L$) and noise component ($\mathbf{X}_R$), respectively.

$$\mathbf{X} = \underbrace{\mathbf{U}_L \Sigma_L \mathbf{V}_L^T}_{\mathbf{X}_L} + \underbrace{\mathbf{U}_R \Sigma_R \mathbf{V}_R^T}_{\mathbf{X}_R} \tag{11}$$

The first term in Equation (11), *i.e.*, $\mathbf{X}_L$, is called the L-rank Eckart–Young–Mirsky (EYM) approximation of $\mathbf{X}$ and represents the higher signal-to-noise ratio (SNR) representation of $\mathbf{X}$. Matrix $\mathbf{\Sigma}_L$ is a diagonal matrix, and it contains the first L singular values corresponding to the signal component; and $\mathbf{U}_L$ and $\mathbf{V}_L$ correspond to the left and right singular vectors, respectively. Similarly, $\mathbf{\Sigma}_R$ is a diagonal matrix containing the last $K - L$ singular values corresponding to the noise part, and $\mathbf{U}_R$ and $\mathbf{V}_R$ correspond to the left and right noise singular vectors, respectively. Hence, the space of the observed measurements is approximately decomposed into two separate subspaces: signal and noise subspace, respectively. If we still further denote $\mathbf{X}$ as the product of the two matrices $\mathbf{A} \in \mathbb{R}^{N \times M}$ and $\mathbf{S} \in \mathbb{R}^{M \times K}$, *i.e.*, $\mathbf{X} = \mathbf{AS} + \mathbf{\Gamma}$ as shown in Equation (1), it is shown in [32] that $\mathbf{U}_R$ represents a robust approximation of the left null space of $\mathbf{A}$ in the case $L = M$.

4.2.2. FastNCA

FastNCA [31] provides a closed form solution to NCA, and it overcomes in the same time the speed limitations of the original NCA. FastNCA employs a series of matrix partitionings and orthogonal projections to estimate the connectivity matrix on a column-by-column basis. Once matrix $\mathbf{A}$ is estimated, matrix $\mathbf{S}$ is estimated by a direct application of the least-squares principle:

$$\mathbf{S} = A^{\dagger}\mathbf{X} \tag{12}$$

Next, a detailed explanation of the FastNCA approach to estimate the first column of $\mathbf{A}$, *i.e.*, $\mathbf{a}_1$, in both the noiseless and noisy case is presented. The same analysis can be repeated for the remaining columns, since the columns in $\mathbf{A}$ can be re-ordered by appropriately changing the rows of $\mathbf{S}$.

In the ideal case where no noise exists, the system model in Equation (1) assumes the form:

$$\mathbf{X} = \mathbf{AS} \tag{13}$$

Without loss of generality, the elements in $\mathbf{a}_1$ are rearranged, such that the nonzero elements are located at the beginning of the vector and the zero elements are placed at the end:

$$\mathbf{a}_1 = \begin{bmatrix} \widetilde{\mathbf{a}}_1 \\ 0 \end{bmatrix} \tag{14}$$

Then, Equation (13) can be partitioned as:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_c \\ \mathbf{X}_r \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{a}}_1 & \mathbf{A}_c \\ 0 & \mathbf{A}_r \end{bmatrix} \begin{bmatrix} \mathbf{s}_1^T \\ \mathbf{S}_r \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{a}}_1 \mathbf{s}_1^T + \mathbf{A}_c \mathbf{S}_r \\ \mathbf{A}_r \mathbf{S}_r \end{bmatrix} \tag{15}$$

Taking the transpose of Equation (20) results in:

$$\mathbf{X}_c^T = \mathbf{s}_1 \widetilde{\mathbf{a}}_1^T + \mathbf{S}_r^T \mathbf{A}_c^T \tag{16}$$

$$\mathbf{X}_r^T = \mathbf{S}_r^T \mathbf{A}_c^T \tag{17}$$

Extracting $\widetilde{\mathbf{a}}_1$ is possible if the term $\mathbf{S}_r^T \mathbf{A}_c^T$ in Equation (16) can be eliminated. This can be determined by using an orthogonal matrix projection. Assuming the orthogonal projection matrix onto $\mathbf{S}_r^T$ is $\mathbf{P}^{\perp}_{\mathbf{S}_r^T}$ and multiplying Equation (16) by $\mathbf{P}^{\perp}_{\mathbf{S}_r^T}$ leads to:

$$\mathbf{P}^{\perp}_{\mathbf{S}_r^T} \mathbf{X}_c^T = \mathbf{P}^{\perp}_{\mathbf{S}_r^T} \mathbf{s}_1 \widetilde{\mathbf{a}}_1^T = \widetilde{\mathbf{s}}_1 \widetilde{\mathbf{a}}_1^T \tag{18}$$

where $\tilde{s}_1 = \mathbf{P}^{\perp}_{\mathbf{S}_r^T} \mathbf{s}_1$. Therefore, the challenge is to find $\mathbf{P}^{\perp}_{\mathbf{S}_r^T}$. From Equation (17), the range space of $\mathbf{S}_r^T$ and the left null space of $\mathbf{X}_r^T$ are the same, since $\mathbf{S}_r^T$ is full column rank (the third NCA criterion). Furthermore, $\mathbf{A}_r^T$ is full row rank (first NCA criterion). Hence, $\mathbf{P}^{\perp}_{\mathbf{S}_r^T} = \mathbf{P}^{\perp}_{\mathbf{X}_r^T}$. $\mathbf{P}^{\perp}_{\mathbf{S}_r^T} \mathbf{X}_c^T$ is known. Therefore, a rank-one factorization of $\mathbf{P}^{\perp}_{\mathbf{S}_r^T} \mathbf{X}_c^T$ yields an estimate of $\widetilde{\mathbf{a}}_1^T$ up to a scalar ambiguity, and it represents the first right singular vector of $\mathbf{P}^{\perp}_{\mathbf{S}_r^T} \mathbf{X}_c^T$.

In the noise case, as shown in Equation (1), FastNCA handles the noise in the gene expression measurements by using the concept of subspace separation. This is done by replacing the noisy observation data $\mathbf{X}$ with its L-rank EYM approximation $\mathbf{X}_L$ (see Equation (11)). In this way, it follows that:

$$\mathbf{X} = \mathbf{U}_L \Sigma_L \mathbf{V}_L^T$$

and moreover:

$$\mathbf{W} = \mathbf{U}_L = \mathbf{X}\mathbf{V}_L \Sigma_L^{-1} = (\mathbf{A}\mathbf{S} + \mathbf{\Gamma})\mathbf{V}_L \Sigma_L^{-1} = \mathbf{A}\widetilde{\mathbf{S}} + \widetilde{\mathbf{\Gamma}} \tag{19}$$

where $\mathbf{U}_L$ is represented by $\mathbf{W}$ for simplicity, $\widetilde{\mathbf{S}} = \mathbf{S}\mathbf{V}_L \Sigma_L^{-1}$ and $\widetilde{\mathbf{\Gamma}} = \mathbf{\Gamma}\mathbf{V}_L \Sigma_L^{-1}$.

Partitioning $\mathbf{W}$ in the same way as in Equation (20) yields:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_c \\ \mathbf{W}_r \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{a}}_1 & \mathbf{A}_c \\ 0 & \mathbf{A}_r \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{s}}_1^T \\ \widetilde{\mathbf{S}}_r \end{bmatrix} + \begin{bmatrix} \widetilde{\mathbf{\Gamma}}_c \\ \widetilde{\mathbf{\Gamma}}_r \end{bmatrix} \tag{20}$$

which further results in:

$$\mathbf{W}_c^T = \widetilde{\mathbf{s}}_1 \widetilde{\mathbf{a}}_1^T + \widetilde{\mathbf{S}}_r^T \mathbf{A}_c^T + \widetilde{\mathbf{\Gamma}}_c^T \tag{21}$$

$$\mathbf{W}_r^T = \widetilde{\mathbf{S}}_r^T \mathbf{A}_r^T + \widetilde{\mathbf{\Gamma}}_r^T \tag{22}$$

Due to noise, a direct repetition of the noiseless case analysis is not applicable, because $\mathbf{P}^{\perp}_{\widetilde{\mathbf{S}}_r^T} \neq \mathbf{P}^{\perp}_{\mathbf{W}_r^T}$. The subspace separation principle provides an estimate of $\mathbf{P}^{\perp}_{\widetilde{\mathbf{S}}_r^T}$. Consider the following SVD of $\mathbf{W}_r$:

$$\mathbf{W}_r = \mathbf{U}_1 \Sigma_1 \mathbf{V}_1^T + \mathbf{U}_0 \Sigma_0 \mathbf{V}_0^T \tag{23}$$

where Σ_1 and Σ_0 contain the leading $M-1$ and last $L-M+1$ singular values, respectively. Then, an estimate of $\widehat{\mathbf{P}}^{\perp}_{\widetilde{\mathbf{S}}_r^T}$ is given by:

$$\widehat{\mathbf{P}}^{\perp}_{\widetilde{\mathbf{S}}_r^T} = \mathbf{V}_0 \mathbf{V}_0^T \tag{24}$$

Similar to the noiseless case, $\widetilde{\mathbf{a}}_1^T$ can be obtained by applying a rank-one factorization for $\widehat{\mathbf{P}}^{\perp}_{\widetilde{\mathbf{S}}_r^T} \mathbf{W}_c^T$.

4.2.3. Positive NCA, Non-Negative NCA and Non-Iterative NCA

PosNCA [33] modifies the original NCA algorithm in two regards. The first aspect pertains to evaluating matrix $\mathbf{A}$ via a convex optimization (instead of ALS, as in the original NCA). The second aspect refers to the addition of the positivity constraints on all of the nonzero elements in the connectivity matrix. This assumption has a biological support [35]. The positivity constraint is valid only in situations where all TFs play the same role (*i.e.*, activating or deactivating) on their corresponding targeted genes. If all of the TFs regulate the genes in a negative way (deactivating), the positivity assumption is maintained by multiplying the activity value in the signal matrix by the value -1. This positivity assumption is a convex constraint, which perfectly integrates with the convex formulation of the problem.

The essence of the formulation of PosNCA as a convex optimization problem relies on the orthogonality between the range space and the left null space. However, the challenge is to find a basis for the left null space of $\mathbf{A}$. Consider $\mathbf{C}$ to be a basis for the left null space of $\mathbf{A}$; then, it follows that:

$$\mathbf{C}^T\mathbf{A} = 0. \tag{25}$$

In the ideal case ($\mathbf{X} = \mathbf{AS}$), the range space and left null space of $\mathbf{A}$ are the same as those of $\mathbf{X}$. This is because $\mathbf{A}$ is a full column rank (first criterion of NCA) and $\mathbf{S}$ is full row rank (third criterion of NCA). Therefore, $\mathbf{C}$ is obtained directly from $\mathbf{X}$. In contrast to the noiseless case, there is no direct access to $\mathbf{C}$ in the noisy case. Alternatively, SSP provides a robust approximation of $\mathbf{C}$. Consider the SVD $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$, and let $\mathbf{U}$ be partitioned as $\mathbf{U} = [\mathbf{U}_L, \mathbf{U}_R]$, where $\mathbf{U}_L$ is of dimensions $N \times M$ and $\mathbf{U}_R$ is of dimensions $N \times (N - M)$. Then, based on the discussion in Section 4.2.1, $\mathbf{U}_R$ represents an approximation of $\mathbf{C}$ ($\widehat{\mathbf{C}} = \mathbf{U}_R$). Therefore, $\mathbf{A}$ can be estimated by minimizing the Frobenius norm of $||\widehat{\mathbf{C}}^T\mathbf{A}||_F$, while maintaining both constraints, *i.e.*, the structure of the connectivity matrix and the positivity of all nonzero elements in the connectivity matrix. Mathematically, this problem can be formulated as follows:

$$\begin{aligned} \hat{\mathbf{A}} &= \arg\min_{\mathbf{A}} ||\widehat{\mathbf{C}}^T\mathbf{A}||_F \\ \text{s.t.} \quad & \mathbf{A}(I) = 0, \quad \mathbf{A}(J) \geq c \end{aligned} \tag{26}$$

where J stands for the set of indices of the nonzero elements in $\mathbf{A}$ and c is small positive constant. The optimization problem in Equation (26) is a convex optimization problem, since both the objective function and constraints are convex. The authors of [33] used an interior point-based method [24] to solve Equation (26). After evaluating $\mathbf{A}$, the signal matrix is estimated using the traditional ALS:

$$\mathbf{S} = \mathbf{A}^{\dagger}\mathbf{X} \tag{27}$$

The authors of [34] pioneered nnNCA, which utilizes the separable nature of the estimation problem corresponding to the matrix $\mathbf{A}$ in Equation (26) to achieve a computationally-efficient version of their previously-reported algorithm PosNCA. In PosNCA, matrix $\mathbf{A}$ is estimated in one shot by solving the optimization problem Equation (26). On the other hand, nnNCA estimates the columns in $\mathbf{A}$ in parallel, since each column of the connectivity matrix can be estimated independently of the other columns [34].

Later, NINCA [32] was proposed to further improve the computational efficiency and the estimation accuracy of the framework reported in PosNCA. Analogous to nnNCA, NINCA estimates matrix $\mathbf{A}$ on

a column-by-column basis. In addition, NINCA does not assume a positive constraint on the non-zero elements of the connectivity matrix and further imposes the constraint $\mathbf{1}^T \cdot \mathbf{a}_i = 1$ for each column of matrix $\mathbf{A}$ to avoid the trivial solution. In terms of the procedure to estimate the TF matrix $\mathbf{S}$, instead of using the traditional least-squares error adopted in [33] and [34], NINCA employs a total least-squares (TLS) algorithm [36] that not only considers the error in $\mathbf{S}$, but also weighs the error in $\mathbf{A}$.

5. Simulation Results

In this section, computer simulations are carried out to compare and evaluate the performance of major NCA algorithms. As discussed in Section 3, mNCA, gNCA, NCAr and gfNCA utilize the same estimation method as NCA. Additionally, PosNCA and nnNCA rely on the same framework, *i.e.*, minimizing the Frobenius norm of $\widehat{\mathbf{C}}^T\mathbf{A}$, where $\widehat{\mathbf{C}}$ denotes the estimated left null space of the connectivity matrix $\mathbf{A}$ and estimating $\mathbf{S}$ via the least-squares method. Therefore, only the simulation results pertaining to NCA, FastNCA, ROBNCA, NINCA and PosNCA will be presented in this section.

The synthetic data widely used in [17,29,31,32] are tested. This spectroscopy data contain $M = 3$ hemoglobin solutions obtained by mixing up $N = 7$ pure hemoglobin components, and the absorption spectra consist of K = 300 experiment points, which are measured for wavelengths in the range of 380–700 nm [29]. The aforementioned algorithms are tested when the observed data are corrupted with different levels of Gaussian noise and when the observations contain both Gaussian noise and outliers. The normalized mean square error (NMSE) and the data fitting error (DFE), *i.e.*, $||\mathbf{AS}-\mathbf{X}||_F$, are adopted herein to measure the estimation accuracy. The simulation results are averaged over 50 iterations. The algorithms are first simulated by varying the SNR from -10 dB–20 dB. The NMSE for matrices $\mathbf{A}$ and $\mathbf{S}$ and the DFE are illustrated in Figures 4 and 5, respectively. In terms of the test against both noise and outliers, the outliers are manually added into the observations by modeling them as a Bernoulli process with probability 0.1. The simulation results with respect to NMSE and DFE are depicted in Figures 6 and 7, respectively. Under 10 dB SNR, a comparison of the performance of NCA-based algorithms in both the noise case and noise + outliers case is shown in Table 1. These experiments are performed in MATLAB 7.12.0 with a 2.5-GHz Intel Core i5 processor, and the computation time stands for the average time to perform one iteration of simulation experiments.

Table 1. Normalized mean square error (NMSE), data fitting error (DFE) and computation time for different algorithms under 10 dB SNR. NINCA, non-iterative NCA; ROBNCA, robust NCA; PosNCA, positive NCA.

Algorithm	ANSME		SNSME		Data Fitting Error		Computation Time
	Noise	Noise + Outliers	Noise	Noise + Outliers	Noise	Noise + Outliers	
FastNCA	0.0571	0.0500	0.2544	0.2666	1.6973	4.4193	0.0005
NINCA	0.0037	0.0134	0.2250	0.2280	1.7361	4.7164	0.0119
ROBNCA	0.0033	0.0044	0.2218	0.2062	1.7141	4.5630	0.0080
NCA	0.0033	0.0060	0.2217	0.2068	1.7139	4.4809	6.6728
PosNCA	0.0031	0.0055	0.3896	0.3451	1.8200	4.7275	0.2648

Furthermore, we also employ these algorithms to quantitatively analyze a real dataset. In particular, a plant TRN in floral development using the *Arabidopsis thaliana* dataset housed in the Arabidopsis Gene Regulatory Information Server (AGRIS) [37] is analyzed. The initial dataset consists of 10 TFs and 57 genes. However, only seven TFs, namely LFY, AG, SEP3, AP2, AGL15, HY and AP3/PI, and 55 genes were found to be compliant with the NCA framework [38]. The simulation results to reconstruct the aforementioned seven TFs are depicted in Figure 8. It can be seen that NCA, ROBNCA and NINCA almost obtain an identical estimate for SEP3, and they share a similar trend for the reconstruction of other TFs. On the other hand, with respect to the estimation of SEP3, the results obtained by FastNCA and PosNCA are different from those exhibited by NCA, ROBNCA and NINCA.

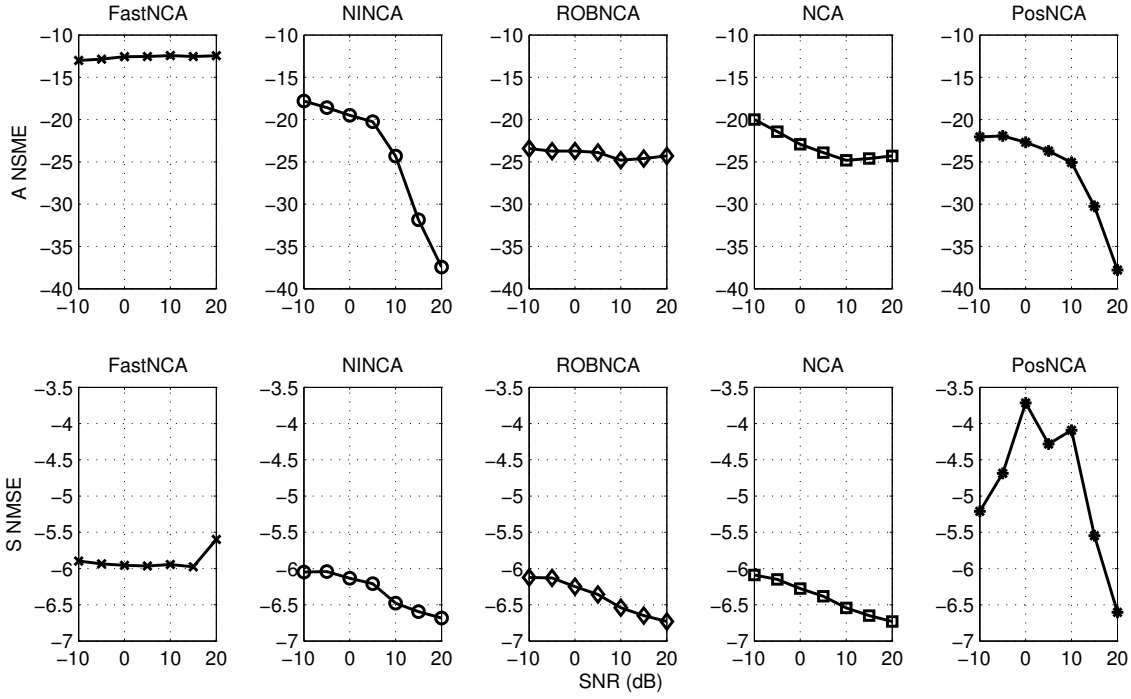


Figure 4. NMSE for different algorithms with respect to SNR from -10 dB–20 dB.

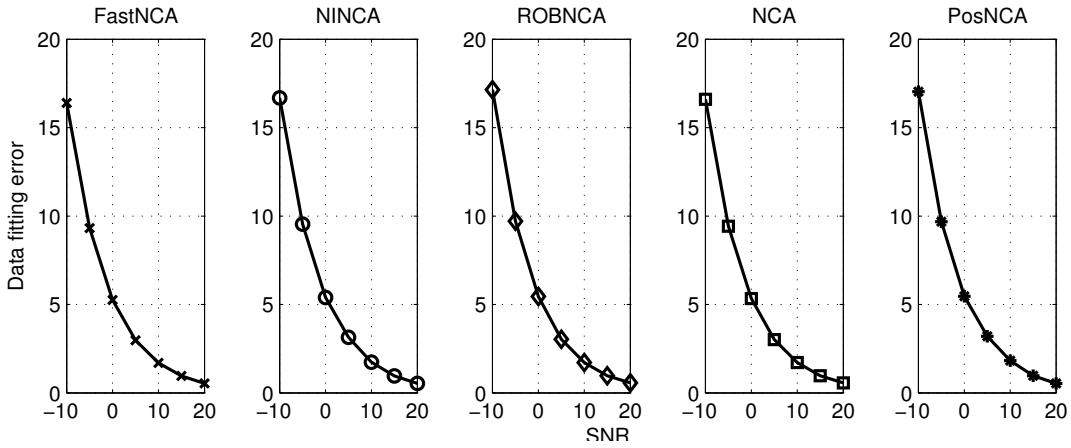


Figure 5. Data fitting error for different algorithms with respect to SNR from -10 dB–20 dB.

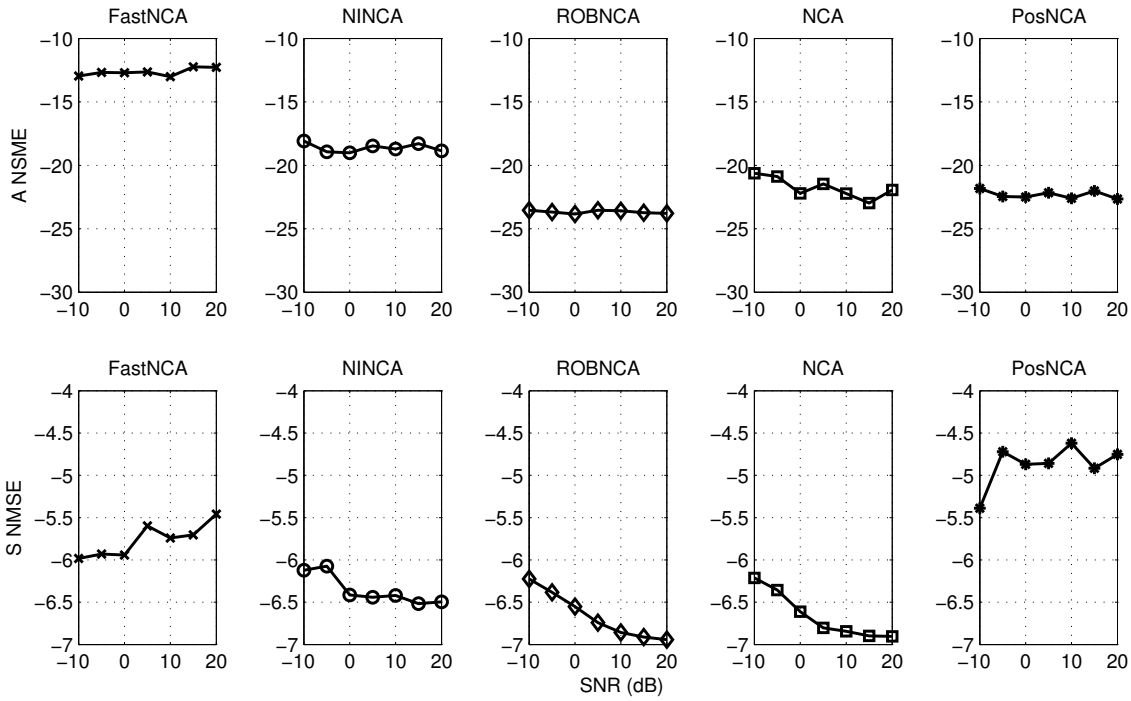


Figure 6. NMSE for different algorithms with respect to SNR from -10 dB–20 dB and outliers with probability 0.1.

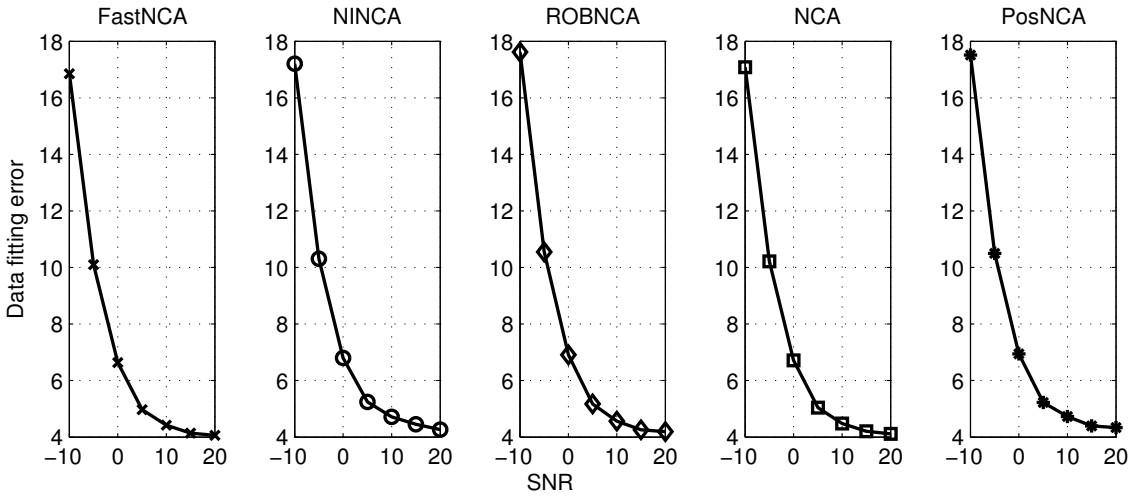


Figure 7. Data fitting error for different algorithms with respect to SNR from -10 dB–20 dB and outliers with probability 0.1.

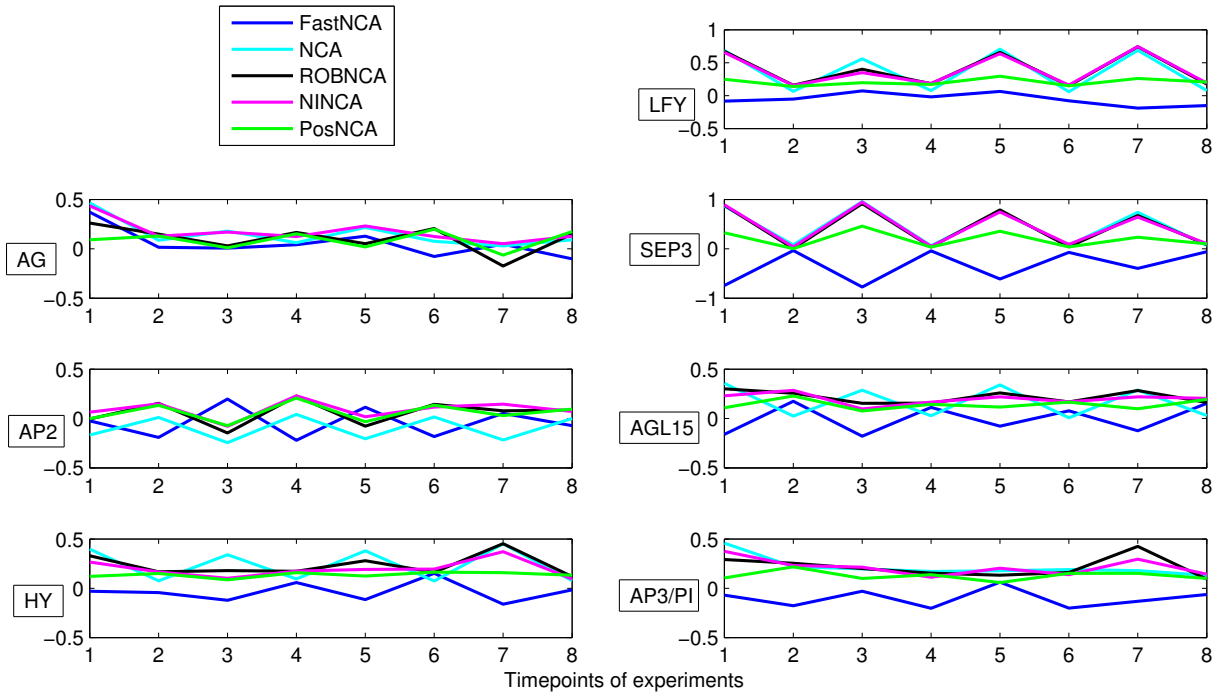


Figure 8. TFA reconstruction: estimation of seven TFAs of the Arabidopsis Gene Regulatory Information Server (AGRIS).

6. Comparison of NCA-Based Algorithms

Based on the simulation results in Figures 5 and 7, it can be concluded that all algorithms achieve a similar performance in terms of the data fitting error; or roughly speaking, these algorithms obtain a similar estimate of the product of $\mathbf{A}$ and $\mathbf{S}$. However, it can be observed from Figures 4 and 6 that different estimation results are obtained when NCA-based algorithms try to separate the product of $\mathbf{A}$ and $\mathbf{S}$, respectively. Therefore, in the following subsections, we mainly investigate the simulation performance on estimating the connectivity matrix $\mathbf{A}$ and the TF matrix $\mathbf{S}$ using the synthetic data. Additionally, some recommendations on how to choose the appropriate algorithm in different scenarios are also discussed.

6.1. Estimating the Connectivity Matrix

In terms of performance in the presence of additive noise, Figure 4 depicts that NINCA and PosNCA achieve a higher degree of accuracy when the SNR is high. Moreover, the performance of ROBNCA and NCA is also accurate and consistent compared to FastNCA. When the data are corrupted with both noise and outliers, according to Figure 6, ROBNCA achieves the best performance against outliers. The NSME of NINCA and PosNCA increases significantly compared to the case without outliers, especially when the SNR is high. Even though the performance of FastNCA does not degenerate when outliers exist, the NMSE is still relatively large compared to other algorithms.

6.2. Estimating the TF Matrix

Based on the simulation results in Figures 4 and 6, it can be observed that ROBNCA and NCA achieve the minimum NSME in both the noise case and noise + outliers case. Moreover, the existence of outliers does not have an obvious impact on the performance of ROBNCA and NCA. In contrast, the performance of FastNCA and NINCA for estimating the TF matrix $\mathbf{S}$ is not robust to outliers. Unlike the good simulation results in estimating $\mathbf{A}$, the performance of PosNCA for estimating $\mathbf{S}$ is significantly inferior to all other algorithms. That is probably because PosNCA utilizes the least-squares solution to derive $\mathbf{S}$ once obtaining the estimate of $\mathbf{A}$, which is numerically unstable.

6.3. Recommendations on Choosing the Appropriate Algorithm

In terms of the average computational time from Table 1, FastNCA is faster than the other four algorithms. However, FastNCA is not recommended herein, since it shows a high degree of inconsistency and inaccuracy. Moreover, even though NCA performs very well in both the noise and noise + outliers cases, the run time of NCA is hundreds and thousands of times slower than the other four algorithms using the small-dimensional synthetic data. It can be inferred that NCA is more computationally inefficient for reconstructing large-dimensional TRNs. In the case where the accuracy of the connectivity matrix is the first priority, PosNCA is recommended due to the fact that PosNCA has a high degree of accuracy in estimating $\mathbf{A}$, especially in the scenario where the SNR is high. NINCA and ROBNCA can be selected as the general methods to solve the TRN inference problem, since they are consistent and accurate in both the noise and noise + outliers cases. Moreover, the run time of NINCA and ROBNCA is also comparable to FastNCA. Between these two algorithms, ROBNCA is more preferable if the existence of outliers is known *a priori*.

7. Conclusions

This paper surveys the state-of-the-art NCA-based algorithms proposed in the literature. These algorithms rely on a linear model and concentrate on reconstructing the TFA matrix and the connectivity matrix by using the information provided by microarray gene expression data. The algorithms reviewed herein can be divided broadly into two categories: iterative and non-iterative methods. For the iterative methods, the estimation process for the connectivity matrix and TFA matrix starts with an initial guess, and then, it proceeds through a sequence of iterative steps. The output of each step is fed as an input to the next step. On the other hand, the non-iterative methods aim to overcome the drawbacks of iterative methods, especially to reduce the high computational complexity by reformulating the NCA problem. A summary of the surveyed NCA-based algorithms is illustrated in Table 2 for further details.

Table 2. Summary of NCA-based algorithms. mNCA, motif-directed NCA; gNCA, generalized NCA; NCAr, revised NCA; gfNCA, generalized-framework NCA; nnNCA, non-negative NCA; ALS, alternate least-squares; SSP, subspace separation principle; TLS, total least-squares.

Algorithm	Category	Estimation Technique	Contribution
NCA [17]	Iterative	ALS	Proposed the NCA framework and criteria, motivated other NCA algorithms
mNCA [25]	Iterative	ALS	Incorporated motif information to obtain the prior connectivity information
gNCA [26]	Iterative	ALS	Incorporated the prior information about the TFA matrix
NCAr [27]	Iterative	ALS	Revised and extended the third identification criterion
gfNCA [28]	Iterative	ALS	Modified the criteria of NCA, such that they are only related to the prior connectivity information
ROBNCA [29]	Non-iterative	Alternate optimization	Reduced the computational complexity and improved the robustness against outliers
FastNCA [31]	Non-iterative	SSP, rank-1 factorization	Reduced the computational complexity
PosNCA [33]	Non-iterative	SSP, convex optimization	Combined additional prior information to reduce the complexity
nnNCA [34]	Non-iterative	SSP, convex optimization	Combined additional prior information and reduced the complexity
NINCA [32]	Non-iterative	SSP, convex optimization, TLS	Combined additional prior information, reduced the complexity and improved the estimation accuracy

Acknowledgments

This work was supported by NSF Award No. 1318338.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Latchman, D.S. Transcription factors: An overview. *Int. J. Biochem. Cell Biol.* **1997**, *29*, 1305–1312.
2. Lähdesmäki, H.; Rust, A.G.; Shmulevich, I. Probabilistic inference of transcription factor binding from multiple data sources. *PLoS ONE* **2008**, *3*, e1820.
3. Chen, H.C.; Lee, H.C.; Lin, T.Y.; Li, W.H.; Chen, B.S. Quantitative characterization of the transcriptional regulatory network in the yeast cell cycle. *Bioinformatics* **2004**, *20*, 1914–1927.
4. Sasik, R.; Iranfar, N.; Hwa, T.; Loomis, W. Extracting transcriptional events from temporal gene expression patterns during Dictyostelium development. *Bioinformatics* **2002**, *18*, 61–66.
5. Shmulevich, I.; Dougherty, E.R.; Kim, S.; Zhang, W. Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics* **2002**, *18*, 261–274.

6. Akutsu, T.; Miyano, S.; Kuhara, S. Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics* **2000**, *16*, 727–734.
7. Zare, H.; Sangurdekar, D.; Srivastava, P.; Kaveh, M.; Khodursky, A. Reconstruction of Escherichia coli transcriptional regulatory networks via regulon-based associations. *BMC Syst. Biol.* **2009**, *3*, 39, doi:10.1186/1752-0509-3-39.
8. Butte, A.J.; Tamayo, P.; Slonim, D.; Golub, T.R.; Kohane, I.S. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proc. Natl. Acad. Sci.* **2000**, *97*, 12182–12186.
9. Butte, A.J.; Kohane, I.S. Mutual information relevance networks: Functional genomic clustering using pairwise entropy measurements. *Pac. Symp. Biocomput.* **2000**, *5*, 418–429.
10. Liang, S.; Fuhrman, S.; Somogyi, R. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pac. Symp. Biocomput.* **1998**, *3*, 18–29.
11. Markowetz, F.; Spang, R. Inferring cellular networks—A review. *BMC Bioinform.* **2007**, *8*, doi:10.1186/1471-2105-8-S6-S5.
12. Friedman, N. Inferring cellular networks using probabilistic graphical models. *Science* **2004**, *303*, 799–805.
13. Emmert-Streib, F.; Glazko, G.; de Matos Simoes, R. Statistical inference and reverse engineering of gene regulatory networks from observational expression data. *Front. Genet.* **2012**, *3*, doi:10.3389/fgene.2012.00008.
14. Friedman, N.; Linial, M.; Nachman, I.; Pe'er, D. Using Bayesian networks to analyze expression data. *J. Comput. Biol.* **2000**, *7*, 601–620.
15. Xiong, M.; Li, J.; Fang, X. Identification of genetic networks. *Genetics* **2004**, *166*, 1037–1052.
16. Liu, B.; de La Fuente, A.; Hoeschele, I. Gene network inference via structural equation modeling in genetical genomics experiments. *Genetics* **2008**, *178*, 1763–1776.
17. Liao, J.C.; Boscolo, R.; Yang, Y.L.; Tran, L.M.; Sabatti, C.; Roychowdhury, V.P. Network component analysis: Reconstruction of regulatory signals in biological systems. *Proc. Natl. Acad. Sci.* **2003**, *100*, 15522–15527.
18. Lee, T.I.; Rinaldi, N.J.; Robert, F.; Odom, D.T.; Bar-Joseph, Z.; Gerber, G.K.; Hannett, N.M.; Harbison, C.T.; Thompson, C.M.; Simon, I.; *et al.* Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science* **2002**, *298*, 799–804.
19. Roth, F.P.; Hughes, J.D.; Estep, P.W.; Church, G.M. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat. Biotechnol.* **1998**, *16*, 939–945.
20. Bussemaker, H.J.; Li, H.; Siggia, E.D. Building a dictionary for genomes: Identification of presumptive regulatory sites by statistical analysis. *Proc. Natl. Acad. Sci.* **2000**, *97*, 10096–10100.
21. Bussemaker, H.J.; Li, H.; Siggia, E.D. Regulatory element detection using correlation with expression. *Nat. Genet.* **2001**, *27*, 167–174.
22. Jolliffe, I. *Principal Component Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2002.

23. Hyvärinen, A.; Karhunen, J.; Oja, E. *Independent Component Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2004; Voluem 46.
24. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
25. Wang, C.; Xuan, J.; Chen, L.; Zhao, P.; Wang, Y.; Clarke, R.; Hoffman, E. Motif-directed network component analysis for regulatory network inference. *BMC Bioinform.* **2008**, *9*, doi:10.1186/1471-2105-9-S1-S21.
26. Tran, L.M.; Brynildsen, M.P.; Kao, K.C.; Suen, J.K.; Liao, J.C. gNCA: A framework for determining transcription factor activity based on transcriptome: Identifiability and numerical implementation. *Metab. Eng.* **2005**, *7*, 128–141.
27. Galbraith, S.J.; Tran, L.M.; Liao, J.C. Transcriptome network component analysis with limited microarray data. *Bioinformatics* **2006**, *22*, 1886–1894.
28. Boscolo, R.; Sabatti, C.; Liao, J.C.; Roychowdhury, V.P. A generalized framework for network component analysis. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2005**, *2*, 289–301.
29. Noor, A.; Ahmad, A.; Serpedin, E.; Nounou, M.; Nounou, H. ROBNCA: Robust network component analysis for recovering transcription factor activities. *Bioinformatics* **2013**, *29*, 2410–2418.
30. Finegold, M.; Drton, M. Robust graphical modeling of gene networks using classical and alternative t-distributions. *Ann. Appl. Stat.* **2011**, *5*, 1057–1080.
31. Chang, C.; Ding, Z.; Hung, Y.S.; Fung, P.C.W. Fast network component analysis (FastNCA) for gene regulatory network reconstruction from microarray data. *Bioinformatics* **2008**, *24*, 1349–1358.
32. Jacklin, N.; Ding, Z.; Chen, W.; Chang, C. Noniterative convex optimization methods for network component analysis. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2012**, *9*, 1472–1481.
33. Chang, C.; Hung, Y.S.; Ding, Z. A new optimization algorithm for network component analysis based on convex programming. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009; pp. 509–512.
34. Dai, J.; Chang, C.; Ye, Z.; Hung, Y.S. An efficient convex nonnegative network component analysis for gene regulatory network reconstruction. In *Pattern Recognition in Bioinformatics*; Springer: Berlin, Germany, 2009; pp. 56–66.
35. Alon, U. *An Introduction to Systems Biology: Eesign Principles of Biological Circuits*; CRC Press: Boca Raton, FL, USA, 2007.
36. Golub, G.H.; van Loan, C.F. An analysis of the total least squares problem. *SIAM J. Numer. Anal.* **1980**, *17*, 883–893.
37. Palaniswamy, S.K.; James, S.; Sun, H.; Lamb, R.S.; Davuluri, R.V.; Grotewold, E. AGRIS and AtRegNet. A platform to link *cis*-regulatory elements and transcription factors into regulatory networks. *Plant Physiol.* **2006**, *140*, 818–829.

38. Misra, A.; Sriram, G. Network component analysis provides quantitative insights on an Arabidopsis transcription factor-gene regulatory network. *BMC Systems Biology* **2013**, *7*, doi:10.1186/1752-0509-7-126.