

## MANIA: A GENE NETWORK REVERSE ALGORITHM FOR COMPOUNDS MODE-OF-ACTION AND GENES INTERACTIONS INFERENCE\*

DARONG LAI<sup>†,‡,§</sup> and HONGTAO LU<sup>†</sup>

<sup>†</sup>*Department of Computer Science and Engineering,  
Shanghai Jiao Tong University, China*

<sup>‡</sup>*CAS-MPG Partner Institute for Computational Biology,  
Yue Yuan Road 320, Shanghai, China*

<sup>§</sup>*darong.lai@gmail.com*

MARIO LAURIA and DIGEO DI BERNARDO

*TIGEM, 111 Via Pietro Castellino, Naples, Italy*

CHRISTINE NARDINI

*CAS-MPG Partner Institute for Computational Biology,  
Yue Yuan Road 320, Shanghai, China*

*christine@picb.ac.cn*

Received 8 May 2009

Revised 7 January 2010

Understanding the complexity of the cellular machinery represents a grand challenge in molecular biology. To contribute to the deconvolution of this complexity, a novel inference algorithm based on linear ordinary differential equations is proposed, based solely on high-throughput gene expression data. The algorithm can infer (i) gene–gene interactions from steady state expression profiles and (ii) mode-of-action of the components that can trigger changes in the system. Results demonstrate that the proposed algorithm can identify *both* information with high performances, thus overcoming the limitation of current algorithms that can infer reliably only one.

*Keywords:* Gene network; gene expression; reverse engineering; ordinary differential equations (ODE); compound mode-of-action.

### 1. Introduction

Thanks to the fast moving and recent advancements in technology, our society is assisting to an unprecedented high-throughput production of information coming from a variety of areas of human activity. This comprises, but is not limited to, economic,

\*From Complex 2009 — the First International Conference on Complex Sciences: Theory and Applications.

social, and biological data. In particular, we focus our attention on biomolecular data. To deconvolute the structure underlying such data, cross fertilization from diverse areas of research, and notably the introduction of exact sciences in the realm of biology, has been a fundamental requirement to mine the complex interaction that explains the data we observe. However, the task is far from completed, and although economical, sociological, and molecular systems own peculiar characteristics, advances in the deconvolution of the complexity in any of these areas bears the potential to significantly contribute to explain the complexity of the global system we live in. In the area of molecular biology, several high-throughput platforms are quickly becoming available [2]; however, gene expression data represents at the moment the most abundant source of molecular high-throughput information. This work focuses on the identification of networks of interaction among genes. Networks of interactions identify general relationships among the nodes of the network (genes), thus, a link in the network may not represent a physical interaction (carried on by intermediate molecules such as proteins). However, these algorithms can be extremely powerful in the initial characterization of unknown systems, taking advantage of low-cost, high-throughput screens and generating relevant *in silico* hypotheses that can be further and efficiently tested in *wet lab*. Moreover, these algorithms, thanks to their ability to reconstruct networks on genome-wide data, offer a systemic perspective of the interactions. Depending on the model adopted, these methods can infer a causality in the relationship (directed networks) or rather a simple “connection” among items (undirected networks). Many methods [7] have been proposed to reverse-engineer gene expression data, that can either take advantage of the evolution in time of the state of the system (time series, e.g. see Ref. 8), or of different equilibrium states reached by the system (steady state, e.g. see Ref. 9). Our approach focuses on the latter, more abundant, steady state data. To achieve different equilibrium states of the system, the system is perturbed in different ways (e.g. knock-out, knock-down, alterations in the growing medium) and the resulting expression data is collected once the system has reached the novel equilibrium. Algorithms that handle these data typically output a representation of the gene network in the form of a graph or an adjacency matrix (here called  $A$  [9, 1, 6]). These networks represent the relationships occurring among genes, and offer a first impression of the complex pathways that are being activated in the system under study. Alternatively these algorithms offer an estimation, for example in the form of a ranked list, of the genes that were directly affected by the perturbation in the experiments [3] (here called  $P$ ). When the perturbation is obtained adding a compound in the environment of the cell, the genes identified by  $P$  represent the direct targets of the perturbing agents that have been used to alter the equilibrium. This identifies an information extremely valuable in areas such as *chemogenomics*, where the identification of a small molecule’s direct target (also called *transcriptional perturbations*) can provide fundamental information on its use as a drug. Because of this,  $P$  is also known to represent the *mode-of-action* of the perturbing compound. So far, *a priori* knowledge of the direct targets of perturbation was

required for a proper identification of  $A$  [9], or alternatively, the identification of an estimate of  $P$  was not able to produce a reliable representation of  $A$  [3], due to the high sensitivity of the algorithms to errors in  $P$ . With our novel approach we aim at the identification of both the gene network ( $A$ ) and the single direct target matrix ( $P$ ), overcoming the current limitation, while preserving and improving both performances. Our approach can handle efficiently experiments resulting in single transcriptional perturbations. Single transcriptional perturbations are useful to be quantified when the entity of a single gene knock-down is unknown and when the action of perturbagens is supposed to target predominantly an individual (unknown) transcript or protein, rather than several elements of a pathway. In the following we present related methods (Sec. 2), details of our algorithm (Sec. 3), validation results (Sec. 4), and their interpretation (Sec. 5).

## 2. Related Work

Number of approaches are being designed and tested to uncover the complexity of molecular interactions. In the following we briefly describe currently used tools based on Bayesian theory (Banjo [6]), information theory approach (ARACNe [1]), and ordinary differential equations (ODE, NIR, and MNI [9, 3]), that have proven to be useful in the identification of gene networks or compounds mode-of-action. All the above methods can handle steady state data. Banjo [6] generates a network space and screens then the best network structures attributing the most appropriate conditional density function, by optimization of an objective function (Bayesian Dirichlet equivalence or Bayesian information criterion). Banjo can reconstruct signed directed network indicating regulation among genes, but it cannot infer networks involving cycles (or loops). ARACNe (Algorithm for Reconstruction of Accurate Cellular Networks [1]) is regarded as an information-theoretic approach to gene network inference. It computes mutual information (MI [10]) for all pairs of genes profiles to estimate the independence between genes and uses strategies (Data Processing Inequality, DPI) to successfully filter out the number of false-positive interactions. ARACNe cannot reconstruct directed networks. Our approach is strongly rooted in two ODE-based methods previously developed and validated. Namely, we used as a starting point NIR [9] able to infer the network of genes interactions ( $A$ ), provided the perturbations ( $P$ ) are known, and MNI [3], able to rank the most likely direct target genes of perturbations (estimate of  $P$ ). Briefly, these algorithms aim at the identification of the function that describes the variation of gene expression matrix  $\underline{x}$  over time  $\underline{x}' = f(\underline{x}, \underline{p})$ , with  $\underline{x}$  representing the steady state expressions of the  $N$  genes involved in the network across  $M$  experiments,  $f$  is a nonlinear function that models how the expression values  $\underline{x}$  and  $M$  experiments provoking external influences  $\underline{p}$  modify the genes' activity. Assuming steady state and small perturbations, these equations can be linearized around the equilibrium state, and become, for a scalar element of the expression matrix  $x'_{il} = \sum_j a_{ij}x_{jl} + p_{il} = \underline{a}_i^T \cdot \underline{x}_l + \underline{p}_{il}$  with  $i, j = 1, \dots, N$  indicating genes

and  $l = 1, \dots, M$  experiments. In matricial form and at equilibrium this becomes  $AX = -P$ , with  $A$  being the  $N$  by  $N$  network matrix ( $a_{ij}$  represents the action of gene  $j$  on gene  $i$ ),  $X$  the expression data ( $x_{il}$  represents the expression of gene  $i$  in experiment  $l$ ) and  $P$  the matrix of transcriptional perturbations ( $p_{il}$  represents the transcriptional perturbation of gene  $i$  in experiment  $l$ ), explaining the origin of our notations. These approaches assume that only a limited number of connections among genes are possible, to reflect the structure of the molecular pathways. Based on this sparsity assumption, NIR uses multiple linear regression to infer the connections among genes. Conversely, MNI is trained on the expression data in  $X$  to evaluate  $A$  and  $P$  through an iterative process based on the minimization of an objective function (Sum of Square Errors, SSE).

### 3. Method

Our approach aims at identifying  $A$  and  $P$  based solely from the expression data  $X$ , thus overcoming the necessity to have *a priori* information on the direct target of the perturbation ( $P$ ), which is very often an important unknown of the problem. To achieve this goal, we sought to chain the two algorithms in order to use the prediction of MNI to feed NIR and infer the network. To do so, our algorithm uses iteratively  $M - 1$  experiments in  $X$  to predict the  $M$ th column (experiment) of  $P$ , as a ranked list of most likely targets. Due to the intrinsic noise of the data and the limited deterministic predictive power of MNI, the reliable identification of  $A, P$  is not trivial, especially when predicting complex data, as it can be shown in Sec. 4, in the varying performances of MNI + NIR, which represents the trivial chaining of the two algorithms (output of MNI used directly as  $P$  for NIR, see Fig. 1(a)). For this reason, other strategies have to be integrated, schematically shown in Fig. 1(b). Based on previous acronyms (and on the obsessive search for the network identification) we call this new approach Mode of Action & Network Identification Approach (MANIA).

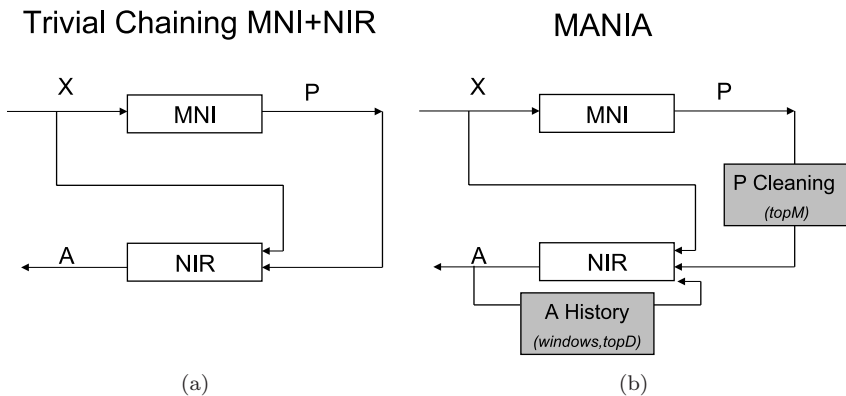


Fig. 1. Schematic view of the trivial chaining of MNI and NIR and of the strategies implemented in MANIA, discussed in Sec. 3.

In this approach, an estimate of  $P$  is produced by MNI, called  $P_{\text{MNI}}$ , this matrix contains the top ranking perturbations (we tested 1 and 10 top best, parameter  $topP$ ), while all other values of  $P_{\text{MNI}}$  are set to zero. When choosing the single top perturbation option ( $topP = 1$ ), the algorithm should perform at its best, provided MNI reliably identifies the correct transcriptional perturbation as the most likely (i.e. the best prediction *is* indeed the gene target). In this case, in fact, no noise is added; however, we also tested the algorithm preserving the top 10 best predictions ( $topP = 10$ ), to offer backup solutions in case MNI is not able to find the correct perturbation as first choice. The core step of the algorithm consists of the strategy used to *clean*  $P_{\text{MNI}}$  from the incorrect predictions, so that only the appropriate perturbation is used in NIR to predict  $A$ . This strategy consists of two steps. The first is the iterative computation of all the solutions for a given row of  $A$ , using all the predictions offered by  $P_{\text{MNI}}$ . NIR calculates each solution of that row of  $A$  by one possible combination of nonzero perturbations in the corresponding row of  $P_{\text{MNI}}$ , if the interactive genes are known. Since NIR assumes that only limited number of connections among genes are possible, i.e. each row of  $A$  has  $restK$  connections including self-connection, it uses a heuristic strategy (Forward- $topD$ - $restK$  [11]) to select the best combination of  $restK$  genes, which allows to start from one gene (used for self-connection) and iteratively ( $restK-1$  times) preserve the best ( $topD$ ) solutions that minimize the objective function (SSE, determined by the linear regression on currently selected genes) until  $restK$  genes are selected. The solutions for that row of  $A$  are then ranked and only the  $topM$  best are preserved (along with the corresponding perturbations) while computing the following rows. However, this step alone is not sufficient, since, often, the solution that minimizes the objective function (SSE) produces a local minimum of the objective function. Choosing this solution can result in the identification of a unique minimum for  $P$  and thus for all the rows of  $A$ . To overcome this issue, information about the previous rows computed in  $A$  are used. Thus, another parameter ( $windows$ ) has been introduced to indicate the number of rows used as previous knowledge to calculate and minimize SSE. In particular, SSE is computed on all the  $topM$  solutions as  $X = -A_{tmp}^{-1}P$ , where  $A_{tmp}$  is the identity matrix (self-relation is always assumed true) with the corresponding  $windows$  rows replaced by the solutions already computed. By construction  $A$  is always invertible. For each row of  $A$  and  $P$  only the best  $topM$  solutions are preserved before computing the following rows of  $A, P$ . In our simulations, we set  $restK = 10$ ,  $windows = topD = 5$ , and  $topM = 200$ . In our experience, these values represent a good compromise between computation time and accuracy. Pseudocode in Algorithm 1 gives more details about the process. The parameters  $restK$  and  $topD$  are absorbed in the calculation of each row of  $A$ , i.e.  $A_{ijk}$ , and not explicitly shown in the Pseudocode.

**Algorithm 1.** Pseudocode for MANIA. Matricial notations follow Matlab syntax:  $M(:, i)$  indicates column  $i$  in matrix  $M$ ,  $M(i, :)$  indicates row  $i$  in matrix  $M$ .

**Input:** gene expression profiles matrix  $X$ , user defined parameters.

**Output:** Adjacency matrix  $A$  of gene network; mode-of-action matrix  $P$ .

$N$ : number of genes;

$M$ : number of experiments;

$topP$ : max number of perturbations proposed by MNI preserved in  $P_{MNI}$  per experiment;

$topM$ : max number of solutions preserved per each row of  $A$ ;

$Windows$ : max number of previously computed row preserved;

**For**  $i \leftarrow 1$  to  $M$  **do**

Compute  $P_{MNI}(:, i)$  with MNI from  $X(:, [1, \dots, i - 1, i + 1, \dots, M])$ ;

**end**

Sort  $P_{MNI}$  columnwise in descending order  $\rightarrow pIdx$  sorted index matrix of  $P_{MNI}$ ;

$arrayA_{1:topM} \leftarrow \text{NULL}$ ;

$arrayP_{1:topM} \leftarrow \text{NULL}$ ;

**For**  $i \leftarrow 1$  to  $N$  **do**

**For**  $j \leftarrow 1$  to  $topP$  **do**

Create perturbation matrix  $Pmat$  in two steps:

(a).  $Pmat \leftarrow O(\text{zero-matrix})$ ;

(b).  $Pmat(pIdx(j, m), m) \leftarrow P_{MNI}(pIdx(j, m), m)$ , where  $1 \leq m \leq M$ ;

Get  $j$ th perturbation vector  $P_i(j, :) = Pmat(j, :)$ ;

**For**  $k \leftarrow 1$  to Total number of all combinations of nonzero perturbations in  $P_i(j, :)$ , say  $P_{ijk}$  **do**

compute  $i$ th row of  $A$ , i.e.  $A_{ijk}$ , with NIR and perturbation  $P_{ijk}$ ;

compute  $A_{tmp,ijk}$  as identity matrix with  $windows$  rows of  $arrayA_h (1 \leq h \leq topM)$  and  $A_{ijk}$  as  $i$ th row;

compute  $P_{tmp,ijk}$  as zeros matrix with previous  $windows$  rows of  $arrayP_h (1 \leq h \leq topM)$  and  $P_{ijk}$  as  $i$ th row;

compute  $SSE_{ijk} = (X - A_{tmp,ijk}^{-1} \cdot P_{tmp,ijk})^2$ ;

**end**

**end**

Rank SSE and select  $topM$   $A_{ijk}$  solutions;

**For**  $h \leftarrow topM$  **do**

$arrayA_h = [arrayA_h(1 : i - 1, :); A_{tmp,h}(i, :)]$ ;

$arrayP_h = [arrayP_h(1 : i - 1, :); P_{tmp,h}(i, :)]$ ;

**end**

**end**

$A = arrayA_1$ ;

$P = arrayP_1$ .

#### 4. Experimental Results

To validate our approach, we used two known benchmark datasets (here called Dataset 1 [7] and Dataset 2 [4]), and compared our performances to state-of-the-art algorithms briefly summarized in Table 1. These algorithms were used with their default parameters values.

**Dataset 1.** This dataset consists of 20 instances of expression matrices  $X$  with 100 genes and 100 experiments, obtained from 20 instances of network matrices  $A$  with sparsity 10 (indicating a maximum of 10 possible interactions for each gene), and single perturbation for  $P$  (identity matrix). Gaussian noise (10%) is added to expression data to better mimic real data. Performances are computed using positive predictive value (PPv, also called *accuracy*) defined as  $TP/(TP + FP)$  and Sensitivity  $TP/(TP + FN)$ , where TP, FP, and FN stand for True Positive, False Positive, and False Negative, respectively. Results were averaged, and proved to be stable with  $st.dev < 0.08$  in all cases (standard deviations of PPv/sensitivity for undirected networks are 0.07/0.07, and 0.06/0.06 for directed networks) (Table 2).

Table 1. Network inference algorithms used for performances comparison.

Software	Download link	Model
BANJO	<a href="http://www.cs.duke.edu/~amink/software/banjo">www.cs.duke.edu/~amink/software/banjo</a>	Bayesian method
ARACNe	<a href="http://www.amdec-bioinfo.cu-genome.org/html">www.amdec-bioinfo.cu-genome.org/html</a>	Information-theory method
MNI/NIR	<a href="http://dibernardo.tigem.it/wiki/index.php">http://dibernardo.tigem.it/wiki/index.php</a>	ODE-based method

Table 2. Matrix  $A$  performance results of Dataset 1.

Algorithm	Directed		Undirected	
	PPv	Sensitivity	PPv	Sensitivity
<b>MNI+NIR1</b>	<b>0.84</b>	<b>0.75</b>	<b>0.86</b>	<b>0.76</b>
MNI+NIR10	0.18	0.15	0.27	0.23
<b>MANIA1</b>	<b>0.89</b>	<b>0.81</b>	<b>0.95</b>	<b>0.81</b>
<b>MANIA10</b>	<b>0.75</b>	<b>0.68</b>	<b>0.79</b>	<b>0.70</b>
<b>NIR</b>	<b>0.96</b>	<b>0.86</b>	<b>0.97</b>	<b>0.87</b>
ARACNe	—	—	0.56	0.28
BANJO	0.42	—	0.71	0.00
Random	0.10	—	0.19	—

*Note:* MNI+NIR represents the trivial chaining of MNI and NIR, with no strategy to identify the best performances and keeping the single first best and 10 first best predictions of MNI (called, respectively, MNI+NIR1 and MNI+NIR10). The same values were used for MANIA. *Random* refers to the expected performances of an algorithm that selects pairs of genes randomly and then infers an edge between them.

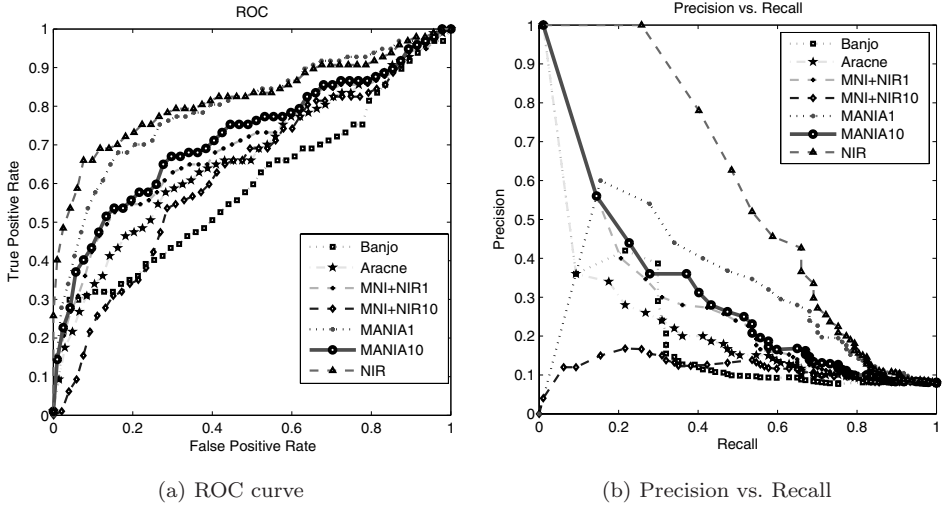
**Dataset 2.** This dataset comes from the Dream 2 Competition (Heterozygous InSilico 1, Challenge 4) organized by the DREAM (Dialogue for Reverse-Engineering Assessments and Methods [4]) consortium, whose objective is to catalyze the interaction among researchers and improve progresses in the area of cellular network inference. Data was generated using simulations of biological interactions. Namely, the rate of synthesis of the mRNA of each gene is considered to be affected by the level of mRNA of other genes. For these reasons, this represents a valuable and challenging benchmark to test reverse-engineering approaches. This dataset contains steady state levels of 50 genes of an hypothetical wild-type organism and 50 heterozygous knock-down strains. All ODE algorithms were tested assuming the number of connections associated with each gene (connectivity of the network and sparsity of the matrix) is 10, including self-connection. Data were preprocessed with log-transformation of the expression ratio for each gene (knock-down versus wild-type strains). Ratios corresponding to null levels of expression in wild-type were treated as unknown values, and were set to zero as it was done in Ref. 9. Standard deviation of each entry of the data matrix  $X$  was computed against the 25-nearest neighbors of the gene of interest, with the approach illustrated in Ref. 9. Finally before computing  $A$ , the absolute value of  $P_{\text{MNI}}$  was normalized column-wise for numeric stability consideration, however, this step does not affect the results. Besides evaluating PPv and Sensitivity (ROC curves), the adjacency matrix  $A$  was also scored following the procedure adopted in the DREAM 2 Challenge, after scoring the connections of  $A$  (normalizing the absolute values). Results were graded using the area under the curve (AUC) for ROC (false positive vs true positive rate) and precision-versus-recall curve (Prec versus Rec) for the whole set of predictions. For the first  $k$  predictions (ranked by score, and for predictions with the same score, taken in the order they were put in the prediction files), Precision was defined as the fraction of correct predictions to  $k$ , and Recall was the proportion of correct predictions out of all the possible true connections.

Figure 2 is the graphical version of the results of Table 3.

Table 4 and Fig. 3 give the results produced by the algorithms for the performances on the directed network. From the tables and the figures, one can find that MANIA always shows good performances. We are discussing more details in Sec. 5.

## 5. Discussion

We have tested our approach against four different algorithms and across two datasets interpreted as directed and undirected networks. In general, MANIA can perform better than the state-of-the-art non-ODE approaches listed in Table 1, which were used by setting parameters to their default values, and comparably well or superiorly to ODE approaches as NIR or MNI+NIR. Our objective was to infer  $A$  with performances as close as possible to NIR, which we considered as our gold standard [7]. Before discussing further the performances, it is worth noting that comparisons between MNI+NIR and MANIA were done with the purpose to

Fig. 2. Performances for  $A$  on Dataset 2. AUC curves for undirected network.Table 3. Performance results on Dataset 2 on  $A$  undirected network.

Algorithm	Precision at $n$ th correct prediction				AUC	
	1st	2nd	5th	20th	Prec vs Rec curve	ROC curve
MNI+NIR1	1.0000	1.0000	0.8333	0.4651	0.2859	0.6965
MNI+NIR10	0.0909	0.0609	0.1219	0.1639	0.1158	0.6230
<b>MANIA1</b>	<b>0.5000</b>	<b>0.4000</b>	<b>0.5556</b>	<b>0.5714</b>	<b>0.3513</b>	<b>0.7957</b>
<b>MANIA10</b>	<b>1.0000</b>	<b>1.0000</b>	<b>0.6250</b>	<b>0.5405</b>	<b>0.3014</b>	<b>0.7191</b>
<b>NIR</b>	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>0.5968</b>	<b>0.8202</b>
ARACNe	1.0000	1.0000	0.5000	0.3279	0.2143	0.6658
BANJO	1.0000	0.3333	0.3125	0.4167	0.1900	0.5925

Table 4. Performance results on Dataset 2 on  $A$  directed network.

Algorithm	Precision at $n$ th correct prediction				AUC	
	1st	2nd	5th	20th	Prec. versus Rec curve	ROC curve
MNI+NIR1	0.5000	0.6667	0.5556	0.3279	0.1921	0.6999
MNI+NIR10	0.1429	0.2222	0.1724	0.1835	0.1107	0.6864
<b>MANIA1</b>	<b>0.5000</b>	<b>0.4000</b>	<b>0.6250</b>	<b>0.3846</b>	<b>0.2258</b>	<b>0.7877</b>
<b>MANIA10</b>	<b>1.0000</b>	<b>0.6667</b>	<b>0.5556</b>	<b>0.3448</b>	<b>0.2066</b>	<b>0.7232</b>
<b>NIR</b>	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>0.5781</b>	<b>0.8314</b>
BANJO	0.5000	0.6667	0.2174	0.0559	0.0724	0.5441

assess the validity of the enhancements proposed, compared with our simpler idea of directly chaining the two approaches (MNI and NIR). We figured that, given the reasonably good performances of MNI on the identification of one perturbation, MNI+NIR would be advantaged when used with the parameter  $topP = 1$ , which offers to NIR the best possible  $P$ . In order to perform a fair comparison,

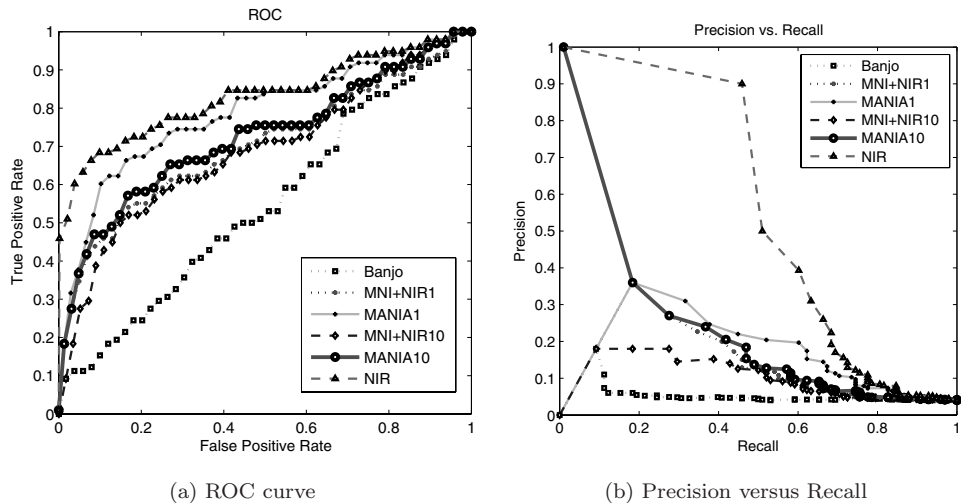


Fig. 3. Performances for  $A$  on Dataset 2. AUC curves for directed network.

MANIA was also tested with this value of the parameter; however, we expected MANIA to perform better when it can take advantage of more proposed solutions. Our final goal was to assess if, despite the expected variable performances of the two algorithms with different parameter setting, MANIA could be able to identify solutions with global better performances. This is indeed true for the final output of  $A$ , and performances remain superior or comparable for the identification of  $P$ , indicating that the modification introduced in the multiple regression step defined in MANIA contributes to improve the final results. Overall MANIA has proved to be comparable or to outperform the simplified approach MNI + NIR, even when using identical parameter  $topP$ , thus, it guarantees more stable performances, and offers results comparable to NIR, with no need for *a priori* information on  $P$ . With respect to the identification of  $P$ , MANIA shows stable and robust results comparable or outperforming MNI + NIR, see Table 5. These results are confirmed when coming to the identification of  $A$ . In particular, when tested against simple models

Table 5. Performance results on Dataset 1 and Dataset 2 for the ODE-based algorithms that can predict  $P$ .

	Dataset 1		Dataset 2	
	PPv	Sensitivity	PPv	Sensitivity
MNI+NIR1	0.870	0.870	0.540	0.540
MNI+NIR10	0.094	0.920	0.066	0.660
MANIA1	<b>0.950</b>	<b>0.860</b>	<b>0.750</b>	<b>0.540</b>
MANIA10	0.798	0.830	0.526	0.600

for simulations (Dataset 1) the performances of MANIA and MNI+NIR are inferior to NIR; however, they do not degrade much and both algorithms have the fundamental advantage to infer  $A$  from expression data only. In the validation on Dataset 2 a more complex and realistic model, these trends are confirmed, with even less variance in the performances, and highlighting the superiority of MANIA. This supports the introduction of the algorithmic variations peculiar to MANIA. Since NIR was the best algorithm tested on this dataset in the Challenge DREAM2, the possibility to preserve or degrade little the performances under more difficult conditions, represents an important achievement. In general, compared to the best performing algorithm NIR, MANIA has comparable performances in accuracy and the great advantage of not requiring *a priori* knowledge on the targets of the perturbations. Compared to NIR + MNI it has higher ability to remove the noise in matrix  $P$ , a characteristic that becomes more and more crucial when the network represents more complex interactions. Although at this level of analysis this is only speculation, it is quite reasonable to assume that real networks are indeed complex ones. At the other hand, the lack of *a priori* information about  $P$  makes MANIA need to search for an optimal  $P$  in a space spanned by  $P_{\text{MNI}}$ , which increases its computation effort when compared to NIR or MNI + NIR. From the Algorithm in Sec. 3, the main time cost is in the line of the computation of SSE involving inversion which takes  $O(N^3)$ , thus, MANIA scales as  $O(N^4)$ . Therefore, MANIA can infer networks with up to about thousands of vertices in reasonable time. However, MANIA is easily parallelized (row-wise computations of  $A$ ), and can thus handle larger networks in reasonable time.

## 6. Conclusion

In this paper, a new reverse-engineering algorithm (MANIA) has been proposed, which effectively couples two assessed approaches MNI and NIR and overcomes their limitations, while preserving their performances. In more detail, MANIA uses the prediction of MNI to feed NIR and infer the network. Compared to NIR, MANIA has comparable performances in accuracy and the great advantage of **NOT** requiring *a priori* knowledge on the targets of the perturbations, which is very often an important unknown of the problem. In our simulation experiments, we have shown that MANIA can identify the network of interactions among genes from steady state experiments provided single perturbations are causing the expression variations. This covers several applications, like single gene knock-down or systematic small molecules testing [5], when assuming the perturbation affects a single target. These are two widely used experimental approaches with applications in chemo- and pharmaco-genomics and model organism research. Although MANIA performs encouragingly, it is worth noting that there is only one single gene perturbed in each experiment in the system under study. Our current work consists in the identification of a proper heuristic for extending this application to multiple perturbation targets and apply the validation to a larger variety of cases.

## Acknowledgments

Darong Lai and Hongtao Lu are supported by the Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP, No. 20050248048), the Program for New Century Excellent Talents in University (NCET-05-0397), and the National Natural Science Foundation of China (NSFC, No. 60873133).

## References

- [1] Margolin, A. A., Nemenman, I., Basso, K., Klein, U., Wiggins, C., Stolovitzky, G., Favera, R. D. and Califano, A., Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context, *BMC Bioinform.* **7** (2006) (Suppl 1):S7.
- [2] Guiducci, C. and Nardini, C., High parallelism, portability and broad accessibility, Technologies for genomics, *ACM J. Emerg. Technol. Comput. Syst.* **4**(1) (2008) Article 3.
- [3] di Bernardo, D., Thompson, M. J., Gardner, T. S., Chobot, S. E., Eastwood, E. L., Wojtovich, A. P., Elliott, S. J., Schaus, S. E. and Collins, J. J., Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks, *Nat. Biotechnol.* **23** (2005) 377–383.
- [4] <http://wiki.c2b2.columbia.edu/dream/index.php/Challenges>, *DREAM2* (2007).
- [5] Lamb, J., Crawford, E. D., Peck, D., Modell, J. W., Blat, I. C., Wrobel, M. J., Lerner, J., Brunet, J.-P., Subramanian, A., Ross, K. N., Reich, M., Hieronymus, H., Wei, G., Armstrong, S. A., Haggarty, S. J., Clemons, P. A., Wei, R., Carr, S. A., Lander, E. S. and Golub, T. R., The connectivity map: Using gene-expression signatures to connect small molecules, genes, and disease, *Science* **313**(5795) (2006) 1929–1935.
- [6] Yu, J., Smith, V. A., Wang, P. P., Hartemink, A. J. and Jarvis, E. D., Advances to bayesian network inference for generating causal networks from observational biological data, *Bioinformatics* **20**(18) (2004) 3594–3603.
- [7] Bansal, M., Belcastro, V., Ambesi-Impiombato, A. and di Bernardo, D., How to infer gene networks from expression profiles, *Mol. Syst. Biol.* **3** (2007), Article 78.
- [8] Bansal, M., Gatta, G. D. and di Bernardo, D., Inference of gene regulatory networks and compound mode of action from time course gene expression profiles, *Bioinformatics* **22**(7) (2006) 815–822.
- [9] Gardner, T. S., di Bernardo, D., Lorenz, D. and Collins, J. J., Inferring genetic networks and identifying compound mode of action via expression profiling, *Science* **301**(5629) (2003) 102–105.
- [10] Cover, T. M. and Thomas, J. A., *Elements of Information Theory* (John Wiley and Sons, 2001).
- [11] van Someren, E. P., Wessels, L. F. A., Reinders, M. J. T. and Backer, E., Searching for limited connectivity in genetic network models, *Proc. Second Int. Conf. on Systems Biology* **8**(9) (2001) 222–230 (Omnipress).