INFERRING GENE REGULATORY NETWORK STRUCTURE

Shreepriya Das, Manohar Shamaiah and Sharayu Moharir

Inferring the network structure of gene regulatory networks is one of the most important problems in contemporary bioinformatics. We analyze different methodologies for inferring small to very large sized gene networks. We use the datasets of DREAM 3 *in-silico* network challenge that is provided online [1]. The challenge involves inferring primarily the network structure from steady state gene expressions, knockout and knockdown and time-series data for gene networks of size 10, 50 and 100. The basis for all such inference methodologies in gene networks is the assumption that the network is sparse [2],[3]. We look at three formulations that involve time series data and knock-out data. We propose a heuristic for inference from knock out data only that provides reasonably good results on the datasets. We also compare the performance of our heuristic with that of other top teams and the results indicate that our proposed methodology fares in the top three rankings consistently.

1 Introduction

Gene regulatory networks (GRN) are systems of genes, mRNA and proteins that interact with each other and through those interactions determine gene expression levels i.e., determine the rate of gene transcription to mRNA and, consequently, the rate of mRNA translation to proteins. The signals in GRN are carried by molecules. Proteins which enable initiation of the gene transcription to mRNA (so-called transcription factors) can be considered as input signals. They bind to the promoter regions adjacent to the regulated gene and, in doing so, enable an RNA polymerase to perform the transcription. The gene expression levels obtained experimentally (typically from microarray experiments) can be regarded as the outputs of the system. Inferring the network structure of gene regulatory networks or determining pathways is crucial for the understanding of lots of important metabolic pathways for drug design, etc. GRNs are typically represented by a graph with the nodes representing the various genes/mRNA whose expression levels have been obtained. The problem then is to infer the presence or absence of edges between such nodes. In more sophisticated applications, it is necessary to determine these edge weights as well so as to predict the response of the network to different conditions.

2 The Data

We use the data provided by the DREAM 3 *in-silico* network challenge looking at 10 gene , 50 gene and 100 gene networks. The challenge provides the following kinds of data:

- Wild type: Provides the steady-state levels of the wild-type (the unperturbed network).

- Knockouts: Provides the steady-state levels of single-gene knockouts (deletions). An independent knockout is provided for every gene of the network. A knockout is simulated by setting the transcription rate of this gene to zero.[4]
- Knockdowns: Provides the steady-state levels of single-gene knockdowns. A knockdown of every gene of the network is simulated. Knockdowns are obtained by reducing the transcription rate of the corresponding gene by half.
- Time series: Provides time courses of the network recovering from several external perturbations. For the networks of size 10, 50 and 100 genes, 4, 23 and 46 perturbations respectively are provided (each one with 21 time points).

The perturbations applied here only affect about a third of all genes, but basal activation of these genes can be strongly increased or decreased. For example, these experiments could correspond to physical or chemical perturbations applied to the cells, which would cause (via regulatory mechanisms not explicitly modeled) some genes to have an increased or decreased basal activation. The genes that are directly targeted by the perturbation may then cause a change in the expression level of their downstream target genes.

3 Prior Work

In this section, we summarize the work of two of the best performers from the DREAM2 challenge.

3.1 NIRest

This approach is based on an ODE model of the network, with the assumption of linearity around the equilibrium point of the cell machinery [5]. This approach cannot be applied on knockout data since knockout is a result of very large perturbations. Therefore, the linearity assumption, generally employed in the case of small perturbations, does not hold. It is however suitable for the knockdown data and multifactorial perturbations. In the original NIR setup, it was assumed that the perturbations were known. This modified approach works without this assumption.
Let,

$$\frac{dx}{dt} = f(x, u), \tag{1}$$

where x is the mRNA concentration and u is a set of transcriptional perturbations. Assuming linearity,

$$\begin{aligned} \frac{dx_{il}}{dt} &= \sum_{j=1}^{m} a_{ij} x_{jl} + u_{il} \qquad (2) \\ &= \mathbf{a}_i \mathbf{x}_l + u_{il}, \\ & \quad i = 1, \ldots, N, l = 1, \ldots, M. \end{aligned}$$

Here, x_{il} is the mRNA concentration of gene i following the perturbation in experiment l, a_{ij} is the influence of gene j on gene i, u_{il} is the external perturbation to gene i in the experiment l. At steady state, $\frac{dx_{il}}{dt} = 0$, and therefore the solution of the system of linear equations $AX = -U$ gives an insight into which genes influence which other genes. NIR solves Eqn 2 by assuming an upper bound of 10 regressors for each predicted gene i.e. for each i, at most 10 $a_{ij}s$ are non-zero. The main contribution of NIRest is an algorithm to construct the matrix U. The first step is to find a rough estimate A_{est} based on the measure of the degree of the similarity between the expression profiles of each pair of genes and then approximate the perturbation matrix as $U_{est} = -A_{est}X$.

3.2 Using Threshold Logic for Pairwise Prediction

This technique [6] proposes the use of threshold logic to determine the network structure of gene networks. The directed graph representing this interaction between related genes is called the gene interaction graph. A gene regulatory model (GRM) describes the temporal and spatial characteristics of the biological system by modeling the genes involved in the processes of that system. One example is that genes have only two levels either on (expressed) or off (not expressed). The effects genes have on each other are captured by boolean functions. The rules involved in the gene regulations are threshold functions. A threshold logic function is completely characterized by the inputs, input weights and threshold. In order to determine the threshold gene regulation rule for the target gene g_t, the set of genes $G(g_t)$ that affect the expression of g_t, the input weights for each gene (in the set $G(g_t)$), and the threshold are determined. The genes are ordered in the decreasing order of their absolute correlation values with g_t. This sorts the genes in the order of their impact on g_t. The inputs that have the least impact on the output have the least absolute weight. The threshold of $T(g_t)$ is determined to be equal to the value of the least one point (The weighted sum =1). Using LMS perceptron, the weights are fine tuned to minimize the number of misclassified points. The gene addition stops if the error increases by the addition. This method is used only on the time-series data.

4 The Model

Gene regulatory networks are modeled in various ways. The most accurate models are based on Gillespie's Stochastic Simulation algorithm (i.e.the Markovian Chemical Master Equation) in which the discrete probabilistic nature of the chemical process is best captured. Often, the stochastic differential equation version is sufficient to capture most of the uncertainty in the time evolution of the system. Further simplification consists of approximating the aforementioned SDE by ODEs that are much more tractable when the number of unknown genes (and hence variables) in the system is large [7]. Finally, Boolean networks may be used where each node can be viewed as being in an "on" or "off" state. Previous studies on gene networks assume a linear model for the rate equations governing the time evolution of gene networks. Unfortunately, there may be significant nonlinearities even in the rate equation. Further, since genes regulate proteins that in turn may regulate other genes, it is possible that two or more genes are simultaneously necessary for regulation. To compound matters, the nonlinearities in certain genes may be so profound that the gene effectively acts as a switch, turning on when its regulatory gene concentration is above a given threshold and not expressing itself otherwise. Finally, since the system is causal, there may be an additional delay between the cause and its effect.

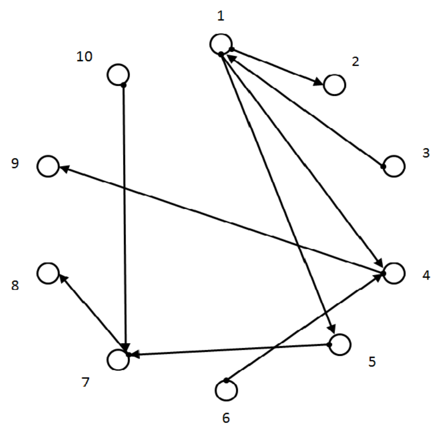

Figure 1: Directed graph for a subset of the yeast network showing sparsity

5 First Approach: Using Sparsity Constraints on Perturbed Time Series Data

In [9], a lasso based approach for gene network inference using time series data was proposed. The main motivation behind this approach was that a given gene is regulated by only few other genes and the lasso formulation exploits this. Our first approach was on similar lines, and we looked at time series data to infer a sparse model. We assume the model is of the form

$$\frac{dx_i}{dt} = \sum_{j=1}^{N} a_{ij} x_j + \epsilon_i \qquad (3)$$
$$i = 1, 2, ...N$$

where N is the number of genes. Alternatively,

$$\dot{x} = Ax + \epsilon \qquad (4)$$

5.1 Objective Formulation

Statistically, the number of genes affecting a given gene is less than 7. We use this sparsity constraint(call it s) to convert the link prediction problem to a lasso regression problem. We formulate the objective as

$$\begin{aligned} & \textit{minimize} \quad && ||Ax_{ss}||_2 + \sum_{i=1}^{T} ||\dot{x} - Ax||_2 \\ & \text{subject to} \quad && diagA \preceq 0 \\ & && \sum_{i=1}^{N}\sum_{j=1}^{N} |a_{ij}| \leq \rho \end{aligned} \tag{5}$$

where x_{ss} is a vector representing the steady state values. Note that the time series data has only 21 time steps. Consequently, for larger networks the system is grossly underdetermined. Even for smaller networks, if we pick data that is close to steady state, we end up with an underdetermined system. However, the sparsity constraint ensures that we get a unique solution under certain conditions.
All optimization problems are solved using *cvx*(a software for disciplined convex optimization). The diagonal constraint ensures that none of the links have autoregulatory behaviour i.e. no positive feedback occurs. As with lasso regression, we are now faced with a model selection problem. While different ρ values lead to different sparsity patterns, we need to choose a single ρ that correlates best with the actual links present in the network. In order to do so, the column elements of A were sorted in descending order of their absolute values. A threshold was defined for each column that was the s^{th} value of the sorted elements in each column. Multiple strategies were then used to ascertain a global threshold. These included the *max* function, the *min* function and the *mean* function. Having selected a threshold, all elements in A that were below this threshold were set to zero and the objective value was ascertained. Subsequently, ρ was swept through a large number of values and the A corresponding to the best objective was selected.

5.2 Comments

Little to no correlation was obtained between the links predicted and the actual links. We tried different window lengths for the time series data; mainly small windows near steady state and a global window which we expected would perform poorly because significant temporal rewiring of links would occur in the process. We also tried to put multiple time series data into the objective to make it overdetermined (this was possible only for the 10 gene network) but did not see any improvement. This was an approach previously tried in [3] but failed in this context. Possibly, sufficient structure was not imposed on the objective. Also, model selection, as always, could have been an issue.

6 Second Approach: Sparse Inverse Covariance Estimation using Graphical Lasso

The primary problem in the DREAM challenge involves finding the undirected links between interacting genes. We formulate the link prediction problem as a sparse graph with nodes corresponding to genes and presence of edges between nodes implying interacting genes. The inherent assumption is that the observations have a multivariate Gaussian distribution.
We use the formulation provided in [2] and use the graphical lasso algorithm described under. For

brevity, we omit the details of derivation (Refer to [8]). The covariance matrix (S) in our case is obtained in the usual manner from the time series data. Let W be the sparse covariance matrix estimate. Let W and S be partitioned as

$$W = \begin{bmatrix} W_{11} & w_{12} \\ w_{12}^T & w_{22} \end{bmatrix}, S = \begin{bmatrix} S_{11} & s_{12} \\ s_{12}^T & s_{22} \end{bmatrix}$$

where the lower block matrices are scalars. Let ρ be the regularizing parameter.

The Graphical lasso algorithm particularly emphasizes on the fact that estimating W involves

Algorithm 1 Graphical Lasso Algorithm

```
1. Start with $W = S + \rho I$. The diagonal of $W$ remains unchanged in what follows.
2. For each $j = 1, 2, ...N, 1, 2, ...N, ...$ solve the lasso problem
$minimize_{\beta} \frac{1}{2}||W_{11}^{\frac{1}{2}}\beta - b||^2 + \rho||\beta||_1$ where $b = W_{11}^{-\frac{1}{2}} s_{12}$
Input: The inner products $W_{11}$ and $s_{12}$
Output: A $N-1$ vector solution $\hat{\beta}$ .
Fill in the corresponding row and column of W using $w_{12} = W_{11}\hat{\beta}$.
3. Continue until convergence.
Convergence criteria: $|W^k - W^{k-1}| \leq t \cdot ave|S^{-diag}|$ where $W^k$ and $W^{k-1}$ are the $k$ and $k-1^{th}$
iterate, $t$ is a tuning parameter set to 0.001 and $ave|S^{-diag}|$ is the average of the absolute values
of the elements of $S$ with its diagonal values set to zero.
```

a coupled lasso problem, which it solves efficiently. Instead of a single ρ, we tried incorporating different ρ's for each column to allow for different sparsity patterns. Also, as in the previous approach, we used different windowing schemes to look for temporal as well as global correlation.

6.1 Comments

The graphical lasso algorithm fared poorly primarily because of the lack of a proper model selection scheme. Also, the covariance matrix obtained is a symmetric metric, whereas in practice, gene networks are rarely bidirectional. The absence of a nonsymmetric correlation metric and the inherent positive definite assumption of the inverse covariance matrix all possibly contribute to the poor performance of the method.

7 Selected Method: Inference using Knock-Out Data

In this section we describe the heuristic based on which we present our results of the gene regulatory network prediction. As already explained there are different types of data available in the challenge. Ideally, we would like to incorporate all this data towards inferring the network so as to exploit the different kinds of information in each of the data sets. But effective merging of the information from the different datasets is not obvious. The poor performance of all our algorithms on time-series data drives us towards using the knock out data for our inference purpose. The rationale behind our approach is as follows. Gene knock-out data consists of steady state gene expression values after setting the transcription rates for a given gene to zero. This contains the information regarding how gene expression values change in the presence and absence of another gene. We expect that a significant deviation from the steady state value of a gene i by knocking out gene j is

Algorithm 2 SMS-DM Algorithm

```
1. Find the gene expression change matrix ($\Delta G$)
2. Find the mean expression values for each gene ($\mu_G$)
3. Divide each row of $\Delta G$ by the corresponding mean expression value of the gene.
4. Find the maximum value of the absolute value of each row ($\beta_G$)
5. Let $\beta_{max}$ be the overall maximum i.e, $\beta_{max} = \max_{\beta_G}$
for target-gene = 1 to $N$ do
  sparsity(target-gene)=0;
  for influencing-gene = 1 to $N$ do
    if $\beta_G$(target-gene)$\leq \alpha_1 \beta_{max}$ then
      LINK(target-gene $\rightarrow$ 1:N)= 0
    end if
    if $|\Delta G$(target-gene,influencing-gene)$| \geq \alpha_2$(target-gene) $\beta_G$(target-gene)
    AND sparsity (target-gene) $\leq \gamma$ then
      LINK(target-gene $\rightarrow$ influencing-gene)=1;
      sparsity (target-gene)=sparsity (target-gene)+1;
    else
      LINK(target-gene $\rightarrow$ influencing-gene)=0;
    end if
  end for
end for
```

an indication of the influence of the gene j on gene i (note that this is similar to “intervention” in the language of graphical models). This information also comes with the limitation that influence might be either due to direct or indirect interaction. Again we suppose that stronger interaction would result in larger changes from the steady state value. We also assume that the number of genes influencing a particular gene is small compared to the total number of genes. This is the sparsity inherent in gene networks and has been corroborated by several studies. Our proposed heuristic to infer the gene regulatory network from the knock-out data is given in algorithm 2. Note that we tried a similar scheme for knock-down data but that fared poorly.

In the algorithm 2, $(\alpha_1, \alpha_2, \gamma)$ are tuning parameters. α_1 is used to directly infer the genes which are not regulated by any other gene (but they can regulate other genes). The corresponding rows in the ΔG matrix contain very small values (relatively) and are eliminated by the constraint imposed by α_1. We can choose α_1 by sorting β_G and setting a threshold such that the least significant values are eliminated. α_2 and γ are used to control the number of genes regulating a given gene. γ directly controls this number and ideally should be around 7. However, such a stringent setting leads to the elimination of lots of correct links and therefore we set a conservative number between 20 and 30. α_2 controls the number of genes regulating a given gene based on the absolute values of the elements in the matrix ΔG. If a particular value is significantly less compared to the maximum in that row, then we eliminate that link. One can obtain this from the empirical evidence from each row independently. For small sized networks (eg 10 gene network), we neglect the sparsity constraint. We empirically choose α_1,α_2 and γ to be 0.01, 0.3 and 30.

We next discuss the metrics and present the results for the different datasets (5 different networks named *Ecoli1, Ecoli2, Yeast1, Yeast2, Yeast3* of sizes 10, 50 and 100) based on this heuristic and also provide the performance comparison of our results with the best performers using the metrics provided by the website [1]. Our results are competitive with the second and third best performers in the competition. The metrics are explained in section 8 and the results are given in section 9.

8 Metrics [1]

- AUROC: Area under the receiver operating characteristic (ROC) curve. The axes of the ROC curve are the false positive rate (x-axis) and the true positive rate (y-axis).
- AUPR: Area under the precision-recall curve. Precision is a measure of fidelity whereas recall is a measure of completeness.
- P-value: The probability that a given or larger area under the curve value is obtained by random ordering of the T potential network links. Distributions for AUROC and AUPR were estimated from 100,000 instances of random network link permutations.
- Overall P-value: The geometric mean of the n individual p-values, computed as $\left(\prod_{i=1}^{n} p_i\right)^{\frac{1}{n}}$.
- Overall AUROC P-value: The geometric mean of the five AUROC p-values (*Ecoli1, Ecoli2, Yeast1, Yeast2, Yeast3*).
- Overall AUPR P-value: The geometric mean of the five AUPR p-values (*Ecoli1, Ecoli2, Yeast1, Yeast2, Yeast3*).
- Score: A log-transformed "average" of the two overall AUROC and AUPR P-values, computed as -0.5 log10 (P1*P2). Larger scores indicate greater statistical significance of the prediction.

9 Results

Data Set	AUPR	AUROC	P-AUPR	P-AUROC
Ecoli1	0.4192	0.8251	0.0017	1.5203e-004
Ecoli2	0.4227	0.8436	0.0022	1.3118e-005
Yeast1	0.4353	0.8581	0.0012	6.9228e-005
Yeast2	0.4504	0.6335	0.0106	0.0240
Yeast3	0.3821	0.5983	0.0292	0.0845

Table 1: Results for Datasets of size 10

Data Set	AUPR	AUROC	P-AUPR	P-AUROC
Ecoli1	0.3439	0.9133	1.5992e-027	1.7069e-025
Ecoli2	0.2879	0.8772	3.9124e-023	2.8353e-025
Yeast1	0.2932	0.8296	5.9570e-023	1.6020e-017
Yeast2	0.1835	0.7239	1.6840e-015	1.9160e-014
Yeast3	0.1852	0.7050	1.8580e-012	2.1271e-015

Table 2: Results for Datasets of size 50

Data Set	AUPR	AUROC	P-AUPR	P-AUROC
Ecoli1	0.1911	0.8583	1.0827e-040	2.7824e-035
Ecoli2	0.1895	0.9127	3.2421e-040	5.7112e-035
Yeast1	0.1660	0.8131	2.6266e-037	9.6214e-041
Yeast2	0.2129	0.7889	2.3393e-092	2.2393e-067
Yeast3	0.1513	0.7114	8.5582e-055	9.5058e-054

Table 3: Results for Datasets of size 100

Team	Data size	Score	Overall AUPR	Overall AUROC
Team 315 (I)	10	5.124	5.925e-06	9.523e-06
Team 291 (II)	10	3.821	1.085e-04	2.103e-04
SMS-DM Algorithm	10	2.7410	0.0043	7.7523e-004
Team 304 (III)	10	2.188	1.986e-02	2.117e-03
Team 315 (I)	50	39.828	4.246e-54	5.210e-27
Team 291 (II)	50	31.341	2.539e-46	8.192e-18
SMS-DM Algorithm	50	19.2434	2.5903e-020	1.2587e-019
Team 256 (III)	50	17.927	4.956e-25	2.828e-12
Team 315 (I)	100	10e10	0.000e+00	3.112e-71
SMS-DM Algorithm	100	49.0221	2.0068e-046	4.5003e-053
Team 296 (II)	100	45.443	3.982e-56	3.260e-36

Table 4: Comparison with the best performers

10 Effect of tuning parameters α_1 and α_2

Having already obtained our predicted results, we look at the effect of α_1 and α_2 on the performance.
The figures 2 and 3 indicate how crucial the tuning parameter is on AUROC/AUPR. The choice of α_1 is easy in all the datasets that we worked on, but the choice of α_2 is more difficult. However, the plots clearly show that the degradation with α_2 is more gradual. Hence, we expect these tuning parameters to be more robust.

11 Discussion and Future Work

We propose a new heuristic based on knock-out data that performs well for the DREAM 3 challenge. As with heuristics, there are some tuning parameters which need to be empirically decided upon. We plot the effect of these tuning parameters on small and large scale networks and show that we can choose these parameters such that the sensitivity of the performance to these parameters is low.

Although our heuristic based on knock-out data gives competitive results, we were particularly disappointed by the poor performance of some of our other formulations on time series data. In hindsight, any method that seeks to provide structure to the problem tends to perform poorly. Also, the datasets tended to be quite noisy and no serious effort was made on our part to smoothen out the time series data. Significant effort is required to generate correlation metrics from time-series data that detects causality. Another aspect of the problem that we havent looked at is efficient merging of results obtained from different datasets i.e. time-series, knock-out and knock-down.

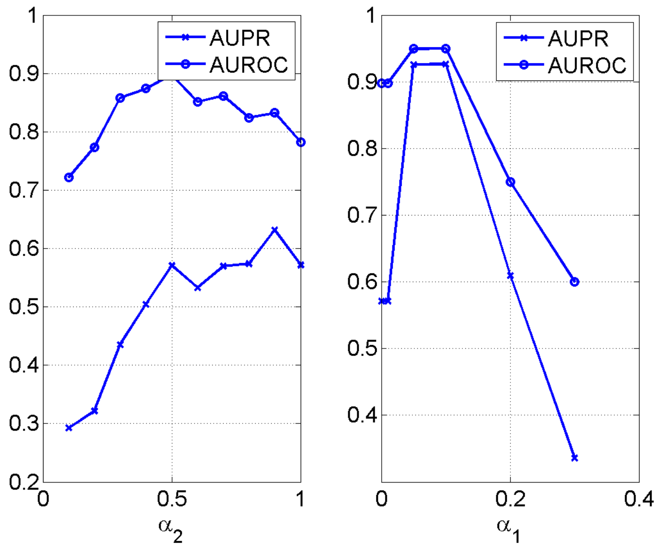

Figure 2: Effect of tuning parameters α_1 and α_2 on $AUROC$ and $AUPR$ for a 10 gene network

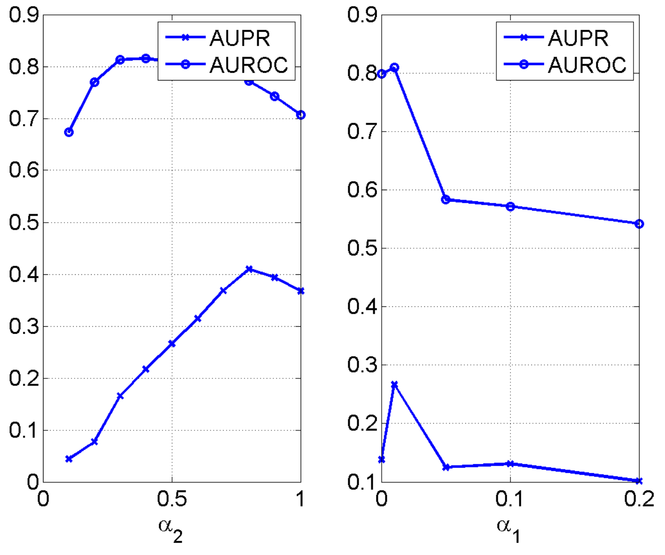

Figure 3: Effect of tuning parameters α_1 and α_2 on $AUROC$ and $AUPR$ for a 100 gene network

Hopefully, we will be able to address these issues in the recent future and obtain results bettering the best performance for these datasets.

References

[1] http://wiki.c2b2.columbia.edu/dream/index.php/BCL6_Transcriptional-Target_Challenge._Description

[2] J Friedman, T Hastie and R Tibshirani, (2007) "Sparse inverse covariance estimation with the graphical lasso", *Biostatistics.*

[3] A. Ahmed and E. P. Xing, (2009) "TESLA: Recovering Time-Varying Networks of Dependencies in Social and Biological Studies", *PNAS.*

[4] F. Geier, J. Timmer, and C. Fleck (2007) "Reconstructing gene-regulatory networks from time series, knock-out data, and prior knowledge", *BMC Systems Biology.*

[5] M. Lauria, F. Iorio and D. di Bernardo (2009) "NIRest: A Tool for Gene Network and Mode of Action Inference", *The Challenges of System Biology.*

[6] T Gowda, S Vrudhula, and S Kim (2009) "Modeling of Gene Regulatory Network Dynamics Using Threshold Logic", *The Challenges of System Biology.*

[7] M. Quach , N. Brunel and F. d'Alch-Buc (2007) "Estimating parameters and hidden variables in non-linear state-space models based on ODEs for biological networks inference", *Bioinformatics.*

[8] O. Banerjee, L. E. Ghaoui, and A. D'aspremont, (2007) " Model selection through sparse maximum likelihood estimation", *The Journal of Machine Learning Research.*

[9] S. Bhadra, C. Bhattacharyya, N. R Chandra and I. S. Mian, (2009) "A linear programming approach for estimating the structure of a sparse linear genetic network from transcript profiling data", *Algorithms for Molecular Biology.*

[10] A. Werhli, M. Grzegorczyk and D. Husmeier (2006) "Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks", *Bioinformatics.*

[11] J. Tegnr and J. Bjrkegren (2006) "Perturbations to uncover gene networks", *Trends in Genetics.*

[12] G. Stolovitzky, R. Prill and A. Califano (2009) "Lessons from the DREAM2 Challenges", *The Challenges of System Biology.*