Proceedings of the 30th Chinese Control Conference
July 22-24, 2011, Yantai, China

An Integrated Gene Regulatory Network Inference Pipeline

WANG Yong[1,2,4], HORIMOTO Katsuhisa[2], CHEN Luonan[3,4]

1. Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, P. R. China
2. Computational Biology Research Center (CBRC), National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, 135-0064, Japan
3. Key Laboratory of Systems Biology, SIBS-Novo Nordisk Translational Research Centre for PreDiabetes, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, 320 Yue-Yang Road, Shanghai 200031, P. R. China
4. Kavli Institute for Theoretical Physics China, CAS, Beijing 100190, P. R. China

E-mail: ywang@amss.ac.cn, k.horimoto@aist.go.jp, lnchen@sibs.ac.cn

Abstract: Rapidly accumulated gene expression data put forward the development of numerous methods for inferring gene regulatory networks and the efforts for critical performance assessment of these methods. In this paper, we propose an integrated pipeline for gene regulatory network inference motivated by the results and follow-up analysis of a blinded, community-wide challenge DREAM (Dialogue on Reverse Engineering Assessment and Methods) project. In particular, we categorize the gene expression data into three types, i.e., steady-state gene expression profile of knockout or knockdown experiments, steady-state gene expression profiles after multi-factorial perturbations, and time-series data after multi-factorial perturbations. Then we analyze the three types of gene expression data by using the combination of fold change and t-test, the path consistency algorithm based on conditional mutual information, and the ordinary differential equation model, respectively. Finally we integrate the three procedures to a pipeline for gene regulatory network inference by considering their complementarities. Performance for the network inference will be improved in the proposed pipeline by maximally utilizing information in the available data, emphasizing the knock-out and knock-down data, and differentiating the direct and indirect regulatory interactions.

Key Words: Gene regulatory network, Reverse engineering, Knock out, Steady state, Time series

1 Introduction

Recent bio-technologies enable us to successfully discover the functional organization of a cell by simultaneously measuring the dynamic change of concentrations of thousands of genes after multi-factorial perturbations (gene, drug, and other environmental factors). Such available input-output data offer us great chances and meanwhile challenges to reconstruct the gene regulatory networks in an accurate and reliable manner. As s result, there are recently a lot of network inference methods that infer gene regulatory network by different strategies including correlation-based methods, information-theoretic methods, Bayesian network predictions, and ordinal differential equation (ODE) models based method [1-3]. However, Marbach et al. indicate that reliable gene network inference from gene expression data remains an unsolved problem by extensively analyzing the different strengths and weakness of individual methods [4]. Their conclusion is based on the efforts of DREAM (Dialogue on Reverse Engineering Assessment and Methods) project, which aims at the development of community-wide challenges for objective assessment of reverse engineering methods for biological networks [5-7].

Most importantly, the DREAM project indicates several potential ways of network reconstruction improvements. In particular, the two major difficulties have been realized in gene-network reverse engineering. One is the limitations of available gene expression data, which may make the inference problem mathematically underdetermined. The other is the difficulty of distinguishing between direct and indirect regulations.

Here we propose an integrated network inference pipeline motivated by the lessons learned from DREAM project. to overwhelm the above difficulties. Indeed, the main contributions in our new pipeline are as follows: 1), we maximally utilize the information on the limited gene expression data by categorizing the data into three types and developing three methods accordingly for information mining;2), we propose a path consistency algorithm based on conditional mutual information to differentiate the direct and indirect regulatory interactions; and 3), we integrate three methods into a pipeline by considering their complementarities, to attain high inference accuracy.

2 Method

One of our main motivations is to take the full advantage of all the available gene expression data, given the fact that data is scarce regarding to the large parameter space for network representation. We observed that three types of steady-state and time-series gene expression data are commonly used for gene regulatory network inference: steady state expression levels of the unperturbed network and minor perturbations, steady-state levels of knockout and knockdown experiments for every gene, and time-series data showing how the network responses from multi-factorial perturbations. It should be noted that

*This work is supported by National Natural Science Foundation of China (NSFC) under Grant 10801131. This research was also supported in part by the Project of Knowledge Innovation Program (PKIP) of Chinese Academy of Sciences, Grant No. KJCX2.YW.W10

knockout and knockdown data are usually regarded as steady data under perturbations. However, recent DREAM project revealed that the steady-state levels of the gene knockout experiments seemed the most informative: the score of the best-performer team in DREAM was mainly due to predictions derived from the knockout datasets and not those from the time-series [4]. Thus, it is important to treat the knockout data separately here.

One of the most remarkable features of our method is that we categorize the gene expression data into three types. The first type is the steady-state gene expression profiles of knock-out or knock down experiments. The second one is the steady-state gene expression profiles after multi-factorial perturbations, and The third onet is the time-series data after multi-factorial perturbations. Accordingly we develop three methods to extract useful information from the three types of gene expression data, respectively, for inferring gene regulatory relationships. Finally we propose an inference pipeline to systematically integrate the predictive results for the three methods and output the reconstructed gene regulatory network. The above procedure is schematically shown in Figure 1.

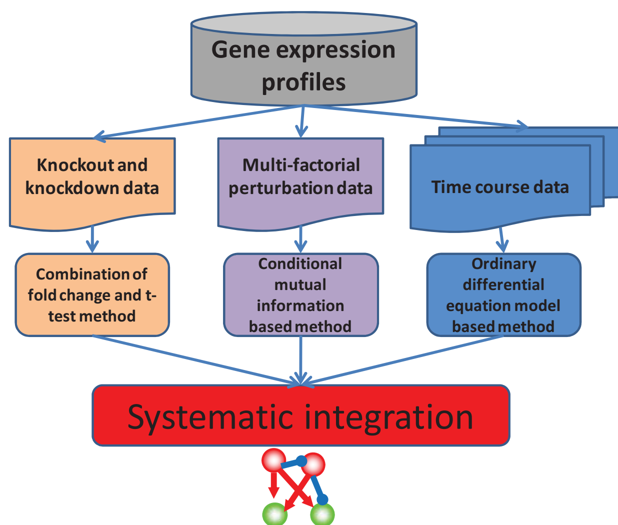

Figure 1. Schematic plot of the proposed gene regulatory network inference pipeline.

2.1 Applying the combination of fold change and t-test method to knockout data

Gene knockout(or knockdown) provides a genetic technique in which one of an organism's genes is made inoperative. Then the follow-up microarray technique measures the expression change of the remaining genes. The idea is natural to identify the influence of one gene on the expressions of the other genes: if a gene x_i is a target gene , then the expression levels of x and the othere genes can be compared with those after the knock out. Here we suppose that the wild-type expression values of gene x_i in N samples are $X^{wt}=(x_{i1}^{wt}, x_{i2}^{wt}, \ldots, x_{iN}^{wt})$ and the measured expression value of x_i after the knockout of x_j is x_{ij}^{ko}. We can simply infer the regulatory relationship in the the following case: gene x_j regulates gene x_i, if the significant change of gene expression of x_j is detected in wild-type and knockout conditions. To this aim, we can then use the t-statistic by calculating the t-score T_{ij} for the regulatory relationship between gene x_j and x_i as,

$$T_{ij}= (x_{ij}^{ko}- (x_{i1}^{wt}+ x_{i2}^{wt}+ \ldots + x_{iN}^{wt})/N)/\sigma \quad (1)$$

where σ is the standard deviation of gene x_i over all wild-type and single gene knockout data. Once a T_{ij} value is determined, a p-value can be found by using a table of values from Student's t-distribution. The expression level of gene x_i is statistically significantly changed if the p-value score is smaller than a chosen cutoff. To further quantify the extent of the gene expression change after the knockout experiment, we also calculate a fold change score F_{ij}, defined as the ratio of the observed gene expression value after knockout versus the wild-type gene expression value, as follows:

$$F_{ij}= x_{ij}^{ko}/(x_1^{wt}+ x_2^{wt}+ \ldots + x_N^{wt})/N) \quad (2)$$

Both p-value and fold change scores are predictive for regulatory relationship of two genes. We further combine the p-value score and the fold change score into a single criterion: if the p-value score and the fold change score are both more significant than the corresponding pre-defined cutoffs (p-value score is less than 0.05. And fold change score is larger than 2.0 when gene x_j negatively regulate x_i and is smaller than 0.5 when gene x_j positively regulate x_i), then two genes have the regulatory interaction.

As a result, the combination of fold change and t-test will generate a list of regulatory relationships, which can be regarded as prior information in our proposed regulatory network inference pipeline. One of the merits in the prior information is that these regulatory interactions have directions, i.e. the knocked out gene will regulate the affected gene. From the expression value change of the target gene, therefore, we can determine the repression (up-regulated) or activation (down regulation) role of the regulation.

2.2 Applying the path consistency algorithm based on conditional mutual information to steady state data

Perturbation is a powerful tool in systems biology study, since cells can be dramatically perturbed by many environmental factors. Usually microarray technique is then followed to capture a snapshot when the cell goes from one steady state to another one. Analyses of multi-factor influenced genome-wide gene expression data offer a first glance at which genes and pathways are affected and the correlation relationships among genes across different conditions.

There are two difficulties for the analysis of the steady state gene expression data under minor perturbations. The first difficulty is how to find the non-linear relationships due to time-delay and other complicated factors. Here we propose a mutual information based framework to identify the possible co-expression interactions among genes. The second difficulty is how to identify the causal relationships

from the large dataset of co-expressions revealed by mutual information. Here we use a simple method, called path consistency algorithm based on conditional mutual information, to infer a causal graph by further assessing the co-expressed gene pairs [8].

The concept of mutual information is quite complex and is the basis of information theory. In essence, mutual information provides a general measure of dependence between two variables. Our aim is to deduce the correlation, dependence, or any of a broad class of statistical relationships between two genes from the observed steady state gene expression samples.

Going beyond the first step of correlation coefficients calculation which tries to determine if two random variables are linearly related, mutual information helps reducing the range of the probability density function (reduction in the uncertainty) for a random variable X if the variable Y is known. The value of I(X,Y) is relative, and the large value indicates the more information that is known of X. Then we can safely deduce if two genes has a regulatory relationship by checking if I(X,Y) is larger than a predefined threshold.

Here we estimate mutual information by using a computational efficient Guassian kernel estimator which is superior to the classical histogram methods in several aspects. Suppose that we have N samples (S_1 S_2 …, S_N), for n genes, the joint probability distribution of is approximated as

$$\hat{p}(S) = \frac{1}{N}\sum_{j=1}^{N} K(S - S_j) \qquad (3)$$

where S is the n-dimensional random vector whose density is being estimated, and K is the kernel with dimensionality n and covariance matrix Σ,

$$K(S) = N(0,\Sigma) = \frac{1}{(2\pi)^{n/2}\,|\Sigma|^{1/2}}\exp(-\frac{1}{2}S^T\Sigma^{-1}S) \qquad (4)$$

Then we can calculate the mutual information of gene X and Y as follows.

1) Form the gene expression value pairs (x_i, y_i), i=1,2, …, N.

2) For i=2,…,N, estimate the probabilities $P_x(x_i)$, $P_y(y_i)$,$P_{x,y}(x_i,y_i)$ at the sample point using (3)-(4) and sample estimates of Σ_x, Σ_y, $\Sigma_{x,y}$.

3) The overall dependence between the two genes can be estimated by mutual information as

$$I(X,Y) = \Sigma_{i,j} P_{x,y}(x_i, y_j)\log_2[\frac{P_{x,y}(x_i, y_j)}{P_x(x_i)P_y(y_j)}] \qquad (5)$$

One limitation of mutual information in detecting gene regulatory relationship is that it can only provide the pairwise correlation information. As we know, gene regulatory network is complex by possessing a lot of feedback loop and feedforward loop structures, which involve the interactions among more than three genes. As a result, some pairwise regulatory relationship identified by mutual information should be carefully checked. For example, Figure 2 shows a feedforward loop structure which is very common in gene regulatory network, so called network motif. By calculating the mutual information, we will find that all of the I(X, Y), I(X,Z), and I(Z,Y) have larger values and naturally deduce the regulatory relationships between gene A and gene B, gene B and gene C, and gene A and gene C. However, we will find that the correlation between gene A and gene C may be caused by the regulatory role of gene B, i.e., it's a correlation relationship but not a causal relationship. The reason is why we didn't consider the effect of gene B when calculating I(X,Y). We then ask the question whether there is still correlation between gene A and gene C after considering the expression pattern of gene B. Conditional mutual information provide a power tool to calculate the expected value of the mutual information of two random variables given the value of a third, and basically serves our need to remove the non-causal relationships.

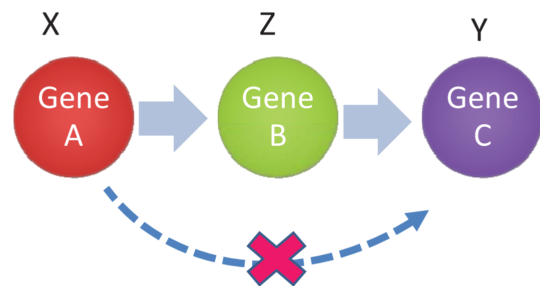

Figure 2. Motivation to introduce conditional mutual information to distinguish the causal relationship from the simple correlation relationship.

Mathematically, conditional mutual information then can be calculated to reveal the dependence of genes X and Y conditioning on a third gene Z as follows,

$$I(X,Y|Z) = H(X,Z) + H(Y,Z) - H(Z) - H(X,Y,Z), \qquad (6)$$

$$H(\mathrm{X}) = -\sum_{i=1}^{N} P(x_i)\ln(P(x_i))$$

In Figure 2, if the calculated I(X, Y|Z) value is small, we can claim that the correlation between gene A and gene C can be explained by gene B. Then the regulatory relationship between gene A and gene C can be safely removed.

In addition to the condition mutual information in equation (6) by considering one gene one time (first-order conditional mutual information), one can also similarly calculate the higher order conditional mutual information by checking whether a given pairwise correlation can be explained by a group of genes. In this way we use conditional mutual

information, in terms of an information theoretic concept, to approximately define causal information.

Given n genes together, we use path consistency algorithm, a systematic procedure, to deduce the regulatory network underlying the n genes. Originally the path consistency algorithm is a method to infer an undirected graph by the partial correlation coefficient. Here we use the conditional mutual information instead. The brief steps for the path consistency algorithm are as follows:

Step 1. Calculate the Pearson correlation coefficients for every pair of genes and reconstruct a gene co-expression network by connecting two genes with higher correlation than a predefined cutoff. Find a complete undirected subgraph (clique) with m nodes in the co-expression network.

Step 2. Calculate the zeroth-order conditional mutual information (for example mutual information of gene X and Y) and delete the edges that are independent by applying a predefined threshold.

Step 3. Calculate the first-order conditional mutual information (for example mutual information of gene X and Y conditioning Z) and delete the edges that are independent.

Step 4. Calculate the higher order conditional mutual information and terminate when there is no edges can be deleted.

Basically we start from the dense co-expression network derived from simple correlation coefficient and gradually filter the redundant edges by recursively applying the conditional mutual information concept. Here we adopt a divide and conquer strategy combined with greedy deletion of edges. The advantage is that we can avoid the intensive computation in the higher order condition mutual information.

As a result, the combination of conditional mutual information concept and a systematic path consistency algorithm will generate a list of regulatory relationships, which will be further feed as prior information in our proposed regulatory network inference pipeline. These regulatory interactions are undirected, i.e., we know two genes are mutually correlated, but we don't know any causal relationship between the two genes. To assemble the regulatory network, we need additional information to assign the directions and also determine the repression or activation roles of these edges.

2.3 Applying ODE model to time-series data

In the previous subsections, we propose two methods to deal with knock-out and multi-factor steady state gene expression data, which can be treated as the static gene expression data. In static expression experiments, a snapshot of the expression of genes in different samples is measured. In this subsection, we will deal with time series of gene expressions , in which a temporal process is measured by time series of expression experiments. An important difference between these two types of data is that while the static data are from a sample population under different perturbations and can be assumed to be independent identically distributed, the time series data exhibit a strong autocorrelation between successive points. In the modeling process, we should utilize the correlation and capture the dynamic behavior of genes.

For the time-series data, we consider the ordinary differential equation model to capture the dynamic relationship among genes [9]. A linear differential equation can be used to represent the rate of synthesis of a transcript as a function of the concentrations of other transcripts in a cell and the external perturbations:

$$\frac{d\mathbf{X}}{dt} = J\mathbf{X} + P\mathbf{C} \tag{7}$$

where $X=(x(t_1),...,x(t_m))$ and $dX/dt=(dx(t_1)/dt,...,dx(t_m)/dt)$ are $n \times n$ matrices with the first derivative of mRNA concentration $dx_i(t_j)/dt=[x_i(t_{j+1})-x_i(t_j)]/[t_{j+1}-t_j]$ for $i=1,...,n$; $j=1,...,m$ is the forward difference approximation. Suppose that there are s external perturbation compounds, then $C=(c(t_1),...,c(t_m))$ is an $s \times m$ matrix representing the s perturbations. The connectivity matrix J and P are unknown to be calculated. In our differential equation model (7) we use the Jacobian matrix J to represent the gene regulatory relationships that can be directed, signed, and weighted. For example, element J_{ij} represents an effect of gene j on gene i, while J_{ji} represents an effect of gene i on gene j. Thus the influence between gene i and gene j is directed. Furthermore, a sign associated with J_{ij} represents a specific role of regulation. For example, if the sign of J_{ij} is positive, gene j is the activator of gene i. On the other hand, if the sign of J_{ij} is negative, gene j is the repressor of gene i. Furthermore the associated weight (the absolute value) of element J_{ij} indicates strength of the regulatory interaction. Obviously, a zero weight of J_{ij} indicates no interaction between two genes.

Assume that there are N microarray datasets $X_1, X_2 \dots, X_N$ with $m_1, m_2, \dots, m_N$ time points, respectively, from one organism. According to Equation (7), N networks can be separately inferred from N time-course datasets (The meaning of equation (8) is detailed in [9]):

$$[J^k, P^k] = \frac{dX^k}{dt} U^k S^{k^{-1}} V^{k^T} = [\overline{J^k}, \overline{P^k}] + Y^k V^{k^T} \tag{8}$$

where the superscript $k=1,...,N$ is the index of the k-th dataset. Next, we will derive a sparse network structure $L=[J, P]=(L_{ij})_{n\times(n+s)}$ that is the most consistent with $L^k=[J^k, P^k]=(L^k_{ij})_{n\times(n+s)}$ for $k=1,...,N$, as well as consistent with the prior information (directed and signed, directed and unsigned regulations, and non-regulatory relationships between genes from literature or other experimental data sources, or computational predictions). Mathematically the problem can be formulated as:

$$\min_{Y^1,Y^2,\ldots,Y^N,L} \sum_{k=1}^{N}\sum_{i=1}^{n}\sum_{j=1}^{n+s}\omega_k\,|L_{ij}-L_{ij}^k| + \lambda \sum_{(i,j)\in\{(i,j)|K_{ij}=0 \text{ or } U_{ij}=0\}} |L_{ij}|$$

$$\begin{aligned} s.t.\quad & L_{ij}>0 \quad \text{if} \quad K_{ij}>0 \quad i,j\in\{1,2,\ldots,n\} \\ & L_{ij}<0 \quad \text{if} \quad K_{ij}<0 \quad i,j\in\{1,2,\ldots,n\} \\ & L_{ij}=0 \quad \text{if} \quad E_{ij}=0 \quad i,j\in\{1,2,\ldots,n\} \end{aligned} \tag{9}$$

where L^k_{ij} is a function of Y^k, and $Y=(Y^1,\ldots,Y^N)$. The objective function has two terms. The first term is a matching term which forces the matching of L and L^k, whereas the second term is a sparseness term which forces L to be sparse as a result of the minimization of the sum of L1 norm. λ is a positive parameter, which balances the matching and sparseness terms in the objective function.

Here the prior information about the regulatory relationships (K, U, and E) is added into the objective function and constraints in the linear programming model in an explicit way. The soft constraints (U) are added into the objective function in an implicit way, by removing the related sparseness terms of the objective function in Equation (9). The hard constraints (K and E) are added as inequality or equality constraints in an explicit way. The first and second constraints are used to add the directed and signed information, and the third one is used to incorporate the non-regulatory relationship information.

The variables in (9) are L_{ij} and all of nonzero Y^k_{ij}. ω_k is a positive weight coefficient for the k-th dataset and $\sum N^k=1\ \omega_k$ =1. Since different datasets may have different data qualities (e.g., different technologies, the number of repeats in measurements, etc.), the weight coefficient is used to represent the reliability of each dataset. The optimization problem (9) is an LP with L1 norm, which is a well-studied problem. It is known that L1 gives a more robust answer compared with L2. The L1-norm is more robust to outliers than the L2-norm and does not penalize large deviations as much as the L2-norm. As a result, the L1-norm pays less attention to the parts of the regulatory interactions that are very different, and focuses more on the parts of the regulatory interactions that are conserved. As a result, this measure is less sensitive toward noise and more robust towards outliers. Generally the optimal solution of (9) sets as many $|L_{ij}- L^k_{ij}|$ and $|L_{ij}|$ to zero as possible, thus ensuring a consistent and sparse structure for the inferred gene regulatory network.

The detailed formulation, the construction of the prior information matrix (U, K, and E), and the efficient algorithm can be referred in [9-12]. In a word, one can derive the matrix L in (9) and then matrix J in equation (7) by solving the optimization problem (9) using a linear programming transformation. Here the final output $J=(J_{ij})_{n\times n}$ is an $n\times n$ connectivity matrix with elements J_{ij} representing the effect of gene j on gene i with a positive, zero, or negative sign, indicating activation, no interaction, and repression, respectively. In this way all the regulatory relationships in J has direction, repression or activation information, and the regulatory strength.

2.4 An integrated network inference pipeline

After categorizing the current available gene expression datasets into three classes, we develop three methods respectively to model and predict the regulatory relationships. Next we will consider propose a gene regulatory network inference pipeline to integrate the three data types of data and the three different methods.

We denote the three types of gene expression data as D_1 (knockout and knockdown data), D_2 (steady state data), and D_3 (time series data). And the three respective methods are M_1 (combination of fold change and t-test), M_2 (path consistency algorithm based on conditional mutual information), and M_3 (ODE modeling of time series data). Then the network inference pipeline to integrate three types of gene expression data are described as the posterior probability of $M=[M_1, M_2, M_3]$ and given data $D=[D_1, D_2, D_3]$ as P(M|D). By applying the Bayesian rule, we have

P(M|D)= P(D|M)P(M)/P(D), where P(M) and P(D) are prior probability. Then by considering the conditional independent relationship among the three different types of gene expression data and corresponding methods, we have the following equation:

$$P(D|M)=P(D_1,D_2,D_3|M_1,M_2,M_3)=P(D_1|M_1,M_2,M_3)P(D_2|D_1,M_1,M_2,M_3)P(D_3|D_1,D_2,M_1,M_2,M_3)=P(D_1|M_1)P(D_2|M_2)P(D_3|D_1,D_2,M_1,M_2,M_3) \tag{10}$$

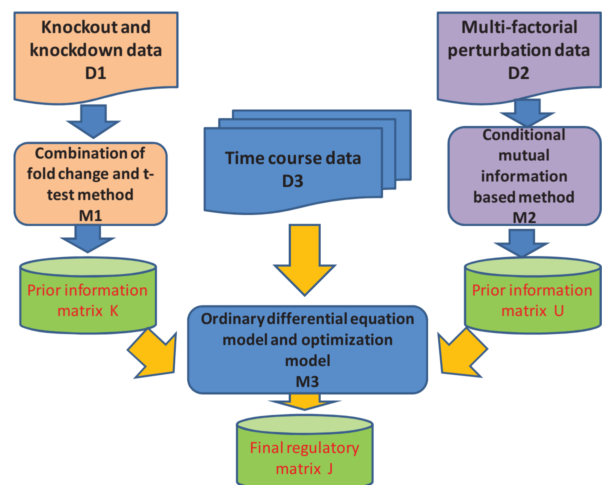

Figure 3. Regulatory network inference pipeline integrates the three types of gene expression data and three corresponding methods.

The above integration pipeline (10) means that we use method M_1 to analyze D_1, M_2 to deal with D_2, and then feed the results of M_1 and M_2 to M_3 as prior information matrix K and U. Finally we use the M_3 to model D_3 by considering the prior information obtained from M_1 and M_2. Figure 3 schematically illustrates the integration strategy. The advantage of this pipeline is that it can fully take advantage of the informative knockout data, the identification of the direct and causal regulations by conditional mutual information, and final assignment of directions and weights to all the regulatory interactions predicted.

3 Discussions and conclusions

In this paper we propose a new network inference pipeline to extensively integrate all available gene expression data and dig all the possible information from the individual data sources. As mentioned in [4], the current inference methods are affected, to various degrees, by three types of systematic prediction errors: the fan-out error (incorrect prediction of interactions between co-regulated genes), the fan-in error (inaccurate prediction of combinatorial regulation), and the cascade error (failure to distinguish direct from indirect regulation). We believe that our conditional mutual information concept is useful to minimize the fan-out error by revealing the hidden nonlinear relationships among genes. Our ODE model with prior information can predict the combinatorial regulations. Our path consistency algorithm can reduce the fan-in error by using conditional independence to remove the indirect regulations. Collectively, performance improvement of the proposed network inference pipeline can be expected by maximally utilizing information in the available data, emphasizing the knock-out and knock-down data, and differentiating the direct and indirect regulatory interactions.

References

[1] H. de Jong. Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol.*, 2002, 9,67-103.

[2] T. S. Gardner, J. J. Faith. Reverse-engineering transcription control networks. *Physics of Life Reviews*, 2005, 2:65-88.

[3] L. Chen, R. S. Wang, X.S. Zhang. Biomolecular Networks: *Methods and Applications in Systems Biology*. John Wiley & Sons, Hoboken, New Jersey. July, 2009.

[4] Marbach, D., et al., *Revealing strengths and weaknesses of methods for gene network inference.* Proceedings of the National Academy of Sciences, 2010. **107**(14): p. 6286.

[5] Baralla, A., W.I. Mentzen, and A. De La Fuente, *Inferring Gene Networks: Dream or Nightmare?* Annals of the New York Academy of Sciences, 2009. **1158**: p. 246-256.

[6] Greenfield, A., et al., *DREAM4: Combining Genetic and Dynamic Information to Identify Biological Networks and Dynamical Models.* PloS one, 2010. **5**(10): p. e13397.

[7] Madar, A., et al., *DREAM3: Network Inference Using Dynamic Context Likelihood of Relatedness and the Inferelator.* PloS one, 2010. **5**(3): p. 159-195.

[8] Saito, S. and K. Horimoto. *Co-expressed gene assessment based on the path consistency algorithm: Operon detention in Escherichia coli.* : Proceedings of the 2009 IEEE international Conference on Systems, Man, and Cybernetics. 2009, p. 4280-4286.

[9] Wang, Y., et al., *Inferring gene regulatory networks from multiple microarray datasets.* Bioinformatics, 2006. **22**(19): p. 2413-2420.

[10] Y. Wang, T. Joshi, D. Xu, X.S. Zhang, L. Chen. Supervised inference of gene regulatory networks by linear programming. Lecture Notes in Bioinformatics, 2006, 4115, 551--561.

[11] Y. Wang, X. S. Zhang and L. Chen. A network biology study on circadian rhythm by integrating various omics data. OMICS: A Journal of Integrative Biology, 2009, 13(4).

[12] Y. Wang, X.-S. Zhang, and Y. Xia. Predicting eukaryotic transcriptional cooperativity by Bayesian network integration of genome-wide data. Nucleic Acids Research, 2009, 37(18):5943--5958.