33rd Annual International Conference of the IEEE EMBS
Boston, Massachusetts USA, August 30 - September 3, 2011

Discovery of Lung Cancer Pathways using Reverse Phase Protein Microarray and Prior-Knowledge based Bayesian Networks

Dong-Chul Kim, Chin-Rang Yang, Xiaoyu Wang, Baoju Zhang, Xiaorong Wu, and Jean Gao

Abstract— **The goal of this paper is to infer the signaling pathway related to lung cancer using Reverse Phase Protein Microarray (RPPM), which provides information on post-translational phosphorylation events. The computational inferring of pathways is obtained by performing Bayesian network in combination with prior knowledge from Protein-Protein Interaction (PPI). A clustering based Linear Programming Relaxation is developed for the searching of optimal networks. The PPI prior knowledge is incorporated into a new scoring function definition based on minimum description length (MDL). In the experiment, we first evaluate the algorithm performance with synthetic networks and associated data. Then we show our signaling network inference for lung cancer using RPPM data. Through the study, we expect to derive new signalling pathways and insight on protein regulatory relationships, which are yet to be known for lung cancer study.**

I. INTRODUCTION

Over the past decade, as high-throughput experimental methods such as gene microarray have been developed and improved, a large amount of biological data like gene expression and Protein-Protein Interaction (PPI) data have been accumulated. It has allowed many researchers in system biology to focus on gene regulatory network inference with plentiful data of transcription factors as well as gene expression [1]. Approaches for signaling pathway inference have been proposed mainly by analyzing PPI data. In this paper, an emerging protein microarray technology, called the Reverse Phase Protein Microarray (RPPM), in conjunction with the quantum dots nano-technology, is used to explore the systemic process of lung cancer signaling pathway. Regarding to the use of RPPM, while conventional data types such as PPI and gene microarray imply only indirect relationships of proteins in signaling pathways, RPPM can provide more immediate information to measure and profile the signaling pathways, providing the data on post-translational phosphorylation events not obtainable by the analysis of gene microarray and PPI.

To infer the signaling pathway with RPPM data, we perform the learning structure of Bayesian networks which have been effectively used to discover the biological networks. In learning Bayesian networks, there are typically two different approaches, conditional independence test based methods and so-called scoring-searching based methods that are time-consuming as a combinatorial optimization problem. We focus on scoring-searching method that can find optimal structure rather than approximate result. To this end, we employ Linear Programming Relaxation (LPR) method, Branch and Bound searching method [2], and Mutual Information (MI) [3]. The scoring function is solved as objective function in linear optimization. LPR and Branch and Bound are used to find optimal structure, and MI and Z-score are also used to select more likely edges as a preprocessing.

Since there are essential limitations of biological data such as the limited number of samples and noise, we integrate RPPM with PPI data as a prior knowledge so as to assure more reliable result in estimations. In this paper, we count the common edges between estimated networks and prior knowledge (PPI) and then reflect it to the score. By doing so, the scoring function is enforced to give a higher score to the estimated network which has more common edges.

Dong-Chul Kim and Jean Gao are with the Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX76019, USA {dkim, gao}@uta.edu

Chin-Rang Yang and Xiaoyu Wang are with Simmons Comprehensive Cancer Center, The University of Texas Southwestern Medical Center, Dallas, TX75390, USA {xiaoyu.wang, chinrang.yang}@utsouthwestern.edu

Baoju Zhang and Xiaorong Wu are with College of Physics and Electronics Information, Tianjin Normal University, Tianjin, China. wdxyzbj@163.com, wu.xiaoyong@sohu.com

II. METHOD

A. Bayesian Networks

We define a set of n nodes as random variables. Each node can have parent nodes, and an edge is oriented from a parent to a child as Bayesian network is Directed Acyclic Graph (DAG). Scoring function which measures the degree of fitness between estimated network and given data, and the goal of learning Bayesian network is to find the optimal network which has maximum score. The score function MDL [4] is defined as follows:

$$\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk}\log(\frac{N_{ijk}}{N_{ij}}) - \frac{1}{2}C(G)\log(N) \qquad (1)$$

where $C(G)$ is network complexity and defined as $C(G) = \sum_{i=1}^{n}(r_i - 1)q_i$. r_i is the number of states for variable X_i and q_i is the number of possible configurations of a parent set of X_i. N_{ijk} is the number of instances in the data set D where the variable X_i takes the value X_{ik} and have the j_{th} $(j = 1, 2, , q_i)$ configuration of the parent set of X_i. N_{ij} is the total number of the j_{th} configuration of X_i. Scoring function is decomposable into each node like $\sum_{i=1}^{n} W_i(s_i)$ where s_i is a parent set of X_i and our goal is to find $S = \{s_1, ..., s_n\}$ maximizing $\sum_{i=1}^{n} W_i(s_i)$. However, graph G should be acyclic with given S. In other words, each s_i cannot be selected independently. This is the most critical problem in learning Bayesian network.

978-1-4244-4122-8/11/$26.00 ©2011 IEEE

B. Preprocessing

Since the number of possible parent node set is 2^{n-1}, exponential searching space is another difficult problem. For this reason, we prune away as many parent sets as possible using Context Likelihood of Relatedness (CLR) [3] and the simple constraint for subset of parent [5]. First, MI is defined as

$$MI(X_i, X_j) = \sum_{X_i, X_j} p(X_i, X_j) \log(\frac{p(X_i, X_j)}{p(X_i)p(X_j)}), \quad (2)$$

and we build MI matrix in which each element MI_{ij} indicates MI value between X_i and X_j. Now we assumed that two nodes, X_i and X_j, are independent if MI_{ij} is relatively lower than other edges. So, Z-score is used to measure relative MI in the column and row of given MI_{ij}.

$$z_i(x_i, x_j) = \max\left(0, \frac{M_{ij} - \frac{\sum_{j'} MI_{ij'}}{N}}{\sigma_i}\right), \quad (3)$$

$$z_j(x_i, x_j) = \max\left(0, \frac{M_{ij} - \frac{\sum_{i'} MI_{i'j}}{N}}{\sigma_j}\right). \quad (4)$$

After setting matrix Z with $z(x_i, x_j) = z_i(x_i, x_j) + z_j(x_i, x_j)$, we can select only the edges whose Z_{ij} is higher than heuristic threshold. Hence, we exclude unnecessary edges in practice so that the number of possible s_i can be reduced effectively. Also, we can remove parent set s'_i if $W_i(s_i) \geq W_i(s'_i)$ for $s_i \subset s'_i$ [5]. In addition, the number of parent of each node can be limited by given a priori number (e.g. 4 in our experiment).

C. Cluster based Linear Programming Relaxation

Cluster based Linear Programming Relaxation we perform has been developed in [6] and applied to learning Bayesian networks in [2].

1) Objective Function: First, we define the objective function as

$$\max \eta \cdot W = \sum_{i=1}^{n} \sum_{s_i \in Pa(i)} \eta_i(s_i) W_i(s_i) \quad (5)$$

where $\eta \in \mathcal{P}$. $\mathcal{P}$ is a polytope of acyclic structures where a vertex corresponds to $\eta = [\eta_1 ... \eta_n]$ and η_i is an indicator (binary) vector of parent selection for node i. Dimension of η_i is $|s_i|$ (number of parent set of node i) and $\eta_i(s_i) = 1$ indicates that s_i is chosen as the parent set of node i.

2) Constraint: Constraint to be relaxed is defined as

$$\sum_{i \in C} \sum_{s_i \in Pa(i)} \eta_i(s_i) I_C(s_i) \geq 1 \quad (6)$$

where C is a cluster (a set of nodes) and $I_c(s_i)$ is an indicator function. If a cluster C includes any node of selected s_i, $I_c(s_i) = 0$. Otherwise, $I_c(s_i) = 1$. This constraint is from the fact that any subset of nodes in acyclic graph has at least one node whose parent is outside of the acyclic graph. So if selected S (parent sets) satisfies the constraint for every possible clusters (all subset of nodes), $\eta(S)$ is a vertex of polytope $\mathcal{P}$. With this constraint, dual problem can be defined as

$$\min \sum_{i=1}^{n} \max_{s_i \in Pa(i)} [W_i(s_i) + \sum_{C: i \in C} \lambda_C I_C(s_i)] - \sum_{C} \lambda_C \quad (7)$$
$$s.t. \lambda_C \geq 0, \forall C \subseteq V$$

where V is all subsets of nodes and λ_C is a dual variable for each cluster (each constraint). Since the number of λ_C is exponential, we initially set all λ_C to zero and $\mathcal{C}$ to $\emptyset$ ($C \in \mathcal{C}$), and then we iteratively add a single cluster into $\mathcal{C}$ and optimize λ_C. In every iteration, the relaxation for a single constraint is performed by adding a cluster and all dual variables (λ_C) is updated (optimized). Until dual value is equal to primal value, cluster is added in $\mathcal{C}$ in each iteration.

3) Update Dual Variables: In order to minimize dual problem, we keep updating all λ in each iteration of adding cluster. The role of λ is to enforce the constraint in the selection of parent set. More precisely, once we increase λ, it is enforced that s_i lied outside the corresponding cluster is selected for $I_c(s_i) = 1$. Reversely, too small λ causes that s_i is selected without considering cluster ($I_c(s_i) = 0$). The part of dual object is given by

$$J_C(\lambda_C) = \sum_{i \in C} \max_{s_i \in Pa(i)} [W_{C;i}(s_i) + \lambda_C I_C(s_i)] - \lambda_C \quad (8)$$

where $W_{C;i}(s_i) = W_i(s_i) + \sum_{C': i \in C'} \lambda_{C'} I_{C'}(s_i)$. To maximize $J_C(\lambda_C)$, we find $i \in C$ that minimizing $\delta_i = W^0_{C;i} - W^1_{C;i}$. Hence, λ_C is $\max\{(\delta_{i_1} + \delta_{i_2})/2, 0\}$ where $\delta_{i_1} < \delta_{i_2} < \ldots < \delta_{i_{|C|}}$. $W^0_{C;i}$ and $W^1_{C;i}$ are defined as

$$W^1_{C;i} = \max_{s_i \in Pa(i): I_C(s_i) = 1} W_{C;i}(s_i), \quad (9)$$

$$W^0_{C;i} = \max_{s_i \in Pa(i): I_C(s_i) = 0} W_{C;i}(s_i). \quad (10)$$

Alternative method is subgradient steps given as

$$\lambda_C \leftarrow \lambda_C + \epsilon \text{ if } \sum_{i \in C} I_C(\hat{s_i}) = 0, \quad (11)$$

$$\lambda_C \leftarrow \max\{\lambda_C - \epsilon, 0\} \text{ if } \sum_{i \in C} I_C(\hat{s_i}) > 1. \quad (12)$$

where s_i is maximizing $W_{C;i}(s_i) + \lambda_C I_C(s_i)$. In updating λ (λ_C vector), step size ϵ is set to $\frac{1}{|\lambda|} \sum_{i=1}^{|\lambda|} |\lambda^{old}_{C_i} - \lambda^{new}_{C_i}|$. $|\lambda|$ is size of λ. The step size decreases to zero and λ can be converged.

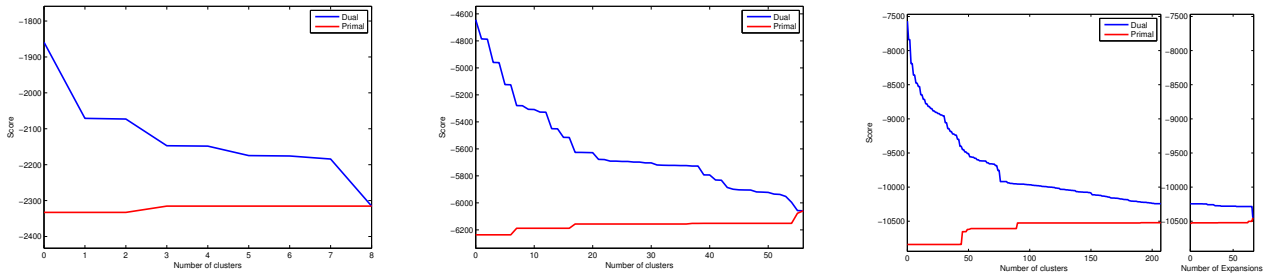
4) Decoding: The goal of decoding is to order the nodes (variables) maximizing primal value. To this end, given the clusters and dual variables in iteration, we can calculate dual score using simple dual form.

$$W_i(s_i; \lambda) = W_i(s_i) + \sum_{C: i \in C} \lambda_C I_C(s_i). \quad (13)$$

We set $P_1 = 0$, and calculate i_t for $t = 1 \ldots n$.

$$i_t = \arg\min_{i \in P_t} R_i, P_{t+1} = P_t \cup \{i_t\} \quad (14)$$

$$R_i = \max_{s_i \in Pa(i)} W_i(s_i; \lambda) - \max_{s_i \in Pa(i), s_i \subseteq P_t} W_i(s_i; \lambda) \quad (15)$$



(a) ASIA (b) CAR2 (c) ALARM

Fig. 1. Linear programming relaxation and Branch and Bound for synthetic network and data

Basically s_i in dual is the best selection of parent set to maximize score and satisfy all constraints for added all clusters at that time. R_i measure the difference of score (so-called regret) between the dual score without additional condition and the score with the exclusion of nodes that are ordered already and added into P.

5) Add Cluster: The goal of this step is to find more likely cycles in the dual. δ_{ij} for each directed edge E_{ji} is calculated as follow:

$$\delta_{ji} = \max_{s_i \in Pa(i): j \in s_i} [W_i(s_i) + \sum_{C:i \in C} \lambda_C I_C(s_i)] - \max_{s_i \in Pa(i): j \notin s_i} [W_i(s_i) + \sum_{C:i \in C} \lambda_C I_C(s_i)] \quad (16)$$

where δ_{ji} indicates whether or not E_{ji} is supported for current dual score. So, we add the cycle that maximize the minimum value of δ_{ji} along the cycle. To this end, first initialize $\Delta_{ji} = \delta_{ji}$ and $p_{ji} = i$. Δ_{ji} is the minimum value of δ along the path from j to i and p_{ji} of pointer matrix is the first node along the path from j to i. Secondly, for $k = 1...n, i = 1...n, j = 1...n$, if $\min\{\Delta_{i\to k}, \Delta_{k\to j}\} > \Delta_{i\to j}$ then $\Delta_{i\to j} = \min\{\Delta_{i\to k}, \Delta_{k\to j}\}, p_{i\to j} = p_{i\to k}$ After initialization, the minimum δ_{kl} (Δ_{kl}) of every possible pairs can be retrieved by tracing the pointers from k to l. Finally, the cycle that has the largest Δ in all cycles is added as new cluster.

D. Branch and Bound

Since dual and primal value may not be converged, we use Branch and Bound to find the optimal(maximum) decoded value. First, the parent set of given node i is divided into two groups as branches. A group are overlapped with a given cluster C that is associated with the node i, and another group are not overlapped with C. Then, we update all λ_C for two branches separately and calculate the dual and primal value with updated λ. Once dual is equal to primal, it is the optimal value. For next expansion, we choose the branch which has higher dual value.

E. Prior Knowledge

For more reliable network inference, we use PPI data as a prior knowledge to estimate an edge of parent set. Concretely, once the edges between node i and selected parent set include more PPI, the scoring function gives higher score to the parent set. As MDL is Maximum Likelihood based scoring function, we manipulate the number of instance set of variable when parent set is given. If $\{i, v \in Pa(i)\} \in PPI$,

$$Score(G:D) = \sum_i^n \sum_j^{q_i} IS_{new}^T \cdot \log(IS_{new} \cdot \frac{1}{\sum_k^{r_i} IS_{new_k}}) \quad (17)$$

where instances set $IS = [N_{ij1}, \ldots, N_{ijr_i}]^T$, $IS_{new} = IS_{old} - \{IS_{old} - \min(IS_{old})\} \cdot \alpha \cdot \beta$, and α is the normalized confidence level of prior knowledge of the edge. $\alpha = \frac{1}{|Pa(i)|} \sum_{j \in Pa(i)} R_{ji}$ where R is confidence level matrix and $|Pa(i)|$ is the size of parent set of node i. β is the parameter that indicates how much reflect the prior knowledge in the scoring function.

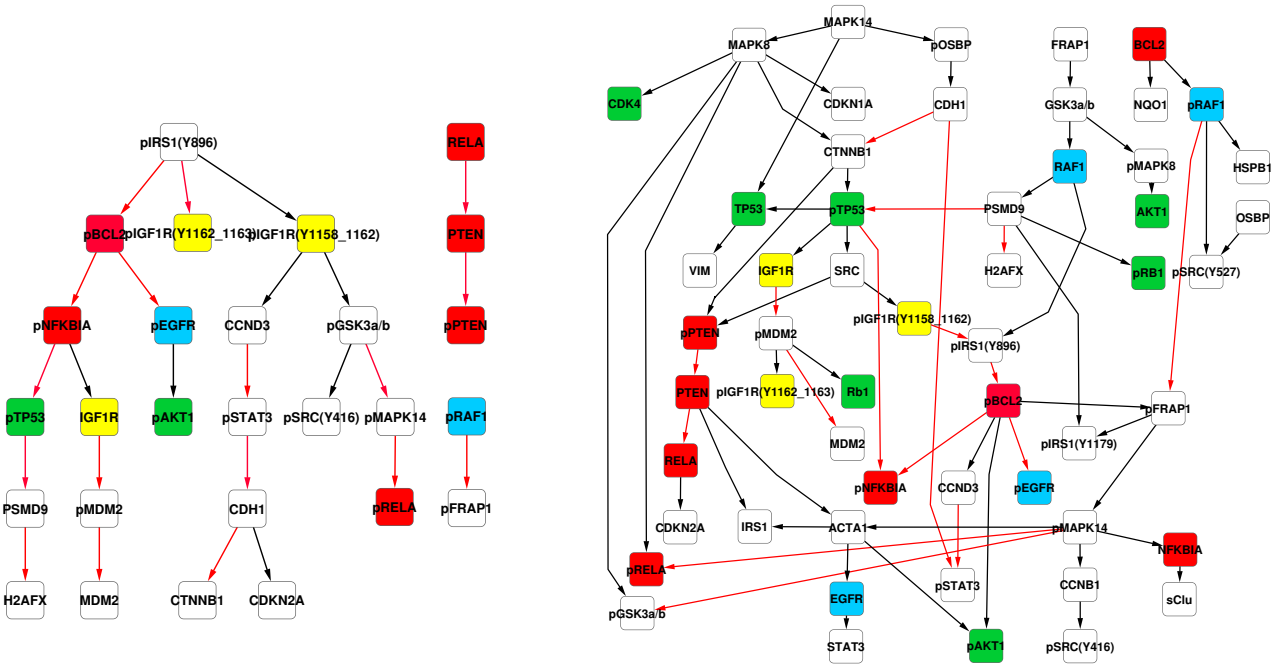
III. Experiments

A. Synthetic Data

Before we apply the method to RPPM data, the method is tested first in three well known network structures, ASIA, CAR2, and ALARM. These networks are frequently used to evaluate the performance of learning Bayesian network method in many literatures. They consist of 8, 18, and 37 nodes and 8, 20, and 46 edges respectively. The dataset of each network can be created by TETRAD4 (http://www.phil.cmu.edu/projects/tetrad/), and each data set has 1,000 samples. In Fig. 1 as a result, the optimal structure of ASIA can be found with only 8 clusters without Branch and Bound searching step in less than one second. For CAR2, it took around 1 minute to find maximum primal value with 56 clusters. In ALARM network, the dual and primal is converged in about 30 minutes with 207 clusters and 74 branch expansions.

B. RPPM Data

We apply the proposed method to lung cancer RPPM data which has 55 antibodies and 75 different lung cancer patient's samples with associated PPI data. In the experiment, each antibody is a random variable (node) in a network, and the expression level of each antibody is discretized into 3 states by K-means clustering as an unsupervised discretization. For PPI information as prior knowledge, STRING PPI database (http://string-db.org) is adopted as it provides the



(a) Inferred network without prior knowledge

(b) Inferred network with prior knowledge

Fig. 2. Inferred networks for lung cancer RPPM data and PPI data (hierarchic layout in Cytoscape)

experimental based confidence level of each interaction as well as the list of protein interactions. This confidence level that is normalized into 0 to 1 is based on only experimental evidence excluding the evidence in curated database. Hence, this confidence level value for each edge of two proteins is α in scoring function. Figure 2 shows the inferred network as a result. The colored label of nodes indicates the lung cancer related proteins in annotated pathway in KEGG database (http://www.genome.jp/kegg/). Red nodes are the proteins of small cell lung cancer pathway, blue means the proteins in non-small cell lung cancer, and green refers the proteins in both pathways. These lung cancer related proteins tend to be connected to each others in the result. The connection between an protein and its phosphorylation is caused by their high dependency rather than actual interaction. An edge may not indicate only the direct but indirect interaction along the pathway. Figure 2-(a) is the estimated network without prior knowledge. Only 27 nodes are appeared with 24 edges. Figure 2-(b) is the inferred network with prior knowledge. β is set to 0.1 and 17 red edges indicate the common edges to Fig. 2-(a). 49 additional new edges are estimated after the integration of 80 PPI and RPPM data. We note IGF-1R (Insulin growth factor type 1 receptor) observing that all three IGF-1R antibodies (yellow nodes) place near lung cancer related proteins in both networks. Recently IGF-1R has attracted attention in cancer therapy research considering that higher levels of IGF can increase the risk of lung cancer. It means that IGF-1R inhibitor may be used potentially as clinical therapy. Through our inferred network, we could also confirm the possibility of potential role of IGF-1R for lung cancer in the result.

IV. CONCLUSIONS AND FUTURE WORKS

In order to discover unknown lung cancer pathways, we perform learning Bayesian network with RPPM data and propose an integration method with a modified scoring function so that we can predict more reliable networks with PPI data as prior knowledge. For the future work, since not all antibodies may be involved in the actual pathway, we could select the partial paths (e.g. IGF-1R) from the result and validate them by biological experiments.

REFERENCES

[1] Riet De Smet and Kathleen Marchal. Advantages and limitations of current network inference methods. *Nature Reviews Microbiology*, 8(10):717–729, 8 2010.

[2] Tommi Jaakkola, David Sontag, Amir Globerson, and Marina Meila. Learning bayesian network structure using lp relaxations. volume 9, pages 358–365. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AI-STATS), 2010.

[3] Aviv Madar, Alex Greenfield, Eric Vanden-Eijnden, and Richard Bonneau. Dream3: Network inference using dynamic context likelihood of relatedness and the inferelator. *PLoS ONE*, 5(3):e9803, 03 2010.

[4] Wai Lam and Fahiem Bacchus. Learning bayesian belief networks: An approach based on the mdl principle. *Computational Intelligence*, 10:269–293, 1994.

[5] Cassio P. de Campos, Zhi Zeng, and Qiang Ji. Structure learning of bayesian networks using constraints. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 113–120, New York, NY, USA, 2009. ACM.

[6] David Sontag, Amir Globerson, and Tommi Jaakkola. Clusters and coarse partitions in lp relaxations. In *In Advances in Neural Information Processing Systems 22*. MIT Press, 2009.