Fiftieth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
October 1 - 5, 2012

A Novel Application of Mixing Coefficients for Reverse-Engineering Gene Interaction Networks

Nitin Singh[1], M. Eren Ahsen[1], Shiva Mankala[1], M. Vidyasagar[1], and Michael A. White[2]

[1]Bioengineering Department, University of Texas at Dallas, Richardson, TX 75080
[2]Cell Biology Department, UT Southwestern Medical Center, Dallas 75390

***Abstract*— In this paper, we present a new application of the so-called phi-mixing coefficient between two random variables. Using the phi-mixing coefficient, as well as an analog of the well-known data processing inequality from information theory, we present a new algorithm for reverse-engineering gene interaction networks (GINs) from expression data, by viewing the expression levels of various genes as coupled random variables. Unlike existing methods, the GINs constructed using the algorithm presented here have edges that are both directed and weighted. Thus it is possible to infer both the direction as well as the strength of the interaction between genes. Several GINs have been constructed for various data sets in lung and ovarian cancer. One of the lung cancer networks is validated by comparing its predictions against the output of ChIP-seq data.**

I. Introduction

Recent advances in experimentation, coupled with dramatic reductions in cost, now permit the biological research community to generate data at an unprecedented pace. One of the most common types of data consists of so-called 'gene expression data', in which a very large number of genes are excited by a mixture of DNA from cancerous and normal tissues. See [1] for a tutorial introduction to this experimental technique, which is also referred to as microarray analysis. The outcome of such an experiment is a set of 'expression levels', one for each gene, that roughly corresponds to the activity level of that particular gene in the cancerous tissue. If the gene is more active in the cancerous tissue than in the normal tissue, then it is said to be 'over-expressed', whereas if the situation is reversed, the gene is said to be 'under-expressed'. The expression study is repeated for several cancerous tissues, all coming from the same type of cancer, so that the scientist is able to generate a pattern of expression behavior for each gene across multiple tumor samples. This multiplicity of measurements compensates, to some extent, for the noisy nature of the measurements that is inherent to microarray analysis. At the end of the study, the data at hand can be viewed as an array of real numbers $\{x_{ij}, i = 1, \ldots, n, j = 1, \ldots, m\}$, where n is the number of genes under study and m is the number of tumor tissues that are analyzed. If the number of genes is close to the total number of genes in the human genome (roughly 22,000), then the study is said to be 'genome-wide'. The number of samples m ranges between several dozen to a few hundred.

There are many possible analyses that can be performed on such genome-wide expression data. One analysis that is potentially the most useful is to infer the manner and extent to which the various genes in the study interact with each other within a cell. Often these interactions are represented in the form of a directed graph, where the nodes represent the genes in the study, and the edges correspond to direct influences of one gene (the source of the edge) on another (the sink of the edge). Ideally, it would also be desirable to assign weights in the form of real numbers to each edge, so that the analysis sheds light not only on the direction of influence but also its strength. Such a network is often referred to in the literature as a 'gene regulatory network' (GRN). From a biologist's standpoint, the phrase 'regulation' encompasses not only the conclusion that gene A influences the behavior of gene B, but also some physical and/or chemical explanation of the *manner* of the regulation. However, methods based on statistical analysis, such as those surveyed in the literature review below, and also the present paper, do not lead to such 'mechanistic' or 'cause and effect' explanations, only an observation that interactions seem to be present. For this reason, we prefer to use the expression 'gene interaction network' (GIN) to denote the networks generated here.

The most common approach to reverse-engineering GINs is to view the expression level of each gene as a random variable, and the measurements of the gene expression levels of all genes in a single tissue as independent samples of the collection of random variables. As before, let n represent the number of genes and m the number of samples, and let $X_1, \ldots, X_n$ denote the random variables corresponding to the expression values of genes 1 through n. Then for a fixed index j, the vector $(x_{1j}, \ldots, x_{nj})$ (which is the outcome of a single genome-wide expression level measurement) is viewed as a realization of the joint random variable $(X_1, \ldots, X_n)$. The assumption is that if j and k are distinct indices, then the realizations $(x_{1j}, \ldots, x_{nj})$ and $(x_{1k}, \ldots, x_{nk})$ are statistically independent. Notice however that it is *not* assumed that the random variables X_i are independent of each other. Indeed, the objective of the exercise is to infer their interdependence from the

available data.

As number of samples (in the hundreds) is two orders of magnitude smaller than the number of genes (in the tens of thousands), it is not possible to infer the joint probability distribution of all random variables. Therefore we simply ask whether, for two distinct indices i and j, the corresponding random variables X_i and X_j are independent. Thus if X_i and X_j are independent random variables, then genes i and j do not interact at all, and there does not exist any path between the associated nodes i and j in the GIN. However, this is far too coarse a representation. At the next level of detail, one can choose three indices i, j, k and ask whether X_i and X_j are *conditionally independent* given X_k, henceforth denoted as $(X_i \perp X_j)|X_k$. If the answer is 'yes', then this would mean that in the associated GIN, the removal of node k and associated edges would disconnect nodes i and j. Therefore all paths from node i to node j and vice versa must pass through node k. Or to put it another way, if X_i and X_j are *conditionally independent* given X_k, then gene i does indeed interact with gene j, but in an indirect fashion, via gene k. It is therefore meaningful to ask: Given a set of whole-genome expression data, what is a *minimal interaction network* that is consistent with the data, in terms of faithfully reproducing the all the conditional independence properties implied by the data? In the present paper, we present an algorithm that answers this question.

To date we have used our method to reverse-engineer several GINs for lung cancer and ovarian cancer. In order to validate the reverse-engineered lung cancer GIN, we have used so-called ChIP-seq data from our collaborators to determine potential target genes around three specific genes, namely ASCL1, PPARG and NKX2-1. These three genes are well-known 'transcription factors', that is, genes that are expected to regulate many other genes. See [2] for a tutorial introduction to the ChIP-seq technology. In a nutshell, the output of a ChIP-seq experiment and subsequent analysis results in a fairly large number of genes that *could be* directly downstream of the transcription factor in the 'true but unknown' GIN. Our collaborators could identify 226 potential downstream target genes of ASCL1, 221 potential downstream target genes of PPARG, and 684 potential downstream target genes of NKX2-1. In the case of ASCL1 our results are truly spectacular, with the P-value (likelihood of getting the match purely through chance) being below machine zero. The P-values of obtaining our predictions purely by chance are about 0.0688 for downstream neighbors of PPARG and about 0.0465 for all neighbors of NKX2-1. The P-value for all neighbors of PPARG is about 0.0858 while the P-value for downstream neighbors of NKX2-1 is quite unimpressive at about 0.2057. Thus, except for the downstream neighbors of NKX2-1, the remainder are enriched at or near the 0.05 level of P-value, which is widely used in biology. As a further 'sanity check', we tested the neighborhood of ASCL1 in the ovarian cancer GIN; it was not in the least enriched for ChIP-seq genes identified for lung cancer. This is as it should be, and lends further credence to our reverse-engineered GIN for lung cancer.

II. Background on Information Theory and the Phi-Mixing Coefficient

A. Mutual Information

Suppose X, Y are random variables assuming values in finite sets $\mathbb{A}, \mathbb{B}$ respectively. Let $\boldsymbol{\theta}$ denote their joint distribution, so that

$$\theta_{ij} = \Pr\{X = i \& Y = j\}.$$

Let $\boldsymbol{\mu}, \boldsymbol{\nu}$ denote the marginal distributions of X and Y respectively. Thus

$$\mu_i = \Pr\{X = i\}, \nu_j = \Pr\{Y = j\}.$$

It is of course well-known that

$$\mu_i = \sum_j \theta_{ij}, \nu_j = \sum_i \theta_{ij}$$

Let

$$H(\boldsymbol{\theta}) = -\sum_{i,j} \theta_{ij} \log \theta_{ij}$$

denote the **Shannon entropy** of the probability distribution of the joint distribution $H(\boldsymbol{\theta})$. We shall also write $H(X, Y)$ in place of $H(\boldsymbol{\theta})$. In other words, we do not distinguish between the entropy of a probability distribution, and the entropy of a random variable having that probability distribution. With these definitions, the quantity

$$\begin{aligned} I(X, Y) &= H(X) + H(Y) - H(X, Y) \\ &= H(\boldsymbol{\mu}) + H(\boldsymbol{\nu}) - H(\boldsymbol{\theta}) \end{aligned}$$

is called the **mutual information** between X and Y. The following facts are well-known in the information theory literature; see for example [3]:

- The mutual information is symmetric; thus $I(X, Y) = I(Y, X)$.
- $0 \leq H(X, Y) \leq \min\{H(X), H(Y)\}$.
- X and Y are independent random variables if and only if $I(X, Y) = 0$.

Another very important fact about mutual information is the so-called 'data processing inequality'; see [3, p. 34]. Suppose X, Y, Z are random variables and that $(X \perp Z)|Y$. Then

$$I(X, Z) \leq \min\{I(X, Y), I(Y, Z)\}.$$

As described in the literature survey, the notion of mutual information and the data processing inequality can be used to reverse-engineer *undirected, unweighted* GINs from expression data. However, our objective is to reverse-engineer *directed and if possible weighted* GINS. So we look for alternate measures of dependence between two random variables that are directional. This leads us to the ϕ-mixing coefficient.

B. The ϕ-Mixing Coefficient: Definition

The ϕ-mixing coefficient was introduced in [4] as a measure of the asymptotic long-term independence of a stationary stochastic process, and was used to prove laws of large numbers for non-i.i.d. processes. See [5, (2.5.3)] for the general definition. This definition can be readily adapted to define a quantitative measure of the dependence between two random variables; see [6]. From the standpoint of reverse-engineering GINs from expression data, the most appealing feature of the ϕ-mixing coefficient is its directionality. Unlike mutual information or Pearson correlation, the ϕ-mixing coefficient distinguishes between the dependence of X on Y and that of Y on X.

If X and Y are random variables assuming values in possibly distinct finite[1] sets $\mathbb{A} = \{1, \ldots, n\}$ and $\mathbb{B} = \{1, \ldots, m\}$ respectively, the ϕ-mixing coefficient $\phi(X|Y)$ is defined as

$$\phi(X|Y) := \max_{S \subseteq \mathbb{A}, T \subseteq \mathbb{B}} |\Pr\{X \in S | Y \in T\} - \Pr\{X \in S\}|. \tag{1}$$

Thus $\phi(X|Y)$ is the maximum difference between the conditional and unconditional probabilities of an event involving only X, conditioned over an event involving only Y. Specifically, the ϕ-mixing coefficient has the following properties:

1) $\phi(X|Y) \in [0, 1]$.
2) In general, $\phi(X|Y) \neq \phi(Y|X)$. Thus the ϕ-mixing coefficient gives directional information.
3) X and Y are independent random variables if and only if $\phi(X|Y) = \phi(Y|X) = 0$.
4) The ϕ-mixing coefficient is invariant under any one-to-one transformation of the data. Thus if $f : \mathbb{A} \to \mathbb{C}, g : \mathbb{B} \to \mathbb{D}$ are one-to-one and onto maps, then
$$\phi(X|Y) = \phi(f(X)|g(Y)).$$

It is evident that $\phi(X|Y)$ measures the degree of inter-dependence between X and Y. Thus, unlike with mutual information, if $\phi(X|Y) < \phi(W|Z)$, then it can indeed be said that X depends less on Y than W does on Z.

C. The ϕ-Mixing Coefficient: Computation

The material presented above is all standard. Now we review two new results from [7] that are crucial for the algorithm being proposed here.

While (1) is suitable for *defining* the quantity $\phi(X|Y)$, it cannot be directly used to *compute it*. This is because (1) requires us to take the maximum over all subsets of $\mathbb{A}$ and $\mathbb{B}$, and would thus require $2^{|\mathbb{A}|+|\mathbb{B}|}$ computations. However, in the special case where the marginal distribution of Y is the uniform distribution, then it is quite easy to compute the associated coefficient $\phi(X|Y)$. Specifically, let $\Theta \in [0, 1]^{n \times m}$ denote the joint distribution of X and Y written out as an $n \times m$ matrix. Define $\Psi \in [0, 1]^{n \times m}$ as the outer product of the two marginal distributions $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$; thus

$$\psi_{ij} = \mu_i \nu_j, \ \forall i, j.$$

Then Ψ is a rank one matrix, and is the joint distribution that X and Y would have if they were independent. Define $\Lambda \in [-1, 1]^{n \times m}$ by

$$\lambda_{ij} = \theta_{ij} - \psi_{ij}, \ \forall i, j.$$

Thus Λ would be the zero matrix if X and Y were independent. Define

$$\|\Lambda\|_{i1} := \max_{j=1,\ldots,m} \sum_{i=1}^{n} |\lambda_{ij}|$$

to be the matrix norm of Λ induced by the ℓ_1 vector norm. With these definitions, the following facts are established in [7]:

Theorem 1: With the notation as above, it is the case that

$$\frac{0.5\|\Lambda\|_{i1}}{\max_j \nu_j} \leq \phi(X|Y) \leq \frac{0.5\|\Lambda\|_{i1}}{\min_j \nu_j}. \tag{2}$$

In particular, if $\boldsymbol{\nu}$ is the uniform distribution so that $\min_j \nu_j = \max_j \nu_j = 1/m$, then

$$\phi(X|Y) = 0.5m\|\Lambda\|_{i1}. \tag{3}$$

The most important property of the ϕ-mixing coefficient is given next. Again, the proof can be found in [7].

Theorem 2: Whenever $(X \perp Z)|Y$, the following inequality holds:

$$\phi(X|Z) \leq \min\{\phi(X|Y), \phi(Y|Z)\}. \tag{4}$$

Note that (4) is entirely analogous in appearance to ([3], pg 34). For this reason, we will refer to (4) as the data processing inequality for the ϕ-mixing coefficient. The observation that the ϕ-mixing coefficient satisfies an analog of the DPI is new, and a proof of (4) can be found in [7].

III. Literature Survey

The problem of inferring GINs from expression data is obviously not new, and several researchers have attempted to study this problem. Most existing methods can be grouped into one of two categories, namely: those based on mutual information, and those based on Bayesian networks. Papers such as [8], [9], [10], [11] are representatives of methods based on mutual information, while [12], [13], [14], [15] are representatives of methods based on Bayesian networks. Both classes of methods impose some biologically unrealistic conditions mainly to facilitate the statistical analysis. Specifically, methods based on Bayesian networks require the graph to be acyclic, while methods based on mutual information will result in graphs that are undirected. Neither

[1]The assumption that both random variables are finite-valued is made purely for convenience in exposition. In the general case, the sets S and T would have to belong to the σ-algebras generated by the random variables X and Y respectively, and the maximum would have to be replaced by the supremum.

assumption is justifiable on biological grounds. The Bayesian paradigm, with its information flow restricted to be in one direction, is useful for hierarchical decomposition of GINs into 'clusters' of genes where each cluster of genes controls lower-level clusters. This is a much coarser picture of a GIN than the ones obtained by using mutual information-based methods. For this reason, we do not discuss Bayesian-based methods hereafter, and confine ourselves to discussing methods based on mutual information. Please see [3] for a thorough treatment of all information-theoretic concepts.

Apparently the first paper to use the concept of mutual information to construct GINs is [8]. In that paper, the authors compute the mutual information between every pair of genes, and introduce an undirected edge between nodes i and j if and only if the mutual information $I(X_i, X_j)$ between the corresponding random variables X_i and X_j is positive.[2] Actually, they select a small threshold ϵ and introduce an undirected edge between nodes i and j if and only if $I(X_i, X_j) \geq \epsilon$. They refer to the resulting (undirected) graph as an 'influence network'. Indeed, in their framework, the presence of an (undirected) edge between two nodes i and j makes no distinction between gene i influencing gene j or vice versa. Also, no distinction is made between direct and indirect influence. As a consequence, the influence networks produced by the method in [8] are overly dense.

In [9], the authors develop a method referred to as ARACNE to distinguish between direct and indirect influence by making use of the data processing inequality [3, p. 34]. Specifically, for each triple i, j, k, they compute all the three mutual informations $I(X_i, X_j), I(X_i, X_k)$ and $I(X_j, X_k)$. Since the exact probability distributions are not known and only samples are available, they use Gaussian kernel approximations for the various joint distributions. Then they identify the smallest amongst the three numbers and discard the corresponding edge. Thus if

$$I(X_i, X_k) \leq \min\{I(X_i.X_j), I(X_j, X_k)\},$$

then they discard the edge between nodes i and k.

Thus the network produced by ARACNE tells us only that nodes i and j interact, but does not give further information as to the *directionality* of the interaction. It would be highly desirable to develop an algorithm that is able to identify the directionality of interaction between genes. Moreover, it is explicitly stated in [8] (and implicitly assumed in [9]) that mutual information can be used as a measure of the strength of interaction between two random variables. But this statement is only partially true. It is true that if $I(X, Y) < I(X, Z)$, then Z tells us more about X than Y does. So in this sense X depends more on Z than on Y. However, if $I(X, Y) < I(W, Z)$, it is *not correct* to conclude that X depends less on Y than W depends on Z. The algorithm presented here addresses both of these limitations.

Proceeding further, in all the mutual information-based approaches, the most computationally intensive step is the computation of the pairwise mutual informations. In [17], the authors take the given sample pairs $\{(x_{il}, x_{jl}), l = 1, \ldots, m\}$ for each pair of indices i, j, and then fit them with a two-dimensional Gaussian kernel. Then they apply a copula transformation so that the sample space is the unit square, and the marginal probability distribution of each random variable is the uniform distribution. In [18], the authors propose a window-based approach for computing the pairwise mutual informations. It is claimed in this paper that the proposed method results in roughly an order of magnitude reduction in computing effort. Finally, in a very recent paper [16], the authors bin the samples into just three bins irrespective of how many samples there are, and propose a highly efficient parallel architecture for computing the pairwise mutual informations. While the proposed architecture is very innovative, it appears to the present authors that quantizing the expression values into just three bins could result in misleading conclusions, because the gene expression level is essentially a *real-valued* random variable. In the method proposed here, we also discretize the samples by percentile binning. However, in our case the number of bins increases as the number of samples increases, thus giving a more realistic discretized representation of a real-valued random variable.

IV. Algorithm for Reverse-Engineering GINs

In the present subsection we present the algorithm, assuming that we know the actual coefficient $\phi(X_i|X_j)$ for each pair of indices i, j. However, in actual implementation, these values are estimated using (2), after percentile binning of a finite number of sample data points. Percentile binning is chosen as it ensures that the joint distribution of the discretized pairs (X_i, X_j) remains invariant under any monotonic transformations of the data. It is important to note here that the invariance property holds even if different monotone transformations are applied to different expression variables.

So we begin with estimated value of all $n(n-1)$ coefficients $\phi(X_i|X_j)$ for each pair of indices $i, j, i \neq j$. Then we proceed as follows: Start with a complete graph of n nodes, where there is a directed edge between every pair of distinct nodes ($n(n-1)$ edges). For each triplet i, j, k, check whether the DPI-like inequality

$$\phi(X_i|X_k) \leq \min\{\phi(X_i|X_j), \phi(X_j|X_k)\} \qquad (5)$$

holds. If so, discard the edge from node k to node i, but retain a 'phantom' edge for future comparison purposes.

[2] It is not mentioned in the paper how precisely they compute the mutual information from the samples. As shown in [7], it makes a difference whether they use the samples 'as given' and construct a stair-case like joint distribution, or use some smoothing as in [9], or bin the samples as in [16] or as we do here.

This step is referred to as 'pruning'. Note that the pruning operation can at best replace a direct path of length one (i.e. an edge) by an indirect path of length two. Hence the graph that results from the pruning operation is still strongly connected. Also, since any discarded edges are still retained for the purposes of future comparisons, it is clear that the order in which the triplets are processed does not affect the final answer. Note that the complexity of this operation is cubic in n.

At this stage, one can ask whether the graph resulting from the pruning operation has any significance. It is now shown, by invoking the Occam's razor principle (giving the simplest possible explanation), that the graph resulting from pruning is *a minimal graph* consistent with the data set. For this purpose, we define a partial ordering on the set of directed graphs with n nodes whereby $\mathcal{G}_1 \leq \mathcal{G}_2$ if $\mathcal{G}_1$ is a subgraph of $\mathcal{G}_2$, ignoring weights of the edges. For a given triplet i, j, k, it is obvious that $(X_i \perp X_k)|X_j$ if and only if every directed path from node i to node k passes through node j, and also every directed path from node k to node i passes through node j. Now, it follows from the DPI that if $(X_i \perp X_k)|X_j$, then (5) holds. Taking the contrapositive shows that if (5) is false, then $(X_i \not\perp X_k)|X_j$. Consequently, if (5) is false, then there must exist a path from node i to node k that does not pass through node j. Given the sequential nature of the pruning algorithm, when (5) is checked for a specific triplet (i, j, k), there already exist edges from node i to node j and from node j to node k; that is, there exists a path of length two from node i to node k. Now, if (5) is false, then there must exist another path from node i to node k that does not pass through node j. It is of course possible that this path consists of many edges. However, by the Occam's razor principle, the simplest explanation would be that there is a shorter path of length one, i.e. a directed edge from node k to node i.

What has been shown is that, under the Occam's razor principle, the graph that results from pruning is *minimal* in the following sense. First, it is consistent with the ϕ-mixing coefficients, and second, any other graph that is 'less than' this graph in the partial ordering defined above would not be consistent with the ϕ-mixing coefficients. Thus, if any edges are removed from the graph that results from applying the pruning step, then some other edges would have to be added in order for the graph to be consistent with the ϕ-mixing coefficients. Note that we are obliged to say *a* and not *the* minimal graph, because there might not be a unique minimal graph. Nevertheless, it is obvious that the application of the algorithm will result in a unique graph, irrespective of the order in which all the triplets (i, j, k) are examined.

Since we estimate interval in which value of ϕ-mixing coefficient lies, there are multiple ways of checking (5) while pruning. Here presented results compare upper bound of $\phi(X_i|X_k)$ with midpoint of intervals in which $\phi(X_i|X_j)$ and $\phi(X_j|X_k)$ lies.

Finally, a thresholding step is employed to eliminate weak interactions represented by low-weight edges, while still ensuring that the graph remains strongly connected.

TABLE I
NUMBER OF POTENTIAL TARGET GENES FOR VARIOUS TRANSCRIPTION FACTORS

Item	ASCL1	PPARG	NKX2-1
Total no. of genes	19,579	19,579	19,579
Total no. of ChIP genes	226	221	684
Prob. of being a ChIP gene	0.0115	0.0112	0.0348

V. RESULTS AND VALIDATION

Using the algorithm presented here, we have reverse-engineered several GINs for both lung cancer and ovarian cancer. It is virtually impossible to validate an entire GIN since there is no known and confirmed 'absolute truth' against which predictions can be confirmed. After considerable thought, we chose to use evidence from so-called ChIP-seq tests for a few transcription factors. ChIP-seq experiment produces a set of downstream target genes whose transcription is controlled by the given transcription factor with high probability. We obtained ChIP-seq data for three transcription factors in lung cancer tissues, namely: ASCL1, PPARG, and NKX2-1. Table I shows the number of experimentally determined ChIP-seq genes for these three transcription factors.

The validation of the GIN consists of seeing whether the set of first-order neighbors in the GIN is 'enriched' for ChIP-seq genes. For this purpose, we employ a hypergeometric distribution model to calculate likelihood of observing such an enrichment by chance. Thus, if N denotes the total number of genes in the geome-wide study, C denotes the number of ChIP-seq genes, N_d denotes the number of downstream genes in the GIN, and C_d denotes the number among the N_d that are in the ChIP-seq list, then the likelihood of enrichment value is computed using the Matlab command as

$$L_d = 1 - \text{hygecdf}(C_d - 1, N - 1, C, N_d).$$

Table II shows the enrichment of ChIP genes amongst the first-order downstream neighbors, and amongst all neighbors, for these three transcription factors. In this table, an entry of zero for the likelihood means that the number is smaller than machine zero.

In biology, a likelihood (P-value) less than 0.05 is considered significant. For ASCL1 our results for truly spectacular, while for the other two genes the results are decent but not spectacular. It should be mentioned that for ASCL1, the ChIP-seq genes were determined using the peak-calling routine GREAT [19]. However, we are not aware of the manner in which the ChIP-seq genes were determined for PPARG and NKX2-1. The difference in performance of our algorithm can perhaps

TABLE II
ENRICHMENT OF NEIGHBORS FOR CHIP GENES

Gene Name	Downstream Neighbors	ChIP Genes	L_d
ASCL1	690	84	0
PPARG	84	3	0.0688
NKX2-1	114	6	0.2057
Gene Name	**Total Neighbors**	**ChIP Genes**	L_t
ASCL1	766	87	0
PPARG	208	5	0.0858
NKX2-1	244	14	0.0465

be attributed to the differences between the peak-calling routines employed.

VI. CONCLUSION

In this paper, we have presented a new algorithm for reverse-engineering gene interaction networks (GINs) from gene expression data that produces GINs that have both directed and weighted edges. To validate the algorithm, we constructed a GIN for lung cancer data set and compared the neighborhood of three genes in this network with experimental ChIP-seq data. Validation results show that these neighborhoods are reasonably enriched with ChIP-seq downstream target genes.

ACKNOWLEDGEMENTS

The authors thank Prof. John Minna and Mr. Alex Augustyn for the expression data for various lung cancer cell lines; Prof. Jane Johnson and Mr. Mark Borromeo for the ChIP-seq data on ASCL1; Prof. Ralf Kittler and Dr. Rahul Kollipara for the ChIP-seq data on PPARG and NKX2-1; and Drs. Keith Baggerly and Anna Unruh for identifying the responders and non-responders to platinum chemotherapy in ovarian cancer patients.

This work was supported by the National Science Foundation Award #1001643 to MV and by grants from Welch Foundation #I-1414 and National Cancer Institute #CA71443 to MAW.

REFERENCES

[1] M. M. Babu, "An introduction to microarray data analysis," in *Computational Genomics: Theory and Application*. Horizon Bioscience, 2004.

[2] E. T. Liu, S. Pott, and M. Huss, "Q&a: Chip-seq technologies and the study of gene regulation," *BMC Biology*, vol. 8:56, pp. http://www.biomedcentral.com/1741–7007/8/56, 2010.

[3] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Second Edition)*. Wiley Interscience, New York, 2006.

[4] I. A. Ibragimov, "Some lilmit theorems for stationary processes," *Theory of Probability and its Applications*, vol. 7, pp. 349–382, 1962.

[5] M. Vidyasagar, *Learning and Generalization: With Applications to Neural Networks and Control Systems*. Springer-Verlag, London, 2003.

[6] P. Doukhan, *Mixing: Properties and Examples*. Springer-Verlag, Heidelberg, 1994.

[7] M. E. Ahsen and M. Vidyasagar, "Mixing coefficients between discrete and real random variables: Computation and properties," *arxiv*, p. 1208:1720, 2012.

[8] A. J. Butte and I. S. Kohane, "Mutual information relevance networks: Functional genomic clustering using pairwise entropy measures," *Pacific Symposium on Biocomputing*, pp. 418–29, 2000.

[9] A. A. Margolin *et al.*, "Aracne: An algorithm for the reconstruction of gene regulatory networks in a cellular context," *BMC Bioinformatics*, vol. 7(Supplement 1):S7, 20 March 2008.

[10] K. Wang *et al.*, "Genome-wide identification of post-translational modulators of transcription factor activity in human b cells," *Nature Biotechnology*, vol. 27(9), pp. 829–839, September 2009.

[11] T. S. Gardner *et al.*, "Inferring genetic networks and identifying compound mode of action via expression profiling," *Science*, vol. 301, no. 5629, 2003.

[12] N. Friedman *et al.*, "Using bayesian networks to analyze expression data," *J. Comput. Biol.*, vol. 7(3-4), pp. 601–620, June-August 2000.

[13] Y. Barash and N. Friedman, "Context-specific bayesian clustering for gene expression data," *J. Comput. Biol.*, vol. 9(2), pp. 169–191, 2002.

[14] N. Friedman, "Inferring cellular networks using probabilistic graphical models," *Science*, vol. 303, pp. 799– 805, February 2004.

[15] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, MA, 2009.

[16] V. Belcastro *et al.*, "Reverse engineering and analysis of genome-wide gene regulatory networks from gene expression profiles using high-performance computing," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9(3), pp. 668–674, May/June 2012.

[17] W. Zhao, E. Serpedin, and E. R. Dougherty, "Inferring connectivity of genetic regulatory networks using information theoretic criteria," *IEEE/ACM Trans. Comput. Biol. and Bioinf.*, vol. 5(2), pp. 262–274, April-June 2008.

[18] P. Qiu, A. J. Gentles, and S. K. Plevritis, "Reducing the computational complexity of information theoretic approaches for reconstructing gene regulatory networks," *J. Comput. Biol.*, vol. 17(2), pp. 169–176, February 2010.

[19] C. Y. McLean *et al.*, "Great improves functional interpretation of cis-regulatory regions," *Nature Biotechnology*, vol. 28(5), pp. 495–501, 2010.