Proceedings of the 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE), Larnaca, Cyprus, 11-13 November 2012

sCoIn: A Scoring algorithm based on COmplex INteractions for reverse engineering regulatory networks

Vijender Chaitankar, Preetam Ghosh
Department of Computer Science
Virginia Commonwealth University
Richmond, VA, USA
chaitankarv@vcu.edu
pghosh@vcu.edu

Mohamed O. Elasri
Department of Biological Sciences
University of Southern Mississippi
Hattiesburg, MS, USA
mohamed.elasri@usm.edu

Kurt A. Gust, Edward J. Perkins
Environmental Laboratory
E.R.D.C, U.S. Army
Vicksburg, MS, USA.
kurt.a.gust@usace.army.mil
edward.j.perkins@usace.army.mil

Abstract— **Structural analysis over well studied transcriptional regulatory networks indicates that these complex networks are made up of small set of reoccurring patterns called motifs. While information theoretic approaches have been immensely popular, these approaches rely on inferring the regulatory networks by aggregating pair-wise interactions. In this paper, we propose novel structure based information theoretic approaches to infer transcriptional regulatory networks from the microarray expression data. The core idea is to go beyond pair-wise interactions and consider more complex structures as found in motifs. While this increases the network inference complexity over pair-wise interaction based approaches, it achieves much higher accuracy and yet is scalable to genome-level inference. Detailed performance analyses based on benchmark precision and recall metrics on the known *Escherichia coli's* transcriptional regulatory network indicates that the accuracy of the proposed algorithms is consistently higher in comparison to popular algorithms such as context likelihood of relatedness (CLR), relevance networks (RN) and GEneNetwork Inference with Ensemble of trees (GENIE3). In the proposed approaches the size of structures was limited to three node cases (any node and its two parents). Analysis on a smaller network showed that the performance of the algorithm improved when more complex structures were considered for inference, although such higher level structures may be computationally challenging to infer networks at the genome scale.**

Index Terms—**Information theory, complex interactions, regulatory networks, inference.**

I. INTRODUCTION

A transcriptional regulatory network (TRN) represents the regulation of genes by transcription factors [1-3]. It is due to this regulation mechanism, cells have the ability to adapt to their external environment [1]; hence inferring the transcriptional regulatory networks is of great importance as they elucidate the behavior of cells. While a number of TRN inference approaches exists [4], this paper focuses on the microarray expression data based inference approaches. In this particular approach, patterns are scanned in the microarray expression data [4] and are aggregated in to a final network.

Various classes of algorithms exist for TRN inference based on microarray expression data. Some of these popular classes are Bayesian networks [5], Dynamic Bayesian networks [6], Boolean networks [7], probabilistic Boolean networks [8], differential equations models [9] and information theoretic models [10-15]. Information theoretic models in particular have gained substantial attention due to their unique ability to exploit genome scale expression data. In this context, relevance networks (RN) [10-11], a general reverse engineering algorithm for inference of genetic network architectures (REVEAL) [12], algorithm for the reconstruction of accurate cellular networks (ARACNE) [13], and Context likelihood of relatedness (CLR) [14] are some of the popular information theoretic algorithms.

A fundamental motivation of our proposed approach comes from Babu et al. [4] work that structurally organized a TRN in the following categories: (i) A basic unit, which consists of a single regulatory mechanism; (ii) Motifs, which are small reoccurring patterns of regulatory interactions and (iii) Global structure, which is essentially set of all regulatory interactions. Other structural analysis on complex networks [1, 16] including TRNs suggests that motifs form their building blocks and entire motif structures work together to achieve particular regulatory effects. Hence our premise for this work is based on the hypothesis that TRN inference algorithms must also consider scanning the microarray data for structures. With the exception of REVEAL, all other information theoretic algorithms build TRN's by scanning for individual edge patterns. While REVEAL does consider scanning for structures, it does not limit it to smaller structures (like motifs), which increases its computational complexity and hence is not applicable towards inference of large networks. Also, with different ordering of input data, REVEAL may infer a different network, and hence is not robust [15].

In this paper, we propose novel structure based information theoretic approaches to infer TRN's. In order to assess the performance of our proposed approaches we have chosen three other popular state of the art approaches: (i) CLR (ii) GEneNetwork Inference with Ensemble of trees (GENIE3) [17] and (iii) RN. CLR is one of the most popular information theoretic algorithm that performs quite well at the genome-scale and is robust to both time-series and non time-series gene expression datasets. GENIE3 a recently proposed algorithm uses a fundamentally different data mining based approach to infer regulatory networks and provides very high inference

	Bin 1			Bin 2			…	Bin m		
	T_1	…	T_t	T_1T_2	…	$T_{t-1}T_t$		$T_1..T_m$	…	$T_{t-1+m}..T_m$
G_1	$I(G_1,T_1)$		$I(G_1,T_t)$	$I(G_1, T_1T_2)$		$I(G_1,T_{t-1}T_t)$		$I(G_1, [T_1..T_m])$		$I(G_1, [T_{t-1+m}..T_m])$
. .										
G_g	$I(G_g,T_1)$		$I(G_g,T_t)$	$I(G_g, T_1T_2)$		$I(G_g,T_{t-1}T_t)$		$I(G_g, [T_1..T_m])$		$I(G_g, [T_{t-1+m}..T_m])$
T_1	$I(T_1,T_1)$		$I(T_1,T_t)$	$I(T_1, T_1T_2)$		$I(T_1,T_{t-1}T_t)$		$I(T_1, [T_1..T_m])$		$I(T_1, [T_{t-1+m}..T_m])$
. .										
T_t	$I(T_t,T_1)$		$I(T_t,T_t)$	$I(T_t, T_1T_2)$		$I(T_t,T_{t-1}T_t)$		$I(T_t, [T_1..T_m])$		$I(T_t, [T_{t-1+m}..T_m])$

Figure 1. MI Computation Scheme

accuracy. GENIE3 actually is the top-ranked inference algorithm that won the Dream-4 international reverse engineering contest [18] where it consistently outperformed other approaches. Our performance analysis based on benchmark precision and recall metrics on the *Escherichia coli's* network and data obtained using the standardized genenetweaver tool [19] showed that the proposed methods infer networks with higher accuracy at the genome scale as compared to both CLR and GENIE3 algorithms.

This paper is organized as follows: Section II discusses the information theory based metrics used in the relevance network class of algorithms and outlines the proposed algorithms; Section III presents the results and detailed performance analysis of the proposed algorithms in comparison to CLR, RN and GENIE3; Section IV presents the conclusion and future directions based on this work.

II. METHODS

A. Data and Network Formulation

Genes and transcription factors are two kinds of nodes in a transcriptional regulatory network (TRN). Considering g number of genes and t number of transcription factors, the total number of nodes in a network is $N= t+g$. A graph $G(E,V)$ represents a TRN, where E is the set of edges and N is the set of vertices. Vertices here are genes and transcription factors whereas an edge represents regulation of a gene by a transcription factor. We further use the notations G_i, where $i = 1..g$ and T_j, where $j = 1..t$ to represent the set of all genes and transcription factors respectively.

B. Information theoretic metrics

Here we discuss some of the information theory metrics used in the reverse engineering algorithm proposed in this paper.

1) Entropy

Entropy (H) is the measure of average uncertainty in a random variable. Entropy of a random variable X with probability mass function $p(x)$ is defined [20] as:

$$H(X) = -\sum_{x\in X} p\,(x) * log[p(x)] \quad (1)$$

The entropy of a random variable is maximum when the states are equiprobable. It should be noted that entropy is a positive quantity and as the bias in the system increases, the entropy decreases.

The concept of entropy over a single random variable can be further extended to a pair of random variables to obtain the joint entropy. The joint entropy $[H(X,Y)]$ of a pair of discrete random variables (X,Y) with a joint distribution $p(x,y)$ is defined as:

$$H(X,Y) = -\sum_{x\in X,y\in Y} p\,(x,y) * log[p(x,y)] \quad (2)$$

Entropy computation as given in equation 2 can be extended to higher dimension variables:

$$H(X,Y,\cdots,Z) = -\sum_{x\in X,y\in Y,\cdots,z\in Z} p\,(x,y,\cdots,z) * log[p(x,y,\cdots z)] \quad (3)$$

2) Mutual Information

Mutual Information (MI) measures the amount of information that can be obtained about one random variable by observing another one. MI is defined [20] as:

$$I(X,Y) = -\sum_{x\in X,y\in Y} p\,(x,y) * log\frac{[p(x,y)]}{p(x)*p(y)} \quad (4)$$

MI can also be defined in terms of entropies as:

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \quad (5)$$

In other words, the shared or mutual information between two variables is obtained by subtracting the joint uncertainty from the sum of individual uncertainties. Given the expression data, the relationship between the genes is scored using the MI metric.

Bin 1

Combination	Score: $S^1_{1,j}$
T_1	0.5
T_2	1.00
T_3	0.25
T_4	0.5

Bin 2

Combination	Score: $S^2_{1,j}$
T_1,T_2	1.00
T_1,T_3	0.33
T_1,T_4	0.65
T_2,T_3	0.4
T_2,T_4	0.80
T_3,T_4	0.1

Regulators/ Bin	T_1	T_2	T_3	T_4
$BS^1_{1,j}$	0.222	0.444	0.111	0.222
$BS^2_{1,j}$	0.301	0.335	0.126	0.236
Final Score $F_{1,j}$	0.523	0.779	0.237	0.458

Figure 2. Example Scoring Scheme

The MI computation as given in equation 5 can be easily extended to higher dimensions:

$$I(X,[Y,\cdots,Z]) = H(X) + H([Y,\cdots,Z]) - H(X,Y,\cdots,Z) \quad (6)$$

While a number of continuous MI estimation approaches exist, we chose to implement the B-Spline approach as described in Daub et al. [21]. Unlike binning techniques, the B-Spline is not sensitive to the number of bins chosen. Gausian kernel density estimators have a better accuracy; however, their computational complexity is high. It has been observed that B-Splines produces results as accurate as Gaussian estimators but with lower computational complexity [22]. Hence we have chosen the B-Spline based density estimation approach for MI computations. Moreover, we have chosen the number of bins as five and the spline order to two in all the algorithm implementations. However, it should be noted that any other MI estimator can also be seamlessly used in our proposed algorithms; the contributions of this paper builds on the computed MI values irrespective of the underlying method used for the same.

3) Z-Scores

Z-score, also known as standard score, is a statistical measure which normalizes a variable based on its mean and variance. Z-score's for network inference was first implemented in the CLR algorithm. CLR applies an adaptive background correction step to eliminate false connections and indirect influences which involved the computation of Z-scores. For computing the CLR score between a pair of nodes *(i,j)* in the TRN, *Z*-score (Z_1) over the MI values of all pairs in which the first variable is *i* is computed, then Z-score (Z_2) over the MI values of all pairs in which the second variable is *j* is computed. After computing Z_1 and Z_2, the final score for an interaction is given as $\sqrt{Z_1^2 + Z_2^2}$. It should be noted that if Z_1 or Z_2 are negative they are set to zero i.e. they are not considered in the CLR score computation. We will be using two simple implementations of these Z-scoring schemes in our proposed approaches.

C. Proposed Approaches: sCoIn sCoIn Z_1 and sCoIn Z_2

The sCoIn (Scoring algorithm based on COmplex INteractions) algorithm, like the relevance network class of algorithms, is based on the mutual information metric. However, unlike these algorithms and our proposed approaches in past [23-28], in which network inference is based on just pair-wise MI computations, sCoIn considers inference of more complex structures. sCoIn starts with computing the mutual information between a gene/transcription factor and every other transcription factor that are stored in bin-1. After computing the pair-wise scores, sCoIn then computes MI between a gene/transcription factor and every two node combination of transcription factors and store them in bin-2. The algorithm then computes the MI between a gene/TF and all possible three node combination of TF's and stores them in bin-3. In this fashion, sCoIn computes MI between a gene/TF and various number of its potential regulator combinations. Figure 1 illustrates this MI computation scheme; the MI's between a gene/TF and every regulator is stored in bin 1, the MI's between gene/TF's and every possible two node combination is stored in bin 2 and so on. The number of bins, *M,* is a user selected threshold. The number of combinations in each bin, *b (b=1,…,M)*, is given as follows:

$$n_b = \binom{t}{b} \quad (7)$$

Hence, the total number of TF combinations considered across all the bins for any particular gene/TF is given by:

$$n_c = \sum_{b=1}^{M} n_b \quad (8)$$

Every MI score between a gene/TF (*i*) and its possible regulators, i.e., TF combinations in bin (*k*) is designated as $S^k_{i,j}$.

As the MI values between higher number of variables is usually higher [29], i.e. MI values between different number of variables are at different scales, the scores in any bin, k, ($S^k_{i,j}$) needs to be normalized first; such bin-wise normalized scores can then be combined to achieve a final score for each regulator that designate its suitability of acting as a regulator for the gene/TF under question. The bin-wise normalized score between a gene/TF (*i*) and every other TF (*j*) in bin k is given by:

$$BS^k_{i,j} = \frac{\sum_{a=1}^{n_k} S^k_{i,a}, where T_j \in combination - a}{k * \sum_{a=1}^{n_k} S^k_{i,a}}; i = 1,\dots,n; j = 1,\dots,t; k = 1,\dots,M \quad (9)$$

Once the bin-wise score is computed the final score between a gene/TF (*i*) and its potential regulators, i.e., TF (*j*) across all the bins is given by:

$$F_{i,j} = \sum_{k=1}^{M} BS^k_{i,j}; i = 1,\dots,n; j = 1,\dots,t; i = 1,\dots,N \quad (10)$$

Hence, the scores are computed for each gene/TF individually; however, for each such node, we first compute the bin-wise score of each TF showing its potential to regulate

```
INPUT :
 1. Expression Data : data
 2. Number of nodes : N
 3. Regulator List : T
 4. Number of regulators : t
 5. Maximum number of regulator combination : M

OUTPUT :
 N X t score matrix : F

ALGORITHM :
 Generate list of all possible regulator combinations
 %Initialize mutual information matrix
 muInfoMat ( N , n_c ) = 0;
 for i = 1 to N
  for j = 1 to n_c
   if i is not in regulator combination
    %here we can have the MI or Z - scores
    muInfoMat ( i , j ) = mutual information using eqn 6;
   else
    muInfoMat ( i , j ) = 0;
   end
  end
 end
 for i = 1 to N
  for j = 1 to t
   Obtain scores in F using equation 10 ;
  end
 end
 return F ;
```

Figure 3. Pseudo-code for sCoIn approaches

this node and then sum up such normalized scores across all bins to find the over-all suitability of this TF to regulate the node under consideration. Thus, our simple scoring scheme combines the effects of higher level structures (stored in each higher order bin) for each gene/TF individually. The final scoring matrix is exactly similar to any other relevance network based scheme, where the rows designate a node (gene/TF) and the columns designate all of possible regulators, i.e., TFs. Hence, we can apply a threshold on this final scoring matrix to obtain the final connectivity matrix. Figure 2 gives an example computation scheme of scores. In this example, scores were computed between a gene node and four TF's. Although the final scoring matrix will have five rows, this example is only showing the computations for the first row which designates the gene node.

SCoIn Z_1 and sCoIn Z_2 are extensions of sCoIn algorithm in which the initial MI matrix is further updated using Z-scores before our scoring scheme is applied. Again as MI values between different numbers of variables are at different scales, Z-score computations were performed for each bin separately. In sCoIn Z_1 algorithm, the MI values are standardized for each gene/TF separately considering the computed means and standard deviations for each row in bin k separately, as highlighted in figure 1, where as in sCoIn-Z_2, the MI values were standardized based on the computed means and standard deviations across both the bin-wise row and column (i.e., for all n nodes) levels as in the CLR algorithm. Hence $S_{i,j}^{k}$ now denotes the z-scores (instead of the MI values as done earlier), and we next apply our scoring scheme to compute the normalized scores between genes/TF's

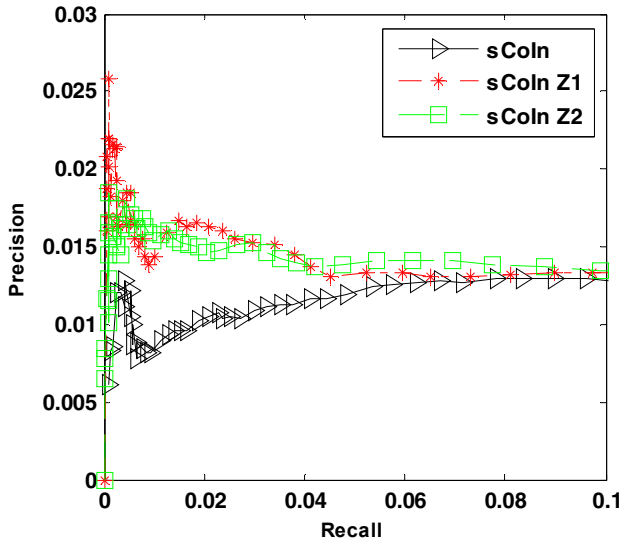

Figure 4. comparison of sCoIn approaches

and their regulator TF's as described in the sCoIn algorithm. The pseudo-code for the algorithms is given in figure 3.

Note that while we the proposed approaches infer structures, they do not attempt to infer a specific kind of motif.

D. Network and Data Sets

1) Escherichia Coli Network and Data using the genenetweaver tool:

The known complete *Escherichia coli's* network and data sets was obtained using the popular genenetweaver tool. The tool provided six types of data sets viz. wild-type, knockouts, knockdowns, multi-factorial perturbations, and time series. We only picked the time series data sets for our analysis (the algorithm works for non time-series data too). All the data sets were generated under the DREAM 4 2)challenge settings. The time series data set had ten different gene expression matrices under different perturbations and we selected the first data set for our analysis. The time series data had 21 time points with a time interval of ten minutes between each time point. The first time point in the data is the control where the expression levels of untreated cells are recorded. For performance analysis, a simple fold change model was implemented. Using the control time point, the fold-change for every time point was computed. The fold change is defined as the ratio between the expression levels of a gene/TF at that time point and the control time point. The network provided by genenetweaver had a total of 1389 genes and 176 transcription factors.

3) Performance analysis metrics

The benchmark precision and recall metrics were used in the sensitivity analysis of the proposed approaches. While a number of definitions exist for such precision and recall metrics [30], in this paper, recall is defined as *Te*/(*Te*+*Me*) and precision is defined as *Te*/(*Te*+*Fe*); where *Te* is the sum of correctly inferred edges, *Fe* is the sum of wrongly inferred edges and *Me* is the sum of edges that existed in the actual network but were not inferred by the algorithm.

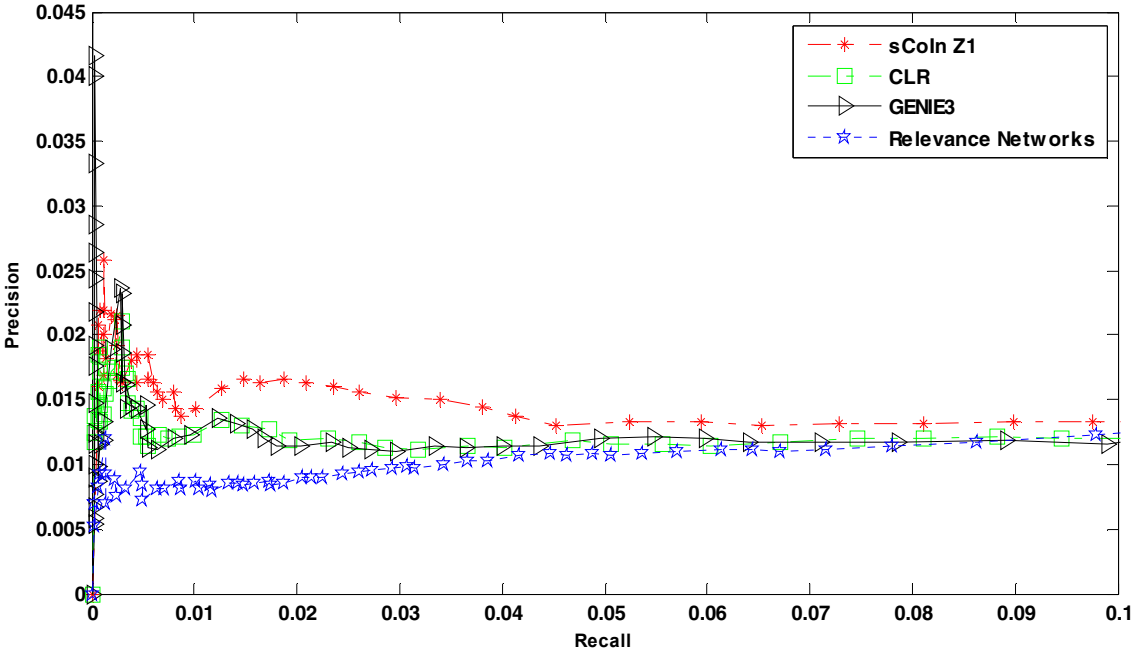

Figure 5. Accuracy comparison of sCoIn-Z1, GENIE3 and CLR

III. RESULTS AND DISCUSSION

1) Comparison Analysis of SCoIn, sCoIn Z_1, and sCoIn Z_2

Comparison analysis of all the three proposed algorithms was performed to choose the best performing scheme. Hundred different thresholds were applied over final score matrix obtained from each of these schemes using the genome-scale *Escherichia coli's* dataset and the precision and recall values were computed for every threshold used. The precision-recall plots of the algorithms are shown in figure 4. The plot shows that sCoIn- Z_1 consistently gives higher precision values for the same recall as compared to the other two proposed algorithms. sCoIn- Z_1 will hence be used for rest of the analysis.

2) Comparison analysis of sCoIn Z1, CLR, RN, and GENIE3

Comparison analysis of the three popular existing approaches CLR, RN, and GENIE3 with sCoIn-Z_1 was performed using precision recall metrics as discussed in previous section. GENIE3 and CLR showed similar accuracy behavior. With initial peaks i.e. higher precision at lower recall values, GENIE3 had a slight edge over CLR. SCoIn-Z_1 consistently showed higher precision values as compared to CLR and GENIE3. The plot for this analysis is shown in figure 5.

As mentioned earlier the number of bins and spline order parameters of the B-Spline MI computation approach were set to five and two respectively. Also when a very high threshold is used a single addition of a correctly inferred edge or a wrongly inferred edge will increase or decrease the precision by a huge margin as the number of edges inferred is small, due to this we observe a very high rise/fall in precision of the algorithms when higher thresholds are used.

3) Sensitivity analysis on number of bins

Note that the earlier results were generated by considering only up to 2 bins in the proposed sCoIn class of algorithms. However, as sCoIn can potentially consider even higher number of TF combinations for inference, in this analysis we study the accuracy behavior of the sCoIn-Z1 algorithm by considering up to two, three and four number of bins respectively. As the computational complexity of the scoring scheme increases exponentially with more number of bins, a smaller sized network with 100 genes and 16 TF's was chosen for this particular analysis. Again the network and data were obtained using the genenetweaver tool. Plots in figure 6 show that there was a small improvement in accuracy when higher numbers of bins were considered for inference. Hence, to keep our approach scalable to genome-scale inference, it is

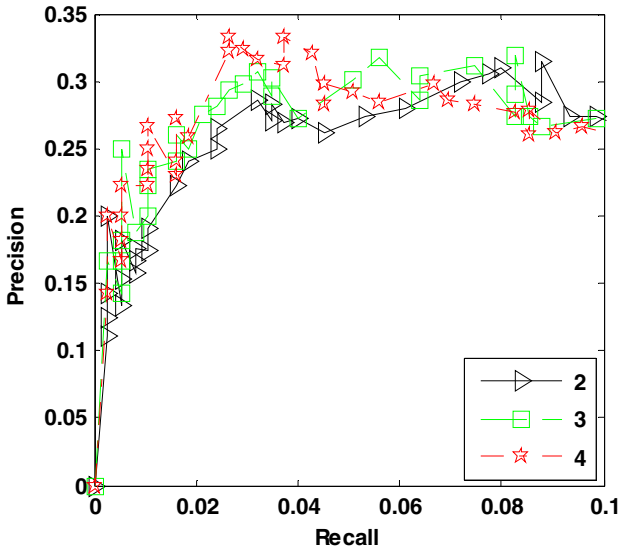

Figure 6. Bin size sensitivity for sCoIn-Z1

sufficient to consider up to 2-bins only; higher order bins may only give a slight improvement in the accuracy. We however plan to implement a parallelized version of sCoIn to validate this observation for even genome-scale networks.

IV. Conclusions and Future Work

In this paper, we have proposed a novel scoring scheme to design inference methods that can consider more complex regulatory interactions. Unlike existing approaches, the proposed algorithms do not assume all interactions to be pair-wise. Based on the benchmark metrics of precision and recall, we have shown that the proposed approaches infer networks with higher accuracy compared to popular state of art approaches at the genome-scale. In fact, our results demonstrate that the proposed approach performs even better than GENIE3 which has reported the best performance till date for reverse engineering gene regulatory networks. Moreover, restricting the proposed scoring scheme to considering only up to two bins provides sufficient inference accuracy and can easily scale to genome-level inference as demonstrated in our results. Our structure based inference approach as presented here is only a first step towards designing more accurate regulatory network inference algorithms which continue to be an area of active research.

References

[1] Alon, U. "Introduction to Systems Biology: Design Principles Of Biological Circuits". CRC Press, 2006.

[2] Levine, M. & Davidson, E. H. "Gene regulatory networks for development". Proc. Natl Acad. Sci, 2005, 4936–4942.

[3] Thieffry, D., Huerta, A. M., Perez-Rueda, E. & Collado-Vides, J. "From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in Escherichia coli". Bioessays 1998, 20:433–440.

[4] Babu MM, Lang B, Aravind L. "Methods to reconstruct and compare transcriptional regulatory networks". Methods Mol Biol. 2009, 541:163-80.

[5] Imoto S, Goto T, Miyano S. "Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression". Pacific Symposium on Biocomputing, 2002, 175–186.

[6] Murphy K, Mian S. "Modelling gene expression data using dynamic Bayesian networks". In Technical report 1999, Computer Science Division University of California, Berkeley, CA.

[7] Kauffman SA. "Metabolic stability and epigenesis in randomly constructed genetic nets". J Theor Biol, 1969, 22:437-467.

[8] Schmulevich I, Dougherty ER, Kim S, Zhang W. "Probabilistic Boolean Networks: a rule-based uncertainty model for gene regulatory networks". Bioinformatics, 2002, 18(2):261-274.

[9] Chen T, He HL, Church GM. "Modeling gene expression with differential equations". Pacific Symposium on Biocomputing, 1999, 4:29-40.

[10] Eisen MB, Spellman PT, Brown PO, Botstein D. "Cluster analysis and display of genome-wide expression patterns". Proc Natl Acad Sci USA, 1998, 95: 14863–14868.

[11] Butte AJ, Kohane IS. "Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements". Pacific Symposium on Biocomputing, 2000: 418–429.

[12] Liang S, Fuhrman S, Somogyi R. REVEAL. "A general reverse engineering algorithm for inference of genetic network architectures". Pacific Symposium on Biocomputing. 1998;3:18–29.

[13] Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Dalla Favera R, Califano A. "ARACNE: An algorithm for reconstruction of genetic networks in a mammalian cellular context". BMC Bioinformatics, 2006, 7 Suppl 1:S7.

[14] Faith J.J, Hayete B, Thaden J.T, Mogno I, Wierzbowski J, Cottarel G, Kasif S, Collins J.J, and Gardner T.S. "Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles". PLoS Biology, 2007, 5(1):e8.

[15] Chaitankar V, Ghosh P, Elasri MO, Perkins EJ. "sREVEAL: Scalable extensions of REVEAL towards regulatory network inference". IEEE ISDA, 2011, pp. 1365-1370.

[16] Lee, T.I. et al. "Transcriptional regulatory networks in Saccharomyces cerevisiae". Science, 2002, 298:799-804.

[17] Huynh-Thu VA, Irrthum A, Wehenkel L, Geurts P (2010) Inferring Regulatory Networks from Expression Data Using Tree-Based Methods. PLoS ONE 5(9): e12776. doi:10.1371/journal.pone.0012776

[18] Marbach D, Prill RJ, Schaffter T, Mattiussi C, Floreano D, and Stolovitzky G. Revealing strengths and weaknesses of methods for gene network inference. PNAS 2010, 107(14):6286-6291.

[19] Marbach D, Schaffter T, Mattiussi C, and Floreano D. Generating Realistic in silico Gene Networks for Performance Assessment of Reverse Engineering Methods. Journal of Computational Biology 2009, 16(2) pp. 229-239.

[20] Cover T M, Thomas J A: Elements of information theory. Wiley-Interscience, New York, 1991.

[21] C. Daub et al., "Estimating Mutual Information Using B-spline Functions—an Improved Similarity Measure for Analysing Gene Expression Data," BMC Bioinformatics, vol. 5, p. 118, 2004.

[22] J. Zola, M. Aluru, A. Sarje, and S. Aluru, "Parallel Information Theory Based Construction of Genome-wide Gene Regulatory Networks," IEEE Transactions on Parallel and Distributed Systems, vol. 21, iss. 12, pp. 1721-1733, 2010.

[23] Chaitankar V et al. "A scalable information theory based gene regulatory network inference method from time series and knock-out data" BICoB, 2011, pp.74-79.

[24] Chaitankar et al. "Gene regulatory network inference using predictive minimum description length principle and conditional mutual information". IJCBS, 2009, pp.487-490.

[25] Chaitankar et al. "Effects of cDNA microarray time-series data size on gene regulatory network inference accuracy". ACM-BCB, 2010, pp.410-413.

[26] Chaitankar et al. "A novel gene regulatory inference algorithm using predictive minimum description length approach". BMC Systems Biology, 2010, 4 Suppl 1:S7.

[27] Chaitankar et al. "Time lagged information theoretic approaches to the reverse engineering of gene regulatory networks". BMC Bioinformatics, 2010, 11 Suppl 6:S19.

[28] Chaitankar et al. "Predictive minimum description length principle approach to inferring gene

[29] A. Kraskov, H. Stögbauer, and P. Grassberger. "Estimating Mutual information", Phys. Rev. E 69 (6) 066138, 2004

[30] Zhang X, Baral C, Kim S. "An Algorithm to Learn Causal Relations Between Genes from Steady State Data": Simulation and Its Application to Melanoma Dataset. Proceedings of 10th Conference on Artificial Intelligence in Medicine (AIME 05), Aberdeen, Scotland. 2005. pp. 524–534.