Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

A Neural Network Based Approach for Inference and Verification of Transcriptional Regulatory Interactions

S. Knott[1], S. Mostafavi[2], P. Mousavi[3], *Member, IEEE*

***Abstract*—In this paper, we present a comprehensive neural network based modeling and validation framework for reverse engineering gene regulatory interactions. We employ two approaches, Gene Set Stochastic Sampling and Sensitivity Analysis, to infer these interactions. We first apply these methods to a simulated artificial dataset to ensure their correctness and accuracy. True biological interactions are then modeled by analyzing a rat hippocampus development dataset. Finally, we present a thorough computational methodology to test the validity and robustness of the inferred regulations through novel assemblies of relevant testing datasets.**

I. Introduction

The control of transcription is an integrated mechanism involving regulatory interactions between genes [1, 2]. Genes that regulate one and other comprise a genetic network [3]. Given the relative expression levels (mRNA or protein levels) of a set of interacting genes at different time points, a model of the gene interactions can be developed through different reverse engineering techniques [4].

In recent years, many reverse engineering techniques have been applied to this problem. These include Boolean networks [5], mutual information based techniques [6, 7], Bayesian networks [8, 9], local invariant methods [10], additive models [11, 12], genetic algorithms [13], neural networks [14] and neural genetic hybrids [15, 16].

Additive models represent changes in the expression level of each gene at a given time point as a weighted sum of all of its regulatory inputs at previous time points [11]. It is known that genetic networks display complex non-linear network dynamics. Thus non-linear additive models are the most biologically plausible [17]. Artificial neural networks are powerful mathematical tools that can be used to learn complex non-linear functions, and can therefore be used to simulate non-linear additive models to model genetic regulatory networks.

In the current study we present a comprehensive reverse engineering and computational-validation approach to modeling genetic networks as non-linear systems. We utilize two algorithms to identify genetic interactions from trained neural networks: Sensitivity Analysis (SA) [18], a heuristic search approach and Gene Set Stochastic Sampling (GSSS), a stochastic search approach. First we validate these methods on a simulated dataset. Following this, we apply these methods to a biological dataset. Finally, we present an extensive test strategy to evaluate the accuracy of the inferred interactions.

Manuscript received March 4, 2006.

S. Knott is with the Dept of Computer Science, Queen's University, Kingston, Ontario, Canada, K7L 3N6 (email: knott@cs.queensu.ca).

S. Mostafavi is with the Dept of Computer Science, Queen's University, Kingston, Ontario, Canada, K7L 3N6 (email: sara@cs.queensu.ca).

P. Mousavi is with the Dept of Computer Science, Queen's University, Kingston, Ontario, Canada, K7L 3N6 (phone: 613 533 6070) (email: pmousavi@cs.queensu.ca).

II. Methods

II.1 Modeling gene interactions using neural networks:

Neural network performance is assessed by the sum squared error (SSE) of the network predictions. We employ feed-forward and Elman network architectures in our modeling strategies. In the feed-forward network, input nodes are directly connected to one output node with a non-linear transfer function (tan-sigmoid function). In contrast the Elman network contains a hidden node that is connected to itself through a context unit. The recurrent layer in the Elman network allows it to make predictions based on all previously presented data. Both neural networks were trained employing Bayesian regularization which offers fast convergence and a high degree of generalization [19].

Gene Set Stochastic Sampling

GSSS infers regulatory interactions by predicting target gene profiles with small input gene subsets. An exhaustive search will identify a gene subset that is the most predictive of the target gene expression profile (revealing its most likely regulators). Due to the computational infeasibility of an exhaustive search, GSSS performs stochastic samplings of genes and tests their ability to predict a target gene profile. An SSE threshold, τ, is dynamically assigned to select the 1000 most predictive subsets and the most likely regulators of the target gene are then identified based on their occurrence in these most predictive subsets. For a more detailed description of the algorithm see [18].

Sensitivity Analysis

SA infers regulatory interactions by systematically perturbing the expression profile of each input gene and examining the resultant error in a trained network. A neural network is trained with the expression profiles of input genes to predict a target gene profile. For each input gene a 'jitter' equal to +/- the standard deviation of its expression level divided by four is added to its profile. Perturbed genes that cause the greatest increase in prediction error of a target profile can be inferred as its regulators. The 'jitter' is random in nature, therefore each run of the algorithm can result in slightly different perturbed patterns for each gene. To design a robust methodology for regulator inference, SA is performed multiple times, each time taking advantage of a small difference in the perturbation scheme. The genes with the highest occurrence in the resultant subsets are taken as the most likely regulators of the target. For a more detailed description of the algorithm see [18].

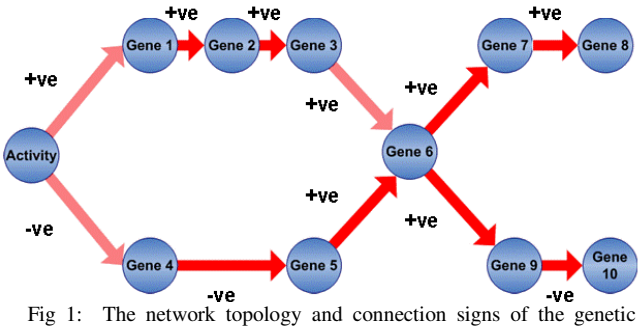

Fig 1: The network topology and connection signs of the genetic regulatory interactions simulated in the artificial dataset. The highlighted connections were identified by both GSSS and SA.

II.II Artificial and Biological Datasets:

Artificial Dataset

The artificial dataset used to validate our methods was produced by Smith *et al.* [20], employing the Bayesian network based BRAINSIM simulator. The simulated dataset consists of measurements of 100 genes taken from 4 simulated tissue regions in 6 simulated patients. The networks underlying the datasets between regions and patients have the same topology, but differ slightly in their weight magnitudes. Only 10 of the simulated genes are associated with one and other through regulatory interactions while the remaining 90 genes serve as distracters and represent 'noisy' and irrelevant gene measurements. The expression levels of two genes in the network are directly affected by an activity node representing a stimulated (on) biological system and non stimulated (off) biological system. These two genes, in turn, affect the expression levels of 8 downstream genes (Fig 1). At each time point, the expression levels of genes in the network are governed by the expression levels of their regulators, a degradation factor and a noise factor. Whereas, the expression levels of the 90 other genes randomly fluctuate within the upper and lower expression level bounds.

Due to the stochastic nature of the added noise in the system, when the model is run on separate instances the simulated output will differ slightly but will be governed by the same underlying genetic network interactions. We use 20 datasets generated from 20 different BRAINSIM runs, where each dataset is comprised of 20 time points, sampled at simulated 5 minute intervals. Twelve of these datasets are concatenated and used for training and the remaining 8 are concatenated and used for testing. For a more detailed description of BRAINSIM see [20].

Hippocampus development dataset

We apply our methods to the hippocampus development dataset developed in [21]. It is comprised of 70 genes whose expression levels are measured by real time polymerase chain reaction (RT-PCR) over 11 non-uniformly separated time points, namely 0.25h, 0.5h, 1.5h, 3h, 6h, 24h, 48h, 10d, 21d, 32d, 49d (where h is hours and d is days). The dataset is low in experimental noise and accurate, therefore favorable for testing modeling strategies [22]. In addition, several other studies have previously explored this dataset making a comparative analysis available [10, 15, 16, 22].

To obtain sufficient training and testing data points, piecewise cubic Hermite Spline interpolation is employed to approximate measurements at hour intervals between the measured time points. This technique fits a polynomial of third degree between every two consecutive time points. Hermite Spline interpolation is the preferred method, as Hermite curves do not oscillate when the underlying function is not smooth and are able to fit the data without under/over shooting between the basis points [23].

A. II.III Validation of acquired gene regulatory networks:

Training and Test data

The lack of replicates in the hippocampus development dataset requires that dataset partitioning schemes be developed to separate training and testing data. In the first method, the dataset is separated into training and test sets by selecting measurements from alternating time points. Training time points are selected at t=1h, t=3h t=5 h… and testing time points are selected at t=2h, t=4h and t=6h...

In a second testing method, a 'noisy' dataset similar to the original data is derived by adding Gaussian noise to each profile, with a mean of zero and a standard deviation proportional to the fluctuation of the corresponding profile. The change in SSE of a robust network when simulated on a 'noisy' input dataset should be insignificant.

The interpolation process increases the similarity between neighboring data points. To ensure that time-closeness between interpolated training and test samples does not cause favorable testing results, a partitioned separation scheme is devised. In this technique, training data is constructed using only the first 60% of the available time points and the testing data is comprised of the second 40%.

Reverse Prediction

Causal relationships are time dependent, thus neural networks predicting based on causality should be able to predict target profiles at time t only when given the input gene profiles at time $t - \Delta t$. In contrast, if a neural network predicts based on correlations between its inputs and outputs it should predict expression levels at time t given the expression levels of the input genes at time $t - \Delta t$ or time $t + \Delta t$. To ensure that only causal relationships are being inferred, we implement reverse prediction.

In one reverse prediction scheme, SA is applied to a reversed gene expression dataset where the original final time point measurement becomes the initial time point and the order of the data points is reversed. If analysis on a reversed dataset results in overlapping regulators with the regular dataset, correlation based predictions are likely. In a second method, neural networks trained and tested with regulators inferred on an original dataset, are tested on corresponding reversed time point data to compare SSEs. In a more stringent analysis, the inferred regulators that are most correlated with the target genes are removed from the input space and resultant testing SSEs on the normal and reversed dataset are compared.

Table I: Inferred regulators of preGAD67 and G67186 resulting from Gene Set Stochastic Sampling.

Network Architecture	Regulating Genes of PreGAD67	Regulating Genes of G67186
Unique Regulators Inferred by the Elman Network	TCP	TH
	IGFR1	NFH
	GAD67	-
	mGluR3	-
Unique Regulators Inferred by the Feed-Forward Network	-	5HT2
	-	IGFR1
Intersection of Regulators Inferred by the Elman Network and Feed-Forward Network	COCO2	COCO1
	IGF2	Cyclin A
	mAChRa3	IGF2
	TH	mACHR1
	preGAD67	preGAD67

III. RESULTS

III.I Results – Artificial dataset

GSSS and SA were applied to each 'network' gene in the artificial dataset. GSSS and SA were able to successfully infer 8 of the 9 regulatory interactions as shown in Fig 1. In all cases our analysis was unable to identify gene 3 as a co-regulator of gene 6.

III.II Results - Hippocampus development dataset

Gene Set Stochastic Sampling

For each target gene, GSSS was performed, sampling 20,000 gene sets of size 7, using both Elman and feed-forward network architectures. The maximum permissible τ was set to 0.09, as higher SSEs resulted in insufficient prediction accuracies. Genes that had a percent occurrence greater than 0.75 in the predictive gene sets were inferred as regulating genes (see Table I). Finally the network was retrained with the most probable regulating genes and corresponding training and testing errors were recorded.

For preGAD67, inferred regulators were trained and tested using all training/testing data separation techniques discussed in *II.III*. The SSEs on all test sets were below 0.07 for the feed-forward networks and below 0.02 for the Elman network architecture (Fig 2). Similar results were seen when the same analysis was applied to G67186.

Sensitivity Analysis

SA was applied 100 times to identify regulators for each gene in the dataset. The algorithm was terminated when the number of remaining probable regulators of the target gene reached 10. The reported regulating genes had a probability occurrence of at least 0.75 in the 100 runs (Table II). For preGAD67, inferred regulators were trained and tested using all training/testing data separation techniques discussed in *II.III*. The SSE on the test sets were below 0.06 for the feed-forward networks and below 0.02 for the Elman network architecture.

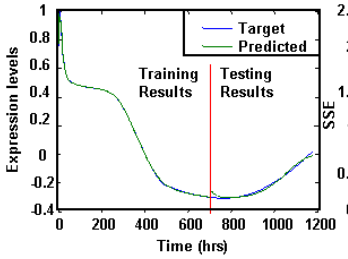

Fig 2: The training and testing output of an Elman network for preGAD67 using only the regulators that were inferred through GSSS as it was applied to the partitioned dataset (SSE = 0.02)

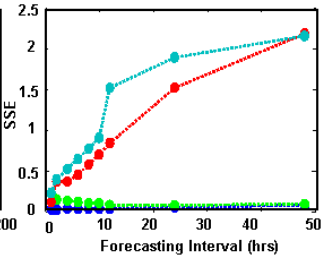

Fig 3: SSEs for network predictions of G67186. Testing SSE (blue) reverse testing SSE (Red) testing SSE with correlated genes removed (green) reverse testing with correlated genes removed (aqua).

Table II: Inferred regulators of preGAD67 and G67186 as resulting from Sensitivity Analysis.

Network Architecture	Regulating Genes of PreGAD67	Regulating Genes of G67186
Unique Regulators Inferred by the Elman Network	GRa4	TH
	5HT2	GRa2
	-	H4
Unique Regulators Inferred by the Feed-Forward Network	Nestin	CyclinA
	GRg1	mACHR1
	-	mACHR2
Intersection of Regulators Inferred by the Elman Network and Feed-Forward Network	InsR	InsR
	IGFR2	IGFR2
	IGF2	IGF2
	mACHR1	PDGFb
	CyclinB	NFH
	NFL	IP3R3
	nmACHRa7	nACHRa7
	Brm	-

III.III Results: Reverse Prediction

SA was applied to the reversed time dataset to infer the probable regulating genes of the target gene G67186. The results had only one gene in common with the predicted regulatory set from analysis on the original dataset.

In a second experiment the inferred regulators of G67186, via SA, were used to train a feed-forward neural network to predict the expression profile of G67186 1h, 2h, 4h, 6h, 10h, and 24h and 48h ahead of time. These networks were then simulated on the reversed time point dataset resulting in a prediction SSE of at least 5 orders of magnitude larger than the training SSE (Fig 8). Finally, the three regulatory genes most correlated with G67186 (mAChR1, PDGFb, IP3R3) were removed form the input space and simulations were performed with the original and reversed datasets. Simulation SSEs for the original data remained low. However, simulation SSEs increased dramatically on the reversed dataset, inferring that predictions are not being made based on correlated expression profiles, but on causal relationships (Fig 3).

IV. Discussion

GSSS and SA were validated on their ability to capture known interactions in the artificial dataset, as they captured eight of the nine interactions represented in the dataset. When this dataset was analyzed previously in [20], the Bayesian network based NETWORKINFERENCE algorithm was also unable to predict the co-regulation of gene 6 by gene 3. Gene 5 and gene 3 both regulate gene 6 in a coordinated fashion with the minimum expression level of the pair serving as the regulation limiting factor. By examining the expression data across temporal samples, it was observed that gene 5 had a lower expression level than gene 3 in ~89 % of the time points, thus, gene 5 served as the effective regulator of gene 6 [20]. This confirms the reliability of viability of GSSS and SA.

When GSSS and SA were performed on the rat hippocampus development dataset, it became apparent that the two network architectures (feed-forward and Elman) produced overlapping results in their inferred regulatory genes for each target gene (Table I and Table II), indicating robustness to the choice of network architecture.

The two developed modeling approaches (GSSS and SA) also produced overlapping results when applied to infer regulating genes of a common target gene as three of the ten predicted regulators of G67186 were shared between the two methods (Table I and Table II).

The accuracy and robustness of the inferred regulatory interactions were confirmed through testing trained networks with 'noisy' input data, as well as alternate testing and partitioned testing data. SSEs on the testing datasets had a maximum of 0.07 and a minimum of 0.02 which result in expression profile predictions that very closely follow the target profiles (Fig 2).

Causal relationships between target genes and their inferred regulators were verified through reverse prediction analysis. Inferred regulators were unable to accurately predict the expression profiles of their target genes in a reversed time point manner. The causality of the inferred relationships was verified further, when we eliminated the inferred regulators that were most correlated with a target gene and observed that the SSEs for prediction with the original data remained low, while the SSEs on the reverse time point testing set increased in magnitude (Fig 3).

In conclusion, two approaches involving neural networks aimed at inferring genetic regulatory networks were validated on an artificial dataset. Both methods were able to detect all but one of the artificial genetic interactions represented in the artificial dataset, verifying their validity. The methods were then applied to the rat hippocampus dataset [21]. The modeling approaches were shown to be robust to the architecture of the neural networks employed. The resultant networks were then tested and verified, by employing a thorough computational methodology to determine the validity and robustness of the inferred regulations through novel assemblies of relevant testing datasets.

References

[1] McClean, P. *Prokaryotic GeneExpression.* Available: "www.ndsu.nodak.edu/instruct/mcclean/plsc431/prokaryo1.htm" 1997

[2] King M.W. "Introduction to Gene Expression and Regulation." Available: "http://web.indstate.edu/thcme/mwking/gene-regulation.html"

[3] Hunter, L. "Molecular Biology for Computer Scientists." *Artificial Intelligence for Molecular Biology*, Ed. L. Hunter, pp. 1-46, AAAI Press, 1993.

[4] Keedwell, E and Narayanan, A. "Genetic Algorithms for Gene Expression Analyasis." *Applications of Evolutionary Computing* LNCS 2611 Gunther Raidle *et al.* (Eds.), *proceedings of EvoBIO2003 1st European Workshop on Evolutionary Bioinformatics 76-86*, 2003.

[5] Kauffman, S.A. "Metabolic stability and epigenesis in randomly constructed genetic nets." *Journal of Theoretical Biology*, *22*, 437-467, 1969.

[6] Margolin, A., Nemenman, I, Wiggins, C, Stolovitzky, G, and Califano, A. "On the reconstruction of interaction networks with applications to transcriptional regulation"HHHH. In G Chechik, C Leslie, G Ratsch, and K Tsuda, editors, TTT*NIPS'04 Computational Biology Workshop*TTT, Available: "HHTTUhttp://arxiv.org/abs/q-bio/0410036TUTHH"

[7] Chaudhari, A. "Reverse Engineering of Genetic Networks using Information Theory." Available: "www-scf.usc.edu/~ajchaudh/website_html/reverse.PDF"

[8] Husmeier D. "Reverse Engineering of Genetic Networks with Bayesian Networks." *Biochem Soc Trans.* Dec;31(Pt 6):1516-8, 2003.

[9] Husmeier D. *"Sensitivity and specificity of inferring Genetic regulatory interactions from microarray experiments with Dynamic Bayesian networks."* Bioinformatics: 19(17):2271-82. Dec. 2003.

[10] Fofanov, Y. Montgomery Pettit, B. "Reconstruction of the genetic regulatory dynamics of the rat spinal cord development: Local Invariants approach." *Journal of Biomedical Informatics* (5-6): 343-351. Oct. 2002.

[11] D'haeseleer, P., Wen, X., Fuhrman, S., and Somogyi, R. "Linear Modeling of mRNA Expression Levels during CNS Development and Injury." *Pacific Symposium on Biocomputing,* pp. 41-52, 1999.

[12] Chen, T. He, HL., Church, GM. 1999. Modeling gene expression with differential equations. *Pacific Symposium on Biocomputing,* 4, 29-40. http://www.smi.stanford.edu/projects/helix/psb99/Chen.pdf

[13] Ando, S., and Iba, H. "Inference of Gene Regulatory Models by Genetic Algorithms ." *Proceedings of Conference on Evolutionary Computation* pp: 712-719. 2001.

[14] Keedwell, E., Narayanan, A, Savic, D. "Constructing gene regulatory networks using artificial neural networks." *The Proceedings of the 2002 International Joint Conference on Neural Network.* pp:183-188. 2002.

[15] Keedwell, E. and Narayanan, A. "Genetic Approaches to Reverse Engineering Regulator Networks from Gene Expression Data." Available at: "www.dcs.ex.ac.uk/~anarayan/publications/IEEETransac2.pdf" 2004.

[16] Wahde, M., and Hertz, J. "Coarse-grained Reverse Engineering of Genetic Regulatory Networks." *Biosystems:* 55, pp. 129-136. 2000.

[17] De Jong, H. and Page, M. "Qualitative Simulation of Large and Complex Genetic Regulation Systems." *ECAI 2000: 141-145,* 2000.

[18] Knott

[19] Forsee, F.D. and Hagan MT. "Gauss Network Approximation to Bayesian Learning." *Proceedings of the 1997International Joint Conference on Neural Networks.*pp. 1930-35. 1997.

[20] Smith, VA., Jarvis, ED., Hatermink, AJ. 2002. Evaluating functional network inference using simulation of complex biological systems. *Bioinformatics*. 18 suppl. 1:S216-S224.

[21] Wen, X., Fuhrman, S., Michales, G.S., Carr, D.B., Smith, S., Barker, J., and Somogyi, R. "Large-scale Temporal Gene Expression Mapping of Central Nervous System Development." *Proc.Natl.Acad.Sci:* 95:334-339, 1998.

[22] Deng, X., and Ali, H. "A Computational Approach to Reconstructing Gene Regulatory Networks." *CSB 2003:*413-414. 2003.

[23] Pipenbrinck N. "Hermite Curve Interpolations." Available: HHTTU"www.cubic.org/~submissive/sourcerer/hermite.htmTUTHH. "1998.