2014 IEEE 27th International Symposium on Computer-Based Medical Systems

Towards using Probabilities and Logic to Model Regulatory Networks

António Gonçalves*, Irene Ong†, Jeffrey A. Lewis‡ and Vítor Santos Costa*
*CRACS INESC-TEC and Department of Computer Science
Faculty of Sciences, Universidade do Porto, Portugal 4169-007
Email: {up201008720@alunos.dcc.fc.up.pt, vsc@dcc.fc.up.pt}
†Great Lakes Bioenergy Research Center
University of Wisconsin, Madison, WI 53706
Email: ong@cs.wisc.edu
‡Department of Biological Science
University of Arkansas, Fayetteville, AR 72701
Email: lewisja@uark.edu

***Abstract*—Transcriptional regulation plays an important role in every cellular decision. Unfortunately, understanding the dynamics that govern how a cell will respond to diverse environmental cues is difficult using intuition alone. We introduce logic-based regulation models based on state-of-the-art work on statistical relational learning, and validate our approach by using it to analyze time-series gene expression data of the $\mathrm{Hog1}$ pathway. Our results show that plausible regulatory networks can be learned from time series gene expression data using a probabilistic logical model. Hence, network hypotheses can be generated from existing gene expression data for use by experimental biologists.**

***Keywords*-Bioinformatics; Gene Regulation; Genomics; Network/Pathway Analysis; Statistical Relational Learning;**

I. Introduction

Many major cellular decisions involve changes in networks of proteins that perform transcriptional regulation. With the advent of high-throughput technologies and advanced measurement techniques molecular biologists and biochemists are rapidly identifying components of these networks and determining their biochemical activities, but understanding how these complex multicomponent networks govern how a cell will respond to diverse environmental cues is difficult using intuition alone. Thus, our goal is to build probabilistic logical models to uncover the structure and dynamics of such networks and how they regulate their targets.

Despite the challenge of inferring genetic regulatory networks from gene expression data, various computational models have been developed for regulatory network analysis. Examples include approaches based on logical gates [1], [2], and probabilistic approaches, often based on Bayesian networks [3]. On one hand, logic gates provide a natural, intuitive way to describe interactions between proteins and genes. On the other hand, probabilistic approaches can handle incomplete and imprecise data in a very robust way.

Our main contribution is in introducing a model that combines the two approaches. Our approach is based on the probabilistic logic programming language ProbLog [4], [5]. In this language, we can express true logical statements (expressed as *true rules*) about a world where there is uncertainty over data, expressed as *probabilistic facts*. In the setting of gene expression, this corresponds to establishing:

- a set of *true rules* describing what are the possible interactions existing in a cell;
- a set of *uncertain facts* describing which possible rules are applicable to a certain gene or set of genes.

Given time-series gene expression data, we want to choose the probability parameters that best describe the data. ProbLog's approach is to reduce this problem to an optimization problem, and use a gradient ascent algorithm to estimate a local solution [6] in the style of logistic regression.

We validate our approach by using it to study expression data on an important gene-expression pathway, the $\mathrm{Hog1}$ pathway [7] in budding yeast. It is well known that under conditions of osmotic stress, the protein kinase $\mathrm{Hog1}$, and the paralogous proteins $\mathrm{Msn2}$ and $\mathrm{Msn4}$ interact to create a response that envolves the expression of a large number of proteins. We model these pathways by also incorporating the two transcription factors activated by $\mathrm{Hog1}$: $\mathrm{Hot1}$ and $\mathrm{Sko1}$.

II. Related Work

Despite the difficulty of deciphering genetic regulatory networks from microarray data, numerous approaches to the task have been quite successful. Friedman *et al.* [3] were the first to address the task of determining properties of the transcriptional program of *S. cerevisiae* (yeast) by using Bayesian networks (BNs) to analyze gene expression data. Other approaches include Boolean networks (Akutsu *et al.* [8], Ideker *et al.* [9]) and other graphical approaches (Tanay and Shamir [10], Chrisman *et al.* [11]).

The methods above can represent the dependence between interacting genes, but they cannot capture causal relationships. In our previous work [12], we proposed that the

1063-7125/14 $31.00 © 2014 IEEE
DOI 10.1109/CBMS.2014.9

analysis of time series gene expression microarray data using Dynamic Bayesian networks (DBNs) could allow us to learn potential causal relationships. A more recent approach was done by Marshal *et al.* [13], where they developed a decision theoretic method that exploits publicly available information, captured in a comprehensive interaction network to obtain a mechanistic view of genes.

III. Methods

The ProbLog [4], [5] language was initially motivated by the problem of representing a graph where there is uncertainty over whether edges exist or not. As a straightforward example consider the directed graph in Figure 1.

Notice that each edge has a probability of being true. As an example, starting from a we can reach b with probability 0.2 and c with probability 0.5. We assume that all the different probabilities are *independent*.

ProbLog represents edges as probabilistic facts which allows us to establish true rules for defining a path between any two nodes:

```
0.2::edge(a,b).
0.5::edge(a,c).
0.7::edge(b,c).
....

path(N,N).
path(N,E) :-
     edge(N,M),
     path(M,E).
```

The first rule says that there is a path between node N and E if $N = E$. The second rule defines path recursively: there is a path from N to E, if from N one can reach M and from M there is a path to E. Thus, in this example, to find a path between d and e we would call the second rule to reach e and then just use the first rule as the base case. To find a path between c and e we could try $M = \mathtt{d}$ and we are back to d to *e*, or we can just follow the edge from c to e. Given this program, ProbLog allows one to answer several queries, such as *what is the most likely path between two nodes*, and *what is the total probability that there is a path between two nodes*.

A. Learning ProbLog Programs

Arguably, the most fundamental task in learning ProbLog programs is *parameter learning*: to obtain the best values for the fact probabilities, given a program structure. There are a number of approaches to this problem. LeProbLog [6] provides a very natural and general algorithm, that is the one that best suits our task. We refer the reader to Gutmann et. al. [6] for a complete discussion and detailed derivation assuming the parameters follow a sigmoidal function. In practice, this result means that to compute the gradient over some parameter θ_i we simply have to follow a bottom-up dynamic programming algorithm, very much in the same style as what we had to do to compute the probability.

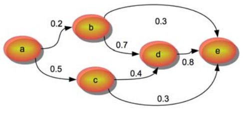


Figure 1. A simple directed graph, where each edge has a probability of being true.

IV. Experimental Methodology

We obtained time-series gene expression data from Lee et al. [14] for our experiments. The experiments followed the response of actively growing Saccharomyces cerevisiae to an osmotic shock of 0.7 M NaCl. The dose of salt was selected by the experimentalists to provide a robust physiological response but allow high viability and eventual resumption of cell growth. The samples were collected before and after NaCl treatment at 30, 60, 90, 120, and 240 min (measuring the peak transcript changes that occurs at or after 30 min) [15]. We focused our attention on the 270 genes of the Hog1 Msn2/4 pathway from Capaldi [7] for which we have expression data.

We performed three types of experiments:

1. Compared the three time-series to validate data quality and better understand the general patterns in the data.
2. Computed correlations to obtain a first, robust, quantification of the main relationships in the pathways.
3. Used the temporal data to better estimate the relationship patterns from the data.

In the first and second step we use correlations (*Pearson product-moment correlation coefficient*) between normalized gene expression values. We first compute correlations between genes in the same experiment, where X and Y range over the temporal data in order to calculate the *r* between each pair of genes. We experimented with both correlating X_t and Y_t (same time-step) and correlating X_t and Y_{t+1} (next time-step).

In our first experiment, we further compared the r-coefficient obtained between all correlations in the three different experiments. Although all three experiments followed the same methodology, variations in initial conditions can significantly affect gene expression. We study this effect in order to understand the inherent variability existing in the data.

In our second experiment, we assume that if a pair of genes has high absolute correlation it also has a high probability of being connected in the pathway [16], resulting in two possible connected nodes in the gene interaction network. This information is then used as training examples for the ProbLog learner. The ProbLog background knowledge is a variation of the graph background we used before and is of the form:

```
path(X,Y,S) :-
        hog1_promoter(X),
        hog1_promoter(Y),
        edge(X,Y,S).
path(X,Y,S) :-
        hog1_promoter(X),
        path(X,W,_),
        edge(W,Y,S).
```

The *hog1_promoter* genes are Hog1, Msn2, Msn4, Hot1, and Sko1. The S flag indicates whether the edge corresponds to activation or inhibition. The `Connected` relationship is defined either as:

```
edge(X,Y,S) :- connected(Y,X,S).
edge(X,Y,S) :- connected(X,Y,S).
```

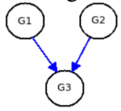
in the case of learning an undirected graph (same time-step correlations), and as:

```
edge(X,Y,S) :- connected(Y,X,S).
```

for a directed graph.

The third experiment aims for a more detailed picture of the learned network by using the temporal nature of the data. The output generated is a weighted, directed gene network, but nodes are connected as a gated network:

AND: two promoter genes need to be active in order to activate a gene, as shown in the graph.



```
active(G3,T1,Z) :-
        next_step(T0,T1),
        and(G1,G2,G3),
        active(E,T0,G1),
        active(E,T0,G2).
```

There are a total of 4 gates, the **AND**, **OR**, **XOR** and **Single**.

We use two different forms of temporal data: expression level (E), and variation (Δ). We experimented with three different approaches:

- Level influences variation (LV).
- Variation influences variation (VV).
- Level influences level (LL).

One important advantage of the approach is that it allows us to implement *soft constraints* on the probability distribution. These constraints are implemented by saying that satisfying some rule must have probability 1 or 0. In experiment three, we implement constraints saying that a *gene must be explained by a single rule*. In practice, we must be careful not to flood the system with soft constraints. In our experiment we implemented one joint soft constraint per gene.

A. Implementation

We fed the expression data to a domain pre-processor that generates the E and Δ data, and computes correlations. We then use ProbLog, implemented using YAP Prolog [17] and CUDD [18] to generate and output graph and some statistics. The output graph is translated to graphviz format for visual inspection.

V. Experimental Results

Our first experiment was designed to understand data variation between three experiments. To do so, we simply compute the correlation between gene expression level between every pair of experiments as the experiment proceeds across time. We compute this expression between all genes, and only between the genes in the Hog1/Msn24 network.

Table I
Experiment 1: Variation in Correlation Between All Genes and Experiments Along Time

Time	Experiment Pair					
	All Genes			*Hog1 Genes*		
	$1-2$	$1-3$	$2-3$	$1-2$	$1-3$	$2-3$
30	0.84	0.88	0.91	0.83	0.89	0.90
60	0.64	0.77	0.80	0.84	0.88	0.89
90	0.71	0.79	0.79	0.87	0.88	0.87
120	0.71	0.75	0.77	0.84	0.84	0.83
240	0.52	0.65	0.53	0.80	0.80	0.73

Table II
Experiment 2: Proposed parents with threshold $0.7 \leq P \leq 0.9$

Gene(s)	90%	80%	70%
Msn2	7	37	61
Msn4	1	22	68
Hog1	0	14	35
Hot1	0	8	38
Sko1	1	21	50
Msn2/Msn4	0	8	31
Msn2/Sko1	0	9	28
Msn4/Sko1	0	9	22
Msn2/Hog1	0	3	9
Total	9	102	153

The experiments show initially a strong correlation, close to 90% after 30 min. The correlation decreases to 50% after 120 min. The decrease is noticeable in every experiment: no experiment seems different from the others. Regarding the genes in the Hog1 pathway, the set of genes remain correlated up until 120 min.

Experiment two is about obtaining the main correlation between genes and their direct regulators, by using the correlations between the genes as examples. We were able to get a network that is very similar to one presented by Capaldi et al. [7].

Next, we try to find out which promoter gene regulates a Hog1 or Msn2/4 gene. Table II shows what promoters are proposed as parents, with the threshold ranging from $0.7 \leq P \leq 0.9$. They also show the major cases where we have multiple parents for a node. The results suggest that Msn2/4 and Hog1 have the most direct impact on gene expression. Moreover, as we lower the threshold it becomes harder to distinguish the influence of individual promoters.

The algorithm finds 107 genes with parents at $P > 0.3$. Of those 107, 9 have > 2 parents, indicating the algorithm has not converged yet. The distribution of gates over the remaining 98 is given in Table III. Most target genes are controlled positively by Msn2/4 or Hot1. It is interesting to observe the repression from Hog1/Sko1 [19].

VI. Conclusion

Learning regulatory networks from gene expression is a hard problem. Data is noisy and relationships between genes highly complex. We present a statistical relational approach to modelling pathways. Our approach allows us to design a coarser and a more fine grained model, based on probabilistic gates. We show that the latter model has

Table III
EXPERIMENT 3: PROPOSED PARENTS PER GATE. NOTICE THAT $\oplus+$ REPRESENTS THE POSITIVE PARENT, AND $\oplus-$ THE NEGATIVE PARENT.

Gene	$\wedge$	$\vee$	$\oplus+$	$\oplus-$
Msn2	22	4	2	
Msn4	16		23	
Hog1	9	7	4	18
Sko1	9	14		10
Hot1	29	10	2	4

predictive performance on the time-series data, and recovers important relationships in the data.

We plan to continue improving the model quality and experiment with new data. Specifically, we would like to experiment with implementing a regression based approach, as it fits our framework naturally. In addition, we would like to experiment with different pathways and with proteomic data. Last, but not least, we would like to investigate how to reduce the number of parameters in the model by exploiting strong correlations between gene expression.

ACKNOWLEDGMENT

This work was funded by National Funds through the FCT - Fundação para a Ciência e Tecnologia (Portuguese Foundation for Science and Technology) within project ADE: PTDC/EIA-EIA/121686/2010 (FCOMP-01-0124-FEDER-020575), project ABLe: PTDC/EEI-SII/2094/2012 (FCOMP-01-0124-FEDER-029010) and by the US 760 Department of Energy (DOE) Great Lakes Bioenergy Research Center (DOE BER 761 Office of Science DE-FC02-07ER64494).

REFERENCES

[1] L. Glass and S. Kauffman, “A logical analysis of continuous, non-linear biochemical control networks,” *Journal of Theoretical Biology*, vol. 39, pp. 103–129, 1973.

[2] R. Thomas, “Boolean formalization of genetic control circuits,” *Journal of Theoretical Biology*, vol. 42, pp. 563–585, 1973.

[3] N. Friedman, M. Linial, I. Nachman, and D. Pe’er, “Using Bayesian networks to analyze expression data,” *Journal of Computational Biology*, vol. 7, no. 3/4, pp. 601–620, 2000.

[4] L. D. Raedt, A. Kimmig, and H. Toivonen, “Problog: A probabilistic prolog and its application in link discovery,” in *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, M. M. Veloso, Ed., 2007, pp. 2462–2467.

[5] A. Kimmig, V. Santos Costa, R. Rocha, B. Demoen, and L. D. Raedt, “On the Implementation of the Probabilistic Logic Programming Language ProbLog,” *Theory and Practice of Logic Programming Systems*, vol. 11, pp. 235–262, 2011.

[6] B. Gutmann, A. Kimmig, K. Kersting, and L. D. Raedt, “Parameter learning in probabilistic databases: A least squares approach,” in *ECML/PKDD–08*, vol. LNCS 5211. Antwerp, Belgium: Springer, September 15–19 2008, pp. 473–488.

[7] A. Capaldi, T. Kaplan, Y. Liu, N. Habib, A. Regev, N. Friedman, and E. O’Shea, “Structure and function of a transcriptional network activated by the mapk hog1,” *Nature Genetics*, vol. 40, pp. 1300–1306, 2008.

[8] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano, “Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions,” in *Proc. the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998, pp. 695–702.

[9] T. Ideker, V. Thorsson, and R. Karp, “Discovery of regulatory interactions through perturbation: Inference and experimental design,” in *Pacific Symposium on Biocomputing*, 2000, pp. 302–313.

[10] A. Tanay and R. Shamir, “Computational expansion of genetic networks,” in *Bioinformatics*, vol. 17, 2001. [Online]. Available: citeseer.ist.psu.edu/tanay01computational.html

[11] L. Chrisman, P. Langley, S. Bay, and A. Pohorille, “Incorporating biological knowledge into evaluation of causal regulatory hypotheses,” in *Pacific Symposium on Biocomputing (PSB)*, January 2003. [Online]. Available: citeseer.ist.psu.edu/568177.html

[12] I. Ong, J. Glasner, and D. Page, “Modelling regulatory pathways in *Escherichia coli* from time series expression profiles,” *Bioinformatics*, vol. 18, pp. S241–S248, 2002.

[13] D. De Maeyer, J. Renkens, L. Cloots, L. De Raedt, and K. Marchal, “Phenetic: network-based interpretation of unstructured gene lists in e. coli,” *Mol. BioSyst.*, vol. 9, pp. 1594–1603, 2013.

[14] V. Lee, S. Topper, S. Hubler, J. Hose, C. Wenger, J. Coon, and A. Gasch, “A dynamic model of proteome changes reveals new roles for transcript alteration in yeast,” *Molecular Systems Biology*, vol. 7, no. 514, 2011.

[15] D. B. Berry and A. P. Gasch, “Stress-activated genomic expression changes serve a preparative role for impending stress in yeast,” *Molecular Biology of the Cell*, vol. 19, no. 11, pp. 4580–4587, 2008.

[16] A. Grigoriev, “A relationship between gene expression and protein interactions on the proteome scale: analysis of the bacteriophage t7 and the yeast saccharomyces cerevisiae,” *Nucleic Acids Research*, vol. 29, no. 17, pp. 3513–3519, 2001.

[17] V. Santos Costa, L. Damas, and R. Rocha, “The yap prolog system,” *Theory and Practice of Logic Programming*, vol. 12, no. Special Issue 1-2, pp. 5–34, 2012. [Online]. Available: http://dx.doi.org/10.1017/S1471068411000512

[18] B. Yang, R. E. Bryant, D. R. O’Hallaron, A. Biere, O. Coudert, G. Janssen, R. K. Ranjan, and F. Somenzi, “A performance study of bdd-based model checking,” in *FMCAD ’98*, vol. LNCS 1522, 1998, pp. 255–289.

[19] S. M. ORourke and I. Herskowitz, “The hog1 mapk prevents cross talk between the hog and pheromone response mapk pathways in saccharomyces cerevisiae,” *Genes & Development*, vol. 12, no. 18, pp. 2874–2886, 1998.