Inferring Regulatory Networks from Time Series Expression Data and Relational Data via Inductive Logic Programming

Irene M. Ong[1,2], Scott E. Topper[3], David Page[2,1], Vítor Santos Costa[4]

[1] Department of Computer Sciences,
[2] Department of Biostatistics and Medical Informatics,
[3] Department of Genetics,
University of Wisconsin – Madison, WI 53706 USA
[4] COPPE/Sistemas, UFRJ Centro de Tecnologia, Bloco H-319, Cx. Postal 68511
Rio de Janeiro, Brasil

Abstract. Determining the underlying regulatory mechanism of genetic networks is one of the central challenges of computational biology. Numerous methods have been developed and applied to the important but complex task of reverse engineering regulatory networks from high-throughput gene expression data. However, many challenges remain. In this paper, we are interested in learning rules that will reveal the causal genes for the expression variation from various relational data sources in addition to gene expression data. Following our previous work where we showed that time series gene expression data could potentially uncover causal effects, we describe an application of an inductive logic programming (ILP) system, to the task of identifying important regulatory relationships from discretized time series gene expression data, protein-protein interaction, protein phosphorylation and transcription factor data about the organism. Specifically, we learn rules for predicting gene expression levels at the next time step based on the available relational data and then generalize the learned theory to visualize a pruned network of important interactions. We evaluate and present experimental results on microarray experiments from Gasch *et al* on *Saccharomyces cerevisiae.*

1 Introduction and Motivation

Gaining insight into the underlying regulation of genes within organisms is important not just for understanding the cause of diseases but also for developing treatments. Viruses have been shown to cause cancer by affecting normal regulation in cells, and gaining an understanding of the factors that determine the ability of embryonic stem cells to maintain their self-renewal and pluripotency can significantly advance developmental biology and stem cell research.

For nearly a decade now, DNA microarray technology has enabled the simultaneous measurement of mRNA abundance of genes in an organism under normal conditions or under various treatments or perturbations. However, microarray experiments still have many sources of error: sample preparation, hybridization, scanning, image processing, normalization, etc. Because samples for microarray data are usually obtained by pooling extracts from a population of cells rather than a single cell, in addition to experimental variables and limitations of the technology, the measurements obtained can be noisy. Noisy data inherently makes it more difficult to reverse engineer the underlying regulatory network.

Despite the difficulty of deciphering genetic regulatory networks from microarray data, numerous approaches to the task have been quite successful. Friedman *et al.* [5] 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. Pe'er *et al.* [18] followed up that work by using BNs to learn master regulator sets. Other approaches include Boolean networks (Akutsu *et al.* [1], Ideker *et al.* [11]) and other graphical approaches (Tanay and Shamir [26], Chrisman *et al.* [3]).

The methods above can represent the dependence between interacting genes, but they cannot capture causal relationships. Pe'er *et al.* [19] ingeniously proposed the use of microarray experiments in which specific genes have been deleted (*knockout*) in yeast to obtain causality. The use of perturbations such as gene deletion mutants can allow the BN learning algorithm to learn a directed edge that suggests direct causal influence. This approach of combining observational and interventional data delivered promising results. Unfortunately, a complete library of gene knockouts are not yet available for organisms other than yeast. The advent of small interfering RNA (siRNA) can be used to reduce the expression of a specific gene in organisms other than yeast, however, siRNA does not guarantee complete silencing of the gene. In our previous work [16], we proposed that the analysis of time series gene expression microarray data using Dynamic Bayesian networks (DBNs) could allow us to learn potential causal relationships (Figure 1).

DBN learning can provide more insight into causality than ordinary BNs. An induced arc from gene X_1 to gene X_2 in an ordinary BN simply means that the expression of gene X_1 is a good predictor of the expression of gene X_2 *at the same time* (Figure 2a). While this good prediction may be because expression of gene X_1 influences expression of gene X_2, it could just as easily be because expression of gene X_2 influences expression of gene X_1 or expression of both gene X_1 and gene X_2 are influenced by expression of another gene X_3 (Figure 2b). On the other hand, an induced arc from gene X_1 to gene X_2 in a DBN implies that expression of gene X_1 at one time slice is a consistently good predictor of gene X_2 at *the next time slice*. This good prediction is unlikely to be because expression of gene X_2 influences expression of gene X_1; intuitively, it seems likely to be because expression of gene X_1 influences expression of gene X_2.[5]

[5] An arc in a DBN does not establish causality definitively. Nevertheless, if a learned DBN contains arcs that imply novel potential causal relationships, in some cases biologists can test these novel relationships with additional, more focused (and time-consuming) experiments.

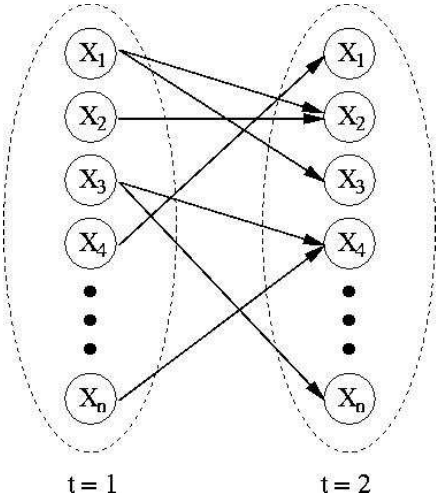

Fig. 1. *Simple DBN model. Labeled circles within a dotted oval represent our variables in one time slice. Formally, arcs connecting variables from one time slice to variables in the next have the same meaning as in a BN, but they intuitively carry a stronger implication of causality. We note that in a DBN with more time slices, the arcs are always the same, e.g., the arc from X_1 at time slice 1 to X_2 at time slice 2 is also present from time slice t to time slice $t + 1$ for all $1 \leq t < T$ where T is the last time slice in the model. This constancy of the arcs is justified by an assumption that the process being modeled is* stationary *though not static. While values of variables may change over time, the manner in which the value of one variable influences the value of a variable at the next time step (i.e., the parents and the conditional probability distribution for the latter variable) will not change.*

While temporal gene expression data contains causal information in the temporal data sequence, the dependence on the appropriate sampling rate, the small sample size, the large number of variables, and the presence of many hidden (signaling and other molecular interactions for which we do not have measurements) variables make it difficult for learning algorithms to completely determine the network.

In this paper, our goal is to utilize the abundant information available from many years of low-throughput as well as recent high-throughput research that are currently available in public databases to infer new relationships that cannot be learned from expression data alone. We are interested in discovering whether ILP is able to infer theories for particular pathways from time series microarray data and use other known relational information about the organism to refine what is already known about that pathway. Specifically, we formulate the learning in the same way as a DBN by learning theories of gene expression that are good predictors of the expression of particular genes at the next time step.

Regulatory sequences control gene expression temporally as well as spatially by *cis*-acting elements and *trans*-acting factors. *Cis*-acting elements are DNA sequences in the vicinity of the target gene, usually within 200 base pairs upstream of the transcription start site. *Trans*-acting factors, bind to the *cis*-acting

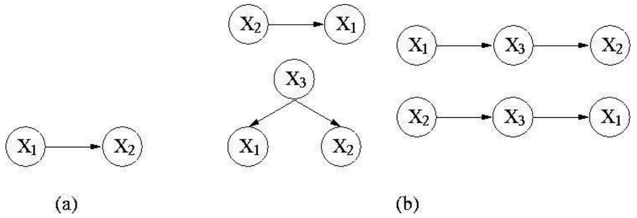

Fig. 2. *(a) X_1 may be a good predictor of X_2, but is X_1* regulating *X_2? (b) Ground truth might be any one of these or a more complicated variant.*

sequences to control gene expression in several ways: the factor may (1) be expressed temporally (specific times in life cycle), (2) be expressed spatially (in a specific location), (3) require modification (phosphorylation), (4) be activated by ligand binding, (5) be sequestered until an environmental signal allows it to interact with the nuclear DNA. Hence, by integrating temporal gene expression data with additional information such as protein-protein interaction, transcription factor and kinase-substrate (phosphorylation) information, we believe we can capture some of these causal relationships and underlying mechanisms.

2 Related Work

Our goal in this paper is similar to that of Tu *et al.* [28]. We are interested in determining whether ILP can learn the pathway links between causal genes and target genes that explain the regulatory relationships between them. In the past few years, we have seen an increase in the use of inductive logic programming (ILP) methods for learning functional genomics [24, 13, 2, 20], metabolic networks [25] and also predicting gene expression levels [17]. Papatheodorou *et al.* [17] used Abductive logic programming (ALP) to learn rules that would explain how gene interactions can cause changes in gene expression levels.

Recently, Fröhler and Kramer [6], applied ILP to the task of predicting up- and down-regulation of gene expression in *S. cerevisiae* under different environmental stress conditions [8] with the use of additional information. Fröhler and Kramer used the data from Middendorf *at al.* [14], where the presence of transcription factor binding sites (pruned list of 354 after removing redundant and rare sites) in the gene's regulatory region and the expression levels of regulators (selected list of 53, 50 of which were top ranking regulators identified by Segal *et al.* [21]) are used to predict gene regulation.

Following Middendorf, Fröhler and Kramer consider 3 classes of gene activity: up-regulation (> 1.2), down-regulation (< -1.2), and no change. The up- and down-regulated genes consist of 5% of all the data points since 95% of the expression were unstimulated. Their results report on discriminating between up- and down-regulation, with excellent results, although the original work from Middendorf's showed that discriminating between the 3 classes is a much harder task. We similarly discretize into 3 classes to reduce noise, but our up- and down-regulated classes are about 20% of the total number of examples, so one would expect the discrimination task in our case to be harder.

Our work differs from that of Fröhler and Kramer in four ways. First, we learn rules to predict the up-regulation of a gene based on the activity and expression of genes from the *previous time step* as in a DBN since we are interested in learning causal relationships from the data. Secondly, we discretize the gene expression data by comparing two consecutive time series measurements under the same experimental condition and determining whether the change in expression was up, down or same based on a threshold of greater than 0.3, less than -0.3, or in between. Thirdly, we use information on transcription factors rather than transcription factor binding sites and we do not restrict the transcription factor or regulator set as our goal is to learn possible new players in the network. Finally, we use Aleph instead of Tilde.

3 ILP and Aleph

Inductive logic programming (ILP) is a popular approach for learning first-order, multi-relational concepts between data instances. ILP uses logic to induce hypotheses from observations (positive and negative examples) and background (prior) knowledge by finding a logical description of the underlying data model that differentiates between the positive and negative examples. The learned description is a set of easily interpretable rules or clauses.

There are many ILP systems available, but we chose to use Aleph [22] because it has been shown to perform well even on fairly large datasets. This is because Aleph implements the Progol algorithm [15], which learns rules from a pruned space of candidate solutions. The Progol algorithm structures and limits the search space in two steps. Initially, it selects a positive instance to serve as the seed example and searches the background knowledge for the facts known to be true about the seed example - the combination of these facts form the example's most specific or saturated clause. Then, Aleph defines the search space to be clauses that generalize a seed example's saturated clause, and performs a general to specific search over this space. The key insight of the Progol algorithm is that some of these facts explain the seed example's classification, thus generalizations of those facts could apply to other examples.

4 Data and Methodology

To test our hypotheses, we use time series gene expression data of environmental stress response experiments, including DNA-damaging agents from Gasch *et al.* [8, 7]. We chose to use this dataset on yeast because yeast is a model organism used for studying many basic cellular processes and there exists many publicly accessible databases containing various sources of data from many years of research. We focused our study on the DNA damage checkpoint pathway because it is an important pathway that has been widely studied. There are about 6500 genes in yeast, 19 of which are considered to be in the "DNA damage checkpoint" pathway based on a recent review by Harrison and Haber [10].

It is well known that a common problem with current microarray data is the small number of sample points and the large number of features or genes. Nevertheless, it is hoped that discretization as well as other sources of information will permit useful results to be obtained. We determined the relative change in expression from one time step to the next by comparing the expression levels between two consecutive time series measurements. The time series data were discretized into one of three possible discrete values by comparing two consecutive time series measurements: if the change increased by 0.3, we consider the expression to be up-regulated, if the change decreased by 0.3, we consider the expression to be down-regulated, otherwise we say the expression stayed the same.

As alluded to earlier, there are many other spatial and molecular interactions that are not captured by expression data. Known transcription factors for specific genes can allow the learning algorithm to focus on specific proteins that are known to interact with the DNA of the target gene. The learning algorithm could also potentially discover combinations of transcription factors (pairs, trios, etc.) required to trigger a change in expression of a particular set of genes. Because transcription factors can also interact with other proteins or metabolites on their way to activating gene expression, background knowledge of proteins that are known to interact with each other can allow for the discovery of novel proteins in the pathway. Furthermore, an estimated 30% of proteins need to be phosphorylated in order to trigger a change in the protein's function, activity, localization and stability [12]. Thus, background knowledge about a large number of protein phosphorylation in yeast was also included [4].

Recent technological advances have produced more high-throughput data that capture different types of interactions. ChIP-chip (chromatin immunoprecipitation, a well-established procedure to investigate interactions between proteins and DNA, coupled with whole-genome DNA microarrays), technology allows one to determine the entire spectrum of in vivo DNA binding sites for any given transcription factor or protein. Mass spectrometry, large-scale two-hybrid screens, single-cell analysis of flow cytometry, and protein microarrays have all been used to generate high-throughput measurements of certain types of molecules such as proteins, metabolites, protein-protein interactions and also signaling events such as phosphorylation within cells. Most of these data are also known to be noisy especially those obtained through high-throughput methods that were conducted in vitro (outside the organism). High-throughput protein-protein interaction and phosphorylation data are especially noisy because the conditions under which the data are collected differs quite significantly from that in a cell, i.e. detecting interactions that would not actually occur in vivo (inside the organism) or missing interactions that actually take place.

We aim to link known interactions with gene expression activity to possibly learn new mechanisms. We do this by associating the up- or down-regulation of specific genes from the previous time step with its transcription factor, a protein it might interact with, or a phosphorylation event. We assume that an event in the previous time step will contribute to the change in expression at the current

Fold	0	1	2	3	4	5	6	7	8	9	Average across all folds
Accuracy	0.73	0.87	0.81	0.72	0.83	0.84	0.73	0.79	0.75	0.78	0.79

Table 1. Cross validation accuracies

time. This assumption does not necessarily hold for all biological activity but a similar assumption, that of using a gene's expression level to approximate the activity of other genes within the same pathway, have been used by others [29].

The MIPS Comprehensive Yeast Genome Database (CYGD) [9] provided much of the information regarding yeast genes, their function, location, phenotype and disruption. We obtained protein-protein interaction data from BioGRID [23], transcription factor data from the YEASTRACT database [27], and over 4000 yeast phosphorylation events from Ptacek *et al.* [4]. The ILP system, Aleph [22], was used to learn rules from the data.

We first learn rules using inductive logic programming (ILP) to predict the discretized gene expression level at the *next time step* as in a DBN. Then we use the learned theory to generate a pruned network or graph that show interactions corresponding to proofs for the rules.

5 Experiments and Results

We performed ten-fold cross validation experiments to learn theories for predicting held-out gene expression values for genes in the DNA damage checkpoint pathway at the *next time step*. The discretized microarray experiments were divided into ten folds, grouping replicate experiments together to avoid bias, based on the different experimental conditions.

We obtained an accuracy of 79% on predicting up-regulated examples averaged over ten folds of the cross-validation procedure (see Table 1).

Examples of some of the rules learned across the folds are:

Rule 1 up(GeneA,Time,Expt) :-
previous(Time,Time1), down(GeneA,Time1,Expt), interaction(tof1,GeneA),
up(tof1,Time1,Expt), function(GeneA,'CELL CYCLE AND DNA PROCESSING:cell cycle:mitotic cell cycle and cell cycle control:cell cycle arrest').

Rule 2 up(GeneA,Time,Expt) :-
previous(Time,Time1), down(GeneA,Time1,Expt),
phosphorylates(GeneA,GeneE), up(GeneE,Time1,Expt),
transcriptionfactor(GeneF,GeneE), down(GeneF,Time1,Expt),
transcriptionfactor(GeneF,cdc20), down(cdc20,Time1,Expt).

Rule 3 up(GeneA,Time,Expt) :-
previous(Time,Time1), down(GeneA,Time1,Expt),
interaction(GeneE,GeneA), down(GeneE,Time1,Expt),

interaction(GeneE,mms4), down(mms4,Time1,Expt),
function(GeneA,'METABOLISM').

These rules all specify the activity of specific genes involved in the larger DNA damage pathway. Tof1 is a subunit of a replication-pausing checkpoint complex (Tof1p-Mrc1p-Csm3p) that acts at the stalled replication fork to promote sister chromatid cohesion after DNA damage, facilitating gap repair of damaged DNA. Cdc20, which is regulated by cell-cycle genes, is an activator of anaphase-promoting complex/cyclosome (APC/C), which is required for metaphase/anaphase transition. It is part of the DNA damage checkpoint pathway and directs ubiquitination of mitotic cyclins, Pds1p, and other anaphase inhibitors. Finally, Mms4 is a subunit of the structure-specific Mms4p-Mus81p endonuclease that cleaves branched DNA and is involved in recombination and DNA repair.

The learned rules prove examples, and proofs generate paths between genes, so using the theories in all the folds, we further generated graphs. The graphs only show links that can be used in proofs for at least 5 examples (train+test). The width of a line in the graph is an indication of the proportion of examples used in the proof. Note that the graph only displays literals that were used in successful proofs. Hence, paths in the graph correspond to proofs and the nodes are examples of literals which were used to prove the rules. The learned graph of interactions amongst the 19 genes in the DNA damage checkpoint pathway are shown in Figure 3. A more detailed graph showing interactions amongst the genes in the DNA damage checkpoint pathway as well as transcription factors and phosphorylators can be seen in Figure 4.

6 Discussion

The DNA damage checkpoint monitors genome integrity, and ensures that damage is corrected before cell division occurs. When DNA damage is detected, the checkpoint network transmits signals that stall the progression of the cell cycle and mobilize repair mechanisms. The graph resulting from our analysis recapitulates many of the central aspects of this signaling network, and connects that network temporally to the normal progression through the cell cycle.

DNA damage (often in the form of a double strand break) is first recognized by MRX, a protein complex consisting of Mre11, Rad50 and Xrs2. These proteins are shown to interact together slightly to the left of the middle of Figure 4, with Mre11 linked to both Xrs2 and Rad50. The MRX complex coordinates the restructuring of the damaged region. MRX stimulates the phosphorylation of histones, H2A, in the region adjacent to the DNA double strand break (via Tel1) and recruits an exonuclease to generate a stretch of single stranded DNA. Our graph does not include physical interactions between Tel1 and the MRX complex, however both are connected through Mec1 and through the DNA binding protein Rap1. Rap1 can act as an inducer or a repressor, and is active in many disparate elements of cell biology, including ribosome synthesis and telomere preservation.

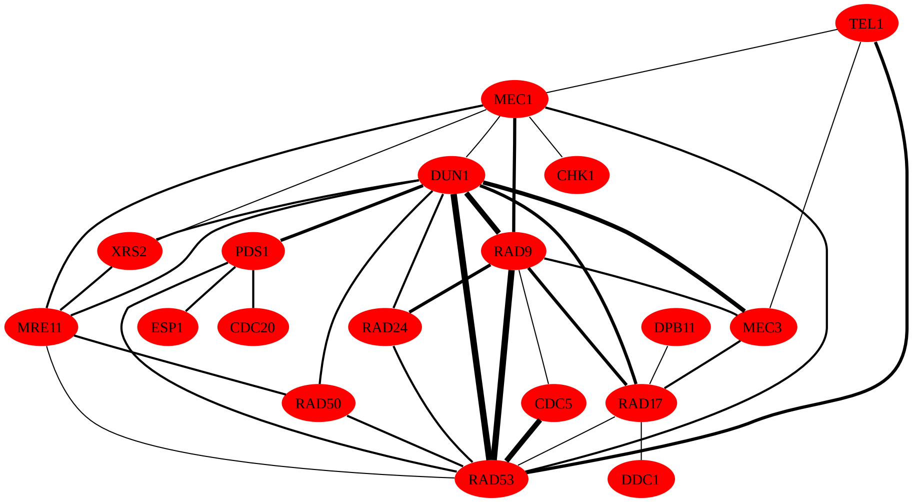

Fig. 3. Learned graph of interactions from successful proofs amongst the 19 genes in DNA damage checkpoint pathway from Harrison and Haber [10]. Straight edges represent protein-protein interactions. The width of a line in the graph is an indication of the number of examples that used this interaction in a proof.

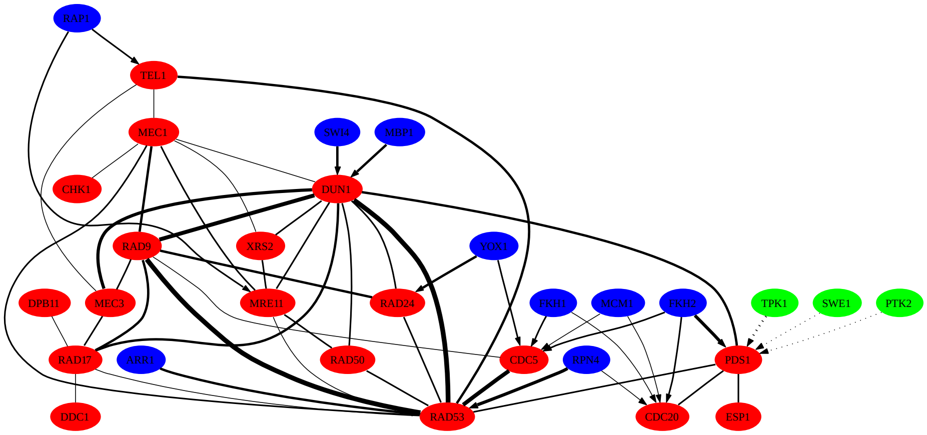

Fig. 4. Learned graph of interactions from successful proofs for the DNA damage checkpoint pathway. Red nodes indicate one of the 19 genes in DNA damage checkpoint pathway from Harrison and Haber [10], blue nodes indicate transcription factors and green nodes kinases. Straight edges represent protein-protein interactions, solid lined arrows represent transcription factor to target gene interaction, and dotted arcs represent kinase to substrate phosphorylation. The width of a line in the graph is an indication of the number of examples that used this interaction in a proof. A larger figure can be found at: http://www.biostat.wisc.edu/˜ong/new2a.ps

Once single stranded DNA is generated, it is bound by the heterotrimer replication protein A (RPA) and two things occur. First, Mec1/Ddc2 binds and activates the signaling cascade. Mec1 phosphorylates Rad9 (shown as physical interaction in the graph), which in turn recruits Rad53. Ddc2 is conspicuously absent from this graph due to the requirement that only links that can be used in proofs for a certain number of examples are displayed. Next, the 9-1-1 clamp, which consists of three proteins Rad17, Mec3 and Ddc1, binds and demarcates the ssDNA/dsDNA junction, and facilitates some of the interactions described above. The 9-1-1 clamp components are grouped at the left side of the graph, linked by protein-protein interactions.

At the heart of the signaling network is Rad53, a well-connected, essential yeast kinase. Rad53 phosphorylates Dun1, a kinase whose activity ultimately controls much of the transcriptional response to DNA damage. Dun1 is also a very central protein in this network, demonstrating interactions with the 9-1-1 clamp, Rad24, the MRX complex, Rad53 and Pds1, a cell cycle control gene. Finally, Rad53 signals cell cycle arrest through Pds1 (via Cdc20), and Cdc5. Pds1 governs entry into mitosis, and Cdc5 controls exit from mitosis. All of these interactions are present in our results.

DNA damage is an inevitable consequence of DNA synthesis, and the graph reveals that the expression of the gene responsible for signaling the induction of DNA repair genes (Dun1) is coordinated by two transcription factors (Swi4 and Mbp1) that are active in the period just before DNA synthesis begins. Likewise, the transcription factors Mcm1, Fhk1 and Fkh2 are known to control the transition from G2 to mitosis, and in our graph these TFs are linked to Cdc5, Cdc20 and Pds1, which govern this transition.

At a broader level, the results shown in Figure 4 illustrates the centrality of Rad9, Rad53 and Dun1. These genes are instrumental in coordinating the various aspects of this response: detection of damage, cell cycle arrest, and mobilization of repair mechanisms.

7 Conclusions and Future Work

As a first step, we concentrated our experiments on learning the DNA damage checkpoint pathway because it is a very important pathway that have been implicated in cancer and aging, and because it has been very well studied. This pathway plays an important role by responding to single and double-stranded DNA breaks, and is therefore often activated in stressful environments. Hence, it involves a lot of signaling kinases that phosphorylates proteins that are already present within the cell or that only require molecular amounts to trigger a response.

After performing our analysis, we found that the phosphorylation dataset from Ptacek *et al.* [4] did not specifically include any phosphorylation relationships for the kinase and substrates in the DNA damage checkpoint pathway. The results we obtained show that our method is quite good at learning important pathway interactions and regulators despite the fact that the data may be noisy or incomplete. This further emphasizes the utility of integrating different data types, since many potential interactions, including those that were not evident from single data sources were identified.

A possible next step will be to perform a comparison with DBNs. We could also explore the larger network of genes that are connected with the core DNA damage checkpoint genes by including more specific background knowledge. This set is likely to include known targets of Dun1 activation, and genes that coordinate the biological processes involved in cell division. It may also include genes heretofore un-implicated in this process, and may provide good starting points for future wet lab experimentation.

In the future, we also plan to study other pathways and organisms, incorporate other sources of relational data including knockout data, and integrate these networks with probabilistic models.

8 Acknowledgments

We gratefully thank Audrey Gasch for useful discussions. Irene Ong is supported by the BACTER institute (DOE-GTL award DE-FG02-04ER25627), support for Scott Topper was provided by the Predoctoral Training Program in Genetics, 5 T32 GM07133 and Vítor Santos Costa is supported by CNPq.

References

1. 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*, pages 695–702, 1998.
2. C.H. Bryant, S.H. Muggleton, S.G. Oliver, D.B. Kell, P.G.K. Reiser, and R.D. King. Combining inductive logic programming, active learning, and robotics to discover the function of genes. *Electronic Transactions in Artificial Intelligence*, 6:1–36, 2001.
3. 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.
4. J. Ptacek et al. Global analysis of protein phosphorylation in yeast. *Nature*, 438:679–684, 2005.
5. N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3/4):601–620, 2000.
6. S. Fröhler and S. Kramer. Inductive logic programming for gene regulation prediction. In *Proceedings of the 16th International Conference on Inductive Logic Programming*, pages 83–85, Santiago de Compostela, Spain, 2006. University of Corunna.
7. A.P. Gasch, M. Huang, S. Metzner, D. Botstein, S.J. Elledge, and P.O. Brown. Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog Mec1p. *Mol Biol Cell.*, 12:2987–3003, 2001.
8. A.P. Gasch, P.T. Spellman, C.M. Kao, O. Carmel-Harel, M.B. Eisen, G. Storz, D. Botstein, and P.O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Mol Biol Cell.*, 11:4241–57, 2000.

9. U. Güldener, M. Münsterkötter, G. Kastenmüller, N. Strack, J. van Helden, C. Lemer, J. Richelles, S.J. Wodak, J. Garcia-Martinez, J.E. Perez-Ortin, H. Michael, A. Kaps, E. Talla, B. Dujon, B. Andre, J.L. Souciet, J. De Montigny, E. Bon, C. Gaillardin, and H.W. Mewes. CYGD: the Comprehensive Yeast Genome Database. *Nucleic Acids Research*, 33:D364–8, 2005.
10. J.C. Harrison and J.E. Haber. Surviving the breakup: The DNA damage checkpoint. *Annu. Rev. Genet.*, 40:209–235, 2006.
11. T.E. Ideker, V. Thorsson, and R.M. Karp. Discovery of regulatory interactions through perturbation: Inference and experimental design. In *Pacific Symposium on Biocomputing*, pages 302–313, 2000.
12. Ptacek J. and Snyder M. Charging it up: global analysis of protein phosphorylation. *Trends in Genetics*, 22:545–554, 2006.
13. R.D. King, K.E. Whelan, F.M. Jones, P.J.K. Reiser, C.H. Bryant, S. Muggleton, D.B. Kell, and S Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427:247–252, 2004.
14. M. Middendorf, A. Kundaje, C. Wiggins, Y. Freund, and C. Leslie. Predicting genetic regulatory response using classification. *Bioinformatics*, 20:i232–240, 2004.
15. S. Muggleton. Inverse entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4):245–286, 1995.
16. I.M. Ong, J.D. Glasner, and D. Page. Modelling regulatory pathways in *Escherichia coli* from time series expression profiles. *Bioinformatics*, 18:S241–S248, 2002.
17. I. Papatheodorou, A. Kakas, and M. Sergot. Inference of gene relations from microarray data by abduction. In *Eighth International Conference on Logic Programming and Non-Monotonic Reasoning*. Springer, 2005.
18. D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. In *Proceedings of the 9th International Conference on Intelligent Systems for Molecular Biology*, pages S215–S224. Oxford University Press, 2001.
19. D. Pe'er, A. Regev, and A. Tanay. Minreg: Inferring an active regulator set. In *Proceedings of the 10th International Conference on Intelligent Systems for Molecular Biology*, pages S258–S267. Oxford University Press, 2002.
20. P.G.K. Reiser, R.D. King, D.B. Kell, S.H. Muggleton, C.H. Bryant, and S.G Oliver. Developing a logical model of yeast metabolism. *Electronic Transactions in Artificial Intelligence*, 5:223–244, 2001.
21. E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman. Module networks: identifying regulatory modules and their condition specific regulators from gene expression data. *Nature Genetics*, 34:166–176, 2003.
22. Ashwin Srinivasan. *The Aleph Manual.* University of Oxford, 2001.
23. C. Stark, B.J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. BioGRID: a general repository for interaction datasets. *Nucleic Acids Research*, 34:D535–539, 2006.
24. J. Struyf, S. Dzeroski, H. Blockeel, and A. Clare. Hierarchical multi-classification with predictive clustering trees in functional genomics. In *Progress in Artificial Intelligence: 12th Portugese Conference on Artificial Intelligence*, pages 272–283. Springer, 2005.
25. A. Tamaddoni-Nezhad, R. Chaleil, A. Kakas, and S.H. Muggleton. Application of abductive ILP to learning metabolic network inhibition from temporal data. *Machine Learning*, 64:209–230, 2006.
26. A. Tanay and R. Shamir. Computational expansion of genetic networks. In *Bioinformatics*, volume 17, 2001.

27. M.C. Teixeira, P. Monteiro, P. Jain, S. Tenreiro, A.R. Fernandes, N.P. Mira, M. Alenquer, A.T. Freitas, A.L. Oliveira, and I. S-Correia. The YEASTRACT database: a tool for the analysis of transcription regulatory associations in *Saccharomyces cerevisiae*. *Nucleic Acids Research*, 34:D446–451, 2006.
28. Z. Tu, Li. Wang, M.N. Arbeitman, T. Chen, and F. Sun. An integrative approach for causal gene identification and gene regulatory pathway inference. *Bioinformatics*, pages e489–e496, 2006.
29. A. Zien, R. Kuffner, R. Zimmer, and T. Lengauer. Analysis of gene expression data with pathway scores. *Proc Int Conf Intell Syst Mol Biol*, 8:407–17, 2000.