Structure learning for biomolecular pathways containing cycles

Solomon Itani, Karen Sachs, Garry P. Nolan, Munther A. Dahleh

Abstract— **Bayesian network structure learning is a useful tool for elucidation of regulatory structures of biomolecular pathways. The approach however is limited by its acyclicity constraint, a problematic one in the cycle-containing biological domain. Here, we introduce a novel method for modeling cyclic pathways in biology, by employing our newly introduced Generalized Bayesian Networks (GBNs) and proposing a structure learning algorithm suitable for the biological domain. This algorithm relies on data and perturbations which are feasible for collection in an experimental setting, such as perturbations affecting either the abundance or activity of a molecule. We present theoretical arguments as well as structure learning results from simulated data. We also present results from a small real world dataset, involving genes from the galactose system in S. cerevisiae.**

I. INTRODUCTION

Since the seminal work by Pe'er [2], Bayesian networks have been used extensively in biology, to model regulatory pathways both in the genetic ([15], [2]) and in the signaling pathway domain ([4], [16]). Bayesian network models encode probabilistic relationships among random variables in a domain, providing a framework for tasks such as *s*tructure learning. In a biological setting, the random variables represented are biologically important entities such as genes, small molecules and activated or phosphorylated proteins. The structure learning task consists of searching the space of possible structures to find the one that bests reflects probabilistic relationships in a biological dataset. Under appropriate conditions and assumptions, the framework of causation can be employed to enable a causal interpretation to these models, indicating that the parent of a variable in the learned graph causally influences the variable's quantity (either directly or indirectly) [13], [10]. Within the framework of causation, structure learning can be used to elucidate the structure of interactions in regulatory pathways.

In spite of their usefulness, Bayesian network models are limited in their applicability in this domain because they are constrained to be acyclic, while positive and negative feedback loops abound in biological pathways. In particular, Bayesian network structure learning will *a*lways yield an inaccurate structure for any cycle containing pathway and, as a result, will fail in its predictive capacity (at minimum) for variables downstream of an incorrectly directed edge. When time course data are available, it is feasible to represent cycles by unrolling them in time, using a Dynamic Bayesian networks (DBNs), or Continuous Time Bayesian networks (CTBNs) [9], [11]. However, DBNs suffer from various computational and data related challenges and, in some domains, the requisite timecourse data are not feasibly attainable in an applicable form. Therefore, it would be useful to find an approach for learning cyclic structures from static 'snapshot' data, collected at a single timepoint from a dynamic system.

We have recently developed a formalism for representing cyclic structures using Generalized Bayesian networks (GBNs), a form of Bayesian networks that we have generalized to encompass cycles [6]. This formalism enables structure learning in a cyclic domain, relying on perturbations which break the cyclic structure. Far from requiring an exhaustive set of perturbations, this algorithm is designed to minimize the amount of interventional data needed, requiring as few as merely one perturbation per cycle for accurate structure learning. However, the generalized algorithm is not necessarily applicable to biology, due in part to the nature of the necessary perturbations, which are often not feasible in biological systems. Here, we present the first ever application of GBNs in a biological context. For this application, we modify the structure learning algorithm to bring it incrementally closer to applicability in a biological domain, by generalizing the algorithm to enable structure learning with biologically feasible data. We present the algorithm as well as results from two simulated synthetic networks containing cycles, and one real world genetic feedback system from the galactose metabolism system in yeast.

A. Bayesian networks and Bayesian network structure learning

Bayesian networks [12], represent probabilistic dependence relationships among multiple interacting components [3], [2], [5]. In our context, the components are biomolecules such as genes or signaling pathway proteins. Bayesian network models illustrate the effects of pathway components upon each other (that is, the dependence of each biomolecule in the pathway on other biomolecules) in the form of an influence diagram- a graph (G), and a joint probability distribution. In the graph, the nodes represent variables (the biomolecules) and the edges represent dependencies (or more precisely, the lack of edges indicate a conditional independency) [12]. For each variable, a conditional probability distribution (*CPD*) quantitatively describes the form and magnitude of its dependence on its parent(s). The graph must be a *DAG*- a directed acyclic graph. By *directed* we mean that the edges must be single-headed arrows, originating from one node (the *parent* node) and ending in another (called the *child* node). *Acyclic* indicates that the graph must not include directed cycles, so it should not be possible to follow a path from any node back to itself.

This work was supported by a Leukemia and Lymphoma Society grant and an LLS post doctoral fellowship to K.S.

These models can be automatically derived from experimental data through a statistically founded computational procedure termed network inference or *structure learning*. The Bayesian network inference algorithm aims to discern a model that is as close as possible to the observations made. The algorithm finds the most likely models by traversing the space of possibilities, via single arc changes that improve the model score, a score which expresses the probability of the model structure being considered, given the dataset at hand. There is a trade-off between simple models and those that accurately capture the empirical distribution observed in the data. The employed Bayesian scoring metric captures this trade-off, thus a high scoring model is a both simple and accurate representation of the data[17].

In a Bayesian network, the graph G encodes the *Markov Assumptions*: Each variable X_i is independent of its nondescendants, given its parents in G.

$$\forall X_i X_i \perp \text{Nondesc} X_i \mid \text{Par} X_i$$

As a consequence of the Markov assumption, the joint probability distribution over the variables represented by the Bayesian network can be factored into a product over variables, where each term is local conditional probability distribution of that variable, conditioned on its parent variables:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|\text{Par}X_i) \qquad (1)$$

This is called the chain rule for Bayesian networks, and it follows directly from the chain rule of probabilities, which states that the joint probability of independent entities is the product of their individual probabilities.

It is this formulation for the joint probability that breaks down in a cyclic domain, and thus necessitates an acyclicity assumption. In our formalism, we present a factorization that applies in the cyclic domain, and, in the context of this framework, we present a cycle-enabling structure learning algorithm.

B. Generalized Bayesian Networks

In this section we present the general model, using formal definitions and remarks. We start with a formal characterization of Bayesian networks that is consistent with our notation in the rest of the paper. Consider a generic problem with M random variables. We would like to define a probability space $(\Omega, \mathscr{F}, \mathbb{P})$ in which to study such a problem. We first restrict ourselves to random variables taking values from finite alphabets. To further simplify the exposition, we use a universal alphabet that we denote by $\mathcal{X}$ and let $L = |\mathcal{X}|$. Therefore, we may choose the outcome space Ω to be the set of all possible configurations of the random variables at hand, i.e. $\Omega = \mathcal{X}^M$. Our probability space then becomes $(\Omega, 2^\Omega, \mathbb{P})$, for a suitable choice of probability measure $\mathbb{P}$, which we can think of as the joint mass function of all random variables. We are interested in different descriptions and characterizations of this measure. We start with some traditional definitions, and proceed with new ones.

We start by introducing some of the terminology used here. For a directed graph $G = (V, E)$, the *parents* of node i are defined as $\pi_i := \{j \in V : (j, i) \in E\}$. A stochastic map $g : \mathcal{X} \to \mathbb{R}_+$ is a nonnegative real-valued map such that $\sum_{\mathcal{X}} g = 1$. A stochastic kernel $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$ is a nonnegative real-valued map such that every restriction of the second argument results in a stochastic map. For finite $\mathcal{X}$ and $\mathcal{Y}$ a stochastic map g is a probability vector, and a stochastic kernel f is a stochastic matrix.

Definition 1: Bayesian Network

A *Bayesian network* is a pair (G, F), where:

- G is a directed acyclic graph $G = (V, E)$ and
- F is a set of stochastic kernels $f_i : \mathcal{X} \times \mathcal{X}^{|\pi_i|} \to \mathbb{R}_+$ indexed by all nodes $i \in V$.

The *probability space induced by a Bayesian network* is an instance of the generic problem, obtained by letting $M = |V|$, associating to every node $i \in V$ a random variable X_i, and defining the following probability measure (which can be easily verified to be valid):

$$\mathbb{P}(X_1 = x_1, \cdots, X_M = x_M) = \prod_{i \in V} f_i(x_i; x_{\pi_i}). \qquad (2)$$

Note that when $\pi_i = \emptyset$, the corresponding stochastic kernel is essentially a stochastic map $f_i : \mathcal{X} \to \mathbb{R}_+$. We use the terminology 'node' to refer, based on context, to both the graph node and the associated random variable.

Definition 2 (Causal Bayesian Network): A *causal Bayesian network* is a Bayesian network in which a given edge from node i to node j means that i causally affects j. Thus in addition to all of the regular probabilistic relations of BN's, a causal Bayesian network incorporates causality, which can be understood by way of possible interventions on the nodes of the graph. Classically, an intervention at node i forces node i to a certain value, thus making it independent of its parents but keeping the dependence between it and its children. These interventions are interventions on the abundance of the node, or its amount. Another type of interventions, which we model differently, is where the *activity* of the node is affected. This type of interventions remove the dependencies between the node and its children, but still keeps its dependence on its parents.

Both of these types of interventions can be studied in the probability framework by endowing the a Bayesian Network with a family of joint distributions, each corresponding to a different intervention (in addition to a joint for the no-intervention case.)

Consider a causal Bayesian network and the two joints corresponding to scenarios that are the same except for an *abundance* intervention at node i. The only difference between those joints (described as in equation (2) above) will be in f_i, where f_i in the case of the intervention on i is independent of i's parents and dictated directly by the intervention. For the case of *activity* intervention at i being the difference, f_j will chance for all nodes j that are children of i. So $f_j(x_j; x_{\pi_j})$ will be replaced by $f_j(x_j; x_{\pi_j \setminus i}, i_0)$, where i_0 is the value that i's activity is forced to be by the intervention. The kernel corresponding to i itself will not change.

We take a deeper look at the activity interventions, since they are as studied in the literature. In what follows we use a partial configuration of the nodes of a Bayesian network which we represent by a pair (I, ξ), where $I \subset V$ and $\xi \in \mathcal{X}^{|I|}$ is an I-indexed tuple, interpreted as specifying the values of the nodes of index I.

We think of (I, ξ) as an activity intervention on nodes I to the value specified by ξ. Note that each perturbation supersedes the node variable itself, and may be interpreted as what is 'seen' by all descendants of that node.

We can make this decoupling more explicit in an alternative characterization. Indeed, given a causal Bayesian network (G, F) describing an M-variable generic problem, and a specific activity intervention (I, ξ), we can construct a regular Bayesian network (G', F') which describes a $2M$-variable generic problem, such that the restriction to M of the variables has a joint distribution evaluating to $\mathbb{P}_{(I,\xi)}$.

The construction of G' does not depend on the perturbation. Let $G' = (V', E')$, where we duplicate each original node: $V' = \{i_\mu : i \in V\} \cup \{i_\sigma : i \in V\}$, and construct edges $E' = \{(i_\sigma, j_\mu) : (i, j) \in E\} \cup \{(i_\mu, i_\sigma) : i \in V\}$ that make G' bipartite, between the σ (*seen*) and μ (*measured*) copies. The $\sigma \to \mu$ edges preserve the parent-child structure, and the $\mu \to \sigma$ edges exist to relate the two under different perturbation settings.

As for F', we populate it such that for all $i \in V$ we have $f_{i_\mu}(x_{i_\mu}; x_{\pi_{i_\mu}}) = f_i(x_{i_\mu}; x_{\pi_{i_\mu}})$ regardless of the perturbation, and:

$$f_{i_\sigma}(x_{i_\sigma}; x_{i_\mu}) = \begin{cases} \eta_{x_{i_\mu}}(x_{i_\sigma}) & \text{if } i \notin I \\ \eta_{\xi_i}(x_{i_\sigma}) & \text{if } i \in I \end{cases} .$$

We call this characterization the $\sigma - \mu$ *characterization* of a causal Bayesian network.

Definition 3 (Generalized Bayesian Network): A *Generalized Bayesian network* is a Bayesian network where we allow G to have directed cycles. A cyclic Bayesian network induces a probability space, with the same outcome space as in the Bayesian network, and with a probability measure which we characterize locally, such that for every $A \subset V$ it obeys the following property:

$$\mathbb{P}(X_{A \cup \pi_A} = x_{A \cup \pi_A}) = \mathbb{P}(X_{\pi_A} = x_{\pi_A}) \prod_{i \in A \setminus \pi_A} f_i(x_i; x_{\pi_i}). \tag{3}$$

In particular, for every i such a probability measure obeys the following property:

$$\mathbb{P}(X_i = x_i, X_{\pi_i} = x_{\pi_i}) = \mathbb{P}(X_{\pi_i} = x_{\pi_i}) f_i(x_i; x_{\pi_i}). \tag{4}$$

Definition 4 (Causal Generalized Bayesian Network): A *causal Generalized Bayesian network* is a generalized Bayesian network with which we associate a collection of joint probability distributions. These joints relate to each other as they do in causal Bayesian networks. Thus for abundance interventions, intervened nodes become independent of their parents and under activity inhibitions they become independent of their children.

Finally, we note that the $\sigma - \mu$ characterization which we gave for acyclic causal Bayesian networks extends here in a straightforward fashion. It is easy to see that this description is compatible with our previous description of GBN's.

II. STRUCTURE LEARNING FOR CYCLIC NETWORKS

In [6], we use the formalism presented above to formulate a structure learning algorithm for cyclic domains. The algorithm relies on the $\sigma - \mu$ characterization, and assumes that the available interventions affect the activity but not the quantity of the targeted biomolecule. This is sometimes the case in biology- for instance, with small molecule kinase inhibitors- but it is often not the case (e.g. gene knockouts). Below, we present a structure learning algorithm modified to enable the use of either *activity* or *abundance* perturbations; perturbations that affect either the activity of a molecule (but not its quantity), or that affect only its quantity, respectively. This incremental improvement renders the algorithm applicable to real-life biological datasets including various feasible forms of perturbations.

Consider a CGBN from which we can sample both observational and experimental data, from an intervention set I and its subsets. Assume that I is 'rich', in the sense that it has at least one representative node with an activity intervention or two nodes with abundance intervention from every cycle in the underlying graph. The following algorithm effectively guides the experimental procedure (or uses previously collected data) and recovers the CGBN's structure. In what follows, we elaborate the subroutines that are used, and show correctness.

Algorithm: Learn CGBN structure

```
0: Start with a CGBN and an intervention set $I$.
1: [Probing experiments] Collect sets of i.i.d. samples under no-
   intervention and single-intervention data, i.e. when node $i$ is
   intervened at, for each $i$ in $I$.
2: Call subroutine 'detect descendants' to recover descendant in-
   formation for all nodes in $I$.
3: Identify the subset of all nodes in $I$ which are in cycles, and
   denote it by $I_C$.
4: [Cycle-breaking experiment] Collect i.i.d. samples when all
   nodes in $I_C$ are intervened at.
5: Recover an embedded DAG.
6: [Leave-one-out experiments] Collect sets of i.i.d. samples when
   nodes in $I_C \setminus \{i\}$ are intervened at, for each $i \in I_C$.
7: Call subroutine 'detect children' to recover child information for
   all nodes in $I_C$.
8: Recover all missing edges in the DAG, and complete the DCG
   structure of the CGBN.
```

The following is the subroutine that obtains descendant information based on no-intervention and single-intervention i.i.d. data. The correctness of the subroutine is similar to the one in [6]. The choice of distance is not critical, and thresholding can be automated.

Subroutine: Detect descendants

```
Start with sets of $n$ i.i.d. samples generated by a CGBN, under no interventions as well as single-interventions at each $i$ in $I$. Initialize a binary $|V| \times |I|$ descendant information matrix.
For each $j \in V$:
    Compute $\hat{\mathbb{P}}^n(X_j)$, the empirical marginal of $X_j$ under no interventions.
    For each $i \in I$:
        Compute $\hat{\mathbb{P}}^n_i(X_j)$, the empirical marginal of $X_j$ under the single-intervention $i$.
        Evaluate some distance between $\hat{\mathbb{P}}^n(X_j)$ and $\hat{\mathbb{P}}^n_i(X_j)$.
        If the distance exceeds a threshold, mark $j$ as a descendant of $i$.
    Next $i$.
Next $j$.
Compute the transitive closure of the descendant information matrix, and return it.
```

$I_{\mathcal{C}}$ can then be identified as the set of all self-descendants. Since the intervention set I has at least one node from each cycle in the underlying graph, $I_{\mathcal{C}}$ constitutes a cycle-breaking intervention set, meaning that if all nodes in $I_{\mathcal{C}}$ are intervened at, the CGBN behaves like a BN. Thus with i.i.d. data obtained as such, we can recover the corresponding embedded DAG using generic BN structure learning, which we do not elaborate further on. Note that I itself is a cycle-breaking intervention set, the merit here being that $I_{\mathcal{C}}$ can be much smaller.

Note that the only edges that are in the underlying graph but are missing from the embedded DAG are those from cycle breakers to their children. The following subroutine obtains a child information matrix, based on $I_{\mathcal{C}}$-intervention and leave-one-out from $I_{\mathcal{C}}$ intervention i.i.d. data. Once this information is obtained, all cycles can be closed in a straightforward fashion, recovering the underlying structure. Once again, the correctness of the subroutine follows from our work in [6] and the convergence of empirical distributions, since only children will exhibit a change in marginal conditional.

Subroutine: Detect children

```
Start with the recovered DAG, and sets of $n$ i.i.d. samples generated by the CGBN, under $I_{\mathcal{C}}$-intervention as well as leave-one-out interventions, i.e. on $I_{\mathcal{C}} \setminus \{i\}$ for each $i$ in $I_{\mathcal{C}}$. Initialize a binary $|V| \times |I_{\mathcal{C}}|$ child information matrix. Denote by $\tilde{\pi}_j$ the parents of node $j$ according to the recovered DAG.
For each $j \in V$:
    For each $\alpha \in \mathcal{X}^{|\tilde{\pi}_j|}$:
        Compute the empirical marginal conditional $\hat{\mathbb{P}}^n_{I_{\mathcal{C}}}(X_j|X_{\tilde{\pi}_j} = \alpha)$, call it $\mathbb{Q}_1$.
        For each $i \in I_{\mathcal{C}}$:
            Compute the empirical marginal conditional $\hat{\mathbb{P}}^n_{I_{\mathcal{C}} \setminus \{i\}}(X_j|X_{\tilde{\pi}_j} = \alpha)$, call it $\mathbb{Q}_2$.
            Evaluate some distance between $\mathbb{Q}_1$ and $\mathbb{Q}_2$.
            If the distance exceeds a threshold, mark $j$ as a child of $i$.
        Next $i$.
    Next $\alpha$.
Next $j$.
Return the completed child information matrix.
```

III. Results

To illustrate the algorithm, we simulated a GBN that has fourteen variables,shown in Figure 3, each with three states $\mathcal{X} = \{0, 1, 2\}$, two cycles $5 \to 6 \to 7 \to 5$ and $8 \to 9 \to 10 \to 11 \to 8$, and nodes 7, 8 and 10 available for intervention. The other network has four cycles. The stochastic kernels were sampled continuously from the 3-simplex. Up to 4000 data points were sampled for every required intervention.

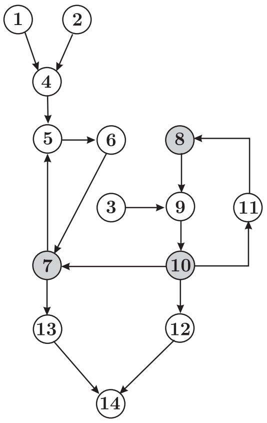

Fig. 1. Test network, recovered exactly by algorithm

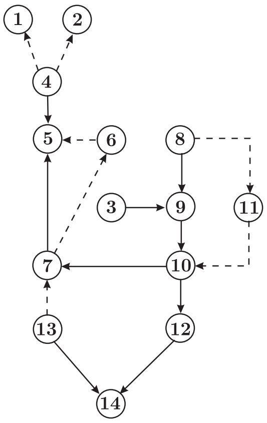

Fig. 2. Best network recovered by BN structure learning

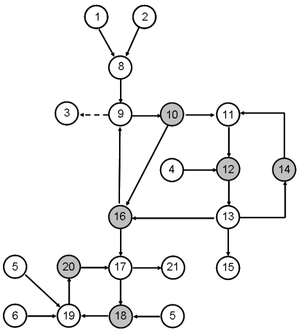

Fig. 3. Network recovered by algorithm

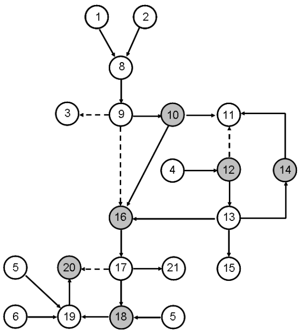

Fig. 4. Best network recovered by BN structure learning

In the tables of Figure 6, we compare the performance of our algorithm to a plain BN structure learning algorithm for the various data sizes. In particular, the tables document the number of true edges that the algorithms uncover, the number of reversed edges that they give, and the number of edges that they add but which are absent in the original

GBN algorithm

Data	Correct	Inverted	Added
1000	14	0	0
2000	15	0	0
4000	16	0	0

BN structure learning

Data	Correct	Inverted	Added
1000	9	3	0
2000	9	7	0
4000	12	4	2

Fig. 5. Performance for 1^{st} Network

GBN algorithm

Data	Correct	Inverted	Added
1000	19	3	2
2000	22	2	0
4000	23	1	0

BN structure learning

Data	Correct	Inverted	Added
1000	19	3	0
2000	20	4	0
4000	20	5	2

Fig. 6. Performance for 2^{nd} Network

graph. Observe that the GBN algorithm recovers the network exactly with 4000 data points. The comparison is inherently unfair, because BN structure learning does not handle cycles, but the emphasis here is on illustrating the type of pitfalls in using BNs to capture data that is generated by a GBN. Using the best recovered DAG in Figure 4, for instance, will mistakenly predict that an intervention at node 10 will not affect node 9.

A. Genetic regulatory networks for Galactose metabolism genes

Since our ultimate goal is structure learning in the biological domain, in this section, we test our algorithm on a small real world dataset from the galactose metabolism system in the organism Saccharomyces cerevisiae (budding yeast). In yeast, when the sugar galactose is present, the enzymes responsible for its metabolism are produced (transcribed). The genetic regulation of these enzymes relies on the protein products of the genes gal4, gal3 and gal80 (as is classically known) and on additional genes such as gal10 (as shown in [1]). We use the dataset generated by [1] to build two small networks from the galactose system. [1] use *abundance perturbations*- a standard perturbation in biology experiments in which a gene or set of genes are *knocked out* or eliminated from the system, such that the actual gene is missing from the perturbed yeast cells. The algorithm presented above is formulated to enable handling of such perturbations.

In the galactose system, the sugar galactose, along with the protein gal4, regulate gal3, gal80 and gal10. These downstream genes also have a regulatory role- they feedback upon the downstream genes, such that gal3, 80 and 10 each regulate gal3, gal80 and gal10 (note that gal4 transcription is *not* affected by these genes). Therefore, the correct model should show galactose and gal4 as parents of gal3, 80 and 10, and gal3, 80 and 10 as parents of each other. The dataset in [1] does not permit us to investigate this entire model, partly because of the dataset's relatively small size, and partly because it lacks the requisite multiple perturbations per sample (i.e. double knockouts, see II). However, we are able to investigate smaller models with one cycle each. We ran standard Bayesian network structure learning on data from [1] for the nodes galactose, gal4, gal3 and gal80 (network I) and for the nodes galactose, gal4, gal3 and gal 10 (network II). We then applied our algorithm and compared the results.

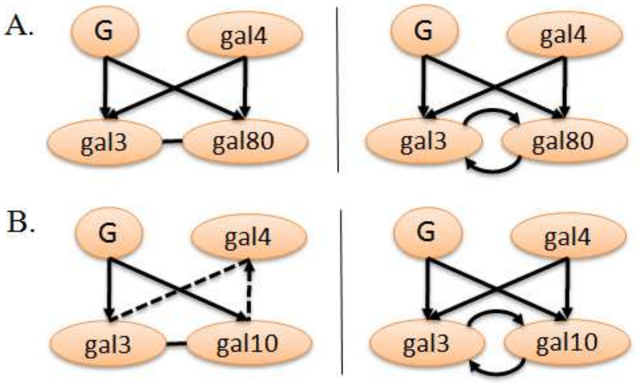

Fig. 7. *Structure learning results for two galactose system networks.* Panel A on the left shows results from standard Bayesian network structure learning performed on network I, including the nodes galactose, gal4, gal3 and gal80. The network is nearly correct, but misses the cycle between gal3 and gal80. The network on the right is correctly inferred by GBN algorithm. Panel B shows results from network II, including the nodes galactose, gal4, gal3 and gal10. The network on the left, from standard structure learning, incorrectly orients two edges and misses the cycle between gal3 and gal10. Results from the GBN, shown on the right, orient the edges correctly and also include the cycle. Undirected edges indicate that the edge direction could not be determined; dotted edges indicate incorrectly oriented edges.

Figure 7 shows the results of our structure learning efforts. Panel A shows network I, with the standard Bayesian network structure learning results on the left, and results from our algorithm on the right. The undirected edge between gal3 and gal80 indicates that the two models (with the edge in either orientation) each scored alike. Thus, the Bayesian network in this case is able to determine the existence of a connection between gal3 and gal80, but it (understandably) cannot assess the correct orientation of this edge. The results from our algorithm, shown on the right, *do* include the correct cycle. The results from network II are shown in panel B, with the results from standard structure learning on the left. In this case, along with missing the cycle, standard structure learning also fails to correctly orient certain edges (indicated by dotted lines). This can occur when the edge direction is not clear cut due to constraints of the model (for instance, the score's inherent complexity penalty will encourage a smaller in-degree, in some cases shifting the preference of the child node to be the node with the smaller overall number of parents). Because our algorithm puts an emphasis on node descendants as indicated by perturbation data, it is able to correctly orient these edges. It is also able to learn the correct cycle as in network I.

IV. Conclusion and future work

In this paper, we demonstrate the first-ever application of cyclic-structure learning with score-based Bayesian networks in a biological context. We present a structure learning algorithm grounded in the GBN formalism, and capable of handling biologically relevant perturbations, namely, perturbations affecting molecule activity, perturbations affecting abundance, or a dataset including a combination of the two. We test our algorithm on two synthetic networks and two small real-world networks using expression data. In each case, our algorithm demonstrates clearly superior performance, both in terms of elucidation of cyclic structures, and in terms of correctly orienting additional model edges.

In the biological domain, we are often interested in a causal model, partly for the insight and understanding such a model conveys with respect to the modeled system, and partly for the possibility for system predictions which it enables. In disease states for instance, a characterization of the altered biological network can serve to guide therapeutic interventions. A truly causal model which includes correctly oriented edges is crucial- with it, a useful target can be identified and potentially detrimental effects can be avoided. Whereas previous attempts at modeling biological pathways with Bayesian networks have yielded useful results, the prevalence of cycles have confounded those efforts, compromising the causal nature of the learned models. With this work, by overcoming the acyclicity constraint, we have brought the structure learning capability incrementally closer to learning truly causal models.

There remains substantial need for increasing the causal interpretability of models. Even without the acyclicity constraint, the presence of hidden variables- rampant in the biological domain- can confound causal model interpretations. In our GBN model presented here, it is necessary to expand it further to enable handling of other experimental constraints, for instance, systems in which only one variable can be perturbed in each experiment (currently not possible with our algorithm). Future improvements to structure learning capability will lead us closer to truly causal models, leading to improved biological understanding and potential for therapeutic benefits.

References

[1] T. Ideker, V. Thorsson, J. A. Ranish, R. Christmas, J. Buhler, J. K. Eng, R. Bumgarner, D. Goodlett, R. Aebersold, L. Hood (2001). Integrated Genomic and Proteomic Analyses of a Systematically Perturbed Metabolic Network. *Science* Vol. 292. no. 5518, pp. 929 - 934.

[2] Friedman, N. and Linial, M. and Nachman, I. and Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *J Comput Biol*, 3-4, Volume 7.

[3] Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, 5659, Volume 303.

[4] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*.

[5] K. Sachs, D. Gifford, T. Jakkola, P. Sorger, and D. A. Lauffenburger(2002). Bayesian network approach to cell signaling pathway modeling. *Sci STKE*.

[6] S. Itani, M. Ohannessian, K. Sachs, G. Nolan and M. Dahleh (2008). Formalism and structure learning for cyclic structures. *Neural Information Processing Systems, submitted*.

[7] G. Cooper and C. Yoo (1999). Causal discovery from a mixture of experimental and observational data. *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 116-125.

[8] N. J. Druzdzel and H. A. Simon (1993). Causality in Bayesian belief networks. *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 3–11.

[9] N. Friedman, K. Murphy, and S. Russell (1999). Learning the structure of dynamic probabilistic networks. *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 139-147.

[10] D. Heckerman, C. Meek and G. F. Cooper (1999). A Bayesian approach to causal discovery. *Computation, Causation, and Discovery*, C. Glymour and G. F. Cooper, Eds., MIT Press, pp 141-166.

[11] U. Nodelman, C. Shelton, and D. Koller (2003). Learning continuous time Bayesian networks. *Nineteenth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 451-458.

[12] J. Pearl (1988). *Probabilistic Reasoning in Intelligent Systems*. Morgan Kauffman.

[13] J. Pearl and T. S. Verma (1991). A theory of inferred causation. *Second KR*, pp. 441-452.

[14] P. Spirtes (1995). Directed cyclic graphical representations of feedback models. *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*, pp. 491–498.

[15] A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, and R. A. Young (2001). Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pac Symp Biocomput*.

[16] P. J. Woolf, W. Prudhomme, L. Daheron, G. Daley,and Q. and D. A. Lauffenburger (2004). Bayesian Analysis of Signaling Networks Governing Embryonic Stem Cell Fate Decisions. *Bioinformatics*.

[17] D. Pe'er (2005). Bayesian network analysis of signaling networks: a primer.*Sci STKE*