Incorporating Topological Prior Information to Learn Modular Gene Regulatory Networks

Viswesh Periyasamy
Department of Computer Science
University of Wisconsin-Madison
vperiyasamy@wisc.edu

December 15, 2018

1 Introduction

1.1 Background

Gene regulatory networks are models which depict the interactions between regulating molecules and their targets (e.g transcription factors and target genes) that drive their expression patterns. Networks are defined by their structure and parameters, where the structure determines which relationships exist between molecules and the parameters define how they interact. Understanding these components could lead to discoveries that highlight which molecules play significant roles in certain biological processes. However, accurately reconstructing these networks, both experimentally and computationally, remains an open problem.

Although both procedures have seen steady improvement over the years, experimental and computational approaches still cannot produce a complete picture. Experimental methods such as ChIP-chip and ChIP-seq are able to recover the structure of the network but cannot easily capture details about the context such as direct and indirect relationships. Furthermore, these experiments are expensive so performing them under a variety of conditions to observe these details is not feasible. On the other hand, computational approaches are inexpensive and different models can capture relationships with higher complexity, but even the best methods have shown lackluster results when compared to biological literature. Although the exact reason for this is unclear, it is well known that high-dimensional data ($n \ll p$) is difficult to work with, so gene expression data alone seems to be insufficient as input to these methods.

1.2 Network motifs as a prior

One approach to improve the accuracy of inferred networks is to incorporate prior knowledge from biological sources. These sources might be derived from literature or experimental networks and can be used during the inference process to guide which edges are learned. For example, one can use known transcription factor binding locations as a form of weighted confidence in edges. However, little work has been done to incorporate priors at a higher level by examining the network topology as a whole or in part.

Network motifs are subgraphs (formed from a subset of the vertices and edges of a graph) which are statistically significant and found to be recurring within a single network or set of networks. One example of a network motif in biology is the Feed-Forward Loop (FFL) depicted in Figure 1 [1]. This motif is comprised of three nodes and three edges. In the context of gene networks, it can be interpreted as Gene A regulating Genes B and C, and Gene B regulating Gene C. By biasing the inference strategy to favor known network motifs such as a FFL, it may be possible to improve the accuracy of learned networks.

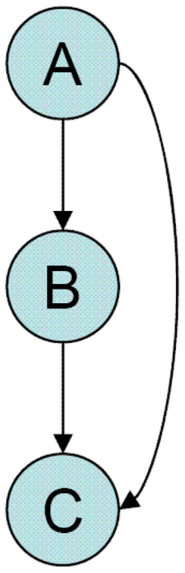

Figure 1: Feed-Forward Loop network motif diagram from [2].

However, directly searching for and enumerating network motifs is computationally expensive and often infeasible, as enumerating all possible k-vertex subgraphs within a network is an $O(n^k)$ procedure. Instead, it may be more reasonable to build this search directly into the chosen inference algorithm to amortize its cost. The aim of this study will be to develop an efficient way to search for network motifs while simultaneously biasing the inference strategy to favor such motifs. To investigate this problem, this work restricts the space of influential network motifs to FFLs in order to fit the scope of this project.

2 Related Work

The literature regarding regulatory network inference using priors is extensive, so this study highlights work regarding network inference using structural priors and network motifs that parallel the goals of this work.

One study by Mahajan [3] took a slightly different approach in modifying their inference approach. They utilize information about known degree distributions in regulatory interactions to enforce a relative sparsity throughout their network. Specifically, they acknowledge that indegree is known to follow a restricted exponential distribution while outdegree follows a scale-free distribution. They then apply these constraints in three different methods, and the authors show success in recovering other motifs such as Fan-in and Cascade motifs, although they concede that they are unable to alleviate error in recovering FFLs.

Another study from Zhang et al. [4] used a variety of machine learning models spanning from Support Vector Machines to Recurrent Neural Networks in order to predict certain kinds of network motifs for given transcription factors. Specifically, they trained classifiers on input network motifs in order to predict when a gene might be important for such a motif and bias their inference that way. They were able to show a reduced error on test set data when network motif information was incorporated with gene expression data than without and especially when considering FFLs. These results motivate further investigation into network motif-based methods.

FANMOD from Wernicke and Florian [5] is a well known tool for fast network motif detection and naturally a good candidate when considering potential approaches. Although it works well, it uses a sampling based approach to enumerate network motifs (since it is computationally intractable to exactly enumerate) and thus was disregarded in this study for two reasons. The first is that its stochastic nature does not lend itself to actually favor FFLs or any network motifs of any kind, and the second is that it operates on a network level meaning it would be difficult to incorporate into the inference problem.

These methods all show promise in the area of network motifs, however none of them are able to incorporate that information at the edge level of network inference. For these reasons, this work extends the existing algorithm of MERLIN+Prior [6] to further study the problem. MERLIN+Prior is an iterative, greedy network inference algorithm which learns modular regulatory networks using gene expression data and different kinds of prior information. Because it already incorporate priors at the edge level, it lends itself nicely to extension of other forms of priors.

3 Approach

At a high level, the goal of this approach is to favor a network with more FFLs as opposed to less. In the context of the MERLIN+Prior, this equates to favoring one regulator-target relationship over another regulator for that same target if the former relationship contributes to more FFLs. To be as explicit as possible, this approach increases the probability of an edge $u \rightarrow v$ proportionally by the number of FFLs that are created by its presence. This measure is calculated with respect to the neighborhood of nodes u and v.

3.1 Defining edge probability

MERLIN+Prior uses probabilistic graphical models in the form of dependency networks to represent its internal structures. Additionally, it incorporates prior information about the network structure from multiple sources. The inference problem then boils down to maximizing the quantity $P(G, \Theta|D)$, where G represents the graph structure, Θ represents the parameters, and D is the given expression data. Using Bayes rule, this quantity can be decomposed to a different but proportional product as given by the following equation:

$$P(G, \Theta|D) \propto P(D|G, \Theta)P(\Theta|G)P(G) \tag{1}$$

where $P(D|G, \Theta)$ is the data likelihood, $P(\Theta|G)$ is the prior distribution of the parameters, and $P(G)$ is the prior distribution of the graph.

To calculate the prior probability of the network structure, MERLIN+Prior uses the following equation:

$$P(G) = \prod\nolimits_{X_j \to X_i \in G} P(X_j \to X_i) \prod\nolimits_{X_j \to X_i \notin G} 1 - P(X_j \to X_i) \tag{2}$$

which can be interpreted as the product of the probability of all existing edges and 1 minus the probability of all absent edges. Thus the problem can be further decomposed to defining the probability of an edge as follows:

$$P(X_j \to X_i) = \frac{1}{1 + \exp(-(p + \beta^R f_{j,i} + \sum_k \beta^k \times w^k_{j,i}))} \tag{3}$$

Multiple priors are encoded by their parameters and accompanying hyperparameters in this logistic function, so this study seeks to extend this approach by defining a new quantity within the denominator to represent the number of FFLs that an edge will contribute to.

3.2 Tracking potential Feed-Forward Loops

In order to keep track of potential FFLs that would be created, three relationships are maintained between triads of nodes:

1. Two nodes are considered to be **co-parents** if they share a common child node.

2. Two nodes are considered to be **co-siblings** if they share a common parent node.

3. Two nodes are considered to be in a **grandparent** relationship if the first regulates some other node and that other node regulates the second.

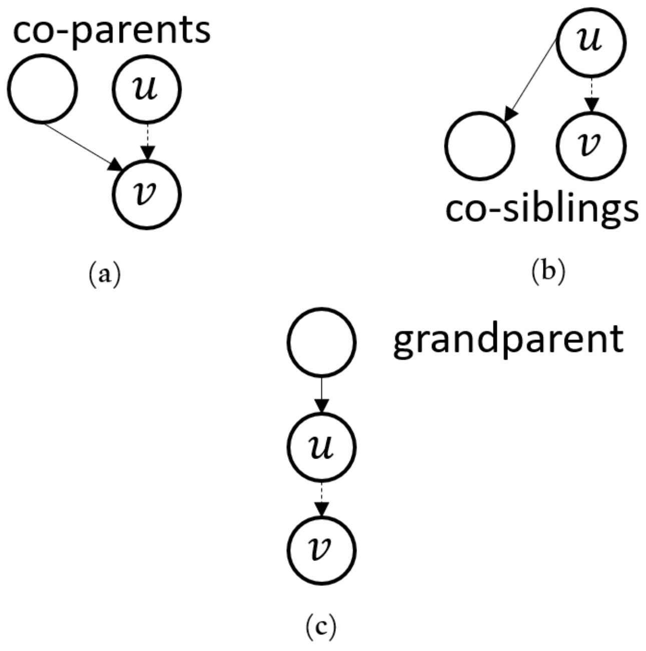

Figure 2: Relationships capturing two-node components of Feed-Forward Loops.

Note that these relationships (depicted in Figure 2) enumerate the pairs of edges which comprise a potential FFL. When adding an edge between any nodes u and v, three situations are considered. First, all parents of v are now said to be maintaining a co-parent relationship with u. Second, all children of u are said to maintain a co-sibling relationship with v. Lastly, all parents of u are said to maintain a grandparent relationship with v.

When actually scoring an edge $u \to v$, the number of FFLs that will be created, call this quantity Z, is the sum result of three scenarios, where **A**, **B**, and **C** here refer to the labels in Figure 1.

1. Node u is **A**, node v is **B**, #FFLs $= |\{c \in C : c = \mathit{coparents}(u, v)\}|$
2. Node u is **B**, node v is **C**, #FFLs $= |\{a \in A : a = \mathit{cosiblings}(u, v)\}|$
3. Node u is **A**, node v is **C**, #FFLs $= |\{b \in B : b = \mathit{grandparents}(u, v)\}|$

3.3 Normalizing potential Feed-Forward Loops

While Z now gives us the number of FFLs this edge will create, one concern is to ensure that the logistic function in Equation 3 is not saturated by dominating the other prior terms when the number of FFLs created gets too large. In order to combat this, a normalization scheme is proposed by counting the total number of potential FFLs that could be created. In other words, this is the maximum value that Z can take on given the current neighborhood of the nodes in the proposed edge. This can again be decomposed into three situations for edge $u \to v$, where the desired quantity is the maximum intersection of either the parent or child set of each node under each situation. A maximum intersection can be thought of as the largest possible intersection between two sets, whose size is simply the cardinality of the smaller set (in which case the smaller set is a complete subset of the other). Given these conditions, the following cases can be summed up to get the normalization constant:

1. Node u is **A**, node v is **B**, #FFLs = $\min(|targets(A), targets(B)|)$
2. Node u is **B**, node v is **C**, #FFLs = $\min(|regulators(B), regulators(C)|)$
3. Node u is **A**, node v is **C**, #FFLs = $\min(|targets(A), regulators(C)|)$

By normalizing Z with this sum, the logistic function in Equation 3 is updated with a new normalized term z:

$$P(X_j \to X_i) = \frac{1}{1 + \exp(-(p + \beta^R f_{j,i} + \sum_k \beta^k \times w^k_{j,i} + \beta^F z_{j,i}))} \tag{4}$$

where β^F is a new hyperparameter controlling the strength of this scheme.

4 Results

To analyze the efficacy of this approach, the FFL-enriched MERLIN+Prior algorithm was applied to the Natural Variation (NatVar) dataset used in the original MERLIN+Prior paper [6]. The NatVar dataset is collected from a multitude of sources and is comprised of gene expression data in Yeast. Experiments were first conducted on a subset of the data to show preliminary results before normalization, where the subset was defined by the intersection of genes present among gold standard networks from MacIsaac et al. [7] and Hu et al. [8]. These results are summarized by Table 1.

Settings	# FFLs	# Edges	Iterations	Recall	Precision	F-score
$\beta^F = 0$	763	1,994	24	0.379121	0.034604	0.063419
$\beta^F = 0.1$	26,007	5,564	46	0.807692	0.026420	0.051166
$\beta^F = 1$	32,985	6,806	42	0.961538	0.025713	0.050086
$\beta^F = 5$	33,395	6,883	42	0.972527	0.025716	0.050106

Table 1: Performance of inferred networks without normalization.

After implementing normalization, the algorithm was run again on the full NatVar set with the MacIsaac network used as the sole gold standard. A hyperparameter value of $\beta^F = 5$ was used in similarity to other default hyperparameters in MERLIN+Prior and was compared against one run with $\beta^F = 0$ as a control. Furthermore, 100 runs were performed on randomly generated subsamples of the data to build a consensus network. The consensus network was then thresholded to accept only edges present in at least 50 percent of inferred networks. Results are depicted below in Table 2.

Settings	# FFLs	# Edges	Iterations	Recall	Precision	F-score
$\beta^F = 0$	271,138	144,911	50	0.219621	0.005762	0.011230
$\beta^F = 5$	173,145	67,857	21	0.101262	0.005674	0.010745
Consensus $\beta^F = 5$	N/A	14,540	N/A	0.0341925	0.008941	0.014175

Table 2: Performance of inferred networks with normalization on NatVar dataset.

Precision-Recall curves were also generated for the three methods for comparison as shown in Figure 3, with accompanying AUPR and AUROC values listed in Table 3.

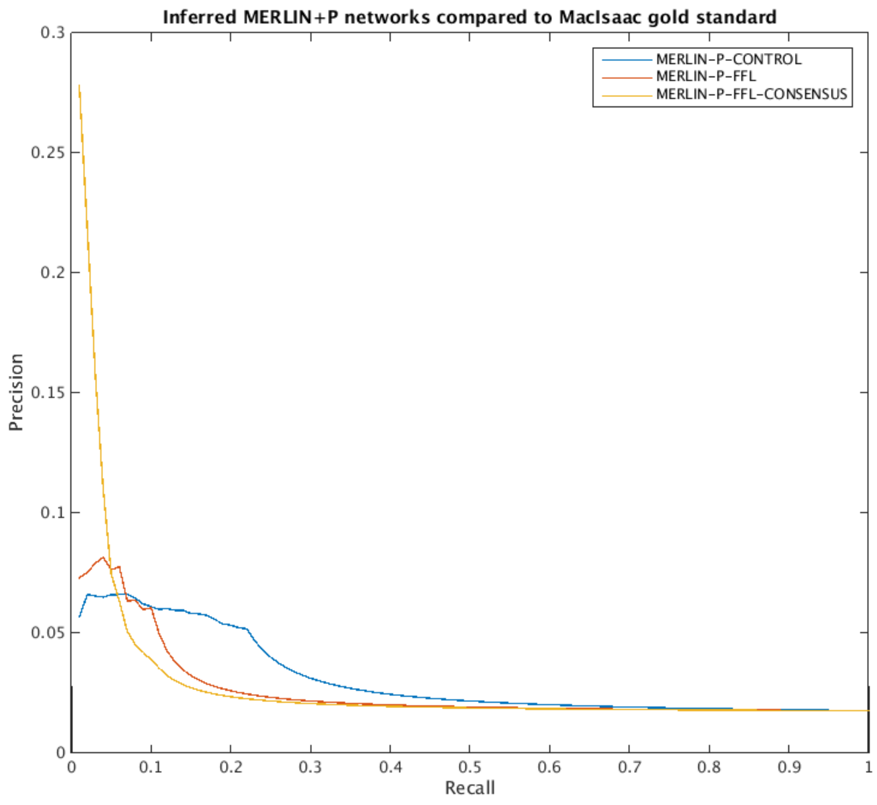

Figure 3: Precision-recall curves for each method as compared to MacIsaac gold standard.

Settings	AUPR	AUROC
$\beta^F = 0$	0.031304	0.574581
$\beta^F = 5$	0.026905	0.536632
Consensus $\beta^F = 5$	0.029804	0.528984

Table 3: AUPR and AUROC values for each method as compared to MacIsaac gold standard.

5 Discussion

5.1 Inferred network results

Initial analysis of the inferred networks across different conditions presents some surprising results. As the numbers show, the approach without normalization clearly favors FFLs but performance was lacking. This leads to the conclusion that the approach was likely learning several meaningless FFLs which polluted the network with false edges. After implementing normalization and enforcing some sparsity, the results still show that the control performed better. While the consensus network outperformed the full run in terms of absolute precision and recall, the curves in Figure 3 show that the consensus network matched even worse than the full run, and both methods paled in comparison to the control run. One reason for this could be that the FFL enriched methods stopped much earlier at around 20 iterations instead of the full 50 iterations that the control run. The default convergence criteria may have caused this, leading to less opportunities to infer the correct edges. That being said, the measures are all still relatively close across methods meaning that the FFL approach isn't far behind the control even in these circumstances.

5.2 Analysis of learned Feed-Forward Loops

The control network performed better in every aspect, but the most striking result is that the control actually inferred many more FFLs as compared to the FFL-enriched networks. This can again be attributed to the much smaller networks due to early-stopping. Investigating the gold network (total of 3,802 edges) shows that while the control recovered 144,911 edges, it missed 2,967 of the gold standard edges. If these had been recovered, it would have resulted in 27,346 more FFLs.

On the other hand, the FFL enriched algorithm only recovered 67,857 edges and missed 3,417 gold edges. However, if it had recovered those, it would've resulted in 10,567 extra FFLs. One conclusion to draw from this is that even though the FFL enriched method recovered about a third of the number of edges as the control, it was able to infer more of the correct FFLs than the control method, since it missed even more gold edges but missed less than half of the amount of FFLs that the control did from gold edges. Normalizing the number of FFLs learned compared to number of inferred shows that the enriched method learned about 2.5 FFLs per edge, as opposed to 1.8 for the control. Thus, although the network inference may have been stopped early, it was still able to favor FFLs during its procedure as well as choose more of the actual FFLs from the gold standard than the control was.

6 Future Work

As this study has depicted, this approach shows promise but needs some fine tuning to be able to draw concrete conclusions. The first concern to investigate is the early stopping. At the time of submission, the convergence criteria has been relaxed and runs are ongoing to collect more accurate results on the NatVar dataset. This will allow a more fair analysis of learned FFLs compared to the gold standard and overall accuracy of inferred networks.

Additionally, more experiments which vary both the hyperparameters and datasets are necessary to gauge the robustness of the system. RNA-Seq data, both bulk and single-cell, has been increasingly replacing microarray expression data as the norm and would provide a good reference point against other algorithms. To determine a good hyperparameter, grid search may be performed although the author is looking into Markov chain Monte Carlo based approaches to learn hyperparameters during inference.

Lastly, the current system is written only in the context of FFLs. A necessary extension will be to generalize the framework to other motifs, such as Fan-in modules, Single-input modules, and Dense overlapping regulons among others. One approach to this might be to read in all of the edge relationships that make up a network motif, and record relationships which agree with the input. Further results will confirm whether these motifs will help improve the accuracy of inferred regulatory networks.

References

[1] Uri Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461, 2007.

[2] Network motif, Nov 2018.

[3] Tarun Mahajan. *Gene regulatory network inference using structural prior on the distribution of edges*. PhD thesis, 2017.

[4] Yuji Zhang, Jianhua Xuan, Benildo G De Los Reyes, Robert Clarke, and Habtom W Ressom. Network motif-based identification of transcription factor-target gene relationships by integrating multi-source biological data. *BMC Bioinformatics*, 9(1):203, 2008.

[5] S. Wernicke and F. Rasche. Fanmod: a tool for fast network motif detection. *Bioinformatics*, 22(9):1152–1153, Feb 2006.

[6] Alireza F. Siahpirani and Sushmita Roy. A prior-based integrative framework for functional transcriptional regulatory network inference. *Nucleic Acids Res*, 45(4):2221–2221, Feb 2017. gkw1160[PII].

[7] K D MacIsaac, D B Gordon, D Gifford, G Stormo, and E Fraenkel. An improved map of conserved regulatory sites for saccharomyces cerevisiae. *BMC Bioinformatics*, 7(113), Mar 2006.

[8] Zhanzhi Hu, Patrick J Killion, and Vishwanath R Iyer. Genetic reconstruction of a functional transcriptional regulatory network. *Nature Genetics*, 39(5):683–687, Aug 2007.