Published in final edited form as:

IET Syst Biol. 2009 September ; 3(5): 404. doi:10.1049/iet-syb.2008.0161.

Scalable learning of large networks

Sushmita Roy[§], Sergey Plis[§], Margaret Werner-Washburne⁺, and Terran Lane[§]

[§] Department of Computer Science, University of New Mexico

⁺ Department of Biology, University of New Mexico

Abstract

Cellular networks inferred from condition-specific microarray data can capture the functional rewiring of cells in response to different environmental conditions. Unfortunately, many algorithms for inferring cellular networks do not scale to whole-genome data with thousands of variables. We propose a novel approach for scalable learning of large networks: Cluster and Infer Networks (CIN). CIN learns network structures in two steps: (a) partition variables into smaller clusters, (b) learn networks per cluster. We optionally revisit the cluster assignment of variables with poor neighborhoods.

Results on networks with known topologies suggest that CIN has substantial speed benefits, without substantial performance loss. We applied our approach to microarray compendia of glucose-starved yeast cells. The inferred networks had significantly higher number of subgraphs representing meaningful biological dependencies than random graphs. Analysis of subgraphs identified biological processes that agreed well with existing information about yeast populations under glucose starvation, and also implicated novel pathways that were previously not known to be associated with these populations.

1 Introduction

Functional networks are networks of genes with edges representing statistical dependencies. The structure of functional networks provides insight into genetic and protein interaction patterns employed in cells to respond to different environmental conditions. Reconstruction of genome-scale functional networks, involving thousands of nodes, can provide a systems-level understanding of cellular response mechanism to different environmental stresses.

Probabilistic graphical models (PGMs) are well-known frameworks for modeling functional networks [5,8,10,26,20], because of their intuitive graph representation of the interaction set, as well as their ability to probabilistically reason with noisy biological data. Similar to clustering approaches of gene expression data, PGMs can identify higher-order statistical dependencies among arbitrarily sized groups of genes. However, PGMs additionally provide fine-grained, network-based description of the dependencies within a gene group.

Network structure inference algorithms are used to infer the graph structure in PGMs. Unfortunately, for genome-scale networks, these algorithms become expensive in time and memory. Structure inference of biological networks is further complicated by the lack of sufficient data to reliably learn graphs over thousands of nodes. Some approaches for addressing the data sparseness problem identify variable clusters and replace the clusters by pseudo variables [27,6]. These approaches pool the data from several variables to robustly estimate parameters. However, they do not capture the fine-grained structure within a cluster.

We present a novel, tractable approach to learn the structure of functional networks represented as undirected graphical models. Our approach, Cluster and Infer Networks (CIN), clusters the

nodes into smaller groups and learns separate networks per cluster. By partitioning the nodes into smaller groups, we avoid searching over the complete node set, resulting in runtime benefits. Furthermore, learning smaller networks using the same amount of data alleviates the data sparseness problem. Because the initial clustering may not be perfect, we iteratively reassign nodes to clusters to improve the quality of the node neighborhoods, repeating the procedure until convergence. As this revisiting has extra computational cost, we make it optional by specifying the number of nodes to be revisited as an input parameter to CIN. The complete network structure is obtained by combining the networks inferred per cluster.

Several previous approaches have been described that combine clustering with network structure inference [28,17]. Toh *et al.* cluster random variables, followed by replacing the cluster with meta random variables representing the average expression profile per cluster. Inoue *et al.* learn a model from time series data capturing the temporal dynamics of inter-cluster dependencies. Both these approaches focus on learning inter-cluster dependencies, whereas we focus on learning the intra-cluster structure. The revisit step additionally allows to establish inter-cluster connectivity.

CIN is a meta-algorithm and can work with any existing algorithm for learning network structure. We used the Markov blanket search (MBS) algorithm [23] for learning the structure of undirected graphs (specifically, Markov random fields). We compared CIN with and without cluster re-assignment, against standard MBS that infers networks over the entire, unpartitioned node set. This comparison was done on simulated data generated from networks of known structure. CIN gave significant speed improvements without significant accuracy loss of the inferred structures. Adding cluster reassignment further improved performance, still keeping CIN faster than MBS with no clustering.

We applied CIN to two yeast microarray compendia of glucose-starvation induced stationary phase [3]. Gene ontology (GO) enrichment analysis of subgraphs identified biological processes that were exclusively enriched in the two populations. These processes agreed well with existing knowledge and included novel findings generating testable hypothesis of response mechanism employed by yeast cells in starvation conditions.

We also compared the properties of the learned graphs against random graphs with same number of edges. Our inferred graphs had significantly more subgraphs that were enriched in at least one GO term. We found that subgraphs that were topologically close (measured by the number of edges across the subgraphs) also exhibited similar process enrichments. The correlation between topological and annotation similarity was significantly higher than random suggesting that the inferred subgraphs were representing biologically meaningful dependencies.

Overall CIN has the following advantages: (a) we provide a scalable approach to learn generic statistical dependencies in biological networks, (b) our approach captures fine-grained dependencies, which cannot be captured by simple clustering approaches, and, (c) subgraphs in our inferred networks represent significantly more biologically meaningful dependencies than in random networks.

2 Learning probabilistic graphical models for biological networks

A probabilistic graphical model (PGM) is associated with two components: a graph $\mathbf{G}$ and a set of functions $\psi = \{\psi_1, \dots, \psi_m\}$. Graph nodes represent random variables, $\mathbf{X} = \{X_1, \dots, X_n\}$, encoding the expression level of genes. The edges of $\mathbf{G}$ describe the statistical dependency structure among the random variables (RVs). Each ψ_i describes the functional relationship between a random variable X_i and its neighbors. In directed PGMs such as Bayesian networks, ψ_i describes the conditional distribution of X_i given its parent variables. In undirected PGMs

such as Markov random fields (MRFs), ψ_i corresponds to potential functions, one for each clique in $\mathbf{G}$.

We focus here on learning functional networks, where the edges correspond to strong statistical correlations among random variables. Such functional interactions represent a broad class of phenomenological associations among network nodes, for which specifying directionality may be difficult or even irrelevant as in protein-protein interactions. Therefore, we learn undirected graphs, where functional interactions are represented by the undirected edges of the graph. However, directed PGMs are often used to learn functional networks because of the decomposability of the likelihood score into local per-family scores, making them easier to learn than undirected PGMs. Although directed PGMs can be used to represent causality, it is difficult to infer causal edges from observational data alone, leaving only a correlation implication for most edges. This difficulty arises due to the presence of many likelihood equivalent structures consistent with the data [16], producing at best a partially directed graph. While the undirected edges of this graph can be oriented using perturbations, doing so relies on the assumptions of no external hidden causes [15], which may be difficult to satisfy from purely observational data.

We learn the structure of MRFs using the Markov blanket search (MBS) algorithm [22,23]. MBS is an extension of Abbeel *et al.*'s Markov blanket canonical parameterization (MBCP) [1]. MBCP exhaustively enumerates over all variable subsets (candidate cliques) up to size s and their Markov blankets (MB) up to size k , where s and k are pre-specified, maximum subset and MB sizes, respectively. In contrast, MBS avoids exhaustive enumeration by establishing an equivalence between MBCP and local per-variable canonical parameters. As a consequence, we need to estimate the MB of only individual RVs, instead of all subsets. MBS additionally learns structurally consistent Markov blankets, by ensuring that if $X_i \in \mathbf{M}_{X_j}$, then $X_j \in \mathbf{M}_{X_i}$, where $\mathbf{M}_{X_i}$ is the MB of X_i , and $\mathbf{M}_{X_j}$ is the MB of X_j . MBS identifies the best, structurally consistent MB for each RV by minimizing an information theoretic measure, *conditional entropy* [7].

2.1 Speeding up structure search using Cluster and Infer Networks (CIN)

Due to the exponential complexity of optimal structure search, most structure learning algorithms, including MBS, learn networks with bounded neighborhood size, k [1,10]. Although this reduces the structure search space, there are still exponentially many candidate networks to be evaluated ($O(2^{kn \log n})$) [11]. Hence, when n becomes very large (several hundreds) and k is of any interesting size ($2 \leq k \leq 10$), as is the case for genome-scale networks, these algorithms become prohibitively expensive in time and memory.

We introduce the CIN approach to further speed up structure learning of graphical models, enabling efficient learning of these models for genome-scale data. CIN comprises two steps: (a) cluster variables, and (b) infer graph structures per cluster. An optional revisit step is included where cluster assignment of variables with poor neighborhoods is updated. In CIN, variables can be clustered using any clustering approach. We use k -means with absolute value of Pearson's correlation as the similarity measure between two variables [14].

CIN is a meta-approach that works with any structure learning algorithm that identifies the best neighbor set (or parent set for Bayesian nets). We illustrate CIN with the MBS algorithm (Algorithm 1). The algorithm takes as input the gene expression data matrix $\mathcal{D} = \{\mathbf{x}^1, \dots, \mathbf{x}^{|\mathcal{D}|}\}$, where $\mathbf{x}^d$, $1 \leq d \leq |\mathcal{D}|$, represents a joint assignment to $\mathbf{X}$ from the d^{th} datapoint. The number of clusters, c , maximum neighborhood size, k , and the number of variable neighborhoods to be revisited, r , are additional inputs to the algorithm.

The algorithm begins with partitioning the n RVs into c clusters, $\mathbf{C}_i$, $1 \leq i \leq c$. This is followed by inferring the structure per cluster $\mathbf{C}_i$ independently. In MBS this entails identifying the MB for each X_i , $\mathbf{M}_{X_i}$, such that $|\mathbf{M}_{X_i}| \leq k$ and the conditional entropy, $H(X_i|\mathbf{X}_i)$ is minimized.

The revisit step is executed if $r > 0$. This selects r variables with the worst neighborhoods (high conditional entropy in MBS) and stores them in $\mathbf{Y}$. For each $Y \in \mathbf{Y}$, we compute the change in conditional entropy, $S_{YZ} = H(Y|\mathbf{M}_Y) - H(Y|\mathbf{M}_Y \cup \{Z\})$, on adding new variables Z to $\mathbf{M}_Y$. Z is selected from clusters other than the current cluster of Y , $\text{curr_cluster}(Y)$. $\mathbf{M}_Y$ is updated to include the variable with the maximal score improvement, Z^* . After considering all $Y \in \mathbf{Y}$, the cluster assignments are updated to incorporate the MB modifications. This in effect creates modified, possibly overlapping clusters. In the actual implementation of CIN with MBS, the revisit step is executed for every l , $1 \leq l \leq k$. We do not include this iteration in Algorithm 1 for ease of exposition.

As structures are inferred per cluster, which are much smaller than the total number of variables, we have more data to learn smaller graphs, thus producing robust structures. The speed up in CIN increases with n . Assuming that we have c clusters each of size m such that $cm = n$, the speed up is $O(\frac{n}{c} 2^{km \log^{a'} b})$, where $a = 1 - \frac{1}{c}$ and $b = c^{\frac{1}{c}}$. Thus, CIN becomes increasingly advantageous with increasing values of n .

3 Experimental strategy

The goal of our experiments was two fold: (a) compare the performance of Cluster and Infer Networks (CIN) against standard graph structure learning algorithms, using both running time and quality of inferred structures, (b) evaluate the biological significance of structures inferred using CIN on real microarray data.

We address (a) using artificial regulatory networks generated from a network simulator [24], allowing us to compare the inferred structures against ground truth. We used small networks (< 200 nodes) so that the structure could be learned using the standard graph structure learning algorithms in reasonable time. We address (b) on two microarray datasets from yeast under glucose starvation conditions [3].

3.1 Performance comparison using simulated networks

To compare the true and inferred graph structures we use different types of scores that capture higher-order dependencies. We use these scores in addition to the traditional edgewise scores where one matches the edges in the true and inferred networks. Our higher-order scores include *pathwise* scores and general *subgraph* scores. We define the precision and recall for each score, and compute the harmonic mean of precision and recall to obtain the F-score.

The pathwise scores extends the traditional edgewise scores by allowing edges from the inferred network to match shortest paths in the true network (and vice versa). The pathwise precision measures how well predicted edges are matched by paths in the true network.

Pathwise precision is: $P_p = \frac{1}{N} \sum_{i=1}^N S_i$, where N is the number of inferred edges and $S_i = \frac{1}{1+(l_i-1)}$, l_i is shortest path length in the true network between nodes of the i^{th} predicted edge. If no path exists, $l_i = n + 1$, where n is the number of nodes in the network. S_i is inversely proportional to l_i and is maximum ($= 1$) when an inferred edge does match a true edge. This definition of precision favors shorter paths, and allows more flexibility when predicted edges correspond to higher-order structures, such as paths, in the true network. Recall is defined in a similar way where edges in the true network are matched against paths in the predicted network.

Our *subgraph* scores match subgraphs between the true and inferred networks, generated from a vertex and its neighbors $\leq r$ steps away. We refer to these as r -n subgraphs. These scores measure the overall ability of algorithms to capture correct subsets of edges. We use a “weight” per subgraph, proportional to the subgraph size, which gives more importance to larger structures that are generally harder to match.

Subgraph recall is: $R_s = \sum_{g \in \mathcal{S}_r} \bar{w}_g \frac{|E_g \cap \widehat{E}_g|}{|\widehat{E}_g|}$, where $\mathcal{S}_r$ denotes a set of r -n subgraphs in the true network, $\bar{w}_g = \frac{|V_g|}{\sum_{f \in \mathcal{S}_r} |V_f|}$ is weight, and V_g is the vertex set of subgraph g . E_g is the edgeset of g , and $\widehat{E}_g$ is the inferred edge set among $v \in V_g$. The precision is defined similarly, except, E_g in the denominator is replaced by $\widehat{E}_g$. These scores lie in the interval $[0, 1]$, and equal 1 when all subgraphs are matched perfectly.

3.2 Validation of inferred graph structures using real microarray data

We developed a global measure of validation of the entire graph structure, *Annotation-Topological Similarity* (ATS), which assesses if subgraphs that are topologically close in the inferred network, are also similarly annotated. ATS is defined as the Pearson’s correlation

coefficient between two vectors, v_A and v_T , each of length $\binom{|\mathcal{S}|}{2}$, where $\mathcal{S}$ is the set of subgraphs generated from the inferred network. Each dimension, $v_A(r)$ is the annotation similarity for r^{th} subgraph pair, where $1 \leq r \leq \binom{|\mathcal{S}|}{2}$. Similarly, each dimension, $v_T(r)$ is the topological distance for r^{th} subgraph pair. We now define annotation and topological similarity.

The topological similarity for each subgraph pair $\{S_i, S_j\}$, is defined as $t_{ij} = \frac{\sum_{v \in V_i, u \in V_j} 2^{-d(u,v)}}{|V_i||V_j|}$, where V_i and V_j are the vertex sets of S_i and S_j respectively, and $d(u, v)$ is the shortest path length between vertices u and v . Subgraphs that are not connected in the graph are not considered for computing the ATS measure.

The annotation similarity, f_{ij} between S_i and S_j , is the Pearson’s correlation co-efficient between two vectors, e_i and e_j . Here, e_i is the Gene Ontology (GO) slim process enrichment vector for S_i . Each dimension, $e_i(t)$, is the logarithm of p -value enrichment for the t^{th} process term. The more negative the ATS, the more likely it is for similarly annotated subgraphs to have a small distance, and differently annotated subgraphs to have a large distance.

We evaluated the statistical significance of the ATS measure per inferred network using a background distribution of ATS from random networks. Specifically, for each inferred network, we generated 100 random networks using a randomization technique that preserved the degree distribution of the networks [21], computed ATS for each random network, and generated the histogram of the ATS values.

3.3 Dataset description

We evaluated the merit of our CIN approach on data generated from networks of known topology. Simulated data was from a regulatory network simulator [24], and provided realistic data from a model that is not in the class of models that we are analyzing. We used simulated data from three networks, ECOLI, G75 and G50, with $n = 188, 150$, and 100 nodes respectively. ECOLI is obtained from the regulatory network of the bacteria *E. coli* [25], and G75 and G50 are artificial regulatory networks generated from a network simulator. Nodes in these networks represent genes or proteins, and edges represent regulatory or protein interactions. These networks are sufficiently large to require our pre-clustering approach, yet small enough to

enable structure learning via an approach without clustering. Each simulated dataset has 1000 datapoints, each representing a steady-state measurement after perturbing all or only transcription factor genes.

We also applied our algorithm to microarray data from two yeast compendia from stationary phase [3]. Each compendium measures the gene expression profile of two yeast populations, quiescent and non-quiescent, isolated from glucose-starved stationary phase cultures. Each microarray corresponds to a gene knockout. Both compendia were done in duplicate with 170 measurements (85 knockouts) for quiescent, and 186 measurements (93 knockouts) for non-quiescent. We selected 2818 genes for each dataset based on reproducibility of gene expression across the duplicates.

4 Results

4.1 CIN has significant speed benefits without substantial performance loss on simulated data

We compared three approaches to infer networks: (a) CIN with MBS with no cluster reassignment (Norevisit), (b) CIN with MBS with cluster reassignment (Revisit), and (c) standard MBS (Nocluster). For (a) and (b) we specified the number of clusters to the k -means algorithm to be 5.

We evaluated the inferred network quality by matching: (a) edges in the true and inferred networks, (b) edges in the true network to paths in the inferred network and vice-versa, (c) 1-n and 2-n subgraphs per vertex from the true network with the inferred network.

The running time of CIN with MBS, Norevisit and Revisit, is significantly smaller than standard MBS (Fig. 1). Further, the speed up for ECOLI (with 188 nodes) is much greater than G50 (with 100 nodes) corroborating our observation that CIN becomes increasingly advantageous with the number of nodes in the network.

The quality of the inferred networks using both CIN approaches are comparable to MBS on the complete variable set (Fig. 2). We show all results other than 1-n subgraphs, which is similar to 2-n. Overall, we found that CIN had significant speed benefits over standard MBS. Revisiting clusters improved results at additional runtime cost, but was still faster than standard MBS.

4.1.1 Performance under limited data conditions—We did preliminary experiments to compare the performance of CIN against the no-clustering approach as a function of decreasing training data (Fig 3). In these experiments, network size was kept the same but the amount of training data d varied, where $d \in \{20, 50, 100, 200, 400, 600, 800\}$. We found that while all approaches suffered performance loss with decreasing amounts of data, the no clustering approach tended to suffer a greater performance deterioration than the CIN approaches. As a result, CIN performed at par for the smaller training data sizes ($d < 50$). This together with the speed benefits increases attraction of CIN under limited data conditions.

4.2 Application to real microarray data

We applied CIN with MBS to two recently generated microarray compendia of yeast in stationary phase [3]. For both datasets, we specified $k = 4$ as the maximum neighborhood size, and $c = 20$ as the number of k -means clusters. Various statistics of the clusters are reported in Table 1. As the true network is not known for these data, we used Gene Ontology (GO) for validation [4].

4.2.1 CIN-inferred networks have non-random topological properties—We first computed the number of 1-n subgraphs that were enriched in a Gene Ontology slim (GO slim)

process term as a function of decreasing p -value (Fig 4). GO slim process is a collapsed single level view of the complete GO biological process, providing high level information of the types of biological processes involving a set of genes. For both datasets, the number of subgraphs enriched in a term was significantly higher than random networks. The results belong to CIN with the revisit step. Results without the revisit step are similar.

We then computed the Annotation-Topological Similarity (ATS) at each p -value using the subgraphs that had an enrichment at that p -value (Fig 5). ATS provides a global measure of biological validity of the inferred networks by evaluating to see if subgraphs that are close together (number of edges crossing them) also participate in the same set of biological processes. The lower the ATS score is, the more meaningful the dependencies captured in the graph.

We found that the inferred networks had higher ATS than random networks, and this was significantly higher ($p < 0.05$, simulation) at GO slim enrichment p -value, $10^{-4} < p < 0.05$. As the p -value decreased the error bars increased due to too few random subgraphs satisfying the threshold. The low ATS in the inferred networks indicates that subgraphs that are topological close, also participate in similar biological processes. Overall this suggests, that networks learned by CIN are significantly different from random, with many subgraphs representing meaningful dependencies. The global structure of the networks also captures the biological intuition that topologically close subgraphs participate in similar processes.

4.2.2 CIN identifies subgraphs involved in meaningful biological processes from yeast stationary phase compendium—We evaluated the 1-n subgraphs inferred using CIN with and without the revisit step using enriched GO process terms (p -value $< 10^{-6}$, and false-discovery rate < 0.05). In comparison to the small set of terms (acetyl-CoA metabolic process) enriched in the k -means clusters, the 1-n subgraphs had a large number of term enrichments, many of which were very specific (tricarboxylic acid cycle). This is expected as the k -means clusters are very coarse without any graph structure. In contrast, learning the internal dependency structure identified smaller, more specific enrichment information. Both versions of CIN, with and without revisit, had similar term enrichments, with revisit identifying slightly more terms.

The specific terms enriched in the quiescent and non-quiescent subgraphs indicated population-specific similarities and differences (Table 2, Fig ??). For example, in the quiescent population a pro-growth pathway was down-regulated, which is known to be crucial in this population to avoid expensive, growth-related processes. In the non-quiescent population, telomere maintenance was up-regulated, which agrees with previous observation that these cells have unstable genomes [2].

We also identified novel processes associated with these populations. In the quiescent population, we found an aging-related subgraph and in the non-quiescent population, we found subgraphs involved in ammonium transport. These cells have been hypothesized as models for aging studies [13], and we provide specific candidate networks that can provide deeper understanding of aging and other cellular processes involved in proper stress response.

5 Discussion

In this paper, we present a novel approach, Cluster and Infer Networks (CIN), for tractable learning of probabilistic graphical models (PGMs) from genome-scale expression data. At the heart of our approach is a *divide and conquer* strategy, where the problem of learning a single large graph is replaced by a set of problems, each learning smaller graphs. This is especially beneficial for biological networks, where the global structure is composed of biologically

meaningful local structures such as pathways or protein complexes. Learning PGMs using CIN has the following advantages: (a) capture generic statistical dependencies among arbitrarily sized groups of genes, (b) learn these dependencies at low runtime, (c) by learning smaller networks with the same amount of data as available for the complete network, we identify many more biologically meaningful subgraphs than in random networks.

CIN is a meta-algorithm combining a clustering algorithm with a graphical model learning algorithm. For this paper, we used the k -means algorithm for clustering. CIN leaves the choice of model order (k) and, indeed, the clustering algorithm itself as free parameters. Clearly, the performance of CIN will depend on these choices, and deliberately so: control over the clustering mechanism provides the user flexibility to express prior domain knowledge, when it is available. The problem of model order selection, when the user wishes to remain agnostic, remains a difficult problem in machine learning, statistics, and pattern recognition literature for decades. While full solution of this problem is beyond the scope of this paper, we note that a number of automated or semi-automated approaches to selecting the model order have achieved wide popularity [29,14].

The revisit step, which updates cluster assignment of variables with poor neighborhood, reduces the dependence of the inferred network upon the initial clustering. Additionally, revisiting produces overlapping clusters, which provides a better global description of how the local parts of the network are connected. As the revisit step has additional running time, we specify the number of variables to be revisited as an input to the algorithm. This allows the revisit step to be executed in a domain-specific manner, depending upon the importance of obtaining the connectivity across clusters.

Our approach is similar to Lee *et al.*'s approach, where a global structure is also learned by identifying graphs per gene cluster [19]. However, Lee *et al.*'s approach generates clusters by finding genes similar in expression and annotation to a set of differentially expressed *seed* genes. This approach relies on both annotation and mRNA similarity to produce clusters. We instead cluster using only mRNA expression, and prefer to use annotation for validation to provide an unbiased quality estimation of inferred structures.

By learning graphs per cluster, CIN prunes the search space of possible graphs by restricting candidate neighbors to the cluster of a variable. This is similar to the Sparse candidate algorithm for learning Bayesian networks where parents per variable are searched within a small set of candidate parents [12]. However, by searching within a cluster, CIN restricts the candidate neighbors upfront (with the exception of variables being revisited), whereas Sparse candidate increments the candidate set by iteratively scanning the complete set of variables. As the candidate set is smaller and remains fixed in CIN, we can allow neighborhoods to be larger than the neighborhoods allowed in the Sparse candidate algorithm. Furthermore, Sparse candidate revisits the candidate parent sets of all n variables. In contrast, CIN revisits the neighbor set of at most r variables, $r \ll n$, making CIN's revisit faster than that of Sparse candidate. Because CIN and Sparse candidate are exploiting a similar heuristic of pruning the search space, comparison of both frameworks on ground truth networks is useful. We plan to address that as future work.

To assess the performance effect of the divide and conquer heuristic of CIN, we compared networks inferred using Markov blanket search (MBS) over the complete graph versus networks inferred using MBS within the CIN framework. We used simulated data from networks of known structure. These networks were large enough to justify using CIN, and small enough to also enable learning the complete graph. Our results indicated that CIN had significant speed improvements without incurring substantial performance loss.

We expect CIN to lose performance as compared to a non-clustering approach. This is because the pre-clustering, which gives us the speed and data benefits, also restricts the search to a smaller space that may miss some good candidate structures. However, as the relative amount of data for a network of given size becomes small, we expect a non-clustering approach to suffer in performance more than CIN. We show preliminary evidence in Section 4.2.2, which suggests that under limited data conditions CIN can perform at par at a lower runtime cost.

We also applied CIN to a large microarray compendium of approximately 2,000 genes measuring expression changes of glucose-starved yeast cells in response to gene knockouts. Analysis of networks inferred from these data identified more biologically meaningful dependencies than in random graphs. Many of the dependencies agreed with existing knowledge, and included novel biological processes not previously known to be associated with these cells. For comparison purposes, we also applied standard MBS with no pre-clustering on one of the compendia datasets (results not shown). In spite of the long run time of learning the complete graph, we found only a modest increase in the number of biological processes that were enriched in different subgraphs. This demonstrated that even for real data, the pruning heuristics employed in CIN successfully identify the important interactions, and do so at low runtime.

Our work can be extended in several directions. We have illustrated CIN with an algorithm for learning undirected graphical models. However, the CIN framework can be applied to directed graphical models as well. Further, for small clusters ($n \leq 40$), we can learn optimal Bayesian networks per cluster via exact structure search [18]. The k -means clustering currently used in CIN produces non-overlapping clusters. CIN can be readily extended to more powerful clustering methods such as non-negative matrix factorization [9], which generates overlapping clusters. Finally, we are applying CIN to expression data from different stress conditions available at the Gene expression omnibus (<http://www.ncbi.nlm.nih.gov/geo/>).

In summary, tractable learning of genome-scale networks is crucial for constructing a complete repertoire of functional interactions among genes. Our CIN approach provides an initial, promising framework to learn these genome-scale networks from microarray data under different conditions. These networks provide fine-grained information of how cells respond to environmental stresses, and can guide future experiments to develop a systems-level understanding of stress response mechanism.

Acknowledgments

This work was supported in part by NIMH grant number 1R01MH076282-03 as part of the NSF/NIH Collaborative Research in Computational Neuroscience Program. This work was also supported in part by the NSF grants IIS-070568 and MCB0734918.

References

1. Abbeel, Pieter; Koller, Daphne; Ng, Andrew Y. Learning factor graphs in polynomial time and sample complexity. *JMLR* 2006;7:1743–1788.
2. Allen C, Büttner S, Aragon AD, Thomas JA, Meirelles O, Jaetao JE, Benn D, Ruby SW, Veenhuis M, Madeo F, Werner-Washburne M. Isolation of quiescent and nonquiescent cells from yeast stationary-phase cultures. *J Cell Biol* July;2006 174(1):89–100. [PubMed: 16818721]
3. Aragon, Anthony D.; Rodriguez, Angelina L.; Meirelles, Osorio; Roy, Sushmita; Davidson, George S.; Allen, Chris; Joe, Ray; Tapia, Phillip; Benn, Don; Werner-Washburne, Margaret. Characterization of differentiated quiescent and non-quiescent cells in yeast stationary-phase cultures. *Molecular Biology of the Cell*. 2008
4. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE,

- Ringwald M, Rubin GM, Sherlock G. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet* May;2000 25(1):25–29. [PubMed: 10802651]
5. Blais, Alexandre; Dynlacht, Brian David. Constructing transcriptional regulatory networks. *Genes and Development* 2005;19
 6. Bonneau, Richard; Reiss, David J.; Shannon, Paul; Facciotti, Marc; Hood, Leroy; Baliga, Nitin S.; Thorsson, Vesteinn. The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biology*. 2006
 7. Cover, Thomas M.; Thomas, Joy A. *Elements of information theory*. Wiley-Interscience; New York, NY, USA: 1991.
 8. de Jong H. Modeling and simulation of genetic regulatory systems: a literature review. *J Comput Biol* 2002;9(1):67–103. [PubMed: 11911796]
 9. Devarajan, Karthik. Nonnegative matrix factorization: An analytical and interpretive tool in computational biology. *PLoS Comput Biol* July;2008 4(7):e1000029. [PubMed: 18654623]
 10. Friedman, Nir. Inferring cellular networks using probabilistic graphical models. *Science* 2004;303:799–805. [PubMed: 14764868]
 11. Friedman, Nir; Koller, Daphne. Being bayesian about network structure; UAI '00: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence; San Francisco, CA, USA. Morgan Kaufmann Publishers Inc; 2000. p. 201-210.
 12. Friedman, Nir; Nachman, Iftach; Pe'er, Dana. Learning bayesian network structure from massive datasets: The sparse candidate algorithm. *Uncertainty in Artificial Intelligence*. 1999
 13. Gray, Joseph V.; Petsko, Gregory A.; Johnston, Gerald C.; Ringe, Dagmar; Singer, Richard A.; Werner-Washburne, Margaret. “sleeping beauty”: Quiescence in *saccharomyces cerevisiae*. *Microbiol Mol Biol Rev* June;2004 68(2):187–206. [PubMed: 15187181]
 14. Hastie, T.; Tibshirani, R.; Friedman, JH. *The Elements of Statistical Learning*. Springer; Aug. 2001
 15. He, Yang-Bo; Geng, Zhi. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research* November;2008 9:2523–2547.
 16. Heckerman, David. Technical Report MSR-TR-95-06. Microsoft research; Mar. 1995 A Tutorial on Learning Bayesian Networks.
 17. Inoue LY, Neira M, Nelson C, Gleave M, Etzioni R. Cluster-based network model for time-course gene expression data. *Biostatistics (Oxford, England)* July;2007 8(3):507–525.
 18. Koivisto, Mikko; Sood, Kismat. Exact bayesian structure discovery in bayesian networks. *J Mach Learn Res* 2004;5:549–573.
 19. Lee, Phil H.; Lee, Doheon. Modularized learning of genetic interaction networks from biological annotations and mrna expression data. *Bioinformatics* June;2005 21(11):2739–2747. [PubMed: 15797909]
 20. Margolin A, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera RD, Califano A. ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* 2005;(Suppl 1):S7.
 21. Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U. Network motifs: simple building blocks of complex networks. *Science* October;2002 298(5594):824–827. [PubMed: 12399590]
 22. Roy, Sushmita; Lane, Terran; Werner-Washburne, Margaret. Technical Report TR-CS-2008-14. University of New Mexico; 2008. Learning structurally consistent undirected probabilistic graphical models.
 23. Roy, Sushmita; Lane, Terran; Werner-Washburne, Margaret; Martinez, Diego. Inference of functional networks of condition-specific response—a case study of quiescence in yeast. *PSB*. 2009
 24. Roy, Sushmita; Werner-Washburne, Margaret; Lane, Terran. A system for generating transcription regulatory networks with combinatorial control of transcription. *Bioinformatics (Oxford, England)*. April;2008
 25. Salgado, Heladia; Gama-Castro, Socorro; Peralta-Gil, Martin; Diaz-Peredo, Edgar; Sanchez-Solano, Fabiola; Santos-Zavaleta, Alberto; Martinez-Flores, Irma; Jimenez-Jacinto, Veronica; Bonavides-Martinez, Cesar; Segura-Salazar, Juan; Martinez-Antonio, Agustino; Collado-Vides, Julio. Regulondb (version 5.0): *Escherichia coli* K-12 transcriptional regulatory network, operon organization, and growth conditions. Vol. 34. *Nucleic Acids Research*; 2006. p. D394

26. Segal E, Shapira M, Regev A, Pe'er D, Botstein D, Koller D, Friedman N. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat Genet* June;2003 34(2):166–176. [PubMed: 12740579]
27. Segal, Eran; Pe'er, Dana; Regev, Aviv; Koller, Daphne; Friedman, Nir. Learning module networks. *Journal of Machine Learning Research* April;2005 6:557–588.
28. Toh H, Horimoto K. Inference of a genetic network by a combined approach of cluster analysis and graphical gaussian modeling. *Bioinformatics (Oxford, England)* February;2002 18(2):287–297.
29. Xing, Eric; Sharan, Roded; Jordan, Michael I. Bayesian haplo-type inference via the dirichlet process. *ICML '04: Proceedings of the twenty-first international conference on Machine learning*; New York, NY, USA. ACM; 2004.

Figure 1.

Run time for different algorithms; lower runtimes are better. Algorithms were compared using three networks of known structure. Per network, two datasets were generated by either perturbing all genes (ECOLI-ALL, G75-ALL, G50-ALL) or only transcription factor genes (ECOLI-TF, G75-TF, G50-TF).

Figure 2. Match scores for different algorithms; higher scores are better. Nocluster: MBS algorithm without pre-clustering. Norevisit: CIN with MBS without cluster reassignment. Revisit: CIN with MBS with cluster reassignment.

Figure 3.

Performance of different approaches as function of increasing data size (x-axis). Performance is measured using Edge-Path and 2nsubgraph match scores defined in Section 3.1. Scores were obtained for two networks (G75 and ECOLI) using data generated after perturbing all genes (ALLPERT) or only transcription factors (TFPERT)

Figure 4. Number of subgraphs that were enriched in a GO slim process term at a specific p -value. Line with errorbars shows the average number of subgraphs enriched in random networks at that p -value. Results are shown on log-log scale.

Figure 5. ATS measure of real and random networks at different p -values. Lower ATS is better. Results are shown on log-log scale.

Table 1

Different statistics of the number of genes per cluster in the quiescent and non-quiescent populations.

Dataset	Mean	Median	Stdev	Max	Min
Quiescent	140.9	146.0	46.69	214	26
Non-quiescent	140.9	147.0	77.41	300	22

Table 2

GO enrichment quiescent (QSCNT) and non-quiescent (N-QSCNT) populations. Up regulated indicates if these subgraphs were significantly upregulated in expression, in addition to representing strong statistical correlation.

QSCNT	GO Process	Hit genes	Up regulated
	negative regulation of Ras protein signal transduction	BCY1, FDH1, IRA1, PXA2, YLR356W	Yes
	replicative cell aging	BOI2, PDR1, SCH9, SNF1, SSD1	No
N-QSCNT	GO Process	Hit genes	Up regulated
	telomere maintenance via recombination	YHL050C, YLL066C, YRF1–4, YRF1–5, YRF1–6	Yes
	ammonium transport	ADY2, ALD4, ATO3, DBR1, TWF1	Yes
	arginine catabolic process	CAR1,CAR2,CUE1	Yes

Algorithm 1

MBS with CIN

Input:

Expression data of n genes, $\mathcal{D} = \{\mathbf{x}^1, \dots, \mathbf{x}^{|\mathcal{D}|}\}$
 maximum neighborhood size, k
 number of clusters, c
 convergence threshold, ε
 number of variables for revisit, r

Output:

Inferred graph structure $\mathcal{G}$.

Partition n variables into c clusters $\mathcal{C} = \{C_1, \dots, C_c\}$, using datapoints as attributes.

while Score change $\geq \varepsilon$ **do**

*/*Infer structure per cluster*/*

for $C_i \in \mathcal{C}$ **do**

 Learn best structure for C_i s.t. $\forall X_j \in C_i, |\mathbf{M}_{X_j}| \leq k$.

end for

*/*Revisit*/*

if $r > 0$ **then**

$\mathbf{Y} = r$ variables with worst MBs

for $Y \in \mathbf{Y}$ **do**

$S_{YZ} = H(Y|\mathbf{M}_Y) - H(Y|\mathbf{M}_Y \cup \{Z\})$, s.t. $\text{curr_cluster}(Z) \neq \text{curr_cluster}(Y)$

$Z^* = \arg \max_Z S_{YZ}$

$\mathbf{M}_Y = \mathbf{M}_Y \cup \{Z^*\}$

end for

 Update $\mathcal{C}$ to account for MB modifications of $Y \in \mathbf{Y}$

end if

end while