Reverse Engineering Molecular Hypergraphs

Ahsanur Rahman, Christopher L. Poirel, David J. Badger, and T. M. Murali
Department of Computer Science
ICTAS Center for Systems Biology of Engineered Tissues
Virginia Tech
Blacksburg, VA, USA
ahsanur@vt.edu, poirel@vt.edu, dbadger@vt.edu, murali@cs.vt.edu

ABSTRACT

Analysis of molecular interaction networks is pervasive in systems biology. This research relies almost entirely on graphs for modeling interactions. However, edges in graphs cannot represent multi-way interactions among molecules, which occur very often within cells. Hypergraphs may be better representations for such interactions, since hyperedges can naturally represent relationships among multiple molecules.

Here we propose using hypergraphs to capture the uncertainty that is inherent in reverse engineering gene-gene networks from systems biology datasets. Some subsets of nodes may induce highly varying subgraphs across an ensemble of high-scoring networks inferred by a reverse engineering algorithm. We provide a novel formulation of hyperedges to capture this uncertainty in network topology. We propose a clustering-based approach to discover hyperedges.

We show that our approach can recover hyperedges planted in synthetic datasets with high precision and recall. We apply our techniques to a published dataset of pathway structures inferred from quantitative genetic interaction data in *S. cerevisiae* related to the unfolded protein response in the endoplasmic reticulum (ER). Our approach discovers several hyperedges that capture the uncertain connectivity of genes in specific pathways and complexes related to the ER.

Our work demonstrates that molecular interaction hypergraphs are powerful representations for capturing uncertainty in network structure. The hyperedges we discover directly suggest groups of genes for which further experiments may be required in order to precisely discover interaction patterns.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and Genetics; E.1 [**Data Structures**]: Graphs and Networks

General Terms

Algorithms, Experimentation

ACM-BCB'12 , October 7-10, 2012, Orlando, FL, USA

1. INTRODUCTION

Interaction networks are increasingly used to represent cellular processes and reason about them [11]. Methods have been developed to reconstruct gene regulatory networks from gene expression profiles [16]; to predict molecular interactions [10]; to classify cellular states [7]; and to compute cellular response networks [25]. An overwhelming majority of these approaches use directed or undirected connections between pairs of molecules to model interaction networks. However pair-wise interactions cannot accurately represent coordinated activity of assemblages of more than two molecules. For instance, pair-wise interactions cannot represent a protein complex that acts as a unit, e.g., the Anaphase-Promoting Complex (APC) that triggers exit from mitosis. Modifier proteins and protein complexes often bind to and modulate the activity of transcription factors. Metabolic reactions may involve multiple substrates and products and be catalyzed by one or more enzymes [17].

Hypergraphs are attractive alternatives to graphs to represent such facets of cellular processes [4, 9, 13, 27]. Informally, a hyperedge (an edge in a hypergraph) is simply a set of one or more nodes; therefore, every edge in a graph is a hyperedge composed of exactly two nodes. Hypergraphs are increasingly being recognized for their utility in accurately representing cellular processes. Many databases and interaction storage formats support hyperedges of different types, either explicitly or implicitly [6, 20]. Such formats have proven useful for converting existing interaction pathways and processes into hypergraph representations.

The power of hypergraphs for representing uncertainty in experimentally and computationally derived interactions is less well recognized. For example, pair-wise interactions are inappropriate for representing protein complexes pulled down by tandem affinity purification; it is widely recognized that the spoke and matrix models [21] are both incorrect representations of purified complexes. While techniques have been developed to infer which pairs of proteins physically interact in each complex [2, 21], representing each protein complex by a hyperedge is natural [19].

More generally, methods that reconstruct gene networks [1, 18] may be able to infer only that there is some set of interactions among a group of molecules but may not be able to precisely discern pair-wise interactions within the group. Furthermore, since experimental data are noisy and limited, there may be multiple network topologies that fit the experimental data equally well. Existing algorithms for inferring and representing molecular interaction networks make simplifying as-

sumptions to account for the under-determined nature of the system [16] or compute a single network that is the consensus of multiple high-scoring networks [8]. *The central thesis of our work is that hypergraphs are natural candidates for representing uncertainty in the topology of the inferred network.*

Contributions. Our primary contribution is formulating the novel problem of reverse engineering hypergraphs from systems biology datasets. Many network inference techniques, for example those discovering Bayesian networks, search the landscape of possible networks until they converge to local optima, thereby generating ensembles of networks with scores that are close to optimal [1, 18]. Although these networks have very similar scores, they may have different dependency and connectivity structures [8, 18]. We take as our starting point a set $\mathcal{G}$ of graphs computed by such an algorithm. In our formulation, a set S of nodes constitutes a hyperedge if S induces very different subgraphs in each of the graphs in $\mathcal{G}$. Intuitively, across the ensemble $\mathcal{G}$, there is no consensus on which specific edges should connect pairs of nodes in S. We deem such a set of nodes to be a hyperedge. We formalize this notion by incorporating parameters that are lower bounds on the number of distinct graphs on S that appear in $\mathcal{G}$ and the number of times each such graph occurs in $\mathcal{G}$.

Our second contribution is an algorithm that discovers hyperedges by computing heavily-weighted clusters in an appropriately defined summary graph. As far as we know, ours is the first paper to explicitly propose using hypergraphs to represent uncertainty in the structure of reverse-engineered gene networks, to propose a formal definition of hyperedges in this context, and to develop an efficient algorithm to compute hyperedges supported by a set of varying graphs.

Results. First, we demonstrate that our approach recovers hyperedges planted in synthetic datasets with high precision and recall, even when there is noise in the data and the planted hyperedges overlap. Second, we highlight an application where we use hyperedges to capture the variations in an ensemble of networks inferred from quantitative genetic interaction (GI) data in *S. cerevisiae* [1]. Upon analysing this data, we observe that our method discovers hyperedges that capture specific pathways and complexes in the ER for whom the GI data do not support well-defined interactions.

2. RELATED RESEARCH

Here, we highlight how our question is conceptually distinct from related areas of research.

Network inference. Our knowledge of molecular interactions that take place within the cell is highly incomplete. To surmount this difficulty, methods have been developed to predict or "reverse engineer" interactions from datasets of information on gene and protein expression. The primary assumption underlying these techniques is that an interaction may be inferred between two genes if they show similar patterns of activities in multiple experimental conditions. Based on this hypothesis, many methods have been developed to infer interactions between pairs of genes [16]. As far as we know, these methods cannot be generalized to predict hyperedges.

Gene modules and network clustering. A functional module may be defined as a set of molecules that interact to execute a discrete biological function. A vast number of approaches have been developed to find modules or communities from one or more molecular networks [14, 23]. All existing methods start from one or more graphs and find dense clusters within these graphs. The clusters may exist within a single graph, be composed of edges arising from different graphs, or occur simultaneously in many graphs (the last version of the problem is often termed frequent subgraph mining in relational graphs). In contrast, in our work, we focus on a completely different type of property: a set of nodes that do not exhibit any consistent pattern of connectivity in any graph.

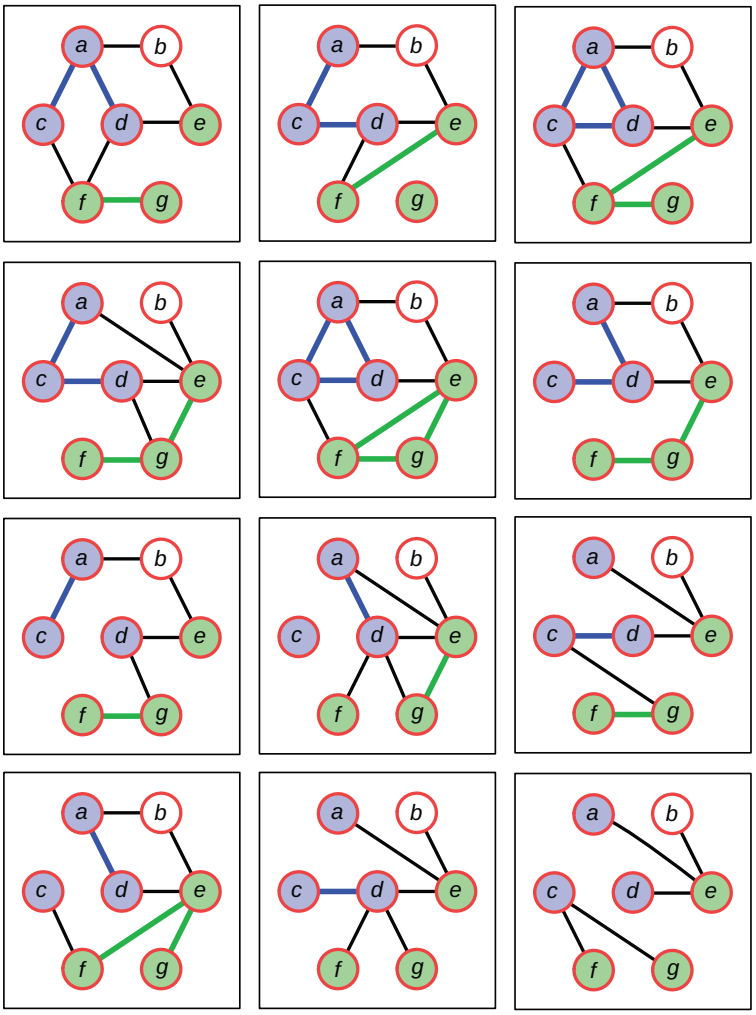
Molecular hyperedges. Some approaches do exist to reverse-engineer specific types of hyperedges from systems biology data. For instance, the MINDY [26] algorithm predicts post-translational modulators of transcription factors (TF). In other words, it predicts directed hyperedges with the TF and its modulator on one side and the target gene on the other. Another example arises in the work by Battle *et al.* on identifying pathways from genetic interaction data [1]. They reconstruct Bayesian networks that represent pathway structures from quantitative phenotypes of double knockout strains of budding yeast. They identify sets of nodes that induce different paths across an ensemble of high-scoring Bayesian networks. In principle, such node sets are similar to the hyperedges we compute. However, the paths among their node sets do not necessarily vary widely across the graphs in $\mathcal{G}$, i.e., only a few of the possible paths among the nodes may be represented in $\mathcal{G}$. Our methodology differs significantly, as we explicitly seek sets of nodes whose induced subnetworks exhibit high variation across the collection of graphs. Moreover, our important contributions include a formal definition of hyperedges and an algorithm to systematically compute hyperedges. In Section 5.2, we demonstrate the value of our approach by applying our hyperedge discovery technique to their ensemble of networks.

3. DEFINITIONS

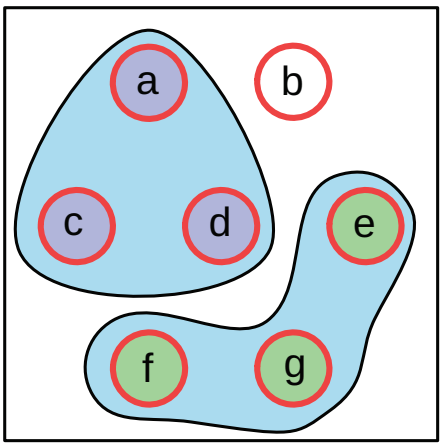
Let $\mathcal{G}$ be a set of n graphs computed by multiple runs of a network inference algorithm. In this paper, we assume that each graph in $\mathcal{G}$ is undirected, unweighted, and has the same set V of vertices. There are a number of ways to define how one set of nodes induces different subgraphs in $\mathcal{G}$. We propose one such formulation in this work.

Given a set $S \subseteq V$ of k nodes and a graph $G \in \mathcal{G}$, let $G(S)$ denote the subgraph of G induced by S, let $\mathcal{G}(S)$ denote the multiset of these subgraphs as we vary the graphs in $\mathcal{G}$, and let $\mathcal{P}(S)$ denote the set of $2^{\binom{k}{2}}$ possible graphs on the nodes in S. Note that the number of distinct subgraphs in $\mathcal{G}(S)$ is at most $\min\left(n, 2^{\binom{k}{2}}\right)$. Consider a graph $H \in \mathcal{P}(S)$. Let $\psi(H)$ denote the number of occurrences of H in $\mathcal{G}(S)$. Ideally, as we vary $H \in \mathcal{P}(S)$, we desire the counts $\psi(H)$ to be as uniform as possible. We capture this notion using the following definition. Given parameters $0 < \beta, \sigma \leq 1$, we say that S is a (β, σ)-*hyperedge* if $\psi(H) \geq \beta n$ for at least $\sigma 2^{\binom{k}{2}}$ graphs in $\mathcal{P}(S)$. The parameters β and σ ensure that the counts $\psi(H)$ are balanced for at least some number of graphs in $\mathcal{P}(S)$.

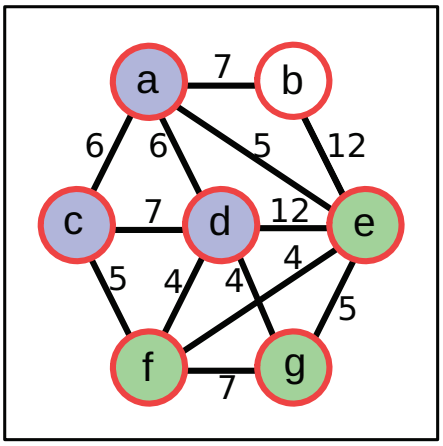
Figure 1(a) illustrates these ideas using a set $\mathcal{G}$ of 12 graphs on the set of nodes $\{a, b, c, d, e, f, g\}$. Consider the set of nodes $\{a, c, d\}$. Each of the eight possible graphs among these



(a) A set $\mathcal{G}$ of graphs to illustrate the definition of a hyperedge. The blue (respectively, green) subgraphs contribute to make the set $\{a, c, d\}$ (respectively, $\{e, f, g\}$) a hyperedge.



(b) The hypergraph containing two $(1/12, 1)$-hyperedges.



(c) The summary graph for the graphs in $\mathcal{G}$ (see section 4.2).

Figure 1: An illustration of a set $\mathcal{G}$ of graphs, two hyperedges defined by $\mathcal{G}$, and the summary graph of $\mathcal{G}$. In Figure 1(c), to aid clarity, we show the number of occurrences of each edge in a graph in $\mathcal{G}$, rather than the fraction of graphs in which the edge occurs.

nodes occurs as a subgraph of at least one graph in $\mathcal{G}$ in the figure, with some graphs occurring exactly once. By our definition, $\{a, c, d\}$ is a $(1/12, 1)$-hyperedge. Figure 1(b) displays all $(1/12, 1)$-hyperedges supported by the set of graphs in Figure 1(a). Observe that four out of the eight possible graphs among $\{a, c, d\}$ appears twice in $\mathcal{G}$. Therefore, $\{a, c, d\}$ is also a $(2/12, 4/8)$-hyperedge. As another example, consider the set of nodes $\{e, f, g\}$. All eight graphs among these nodes appear in $\mathcal{G}$, with two graphs appearing twice each and one graph appearing thrice. Thus, $\{e, f, g\}$ is a $(1/12, 1)$- and a $(2/12, 3/8)$-hyperedge. While this set is also $(3/12, 1/8)$-hyperedge, in practise, we do not consider the pair of parameters $(3/12, 1/8)$ as suggesting a hyperedge, since they mean that only one out of eight possible subgraphs is present in graphs in $\mathcal{G}$. In contrast to these examples, consider the set of nodes $\{a, b, e\}$. Only two of the eight possible graphs occur in $\mathcal{G}$, five and seven times, respectively, making $\{a, b, e\}$ a $(5/12, 2/8)$-hyperedge. Here, the β parameter is quite large $(5/12)$ but the σ parameter is quite small $(2/8)$. This set of nodes is uninteresting for our purpose. For this set of graphs, note that every four-node set is a hyperedge only for values of σ less than $12/64$ $(64 = 2^{\binom{4}{2}})$, since $\mathcal{G}$ contains only 12 graphs; thus, no four-node set is likely to constitute an interesting hyperedge. More generally, the largest (β, σ)-hyperedge has $O\left(\min(\sqrt{-2\log\beta\sigma}, \sqrt{2\log n/\sigma})\right)$ nodes.

We now state the problem we solve in this work:

> *Given a set of graphs $\mathcal{G}$, an integer $k > 0$, and parameters $0 < \beta, \sigma \leq 1$, enumerate all (β, σ)-hyperedges containing k nodes.*

We consider other formulations of the problem in the conclusions (Section 6).

4. ALGORITHM

In this section, we describe an algorithm that formulates the problem of discovering hyperedges in terms of computing clusters in an appropriate summary graph. To motivate the algorithm, consider a (β, σ)-hyperedge S that contains k nodes such that $\sigma = 1$, i.e., each of the $2^{\binom{k}{2}}$ possible graphs on S occurs as a subgraph of some graph in $\mathcal{G}$. For such a hyperedge, the largest possible value of β is $1/2^{\binom{k}{2}}$. In this situation, each pair of nodes in S will appear as an edge in precisely half the graphs in $\mathcal{G}$. Therefore, we can compute such a hyperedge by constructing the average of all graphs in $\mathcal{G}$ and searching for cliques in which each edge has weight equal to 0.5.

We now generalize these observations to arbitrary (β, σ) hyperedges. We first prove lower and upper bounds on the "density" of a hyperedge. We use these bounds to transform the edge weights in the average of all graphs in $\mathcal{G}$. Finally, we use a clustering algorithm to enumerate all dense subgraphs in this transformed graph.

4.1 Bounds on Hyperedge Densities

We start by defining some notation. Given a set $\mathcal{G}$ of undirected, unweighted graphs, let $\mu(G)$ denote the *average* of $\mathcal{G}$, i.e., $\mu(G)$ is an undirected, weighted graph such the edge set of $\mu(G)$ is the union of the edge sets of all the graphs in $\mathcal{G}$ and the weight of each edge in $\mu(G)$ is the fraction of graphs in $\mathcal{G}$ that contain the edge. Given a (β, σ)-hyperedge S, let $\mu_S(G)$ denote the subgraph of $\mu(G)$ induced by the nodes in S. If S contains k nodes, then $\mu_S(G)$ contains at most $\binom{k}{2}$ edges, by definition. In general, any particular edge in $\mu_S(G)$ may have a weight in the interval $[0, 1]$. However, we can establish lower and upper bounds on the density of $\mu_S(G)$, where we define the *density* of a graph to be the total weight of the edges in the graph divided by the number of possible edges in the graph. The following two lemmas state the lower bound and the upper bound, respectively. For the sake of convenience, we assume that $\sigma 2^{\binom{k}{2}}$ is an integer. We provide proofs for both lemmas at: `http://bioinformatics.cs.vt.edu/~murali/supplements/2012-acm-bcb-hypergraphs`.

Lemma 1 *If S is a (β, σ)-hyperedge with k nodes, then the density of $\mu_S(G)$ is at least*

$$\frac{\beta\left(\sum_{i=0}^{l-1} i\binom{\binom{k}{2}}{i} + l\left(\sigma 2^{\binom{k}{2}} - \sum_{i=0}^{l-1}\binom{\binom{k}{2}}{i}\right)[l > 0]\right)}{\binom{k}{2}},$$

where l is the smallest integer such that

$$\sum_{i=0}^{l}\binom{\binom{k}{2}}{i} \geq \sigma 2^{\binom{k}{2}}.$$

In the lemma, [ ] denotes an indicator function, which is true if and only if l is positive.

Lemma 2 *If S is a (β, σ)-hyperedge, then the density of $\mu_S(G)$ is at most*

$$\frac{\beta u\left(\sigma 2^{\binom{k}{2}} - \sum_{i=u+1}^{\binom{k}{2}}\binom{\binom{k}{2}}{i}\right)[u < \binom{k}{2}]}{\binom{k}{2}} + \frac{\beta\left(\sum_{i=u+1}^{\binom{k}{2}-1} i\binom{\binom{k}{2}}{i}\right)}{\binom{k}{2}} + \left(1 + \beta - \beta\sigma 2^{\binom{k}{2}}\right),$$

where u is the largest integer such that

$$\sum_{i=u}^{\binom{k}{2}}\binom{\binom{k}{2}}{i} \geq \sigma 2^{\binom{k}{2}}.$$

Given the parameters $0 < \beta, \sigma \leq 1$ and an integer $k > 0$, let $\lambda(k, \beta, \sigma)$ and $\gamma(k, \beta, \sigma)$ denote the lower and upper bounds defined by Lemma 1 and Lemma 2, respectively, on the density of a (β, σ)-hyperedge with k nodes. For purposes of brevity, we denote the bounds by λ and γ when the parameters are clear from the context. We can prove that $\lambda + \gamma = 1$ (proof omitted).

4.2 Clustering Algorithm

Our algorithm consists of the following steps:

1. Compute $\mu(\mathcal{G}) = \bigcup_{G \in \mathcal{G}} G$, the union of the graphs in $\mathcal{G}$.
2. Assign each edge (u, v) in $\mu(\mathcal{G})$ a weight $w(u, v)$ equal to the fraction of graphs in $\mathcal{G}$ that contain (u, v) as an edge.
3. For each edge (u, v) in $\mu(\mathcal{G})$, transform its weight using the function
$$\frac{1}{1 + e^{\tau \max(\lambda - w(u,v), w(u,v) - \gamma)}},$$
where τ is a large positive number.
4. Compute all highly dense subgraphs of k nodes in $\mu(\mathcal{G})$.

The first two steps simply compute the average $\mu(\mathcal{G})$ of the graphs in $\mathcal{G}$. The third step transforms the edge weights in $\mu(\mathcal{G})$ so that all edge weights in the interval $[\lambda, \gamma]$ are close to 1 and all edge weights outside this interval are small. Note that the value of the maximum in the transformation function is negative iff $w(u, v)$ lies in the interval $[\lambda, \gamma]$. Hence, by choosing $\tau = 100$, we ensure that the transformed weights are close to 1 for edges whose weights lie in the interval $[\lambda, \gamma]$ and are close to 0 otherwise.

Finally, in this transformed graph, we compute all subgraphs with sufficiently high density, and report the node sets of these subgraphs as hyperedges. For a hyperedge S, our intuition is that this transformation will convert $\mu_S(\mathcal{G})$ into a dense (heavily-weighted) subgraph of $\mu(\mathcal{G})$. To enumerate all sufficiently dense subgraphs, we extend the ODES algorithm [15]. ODES hinges on the property that every subgraph with density at least 0.5 contains one node whose removal does not disconnect the graph or decrease the density. While ODES works on unweighted graphs, we were able to extend this property for weighted graphs as well, as long as all edge weights are positive and at most 1 (proof omitted). We downloaded the ODES software and modified it to work on weighted graphs. We also

modified ODES to compute dense subgraphs with exactly k nodes rather than enumerate all dense subgraphs.

Remarks. Our algorithm is a heuristic that is not guaranteed to compute all (β, σ)-hyperedges. Moreover, some sets of nodes computed by our algorithm may not satisfy the properties of a (β, σ)-hyperedge. This discrepancy can arise because the lower and upper bounds apply to the density of a hyperedge but we transform each edge weight individually. Yet our approach works well on both synthetic and real biological data, as discussed below. The worst-case running time of our algorithm can be exponential in k. However, in practice, our algorithm runs very efficiently, as we report below.

5. RESULTS

We divide our results into two parts: (a) synthetic data (Section 5.1) and (b) the pathway structures inferred from double knockout budding yeast strains [1] (Section 5.2). We used a Dell R515 server with 2 x 2.8GHz AMD Opteron 4184 CPUs (12 cores) for all operations. In each execution of our algorithm, we computed all subgraphs with density at least 0.9.

5.1 Synthetic Data

Generation. We created six synthetic datasets to test our algorithm. Each dataset contained 100 sets of graphs, i.e., 100 instances of $\mathcal{G}$. We first describe how we created instances of $\mathcal{G}$ without any overlapping hyperedges or noise. To create such an instance of $\mathcal{G}$, we generated 2048 networks by systematically perturbing the BioGRID protein interaction network for *S. cerevisiae* [24]. This network contains 168,599 interactions among 6063 genes. We initialized each instance of $\mathcal{G}$ with 2048 copies of the BioGRID network. Then, we planted 10 hyperedges of varying sizes (3, 4, or 5 nodes) within each instance of $\mathcal{G}$ as follows. We randomly selected $k \in \{3, 4, 5\}$ nodes and replaced the subgraph induced by these nodes in each of the 2048 networks in $\mathcal{G}$ with a random subgraph generated by the Erdös-Rényi $(k, 0.5)$ model. By adding each possible edge with probability 0.5, we aimed to ensure that the distribution of edges within a hyperedge was relatively uniform. Note that we selected 2048 networks so that we could support $(1, 1/2^{\binom{5}{2}})$-hyperedges with five nodes.

In our first dataset, we did not allow hyperedges to overlap (i.e., share nodes) and did not add any noise. We created five additional datasets that introduced noise and/or overlapping hyperedges. We used two parameters, namely ω and η, to control the amount of overlap and noise, respectively. The parameter ω specified the maximum fraction of nodes within a hyperedge allowed to belong to other hyperedges. We ensured that no hyperedge was fully contained in another hyperedge (thereby forming a Sperner hypergraph). The parameter η specified the number of false positive and false negative interactions that we added to each graph in each instance of $\mathcal{G}$ as a fraction of the total number of node pairs within the implanted hyperedges. For example, if we planted ten five-node disjoint hyperedges, then we added $10 \times \binom{5}{2}\eta = 100\eta$ false positives and false negatives to each graph. To create a false negative (respectively, false positive) interaction in a graph, we randomly selected a pair of nodes in the graph that were connected (respectively, disconnected) and removed that edge (respectively, connect them by an edge). We selected ω from the set $\{0, 0.4\}$ and η from the set $\{0, 0.1, 0.2\}$, thereby obtaining a total of six synthetic datasets.

Evaluation. We applied our algorithm on each of these datasets with $k = 3, 4, 5$, seven values of σ, $\{i/8, 1 \leq i \leq 7\}$, and $\beta = 1/\sigma 2^{\binom{k}{2}}$ (its largest feasible value). To compare the computed hyperedges with the planted hyperedges, we defined precision and recall in the following manner. Let R_i denote the ith planted hyperedge and C_j denote the jth computed hyperedge, where i ranges over the planted hyperedges and j over the computed hyperedges. We defined

$$\text{precision} = \frac{\sum_j \max_i |R_i \cap C_j|}{\sum_j |C_j|}$$

$$\text{recall} = \frac{\sum_i \max_j |R_i \cap C_j|}{\sum_i |R_i|}$$

Note that the numerators of both quantities measured the overlap between the planted hyperedges and the results yielded by our algorithm. For precision, we compared the overlap to the total sizes of computed hyperedges, whereas for recall, we compared the overlap to the total sizes of the planted hyperedges. For each dataset, we measured precision and recall by taking the union of all computed hyperedges (irrespective of their sizes) and by combining the results for all 100 instances of $\mathcal{G}$ within the dataset.

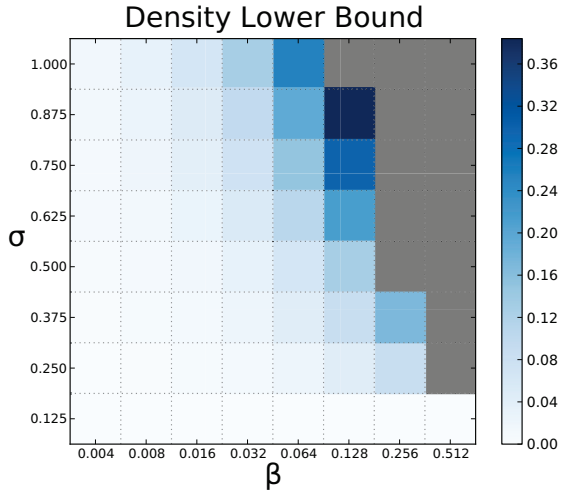
Results. The average time for computing hyperedges in an instance of $\mathcal{G}$ ranged from 1.57 seconds to 2.98 seconds with a standard deviation of 0.12. Within this small range, we observed that the running time generally increased as we increased σ, η, or ω. Across the six datasets, as we varied σ from $1/8$ to $7/8$, precision always had a value of 1, whereas recall varied from 0.71 to 1. Recall was smallest (respectively, largest) for $\sigma = 1/8$ (respectively, $\sigma = 7/8$). Both precision and recall did not vary with ω or η. These results suggest that our method can recover hyperedges planted in synthetic data accurately. In future work, we plan to make the synthetic data generation more sophisticated by incorporating noise in the proximity of the implanted hyperedges, rather than strewing it across the entire network.

5.2 Analysis of Battle *et al*. [1] Data

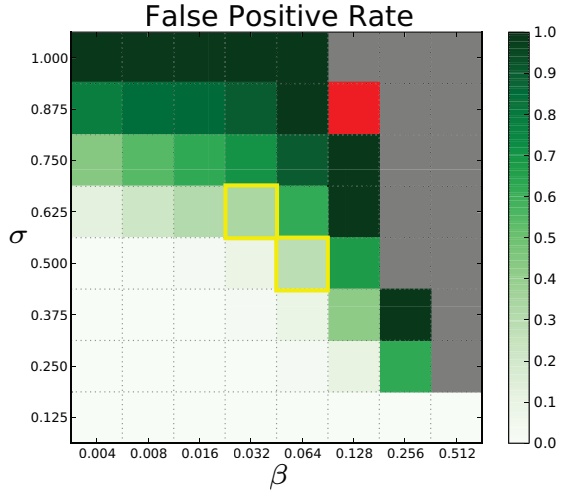
Given quantitative phenotype measurements for a set of single and double knockout organisms, Battle *et al*. computed activity pathway networks (APNs) that represented functional dependencies between genes and their combined effects on the phenotype. Each APN terminated in a node called "Reporter" that represented the quantitative phenotype. They sampled the space of APNs using a Markov chain Monte Carlo method, thereby creating an ensemble of networks . Analogous to our definition of hyperedges, they were interested in a set G of genes that occurred in a single linear chain (in any order). They computed the probability that the genes in G occurred in a linear chain across the ensemble of APNs. When this probability was at least 0.6 and exceeded the probability of occurrence of any specific linear ordering of the genes in G by a factor of 1.8, they collapsed G into a structure similar to our notion of hyperedge. They applied their method to quantitative GI data between pairs of genes [12] whose single mutants upregulated the unfolded protein response (UPR) in the endoplasmic reticulum.

We obtained the 500 APNs computed by them. We treated each APN as an undirected, unweighted graph so as to focus purely on the network topology rather than on the direction-

ality of the probabilistic dependencies. Here we discuss the properties of the hyperedges we computed and what light they shed on the interactions between these genes. Battle *et al.* highlighted examples of interactions and network structures that have support in the literature. In contrast, we focus our attention on groups of genes and related processes among whom pair-wise interactions are difficult to discern in the ensemble of APNs. Our intent is to demonstrate the utility of our approach to interpret this ensemble of networks, rather than to explicitly compare our results to those of Battle *et al.*



(a) Theoretical lower bounds (λ) on density for various (β, σ)-hyperedges of size three.



(b) The FPR for various choices of β and σ. Red cells correspond to parameter values for which our algorithm did not compute any clusters. Gold boxes highlight pairs of β and σ we used for further biological analysis.

Figure 2: (a) Analysis of the theoretical lower bounds on hyperedge density and (b) visualization of the false positive rates in the ensemble of APNs. Grey cells indicate pairs of β and σ that are invalid, i.e., no collection of graphs can support hyperedges with such high values of β and σ.

Evaluation strategy. We executed our algorithm on the ensemble of 500 networks with $k = 3, 4$, and 5, eight values of σ, $\{i/8, 1 \leq i \leq 8\}$, and $\beta = 2^j/500, 1 \leq j \leq 2^{\binom{k}{2}}$. For each dense subgraph computed by our algorithm, we evaluated whether it was truly a (β, σ)-hyperedge. If it was not, we deemed this hyperedge a false positive, and measured the false positive rate as the ratio of the number of false positives to the total number of computed dense subgraphs. As pointed out earlier, our approach may also have false negatives. However, we do not have a method for estimating their count.

This set of 500 networks did not support any four-node hyperedges unless the values of β and σ were very small, which are uninteresting for our purpose. Hence, we focused our attention on three-node hyperedges. Such hyperedges contain too few nodes to support functional enrichment of GO terms. Therefore, we asked if genes that belonged to multiple hyperedges were enriched in any biological function. Accordingly, we computed the degree distribution of the hypergraph, i.e., for every gene we computed the number of hyperedges in which it participated. We computed GO terms enriched in this ranked list of genes using the FuncAssociate algorithm [3], which can take ranked lists of genes as input. We reasoned that this analysis would help us identify biological processes whose genes participated in numerous hyperedges, i.e., processes whose genes were connected in multiple ways both to each other and to genes external to the process.

Results. Figure 2(a) displays how the lower bound on density $\lambda(3, \beta, \sigma)$ varies with the parameters. In general, after fixing β (or σ),λ density monotonically increases with an increase in σ (or β, respectively). For small values of β or σ, the lower bound is 0. Note that for a given value of β and σ, we can prove that the sum of λ and γ is 1; thus, we only plot the lower bounds in (a).

Figure 2(b) illustrates how the false positive rate (FPR), i.e., the fraction of dense subgraphs discovered in the summary graph that are not (β, σ)-hyperedges, varies with β and σ. When both parameters are small (lower left corner), the FPR is close to 0. When σ is high (top) or when both parameters are high (centre), the FPR is close to 1, suggesting that our algorithm computes many dense subgraphs that do not satisfy the constraints laid down by β and σ. We selected two values of the parameters that had FPR less than 0.5: $(0.032, 0.625)$ and $(0.064, 0.5)$, with FPR 0.34 and 0.29, respectively. In the first case, $5/8$ possible subgraphs each appear at least 16 times in the set of 500 graphs. In the second case, $4/8$ possible subgraphs each appear at least 32 times in the 500 graphs. The first case has the advantage of having a higher value of σ. The second case has a lower value of σ, but involves more networks overall from the set of graphs. We examined functions enriched in the list of genes sorted in decreasing order of the number of hyperedges that contained them. We did not observe substantial differences in the results for the two sets of parameters highlighted above. Hence, we focused our attention on the parameters $(\beta = 0.032, \sigma = 0.625)$. Using these parameter-values, we obtained 398 3-node hyperedges. Our method took 1.54 seconds to compute these hyperedges.

The most enriched function was the GARP complex (LOD = 3.1, adjusted p-value = 0). The genes involved in this complex include VPS51, VPS52, VPS53, and VPS54. The Golgi-associated retrograde protein (GARP) complex is required for the recycling of proteins from endosomes to the late Golgi. The original publication of the genetic interaction data [12] noted that a significant set of genes whose deletion caused up-

regulation of the UPR were involved in the late Golgi. Our result suggests that the precise connections among these genes are unclear, at least from the genetic interaction data. This possibility is supported by the fact that the vps52/53/54 triple mutant strain is phenotypically indistinguishable from each of the single mutants [5]. Moreover, each of these genes participates in at least 7 and as many as 18 hyperedges, indicating that the interactions between the GARP complex and other ER proteins are also quite unclear.

The second most enriched process was the GET complex (LOD = 3.08, p-value 0.0024). This complex, which includes the Get1, Get2, and Get3 proteins, is involved in Golgi to ER Traffic, especially in facilitating insertion of tail-anchored proteins into the ER membrane [22].

Another highly enriched GO term was "protein glycosylation" (rank 8, LOD = 1.9, adjusted p-value 0). This term annotated 11 genes (ALG5, ALG6, ALG8, ALG9, ALG12, BST1, DIE2, ERD1, OST3, PMT1, PMT2). The majority of proteins synthesized in the rough ER undergo protein glycosylation. Interestingly, Battle *et al*. reported that their method accurately predicted the ordering of these genes. Their observation appears to contradict our results, which suggest that the pair-wise connections among these genes are difficult to estimate. Upon examining the subgraphs induced by this set of genes in the ensemble of 500 networks, we observed that they involved only 16 unique edges (reinforcing their findings) but in numerous combinations (supporting our result), thus resolving the apparent contradiction. The SWR1 complex, which is involved in chromatin remodeling, was also highly enriched (rank 11, LOD = 1.7, adjusted p-value = 0.003).

In summary, we were able to use the FPR to select appropriate values of the parameters β and σ. We could settle on the value of k by examining the largest value for which our method was able to compute hyperedges. Examination of functional enrichment trends in the list of genes ranked in order of the number of hyperedges they participated in allowed us to discover several ER-related processes among which the genetic interaction data did not support precise pair-wise interactions. These results suggest that more in-depth experiments may be needed to resolve the ambiguity in the connections among these genes.

6. CONCLUSIONS

In this paper, we have proposed hypergraphs as a novel representation for capturing the uncertainty inherent in inferring gene interaction networks from systems biology datasets. Our main theoretical contributions are two-fold: a formal definition of (β, σ)-hyperedges supported by an ensemble of networks and an algorithm for computing (β, σ)-hyperedges of a fixed size. Applying these techniques to a dataset of 500 APNs inferred from quantitative genetic interaction data, we discovered 398 hyperedges. Each hyperedge included genes for which the APNs could not infer precise pair-wise interactions.

We envision that this paper will serve as the basis for a rich body of research. Several extensions and generalizations of our ideas are immediate. For instance, we would ideally like to compute maximal hyperedges (those that are not contained in any other hyperedges). We would also like to systematically enumerate all hyperedges. It may be possible to employ the ideas from itemset mining here. Formulations of the problems other than enumeration are also interesting, e.g., finding the (β, σ)-hyperedge with the largest number of nodes or computing a set of non-redundant (β, σ)-hyperedges or discovering statistically significant hyperedges. We plan to address these problems in the future. We are also considering extensions to weighted and directed graphs.

Ultimately, we are interested in directly inferring hyperedges from diverse datasets without going through the intermediate step of inferring an ensemble of graphs. By discovering such hypergraphs, we hope to pinpoint which set of genes and proteins might be ideal for further experimentation. Incorporating the data from these experiments might help to refine hyperedges and resolve the pair-wise interactions among the nodes, resulting in a fruitful interplay and feedback between computational and experimental scientists.

Acknowledgments. National Institutes of Health grant R01-GM095955-01 and National Science Foundation grants CBET-0933225 and DBI-1062380 supported this work. An NSF Graduate Research Fellowship supported Chris Poirel. We thank Alexis Battle for sharing the dataset of 500 APNs.

References

[1] A. Battle, M. C. Jonikas, P. Walter, J. S. Weissman, and D. Koller. Automated identification of pathways from quantitative genetic interaction data. *Molecular Systems Biology*, 6(1):379, 2010.

[2] A. Bernard, D. S. Vaughn, and A. J. Hartemink. Reconstructing the Topology of Protein Complexes. In *Proceedings of the 11th Annual International Conference on Research in Computational Molecular Biology (RECOMB'07)*, pages 32–46. Springer, 2007.

[3] G. F. Berriz, J. E. Beaver, C. Cenik, M. Tasan, and F. P. Roth. Next generation software for functional trend analysis. *Bioinformatics*, 25(22):3043–3044, 2009.

[4] T. Christensen, A. Oliveira, and J. Nielsen. Reconstruction and logical modeling of glucose repression signaling pathways in *Saccharomyces cerevisiae*. *BMC Systems Biology*, 3(1):7, 2009.

[5] E. Conibear and T. Stevens. Vps52p, Vps53p, and Vps54p form a novel multisubunit complex required for protein sorting at the yeast late golgi. *Molecular Biology of the Cell*, 11(1):305–323, 2000.

[6] E. Demir *et al*. The BioPAX community standard for pathway data sharing. *Nature Biotechnology*, 28(9):935–942, 2010.

[7] J. Dutkowski and T. Ideker. Protein Networks as Logic Functions in Development and Cancer. *PLoS Comput. Biol.*, 7(9):e1002180, 2011.

[8] N. Friedman and D. Koller. Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine learning*, 50(1):95–125, 2003.

[9] L. S. Heath and A. A. Sioson. Semantics of multimodal network models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(2):271–280, 2009.

[10] C. Huttenhower, E. M. Haley, M. A. Hibbs, V. Dumeaux, D. R. Barrett, H. A. Coller, and O. G. Troyanskaya. Exploring the human genome with functional maps. *Genome Res*, 19(6):1093–106, 2009.

[11] T. Ideker and R. Sharan. Protein networks in disease. *Genome Res.*, 18(4):644–652, 2008.

[12] M. C. Jonikas, S. R. Collins, V. Denic, E. Oh, E. M. Quan, V. Schmid, J. Weibezahn, B. Schwappach, P. Walter, J. S. Weissman, and M. Schuldiner. Comprehensive characterization of genes required for protein folding in the endoplasmic reticulum. *Science*, 323(5922):1693–1697, 2009.

[13] S. Klamt, U.-U. Haus, and F. Theis. Hypergraphs and cellular networks. *PLoS Comput. Biol.*, 5(5):e1000385, 2009.

[14] W. Li, C. Liu, T. Zhang, H. Li, M. Waterman, and X. Zhou. Integrative analysis of many weighted co-expression networks using tensor computation. *PLoS Computational Biology*, 7(6):e1001106, 2011.

[15] J. Long and C. Hartman. ODES: an overlapping dense sub-graph algorithm. *Bioinformatics*, 26(21):2788–2789, 2010.

[16] F. Markowetz and R. Spang. Inferring cellular networks - a review. *BMC Bioinformatics*, 8(Suppl 6):S5, 2007.

[17] A. Mithani, G. M. Preston, and J. Hein. Rahnuma: hypergraph-based tool for metabolic pathway prediction and network comparison. *Bioinformatics*, 25(14):1831–1832, 2009.

[18] D. Pe'er. Bayesian network analysis of signaling networks: a primer. *Science's STKE: Signal Transduction Knowledge Environment.*, 2005(281):pl4, 2005.

[19] E. Ramadan, A. Tarafdar, and A. Pothen. A hypergraph model for the yeast protein complex network. In *Proceedings of the 18th International Parallel and Distributed Processing Symposium*, pages 189–196. IEEE, 2004.

[20] C. F. Schaefer, K. Anthony, S. Krupa, J. Buchoff, M. Day, T. Hannay, and K. H. Buetow. PID: the Pathway Interaction Database. *Nucleic Acids Res*, 37(Database issue):D674–9, 2009.

[21] S.-E. Schelhorn, J. Mestre, M. Albrecht, and E. Zotenko. Inferring physical protein contacts from large-scale purification data of protein complexes. *Molecular & Cellular Proteomics*, 10(6):M110.004929, 2011.

[22] M. Schuldiner, J. Metz, V. Schmid, V. Denic, M. Rakwalska, H. Schmitt, B. Schwappach, and J. Weissman. The GET complex mediates insertion of tail-anchored proteins into the ER membrane. *Cell*, 134(4):634–645, 2008.

[23] R. Sharan and T. Ideker. Modeling cellular machinery through biological network comparison. *Nat Biotechnol*, 24(4):427–33, 2006.

[24] C. Stark, B.-J. J. Breitkreutz, A. Chatr-Aryamontri, L. Boucher, R. Oughtred, M. S. Livstone, J. Nixon, K. Van Auken, X. Wang, X. Shi, T. Reguly, J. M. Rust, A. Winter, K. Dolinski, and M. Tyers. The BioGRID interaction database: 2011 update. *Nucleic Acids Research*, 39(Database issue):D698–D704, 2011.

[25] I. Ulitsky, A. Krishnamurthy, R. M. Karp, and R. Shamir. DEGAS: De Novo Discovery of Dysregulated Pathways in Human Diseases. *PLoS ONE*, 5(10):e13367, 2010.

[26] K. Wang, M. Saito, B. C. Bisikirska, M. J. Alvarez, W. K. Lim, P. Rajbhandari, Q. Shen, I. Nemenman, K. Basso, A. A. Margolin, U. Klein, R. Dalla-Favera, and A. Califano. Genome-wide identification of post-translational modulators of transcription factor activity in human B cells. *Nature Biotechnology*, 27(9):829–837, 2009.

[27] W. Zhou and L. Nakhleh. Properties of metabolic graphs: biological organization or representation artifacts? *BMC Bioinformatics*, 12(1):132, 2011.