

Frequent Pattern Discovery in Multiple Biological Networks: Patterns and Algorithms

Wenyuan Li · Haiyan Hu · Yu Huang · Haifeng Li ·
Michael R. Mehan · Juan Nunez-Iglesias · Min Xu ·
Xifeng Yan · Xianghong Jasmine Zhou

Received: 9 June 2011 / Accepted: 2 December 2011 / Published online: 29 December 2011
© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract The rapid accumulation of biological network data is creating an urgent need for computational methods capable of integrative network analysis. This paper discusses a suite of algorithms that we have developed to discover biologically significant patterns that appear frequently in multiple biological networks: coherent dense subgraphs, frequent dense vertex-sets, generic frequent subgraphs, differential subgraphs, and recurrent heavy subgraphs. We demonstrate these methods on gene co-expression networks, using the identified patterns to systematically annotate gene functions, map genome to phenome, and perform high-order cooperativity analysis.

Keywords Frequent pattern · Integrative network analysis · Coherent dense subgraph · Frequent dense vertex-set · Generic frequent subgraph · Differential subgraph · Recurrent heavy subgraph · Tensor representation of multiple networks

W. Li · Y. Huang · M.R. Mehan · J. Nunez-Iglesias · M. Xu · X.J. Zhou (✉)
Program in Computational Biology, Department of Biological Sciences, University of Southern
California, Los Angeles, CA 90089, USA
e-mail: xjzhou@usc.edu

H. Hu
School of Electrical Engineering and Computer Science, University of Central Florida, Orlando,
FL 32816-2005, USA

H. Li
Motorola Labs, 2900 S Diablo Way, Tempe, AZ 85282, USA

X. Yan
Computer Science Department, University of California at Santa Barbara, Santa Barbara,
CA 93106-5110, USA

1 Introduction

The advancement of high-throughput technology has resulted in the rapid accumulation of data on several kinds of biological network, including co-expression networks, protein-interaction networks, metabolic networks, genetic interaction networks, and transcription regulatory networks. They are continuously being generated for a wide range of organisms under various conditions. This wealth of data represents a great opportunity, to the extent that network biology is rapidly emerging as a discipline in its own right [4, 24]. Thus far, most of the computational methods developed in this field have focused on the analysis of individual biological networks. However, a single network is often insufficient to discover patterns with multiple facets and subtle signals. There is an urgent need for methods supporting the integrative analysis of *multiple* biological networks.

On account of the noisy nature of high-throughput data, biological networks contain many spurious edges which may lead to the discovery of false patterns. However, since biological modules are active across multiple conditions, we can easily filter out spurious edges by looking for patterns that appear frequently in multiple biological networks. In this article, we review algorithms for discovering several types of frequent patterns defined on multiple biological networks: coherent dense subgraphs [12], frequent dense vertex-sets [32], generic frequent subgraphs [13], differential subgraphs [21] and recurrent heavy subgraphs [20]. Although the methods described in this paper are applicable to any type of genome-wide network, we shall demonstrate our algorithms using co-expression networks due to their wide availability. That is, we model each microarray dataset as a co-expression network, whose nodes represent genes and whose edges can be either weighted or unweighted. In a weighted co-expression network, the edge weights can be expression correlation coefficients. In an unweighted network, two genes are connected only if their expression correlation is higher than a given threshold. It is known that absolute expression values of a gene cannot be compared across microarray datasets generated by different platforms or in different labs, because systematic variations among datasets are often beyond the capability of statistical normalization. However, the expression correlations of a gene pair in different datasets are comparable because they are unitless measures. As co-expression networks are constructed from expression correlations of gene pairs, their comparisons are not affected by inter-dataset variations. Thus, modeling microarray datasets as co-expression networks provides an effective way to integrate a large number of microarray experiments conducted in different laboratories, at different times, and using different technology platforms.

Given K microarray datasets, we can construct K networks with the same node set but different edge sets. In a co-expression network, each gene occurs once and only once. The networks therefore have distinct node labels, and the problem of discovering frequent patterns in a set of these networks doesn't have "subgraph isomorphism" that is a well-known NP-hard problem. There are two classes of studies in the literature that are related to our problem: (1) "Frequent subgraph discovery with isomorphism in a set of small graphs" is one of most explored and studied topic in the field of data mining in recent years and has been applied to small-scale networks from diverse scientific domains, such as molecule structures and chemical compounds.

Table 1 Differences of three problems. Note that “dense cluster” is a subgraph whose nodes are densely interconnected, and “connected component” is a subgraph in which any two nodes are connected by paths

Problem	#Node	#Network	Subgraph topology	Isomorphism?
Frequent subgraph discovery without isomorphism (our problem)	Thousands or more	Hundreds or more	Dense cluster, connected component, differential subgraph	No
Frequent subgraph discovery with isomorphism	Tens	Hundreds or more	Connected component	Yes
Multiple network alignment	Thousands or more	Tens	Dense cluster, connected component	Yes

Its input data are a large number of small graphs with tens of nodes on average. A plenty of data mining algorithms [19] have been developed by employing different search strategies, such as A priori-based approach and pattern growth. (2) “Multiple network alignment in a small number of massive networks” focuses on comparing biological networks from different species [14, 16–18, 27]. Its task is to extract conserved subgraphs across networks of multiple species, where the nodes in different networks can have a many-to-many mapping based on the orthologous gene relationships, thus involving the “subgraph isomorphism” problem. Particularly, unlike the network alignment problem which compares networks from different species, the methods described here focus on comparing networks from the same species, but generated under different conditions. We list the differences between these two problems and our problem in Table 1. In summary, the main differences between our problem and other two problems are of “isomorphism” and “data scale and type”: i.e., our problem centers on a large number of massive networks without isomorphism, whereas other two classes of problems are devoted for small networks or small number of networks with isomorphism. Therefore, algorithms of those two classes of problems are not applicable to our problem.

We begin by defining a coherent dense subgraph pattern and describing its discovery algorithm. This section is followed by three other types of frequent patterns: frequent dense vertex-sets, generic frequent subgraphs, and differential subgraphs. As these patterns and algorithms were developed on unweighted networks, they cannot be easily extended to weighted networks. We then propose to use a mathematical concept “tensor” for modeling multiple weighted networks, and describe a tensor-based computational method to efficiently discover frequent patterns from a large collection of massive, weighted networks. In these sections, we provide evidence that these frequent network patterns are biologically meaningful in the form of gene function predictions, transcriptome to phenome mapping, and a high-order cooperativity analysis.

2 Coherent Dense Subgraph

Coherent dense subgraphs is a pattern that occurs in multiple unweighted networks. It satisfies two criteria: (1) the nodes of the subgraph are densely interconnected, and

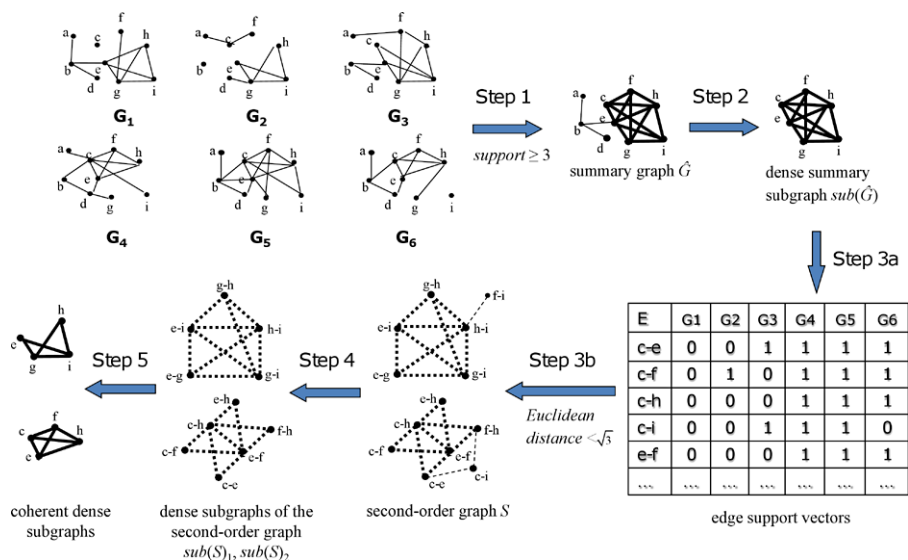


Fig. 1 CODENSE: an algorithm to discover coherent dense subgraphs across multiple networks (the dense subgraphs are marked with bold edges). The element of an edge occurrence profile is 1 if this edge occurs in the network, otherwise it is 0

(2) all of its edges should exhibit correlated occurrences or occur together among multiple networks.

Let $D = \{G_i = (V, E_i)\}$, where $i = 1, \dots, n$ and $E_i \subseteq V \times V$, denote a set of n undirected networks with unweighted edges and sharing the common node set V . To achieve the first criterion, the nodes in the pattern should be densely connected to each other in the “summary graph” $\hat{G} = (V, \hat{E})$, which shares the same node set V but contains only those edges present in at least k networks of D . A *second-order graph* S can be constructed to describe the correlated occurrences of edges and achieve the second criterion. In the second-order graph $S = (V \times V, E_s)$, each node represents an edge of D and two nodes of S are connected if their corresponding edges in D occur together in enough networks. Specifically, the co-occurrence of two edges in D can be measured by Euclidean distance or Pearson’s correlation of their edge occurrence profiles across networks in D . This process of constructing S is illustrated in Fig. 1. To contrast with the second-order graph, we term the original networks G_i first-order graphs. This use of the second-order graph is just one type of second-order analysis, a concept proposed in one of our previous publications [34]. If the first-order graphs G_i are large and dense, S will be impractically large. To more efficiently analyze D , we only construct second-order graphs S for densely connected subgraphs of the summary graph \hat{G} .

We have developed a scalable algorithm to discover coherent dense subgraphs [12]. It is based on two observations concerning the relationships between a coherent dense subgraph, the summary graph, and the second-order graph.

1. If a frequently occurring subgraph of D is dense, then it must also exist as a dense subgraph in the summary graph. However, a dense subgraph of the summary graph may be neither frequent nor dense in D .
2. If a subgraph is coherent (i.e., if its edges are strongly correlated in their occurrences across networks of D), then its second-order graph must be dense.

These two facts permit the mining of coherent dense subgraphs with reasonable computational cost. According to Observation 1, we can begin our search by finding all dense subgraphs of the summary graph. Then we can single out coherent subgraphs by examining their corresponding second-order graphs. Our algorithm, named **CODENSE**, consists of the five steps outlined below and illustrated in Fig. 1. In Steps 2, 4 and 5, we employ a mining algorithm that allows for overlapping dense subgraphs.

Step 1 CODENSE builds a summary graph by eliminating infrequent edges.

Step 2 CODENSE identifies dense subgraphs (which may overlap) in the summary graph. Although these dense subgraphs may not frequently occur in D , they are a superset of the true frequent dense subgraphs.

Step 3 CODENSE builds a second-order graph for each dense summary subgraph.

Step 4 CODENSE identifies dense subgraphs in each second-order graph S . A high connectivity among vertices in a second-order graph indicates that the corresponding edges have high similarity in their occurrences across the original graphs.

Step 5 CODENSE discovers the real coherent dense subgraphs. Although a dense subgraph $sub(S)$ found in Step 4 is guaranteed to have the co-occurrent edges in D , those edges may not form a dense subgraph in the original summary graph. To eliminate such cases, we convert the nodes in $sub(S)$ back to edges and apply the overlapping dense subgraph mining algorithm once more. The resulting subgraphs will satisfy both criteria for coherent dense subgraphs: (1) they are dense in many of the original networks, so all of their edges occur frequently; and (2) their edges are highly correlated or co-occur across the networks of D .

The software of this method is freely downloaded at <http://zhoulab.usc.edu/CODENSE/>.

We used co-expression networks derived from 39 yeast microarray datasets as a testing system for CODENSE. Each dataset contains the expression profiles of 6661 genes in at least eight experiments. These data were obtained from the Stanford Microarray Database [10] and the NCBI Gene Expression Omnibus [9]. The similarity between two gene expression profiles in a microarray dataset is measured by Pearson's correlation (denoted r). We apply the transformation $t = \sqrt{\frac{(n-2)r^2}{1-r^2}}$, and model t as a t -distribution with $n - 2$ degrees of freedom (n is the number of measurements used to compute r). We then construct an unweighted relation network for each microarray dataset, connecting two genes if their correlation is significant at the $\alpha = 0.01$ level. The summary graph \hat{G} is constructed by collecting edges with a support of at least six out of 39 graphs. At all steps where dense subgraph mining is performed, the density threshold is set to 0.4.

To assess the quality of the coherent dense subgraphs, we calculated the percentage of patterns whose gene cluster are functionally homogeneous. Based on the Gene

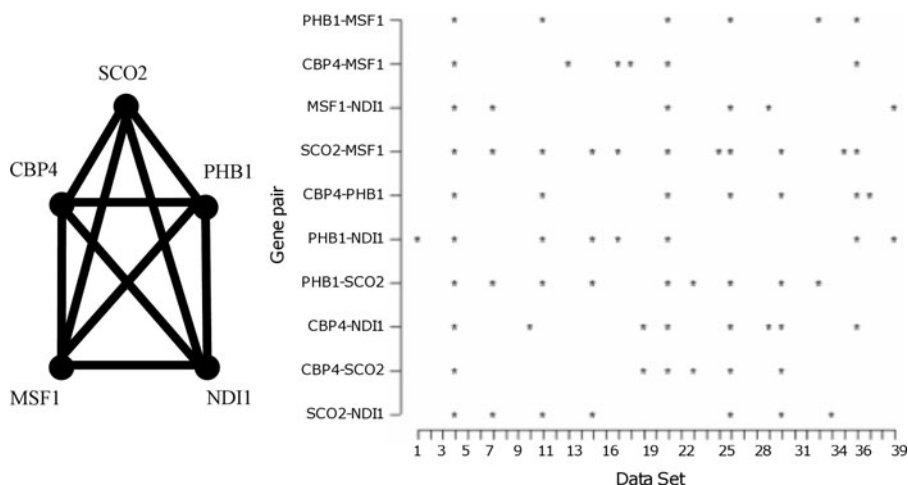


Fig. 2 The edge occurrence profiles of a five-gene clique in the summary graph. A *dot* represents the occurrence of an edge in a network

Ontology (GO) biological process annotations [3], we consider a gene cluster functionally homogeneous if (1) the functional homogeneity, as modeled by the hypergeometric distribution [30], is significant at $\alpha = 0.01$ level; and (2) at least 40% of its member genes with known annotations belong to a specific GO functional category.

Within the hierarchy of GO biological process annotations, we define *specific functions* to be those associated with GO nodes that are more than 5 levels below the root. CODENSE identified 770 clusters with at least 4 annotated genes. Of these clusters, 76% are functionally homogeneous. If we stop at Step 2 of the algorithm, obtaining dense subgraphs of the summary graph, only 42% are functionally homogeneous. This major improvement in performance can be attributed to the power of second-order clustering as a tool for eliminating subgraphs whose edges do not co-occur across networks. As an example of a spurious cluster, consider the five-gene clique in the summary graph composed of *MSF1*, *PHB1*, *CBP4*, *NDI1*, and *SCO2*, depicted in Fig. 2. These genes are annotated with a variety of functional categories such as ‘protein biosynthesis’, ‘replicative cell aging’ and ‘mitochondrial electron transport’, so the subgraph is not functionally homogeneous. As it turns out, although all edges of this clique occur in at least six networks, their co-occurrence is not significant across the 39 networks (Fig. 2). Analyzing clusters in the second-order network can reveal such pseudoclusters, providing more reliable results.

The large set of functionally homogeneous clusters identified by CODENSE provide a solid foundation for the functional annotation of uncharacterized genes. If a GO functional category is significantly overrepresented (Bonferroni-corrected hypergeometric p -value < 0.01) in a cluster, then we can confidently annotate unknown genes in the cluster with that function. To assess the prediction accuracy of our method, we employed a “leave-one-out” approach: a known gene is treated as unknown before analyzing the coherent dense subgraphs, then annotated based on the remaining known genes in the cluster. We consider a prediction correct if the lowest common ancestor of the predicted and known functional categories is five levels

below the root in the GO hierarchy. The annotated yeast genes encompass 160 functional categories at level 6 of the GO hierarchy. We predicted the functions of 448 known genes by this method, and achieved an accuracy of 50%. We also made predictions for 169 unknown genes, covering a wide range of functional categories.

3 Frequent Dense Vertex-Set

Although CODENSE has successfully identified coherent dense subgraphs across multiple biological networks, its criteria are too stringent to identify many potential frequent patterns. CODENSE requires that the entire edge set of a pattern show highly correlated recurrence across all the original networks. However, edge occurrences in a biological network can be distorted by measurement noise. In fact, any set of genes that is densely connected in a significant number of networks is likely to form a functional and transcriptional module, even if the edges differ from network to network. Such a gene set is another kind of frequent pattern and is worthy of attention. We call them “frequent dense vertex-sets” (FDVSs). FDVS has only one criterion: its vertices must be densely connected in at least k networks. The following data mining approach is designed to identify FDVSs [32].

Step 1 Construct summary graphs Given a collection of networks, remove all infrequent edges and then aggregate the networks to form a summary graph \hat{G} . To improve accuracy, this step can be further improved by organizing the original networks into groups, creating a summary graph for each group, then re-weighting each summary graph.

Step 2 Mine dense subgraphs from the summary graph Apply the overlapping dense subgraph mining algorithm to \hat{G} . This step yields a set of dense subgraphs.

Step 3 Refine Extract the truly frequent dense vertex-sets from each dense subgraph.

The entire mining process is illustrated in Fig. 3. The summary graph can be constructed by a procedure that is more complicated than the simple method introduced in Sect. 2, but also more effective. As shown in Fig. 3, first we divide the original networks into groups. Then we create a summary graph for each group. Next we re-weight each summary graph to become a *neighbor association summary graph*. This increases the signal-to-noise ratio and reduces the impact of noisy edges.

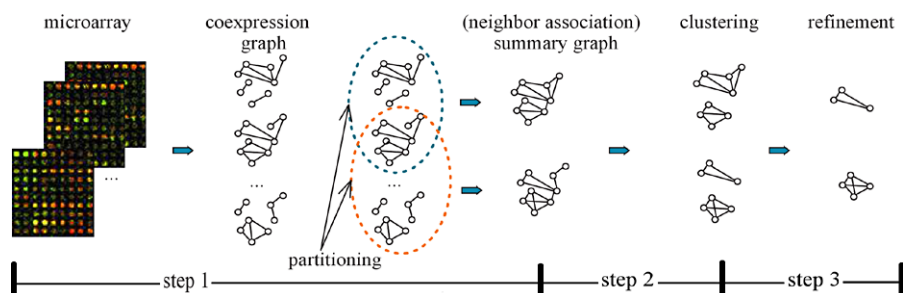


Fig. 3 The pipeline of our frequent dense vertex-set mining algorithm

We selected 105 human microarray datasets generated by the Affymetrix U133 and U95Av2 platforms. Each microarray dataset is modeled as a co-expression network following the method introduced in Sect. 2. In this study, only the most significant correlations with p -values less than 0.01 (the top 2%) are included in each network. Applying our approach to discovering frequent dense vertex-sets in these networks, we identified 4,727 recurrent co-expression clusters. Each cluster's density is greater than 0.7 in at least 10 supporting datasets. To assess the quality of the clusters, we tested their member genes for enrichment of the same bound transcription factor. The transcription factors to target gene relationships were ascertained through ChIP-Chip experiments, which contain 9,176 target genes for 20 TFs covering the entire human genome. A recurrent cluster is considered a potential transcriptional module if (1) $>75\%$ of its genes are bound by the same transcription factor, and (2) the enrichment of the particular TF in the cluster is statistically significant with a hypergeometric p -value <0.01 relative to its genome-wide occurrences. Among the identified clusters, 15.4% satisfied both criteria. This is a high hit rate, considering that we only tested for 1% of the approximately 2000 transcription factors estimated to exist in the human genome. On average, the permuted set of clusters was enriched only 0.2% for a common transcription factor. This result demonstrates that our approach can reliably reconstruct regulatory modules.

4 Generic Frequent Subgraph

Biological modules may be not only frequent dense subgraphs, but also frequent subgraphs with any topology (for example, a path). Such patterns are termed *generic frequent subgraphs*. If they occur in many networks, they can be easily differentiated from the background and have a large probability being a biological module.

The process of finding generic patterns begins with searching for frequent edge sets that are not necessarily connected. Then we to extract their connected components, which form generic frequent subgraphs [13]. The first step of discovering frequent edge sets can be formulated as a biclustering problem; the second step is a typical graph problem of detecting connected components. Given n unweighted biological networks, a matrix of edge occurrence profiles can be constructed where the rows are edges (i.e., node pairs), columns are networks, and each entry (1 or 0) indicates whether the edge appears in that network. An example is shown in Fig. 1. Thus, each frequent edge set corresponds to a submatrix with a high density of 1's that can be found by the biclustering method. We have developed a biclustering algorithm based on simulated annealing to discover frequent edge sets.

We collected 65 human microarray datasets, including 52 Affymetrix (U133 and U95 platforms) datasets and 13 cDNA datasets from the NCBI Gene Expression Omnibus [9] and SMD [10] databases (December 2005 versions). Each microarray dataset is modeled as a co-expression network following the procedure introduced in Sect. 2. The biclustering algorithm described above yields a total of 1,823,518 patterns (modules) which occur in at least five networks. After merging patterns with similar topologies and dataset recurrence, we are left with 143,400 distinctive patterns involving 2769 known and 1054 unknown genes. The numbers of genes in the patterns vary from 4 to 180.

We define a module to be functionally homogeneous if the hypergeometric p -value after Bonferroni correction is <0.01 . Among the identified network patterns, 77.0% are functionally homogeneous by this standard. We found that patterns which occur more frequently are more likely to be functionally homogeneous. This observation supports our basic motivation for using multiple microarray datasets to enhance functional inferences. By considering pattern recurrence across many networks, we can enhance the signal of meaningful structures. The identified network modules have a wide range of topologies. In fact, 24% of them have connectivities <0.5 .

To explore relationships other than co-expression, we resort to the only available large-scale source: protein-interaction data. We retrieved human protein-interaction information from the EBI (European Bioinformatics Institute)/IntAct database (version 2006-10-13) [11]. For each of the 143,400 detected patterns, we tested whether protein interactions were over-represented in member genes compared to all human genes using the hypergeometric test to evaluate significance. A total of 60,556 (22.44%) patterns were enriched in protein interactions at a p -value of 0.001. This shows that genes belonging to a module are much more likely to encode interacting proteins. Interestingly, many of the protein-interaction-enriched network modules also fall into functional categories such as protein biosynthesis, DNA metabolism, and so on. There are even many cases where the interacting protein pairs are not co-expressed.

Given a network pattern identified, we can also predict gene functions through a graph method that can fully explore the topology of network pattern. Our method is based on the principle of “guilt by association” and random walks that can efficiently estimate the degree of association or network topology score between two genes. To further improve the method, we included attributes (such as recurrence, density, the percentage of unknown genes, etc.) other than the network topology scores of a network pattern in the final prediction. We used a random forest method¹ to determine whether function assignments based on the network topology score are robust. For each of the 143,400 recurrent network patterns, we identified the function of each member gene with the maximum network topology score. We then trained a random forest and made functional predictions for 779 known and 116 unknown genes with 70.5% accuracy.

5 Differential Pattern

Suppose that a set of biological networks is divided into two classes, e.g. those related to a specific disease and those obtained under normal or unrelated conditions. It is then interesting to identify network patterns whose rate of occurrence differs significantly between the two classes. In fact, it has become clear that many complex conditions such as cancer, autoimmune disease, and heart disease are characterized by specific gene network patterns. We have designed an integrative approach to inferring network modules specific to a phenotype [21]. A series of microarray datasets modeled as co-expression networks is labeled with phenotypic information such as

¹ A random forest is a collection of tree-structured classifiers [5].

the type of biological sample, a disease state, or a drug treatment. For each phenotype, we can partition the microarray datasets into a positive class of datasets appropriately annotated with the phenotype, and a background class containing the rest of the datasets. We have designed a graph-based simulated annealing approach [15] to efficiently identify groups of genes that form dense subnetworks preferentially and repeatedly in a phenotype's positive class.

Consider a collection of unweighted networks $D = \{G_1, G_2, \dots, G_n\}$, where each network $G_i = (V, E_i)$ is annotated with a set of phenotypes. For each phenotype, we partition D into a positive class D_P and a background class $D_P^c = D \setminus D_P$ as described above. Our problem is to identify groups of genes which form dense subgraphs repeatedly in the positive class but not in the background class. More specifically, we aim to satisfy three criteria: first, a gene set must be densely connected in multiple networks; second, the annotations of these networks must be enriched in a specific phenotype; and third, the gene set meeting the first two criteria must be as large as possible. Put simply, the problem is to find modules with three qualities: density, phenotype specificity, and size.

For the first criterion, we might consider a gene set to be densely connected if its density is larger than a hard threshold (typically 0.66). However, because we will use simulated annealing as the optimization method, hard thresholds prevent the algorithm from accepting unfavorable but useful intermediate states that may be unfavorable. We therefore design an objective function f_{dens} with a soft threshold, where unfavorable values of the density increase the cost exponentially. This objective function is defined in (2) below. The other two criteria also use soft thresholds in their objectives. The second criterion (specificity) states that given a phenotype, we wish to find dense gene sets that occur frequently in the positive class but infrequently in the background class. The specificity objective function is defined in (3) below. It uses the hypergeometric test to quantify the significance of phenotype enrichment and favors low p -values, again at an exponential rate. For simplicity and computational considerations, we limited the size of the module to 30 genes. We believe this to be an ample margin for phenotypically relevant gene sets. Equation (1) shows the size objective function, which contains both a linear component (first term) and an exponential component (second term). The exponential component sets a strong preference for low sizes (4–5 nodes), but the linear component continues to reward size increases above this soft threshold.

We supplemented the three main objectives with a fourth: the *density differential* defined in (4). This term compliments the density and specificity objectives by comparing the average density of the cluster in the background datasets to its density in the phenotype datasets. The rationale behind this term is as follows. Since the specificity objective function only takes a state's active datasets as arguments, the transition to a neighboring state may yield a sudden change in the specificity energy because its active datasets are different. However, many neighboring states can have subtle changes in the density distribution among the active and inactive datasets that is not captured by the density and specificity functions alone. The density differential function is therefore designed to reward these subtle density changes, helping direct the simulated annealing process toward more phenotype-specific clusters. We found that using the density differential in combination with the specificity and density criteria caused the algorithm to converge faster, and resulted in better clusters.

The individual objective functions have the following forms:

$$f_{\text{size}}(x) = \exp\left\{-\alpha\left(\frac{|x|}{\gamma} - o_s\right)\right\} \quad (1)$$

$$f_{\text{dens}}(x) = \exp\left\{-\alpha\left(\min_{i \in D_A}(\delta_i(x)) - o_\delta\right)\right\} \quad (2)$$

$$f_{\text{spec}}(x) = \log(\mathbb{P}(Y \geq |D_A \cap D_P|)) \quad (3)$$

$$f_{\text{diff}}(x) = \left(\frac{1}{|D_P^c|} \sum_{i \in D_P^c} \delta_i(x) - \frac{1}{|D_P|} \sum_{i \in D_P} \delta_i(x)\right) \quad (4)$$

where: D_P is the set of networks annotated with the current phenotype; D_A is the set of networks in which the gene cluster is dense; and $Y \sim \text{hypergeometric}(|D_A|, |D_P|, |D_P^c|)$.

The exponential components of these functions prevent the simulated annealing algorithm from settling on an extreme case with just one of the desired qualities (such as a very specific triangle, which is always very dense and small). Improvements to such cases are always rewarded, however, and they are accepted as intermediate steps with good probability. We selected the parameters $\alpha = 20$, $\gamma = 30$, $o_\delta = 0.85$, and $o_s = 0.2$ based on our simulation results comparing biologically validated clusters with clusters arising from random chance.

We combined the four objective functions into a single function using a weighted sum $f(x) = w_1 f_{\text{size}}(x) + w_2 f_{\text{dens}}(x) + w_3 f_{\text{spec}}(x) + w_4 f_{\text{diff}}$. The key difficulty with this approach is determining the weights. In previous studies, this has been accomplished empirically [8]. We do the same, for the following reasons. First, we are interested in finding a single optimal or near-optimal objective function, rather than exploring the extremes of each term. Second, the overall effectiveness of our algorithm turns out to be consistent for a wide range of weights. Finally, although we chose weights based on the algorithm's performance with simulated data, it also behaved well on real data. The weights for size, density, specificity, and density differential are 0.05, 0.05, 5, and 50, respectively.

We used simulated annealing to optimize the objective function for identifying differential patterns. This well-established stochastic algorithm has been successfully applied to many other NP-complete problems [28]. After obtaining differential modules, we merged redundant gene clusters with intersections/unions greater than 0.8.

We selected microarray datasets from NCBI's Gene Expression Omnibus [9] that met the following criteria: all samples were of human origin, the dataset had at least 8 samples (a minimum for accurate correlation estimation), and the platform was either GPL91 (Affymetrix HG-U95A) or GPL96 (Affymetrix HG-U133A). Throughout this study, we only considered the 8,635 genes shared by both platforms (and therefore all datasets). All 136 datasets meeting these criteria on 28 Feb 2007 were used for the analysis described hereafter.

We determined the phenotypic context of a microarray dataset by mapping the Medical Subject Headings (MeSH) of its PubMed record to UMLS concepts. This process is more refined than scanning the abstract or full text of the section, and in practice results in much cleaner and more reliable annotations [6, 7]. UMLS is

the largest available compendium of biomedical vocabulary, spanning approximately one million interrelated concepts, including diseases, treatments, and phenotypic concepts at different levels of resolution (molecules, cells, tissues and whole organisms). In order to infer high-order links between datasets, we annotated datasets with their matched UMLS concepts and, in addition, all their ancestor concepts. This procedure resulted in a total of 467 annotations, of which 80 mapped to more than five datasets, or 60 after merging annotations that mapped onto identical sets of datasets.

Within each dataset, we used the jackknife Pearson correlation as a measure of similarity between two genes (this is the minimum of all possible leave-one-out Pearson correlations). To create the co-expression network, we selected a cutoff that kept the 150,000 strongest correlations (0.4% of the total number of gene pairs: $\binom{8635}{2} \approx 3.73 \times 10^7$). This choice was motivated by an analysis of the statistical distribution of pairwise correlations, which we do not detail here.

We applied our simulated annealing approach to all 136 microarray datasets covering 42 phenotype classes. These classes included a useful diversity of diseases (e.g. leukemia, myopathy, and nervous system disorders) and tissues (e.g. brain, lung, and muscle). The procedure described above identified 118,772 clusters that satisfied our criteria for a concept-specific co-expression cluster. The number of clusters we found for a given phenotype increased with the number of datasets annotated with it: most of the phenotypes with only a few associated datasets yielded few clusters. The most represented phenotype we studied was “nervous system disorders”, with 15 associated datasets and 22,388 differential clusters.

We assessed the functional homogeneity of a cluster by testing for enrichment of its genes in specific Gene Ontology [3] biological process terms. If a cluster is enriched in a GO term with a hypergeometric p -value less than 0.01, we declare the cluster functionally homogeneous. Of the 118,772 clusters found, 78.98% were functionally homogeneous. An advantage of our approach is demonstrated by this validation: since we only looked for clusters specific to only subsets of all our datasets, we were less likely than previous studies to detect constitutively expressed clusters, such as those consisting of ribosomal genes or genes involved in protein synthesis.

While the GO database provides information on gene functions, it fails to describe their phenotypic implications. To map individual genes to phenotypes, we used GeneRIF [22]. The GeneRIF database contains short statements derived directly from publications, describing the functions, processes, and diseases in which a gene is implicated. We annotated genes with phenotypes by mapping the GeneRIF notes to the UMLS metathesaurus terms, as we did with the dataset MeSH headings. Similar to GO annotations, we then assessed the *conceptual homogeneity* of gene clusters in specific UMLS keywords with the hypergeometric test, enforcing a minimum p -value of 0.01. The proportion of modules that were conceptually homogeneous was 48.3%. The fact that clusters show less conceptual homogeneity than functional homogeneity is likely due to the scarcity of GeneRIF annotations. There are cases, however, in which GeneRIF performs very well. For example, many cancer-related phenotypes such as “Neoplasm Metastasis” and “Neoplastic Processes” show higher GeneRIF homogeneity, which could be attributed to the abundance of related literature. The functional and conceptual homogeneity of clusters derived from different phenotype classes is summarized in Fig. 4.

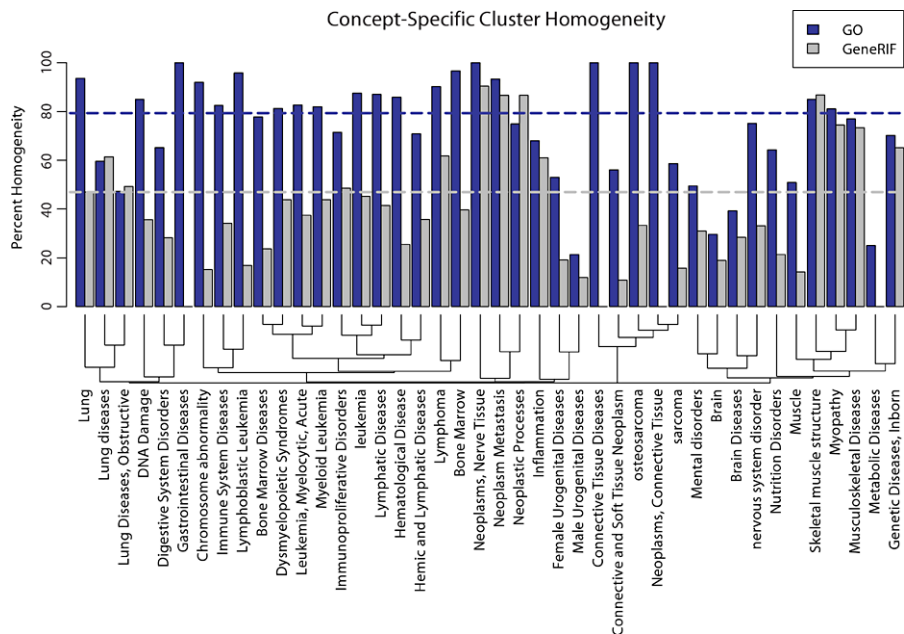


Fig. 4 (Color online) Cluster homogeneity by phenotype. For each phenotype, the proportion of clusters that are significantly enriched (p -value < 0.01) for a GO biological process (blue) or a GeneRIF UMLS concept (gray). The dotted lines show the overall homogeneity for all clusters. The dendrogram shows the distance between phenotypes in terms of dataset overlap

6 Frequent Patterns in Weighted Networks—Tensor Model

In the previous sections, we approached the problems of identifying frequent patterns in multiple unweighted networks through a series of heuristic, graph-based, data mining algorithms. While useful, these methods still face two major limitations. (1) The general strategy is a stepwise reduction of the large search space, but each step involves one or more arbitrary cutoffs. In addition to these, we have the crucial initial cutoff that transforms continuous measurements (e.g. expression correlations) into unweighted edges. The *ad hoc* nature of these cutoffs has been a major criticism directed at this body of work. (2) These algorithms cannot be easily extended to weighted networks. Most graph-based approaches to analyzing multiple networks are restricted to unweighted networks, and weighted networks are often perceived as harder to analyze [23]. However, weighted networks are obviously more informative than their unweighted counterparts. Transforming weighted networks into unweighted networks by dichotomizing weighted edges with a threshold obviously leads to information loss [26], and if there is no reasonable way to choose the threshold, this loss cannot be controlled. This section presents a computational method of analyzing multiple weighted networks that overcomes both of these issues.

Given m networks with the same n nodes but different topologies, we can represent the whole system as a third-order tensor or three-dimensional array $\mathcal{A} = (a_{ijk})_{n \times n \times m}$ (Fig. 5 shows an example). Each element a_{ijk} is the weight of the edge between

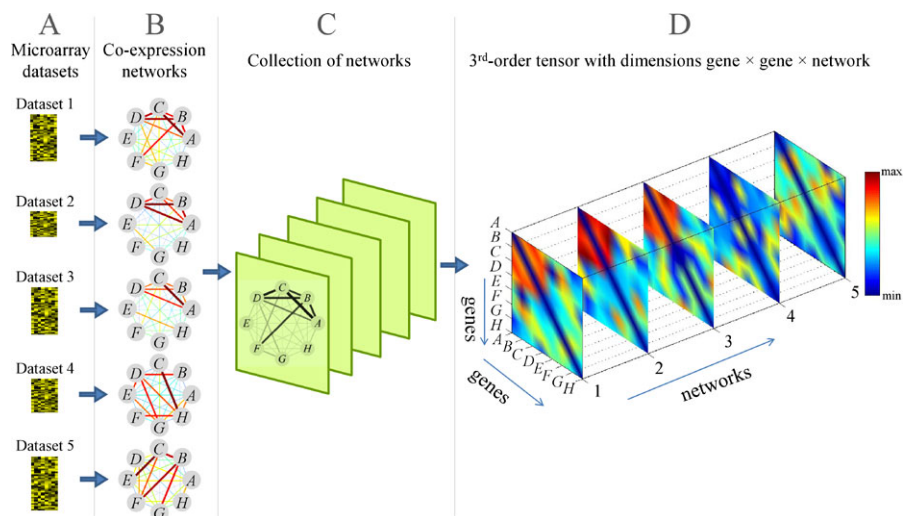


Fig. 5 (Color online) Illustration of the tensor representation for multiple networks and a recurrent heavy subgraph. **(A)** Microarray datasets are modeled as **(B)** a collection of co-expression networks. **(C)** These co-expression networks can be “stacked” into **(D)** a third-order tensor such that each slice represents the adjacency matrix of one network. The weights of edges in the co-expression networks and their corresponding tensor elements are indicated by the color scale to the right of the figure. In **(D)**, after reordering the tensor using the gene and network membership vectors, it becomes clear that the subtensor in the top-left corner (formed by genes A, B, C, D in networks 1, 2, 3) corresponds to a recurrent heavy subgraph

nodes i and j in the k th network. By representing a set of networks in this fashion, we can reformulate a discrete graph problem as a continuous optimization problem. This shift of perspective grants us access to a wealth of numerical methods. Advanced continuous optimization techniques require very few ad hoc parameters, in contrast with most heuristic graph algorithms.

We developed a tensor-based computational method [20] to identify a frequent pattern in multiple *weighted* networks, a so-called *recurrent heavy subgraph* (RHS). A *heavy subgraph* (HS) is a subset of heavily interconnected nodes in a single network. We define a RHS as a HS that appears in a subset of multiple networks. The nodes of a RHS must be the same in each occurrence, but the edge weights may vary between networks. As shown in Fig. 5, a RHS intuitively corresponds to a heavy region of the tensor (a heavy subtensor). Therefore, we formulate the RHS discovery problem as an optimization problem based on tensor representation. In more detail, any RHS can be described by two membership vectors: (i) the *gene membership vector* $\mathbf{x} = (x_1, \dots, x_n)^T$, where $x_i = 1$ if gene i belongs to the RHS and $x_i = 0$ otherwise; and (ii) the *network membership vector* $\mathbf{y} = (y_1, \dots, y_m)^T$, where $y_j = 1$ if the RHS appears in network j and $y_j = 0$ otherwise. The summed weight of all edges in the RHS is

$$H_{\mathcal{A}}(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^m a_{ijk} x_i x_j y_k \quad (5)$$

Note that only the weights of edges a_{ijk} with $x_i = x_j = y_k = 1$ are counted in H_A . Thus, $H_A(\mathbf{x}, \mathbf{y})$ measures the “heaviness” of the RHS defined by \mathbf{x} and \mathbf{y} . Discovering a recurrent heavy subgraph can be formulated as a discrete combinatorial optimization problem: *among all RHSs of fixed size (K_1 member genes and K_2 member networks), we look for the heaviest*. More specifically, this is an integer programming problem: we are looking for the binary membership vectors \mathbf{x} and \mathbf{y} that jointly maximize H_A under the constraints $\sum_{i=1}^n x_i = K_1$ and $\sum_{j=1}^m y_j = K_2$. However, the integer programming problem is NP-hard and has two parameters (K_1 and K_2) that are difficult to estimate. We can instead solve a continuous optimization problem with the same objective, simply by relaxing the integer constraints to continuous constraints. It is formally expressed as follows:

$$\begin{aligned} & \max_{\mathbf{x} \in \mathbb{R}_+^n, \mathbf{y} \in \mathbb{R}_+^m} H_A(\mathbf{x}, \mathbf{y}) \\ & \text{subject to} \quad \begin{cases} \alpha \|\mathbf{x}\|_p + (1 - \alpha) \|\mathbf{x}\|_2 = 1 \\ \|\mathbf{y}\|_q = 1 \end{cases} \end{aligned} \quad (6)$$

where \mathbb{R}_+ is a non-negative real space, and $\|\mathbf{x}\|_p = (\sum_i |x_i|^p)^{1/p}$ is a vector norm. After studying the performance of the algorithm on simulated data, we adopted the parameters $p = 0.8$, $\alpha = 0.2$, and $q = 10$. These equations define a tensor-based formulation of the RHS identification problem. The RHSs can be intuitively obtained by including those genes and networks with large membership values. A pair of gene and network membership vectors $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, i.e., the solution of (6), can result in multiple RHSs whose “heaviness” is greater than a specified value (i.e., \geq a threshold). Here, the “heaviness” of a RHS is defined as the average weight of all edges in the RHS. In practice, we rank genes and networks in decreasing order of their membership values in $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$, then two overlapping RHSs are extracted: the RHS with the smallest number of top-ranking genes (but at least the minimum number required) that appears most often in top-ranking networks, and the one with the maximum number of genes that appears in the smallest number of top-ranking networks (but again, at least the minimum number required). After discovering a RHS, we can mask its edges in those networks where it occurs (replacing those elements of the tensor with zeroes) and optimize (6) again to search for the next heaviest RHS. The software of this method is freely downloaded at <http://zhoulab.usc.edu/tensor/>.

Since the constraint in (6) is non-convex, our tensor method employs a recently proposed optimization framework known as multi-stage convex relaxation [33] which has good numerical properties. To further speed computation, with an acceptable and controllable loss of accuracy, we use edge sampling techniques. This approach has been shown to provide an efficient approximation to many graph problems [1, 29]. We adopt the random-sampling-based single-pass sparsification procedure introduced in [2].

We selected every microarray dataset from NCBI’s Gene Expression Omnibus that met the following criteria: all samples were of human origin; the dataset had at least 20 samples, to guarantee robust estimates of expression correlations. The 130 datasets that met these criteria on 28 January 2008 were used for the analysis described hereafter. Each microarray dataset is modeled as a co-expression network

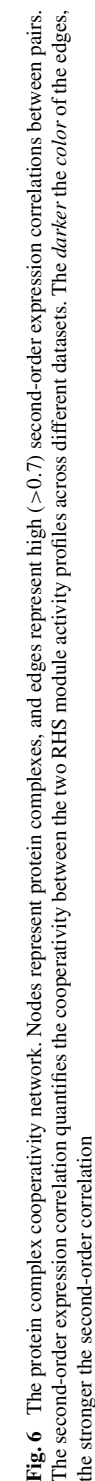
by using jackknife Pearson correlation. To make the correlation estimates comparable across datasets, we “normalize” correlations by using the procedure introduced in [31]. Finally, the absolute value of the normalized correlation is used as the edge weight of co-expression networks.

After applying our method to 130 microarray datasets generated under various experimental conditions, we identified 4,327 RHSs. Each RHS contains ≥ 5 member genes, appears in ≥ 5 networks, and has a “heaviness” ≥ 0.4 . The average size of these patterns is 8.5 genes, and the average recurrence is 10.1 networks. To assess the biological significance of the identified RHSs, we evaluate the extent to which these RHSs represent functional modules and protein complexes.

We evaluated the functional homogeneity of genes in each RHS using the Gene Ontology (GO) [3] biological process terms associated with ≤ 500 genes. If the member genes of a RHS are found to be significantly enriched in a GO term with a q -value < 0.05 (the q -value is the hypergeometric p -value after a False Discovery Rate multiple testing correction), we declare it to be functionally homogeneous. We found that 39.9% of the RHSs were functionally homogeneous in this sense. In an ensemble of randomly generated RHSs having the same size distribution as our RHSs, only 1.2% of them were functionally homogeneous. Not only RHSs with greater heaviness, but also those with more frequent recurrence among datasets, are more likely to be functionally homogeneous. For example, 40%/71%/90%/98% of the patterns appearing in 5/10/20/30 datasets, respectively, were functionally homogeneous. In contrast, only 4.30% of patterns with a single occurrence were functionally homogeneous. This strong dependence highlights the importance of pursuing integrative analysis of *multiple* networks.

We applied our method to the Comprehensive Resource of Mammalian protein complexes (CORUM) database (September 2009 version) [25]. 27.8% of RHSs are significantly enriched with a q -value < 0.05 in genes belonging to a protein complex compared to only 0.16% of randomly generated patterns. The protein complexes are diverse and have a variety of functions. For example, a series of RHSs covered different parts of large complexes such as ribosome (both the small 40s unit and the large 60s unit), proteasome (the 20s core unit and the 19s regulatory unit), and spliceosome.

The discovery of numerous RHSs spanning a variety of experimental or disease conditions enables us to investigate high-order coordination among those modules. We applied our previously proposed *second-order analysis* [34] to study cooperativity among the protein complexes. We define a first-order expression analysis as the extraction of patterns from one microarray dataset, while a second-order expression analysis studies the correlated occurrences of patterns (e.g. heavy subgraph recurrence) across multiple datasets. For each identified RHS, we constructed a vector \mathbf{h} of length n storing its heaviness factors in the n datasets. The heaviness factor is the module’s first-order average expression correlation, so \mathbf{h} can be interpreted as the activity profile of the module across different datasets. To quantify the cooperativity between two modules, we calculated the correlation between their vectors \mathbf{h} . This quantity is denoted the *second-order expression correlation* of two modules. Figure 6 shows a cooperativity map of all protein complexes represented by the RHSs that have high (> 0.7) second-order expression correlations with at least one other protein complex. The most striking feature of this map is a large and very heavily connected sub-network of 32 complexes, all involved in the cell cycle. Seventeen of the complexes



(including *CDC2_Complex*, *CCNB2_CDC2_Complex*, *CDK4_Complex*, *Chromosomal_Passenger_Complex*, and *Emerin_Complex_24*) form a tight core with very strong second-order expression correlations (≥ 0.95). This structure highlights the strict transcription regulation of cell cycle processes. Two other dense subnetworks contain protein complexes involved in the respiratory chain and others involved in translation (e.g. the ribosomal complex, the *NOP56* associated pre-RNA complex, and *TRBP* complex associated with miRNA dicing). Numerous other protein complexes (e.g. the *FIB*-associated complex and the *CCT* complex) connect these dominant subnetworks or supercomplexes into an integrated network. Thus, our approach not only provides a comprehensive catalogue of modules that are likely to represent protein complexes, but also the very first systematic view of how protein complexes dynamically coordinate to carry out major cellular functions. That is, by integrating data generated under a variety of conditions, we have gained a glimpse into the activity organization chart of the proteome.

7 Conclusion

Biological network data are rapidly accumulating for a wide range of organisms under various conditions. The integrative analysis of multiple biological networks is a powerful approach to discover meaningful patterns, including subtle structures and relationships that could not be discovered in a single network. In this paper, we proposed several novel types of frequently occurring patterns and described algorithms to discover them. We also demonstrated that the identified patterns can facilitate functional discovery, regulatory network reconstruction, and phenotype characterization. Although we used co-expression networks as examples throughout this work, our methods can be applied to other types of relational graphs for pattern discovery. New challenges will arise as the quantity and complexity of biological network data continue to increase. The wealth of biological data will certainly push the scale and scope of graph-based data mining to the next level.

Acknowledgements The work presented in this paper was supported by National Institutes of Health Grants R01GM074163 and NSF Grant 0747475.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Achlioptas D, McSherry F (2007) Fast computation of low-rank matrix approximations. *J ACM* 54(2):9
2. Arora S, Hazan E, Kale S (2006) A fast random sampling algorithm for sparsifying matrices. In: Approximation, randomization, and combinatorial optimization. Algorithms and techniques. Springer, Berlin, pp 272–279
3. 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 (2000) Gene ontology: tool for the unification of biology. The gene ontology consortium. *Nat Genet* 25(1):25–29

4. Barabasi A, Oltvai Z (2004) Network biology: understanding the cell's functional organization. *Nat Rev Genet* 5(2):101–113
5. Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
6. Butte AJ, Chen R (2006) Finding disease-related genomic experiments within an international repository: first steps in translational bioinformatics. In: *AMIA Annual Symposium proceedings*, pp 106–110
7. Butte AJ, Kohane IS (2006) Creation and implications of a phenome-genome network. *Nat Biotechnol* 24(1):55–62
8. Collette Y, Siarry P (2003) *Multiobjective optimization: principles and case studies*. Springer, Berlin
9. Edgar R, Domrachev M, Lash AE (2002) Gene expression omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res* 30(1):207–210
10. Gollub J, Ball CA, Binkley G, Demeter J, Finkelstein DB, Hebert JM, Hernandez-Boussard T, Jin H, Kaloper M, Matese JC, Schroeder M, Brown PO, Botstein D, Sherlock G (2003) The Stanford microarray database: data access and quality assessment tools. *Nucleic Acids Res* 31(1):94–96
11. Hermjakob H, Montecchi-Palazzi L, Lewington C, Mudali S, Kerrien S, Orchard S, Vingron M, Roechert B, Roepstorff P, Valencia A, Margalit H, Armstrong J, Bairoch A, Cesareni G, Sherman D, Apweiler R (2004) IntAct: an open source molecular interaction database. *Nucleic Acids Res D* 32:452–455
12. Hu H, Yan X, Huang Y, Han J, Zhou XJ (2005) Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics* 21(Suppl 1):213–221
13. Huang Y, Li H, Hu H, Yan X, Waterman MS, Huang H, Zhou XJ (2007) Systematic discovery of functional modules and context-specific functional annotation of human genome. *Bioinformatics* 23(13):222–229
14. Kelley B, Sharan R, Karp R, Sittler T, Root D, Stockwell B, Ideker T (2003) Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proc Natl Acad Sci USA* 100(20):11394–11399
15. Kirkpatrick S, Gelatt C, Vecchi M (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
16. Koyutürk M, Grama A, Szpankowski W (2004) An efficient algorithm for detecting frequent subgraphs in biological networks. *Bioinformatics* 20(Suppl 1):200–207
17. Koyutürk M, Kim Y, Subramaniam S, Szpankowski W, Grama A (2006) Detecting Conserved Interaction Patterns in Biological Networks. *J Comput Biol* 13(7):1299–1322
18. Koyutürk M, Kim Y, Topkara U, Subramaniam S, Szpankowski W, Grama A (2006) Pairwise alignment of protein interaction networks. *J Comput Biol* 13(2):182–199
19. Krishna V, Suri NNR, Athithan G (2011) A comparative survey of algorithms for frequent subgraph discovery. *Curr Sci* 100(2):190–198
20. Li W, Liu CC, Zhang T, Li H, Waterman MS, Zhou XJ (2011) Integrative analysis of many weighted co-expression networks using tensor computation. *PLoS Comput Biol*. doi:[10.1371/journal.pcbi.1001106](https://doi.org/10.1371/journal.pcbi.1001106)
21. Mehan MR, Nunez-Iglesias J, Kalakrishnan M, Waterman MS, Zhou XJ (2009) An integrative network approach to map the transcriptome to the phenome. *J Comput Biol* 16(8):1023–1034
22. Mitchell JA, Aronson AR, Mork JG, Folk LC, Humphrey SM, Ward JM (2003) Gene indexing: characterization and analysis of nlm's generifs. In: *AMIA Annual Symposium proceedings*, pp 460–464
23. Newman MEJ (2004) Analysis of weighted networks. *Phys Rev E* 70(5):056,131
24. Papin J, Price N, Wiback S, Fell D, Palsson B (2003) Metabolic pathways in the post-genome era. *Trends Biochem Sci* 28(5):250–258
25. Ruepp A, Waegle B, Lechner M, Brauner B, Dunger-Kaltenbach I, Fobo G, Frishman G, Montrone C, Mewes H (2010) CORUM: the comprehensive resource of mammalian protein complexes–2009. *Nucleic Acids Res* 38:D497–D501
26. Serrano MA, Boguñá M, Vespignani A (2009) Extracting the multiscale backbone of complex weighted networks. *Proc Natl Acad Sci USA* 106(16):6483–6488
27. Sharan R, Suthram S, Kelley RM, Kuhn T, McCuine S, Uetz P, Sittler T, Karp RM, Ideker T (2005) Conserved patterns of protein interaction in multiple species. *Proc Natl Acad Sci USA* 102(6):1974–1979
28. Suman B, Kumar P (2006) A survey of simulated annealing as a tool for single and multiobjective optimization. *J Oper Res Soc* 57(10):1143–1160
29. Tsay AA, Lovejoy WS, Karger DR (1999) Random sampling in cut, flow, and network design problems. *Math Oper Res* 24(2):383–413

30. Wu LF, Hughes TR, Davierwala AP, Robinson MD, Stoughton R, Altschuler SJ (2002) Large-scale prediction of *saccharomyces cerevisiae* gene function using overlapping transcriptional clusters. *Nat Genet* 31(3):255–265
31. Xu M, Kao M, Nunez-Iglesias J, Nevins J, West M, Zhou X (2008) An integrative approach to characterize disease-specific pathways and their coordination: a case study in cancer. *BMC Genomics* 9(Suppl 1):S12
32. Yan X, Mehan MR, Huang Y, Waterman MS, Yu PS, Zhou XJ (2007) A graph-based approach to systematically reconstruct human transcriptional regulatory modules. *Bioinformatics* 23(13):577–586
33. Zhang T (2010) Analysis of multi-stage convex relaxation for sparse regularization. *J Mach Learn Res* 11:1081–1107
34. Zhou X, Kao M, Huang H, Wong A, Nunez-Iglesias J, Primig M, Aparicio O, Finch C, Morgan T, Wong W et al (2005) Functional annotation and network reconstruction through cross-platform integration of microarray data. *Nat Biotechnol* 23:238–243