Finding Biologically Accurate Clusterings in Hierarchical Decompositions Using the Variation of Information

Saket Navlakha, James White, Niranjan Nagarajan, Mihai Pop, and Carl Kingsford*

Center for Bioinformatics and Computational Biology, Institute for Advanced Computer Studies, and Department of Computer Science, University of Maryland, College Park

Abstract

Hierarchical clustering is a popular method for grouping together similar items based on a distance measure between them. These clusters can be used to infer annotations for uncharacterized items. However, in many cases, annotation information for some elements is known beforehand. We present a novel approach for decomposing a hierarchical clustering into the optimal clusters that match a set of known annotations, as measured by the variation of information metric. Our approach is general, and we apply it to two biological domains: finding protein complexes within protein interaction networks and identifying species within metagenomic DNA samples. For both applications, we test the quality of our clusters by using them to predict complex and species membership. We find that our approach generally outperforms the commonly used heuristic methods.

1. Introduction

Hierarchical (or agglomerative) clustering is an important tool in many applications. One application where it has been particularly useful is predicting membership in complexes for proteins using protein-protein interaction (PPI) networks. High-throughput experimental protocols are producing information on thousands of PPIs [53]. Embedded within the networks revealed by these interactions are protein complexes, stable groups of interacting proteins that perform some biological function in the cell. Complex membership is known for some proteins, but even for well-studied species like *S. cerevisiae*, 70-80% of proteins have no complex annotation according to MIPS [18]. Consequently, computational methods for determining to which complexes each protein belongs have recently been developed (e.g. [3, 30, 34, 37, 52]). A common approach to this problem is to identify clusters in the network [2, 4, 33, 34, 36, 49]. Often these clusters are detected by hierarchically clustering the graph [1, 5, 40, 41] based on a topological distance measure such as the Czekanowski-Dice [5] or Jaccard [20] distances. Complexes are then transferred to unannotated proteins by considering common known annotations within their clusters [2, 27, 34]. This leads to the following computational problem:

Problem 1 (Predicting Protein Complexes). *Given a hierarchical clustering of a PPI network for which protein complex annotations are known for some of the proteins, predict complex membership for the unannotated proteins.*

A second application of hierarchical clustering is predicting bacterial species for uncharacterized DNA sequences obtained from metagenomic and environmental samples [42, 45, 51]. In the expanding field of metagenomics, the composition of microbial communities is examined by sampling DNA from the environment. A typical diversity study involves targeted 16S rRNA gene sequencing using universal primers, a method that has successfully been used to describe bacterial communities in environments ranging from the ocean to soil to the human gut [12, 15, 43]. The standard methodology for 16S sequence analysis begins with a multiple sequence alignment containing both the environmental samples and several sequences of known origin. An approximate evolutionary distance is computed between every pair of sequences using a distance correction measure such as Jukes-Cantor [22], Kimura 2-parameter [26], or Felsenstein-84 [14]. A

*Corresponding author, carlk@cs.umd.edu

hierarchical clustering is then created from these distances, which is analyzed to identify which operational taxonomic units (OTUs; the more precise analog of species in the bacterial world) are present in the sample. Thus, the approach to this problem is similar to that for complex prediction from PPI networks: uncharacterized sequences are clustered (along with some sequences from known species), and are then assigned to species based on annotated sequences nearby in the clustering. By also approximating the species-level composition of a microbial community, comparisons can be made of the wealth of organisms in different environments, thus leading to estimations of the overall richness and diversity of communities. The accuracy of this analysis is vital for researchers examining environments with unknown composition. This leads to the following computational problem:

Problem 2 (Predicting Species for Uncharacterized DNA). *Given a hierarchical clustering of DNA sequences, some of which are derived from known species, predict the species to which the uncharacterized sequences belong and estimate the number of OTUs in the sample.*

In this paper, we give improved methods for applying hierarchical clustering to both of these applications. In general, hierarchical clustering algorithms are based on one of two types of operations: top-down splitting or bottom-up merging. In top-down approaches, all nodes start in one cluster, and in each step, a cluster from a previous step is split into two. In bottom-up approaches, each node starts in its own cluster, and in each step a pair of clusters are merged into a single, larger cluster. In the network clustering setting, for example, clusters may be split based on network modularity [36] or minimum cuts [10]. Clusters to merge may be chosen based on distances such as the Dice coefficient [5], the Jaccard index [20], or correlation of shortest-path profiles [40], among others. If either the top-down or bottom-up approach is carried out until no more splits or merges are possible, the resulting hierarchical clustering produces a tree ranging from the root (all nodes in one cluster) to the leaves (each node in its own cluster).

In order to apply most methods for predicting new annotations (either a complex for a protein or a species for a sequence), the hierarchical clustering must be converted into a flat grouping of the elements. Typically, this is done by choosing a set of nodes in the tree (called a *node-cut*) such that the path from each leaf to the root of the tree passes through exactly one chosen tree node. Each tree node chosen yields a cluster consisting of all the leaves in the subtree rooted at that node. We refer to such a flat, non-overlapping grouping of elements simply as a *clustering*. To avoid confusion, we refer to hierarchical clustering as "hierarchical decomposition." Some hierarchical decomposition algorithms provide a natural stopping point that can be used to choose a clustering. Newman's spectral partitioning [36], for example, is a top-down approach for hierarchically decomposing nodes in a network that stops splitting when any split would decrease the modularity of the clustering. Graph summarization [33], a bottom-up approach, stops merging when a particular cost function is minimized. However, many algorithms do not have natural stopping points [1, 10, 23]. Instead, they require the user to estimate the number of clusters beforehand, or they require a threshold and stop when no split or merge satisfying the threshold can be found. In general, it is not clear how to choose a number of clusters or an appropriate distance threshold. Therefore, choosing an appropriate clustering implied by the hierarchy is generally a stumbling block to their application.

In many applications, annotations are known for some of the elements being clustered, and these partial annotations can help determine which clustering compatible with the hierarchical decomposition is the most biologically reasonable. For example, Figure 1 shows a small PPI and its natural hierarchical decomposition. The PPI topology alone suggests a different clustering than the one that makes the most sense when the known annotations are taken into account.

Our contributions. In this paper, we propose a novel method, VI-Cut, to choose a clustering from a hierarchical tree decomposition based on how well the clusters induced by a cut in the tree match known annotations, as measured by the variation of information (VI, [32]) metric. The cut is chosen such that each node is placed in a cluster. Hence, nodes with unknown annotations can be together in a cluster with nodes

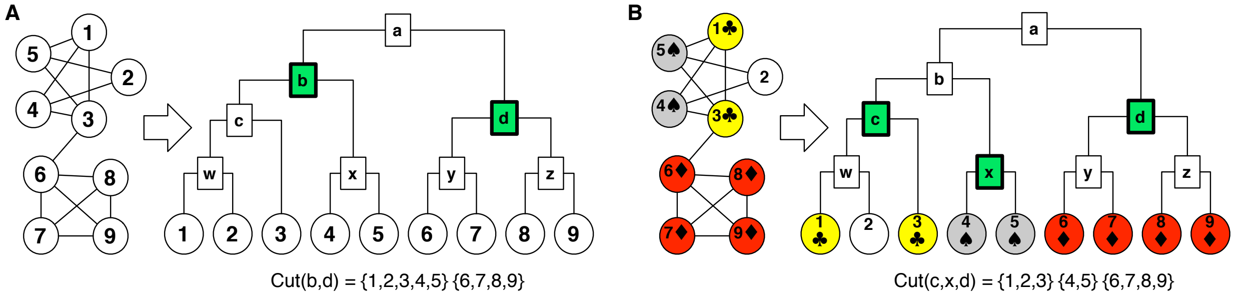
A
1
2
3
4
5
6
7
8
9
a
b
c
d
w
x
y
z
Cut(b,d) = {1,2,3,4,5} {6,7,8,9}
B
1♣
2
3♣
4♠
5♠
6♦
7♦
8♦
9♦
Cut(c,x,d) = {1,2,3} {4,5} {6,7,8,9}

Figure 1: Example PPI network where use of known annotations can produce a better clustering. (A) The network consists of two dense subgraphs that in most approaches would result in the hierarchical decomposition shown. By looking at the topology of the graph, it is reasonable to place proteins $\{1, 2, 3, 4, 5\}$ into one cluster and proteins $\{6, 7, 8, 9\}$ into a separate cluster by choosing cut $\{b, d\}$. (B) If some annotations are known (indicated in the figure by ♣, ♠, ♦), we want to choose a cut that not only abides by the topology, but also matches the known annotations as closely as possible. Here, cut $\{b, d\}$ is not ideal because it places proteins $\{1, 3\}$ and $\{4, 5\}$ together, which have different known annotations (node 2 has no known annotation). The better cut is $\{c, x, d\}$, which induces clusters $\{1, 2, 3\}$, $\{4, 5\}$, and $\{6, 7, 8, 9\}$. A method that only considers topology will be unable to reconstruct this clustering.

with known annotations. We can thus test the quality of a cluster based on how well we can use the clusters to predict annotations for nodes with unknown annotations (e.g. node 2 in Figure 1B). We apply this methodology to predicting membership in protein complexes and to predicting and estimating species composition for metagenomic DNA sequences. In each case, the VI-Cut method outperforms the most commonly used methods by making predictions with higher precision or recall. The VI-Cut is a very natural approach which gives a principled, mathematically sound way to convert a hierarchical decomposition to a flat clustering. To prove its generality, we show that it can be successfully applied in two very different biological problems, and we expect the approach will be applicable to other domains besides the two considered here.

Improvement in predicting protein complexes. We apply our VI-Cut method to two different hierarchical decompositions of a PPI network for the yeast *S. cerevisiae*. The first hierarchical decomposition is created using the Czekanowski-Dice distance between network nodes and applying a neighbor-joining algorithm, following the approach of Brun et al. [5]. The second hierarchical decomposition is created using graph summarization [33], which was recently shown [34] to outperform other graph clustering approaches such as MCL [49], MCODE [2], and Newman's spectral partitioning [36] at the task of predicting membership in protein complexes. For both types of hierarchical decompositions, we compare against the heuristics proposed by Brun et al. [5], Dotan-Cohen et al. [11], and a simple approach that chooses clusters from the tree that are sufficiently enriched according to the hypergeometric function. In the second case, we also compare against the clustering induced by the natural stopping point of the graph summarization algorithm. Unlike any other method, the VI-Cut produces clusters which perform well in terms of precision and recall of predicted annotations on both trees.

Improvement in predicting species. We also applied VI-Cut to predict species annotations for a simulated metagenomic sample created from 1677 real 16S rRNA gene sequences. The sample contains 49 species in various proportions. DOTUR [42] is the most common software for dividing input sequences into OTUs. DOTUR takes as input a distance matrix (derived from a multiple sequence alignment and distance correction) and a distance threshold to define when to stop merging clusters. We replicated six different methodologies for creating input to DOTUR that have been used in recent 16S rRNA studies [8, 15, 24, 43, 45, 51]. Each methodology uses a different multiple sequence alignment algorithm, distance correction, and distance threshold. None of these methods, however, take known OTU annotations into account. There are some semi-supervised clustering methods which take known annotations into account in the form of "must-link" and "cannot-link" constraints, but these algorithms typically use probabilistic approaches [28, 35], are not hierarchical, and have not been applied in these domains. We compare the OTU predictions made by VI-Cut with the clusters produced by each of these six methodologies. In each case, the clusters created by VI-Cut produce predictions with about the same precision as the previous methodologies, but with large increases in recall. Further, the VI-Cut clusters provide a much better estimate of the true number of species embedded within the data set.

1.1 Related work: semi-supervised clustering

Several previous attempts have been made to apply semi-supervised hierarchical clustering to gene expression data. To produce a flat clustering from a hierarchical tree decomposition derived from expression data, several methods assign an *enrichment* score to each internal tree node based on the partial, known annotations, signifying the functional coherence of the cluster [6, 46, 48]. Clusters are then chosen by iteratively choosing high-scoring subtrees, subtrees with uniquely enriched annotations, or other similar heuristics. Raychaudhuri et al. [39] mine medical literature and assign a cluster coherence score based on documents which relate genes. Of these methods, Tan et al. [46] was the only one that was applied to predicting protein function from gene expression data. They report an accuracy of only 50-60%.

Recently, Dotan-Cohen et al. [11] proposed a semi-supervised approach based on choosing a subset of edges in the tree decomposition. Each chosen edge induces a connected component in the tree which corresponds to a cluster. Their goal is to choose the minimum number of edges such that each cluster consists of genes which all share at least one annotation, allowing genes that are unannotated to take on any annotation. After clustering, they predict GO processes for unannotated genes by choosing the shared annotation in the gene's cluster. They applied this approach to tree decompositions derived from gene expression data.

All of the above approaches differ from VI-Cut in the objective function used to produce a clustering from the tree, and are only applied to clusterings derived from gene expression or literature similarity. No previous studies have predicted OTU annotations using a semi-supervised approach. Several studies have looked at network modules. Brun et al. [5] use PPI data to build a tree decomposition and extract clusters which have a majority annotation. Other heuristics have been proposed to choose a clustering from a network decomposition [1, 40, 41], however, they either rely on manual inspection of the hierarchical decomposition [40], or require a similarity threshold to be input by the user [1, 41].

Finally, there are semi-supervised clustering methods which take known annotations into account in the form of "must-link" and "cannot-link" constraints, but these algorithms typically use probabilistic approaches [28, 35], are not hierarchical, and have not been applied in these domains, and can create clusters that artificially merge items that are distant but have similar labels.

2. Methods

2.1 Finding the clustering that best matches known annotations (VI-Cut)

Criteria for choosing a clustering. A hierarchical decomposition is specified by a tree T where the leaves correspond to the elements being clustered. A *node-cut* is a subset K of tree nodes such that the path from every leaf of T to Root(T) passes through some node in K and such that there is no pair of nodes $x, y \in K$ where x is an ancestor of y. Every node-cut K of the tree induces a clustering C_K: each node $x \in K$ yields one cluster that contains the elements corresponding to the leaves in the subtree rooted at x. Despite the simple structure, there are an enormous number of possible node-cuts even for short, binary trees. A complete binary tree of height 7, e.g., induces $44, 127, 887, 745, 906, 175, 987, 802$ possible clusterings.

We assume that some (but not all) of the elements that we are interested in clustering are already annotated. Let D be the partial clustering defined by these known annotations by grouping those with the same annotation together. Among all the possible choices for a node-cut K, we desire the one that induces a clustering C_K that best matches the known partial information D. A natural measure for how well C_K agrees with D is given by the variation of information (VI, [32]) distance metric between the two clusterings:

$$VI(C_K, D) \doteq H(C_K) + H(D) - 2I(C_K, D)\,. \tag{1}$$

In the definition of VI, the clusterings C_K and D are represented as random variables that take on values from $\{1, \ldots, |C_K|\}$ and $\{1, \ldots, |D|\}$, respectively, where we ignore unannotated proteins. $H(X)$ denotes the entropy of random variable X, and $I(X, Y)$ denotes the mutual information between the random variables X and Y. The mutual information gives the reduction in uncertainty about random variable D if the value of random variable C_K is given. The VI distance, which is a metric, is rapidly becoming a standard measure with which to compare clusterings. In the following, we exploit the fact that it can be rewritten such that the total distance is the sum of each cluster's contribution. Other properties of VI are explored by Meila [32].

Because $I(X, Y) = H(X) + H(Y) - H(X, Y)$, where $H(X, Y)$ is the joint entropy, we can rewrite $VI(C_K, D)$ to be $2H(C_K, D) - H(C_K) - H(D)$. Over possible choices of K, $H(D)$ remains constant. Therefore, $\min_K VI(C_K, D)$ is achieved for the same K that minimizes

$$\min_K 2H(C_K, D) - H(C_K)\,. \tag{2}$$

To find a node-cut that minimizes this value, we assign a *quality score* $q(x)$ to each node x in the hierarchical decomposition T. The function $q(x)$ will be chosen so that the sum of the quality scores for nodes in a node-cut K will equal $2H(C_K, D) - H(C_K)$. Define $L(x)$ to be the set of leaves in the subtree rooted at node x that are annotated with some known annotation, and $A(d)$ to be the set of leaves (from the whole tree) that are known to have annotation d. Define $n = |L(\text{Root}(T))|$, the number of elements that have a known annotation. We then set $q(x)$ to be

$$q(x) \doteq p(x) \log p(x) - 2 \sum_{d \in D} p(x, d) \log p(x, d)\,, \tag{3}$$

where the probabilities are defined as

$$p(x) = |L(x)|/n\,, \tag{4}$$

$$p(x, d) = |L(x) \cap A(d)|/n\,. \tag{5}$$

The value $p(x)$ is the probability that an element with a known annotation would fall into the cluster induced by x. The joint probability $p(x, d)$ is the probability that a random annotated element falls into cluster x and has annotation d. Note that $H(C_K) = -\sum_{x \in C_K} p(x) \log p(x)$ and $H(C_K, D) = -\sum_{x \in C_K} \sum_{d \in D} p(x, d) \log p(x, d)$ so that (3) implies that $\sum_{x \in C_K} q(x) = 2H(C_K, D) - H(C_K)$, which is the value we are attempting to minimize in (2). Therefore, the node-cut whose quality scores sum to the smallest number corresponds to the clustering that best matches the known annotations according to the VI distance.

Algorithm to find the best cut in a hierarchical decomposition. We can find a node-cut K in a tree so that $\sum_{x \in K} q(x)$ is minimized (a "min-node-cut") using standard dynamic programming. Let $\mathbf{Children}(x)$ denote the children of a tree node x. We can compute the minimum-weight node-cut recursively:

$$\text{CutDist}(x) = \min \begin{cases} q(x) & \text{case I} \\ \sum_{y \in \mathbf{Children}(x)} \text{CutDist}(y) & \text{case II} \end{cases} \tag{6}$$

The min-node-cut of a subtree S either chooses the root x of S with a weight of $q(x)$ (case I) or it does not choose the root and chooses instead the min-node-cut in each of the subtrees rooted at the children of x (case II). If x is a leaf node, the min-node-cut defaults to $q(x)$. Therefore, the value of $\text{CutDist}(\text{Root}(T))$ is weight of the smallest weight node-cut. To find the actual choice of nodes corresponding to the node-cut of this weight we can backtrack through which cases occurred during the recursive calls. We have flexibility in how we break ties when the value of case I equals the value of case II. If we always break ties in favor of case I, we will choose the highest min-node-cut in the tree. Alternatively, if we always choose case II, we choose the lowest min-node-cut in the tree.

2.2 Handling multiple annotations on some elements

Up to this point, we have assumed that each element has at most one known annotation. This is true by definition in the OTU clustering problem and, of all yeast proteins annotated with some MIPS complex, only 11% are annotated with more than one complex. Hence, for the applications we consider in this paper, the assumption of a single annotation on each element is mostly justified. On the other hand, multiple annotations are present in other applications. They can be used to model either uncertainty in the truth or genuine membership in multiple clusters. A natural way to handle multiple annotations on each element is to look for the tree cut K that induces a clustering C_K that minimizes the VI distance between C_K and the closest clustering compatible with a choice of a single annotation for each element. Unfortunately, even computing the minimum distance between a given clustering C and a clustering compatible with a set of annotations is NP-complete. Correspondingly, computing the optimal tree cut under this scoring function is also NP-complete. This is formalized and shown in the rest of this subsection.

Definition 1 (annotation collection). *Given a set of elements E and a set of annotations L, an* annotation collection *is a collection of subsets $A_\ell \subseteq E$ for each $\ell \in L$ such that every $e \in E$ is in at least one A_ℓ.*

An annotation collection defines which annotations apply to each of the elements of E. Each A_ℓ consists of the elements that are annotated with ℓ. An annotation collection implicitly specifies many possible clusterings for E: a choice of a single annotation $\ell(e)$ for every $e \in E$ such that $e \in A_{\ell(e)}$ induces a clustering that groups all elements with the same annotation together. Let Compatible$(\mathcal{L})$ be the set of clusterings induced in this way by an annotation collection $\mathcal{L}$. The natural measure of how well a given clustering C matches an annotation collection $\mathcal{L}$ is to compute the minimum VI distance between C and some clustering in Compatible$(\mathcal{L})$. Formally, we define:

Problem 3 (MIN-VI LABEL CHOICE). *Given a set of elements E, a clustering C of E, an annotation collection $\{A_\ell \subseteq E : \ell \in L\}$ over a set of annotations L, compute $\min_{D \in \textit{Compatible}(\mathcal{L})} VI(C, D)$.*

The problem of computing this minimum distance is NP-complete and so is the related problem of finding a cut in a tree to minimize the VI distance. See appendix for proofs of the following theorems.

Theorem 1. *The decision version of* MIN-VI LABEL CHOICE *is NP-complete.*

Problem 4 (MIN-VI TREE CUT WITH LABEL CHOICE). *Given a set of elements E, a hierarchical clustering T of E, and an annotation collection $\mathcal{L} = \{A_\ell \subseteq E : \ell \in L\}$ over a set of labels L, compute $\min_{K \in \textit{Cut}(T), D \in \textit{Compatible}(\mathcal{L})} VI(K, D)$.*

Theorem 2. *The decision version of* MIN-VI TREE CUT WITH LABEL CHOICE *is NP-complete.*

Given these hardness results, we are forced to consider heuristics to handle the few proteins that belong to multiple MIPS complexes. We cannot use equation (5) directly to compute $p(x, d)$ because it will not yield a probability distribution. Instead, if protein i has k_i annotations, we count each of its annotations as $1/k_i$. In other words, $p(x, d) = (1/n) \sum_{i \in L(x) \cap A(d)} 1/k_i$. This way $p(x, d)$ defines a probability distribution even if proteins belong to multiple complexes, and we can use the method of the previous section as a heuristic to find a clustering that matches the given annotations well. This is the approach we follow for the complex membership prediction experiments below.

2.3 Predicting new annotations

Ultimately, our goal is to use the clusters found to make new predictions for protein or sequence membership within complexes or OTUs. A common approach, here called “majority,” transfers an annotation A to every unannotated element in a cluster if more than 50% of the annotated elements in the cluster are annotated with A. If no annotation exists on more than 50% of the annotated elements, no predictions are made. Clusters consisting of a single annotated element are ignored.

To test the efficacy of the various clustering methods and annotation transfer rules, we omit the known annotations from a fraction of the elements. The omitted annotations are the "test set," and the remaining annotations are the "training set." The best VI-Cut is found based only on the known annotations in the training set. We vary the size of the training set from 10% to 90% of the total number of elements with known annotations, chosen randomly. For each element x in the test set, the majority annotation is computed and then transferred to x as predicted annotations. If multiple annotations were transferred, each transferred annotation counted as one prediction. A prediction is correct if the protein or sequence is known to belong to that complex or OTU, and incorrect if it is only known to belong to other complexes or OTUs. Naturally, given the incomplete state of knowledge, some "incorrect" predictions may in fact be correct. For each size of the training set, we measure performance by the precision and recall of the predictions made over 500 random samplings (for the Snip approach we only took 10 samplings). Precision is the probability that a predicted annotation is correct. Recall is the average number of elements in the test set for which a correction annotation was made divided by the total number of elements in the test set.

2.4 Application to predicting protein complex annotations

Protein networks. We constructed a protein interaction network for *S. cerevisiae* using all edges in the IntAct [25] database. This network contains 5,492 proteins with 40,332 interactions. For the hierarchical decomposition, we consider only the main component of the network (which we refer to as Y_{ppi}), which contains 5,462 proteins and 40,311 interactions. Most of these interactions were determined using yeast two-hybrid or TAP assays, while a smaller number were derived from traditional, low-throughput experiments. Interactions obtained from high-throughput assays, however, are typically very noisy with potentially a 90% false positive rate [19]. Hence, we created a high-confidence yeast interaction network that only includes edges from IntAct which are supported by at least two PubMed identifiers. (Identifiers listed twice by IntAct for the same interaction were included in the network.) Interactions confirmed to occur using more than one experiment are more likely to be true than interactions found by one experiment. The full high-confidence network contains 2,604 proteins and 8,341 interactions, fewer than half the proteins of the Y_{ppi} network. The main component, which we call $Y_{\text{high-conf}}$, contains 2,378 proteins and 8,189 interactions. We performed our experiments on both of these networks to probe the effects of noise.

Protein complexes. Annotations for yeast complexes were taken from MIPS [18], ignoring the "550" section of the catalog, which represent computationally inferred complexes. This set of complexes has been widely used to assess computational methods [21,38,52]. To make the most specific predictions possible we use the lowest-level complexes in the catalog. Of the 5,462 and 2,378 proteins in Y_{ppi} and $Y_{\text{high-conf}}$ 1191 and 930 proteins, respectively, have some known complex annotation. Of the 267 complexes, 266 and 230 are represented by at least one protein in the Y_{ppi} and $Y_{\text{high-conf}}$ network, respectively.

Hierarchical decomposition of the PPI network. We use two approaches to generate two different hierarchical network decomposition trees. The first tree, called T_{Dice}, is built by applying the neighbor-joining algorithm BIONJ [17] to distances between proteins computed by the Czekanowski-Dice [5] distance. Self-loops were added to each protein to decrease the distance between proteins that interact. This is the approach followed by Brun et al. [5] for predicting the cellular function of proteins. The second tree, called T_{GS}, is built using the greedy graph summarization algorithm (GS, [33,34]). The GS process has a natural stopping point (when there is no longer any compression benefit to merging two nodes). We modified the algorithm so that it continues to merge the lowest cost pair of nodes until all nodes are placed in a single cluster.

Comparison methods. For the T_{Dice} tree, we compare the VI-Cut approach against three other methods. Brun et al. filter false edges from their PPI by removing proteins which take part in fewer than 3 interactions. In our setting, we simply use the high-confidence network, $Y_{\text{high-conf}}$. Brun et al. [5] extract clusters from their hierarchical network decomposition by selecting the largest subtrees that contain at least 3 proteins that all share the same annotation and that make up the majority annotation in the subtree. Dotan-Cohen et al. [11] choose the minimum number of edges in the tree to "snip" such that each cluster induced by the snip contains proteins that all share at least one annotation. Another popular approach we try involves using the hypergeometric P-value to assignment an enrichment score to each internal node in the tree. We then do a breadth-first walk down the tree from the root, choosing clusters if they are enriched past a pre-defined threshold ($P \leq 0.01$). The computed P-values are Bonferroni corrected to account for multiple-testing. We refer to these methods by Brun, Snip, and Enrich, respectively. When considering T_{GS}, we also compare with the clustering induced by the natural stopping point of the unmodified greedy GS process. For the VI-Cut on both trees we select the lowest min-node-cut.

2.5 Application to annotating operational taxonomic units (OTUs)

Creation of simulated 16S sample. We obtained 1860 partial 16S rRNA gene sequences from the Ribosomal Database Project II (release 9.57 [7]) with complete taxonomic identification. These sequences were then screened for conflicting annotation information using the RDP Bayesian classifier [50], and selected for length and quality, resulting in a final set of 1677 sequences. This dataset is designed to simulate a microbial environment of moderate complexity spanning seven phyla with several dominant and rare species. Nine species are only observed once in the data, while eight species have more than 90 observations. Though no single species represents more than 6% of the sample, 66% of the sample is Proteobacteria with roughly equally distributions of Alpha-, Beta-, and Gammaproteobacteria. By using real 16S rRNA sequences, we accurately model the nucleotide divergence we expect to see within any species. This approach has been successfully used to provide high-quality benchmarks for metagenomic assembly and gene-finding [31].

Hierarchical decomposition of OTU sequences. Sequences were oriented and subsequently aligned using a multiple-sequence alignment (MSA) algorithm (such as ClustalW [47], NAST [9], or MUSCLE [13]). MSAs were trimmed so that each sequence spanned the entire alignment. For the NAST MSA, columns containing only gaps were removed. From the alignment, we then used DNADIST with default parameters from the PHYLIP package [14] to compute distance matrices using the Felsenstein-84 [14] or Jukes-Cantor [22] distance correction techniques. The distance matrices were then fed into DOTUR [42], an OTU clustering algorithm, which assigns sequences to OTUs using the furthest-neighbor algorithm. The clusters returned by DOTUR depend on a user-defined distance threshold. If the threshold is set to 0.03, for example, an OTU cluster is defined as a set of sequences which are each no more than 3% different from each other. We modified the DOTUR program to output the full hierarchical tree decomposition, which we use to find the VI-Cut clusters based on partial, known annotations.

Comparison methods. We consider six recently published methods for identifying OTUs that illustrate the current range of OTU-analysis used in the field of metagenomics. These methods differ in the MSA, distance correction, and distance threshold used to define OTUs. The six methods we consider are: Kennedy et al. [24], Fulthorpe et al. [15], Schloss et al. [43], Corby-Harris et al. [8], Sogin et al. [45], and Warnecke et al. [51]. We refer to each by their first author. See Table 1 for their parameters. Other approaches, such as by Gordon [29], for example, manually edit alignments and use specific lane masks (which reduces an MSA to only highly conserved columns), whereas our method is fully automated. We compare the VI-Cut clusters, obtained using the highest min-node-cut, with the threshold-derived clusters of these six methodologies based on their predictive ability and estimation of the number of OTUs present in the sample.

3. Results and Discussion

3.1 VI-Cut yields better predictions for protein complexes

We created a hierarchical decomposition T_{Dice} based on the Czekanowski-Dice distance between proteins in $Y_{\text{high-conf}}$, following the same procedure described by Brun et al. [5] (see Section 2.4). From T_{Dice}, for various sizes of training sets, we compute four clusterings derived from the methods of Brun et al, Dotan-Cohen et. al., the Enrich approach described in Section 2.4, and the VI-Cut approach described in Section 2.1. Using these clusterings, we predict membership in MIPS protein complexes using the "majority" annotation transfer rule. The precision and recall of these predictions are shown in Figure 2A. The

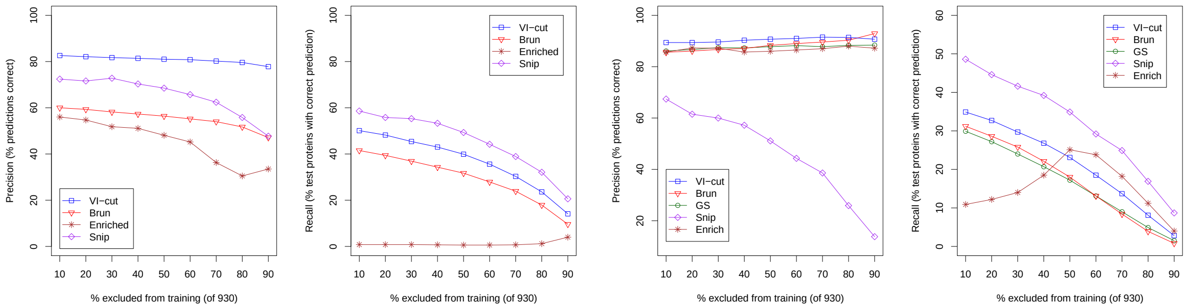

Figure 2: Precision and recall for various sizes of training sets on (A) the T_{Dice} tree, and (B) the T_{GS} tree.

x-axis of these plots gives the percentage of annotations that were excluded from the annotation set when choosing a clustering; larger values indicate tests where there are fewer known annotations. The y-axis shows the precision and recall of the predictions — in both cases, larger numbers are preferred.

While both Brun et al. and VI-Cut take into account the known annotations when defining their clusters from the tree, the cut chosen by minimizing the VI distance is able to make considerably more accurate predictions than the clusters created by the Brun et al. heuristic. Over all tested sizes of training set, the predictions made by the VI-Cut approach are more accurate by at least 22 percentage points. Further, when the number of known annotations is very small, the improvement of the VI-Cut method is even greater. At the fewest number of known annotations (90% annotations excluded) the VI-Cut method is almost 30% more precise in its predictions. The VI-Cut method also makes more correct annotations (larger recall) over the entire range of sizes for the training set. The Enrich approach is even less precise than Brun et al. and with a significantly lower recall. This is largely because the enrichment approach returns a few number of large modules for which very few predictions can be made. The Snip approach yields a higher recall than VI-Cut but with a greater loss in precision.

The robustness of the VI-Cut approach is not limited to hierarchical decompositions that are derived from the Czekanowski-Dice distance. We repeated the prediction experiments using the tree T_{GS} built by the greedy graph summarization (GS) technique. Figure 2B shows the precision and recall achieved by the four previously mentioned methods, and the clustering induced by the natural stopping point of the GS procedure. The clusters produced by the natural stopping point are the same regardless of the training set because annotations are not considered when the GS algorithm is applied. Precision and recall can still vary, however, as predictions in majority annotations change within each cluster. As shown in Figure 2B, the predictions made by the VI-Cut are almost always more precise than every other method. The Snip method has a larger recall, but this is negated by its extremely poor precision. Interestingly, the Enrich approach initially has an increase in recall with less training data. With both large and small amounts of training data, few modules are chosen (excluded singleton modules) indicating that it is rather easy to significantly deviate from random expectation according to the hypergeometric function. Hence, very few predictions are made. However, in the middle regions, more reasonably sized modules are chosen, allowing for greater recall.

In general, the predictions made on T_{GS} are much more precise than those made on T_{Dice}. This suggests that the hierarchical decomposition defined by GS better represents the protein complexes within the PPI. Interestingly, for T_{GS}, the precision of all approaches except for Snip slightly increases as less training data is available. This may imply that with smaller training sets, only easy predictions are made. As the size of the training set increases, however, more difficult predictions are attempted, for which accuracy is generally lower. Further, for T_{GS}, Enrich especially benefits by choosing a larger number of clusters which are smaller and more enriched. Overall, VI-Cut makes precise predictions covering many proteins on both trees, unlike

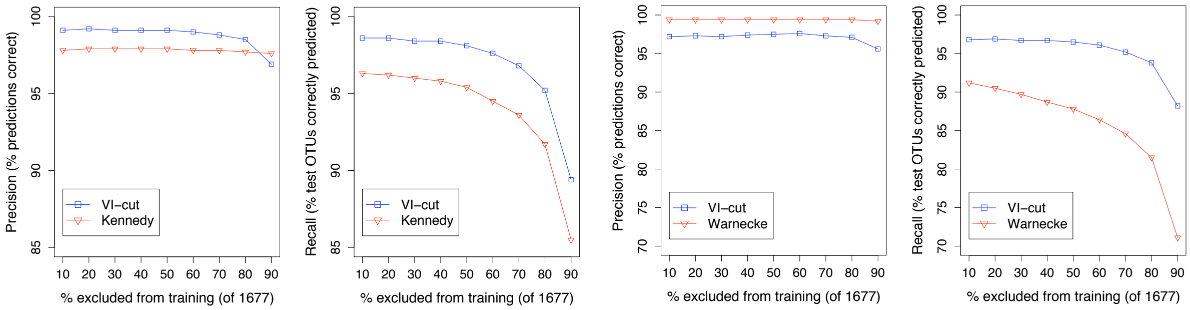

Figure 3: Precision and recall comparison of the Kennedy and Warnecke OTU clustering methods. Although Kennedy and Warnecke produce the same clusters regardless of the training set, the predictions they make vary due to differences in the majority annotation within each cluster.

any other method.

Variations. We performed the experiments above on the unfiltered Y_{ppi} network, as well. The results are not shown here due to lack of space, but they echo the results obtained on the $Y_{\text{high-conf}}$ network. Further, we tested two other annotation transfer rules in addition to the majority rule: plurality and the hypergeometric P-value [44]. Plurality transfers the most common annotation within a cluster and hypergeometric P-value transfers annotations based on their statistical enrichment in a cluster. Although the VI-Cut continued to outperform the other methods, the performance of these two rules was not generally preferable to the majority rule. The hypergeometric P-value was always less precise than the majority rule, and, of the correct predictions made, 94% and 98.1% were also made by the majority rule for the Brun and VI-Cut respectively, averaged over all sizes of the training set and over both trees. For the Brun and VI-Cut methods, the predictions made by plurality were 97.3% and 99.5% of the time the same, respectively, as the predictions made by the majority rule averaged the same way. For the natural GS clusters, the plurality rule resulted in a 15-16% higher recall but with a loss of 14-16% in precision with respect to VI-Cut.

We also attempted to create clusters using a simpler approach that chooses a cut in the tree by a breadth-first search, adding nodes to a queue one at a time (creating a cluster for each leaf encountered) until the length of the queue equals a certain number of clusters (which we varied). This method, however, induced clusters which proved to make few correct predictions and consistently achieved less than 50% precision.

3.2 VI-Cut yields better prediction of OTUs

We apply the same tests to predict OTU annotations for 16S DNA sequences. Predictions were made in the same way as with protein complexes, but instead of complex-membership annotations, we use known OTU annotations and transfer them to sequences with no OTU annotation. We again use the majority annotation transfer rule. We compare the predictive ability of the VI-Cut method for clustering metagenomic samples with previously published methods, including Kennedy [24], Fulthorpe [15], Schloss [43], Corby-Harris [8], Sogin [45], and Warnecke [51], as described in 2.3. We also compare each method's ability to estimate the correct number of OTUs present in the sample. The Corby-Harris approach resulted in nearly identical predictions and estimations as the Kennedy method. We therefore omit discussion of those results.

The VI-Cut generally outperforms each of these methods. Of the methods we compared against, Kennedy and Warnecke had the best overall recall and precision, respectively. Figure 3 compares these methods with the VI-Cut. Compared to Kennedy, the VI-Cut mostly makes more precise predictions, and covers a larger number of OTUs. Although Warnecke makes slightly more precise predictions (average gain of 2%), the VI-Cut method has significantly greater recall. For example, with 80% of the known OTU sequences excluded from the training set, the VI-Cut makes correct predictions for 1256 OTUs, compared to just 1093 by Warnecke.

	80% Annotations				90% Annotations			
Method	**# OTUs**	**Prec.**	**Recall**	**Avg. VI**	**# OTUs**	**Prec.**	**Recall**	**Avg. VI**
Tree 1: ClustalW, Felsenstein								
Kennedy (0.03)	70	97.7	91.7	0.087	70	97.6	85.5	0.087
VI-Cut	45	98.5	95.2	0.031	42	96.9	89.4	0.050
Tree 2: NAST, Felsenstein								
Fulthorpe (0.00)	386	99.2	63.0	0.646	386	98.9	49.6	0.646
VI-Cut	47	97.3	93.7	0.053	45	95.6	87.9	0.073
Tree 3: NAST, Jukes-Cantor								
Schloss (0.03)	99	97.7	88.1	0.157	99	97.5	80.5	0.157
VI-Cut	45	97.1	93.8	0.054	42	95.6	88.2	0.073
Tree 4: NAST, Jukes-Cantor								
Warnecke(0.01)	185	99.4	81.5	0.320	185	99.2	71.1	0.320
VI-Cut	45	97.1	93.8	0.054	42	95.6	88.2	0.073
Tree 5: MUSCLE, Jukes-Cantor								
Sogin (0.03)	96	97.7	87.5	0.190	96	97.5	78.2	0.190
VI-Cut	45	97.5	93.8	0.046	43	96.1	88.2	0.046

Table 1: Comparison of VI-Cut with previous OTU clustering approaches applied to trees constructed from DOTUR with various parameters. Distance threshold used for DOTUR is shown in parentheses. Performance is presented for both 80% and 90% annotations excluded, average over 100 trials. **# OTUs** shows the average number of OTUs predicted by each method. The correct number of OTUs is 49. **Prec.** and **Recall** show the precision and recall for each approach. **Avg. VI** shows the VI distance of the clustering to the actual OTUs.

For all six trees, we find that our VI-Cut based approach yields not only a closer VI distance to the true clustering, but also a much closer approximation to the true number of OTUs. There are 49 true OTUs in the data set and the VI-Cut estimates between 42 and 47, depending on which tree is used. This is a far better estimate of the true diversity of the population than the estimates of the other methods, which range between 70 and 386. The number of OTUs predicted are shown for test set size equal to 80% or 90% in Table 1. While it is true that our method starts with known annotations that hint at the number of true OTUs present in the sample beforehand, the average number of unique OTUs in the training set was only 36. Yet, VI-Cut was still able to identify that other OTUs exist, based on their topological non-compatibility with known annotations in the tree.

4. Conclusion

We presented a framework for finding cut-induced clusters in hierarchical tree decompositions that best match a partial set of known annotations. We take advantage of the property that the variation of information metric can be decomposed into the sum of the contributions of each potential cluster. This allows us to find the optimal clustering that minimizes the distance to the partial known truth. The method is therefore well-founded mathematically. While we showed that a natural generalization that allows more than one annotation per element is NP-hard, our VI-Cut method still performs well in practice. The framework makes improved predictions of proteins' membership in complexes as well as species annotations for metagenomic samples. The success of VI-Cut in two very different domains is evidence of the technique's generality.

Appendix: Proofs of Theorem 1 and 2

Theorem 1. *The decision version of* MIN-VI LABEL CHOICE *is NP-complete.*

Proof. We reduce from EXACT COVER BY 3-SETS (X3C) [16]. Let I be an instance of X3C specified by a set X_I and a collection of 3-tuples $R_I = \{(x, y, z) : x, y, z \in X_I\}$. An I is a "yes" instance if there is a subcollection M of R_I such that every element in X_I belongs to exactly one set in M. We construct an instance of MIN-VI LABEL CHOICE as follows. Take $E = X_I$, and let $C = \{E\}$ be the clustering consisting of a single cluster. For every $(x, y, z) \in R_I$, we create a annotation $A_\ell = \{x, y, z\}$ containing only those 3 elements. The annotation collection $\mathcal{L}_I$ consists of these A_ℓ sets.

We show that there is a clustering $D \in$ Compatible($\mathcal{L}_I$) with $VI(C, D) \leq \log(|E|/3)$ if and only if I belongs to X3C. Because $C = \{E\}$, we have $H(C) = 0$, and $VI(C, D) = 2H(C, D) - H(C) - H(D) = H(D)$. If there is an exact cover D, it consists of a set of $|E|/3$ clusters of size 3, yielding $H(D) = -(|E|/3)\,[(3/|E|)\log(3/|E|)] = \log(|E|/3)$. If there is no exact cover, then any clustering D induced by $\mathcal{L}$ must contain some clusters of size ≤ 2. Because $-(3/n)\log(3/n) < -(2/n)\log(2/n) - (1/n)\log(1/n) < -(3/n)\log(1/n)$ for all n, the presence of clusters of size 2 or 1 yields a larger entropy than grouping those elements into sets of size 3. Hence, if there is no exact cover, $H(D) > \log(|E|/3)$ for all D induced by $\mathcal{L}$. In fact, it can be shown that the difference between the minimum VI distance for an instance with an exact cover and an instance without an exact cover is at least $1/|X_I|$, so this difference can be encoded using a polynomial number of bits. □

Theorem 2. *The decision version of* MIN-VI TREE CUT WITH LABEL CHOICE *is NP-complete.*

Proof. As above, we reduce from EXACT COVER BY 3-SETS (X3C) [16] (using the same notation). We construct an instance of MIN-VI TREE CUT WITH LABEL CHOICE as follows. Take $E = X_I \cup Y$ where Y is a set of new elements such that $|Y| = 2|X_I|$ and let the hierarchical decomposition T have a star topology (all leaves connected to the root) with the elements of E as leaves. For every $(x, y, z) \in R_I$, we create a label $A_\ell = \{x, y, z\}$ containing only those 3 elements. The annotation collection $\mathcal{L}_I$ consists of these A_ℓ sets and the set Y.

We show that there is a clustering $D \in$ Compatible($\mathcal{L}_I$) and node cut K for T, with $VI(K, D) \leq 1/3\log(|E|/3) + 2/3\log 3/2$ if and only if I belongs to X3C. It is easy to verify that if there is an exact cover D' then with $D = D' \cup \{Y\}$ and $K = \{E\}$ we get $VI(K, D) = 1/3\log(|E|/3) + 2/3\log 3/2$. Conversely, if there is no exact cover, then any clustering D induced by $\mathcal{L}$ must contain some clusters of size ≤ 2. Using a similar argument as before, we can show that $VI(D \cup \{Y\}, \{E\}) > 1/3\log(|E|/3) + 2/3\log 3/2$. The only other node cut possible is the one which puts every node in E in a seperate cluster and the corresponding optimal label choice gives a VI distance $\geq 2/3\log|E| - 2/3\log 3/2 > 1/3\log(|E|/3) + 2/3\log 3/2$ (in the ideal case every element in R_I will have its own label), for $|E| > 2$. Note that the difference between the minimum VI distance for an instance with an exact cover and an instance without an exact cover is still $\geq 1/|X_I|$ and hence can be encoded using a polynomial number of bits. □

References

[1] V. Arnau, S. Mars, and I. Marín. Iterative cluster analysis of protein interaction data. *Bioinformatics*, 21(3):364–378, 2005.

[2] G. D. Bader and C. W. V. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4:2, 2003.

[3] A. Bernard, D. S. Vaughn, and A. J. Hartemink. Reconstructing the topology of protein complexes. In T. P. Speed and H. Huang, editors, *RECOMB*, volume 4453 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 2007.

[4] S. Brohee and J. van Helden. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics*, 7:488+, November 2006.

[5] C. Brun, F. Chevenet, D. Martin, J. Wojcik, A. Guenoche, and B. Jacq. Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome Biology*, 5(1), 2003.

[6] E. C. Buehler, J. R. Sachs, K. Shao, A. Bagchi, and L. H. Ungar. The crasss plug-in for integrating annotation data with hierarchical clustering results. *Bioinformatics*, 20(17):3266–3269, 2004.

[7] J. R. Cole, B. Chai, R. J. Farris, Q. Wang, S. A. Kulam, D. M. McGarrell, G. M. Garrity, and J. M. Tiedje. The ribosomal database project (rdp-ii): sequences and tools for high-throughput rrna analysis. *Nucleic Acids Research*, 33(Database-Issue):294–296, 2005.

[8] V. Corby-Harris et al. Geographical distribution and diversity of bacteria associated with natural populations of drosophila melanogaster. *Applied and Environmental Microbiology*, 73:3470–3479, 2007.

[9] T. Z. DeSantis, P. Hugenholtz, K. Keller, E. L. Brodie, N. Larsen, Y. M. Piceno, R. Phan, and G. L. Andersen. Nast: a multiple sequence alignment server for comparative analysis of 16s rrna genes. *Nucleic Acids Res*, 34(Web Server issue), July 2006.

[10] I. S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(11):1944–1957, 2007.

[11] D. Dotan-Cohen, A. A. Melkman, and S. Kasif. Hierarchical tree snipping: clustering guided by prior knowledge. *Bioinformatics*, 23(24):3335–3342, 2007.

[12] P. B. Eckburg, E. M. Bik, C. N. Bernstein, E. Purdom, L. Dethlefsen, M. Sargent, S. R. Gill, K. E. Nelson, and D. A. Relman. Diversity of the human intestinal microbial flora. *Science*, 308(5728):1635–1638, June 2005.

[13] R. C. Edgar. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*, 32(5):1792–1797, 2004.

[14] J. Felsenstein. PHYLIP — phylogeny inference package (version 3.2). *Cladistics*, 5:164–166, 1989.

[15] R. R. Fulthorpe, L. F. W. Roesch, A. Riva, and E. W. Triplett. Distantly sampled soils carry few species in common. *ISME*, 2:901–910, 2008.

[16] M. R. Garey and D. S. Johnson. *Comptuers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.

[17] O. Gascuel. BIONJ: an improved version of the nj algorithm based on a simple model of sequence data. *Mol Biol Evol*, 14(7):685–695, July 1997.

[18] U. Guldener, M. Munsterkotter, G. Kastenmuller, N. Strack, J. van Helden, C. Lemer, J. Richelles, S. J. Wodak, J. Garcia-Martinez, J. E. Perez-Ortin, H. Michael, A. Kaps, E. Talla, B. Dujon, B. Andre, J. L. Souciet, J. De Mon tigny, E. Bon, C. Gaillardin, and H. W. Mewes. CYGD: the comprehensive yeast genome database. *Nucleic Acids Research*, 33(Supplement 1):D364+, January 2005.

[19] T. G. Hart, A. K. Ramani, and E. M. Marcotte. How complete are current yeast and human protein-interaction networks? *Genome Biology*, 7:120+, December 2006.

[20] P. Jaccard. Nouvelles recherches sur la distribution florale. *Bulletin de la Socit Vaudoise des Sciences Naturelles*, pages 223–270, 1908.

[21] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein. A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302(5644):449–453, October 2003.

[22] T. H. Jukes and C. R. Cantor. *Evolution of Protein Molecules*. Academy Press, 1969.

[23] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 1998.

[24] J. Kennedy et al. Diversity of microbes associated with the marine sponge, haliclona simulans, isolated from irish waters and identification of polyketide synthase genes from the sponge metagenome. *Environmental Microbiology*, 10:1888–1902, 2008.

[25] S. Kerrien, Y. Alam-Faruque, B. Aranda, I. Bancarz, A. Bridge, C. Derow, E. Dimmer, M. Feuermann, A. Friedrichsen, R. Huntley, C. Kohler, J. Khadake, C. Leroy, A. Liban, C. Lieftink, L. Montecchi-Palazzi, S. Orchard, J. Risse, K. Robbe, B. Roechert, D. Thorneycroft, Y. Zhang, R. Apweiler, and H. Hermjakob. IntAct—open source resource for molecular interaction data. *Nucleic Acids Research*, 35(Database issue), January 2007.

[26] M. Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J Mol Evol*, 16:111–120, Dec 1980.

[27] A. D. King, N. Przulj, and I. Jurisica. Protein complex prediction via cost-based clustering. *Bioinformatics*, 20(17):3013–3020, November 2004.

[28] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. Semi-supervised graph clustering: a kernel approach. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 457–464, New York, NY, USA, 2005. ACM.

[29] R. E. Ley, F. Bäckhed, P. Turnbaugh, C. A. Lozupone, R. D. Knight, and J. I. Gordon. Obesity alters gut microbial ecology. *Proc Natl Acad Sci U S A*, 102(31):11070–11075, August 2005.

[30] X. L. Li, C. S. Foo, and S. K. Ng. Discovering protein complexes in dense reliable neighborhoods of protein interaction networks. *Computational systems bioinformatics / Life Sciences Society. Computational Systems Bioinformatics Conference*, 6:157–168, 2007.

[31] K. Mavromatis, N. Ivanova, K. Barry, H. Shapiro, E. Goltsman, A. C. C. McHardy, I. Rigoutsos, A. Salamov, F. Korzeniewski, M. Land, A. Lapidus, I. Grigoriev, P. Richardson, P. Hugenholtz, and N. C. C. Kyrpides. Use of simulated data sets to evaluate the fidelity of metagenomic processing methods. *Nat Methods*, April 2007.

[32] M. Meila. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, May 2007.

[33] S. Navlakha, R. Rastogi, and N. Shrivastava. Graph summarization with bounded error. In *SIGMOD 2008: Proceedings of the 2008 ACM SIGMOD International Conference on Management of data*, pages 419–432, New York, NY, USA, 2008. ACM.

[34] S. Navlakha, M. C. Schatz, and C. Kingsford. Revealing biological modules via graph summarization. *RECOMB Systems Biology, Journal of Computational Biology*, October 2008. (to appear).

[35] B. Nelson and I. Cohen. Revisiting probabilistic models for clustering with pair-wise constraints. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 673–680, New York, NY, USA, 2007. ACM.

[36] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences (PNAS)*, 103(23):8577–8582, June 2006.

[37] P. Pei and A. Zhang. A "seed-refine" algorithm for detecting protein complexes from protein interaction data. *IEEE transactions on nanobioscience*, 6(1):43–50, March 2007.

[38] J. Qiu and W. S. Noble. Predicting co-complexed protein pairs from heterogeneous data. *PLoS Computational Biology*, 4(4), April 2008.

[39] S. Raychaudhuri, J. T. Chang, F. Imam, and R. B. Altman. The computational analysis of scientific literature to define and recognize gene expression clusters. *Nucleic Acids Res*, 31(15):4553–4560, August 2003.

[40] A. W. Rives and T. Galitski. Modular organization of cellular networks. *Proceedings of the National Academy of Sciences (PNAS)*, 100(3):1128–1133, February 2003.

[41] M. P. Samanta and S. Liang. Predicting protein functions from redundancies in large-scale protein interaction networks. *Proceedings of the National Academy of Sciences (PNAS)*, 100(22):12579–12583, October 2003.

[42] P. D. Schloss and J. Handelsman. Introducing DOTUR, a computer program for defining operational taxonomic units and estimating species richness. *Applied and environmental microbiology*, 71(3):1501–1506, March 2005.

[43] P. D. Schloss and J. Handelsman. Toward a census of bacteria in soil. *PLoS Comput Biol*, 2(7), July 2006.

[44] R. Sharan, I. Ulitsky, and R. Shamir. Network-based prediction of protein function. *Nature Molecular Systems Biology*, 3, March 2007.

[45] M. L. L. Sogin, H. G. G. Morrison, J. A. A. Huber, D. M. M. Welch, S. M. M. Huse, P. R. R. Neal, J. M. M. Arrieta, and G. J. J. Herndl. Microbial diversity in the deep sea and the underexplored "rare biosphere". *Proc Natl Acad Sci U S A*, July 2006.

[46] M. Tan, E. Smith, J. Broach, and C. Floudas. Microarray data mining: A novel optimization-based approach to uncover biologically coherent structures. *BMC Bioinformatics*, 9(1), 2008.

[47] J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, 22(22):4673–4680, November 1994.

[48] P. Toronen. Selection of informative clusters from hierarchical cluster tree with gene classes. *BMC Bioinformatics*, 5:32, 2004.

[49] S. van Dongen. A new cluster algorithm for graphs. In *281*, page 42. Centrum voor Wiskunde en Informatica (CWI), ISSN 1386-3681, 31 1998.

[50] C. Wang, C. Ding, Q. Yang, and S. R. Holbrook. Consistent dissection of the protein interaction network by combining global and local metrics. *Genome Biology*, 8:R271+, December 2007.

[51] F. Warnecke, P. Luginbühl, N. Ivanova, M. Ghassemian, T. H. Richardson, J. T. Stege, M. Cayouette, A. C. Mchardy, G. Djordjevic, N. Aboushadi, R. Sorek, S. G. Tringe, M. Podar, H. G. Martin, V. Kunin, D. Dalevi, J. Madejska, E. Kirton, D. Platt, E. Szeto, A. Salamov, K. Barry, N. Mikhailova, N. C. Kyrpides, E. G. Matson, E. A. Ottesen, X. Zhang, M. Hernández, C. Murillo, L. G. Acosta, I. Rigoutsos, G. Tamayo, B. D. Green, C. Chang, E. M. Rubin, E. J. Mathur, D. E. Robertson, P. Hugenholtz, and J. R. Leadbetter. Metagenomic and functional analysis of hindgut microbiota of a wood-feeding higher termite. *Nature*, 450(7169):560–565, 2007.

[52] H. Yu, A. Paccanaro, V. Trifonov, and M. Gerstein. Predicting interactions in protein networks by completing defective cliques. *Bioinformatics*, 22(7):823–829, April 2006.

[53] X. Zhu, M. Gerstein, and M. Snyder. Getting connected: analysis and principles of biological networks. *Genes Dev*, 21(9):1010–1024, May 2007.