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

Saket Navlakha[1,2], James White[2], Niranjan Nagarajan[1,2], Mihai Pop[1,2], and Carl Kingsford[1,2]

[1] Department of Computer Science
{saket,mpop,carlk}@cs.umd.edu
[2] Center for Bioinformatics and Computational Biology,
Institute for Advanced Computer Studies
University of Maryland, College Park
whitej@umd.edu, niranjan@umiacs.umd.edu

Abstract. Hierarchical clustering is a popular method for grouping together similar elements based on a distance measure between them. In many cases, annotations for some elements are known beforehand, which can aid the clustering process. We present a novel approach for decomposing a hierarchical clustering into the clusters that optimally match a set of known annotations, as measured by the variation of information metric. Our approach is general and does not require the user to enter the number of clusters desired. We apply it to two biological domains: finding protein complexes within protein interaction networks and identifying species within metagenomic DNA samples. For these two applications, we test the quality of our clusters by using them to predict complex and species membership, respectively. We find that our approach generally outperforms the commonly used heuristic methods.

Keywords: Hierarchical Tree Decompositions, Variation of Information, Clustering, Protein Interaction Networks, Metagenomics, OTUs.

1 Introduction

Hierarchical clustering is an important tool in many applications. One application where it has been particularly useful is predicting protein membership in complexes using protein-protein interaction (PPI) networks. High-throughput experimental protocols are producing information on thousands of PPIs [52]. Embedded within these networks are protein complexes, i.e. 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 [19]. Consequently, computational methods for determining to which complexes each protein belongs have recently been developed (e.g. [3, 29, 34, 36]). A common approach to this problem is to identify clusters in the network [2, 5, 33, 34, 35, 48]. Often these clusters are

S. Batzoglou (Ed.): RECOMB 2009, LNCS 5541, pp. 400–417, 2009.

detected by hierarchically clustering the graph [1, 6, 39, 40] based on a topological distance measure such as the Czekanowski-Dice [6] or Jaccard distances. Complex memberships are then transferred to unannotated proteins by considering common known annotations within their clusters [2, 28, 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 determining bacterial species for uncharacterized DNA sequences obtained from environmental samples [44, 50, 41]. 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 [13, 16, 42]. 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 evolutionary distance is computed between every pair of sequences using a distance measure such as Jukes-Cantor [23], Kimura 2-parameter [27], or Felsenstein-84 [15]. A 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 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 in the same cluster. By estimating the composition of a microbial community, comparisons can be made of the wealth of organisms present in different environments, leading to estimations of the overall diversity. 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 or two types of operations: top-down splitting or bottom-up merging. In the network clustering setting, for example, clusters may be split based on network modularity [35] or minimum cuts [11]. Clusters to merge may be chosen based on distances such as the Dice coefficient [6], the Jaccard index [21], or correlation of shortest-path profiles [39], among others. The clustering process produces a tree ranging from the root (all nodes in one cluster) to the leaves (the nodes being clustered, each in its own cluster).

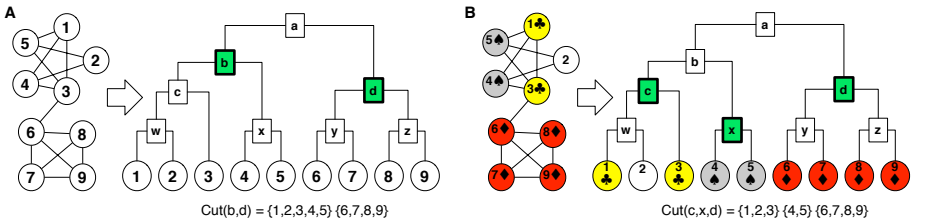

Fig. 1. Example PPI network where use of known annotations can produce a better clustering. (A) The network consists of two dense subgraphs that in most approaches would result in the hierarchical decomposition shown. By looking at the topology of the graph, it is reasonable to place proteins $\{1, 2, 3, 4, 5\}$ 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.

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 chosen tree node 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 clusterings as "hierarchical decompositions." Some hierarchical decomposition algorithms provide a natural stopping point that can be used to choose a clustering. Newman's spectral partitioning [35], for example, is a top-down approach for hierarchically decomposing nodes in a network that stops splitting clusters when any split would decrease the modularity of the clustering. Graph summarization [33], a bottom-up approach, stops merging clusters when a particular cost function is minimized. However, many algorithms do not have natural stopping points [1, 11, 24]. 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 the number of clusters or an appropriate distance threshold. Therefore, choosing an appropriate clustering implied by the hierarchy is generally a stumbling block. 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 network and its natural hierarchical decomposition. The network 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, [31]) metric. The cut is chosen such that each node is placed in a cluster. Hence, nodes with unknown annotations can be placed together in a cluster with nodes with known annotations. We can thus test the quality of a clustering based on how well we can use each cluster to predict annotations for nodes with unknown annotations (e.g. node 2 in Figure 1B). The VI-Cut method 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. [6]. The second hierarchical decomposition is created using graph summarization [33], which was recently shown [34] to outperform other graph clustering approaches such as MCL [48], MCODE [2], and Newman's spectral partitioning [35] at the task of predicting membership in protein complexes. For both types of hierarchical decompositions, we compare against the methods proposed by Brun et al. [6], Dotan-Cohen et al. [12] and also against an approach that chooses statistically enriched clusters. 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 accuracy and coverage 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 [41] 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 that defines when to stop merging clusters. We replicated six different methodologies for creating input to DOTUR that have been used in recent 16S rRNA studies [16,42,25,9,44,50]. 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. We test the quality of the VI-Cut clusters and the clusters produced by each of these six methodologies by using them to predict OTUs. In each case, the clusters created by VI-Cut produce predictions with about the same accuracy as the previous methodologies, but with large increases in coverage. 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 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 [7, 45, 47]. Clusters are then chosen by iteratively choosing high-scoring subtrees, subtrees with uniquely enriched annotations, or other similar heuristics. Recently, Dotan-Cohen et al. [12] 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.

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. No previous studies have predicted OTU annotations using a semi-supervised approach. Brun et al. [6] use PPI network data to build a hierarchical tree decomposition and extract clusters which have a majority annotation (computed using the known annotations). Other heuristics have been proposed to choose a clustering from a network decomposition [39,1,40], however, they either rely on manual inspection of the hierarchical decomposition [39], or require a similarity threshold to be input by the user [1, 40, 4].

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, for example, induces exactly $44,127,887,745,906,175,987,802$ (i.e. 4×10^{22}) 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, [31]) distance metric between the two clusterings:

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

Other methods have been used to measure the distance between clusterings, including pair-counting methods, such as the Rand [38], Mirkin [32], and Jaccard [21] indices. VI is attractive because it is a metric, information-theoretic, and, crucially, can be rewritten such that the total distance between clusterings is the sum of each cluster's contribution. Drawbacks associated with other measures are discussed by Meila [31].

In the definition of VI, the clusterings C_K and D are represented as discrete random variables taking on $|C_K|$ and $|D|$ values, respectively (one value for each cluster in the clustering). Each value corresponds to the probability that a random element chosen belongs to that cluster. This probability is computed by dividing the number of elements in the cluster by the total number of elements. In both clusterings, we ignore unannotated proteins. $H(X)$ denotes the entropy of random variable X. Intuitively, the entropy of a clustering tells us how uncertain we are about which cluster a randomly chosen element lies in. $I(X, Y)$ denotes the mutual information between the random variables X and Y. Intuitively, the mutual information gives the reduction in uncertainty regarding the cluster assignment of an element in D if its assignment in C_K is given, summed over all elements. In the following, we exploit the decomposability property of VI. Other properties of VI are explored by Meila [31].

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 $H(C_K, D) = -\sum_{x,d} p(x, d) \log p(x, d)$ $(x \in C_K, d \in D)$ so that (3) implies that $\sum_{x \in K} q(x) = 2H(C_K, D) - H(C_K)$, 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 tree 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 dynamic programming. Let **Children**(x) denote the children of a tree node x. We can compute the minimum-weight node-cut recursively:

$$\text{CutDist}(x) = \min \begin{cases} q(x) & \text{case I (default if x is a leaf)} \\ \sum_{y \in \textbf{Children}(x)} \text{CutDist}(y) & \text{case II} \end{cases} \tag{6}$$

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 of 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 CutDist(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. This algorithm does not require the user to enter the number of clusters to return.

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 node-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.

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 Annotation 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 $\min_{D \in \text{Compatible}(\mathcal{L})} VI(C, D)$.

Theorem 1. *The decision version of* Min-VI Annotation Choice *is NP-complete.*

Proof. We reduce from Exact Cover by 3-Sets (X3C) [17]. 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 Annotation 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 an 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 $D \in \text{Compatible}(\mathcal{L}_I)$ 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. □

*Problem 4 (*Min-VI Tree Cut With Annotation Choice*).* Given a set of elements E, a hierarchical decomposition T of E, and an annotation collection $\mathcal{L} = \{A_\ell \subseteq E : \ell \in L\}$ over a set of annotations L, compute $\min_{C_K, D} VI(C_K, D)$, where $K \in \text{Cut}(T)$ and $D \in \text{Compatible}(\mathcal{L})$.

Theorem 2. *The decision version of* Min-VI Tree Cut With Annotation Choice *is NP-complete.*

Proof. As above, we reduce from Exact Cover by 3-Sets (X3C) [17] (using the same notation). We construct an instance of Min-VI Tree Cut With Annotation 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 an annotation $A_\ell = \{x, y, z\}$ containing only those 3 elements. The annotation collection $\mathcal{L}_I$ consists of these A_ℓ sets and Y. We show that there is a clustering $D \in \text{Compatible}(\mathcal{L}_I)$ and node-cut K for T which induces a clustering C_K, with $VI(C_K, D) \leq 1/3 \log(|E|/3) + 2/3 \log 3/2$ 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 $C_K = \{E\}$ we get $VI(C_K, D) = 1/3 \log(|E|/3) + 2/3 \log 3/2$. 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 separate cluster and the corresponding optimal annotation 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 annotation), 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. □

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

We can test the quality of our clusters by using them 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, 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." Each method finds its clusters based only on the 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 a predicted annotation. If multiple annotations are transferred, each transferred annotation is 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 accuracy and coverage of the predictions made over 500 random samplings. (For the Snip approach we only took 10 samplings). Accuracy is the probability that a predicted annotation is correct. Coverage is the average number of elements in the test set for which a correct 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 [26] database. This network contains 5,492 proteins with 40,332 interactions. For the hierarchical decomposition, we consider only the largest connected 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 [20]. Hence, we created a high-confidence yeast interaction network from IntAct that only includes edges supported by at least two experiments. The high-confidence network contains 2,604 proteins and 8,341 interactions, fewer than half the proteins of the Y_{ppi} network. Its largest connected component, which we call $Y_{\text{high-conf}}$, contains 2,378 proteins and 8,189 interactions.

Protein complexes. Annotations for yeast complexes are from MIPS [19], ignoring the "550" section of the catalog, which represent computationally inferred complexes. This set of complexes has been widely used to assess computational methods [22, 51, 37]. 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. The average number of proteins per complex in Y_{ppi} is 5.2 (min = 1, max = 78), and in $Y_{\text{high-conf}}$ is 4.7 (min = 1, max = 67).

Hierarchical decomposition of the PPI network. We use two approaches to generate two different hierarchical tree decompositions of a PPI network. The first tree, called T_{Dice}, is built by applying the neighbor-joining algorithm BIONJ [18] to distances between proteins computed by the Czekanowski-Dice [6] 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. [6] 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 pair of nodes that give the least negative benefit 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. [6] filter false edges from their PPI network 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. [6] 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. [12] 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 involves using the hypergeometric P-value to assign 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. For Brun and Enrich, if a protein is not assigned to any chosen subtree, it is placed in a cluster by itself. 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 Predicting Operational Taxonomic Unit (OTUs)

Creation of simulated 16S sample. We obtained 1860 partial 16S rRNA gene sequences from the Ribosomal Database Project II (release 9.57 [8]) with complete taxonomic identification. These sequences were then screened for conflicting annotation information using the RDP Bayesian classifier [49], 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. The other 34% of the sample comes from the following six phyla: Actinobacteria, Bacteroidetes, Chlamydiae, Fibrobacteres, Firmicutes, and Spirochaetes. 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 [30].

Hierarchical decomposition of OTU sequences. Sequences were oriented and subsequently aligned using a multiple-sequence alignment (MSA) algorithm (such as ClustalW [46], NAST [10], or MUSCLE [14]). MSAs were trimmed so that each sequence spanned the entire alignment. From the alignment, we then used DNADIST with default parameters from the PHYLIP package [15] to compute distance matrices using the Felsenstein-84 [15] or Jukes-Cantor [23] distances. The distance matrices were then fed into DOTUR [41], 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 DOTUR to output the full hierarchical tree decomposition, which we use to find the VI-Cut clusters or OTUs 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. [25], Fulthorpe et al. [16], Schloss et al. [42], Corby-Harris et al. [9], Sogin et al. [44], and Warnecke et al. [50]. We refer to each by their first author. See Table 1 for their parameters. The Corby-Harris approach yielded nearly identical results as the Kennedy method, and is therefore omitted from the table. 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. [6] (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 accuracy and coverage of these predictions are shown in Figure 2A. The 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 accuracy and coverage 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

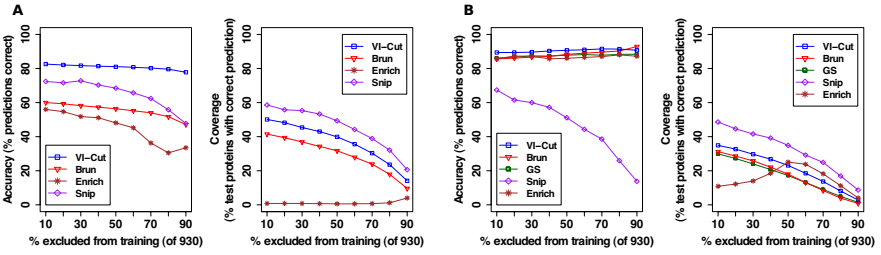

Fig. 2. Accuracy and coverage for protein complex predictions for various sizes of training sets on (A) the T_{Dice} tree, and (B) the T_{GS} tree

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 accurate in its predictions. The VI-Cut method also makes more correct annotations (larger coverage) over the entire range of sizes for the training set. The Enrich approach is even less accurate than Brun et al. and with a significantly lower coverage. 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 coverage than VI-Cut but with a greater loss in accuracy.

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 accuracy and coverage 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. Accuracy and coverage 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 accurate than every other method. The Snip method has a larger coverage, but this is negated by its poorer accuracy. Interestingly, the Enrich approach initially has an increase in recall with less training data before decreasing as one would expect. This is probably because Enrich returns substantially fewer modules as the training set size increases. As a result, the most reasonably-sized clusters are found in the middle ranges.

In general, the predictions made on T_{GS} are much more accurate 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 accuracy 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.

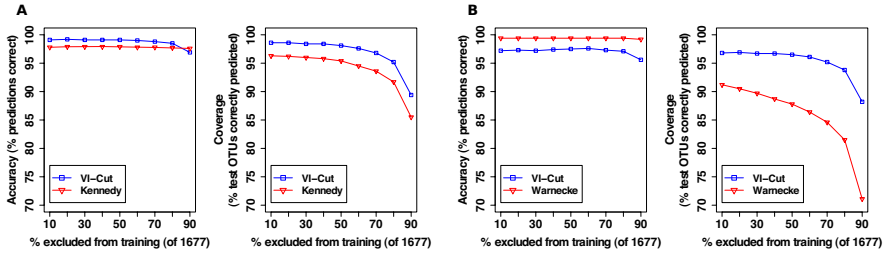

Fig. 3. Accuracy and coverage comparison of the (A) Kennedy and (B) Warnecke OTU clustering methods. Although Kennedy and Warnecke produce the same clusters regardless of the training set, the predictions they make vary due to differences in the majority annotation within each cluster.

Further, for T_{GS}, Enrich especially benefits by choosing smaller, more reasonable clusters. Overall, VI-Cut makes accurate predictions covering many proteins on both trees, unlike any other method.

Variations. Results on Y_{ppi} echo the results obtained on the $Y_{\text{high-conf}}$ network. Further, VI-Cut continued to outperform the other methods when using two other annotation transfer rules: pluarity (transfering the most common annotation), and hypergeometric enrichment [43]. However, the performance of the plurality and hypergeometric rules was generally worse than the majority rule.

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 [25], Fulthorpe [16], Schloss [42], Corby-Harris [9], Sogin [44], and Warnecke [50], described in Section 2.5. We also compare each method's ability to estimate the true 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 coverage and accuracy, respectively. Figure 3 compares these methods with the VI-Cut. Compared to Kennedy, the VI-Cut mostly makes more accurate predictions, and covers a larger number of OTUs. Although Warnecke makes slightly more accurate predictions (average gain of 2%), the VI-Cut has significantly greater coverage. For example, with 80% of the sequences in the test set, the VI-Cut makes correct predictions for 1256 sequences, compared to just 1093 by Warnecke.

Table 1. Comparison of VI-Cut with other OTU clustering approaches applied to trees constructed from DOTUR with various parameters and distance thresholds, shown in parentheses. Performance is presented for 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. **Acc.** and **Coverage** show the accuracy and coverage for each approach. **Avg. VI** shows the VI distance of the clustering to the actual OTUs.

Method	# OTUs	Acc.	Coverage	Avg. VI
Tree 1: ClustalW, Felsenstein				
Kennedy (0.03)	70	97.6	85.5	0.087
VI-Cut	42	96.9	89.4	0.050
Tree 2: NAST, Felsenstein				
Fulthorpe (0.00)	386	98.9	49.6	0.646
VI-Cut	45	95.6	87.9	0.073
Tree 3: NAST, Jukes-Cantor				
Schloss (0.03)	99	97.5	80.5	0.157
VI-Cut	42	95.6	88.2	0.073
Tree 4: NAST, Jukes-Cantor				
Warnecke (0.01)	185	99.2	71.1	0.320
VI-Cut	42	95.6	88.2	0.073
Tree 5: MUSCLE, Jukes-Cantor				
Sogin (0.03)	96	97.5	78.2	0.190
VI-Cut	43	96.1	88.2	0.046

For all six trees, we find that the VI-Cut 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 sample and the VI-Cut estimates between 42 and 45, depending on which tree is used. This is a far better and more robust 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 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 35. 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 optimally match a partial set of known annotations, as measured by the variation of information [31]. Our VI-Cut method makes improved predictions of proteins' membership in complexes and species annotations for metagenomic samples. While we showed that a generalization that allows multiple annotations per element is NP-hard, several open problems exist, such as providing an approximation guarantee on our heuristic that handles multiple annotations, and extensions that allow clusters to overlap. Nonetheless, the success of VI-Cut in two very different domains is evidence of the technique's generality.

Acknowledgements. M.P. and C.K. thank NSF for grant IIS-0812111.

References

1. Arnau, V., Mars, S., Marín, I.: Iterative cluster analysis of protein interaction data. Bioinformatics 21(3), 364–378 (2005)
2. Bader, G.D., Hogue, C.W.V.: An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinformatics 4, 2 (2003)
3. Bernard, A., Vaughn, D.S., Hartemink, A.J.: Reconstructing the topology of protein complexes. In: Speed, T., Huang, H. (eds.) RECOMB 2007. LNCS (LNBI), vol. 4453, pp. 32–46. Springer, Heidelberg (2007)
4. Böhm, C., Plant, C.: HISSCLU: a hierarchical density-based method for semi-supervised clustering. In: Proceedings of the 2008 International Conference on Extending Database Technology, pp. 440–451. ACM Press, New York (2008)
5. Brohee, S., van Helden, J.: Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinformatics 7, 488+ (2006)
6. Brun, C., Chevenet, F., Martin, D., Wojcik, J., Guenoche, A., Jacq, B.: Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. Genome Biol. 5(1), R6 (2003)
7. Buehler, E.C., Sachs, J.R., Shao, K., Bagchi, A., Ungar, L.H.: The CRASSS plug-in for integrating annotation data with hierarchical clustering results. Bioinformatics 20(17), 3266–3269 (2004)
8. Cole, J.R., Chai, B., Farris, R.J., Wang, Q., Kulam, S.A., McGarrell, D.M., Garrity, G.M., Tiedje, J.M.: The ribosomal database project (RDP-II): sequences and tools for high-throughput rRNA analysis. Nucleic Acids Res. 33, 294–296 (2005)
9. Corby-Harris, V., et al.: Geographical distribution and diversity of bacteria associated with natural populations of Drosophila melanogaster. Appl. Environ. Microbiol. 73, 3470–3479 (2007)
10. DeSantis, T.Z., Hugenholtz, P., Keller, K., Brodie, E.L., Larsen, N., Piceno, Y.M., Phan, R., Andersen, G.L.: NAST: a multiple sequence alignment server for comparative analysis of 16s rRNA genes. Nucleic Acids Res. 34(Web Server issue), W394–W399 (2006)
11. Dhillon, I.S., Guan, Y., Kulis, B.: Weighted graph cuts without eigenvectors a multilevel approach. IEEE Trans. Pattern Anal. Mach. Intell. 29(11), 1944–1957 (2007)
12. Dotan-Cohen, D., Melkman, A.A., Kasif, S.: Hierarchical tree snipping: Clustering guided by prior knowledge. Bioinformatics 23(24), 3335–3342 (2007)
13. Eckburg, P.B., Bik, E.M., Bernstein, C.N., Purdom, E., Dethlefsen, L., Sargent, M., Gill, S.R., Nelson, K.E., Relman, D.A.: Diversity of the human intestinal microbial flora. Science 308(5728), 1635–1638 (2005)
14. Edgar, R.C.: MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 32(5), 1792–1797 (2004)
15. Felsenstein, J.: PHYLIP: Phylogeny inference package (version 3.2). Cladistics 5, 164–166 (1989)
16. Fulthorpe, R.R., Roesch, L.F.W., Riva, A., Triplett, E.W.: Distantly sampled soils carry few species in common. ISME J. 2, 901–910 (2008)
17. Garey, M.R., Johnson, D.S.: Comptuers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, New York (1979)
18. Gascuel, O.: BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. Mol. Biol. Evol. 14(7), 685–695 (1997)

19. Guldener, U., Munsterkotter, M., Kastenmuller, G., Strack, N., van Helden, J., Lemer, C., Richelles, J., Wodak, S.J., Garcia-Martinez, J., Perez-Ortin, J.E., Michael, H., Kaps, A., Talla, E., Dujon, B., Andre, B., Souciet, J.L., De Mon tigny, J., Bon, E., Gaillardin, C., Mewes, H.W.: CYGD: the comprehensive yeast genome database. Nucleic Acids Res. 33(suppl. 1), D364+ (2005)
20. Hart, T.G., Ramani, A.K., Marcotte, E.M.: How complete are current yeast and human protein-interaction networks? Genome Biol. 7, 120+ (2006)
21. Jaccard, P.: Nouvelles recherches sur la distribution florale. Bulletin de la Socit Vaudoise des Sciences Naturelles, 223–270 (1908)
22. Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N.J., Chung, S., Emili, A., Snyder, M., Greenblatt, J.F., Gerstein, M.: A Bayesian networks approach for predicting protein-protein interactions from genomic data. Science 302(5644), 449–453 (2003)
23. Jukes, T.H., Cantor, C.R.: Evolution of Protein Molecules. Academic Press, London (1969)
24. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. 20(1), 359–392 (1998)
25. Kennedy, J., 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. Environ. Microbiol. 10, 1888–1902 (2008)
26. Kerrien, S., Alam-Faruque, Y., Aranda, B., Bancarz, I., Bridge, A., Derow, C., Dimmer, E., Feuermann, M., Friedrichsen, A., Huntley, R., Kohler, C., Khadake, J., Leroy, C., Liban, A., Lieftink, C., Montecchi-Palazzi, L., Orchard, S., Risse, J., Robbe, K., Roechert, B., Thorneycroft, D., Zhang, Y., Apweiler, R., Hermjakob, H.: IntAct—open source resource for molecular interaction data. Nucleic Acids Res. 35(Database issue), D561–D565 (2007)
27. Kimura, M.: A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. J. Mol. Evol. 16, 111–120 (1980)
28. King, A.D., Przulj, N., Jurisica, I.: Protein complex prediction via cost-based clustering. Bioinformatics 20(17), 3013–3020 (2004)
29. Li, X.L., Foo, C.S., Ng, S.K.: Discovering protein complexes in dense reliable neighborhoods of protein interaction networks. In: Comp. Syst. Bioinformatics Conference, vol. 6, pp. 157–168 (2007)
30. Mavromatis, K., Ivanova, N., Barry, K., Shapiro, H., Goltsman, E., McHardy, A.C.C., Rigoutsos, I., Salamov, A., Korzeniewski, F., Land, M., Lapidus, A., Grigoriev, I., Richardson, P., Hugenholtz, P., Kyrpides, N.C.C.: Use of simulated data sets to evaluate the fidelity of metagenomic processing methods. Nat. Methods, 495–500 (2007)
31. Meila, M.: Comparing clusterings—an information based distance. J. Multivariate Anal. 98(5), 873–895 (2007)
32. Mirkin, B.: Mathematical classification and clustering. J. Global Optim. 12(1), 105–108 (1998)
33. Navlakha, S., Rastogi, R., Shrivastava, N.: Graph summarization with bounded error. In: Proceedings of the 2008 ACM SIGMOD Conference, pp. 419–432 (2008)
34. Navlakha, S., Schatz, M.C., Kingsford, C.: Revealing biological modules via graph summarization. J. Comp. Biol. 16(2), 253–264 (2009)
35. Newman, M.E.J.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. USA 103(23), 8577–8582 (2006)
36. Pei, P., Zhang, A.: A "seed-refine" algorithm for detecting protein complexes from protein interaction data. IEEE T. Nanobiosci. 6(1), 43–50 (2007)

37. Qiu, J., Noble, W.S.: Predicting co-complexed protein pairs from heterogeneous data. PLoS Comp. Biol. 4(4) (2008)
38. Rand, W.M.: Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. 66(336), 846–850 (1971)
39. Rives, A.W., Galitski, T.: Modular organization of cellular networks. Proc. Natl. Acad. Sci. USA 100(3), 1128–1133 (2003)
40. Samanta, M.P., Liang, S.: Predicting protein functions from redundancies in large-scale protein interaction networks. Proc. Natl. Acad. Sci. USA 100(22), 12579–12583 (2003)
41. Schloss, P.D., Handelsman, J.: Introducing DOTUR, a computer program for defining operational taxonomic units and estimating species richness. Appl. Environ. Microbiol. 71(3), 1501–1506 (2005)
42. Schloss, P.D., Handelsman, J.: Toward a census of bacteria in soil. PLoS Comp. Biol. 2(7), e92 (2006)
43. Sharan, R., Ulitsky, I., Shamir, R.: Network-based prediction of protein function. Nat. Mol. Syst. Biol. 3, 88 (2007)
44. Sogin, M.L.L., Morrison, H.G.G., Huber, J.A.A., Welch, D.M.M., Huse, S.M.M., Neal, P.R.R., Arrieta, J.M.M., Herndl, G.J.J.: Microbial diversity in the deep sea and the underexplored "rare biosphere". Proc. Natl. Acad. Sci. USA 103(32), 12115–12120 (2006)
45. Tan, M., Smith, E., Broach, J., Floudas, C.: Microarray data mining: A novel optimization-based approach to uncover biologically coherent structures. BMC Bioinformatics 9(1), 268 (2008)
46. Thompson, J.D., Higgins, D.G., Gibson, T.J.: 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 (1994)
47. Toronen, P.: Selection of informative clusters from hierarchical cluster tree with gene classes. BMC Bioinformatics 5, 32 (2004)
48. van Dongen, S.: A cluster algorithm for graphs. Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam (2000)
49. Wang, Q., Garrity, G.M., Tiedje, J.M., Cole, J.R.: Naive bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. Appl. Environ. Microbiol. 73(16), 5261–5267 (2007)
50. Warnecke, F., Luginbühl, P., Ivanova, N., Ghassemian, M., Richardson, T.H., Stege, J.T., Cayouette, M., Mchardy, A.C., Djordjevic, G., Aboushadi, N., Sorek, R., Tringe, S.G., Podar, M., Martin, H.G., Kunin, V., Dalevi, D., Madejska, J., Kirton, E., Platt, D., Szeto, E., Salamov, A., Barry, K., Mikhailova, N., Kyrpides, N.C., Matson, E.G., Ottesen, E.A., Zhang, X., Hernández, M., Murillo, C., Acosta, L.G., Rigoutsos, I., Tamayo, G., Green, B.D., Chang, C., Rubin, E.M., Mathur, E.J., Robertson, D.E., Hugenholtz, P., Leadbetter, J.R.: Metagenomic and functional analysis of hindgut microbiota of a wood-feeding higher termite. Nature 450(7169), 560–565 (2007)
51. Yu, H., Paccanaro, A., Trifonov, V., Gerstein, M.: Predicting interactions in protein networks by completing defective cliques. Bioinformatics 22(7), 823–829 (2006)
52. Zhu, X., Gerstein, M., Snyder, M.: Getting connected: analysis and principles of biological networks. Genes Dev. 21(9), 1010–1024 (2007)