Original Paper

A Clustering Method for Analysis of Sequence Similarity Networks of Proteins Using Maximal Components of Graphs

Morihiro Hayashida,[†1] Tatsuya Akutsu[†1]
and Hiroshi Nagamochi[†2]

This paper proposes a novel clustering method based on graph theory for analysis of biological networks. In this method, each biological network is treated as an undirected graph and edges are weighted based on similarities of nodes. Then, maximal components, which are defined based on edge connectivity, are computed and the nodes are partitioned into clusters by selecting disjoint maximal components. The proposed method was applied to clustering of protein sequences and was compared with conventional clustering methods. The obtained clusters were evaluated using P-values for GO (GeneOntology) terms. The average P-values for the proposed method were better than those for other methods.

1. Introduction

Many clustering methods have been developed and/or applied for analyzing various kinds of biological data. Among them, such hierarchical clustering methods as the single-linkage, complete-linkage and average-linkage methods have been widely used [1),2)]. However, these clustering methods are based on similarities between two elements or two clusters, and relations with other elements or clusters are not so much taken into account.

Relations between biological entities are often represented as networks or (almost equivalently) graphs. For example, nodes are proteins in a protein-protein interaction network, and two nodes are connected by an edge if the corresponding proteins interact with each other. For another example, nodes are again proteins in a sequence similarity network of proteins, and two nodes are connected by an edge if the corresponding protein sequences are similar to each other. Moreover, in this case, similarity scores are assigned as weights of edges. Since these networks are considered to have much information, clustering based on network structures might be useful. Of course, conventional clustering methods can be applied to clustering of nodes in these networks [1),2)]. But, information on network structure is not so much taken into account by these methods. For an extreme example, suppose that the network is a complete graph and all edges have the same weight. Then, all the nodes should be put into one cluster and sub-clusters should not be created. However, conventional clustering methods create many sub-clusters. Therefore, clustering methods that utilize structural information on a network should be developed. Though clustering methods utilizing structural information have been developed [3),4)], many of these are heuristic and/or recursive and thus it is unclear which properties are satisfied for the final clusters.

Tuji, et al. applied two clustering methods based on network structure, DPClus algorithm [5)] and Newman algorithm [6)] to protein-protein interaction networks [7)]. DPClus algorithm calculates density and cluster property of each cluster. The density of each cluster is defined to be the ratio of the number of edges present in the cluster to the maximum capable number of edges in the cluster, and the cluster property of each node for each cluster is defined to be the ratio of the total number of edges between the node and each of the nodes of the cluster to the average number of edges of a node in the cluster. Newman algorithm calculates modularity which is defined to be the fraction of edges that fall within clusters, minus the expected value of the same quantity if edges fall at random. Tuji, et al. compared the above clustering methods, and concluded that the results of visualization were quite different depending on clustering algorithms.

On the other hand, in graph theory and graph algorithms, the Gomory-Hu tree is well-known [8)] where it is defined for an undirected

†1 Bioinformatics Center, Institute for Chemical Research, Kyoto University
†2 Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University

network with weighted edges. This tree essentially contains all information on minimum cuts for all pairs of nodes. It is known that a Gomory-Hu tree can be computed efficiently using a maximum flow algorithm. Furthermore, maximal components can be efficiently computed from a Gomory-Hu tree [9], where a maximal component is a set of nodes with high connectivity (the precise definition is given in Section 2). It is known that a set of maximal components constitutes a laminar structure, which is essentially a hierarchical structure.

Based on the above facts, we develop a novel clustering method for an undirected network. In this method, nodes are partitioned into clusters by selecting disjoint maximal components. The method works in $O(n^2 m \log(n^2/m))$ time, where n and m are the numbers of nodes and edges, respectively. The Gomory-Hu tree was already applied to analysis of protein folding pathways [10)–12)]. However, to our knowledge, it was not applied to analysis of large scale protein sequence networks. Moreover, as to be shown in Section 3, our method employs additional ideas to effectively utilize the Gomory-Hu tree.

In this paper, we apply the proposed clustering method to classification of protein sequences and compare with the single-linkage, complete-linkage and average-linkage methods. For that purpose, we construct a sequence similarity network from protein sequences in Saccharomyces Genome Database (SGD) database [13] using BLAST [14] and resulting E-values. We evaluate the computed clusters using P-values for GO (GeneOntology) terms. The results suggest the effectiveness of the proposed method.

The organization of the paper is as follows. In Section 2, we review maximal components of undirected graphs along with related graph theoretical concepts, and conventional clustering methods. In Section 3, we present our proposed method for selecting disjoint clusters from a hierarchical structure representing all maximal components. In Section 4, we show the results on computational experiment. Finally, we conclude with future work.

2. Preliminaries

In this section, we review edge-connectivity and maximal components [9]. We also review three conventional hierarchical clustering methods: single-linkage, average-linkage and complete-linkage clustering methods.

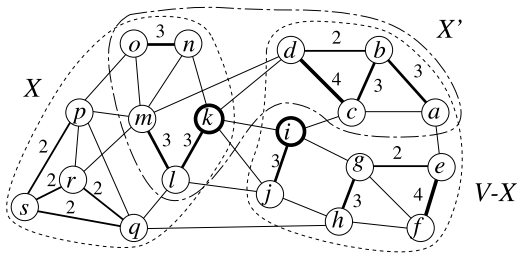

Fig. 1 Illustration for minimum (k, i)-cut of a graph $G = (V, E)$ with $V = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s\}$ and E, where each number denotes the weight of the edge, and edges without numbers are weighted by 1. Each set of nodes surrounded with a dashed line is a maximal component of G. For the set $X = \{k, l, m, n, o, p, q, r, s\}$, $d_G(X) = \sum_{p \in X, q \in V-X} c_G(p, q) = 6$. For the other set $X' = \{a, b, c, d, k, l, m, n, o\}$, $d_G(X') = 9$. $\lambda_G(k, i) = \min_{k \in X, i \in V-X} d_G(X) = 6$.

2.1 Edge-connectivity

Let $G = (V, E)$ be an undirected edge-weighted graph with a vertex set V and an edge set E, where each edge e is weighted by a non-negative real $c_G(e) \in \Re^+$. We define the edge-connectivity $\lambda_G(u, v)$ between two nodes u and v as follows:

$$\lambda_G(u, v) = \min_{\{X \subseteq V | u \in X, v \in V-X\}} \sum_{p \in X, q \in V-X} c_G(p, q). \quad (1)$$

A subset X of V is called (u, v)-cut if $u \in X$ and $v \in V - X$, or $u \in V - X$ and $v \in X$. Among them, a (u, v)-cut X which gives a minimum $\lambda_G(u, v)$ is called a minimum (u, v)-cut. **Figure 1** shows an example of a minimum cut.

2.2 Maximal Components

Definition 1 A subset X of V is called a *maximal component* if it satisfies the following conditions,

$$\forall u, v \in X \quad \lambda_G(u, v) \geq l, \quad (2)$$
$$\forall u \in X, \forall v \in V - X \quad \lambda_G(u, v) < l, \quad (3)$$

where $l = \min_{u,v \in X} \lambda_G(u, v)$. Such a subset X is also called an *l-edge-connected component*.

Figure 2 shows an example of maximal components. Definition 1 means that the internal nodes of a maximal component are connected with each other more strongly than with any other external nodes. Moreover, for relations between maximal components, connectivities between nodes of an internal maximal component are stronger than (and equal to) those between nodes of an external maximal compo-

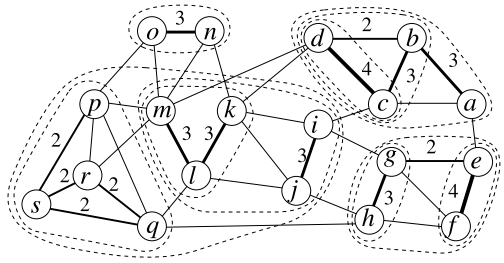

Fig. 2 Illustration for maximal components of the graph G in Fig. 1. For example, the set $X = \{a, b, c, d\}$ is a maximal component because $\lambda_G(u, v) \geq 5$ for any $u, v \in X$, $\lambda_G(u, v) < 5$ for any $u \in X$ and $v \in V - X$, and $\min_{u,v \in X} \lambda_G(u, v) = 5$. It is also called a 5-edge-connected component.

nent which include the internal maximal components.

Definition 2 A family $\boldsymbol{\chi} \subseteq 2^V$ is called *laminar* if $X \cap Y = \emptyset$, $X \subset Y$, or $Y \subset X$ for any distinct sets $X, Y \in \boldsymbol{\chi}$.

In this paper, $X \subset Y$ means that X cannot be equal to Y, and $X \subseteq Y$ means that X can be Y. A laminar family $\boldsymbol{\chi}$ is represented by a rooted tree $\boldsymbol{\tau} = (\boldsymbol{\nu}, \boldsymbol{\epsilon})$. The node set $\boldsymbol{\nu}$ is defined by $\boldsymbol{\nu} = \boldsymbol{\chi} \cup \{V\}$, where V corresponds to the root of $\boldsymbol{\tau}$. Let t_X denote a node corresponding to a set $X \in \boldsymbol{\nu}$. For two nodes t_X and t_Y in $\boldsymbol{\tau}$, t_X is a child of t_Y if and only if $X \subset Y$ holds and $\boldsymbol{\chi}$ contains no set Z with $X \subset Z \subset Y$.

Theorem 1 Let $\boldsymbol{\chi}(G)$ denote the set of all maximal components of G. Then, $\boldsymbol{\chi}(G)$ is a laminar family.

Proof. We assume that there exist three nodes x, y and z so that $x \in X - Y$, $y \in Y - X$, and $z \in X \cap Y$ for two maximal components $X, Y \in \boldsymbol{\chi}(G)$, where X is an l-edge-connected component and Y is an h-edge connected component. We can assume without loss of generality that $l \geq h$. From $x, z \in X$ and Eq. (2) of the definition of maximal components for X, we have $\lambda_G(x, z) \geq l \geq h$. On the other hand, from $x \notin Y, z \in Y$ and Inequality (3) for Y, we have $\lambda_G(x, z) < h$. It contradicts our assumption. □

2.3 Linkage Methods

We briefly review three linkage clustering methods: single linkage (or nearest neighbor method), complete linkage (or farthest neighbor method), and average linkage. Each method starts with a set of clusters, where each cluster consists of a single distinct node. Then, two clusters having the minimum distance are merged into one cluster. This procedure is repeated until there is only one cluster as follows.

Procedure Linkage_Clustering
  Input : a set of nodes V and
    distances $d(x, y)$ for all $x, y \in V$
  Output : a laminar family $\boldsymbol{\tau}$
Begin
  $\boldsymbol{\chi} := \{\{x\} | x \in V\}$
  $\boldsymbol{\tau} = \boldsymbol{\chi}$
  while $|\boldsymbol{\chi}| \geq 2$
    $(X, Y) := \text{argmin}_{(X,Y) \in \boldsymbol{\chi} \times \boldsymbol{\chi}, X \neq Y} D(X, Y)$
    Remove X and Y from $\boldsymbol{\chi}$
    Add $X \cup Y$ into $\boldsymbol{\chi}$ and $\boldsymbol{\tau}$
  end
  return $\boldsymbol{\tau}$
End

In this procedure, the distance $D(X, Y)$ between two clusters X and Y is defined using $d(x, y)$ in a different way depending on a clustering method:

$$D(X,Y) = \begin{cases} \min\limits_{x \in X, y \in Y} d(x, y) \\ \text{(for single linkage)} \\ \max\limits_{x \in X, y \in Y} d(x, y) \\ \text{(for complete linkage)} \\ \dfrac{1}{|X||Y|} \sum\limits_{x \in X, y \in Y} d(x, y) \\ \text{(for average linkage)} \end{cases} . \quad (4)$$

It should be noted that the distance between two nodes should be small if the similarity between these nodes is high, whereas the weight of the edge between these two nodes should be large. Since we are going to use the similarity score (which is high for similar nodes), we use modified versions of these clustering algorithms. In the modified versions, the clusters with the maximum score are merged, instead of the clusters with the minimum distance. Moreover, 'min' and 'max' in Eq. (4) are exchanged.

3. Selection of Disjoint Clusters from Hierarchical Structure

The set of all maximal components $\boldsymbol{\chi}(G)$ of a graph G provides a hierarchical structure which can be represented as a rooted tree $\boldsymbol{\tau}(G)$ because the set $\boldsymbol{\chi}(G)$ is a laminar family. This structure gives a kind of hierarchical clustering. However, what we need is a set of disjoint clusters because we are interested in classification of protein sequences. That is, input nodes should be partitioned into disjoint clus-

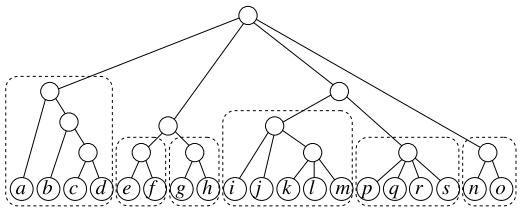

Fig. 3 The rooted tree representation of maximal components $\boldsymbol{\chi}(G)$ of the graph G in Fig. 1. The six sets of nodes surrounded by dashed lines are the resulting clusters provided by the procedure SelectLaminar.

ters. Thus, we propose a method to find disjoint clusters from $\boldsymbol{\chi}(G)$. This selection method is based on a simple idea that a cluster does not include more than one sufficiently clustered set. In our method, a set of maximal components $\boldsymbol{\chi}(G)$ of the graph G is first computed using a Gomory-Hu tree. And then, disjoint maximal components are selected in a bottom-up manner, based on the tree structure $\boldsymbol{\tau}(G)$. The detailed procedure is given below. It should be noted that $|X_t|$ denotes the number of nodes in G that are contained in X_t.

Procedure SelectLaminar
  Input : a laminar family $\boldsymbol{\chi}$
  Output : a set of clusters $\boldsymbol{\chi}_c \subseteq \boldsymbol{\chi}$
Begin
  $\boldsymbol{\tau}$:= (the rooted tree made from $\boldsymbol{\chi}$)
  $\boldsymbol{\chi}_c := \emptyset$
  repeat
    X_p := (a parent node of not marked deepest leaves of $\boldsymbol{\tau}$)
    repeat
      $X_s := X_p$
      X_p := (the parent node of X_s)
    until (X_p has a child X_t except X_s such that $|X_t| \geq 2$)
    Add all the child nodes of X_p to $\boldsymbol{\chi}_c$
    Mark all the descendant leaves of X_p in $\boldsymbol{\tau}$
  until (all the leaves of $\boldsymbol{\tau}$ are marked)
  return $\boldsymbol{\chi}_c$
End

This procedure outputs a subset $\boldsymbol{\chi}_c = \{X_1, \ldots, X_m\}$ from the laminar family $\boldsymbol{\chi}(G)$ of all maximal components of a graph G such that $X_i \cap X_j = \emptyset$ holds for any two sets $X_i \neq X_j \in \boldsymbol{\chi}_c$, and $\bigcup_{i=1}^{m} X_i = V$ holds. **Figure 3** shows an example. This procedure provides the clusters according to the hierarchical structure.

4. Experimental Results

4.1 Data and Implementation

In order to evaluate the proposed clustering method, we applied clustering methods to classification of protein sequences based on the pairwise similarity. We used 5888 protein sequences (The file name is "orf_trans.20040827.fasta") from SGD [13]. This file contains the translations of all systematically named ORFs except dubious ORFs and pseudo-genes. We calculated the similarities between all pairs of the proteins using a BLAST search [14] with an E-value threshold of 0.1. An edge between two nodes exists only when the E-value between the proteins is less than or equal to 0.1. All isolated nodes (i.e., nodes with degree 0) are removed. As a result, 32,484 pairwise similarities and 4,533 nodes were detected.

As an edge-weight, we used the integer part of $-3000 \log_{10} h$ for the E-value h of $10^{-\frac{1000}{3}} < h \leq 0.1$, and 10^6 for $0 \leq h \leq 10^{-\frac{1000}{3}}$. This mapping was injective for all the E-values of the data. It should be noted that the similarity between proteins is large when the E-value is small, and comparison operations of floating point numbers can cause incorrect results.

We solved maximum flow problems with HIPR (version 3.5) [15] which is an implementation of the algorithm developed by Goldberg and Tarjan [16], and constructed a Gomory-Hu tree [8] for an edge-weighted graph G to obtain all the maximal components of G from the tree.

4.2 Results

To evaluate the performance of our clustering method, we used GO-TermFinder (version 0.7) [17]. GO terms are structured and controlled vocabularies which explain gene products with respect to biological processes, cellular components, and molecular functions. A GO term is linked not only to genes and gene products in several databases, but also to other GO terms.

To find the most suitable GO term for a specified list (cluster) of genes, this software calculates a P-value using the hypergeometric distribution as follows:

$$P = 1 - \sum_{i=0}^{k-1} \frac{\left(\begin{array}{c} M \\ i \end{array} \right) \left(\begin{array}{c} N-M \\ n-i \end{array} \right)}{\left(\begin{array}{c} N \\ n \end{array} \right)}$$

Table 1 Results for three ontologies on biological processes, cellular components, and molecular functions, by four clustering methods using maximal components, single linkage, complete linkage, and average linkage.

Method	# of clusters	Process		Component		Function	
Maximal component	869	−8.9462	2,618	−5.9189	2,641	−10.657	2,624
Single linkage	1,176	−5.2346	2,947	−4.5076	2,970	−4.7721	2,903
Complete linkage	1,509	−3.0674	3,258	−2.3149	3,391	−3.8539	3,050
Average linkage	1,440	−3.2556	3,692	−2.4423	3,761	−4.1007	3,508

Left column: the average of logarithm of corrected P-values.
Right column: the total number of proteins included in annotated clusters.

$$= \sum_{i=k}^{n} \frac{\begin{pmatrix} M \\ i \end{pmatrix} \begin{pmatrix} N-M \\ n-i \end{pmatrix}}{\begin{pmatrix} N \\ n \end{pmatrix}} \qquad (5)$$

where N is the total number of genes, M is the total number of genes annotated by the specific GO term, n is the number of genes in the cluster, and k is the number of genes annotated by the specific GO term in the cluster. P-value means the probability of seeing k or more genes with an annotation by a GO term among n genes in the list, given that M in the population of N have that annotation. For example, $P = 1$ holds if none of the genes in the specified list are annotated by the GO term. On the other hand, if all the genes are annotated, $P = \frac{M(M-1)\cdots(M-n+1)}{N(N-1)\cdots(N-n+1)}$ is very small because M is usually much smaller than N.

By performing more than one statistical test, the probability of falsely identifying a test to be statistically significant can increase. In order to avoid that many false positive GO terms are chosen, we need to adjust such a probability or the alpha level (cut-off for P-values), that is often set to 0.05, for multiple hypotheses. The Bonferroni method adjusts the alpha level of each individual test downwards. Alternatively, GO-TermFinder adjusts P-values by multiplying by the number of hypotheses that were tested, and the alpha level can be kept the same level [18]. We employed these corrected P-values to evaluate clustering results.

We used three types of ontologies on biological processes, cellular components, and molecular functions (Their file names are "process.ontology.2005-08-01", "component.ontology.2005-08-01", and "function.ontology.2005-08-01"). We obtained these files also from SGD [13].

We compared the proposed method with other clustering methods using single linkage, complete linkage, and average linkage. These clustering methods usually produce a hierarchical clustering. In order to obtain non-hierarchical clustering results, we applied our proposed procedure in the previous section, SelectLaminar, to their results.

Table 1 shows the averages of logarithms of corrected P-values over all 4,533 proteins and the number of clusters. Among these proteins, there were some proteins which could not be annotated by GO-TermFinder. Therefore, we regarded a corrected P-value as 1 for such proteins, and calculated the averages. We see from the table that our clustering method using maximal components outperformed other methods. For every ontology, the average of our method was lower than that of others. It means that our method classified protein sequences into protein functions better than others.

Figure 4 shows the distributions of the number of proteins in a cluster for resulting clusters of clustering methods using maximal components, single linkage, complete linkage, and average linkage. These methods had a similar distribution and many clusters were concentrated in small sizes. In our selection method, SelectLaminar, a set having more than one node is considered as an independent cluster through the condition $|X_t| \geq 2$, though we can relax the condition like $|X_t| \geq 3$ to obtain larger clusters.

Figures 5, **6** and **7** show logarithms of corrected P-values on 800 lowest proteins for the ontologies on biological processes, cellular components, and molecular functions, respectively. For every ontology, corrected P-values of our method were lower than others. The distributions of complete linkage and average linkage had similar behavior. For the ontologies on biological processes and cellular components, corrected P-values of single linkage were close to those of our method. In particular, our method provided good results for molecular functions.

Tables 2, **3** and **4** show GO terms with low-

Table 2 GO terms with lowest 8 corrected P-value for the ontology of biological processes in resulting clusters of clustering methods using maximal components, single linkage, complete linkage, and average linkage.

Rank	Maximal component	
1	GO:0006319 (Ty element transposition)	2.7522e-190
2	GO:0006468 (protein amino acid phosphorylation)	2.4181e-113
3	GO:0008643 (carbohydrate transport)	1.2509e-43
4	GO:0006865 (amino acid transport)	1.0052e-37
5	GO:0006511 (ubiquitin-dependent protein catabolism)	9.4800e-24
6	GO:0006810 (transport)	1.2224e-21
7	GO:0006081 (aldehyde metabolism)	8.1134e-21
8	GO:0016567 (protein ubiquitination)	1.1405e-19
Rank	Single linkage	
1	GO:0006319 (Ty element transposition)	3.0396e-176
2	GO:0006081 (aldehyde metabolism)	4.0950e-19
3	GO:0006530 (asparagine catabolism)	3.5363e-16
4	GO:0006166 (purine ribonucleoside salvage)	5.0151e-15
5	GO:0045039 (mitochondrial inner membrane protein import)	3.1055e-14
6	GO:0046839 (phospholipid dephosphorylation)	7.8293e-14
7	GO:0005992 (trehalose biosynthesis)	3.4570e-13
8	GO:0006913 (nucleocytoplasmic transport)	4.4109e-13
Rank	Complete linkage	
1	GO:0006319 (Ty element transposition)	3.7058e-81
2	GO:0006319 (Ty element transposition)	9.1783e-55
3	GO:0008645 (hexose transport)	1.1156e-20
4	GO:0006319 (Ty element transposition)	4.1098e-18
5	GO:0000209 (protein polyubiquitination)	5.7229e-17
6	GO:0006530 (asparagine catabolism)	3.5363e-16
7	GO:0006081 (aldehyde metabolism)	2.1634e-15
8	GO:0006166 (purine ribonucleoside salvage)	5.0151e-15
Rank	Average linkage	
1	GO:0006319 (Ty element transposition)	4.4023e-79
2	GO:0006319 (Ty element transposition)	9.1783e-55
3	GO:0008645 (hexose transport)	1.1156e-20
4	GO:0006081 (aldehyde metabolism)	4.0950e-19
5	GO:0006319 (Ty element transposition)	4.1098e-18
6	GO:0006530 (asparagine catabolism)	3.5363e-16
7	GO:0006166 (purine ribonucleoside salvage)	5.0151e-15
8	GO:0045039 (mitochondrial inner membrane protein import)	3.1055e-14

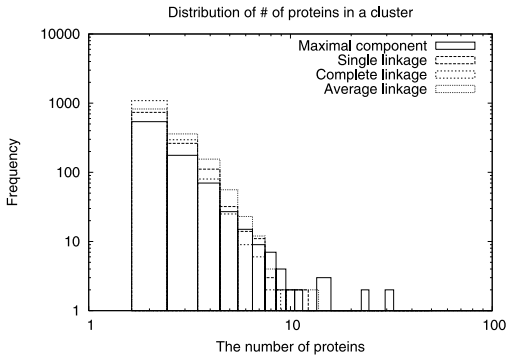

Fig. 4 Distributions of the number of proteins in a cluster for resulting clusters of clustering methods using maximal components, single linkage, complete linkage, and average linkage.

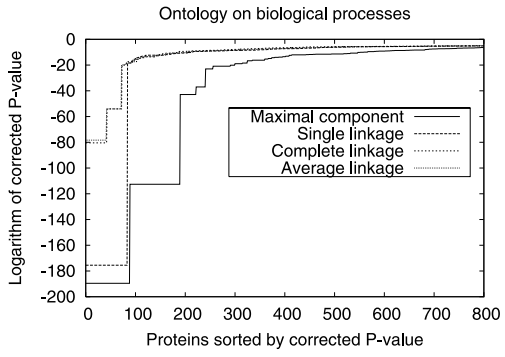

Fig. 5 Logarithms of corrected P-values on 800 lowest proteins for ontology on biological processes.

est 8 corrected P-value in resulting clusters of clustering methods using maximal components, single linkage, complete linkage, and average linkage for the ontology of biological processes, cellular components, and molecular functions, respectively. In both complete linkage and average linkage for the ontology of biological pro-

Table 3 GO terms with lowest 8 corrected P-value for the ontology of cellular components in resulting clusters of clustering methods using maximal components, single linkage, complete linkage, and average linkage.

Rank	Maximal component	
1	GO:0000943 (retrotransposon nucleocapsid)	1.7383e-191
2	GO:0005839 (proteasome core complex)	1.0925e-40
3	GO:0005740 (mitochondrial membrane)	4.6006e-36
4	GO:0005886 (plasma membrane)	2.1740e-25
5	GO:0005886 (plasma membrane)	4.0445e-24
6	GO:0005832 (chaperonin-containing T-complex)	3.7844e-22
7	GO:0042555 (MCM complex)	1.7940e-19
8	GO:0005940 (septin ring)	2.2970e-18
Rank	Single linkage	
1	GO:0000943 (retrotransposon nucleocapsid)	2.7425e-177
2	GO:0019773 (proteasome core complex, alpha-subunit complex)	1.8389e-22
3	GO:0008540 (proteasome regulatory particle, base subcomplex)	1.3396e-17
4	GO:0005946 (alpha,alpha-trehalose-phosphate synthase complex)	1.7976e-13
5	GO:0042719 (mitochondrial intermembrane space protein transporter complex)	1.3137e-12
6	GO:0042555 (MCM complex)	3.7336e-12
7	GO:0019774 (proteasome core complex, beta-subunit complex)	7.7437e-12
8	GO:0000307 (cyclin-dependent protein kinase holoenzyme complex)	1.7423e-11
Rank	Complete linkage	
1	GO:0000943 (retrotransposon nucleocapsid)	1.4767e-81
2	GO:0000943 (retrotransposon nucleocapsid)	4.5547e-55
3	GO:0019773 (proteasome core complex, alpha-subunit complex)	1.8389e-22
4	GO:0000943 (retrotransposon nucleocapsid)	2.6265e-18
5	GO:0005946 (alpha,alpha-trehalose-phosphate synthase complex)	1.7976e-13
6	GO:0019774 (proteasome core complex, beta-subunit complex)	7.7437e-12
7	GO:0000307 (cyclin-dependent protein kinase holoenzyme complex)	1.7423e-11
8	GO:0008540 (proteasome regulatory particle, base subcomplex)	2.7877e-11
Rank	Average linkage	
1	GO:0000943 (retrotransposon nucleocapsid)	1.7874e-79
2	GO:0000943 (retrotransposon nucleocapsid)	4.5547e-55
3	GO:0019773 (proteasome core complex, alpha-subunit complex)	1.8389e-22
4	GO:0000943 (retrotransposon nucleocapsid)	2.6265e-18
5	GO:0000307 (cyclin-dependent protein kinase holoenzyme complex)	1.6203e-14
6	GO:0005946 (alpha,alpha-trehalose-phosphate synthase complex)	1.7976e-13
7	GO:0042719 (mitochondrial intermembrane space protein transporter complex)	1.3137e-12
8	GO:0042555 (MCM complex)	3.7336e-12

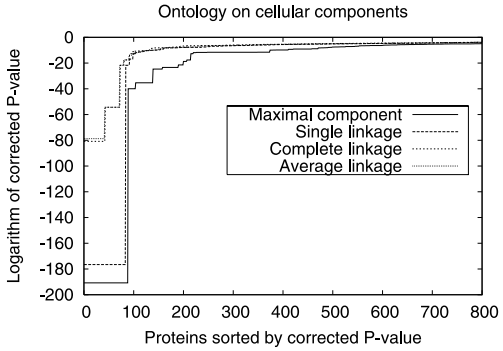

Fig. 6 Logarithms of corrected P-values on 800 lowest proteins for ontology on cellular components.

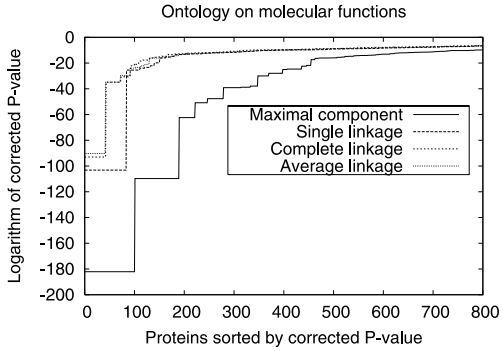

Fig. 7 Logarithms of corrected P-values on 800 lowest proteins for ontology on molecular functions.

cesses, the same GO term (GO:0006319 Ty element transposition) was annotated to the first and second lowest clusters. It means that a cluster having the GO term was divided into two or more clusters by the methods. It was also observed in maximal components, complete linkage, and average linkage for the ontology of cellular components.

Table 4 GO terms with lowest 8 corrected P-value for the ontology of molecular functions in resulting clusters of clustering methods using maximal components, single linkage, complete linkage, and average linkage.

Rank	Maximal component	
1	GO:0004672 (protein kinase activity)	7.6717e-183
2	GO:0003723 (RNA binding)	1.8308e-110
3	GO:0003924 (GTPase activity)	4.1636e-63
4	GO:0003724 (RNA helicase activity)	1.5530e-51
5	GO:0051119 (sugar transporter activity)	2.1704e-48
6	GO:0005215 (transporter activity)	7.4571e-40
7	GO:0015171 (amino acid transporter activity)	1.6321e-39
8	GO:0008639 (small protein conjugating enzyme activity)	1.2381e-38
Rank	Single linkage	
1	GO:0003723 (RNA binding)	6.6055e-104
2	GO:0004386 (helicase activity)	3.3796e-26
3	GO:0004190 (aspartic-type endopeptidase activity)	1.0869e-24
4	GO:0018456 (aryl-alcohol dehydrogenase activity)	4.8306e-24
5	GO:0000293 (ferric-chelate reductase activity)	7.5718e-23
6	GO:0016820 (hydrolase activity)	1.5294e-20
7	GO:0003993 (acid phosphatase activity)	8.5729e-17
8	GO:0004749 (ribose phosphate diphosphokinase activity)	9.6445e-17
Rank	Complete linkage	
1	GO:0003964 (RNA-directed DNA polymerase activity)	1.0121e-93
2	GO:0003723 (RNA binding)	1.5609e-35
3	GO:0004386 (helicase activity)	2.1096e-31
4	GO:0000293 (ferric-chelate reductase activity)	7.5718e-23
5	GO:0005353 (fructose transporter activity)	9.4981e-22
6	GO:0004840 (ubiquitin conjugating enzyme activity)	3.4803e-19
7	GO:0018456 (aryl-alcohol dehydrogenase activity)	1.9535e-18
8	GO:0004190 (aspartic-type endopeptidase activity)	2.2326e-18
Rank	Average linkage	
1	GO:0003964 (RNA-directed DNA polymerase activity)	5.2819e-91
2	GO:0003723 (RNA binding)	1.5609e-35
3	GO:0004386 (helicase activity)	1.9841e-30
4	GO:0004190 (aspartic-type endopeptidase activity)	1.0869e-24
5	GO:0018456 (aryl-alcohol dehydrogenase activity)	4.8306e-24
6	GO:0000293 (ferric-chelate reductase activity)	7.5718e-23
7	GO:0005353 (fructose transporter activity)	9.4981e-22
8	GO:0003756 (protein disulfide isomerase activity)	8.5729e-17

As for CPU time, the proposed method is reasonably fast. Though the worst case time complexity of the proposed method is $O(n^2 m \log(n^2/m))$, it is expected to work faster in practice. Indeed, the proposed method took 6.3 sec. for clustering of a graph with 4,533 nodes on a Linux PC with Xeon 3.6 GHz CPU and 4 GB memory. Though the single-linkage clustering took only 0.024 sec., our proposed method produced better results. Moreover, for memory usage, the space complexity of our method is $O(mn)$. Actually, our method used about 5.5 M bytes memory for the graph with 4533 nodes.

5. Conclusion

We developed a clustering method for analysis of biological data. The proposed method makes use of maximal components, where a maximal component can be characterized as a subgraph having maximal edge connectivity. Since a set of maximal components constitutes a hierarchical structure, we developed a method to select disjoint clusters from the hierarchical structure. We compared the proposed method with the single linkage, complete linkage, and average linkage clustering methods using a sequence similarity network of proteins. Our proposed method outperformed these three methods in terms of the corrected P-values provided by GO-TermFinder, and classified protein sequences into protein functions better than the three methods.

We did not compare clustering methods other than the linkage methods with our method in this study since a number of clustering methods have been proposed and it is unclear which methods are appropriate for analysis of se-

quence similarity networks of proteins. For example, the k-means method [19] is well known as a non-hierarchical clustering method. However, it cannot be directly applied to edge-weighted graphs because it is difficult to define the center of a cluster and the distance between the center and any node in the graph.

We applied our procedure, SelectLaminar, to maximal components and to some linkage methods. However, there may exist better selection methods and/or evaluation methods to compare the results of hierarchical clustering. Study of such methods is left as future work.

One of the most important features of our proposed method is that each cluster has a clear mathematical meaning that each cluster has maximal edge connectivity. Of course, many other clustering methods try to guarantee some mathematical properties [3),6)]. However, in most of these methods, clusters are obtained using recursive procedures and thus the meanings of clusters are unclear if more than 2 clusters are obtained. Of course, having mathematical meanings may not be important from a biological viewpoint. However, the most important contribution of this paper is that we demonstrated that a graph theory based algorithm performed better for protein sequence data than the standard linkage-based clustering methods.

There are several future works. We used log of E-values as edge-weights. However, this weighting method is not necessarily the best. Thus, finding better weighting method is important future work. We developed a simple method in order to select disjoint clusters from a set of maximal components. However, better results may be obtained by using a more elaborated method. Thus, improvement of selection of disjoint clusters should be done. Besides, we did not make much use of hierarchical structure of maximal components. More active use of hierarchical structure should also be examined. We have applied the proposed clustering method to clustering of protein sequences. However, our method is not limited to analysis of protein sequences. For example, clustering of gene expression data is one of extensively studied problems. Therefore, application to analysis of gene expression data is also important future work.

Acknowledgments We would like to thank reviewers for valuable comments. This work is supported in part by Grants-in-Aid #1630092 and "Systems Genomics" from the Ministry of Education, Science, Sports, and Culture of Japan.

References

1) Enright, A.J. and Ouzounis, C.A.: GeneRAGE: A robust algorithm for sequence clustering and domain detection, *Bioinformatics*, Vol.16, pp.451–457 (2004).
2) Koonin, E.V., Tatusov, R.L. and Rudd, K.E.: Sequence similarity analysis of Escherichia coli proteins: Functional and evolutionary implications, *Proc. Natl. Acad. Sci. USA*, Vol.92, pp.11921–11925 (1995).
3) Kawaji, H., Takenaka, Y. and Matsuda, H.: Graph-based clustering for finding distant relationships in a large set of proteins, *Bioinformatics*, Vol.20, pp.243–252 (2004).
4) Yona, G., Linial, N. and Linial, M.: ProtoMap: automatic classification of protein sequences, a hierarchy of protein families, and local maps of the protein space, *PROTEINS: Structure, Function, and Genetics*, Vol.37, pp.360–378 (1999).
5) Amin, M.A., Shinbo, Y., Mihara, K., Kurokawa, K. and Kanaya, S.: Development and implementation of an algorithm for detection of protein complexes in large interaction networks, *BMC Bioinformatics*, Vol.207, No.7, pp.532–538 (2006).
6) Newman, M.E.J.: Fast algorithm for detecting community structure in networks, *Phys. Rev.*, Vol.69, No.066133 (2004).
7) Tuji, H., Altaf-Ul-Amin, MD., Arita, M., Nishio, H., Shinbo, Y., Kurokawa, K. and Kanaya, S.: Comparison of protein complexes predicted from PPI networks by DPClus and Newman clustering algorithms, *IPSJ Transactions on Bioinformatics*, Vol.47, No.SIG 17(TBIO 1), pp.31–41 (2006).
8) Gomory, R.E. and Hu, T.C.: Multi-terminal network flows, *SIAM Journal of Applied Mathematics*, Vol.9, pp.551–570 (1961).
9) Nagamochi, H.: Graph algorithms for network connectivity problems, *Journal of the Operating Research Society of Japan*, Vol.47, pp.199–223 (2004).
10) Kleinberg, J.M.: Efficient algorithms for protein sequence design and the analysis of certain evolutionary fitness landscapes, *Proc. 3rd Int. Conf. Computational Molecular Biology* (*RECOMB*), pp.226–237 (1999).
11) Krivov, S.V. and Karplus, M.: Hidden complexity of free energy surfaces for peptide (protein) folding, *Proc. Natl. Acad. Sci. USA*, Vol.101, pp.14766–14770 (2004).
12) Zaki, M.J., Nadimpally, V., Bardhan, D.

and Bystroff, C.: Predicting protein folding pathways, *Bioinformatics*, Vol.20, pp.i386–i393 (2004).
13) Hong, E.L., Balakrishnan, R., Christie, K.R., Costanzo, M.C., Dwight, S.S., Engel, S.R., Fisk, D.G., Hirschman, J.E., Livestone, M.S., Nash, R., Park, J., Oughtred, R., Skrzypek, M., Starr, B., Theesfeld, C.L., Andrada, R., Binkley, G., Dong, Q., Lane, C., Hitz, B., Miyasato, S., Schroeder, M., Sethuraman, A., Weng, S., Dolinski, K., Botstein, D. and Cherry, J.M.: Saccharomyces Genome Database. ftp://ftp.yeastgenome.org/yeast/
14) Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J.: Gapped BLAST and PSI-BLAST: A new generation of protein database search programs, *Nucleic Acids Res.*, Vol.25, pp.3389–3402 (1997).
15) http://www.avglab.com/andrew/soft.html
16) Goldberg, A. and Tarjan, R.: A new approach to the maximum flow problem, *Journal of the Association for Computing Machinery*, Vol.35, pp.921–940 (1988).
17) Boyle, E.I., Weng, S., Gllub, J., Jin, H., Botstein, D., Cherry, J.M. and Sherlock, G.: GO::TermFinder–open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes, *Bioinformatics*, Vol.20, pp.3710–3715 (2004).
18) http://search.cpan.org/~sherlock/GO-TermFinder-0.7/lib/GO/TermFinder.pm
19) MacQueen, J.B.: Some methods for classification and analysis of multivariate observations, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol.1, pp.281–297 (1967).

(Received August 23, 2007)
(Accepted September 18, 2007)

(Communicated by *Tetsuo Shibuya*)

Morihiro Hayashida received his M.Sci. degree in Information Science from the University of Tokyo in 2002 and his Ph.D. degree in Informatics from Kyoto University in 2005. He is currently an assistant professor in Bioinformatics Center, Institute for Chemical Research, Kyoto University. His current research interests include issues related to protein function prediction and bioinformatics.

Tatsuya Akutsu received his M.Eng degree in Aeronautics in 1996 and a Dr. Eng. degree in Information Engineering in 1989 both from University of Tokyo. From 1989 to 1994, he was with Mechanical Engineering Laboratory. He was an associate professor in Gunma University from 1994 to 1996 and in Human Genome Center, the University of Tokyo from 1996 to 2001 respectively. He joined Bioinformatics Center, Institute for Chemical Research, Kyoto University as a professor in Oct. 2001. His research interests include bioinformatics and discrete algorithms.

Hiroshi Nagamochi received the B.A., M.E. and D.E. degrees from Kyoto University, in 1983, in 1985 and in 1988, respectively. He is a Professor in the Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University. His research interests include network flow problems and graph connectivity problems.