# Clustering Molecular Sequences with Their Components

**Sivasundaram Suharnan** [1]
suharu-s@is.aist-nara.ac.jp

**Takeshi Itoh** [2]
taitoh@lab.nig.ac.jp

**Hideo Matsuda** [3]
matsuda@ics.es.osaka-u.ac.jp

**Hirotada Mori** [4]
hmori@gtc.aist-nara.ac.jp

[1] Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-01, Japan

[2] Center for Information Biology
National Institute of Genetics
1111 Yata, Mishima, Shizuoka 411, Japan

[3] Department of Informatics and Mathematical Science
Graduate School of Engineering Science Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560, Japan

[4] Research and Education Center for Genetic Information
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-01, Japan

**Abstract**

**Motivation:** *Several methods in genetic information have recently been developed to estimate classification of protein sequences through their sequence similarity. These methods are essential for understanding the function of predicted open reading frames (ORFs) and their molecular evolutionary processes. However, since many protein sequences consist of a number of independently evolved structural units (we refer to these units as* components*), the combinatorial nature of the components makes it difficult to classify the sequences.*
**Results:** *This paper presents a new method for classifying uncharacterized protein sequences. As the measure of sequence similarity, we use similarity score computed by a method based on the Smith-Waterman local alignment algorithm. Here we introduce how this method cope when sequences have multi-component structure. This method was applied to predicted ORFs on the Escherichia coli genome and we discuss the algorithm and experimental results.*
**Keywords** *sequence classification, gene component, genome analysis.*

## 1 Introduction

Recent advances in computer technology have led to an exponential growth of DNA sequence analysis in genetic information. This technology has permitted the development of new statistical methods for understanding the unknown function of open reading frames (ORFs). Many proteins consist of a number of independently evolved structural units (so called modules [1] or domains). The functional diversity of protein sequences is partly due to the vast number of possibility to arrange a finite number of these basic units in different combinations. In this paper we refer to these units as *components* since we regard them as the elements to express the function of proteins. The combinatorial nature of proteins is major difficulty when one attempts to classify protein sequences on the basis of sequence similarity.

The detection of similarities between independent sequences is often the first step in the identification of relevant features in newly determined DNA sequences or their translated amino acid sequences. Many different programs have been developed to search for similarities between biological sequences. Generally, methods for classifying protein sequences are basically categorized into two groups.

1. **Similarity-based classification** : clustering sequences with their pairwise similarities. This is widely used for numerical data analysis and sufficient to analyze similarities between sequences that can be aligned over their entire length. However, if the sequences have combinatorial arrangements of several components such as two component system proteins [2], this approach may only detect their regional similarities rather than the similarities of overall lengths in the sequences. In this case, the classification based on similarity scores cannot determine whether a set of sequences share a common component or they have overlapped sets of different components (e.g., for three sequences $P$, $Q$ and $R$ and two components $a$ and $b$, $P$ has $a$, $Q$ has $b$ and $R$ has both $a$ and $b$). In the latter case, the method may classify some sequences that share no common components (e.g., $P$ and $Q$) into the same group through multi-component proteins that have two or more independent components (e.g., $R$).

2. **Pattern-based classification** : detecting commonly shared patterns. This approach piles up the regional similarities among protein sequences into statistically significant character patterns and then classifies the sequences with their patterns into groups [3, 4]. Thus the method can classify multi-component proteins into two or more different groups simultaneously (e.g., $\{P, R\}$ and $\{Q, R\}$ in the previous example). However, since the method explores frequently-occurred fixed-length patterns (so called *blocks*), it may not detect some components that only appeared in a small number of sequence groups. Also since the method does not allow gaps in the blocks usually, it may not detect weakly conserved components in distantly related proteins.

Here we introduce a method that combines these two approaches. Our method performs local alignment for every pair of sequences and extracts a set of regions from the alignment results, which are candidates of the components. Then the method carries out local alignment again between each of the component candidates and each of the sequences for screening spurious component candidates and analyzes the correspondence of the components to the sequences.

Basically our method is regarded as a pattern-based method but it utilizes similarity information generated by pairwise local alignment to obtain component candidates. Also, since the local alignment in our method is based on a rigorous dynamic programming method (the Smith-Waterman algorithm [5]), it allows gaps (i.e., insertions and deletions) in components. This is useful for detecting components from distantly related proteins.

# 2 Algorithm for detecting components

## 2.1 Definitions

The followings are some definitions to formulate the sequence classification problem.

**Definition 1 (Protein sequence)** *A* protein sequence $P = p_1 p_2 \cdots p_n$ *is a sequence such that each* $p_i$ $(1 \leq i \leq n)$ *is a character over an amino acid alphabet* $\sum = \{$A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y$\}$.

In the following definitions, $P = p_1 p_2 \cdots p_n$ and $Q = q_1 q_2 \cdots q_m$ denote two protein sequences of length $n$ and $m$, respectively.

**Definition 2 (Alignment)** *An* alignment *of* $P$ *and* $Q$ *is produced when null elements '–' are inserted into* $P$ *and* $Q$ *so that the resulting sequences* $P^*$ *and* $Q^*$ *have the same length. Here the sub sequence of* $P^*$ *or* $Q^*$ *whose elements are not equal to '–' is the original sequence* $P$ *or* $Q$.

**Definition 3 (Global alignment and global similarity)** *The* global alignment *of $P$ and $Q$ is an alignment of which similarity* (global similarity) *is formulated as follows:*

$$G\_SIM(P,Q) = \max \sum_{i=1}^{L} \sigma(p_i^*, q_j^*),$$

*where $p_i^*$ and $q_i^*$ are the $i$-th element of the resulting sequences $P^*$ and $Q^*$, respectively, whose lengths are the same $L$ and $\sigma(p_i^*, q_j^*)$ denotes similarity on the alphabet $\Sigma \cup \{$ '$-$' $\}$ (amino acid alphabet with the null character).*

**Definition 4 (Local alignment and local similarity)** *The* local alignment *of $P$ and $Q$ is an alignment of which similarity* (local similarity) *is formulated as follows:*

$$L\_SIM(P,Q) = \max\{0, G\_SIM(P[i..j], Q[k..l]) : 1 \le i \le j \le n, 1 \le k \le l \le m\},$$

*where $P[i..j]$ and $Q[k..l]$ are subsequences of $P$ and $Q$, respectively.*

In this paper, we use a procedure for generating a specified number of local alignments for possible subsequences in the descending order of local similarities. The procedure is formulated as follows:

**Definition 5 (Local alignment procedure)** *Given two sequences, $P$ and $Q$, and a positive integer $N$, the procedure*

$$\text{L\_ALIGN}(P,\ Q,\ N\text{: } Sim(k),\ Idt(k),\ s_P(k),\ e_P(k),\ s_Q(k),\ e_Q(k))\ (1 \le k \le N)$$

*generates at most $N$ local alignments whose local similarity is $Sim(k)$ obtained from a subsequence of $P$ ( starting from $s_P(k)$ and ending at $e_P(k)$ ) and a subsequence of $Q$ ( starting from $s_Q(k)$ and ending at $e_Q(k)$ ). Here $Idt(k)$ is the ratio of identical amino acids in the $k$-th local alignment.*

## 2.2 Description of the component-based grouping algorithm

To determine the components based on grouping, we developed a new algorithm. Our method consists of the following three steps.

### 2.2.1 [Step 1.] all-versus-all comparison of the sequences by local alignment

In order to express homology and sequence information, all-versus-all local alignment is carried out (see the pseudocode described below). This step picks up information such as the similarity score, the amino acid identity and the length of the homologous segments.

(1-1) **foreach** $Seq(i)$ in given sequences,
(1-2) **foreach** $Seq(j)$ in given sequences,
(1-3) Compute local alignment between $Seq(i)$ and $Seq(j)$ and output segments whose length, score and amino acid identities are at least given threshold values; $SegLen$, $Sim$ and $Idt$, respectively.

Figure 1 explains that $Seq(i)$ has homologous segments against each of the other sequences and one understands how complicatedly those segments overlap each other.

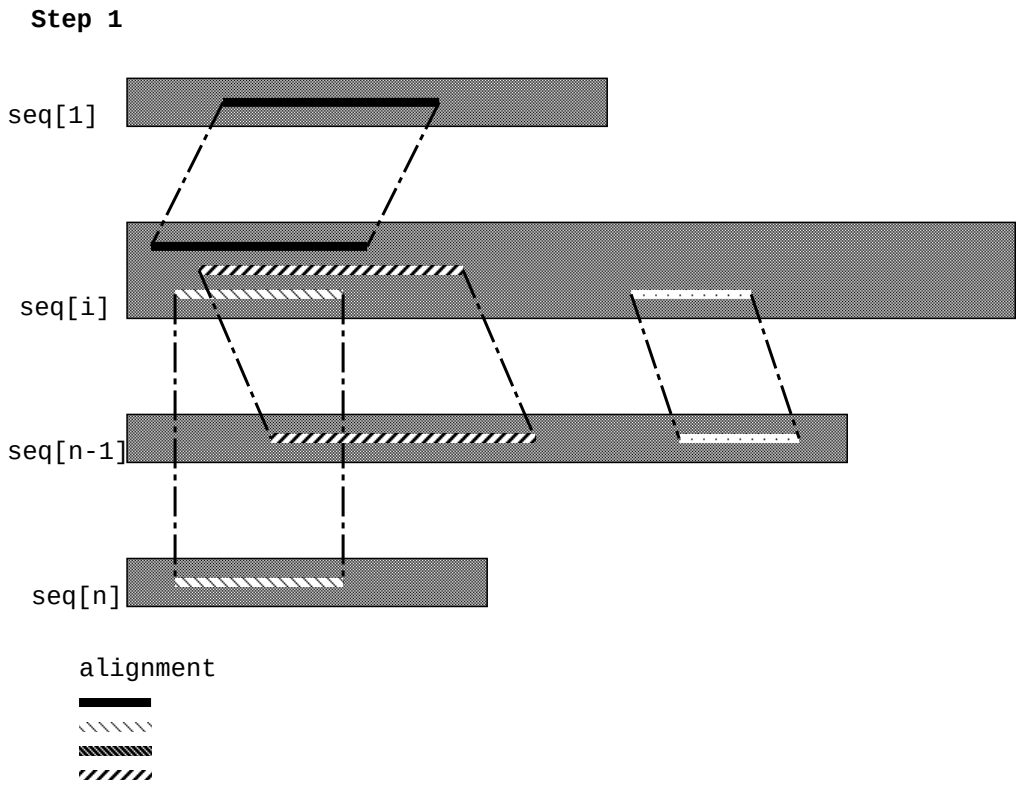


Figure 1: Detecting homologous segments between a sequence, $Seq(i)$, and each of the other sequences, $Seq(j)$ $(1 \leq j \leq n,\ i \neq j)$.

#### 2.2.2 [Step 2.] Construction of component candidates

In this step, overlapping alignment will be pruned down with the rest of the alignments starting positions and ending positions (see the pseudocode described below). If the length of an interval (i.e., a component candidate) is below the threshold cutoff value, that component candidate is removed. Here the cutoff value is a program parameter. This value forbids the generation of too small components and governs the resolution of the components. The generated components are used for the rest of the process. Figure 2 shows what happens to $Seq(i)$ before and after Step 2.

(2-1) Extract all intervals between either of any starting or ending positions of the segments that are output in Step 1.
(2-2) Pick up the intervals whose length is at least given threshold $ComLen$ and store the intervals as $Com(i, j)$ (the $j$-th component candidates in a sequence $Seq(i)$).

#### 2.2.3 [Step 3.] Analyzing the correspondence of component candidates to sequences

This step is composed of three substeps (see the pseudocode described below). Here grouping is carried out on the basis of component information. In Step 3.1 the similarity score of each component candidate and itself (hereafter, self-similarity score) is computed ((3-2) and (3-3)). After this process, local alignment is carried out again between components and the rest of the protein sequences.

**Step 3.1** (Computing self-similarity scores for all intervals)
(3-1) **foreach** $Com(i, j)$ in all component candidates that are output in Step 2,

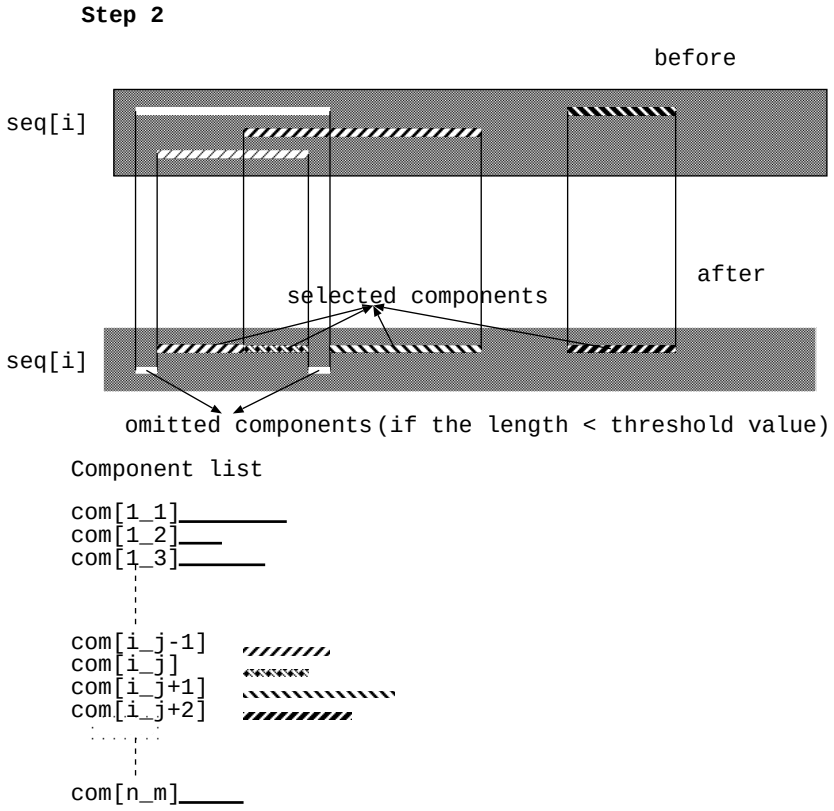


Figure 2: Construction of component candidates Com(i,j) (the j-th component in Seq(i))

(3-2) Extract a subsequence, $Subseq(i,j)$, of a sequence $Seq(i)$ corresponding to $Com(i,j)$.
(3-3) Compute similarity score between $Subseq(i,j)$ and itself and set it to be $SimSelf(i,j)$.

**Step 3.2** (Grouping relevant components)
(3-4) **foreach** $Com(i,j)$ in all component candidates that are output in Step 2,
(3-5) Extract a subsequence, $Subseq(i,j)$, of a sequence $Seq(i)$ corresponding to $Com(i,j)$.
(3-6) **foreach** $Seq(k)$ in given sequences,
(3-7) Compute local alignment between $Subseq(i,j)$ and $Seq(k)$ and output segments whose overlap length is nearly equal to the length of $Subseq(i,j)$ (the difference is within given threshold $LenDiff$), whose score is at least $SimSelf(i,j) * SimDiff$ (a given scale parameter) and whose amino acid identities is at least given threshold $IdtDiff$.
(3-8) Compute self-similarity scores for these segments the same as described in Step 3.1.
(3-9) Grouping these segments and $Com(i,j)$ into a group, $Group(i,k)$, and deduct the $Seq(i)$.

**Step 3.3** (Merging groups)
(3-10) **foreach** $Group(i,j)$ in every groups constructed in (3-9).
(3-11) **foreach** $Group(k,l)$ ($i \neq k$ or $j \neq l$) in every groups constructed in (3-9).
(3-12) **if** $Group(i,j)$ and $Group(k,l)$ have the same members **then** remove $Group(k,l)$.

Figure 3 schematically explains the detail of these steps for each component. First a component $Com(i,j)$ is selected and added to a list and then local alignment will be carried out between each

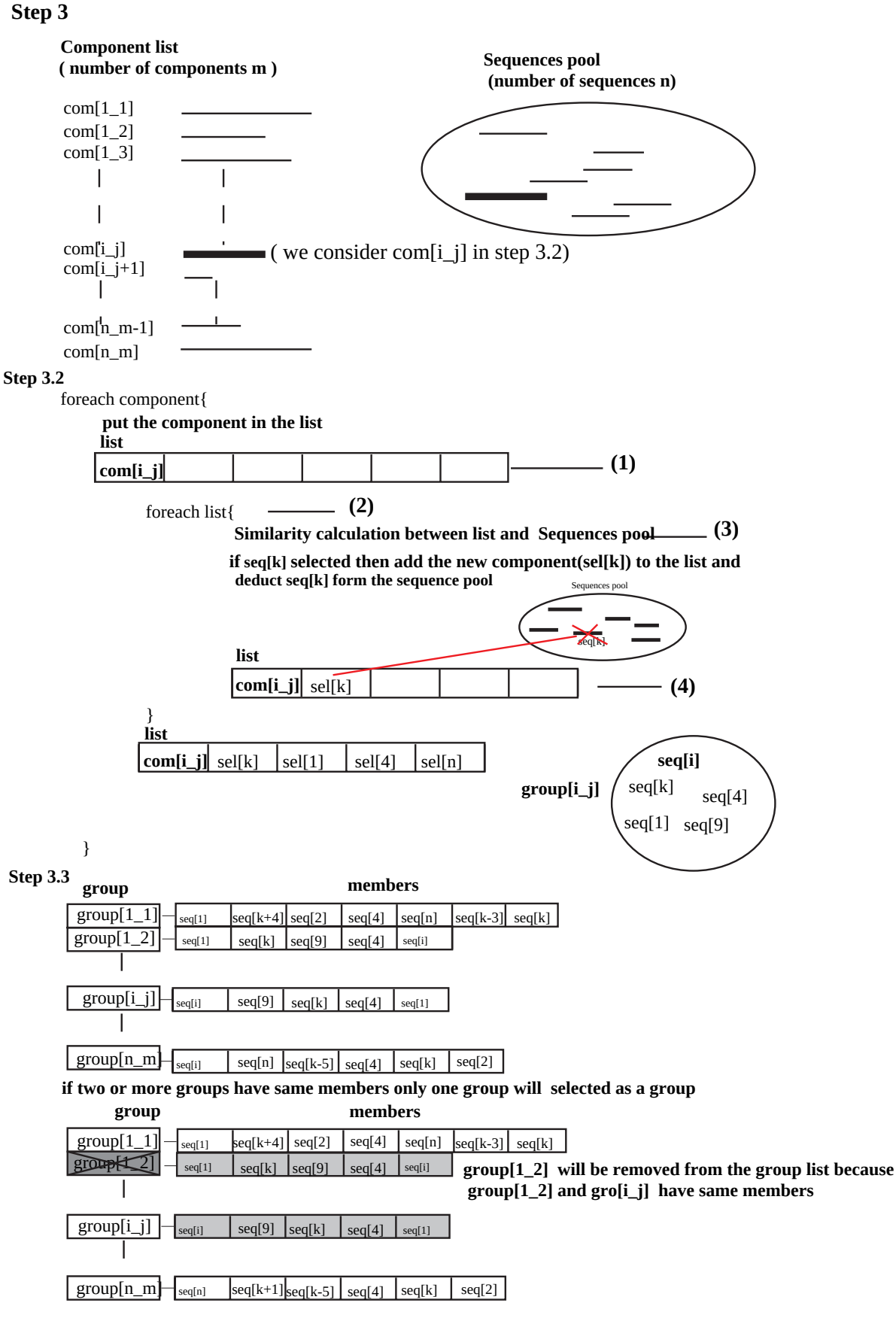


Figure 3: Detecting group member for each component

member of the list and sequences. If one of these sequences indicates the similarity measured by the ratio compared to the self-similarity score of the component, the new component is added to the list and the present sequence is deducted from non-grouped list. The self-similarity score is also computed for the new component. By this process, our algorithm allowing them to collect all possible sequences to become in the same group only if those sequences satisfy the threshold cutoffs. Finally, if two or more groups have the same members we regard them as one group in Step 3.3.

### 2.3 Improvement of the algorithm

Since our method utilizes local alignment based on the rigorous Smith-Waterman algorithm, we believe that the sensitivity of our method is high enough to detect components from distantly-related protein sequences. The local alignment method is, however, very time consuming process. Thus, it may take very long time to detect components from a large number of protein sequences by our method.

To cope with this issue, we introduce the single-linkage clustering method [6] as a pre-process step before analyzing components. By this step, given protein sequences are divided into a number of groups such that each sequence in a group has similarity, which is at least a given threshold, directly or transitively (via some sequence) to the other sequences in the same group (a transitive closure of sequences with pairwise similarity $\geq$ a given cutoff).

This step can be efficiently carried out by using fast similarity search method (e.g., FASTA [7]). Since the time complexity of our method is proportional to at least square of the number of sequences, reduction of the number of sequences by the single-linkage clustering method is effective for decreasing the computation time of our method.

## 3 Experiments and discussions

This application was tested on many amino acid sequences and applied to all of the ORFs deduced from the complete sequence of *E.coli* chromosome compiled by ourselves using the chromosomal sequences from the Japanese *E.coli* sequencing group [8, 9, 10, 11] and *E.coli* sequences in GenBank. We used a total of 4562 ORFs in this analysis. We classified all of them by the following method.

### 3.1 Pre-classification by using single-linkage clustering method

As described in Section 2.3, we carried out single-linkage clustering among the 4562 ORFs before analyzing components in the sequences. In this pre-classification, we used the FASTA program [7] for computing pairwise similarities and we set cutoff similarity score to be 120 in the FASTA opt score. Table 1 shows the result of this pre-classification.

### 3.2 Result of component-based grouping

For further classification we used an improved algorithm described in Section 2.3 and classified each single-linkage clusters into several subgroups based on the distribution of components among the sequences. In our component-based grouping we select the cutoff values described in Section 2.2 as $SegLen = 60$, $Sim = 100$, $ComLen = 10$, $LenDiff = 10$ and $SimDiff = 0.4$. A computer program LALIGN in the FASTA package [7] was used for assisting the local alignment analysis.

In our grouping process we used the same cutoff value for all single-linkage clusters. However if we use the suitable cutoff value for every group, we may get better result than what we got at present. Extracted view of an example of the single-linkage result in *E.coli* group is shown in Figure 4. From this figure one can find out a number of protein sequences appeared in different groups. This is because each member has more than one component and members belong to each component are different from others.

Table 1: Result of single-linkage classification with cutoff similarity score 120

| No. of clusters | No. of members |
|---|---|
| 1 | 435 |
| 1 | 358 |
| 1 | 57 |
| 1 | 52 |
| 1 | 42 |
| 1 | 29 |
| 1 | 27 |
| 1 | 23 |
| 1 | 20 |
| 1 | 16 |
| 4 | 17 |
| 2 | 15 |
| 2 | 14 |
| 7 | 11 |
| 4 | 10 |
| 9 | 8 |
| 9 | 6 |
| 11 | 9 |
| 11 | 7 |
| 17 | 5 |
| 52 | 4 |
| 87 | 3 |
| 244 | 2 |
| 1910 | 1 |

Figure 5 shows the deduced major components of the DnaK homologues by our classification. A study based on the tertiary structure analysis of those proteins [14] reports that all the homologues have five motifs named **Phosphate 1, Connect 1, Phosphate 2, Adenosine** and **Connect 2**. However, previous computational methods for extracting components [3, 4] can detect only a part of the motifs in DnaK, HscA and MreB, and none of them in FtsA.

As shown in Figure 5, the layout of DnaK components is very similar to that of HscA components and both have the five motifs in their components. However, MreB has three motifs (Phosphate 1, Connect 1 and Phosphate 2) and FtsA has only one motif (Connect 1) in their components. We could not get the rest of the motifs for MreB and FtsA because the similarities among the regions including these motifs are very weak compared to those among the other protein sequences. However we succeeded in putting all the four homologues into the same group.

# 4 Conclusion

Several studies are reported for a systematic classification of protein sequences. In one study [12, 13], they reported signatures of protein sequences that are indeed highly informative. However because they derive from non-gapped sequence alignment, these signatures may not correspond to entire component. Another study [15] utilizes consensus matrix with gap to find out conserved regions among sequences. However, the method is motivated for detecting regions conserved among as many sequences as possible to carry out multiple alignment of those sequences. While our algorithm is designed for detecting as many components as possible among two or more sequences.

In other studies [3, 4], they designed domain based clustering algorithms. However they did not attempt to infer the gaps and the component size is fixed all the time. On the other hand, our

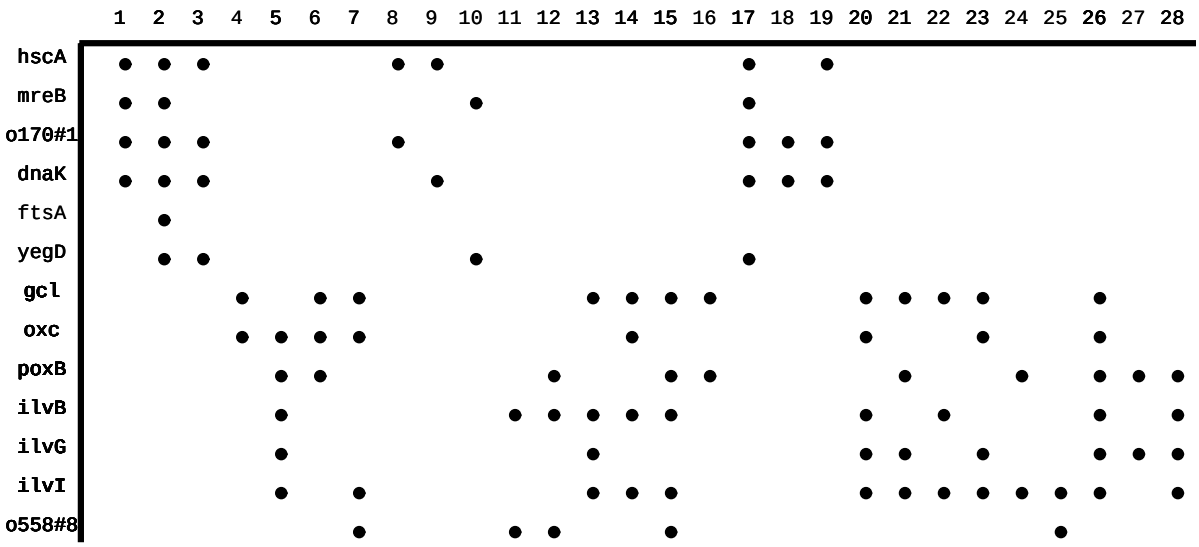


Figure 4: Example of extracted components from a set of sequences classified by the single-linkage clustering method.

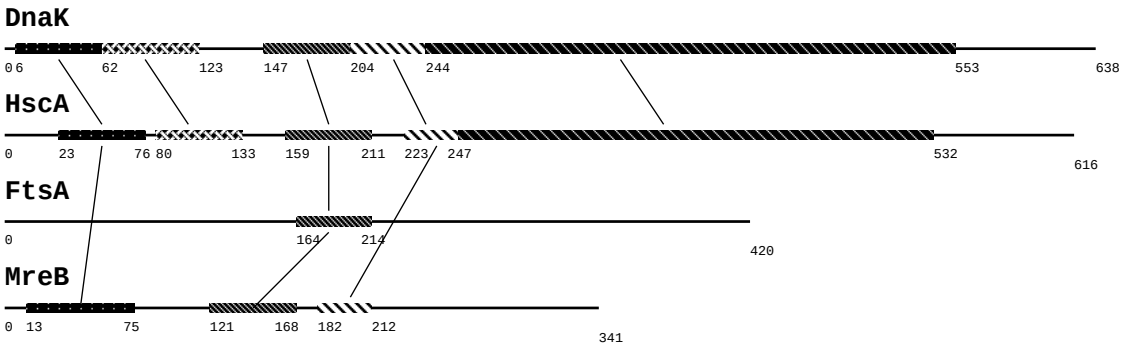


Figure 5: DnaK homologues and their components.

algorithm includes such information and already operational in our laboratories for grouping and functional analysis.

In the latter studies [3, 4], both of their methods classifies the proteins DnaK, HscA, MreB and FtsA that are reported to have the identical ATPase domain [14] into two separate groups (FtsA and the others), whereas our method successfully classified them into the same group through the local alignment that allows gaps.

However the method to determine suitable threshold cutoff values for detecting components among distantly related proteins remains as one of our future works. Also the number of component seems to be too high. This is because we tried to find out all the possible conserved regions. More sophisticated method may be required for only detecting components whose number is sufficient to do functional assignment of uncharacterized protein sequences.

## Acknowledgments

This work was supported in part by a Grant-in-Aid (08283103) for Scientific Research on Priority Areas from The Ministry of Education, Science, Sports and Culture in Japan. The authors would like to thank Jun-ichi Takeda and Michiko Muraki (Nara Institute of Science and Technology) for their valuable comments.

## References

[1] M. Baron, D. G. Norman, L. D. Campbell, Protein modules, *Trends Biochem. Sci.*, (1991) **16** 13–17.

[2] T. Mizuno, Compilation of all genes encoding two-component phosphotransfer signal transducers on the genome of *Escherichia coli* , *DNA Research*, **4** (1997) 161–168.

[3] E. L. L. Sonnhammer and D. Kahn, Modular arrangement of proteins as inferred from analysis of homology, *Protein Science*, **3** (1994) 482–492.

[4] R. L. Tatusov, S. F. Altschul and E. V. Koonin, Detection of conserved segments in proteins: iterative scanning of sequence databases with alignment blocks, *Proc. Natl. Acad. Sci. USA*, **91** (1994) 12091–12095.

[5] T. F. Smith and M. F. Waterman, Identification of common molecular subsequences, *J. Mol. Biol.*, **147** (1981) 195–197.

[6] J. A. Hartigan, *Clustering Algorithms*, John Wiley & Sons, Inc., New York (1975).

[7] W. R. Pearson and D. J. Lipman, Improved tools for biological sequence comparison, *Proc. Natl. Acad. Sci. USA*, **85** (1988) 2444–2448.

[8] T. Oshima, et al., A 718-kb DNA sequence of the *Escherichia coli* K-12 genome corresponding to the 12.7–28.0 min region on the linkage Map, *DNA Research*, **3** (1996) 137–155.

[9] H. Aiba, et al., A 570-kb DNA sequence of the *Escherichia coli* K-12 genome corresponding to the 28.0–40.1 min region on the linkage map, *DNA Research*, **3** (1996) 363–377.

[10] T. Itoh, et al., A 460-kb DNA sequence of the *Escherichia coli* K-12 genome corresponding to the 40.1–50.0 min region on the linkage Map, *DNA Research*, **3** (1996) 379–392.

[11] Y. Yamamoto, et al., Construction of a contiguous 874-kb sequence of the *Escherichia coli* K-12 genome corresponding to the 50.0–68.8 min on the linkage map and analysis of its sequence features, *DNA Research*, **4** (1997) 91–113.

[12] N. L. Harris, L. Hunter, D. J. States, Mega-classification: discovering motifs in massive datastreams, *Proc. National Conf. on Artificial Intelligence (AAAI)* (1992) 837–842.

[13] R. Sheridan, R. Venkataraghavan, A systematic search for protein signature sequences, *Proteins Struct. Funct. Genet.* (1992) 1433–1438.

[14] P. Bork, C. Sander and A. Valencia, An ATPase domain common to prokaryotic cell cycles proteins, sugar kinases, actin, and hsp70 heat shock proteins, *Proc. Natl. Acad. Sci. USA*, **89** (1992) 7290–7294.

[15] N. N. Alexandrov, Local multiple alignment by consensus matrix, *Comput. Appl. Biosci* **8** (1992) 339–345.