情報処理学会研究報告
IPSJ SIG Technical Report

Vol.2009-MPS-74 No.3
2009/7/13

Gibbs-DMGG による類似部分配列の抽出方式

河野修久[†]　北上始[†]　田村慶一[†]　森康真[†]

配列データマイニング処理では，配列データベースから非常に多くの頻出配列パターンが抽出される．著者らは，既に，配列データベースに対してギブスサンプリング(GS)を適用し，頻出配列パターンを削減する方法を提案している．しかしながら，この方法では，予め，抽出する部分文字列の長さをユーザ側で指定する必要があるほか，必ずしも精度(再現度)が良いとは限らないという問題がある．本稿では，これら 2 つの問題を解決するために，遺伝的アルゴリズムの世代交代モデルの 1 つである Minimal Generation Gap (MGG)と，分散遺伝的アルゴリズム(島モデル)の考え方を GS に応用した新しい類似部分配列抽出法 Gibbs-DMGG を提案する．また，この提案手法が従来手法よりの有効であるかを確認するために，両者の性能評価・考察を行ったので，その結果についても報告する．

Extracting Similar Subsequences by Gibbs Sampling with Distributed MGG

NOBUHISA KONO[†]　HAJIME KITAKAMI[†]
KEIICHI TAMURA[†]　YASUMA MORI[†]

In the field of sequence data mining, a large quantity of frequent sequential patterns are extracted from a sequence database. In order to significantly reduce these frequent sequential patterns, we have already proposed a method of applying Gibbs sampling (GS) to the sequence database. However, this method involves problems such as the necessity of setting the length of the extracted substring beforehand, and the possibility of insufficient accuracy (Recall). In order to solve these problems, we propose in this paper a new, similar subsequence extraction method called Gibbs-DMGG. This method applies Minimal Generation Gap (MGG), a generation alternation model of the genetic algorithm, and a distributed population scheme called the Island model. Experiments were used to evaluate our proposed method.

1. Introduction

A method for extracting frequent sequential patterns from sequence databases is useful in many application domains. Among these uses are finding regularity in text databases and finding motif patterns in molecular subsequences. Motifs discovered by many biologists appear in PROSITE [1] and in Pfam [2][3] and are regarded as protein functions that have been conserved in the process of molecular evolution. There are two methods of representing a motif in the natural world: regular, and probabilistic expression. This is due to the fact that motifs include ambiguities such as a variable wildcard regions and approximate expressions.

A sequential pattern mining method for molecular sequences was developed to extract variable wildcard regions included in the regular expression of frequent sequential patterns.[4][5] However, because the method extracts a large quantity of junk patterns, we have focused on reducing the number of frequent sequential patterns extracted by the method.

The Gibbs sampling method, [6] called GS, is a key technology used to solve problems of frequent sequential pattern reduction. GS has the capability of extracting similar subsequences from sequence databases. Therefore, we have proposed a reduction of input data for sequential pattern mining using GS. However, two problems arise from using GS. The first problem is that the length of the subsequence extracted from the sequence databases must be specified in advance. The second problem is that the accuracy of extracting similar subsequences is not always stable.

In this paper, we propose a novel method called Gibbs-DMGG (**Gibbs** sampling with **D**istributed **MGG**, where MGG is a GA with Minimal Generation Gap). [7][8] This method not only has the capacity to automatically determine the length of a similar subsequence but also provides stable accuracy in similar subsequence extraction.

Gibbs-DMGG solves an optimization problem for GS using a genetic algorithm with the Minimal Generation Gap model (MGG) [7][8] and a distributed population scheme called the Island model GA.[9][10] MGG is a generation alternation model capable of avoiding early convergence and suppressing evolutionary stagnation through the dynamics of the best value and the variance of fitness distributions. However, because MGG is not capable of providing stable accuracy, we use MGG with a distributed population scheme.

The remainder of the present paper is organized as follows: A discussion of related research is presented in Section 2. The Gibbs sampling algorithm is described in Section 3. The proposed method, Gibbs-DMGG, is described in Section 4. The performance of the proposed method is evaluated in Section 5, and the results of the present study are summarized in

[†] 広島市立大学情報科学研究科
Graduate School of Information Science, Hiroshima City University

Section 6.

2. Related Work

Since the sequential pattern mining method for molecular sequences extracts a large quantity of junk patterns,[4][5] we have focused on the Gibbs sampling method of reducing the number of frequent sequential patterns extracted.[6] The Gibbs sampling method is a key technology for solving problems of reducing the number of frequent sequential patterns.

In 1993, Lawrence et al. proposed the use of GS to extract similar subsequences from sequence databases.[6][11] This earlier GS method is not capable, however, of extracting multiple motifs from any sequence in the sequence database. Liu et al. proposed Greedy Two-stage Gibbs Sampling, a method capable of extracting multiple motifs from each sequence.[12] Moreover, stable accuracy to solve the optimization problem for GS was enhanced by a simulated tempering [13] and a generation alternation model called Minimal Generation Gap (MGG).[7][8] However, two problems with the existing methods remain. One is that the lengths of similar subsequences extracted by each method have to be given by the user even though they may be unknown. Another problem is that existing methods do not always have stable accuracy and there is no obvious analogy for temperature T with respect to free parameters in the simulated annealing method.

The proposed method, Gibbs-DMGG, provides a solution to these problems. Gibbs-DMGG has not only the capacity to determine the lengths of the similar subsequences automatically but also stable accuracy necessary to extract similar subsequences. This capability is achieved through applying the island model GA to provide stable accuracy and improving Gibbs-MGG so that it can automatically determine the length of similar subsequences.

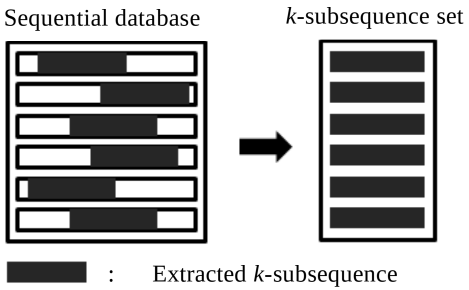

Figure 1 Sequential database and *k*-subsequence set

3. Existing Gibbs Sampling

This section describes a method for extracting similar subsequences using existing Gibbs sampling GS and a method for evaluating similar subsequences (known as *k*-subsequences) extracted by GS. A *k*-subsequence is a substring that is extracted from each sequence in the sequence database, wherein the user provides the value for the length *k*. Additionally, the set of *k*-subsequences extracted from each sequence in the database is called the *k*-subsequence set. The relationship between the sequence database and the *k*-subsequence set is shown in Figure 1.

3.1 Gibbs sampling

Consider that each sequence of the sequential database *DB* is defined alphbetically $\sum = \{a_1, a_2, \ldots, a_n\}$, where *DB* consists of *t* sequences. Gibbs sampling GS is the method used to find one *k*-subsequence as similar as possible to each sequence of the sequential database *DB*, where the user has to specify beforehand the length *k* of the subsequence extracted by GS. In order to find the most similar *k*-subsequences, GS requires a measure to evaluate *k*-subsequences extracted as candidate solutions. To compute the measure, GS includes three quantities related to statistical probability: a score matrix, frequency, and background frequency.

The three quantities related to statistical probability in a *k*-subsequence set are defined as follows:

(1) Score matrix

The score matrix $A = (A_{i,j})$ of the *k*-subsequence set is called a profile, and the matrix element $A_{i,j}$ is the frequency of letter a_j appearing in position i for the *k*-subsequence set.

(2) Frequency

Frequency A_x of *k*-subsequence $x = <a_1 a_2 \ldots a_k>$ is defined as the calculation $A_{1,1} \times A_{2,2} \times \ldots \times A_{k,k}$, which means that the frequency of the *k*-subsequence x has a higher probability if it is similar to the consensus of *k*-subsequences used in computing the score matrix. On the other hand, the *k*-subsequence x has a lower probability if the frequency of x is dissimilar to the consensus.

(3) Background frequency

The background frequency of letter a_j is computed by dividing the total number Pa_j of letter a_j into the total number of letters in *BS*, where *BS* is defined as a set of subsequences that are collected by removing all candidate solutions from *DB*. The background frequency P_x of *k*-subsequence $x = <a_1 a_2 \ldots a_k>$ is defined as the calculation $Pa_1 \times Pa_2 \times \ldots \times Pa_k$.

1. Randomly choose a starting position $ST = (st_1, st_2, \ldots, st_t)$ in t sequences and extract the set $S = \{s_1, s_2, \ldots, s_t\}$ of k-subsequences using the starting positions ST from t sequences.
2. Randomly select one sequence Z from *DB* and compute $DB' = DB - \{Z\}$, where $\|DB'\| = t-1$. Compute $S' = S - \{Z'\}$, where Z' is the k-subsequence extracted from sequence Z based on the starting point *st* selected in step 1 (above).
3. Calculate the score matrix $A = (A_{i,j})$ from the k-subsequence set S' consisting of t-1 elements.
4. Calculate the background frequency Pa_j of each letter a_j using set BS', where BS' is collected by removing S' from DB'.
5. Calculate the evaluation value $U_x = A_x \div P_x$ for each of $\|Z\|-k+1$ k-subsequences x with the starting position i in the sequence Z, where $1 \le i \le \|Z\|-k+1$.
6. Randomly choose the starting position st'_m, according to the distribution proportional to the evaluation value set that is $\{U_x \mid x$ belongs to the set of $\|Z\|-k+1$ k-subsequences with the starting position i in sequence $Z\}$. Update S, where $1 \le m \le \|Z\|-k+1$.
7. Repeat steps 2~6 a preset number of times.

Figure 2 Gibbs sampling algorithm

Let $ST = (st_1, st_2, \ldots, st_t)$ be the starting position of the chosen k-subsequence in *DB* stored with t sequences. Gibbs sampling GS, shown in Figure 2, computes the starting positions *ST* for every iteration processing. The processing is carried out to find the k-subsequence with the starting position from one sequence Z randomly chosen from among the t sequences. The k-subsequence extracted from the starting position is characterized as the k-subsequence with higher frequency but lower background frequency.

3.2 Evaluation of subsequence

In order to appraise similar k-subsequences which are extracted from the sequential database *DB*, relative entropy, called the F value, is used. The relative entropy is the difference in the distribution of the places of similar k-subsequences extracted from *DB* and the dissimilar parts that remain without being extracted. Therefore, the F value is defined as follows:

$$F = \sum_{i=1}^{k} \sum_{j=1}^{20} C_{i,j} \log \frac{Q_{i,j}}{P_j} \tag{1}$$

P_j is the same as the background frequency P_j of the letter a_j. $Q_{i,j}$ is defined as follows:

$$Q_{i,j} = \frac{C_{i,j} + b_j}{N - 1 + B} \tag{2}$$

$C_{i,j}$ is a score matrix element to the portion of the similar k-subsequence extracted from *DB* and represents the number of the letter j which exists in a position i. The pseudo-count b_j is determined by $f_i \times B$. Pseudo-counts are used to avoid a status in which $C_{i,j} = 0$ becomes $Q_{i,j} = 0$. f_i is determined by the relative frequency of the letter a_j in *DB*. Moreover, B will be experimentally set to $\sqrt{N}$, where N is the number of sequences in *DB*.

4. New Method

In this section, we propose a novel method called Gibbs-DMGG. This method not only is capable of automatically determining the length k of the similar subsequence but also provides stable accuracy for similar subsequence extraction. First, we introduce the optimization methods MGG and the Island model GA, and we describe the Gibbs sampling method using MGG (Gibbs-MGG). In order to automatically determine the length k of the subsequence, a method for improving Gibbs-MGG is proposed. Hereafter, this improved Gibbs-MGG is called Modified Gibbs-MGG. Finally, to secure stable extraction accuracy, we propose using Gibbs-DMGG by combining the Modified Gibbs-MGG with the Island model GA.

4.1 Optimization method

In this section, both MGG and the Island model GA, used for optimal Gibbs sampling, are described.

4.1.1 MGG

MGG, shown in Figure 3, is a generation alternation model of a GA which generates multiple individuals in a population and repeats the process of selection for reproduction, crossover, mutation, and selection for survival. The selection for reproduction denotes the selection of a pair of individuals by random sampling without replacement from the population. MGG is capable of avoiding premature convergence; since the selected pair of individuals in a population is randomly chosen as an object of alternation of generation, MGG yields solutions with low precision. Moreover, MGG suppresses evolutionary stagnation by preserving a specific variance of fitness distributions. Through using elite selection and roulette-wheel selection for survival, specific variance is preserved.

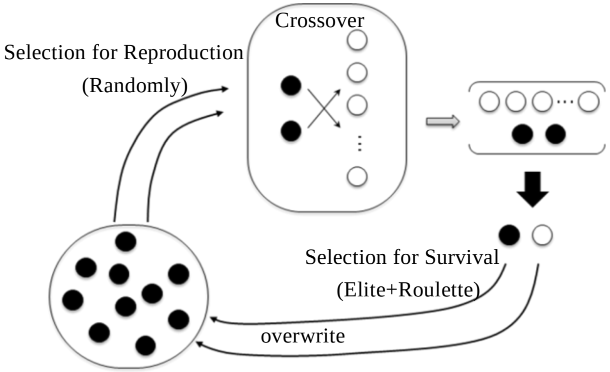

Figure 3　Diagram of processing in MGG

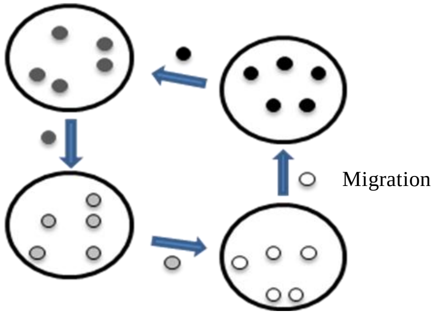

Figure 4　Concept of the Island model GA

4.1.2 Island model GA

The Island model is a distributed model GA. In this model, subpopulations evolve separately, and their individuals migrate among these subpopulations in certain generations. The Island model has parameters associated with migration: the migration interval and the migration rate. The migration interval is the number of generations between each migration, and the migration rate is the number of individuals selected for migration. The concept of the Island model is shown in Figure 4.

In the Island model, the search advances to independence with the respective island. For the individual, each island differs significantly. Therefore, since the diversity is greater than that of a single population, a stable improvement in accuracy is expected.

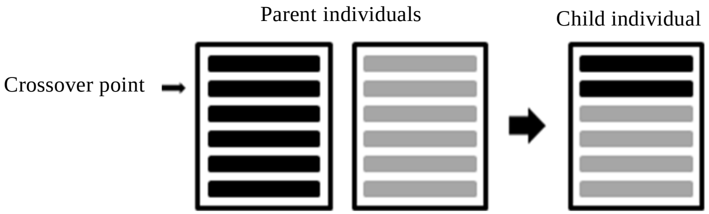

Figure 5　Method for generating one child individual using crossover operation

1. Randomly generate *N* individuals as a population of the initial condition.
2. Selection for reproduction: randomly choose two individuals in the population.
3. Crossover: recombine the two individuals selected by reproduction and generate *M* child individuals.
4. Selection for survival: calculate the *F* value of parent individuals and *M* child individuals. Return the individual whose fitness is highest and the individual chosen by roulette selection.
5. Mutation: randomly choose *L* individuals in the population. Mutation is complete.
6. Repeat steps 2~5 for *m* iterations, where *m* is given by the user.

Figure 6　The Gibbs-MGG algorithm

4.2 Gibbs-MGG

Gibbs-MGG is proposed to improve the extraction accuracy of GS. The individual that is used with MGG is defined as a *k*-subsequence set. The *F* value of Equation (1) is used to compute the fitness of the individual. The processing of MGG is carried out by crossover, selection, and mutation operations for the individually prepared *N* units.

In the crossover operation, the crossover point is randomly decided for each of *M* generated individuals, where *M* is given by the user beforehand. Based on *M* crossover points, *M* child individuals are generated from two individuals chosen by selection for reproduction. If two individuals are the same, the selection for reproduction is repeated until two distinct individuals are selected. Figure 5 shows an example of one child individual made from two individuals.

In the selection operation for survival, we return two new individuals to the population, where one is chosen by highest fitness (elite) selection and the other is chosen by roulette selection. They are chosen from two older individuals obtained by selection for reproduction and *M* child individuals generated by the crossover operation.

In the mutation operation, *L* individuals are selected from the population at random, and the iteration processing in the GS is done only once, where *L* is given by the user beforehand. In other words, steps 2~6 of the Gibbs sampling algorithm shown in Figure 2 are repeatedly executed for each *L* individual. Figure 6 shows the Gibbs-MGG algorithm.

4.3 Proposed method

In the Gibbs sampling algorithm, the length *k* of the subsequence extracted from sequences has to be given by the user beforehand. In order to decide the length *k* automatically, we need to modify the Gibbs-MGG algorithm described in the previous section. Furthermore, we propose a new method constructed by combining the Modified Gibbs-MGG with the Island model GA.

In the Gibbs-MGG algorithm, subsequences of any individual in the population have the same length given by the user, and the length does not change during processing. In contrast to the Gibbs-MGG algorithm, the Modified Gibbs-MGG algorithm is flexible in that subsequences allow for different lengths in the population since an optimal length for the subsequence must be found. Therefore, we modify the crossover and mutation operations for flexibility.

In the existing crossover operation, a problem arises in that subsequences in the child individual (as shown in Figure 5) include two different lengths if two individuals chosen from the population have two different lengths on the subsequence. Unification of the two different lengths is proposed as a solution to the problem.

In the modified crossover operation, unification is a method for randomly selecting either of two different lengths. The method is executed before the existing crossover operation.

In the modified mutation operation, we propose randomly selecting a subsequence from subsequences in the individual and applying it to the change operation for the length of the subsequence. Figure 7 shows the change operation. Drawing a conclusion from the experiment, we avoid a drastic change in the length of the subsequence in the Gibbs sampling algorithm. Therefore, we randomly change the start and end positions of the subsequence within the range from -*k*/4 to *k*/4, where *k* denotes the length of the original subsequence before the change operation is executed. Therefore, the length of the subsequence after the change operation is the range from *k*/2 to 3*k*/2.

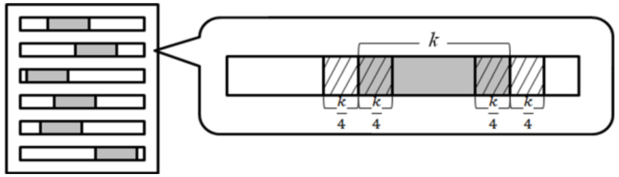

Figure 7 The change operation for the length of the subsequence

```
1    Randomly generate I islands (populations) containing N
     individuals as population groups for the initial condition.
2    Perform the following work in each population:
  2.1    Selection for reproduction: randomly choose two
         individuals per population.
  2.2    Crossover: recombine two individuals selected randomly
         on the condition that the length of each subsequence is
         selected as either of two and M child individuals are
         generated.
  2.3    Selection for survival: calculate the F value of parent
         individuals and M child individuals and return the
         individual whose fitness is highest and the individual
         chosen by roulette selection.
  2.4    Migration: randomly choose L individuals in the
         population. Mutation is complete.
  2.5    Modification of sequential length: change the
         subsequence length once per several generations.
3    Migration: migration should occur once per several
     generations.
4    Repeat steps 2~5 a preset number of times.
```

Figure 8 The Gibbs-DMGG algorithm

As shown in Figure 7, the start and end positions of the subsequence are determined inside the respective shaded areas. The change operation for the length of the subsequence is not necessarily made by the mutation operation each time. It refers to the number of sequences in the sequence database for each of several generations.

When the subsequence length is determined automatically, the initial value of the length becomes very important. One problem is that, if the initial length is too long, Gibbs-MGG is incapable of finding a length shorter than the initial length of the subsequences. On the other hand, if the initial length is too short, Gibbs-MGG is incapable of finding a highly accurate length.

In order to solve this problem, the Island model is used in conjunction with Modified Gibbs-MGG. In the novel Gibbs-DMGG method, to prevent aggravation of the extraction accuracy, we prepare multiple populations (as shown in Figure 8) by applying an initial value of the subsequence length to a different item in each population. For each population, processing is usually done independently. However, individual movement in each population, called migration, is performed once per several generations.

5. Performance Evaluation

To confirm the effectiveness of the new similar subsequence extraction method, an evaluation experiment was conducted using the *Leucine Zipper* dataset (registration number PS00036, from PROSITE). The computer environment used for the evaluation was a 2.66-GHz Intel® Core™2 Quad with 2 GB of memory, 2 GB of SWAP memory, a 227 GB HDD, and Fedora 9 as the operating system. The characteristics of the *Leucine Zipper* dataset in PROSITE are shown in Table 1. The motif appearing in the dataset is formed with a maximum of 16 letters and is represented as follows: <[*KR*]-*x*(1,3)-[*RKSAQ*]-*N*-*x*(2)-[*SAQ*](2) -*x*-[*RKTAENQ*]-*x*-*R*-*x*-[*RK*]>. The symbol [*KR*] is an ambiguous character and denotes allowance of the selection of any character included in the set {*K*, *R*}. The symbol *x*(1,3) between two ambiguous characters, [*KR*] and [*RKSAQ*], denotes a range from one to three wildcards, where the wildcard is a special character that can be used to substitute for any other character. The symbol [*SAQ*](2) denotes [*SAQ*]-[*SAQ*].

5.1 Performance measure

In order to evaluate the performance of Gibbs-DMGG, we define both Recall and Precision as performance measures. Precision can be seen as a measure of exactness or fidelity, whereas Recall is a measure of completeness.

The number of sequences included in the sequence database is assumed to be n. A set of the motif domain which exists in the sequence database is represented as $A = \{A_1, A_2, \ldots, A_n\}$, where A_i is a motif region appearing in the sequence with the value i as the sequence identifier *sid* and $1 \leq i \leq n$. A set of regions extracted from the sequence database is represented as $B = \{B_1, B_2, \ldots, B_n\}$, where B_i is a region extracted from the sequence with the value i of the sequence identifier *sid* using Gibbs-DMGG. $\|A\|$ is defined as $\sum\|A_i\|$ $[1 \leq i \leq n]$, where $\|A_i\|$ denotes the length of region A_i. Furthermore, $A \otimes B$ is defined as $\{C_1, C_2, \ldots, C_n\}$, where $C_i = A_i \otimes B_i$ denotes the region of overlap between A_i and B_i. At this time, Recall is defined as $\|C\|/\|A\|$, and *Precision* is defined as $\|C\|/\|B\|$.

Table 1 Characteristics of the *Leucine Zipper* dataset

Number of sequences	Maximum length	Minimum length	Total length
188	1383	125	73673

Table 2 Relationship between the sequence database and extracted subsequence

sid	Sequences	Extracted subsequences
1	TATKFATFKT	ATFK
2	KATFAFTFAF	FAFT
3	AAKAKATFTK	AKAK
4	FAKATATFAA	ATFA
5	AATFTKFTTF	AATF

Consider computing Recall and Precision using Table 2, where the length k of the subsequence is the value of 4, the motif is represented as <*ATF*>, and the computational results are rounded to hundredths.

The total number $\|A\|$ of letters included in motif regions can be computed from $\|A_1\|+\|A_2\|+\ldots+\|A_5\|$ with $\|A\| = 15$. Since the total number $\|B\|$ of letters in the extracted regions can be calculated equally, the result is $\|B\| = 20$ using $k = 4$. Since all motif domains are contained in the subsequence of *sid* = 1, the result is $\|C_1\| = 3$. Since only "*F*" is contained in the subsequence of the sequence of *sid* = 2 among motifs <*ATF*>, the result is $\|C_2\| = 1$. In the following, since the result is $\|C_3\| = 0$, $\|C_4\| = 3$, and $\|C_5\| = 3$, $\|C\|$ is calculable with 10. Therefore, the Recall and Precision of this subsequence are calculated as Recall = 10/15 = 66.7% and Precision = 10/20 = 50.0%.

5.2 Experimental results

In order to conduct a performance evaluation of the proposed method, we changed each parameter of Gibbs-DMGG. First, we experimented by changing the number of islands. Table 3 gives the following parameters: 5 individuals in the population; 5 child individuals generated from a pair of parents; one mutation for one generation; 4800 generations until the termination of the execution; the number of islands is the value of 6, 10, 15, and 30; the number of trials is the value of 10 for computing the average Recall and the average Precision.

Regarding the number of islands, an average Recall of 99% or more denotes quite good extraction accuracy. In terms of Precision, the best value was achieved with 15 islands.

Table 3 Comparison of extraction performance by changing the number of islands

Number of islands	The number of computational results with the same range of length					Recall	Precision	CPU time (sec)	Number of generations
	54~60	61~70	71~80	81~90	91~114				
6	1	2	4	2	1	99.33	21.19	600.0	4800
10	0	3	5	1	1	99.29	20.67	985.8	4800
15	1	4	4	1	0	99.28	22.62	1447.7	4800
30	0	1	8	0	1	99.33	20.62	2670.6	4800

Table 4 Comparison of extraction performance by changing the number of individuals in the population

Number of individuals	The number of computational results with the same range of length				Recall	Precision	CPU time (sec)	Number of generations
	54~60	61~70	71~80	81~83				
5	1	4	4	1	99.28	22.62	1447.7	4800
10	0	5	5	0	99.26	21.64	3119.6	9600
20	0	1	8	1	99.12	20.93	6870.0	19200

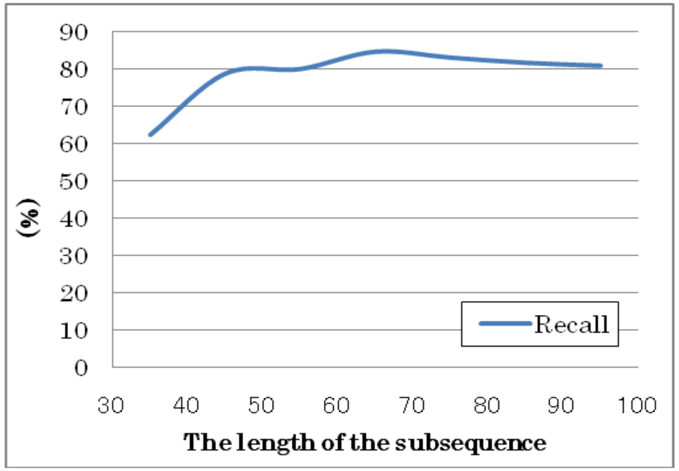

Figure 9 The extraction performance of Gibbs sampling

The difference in extraction results when the number of individuals in a population is changed was investigated using the case of the 15 islands that produced the best result (Table 3). Parameters other than the number of islands, the number of individuals, and the number of generations were used in the experiment (Table 3). The number of individuals in a mother group was changed to 5, 10, and 20, and the number of trials was 10. The number of generations was proportional to the change in the number of individuals and caused them to increase. The length of the subsequence, the average Recall, and the average Precision are shown in Table 4. Finally, the extraction performance of Gibbs sampling is shown in Figure 9.

Based on the information in Table 4, it is clear that the increase in the population is not proportional to the improvement in Precision. When the number of individuals is increased, however, the length in extracted substrings becomes larger.

The experimental results show that the extraction of a similar subsequence was most accurately attained with 15 islands and 5 individuals. At this time, the length of the subsequence extracted was approximately 70, and the Recall was 99% or greater. Based on the information in Figure 9, the Recall of Gibbs Sampling is around 85%. It is clear that the proposed method improves extraction performance. Moreover, the length of similar subsequences extracted by Gibbs-DMGG frequently results in almost 70 and is approximately equal to the user defined length providing for the maximum recall in executing the existing method. Therefore, it can be said that the partial array length extracted by this method is appropriate.

We successfully obtained similar subsequences with a region 1/5 that of the original database and 5 times the length of the motif (where the *Leucine Zipper* dataset includes a motif length ranging from 14 to 16 and an average of 392 letters for one sequence). As the extracted subsequence has not resulted in considerable, unnecessary, or even partial deletion from the original sequence database, the proposed method is considered an effective similar subsequence extraction method.

6. Conclusion

We have proposed a new similar subsequence extraction method, Gibbs-DMGG, which applies MGG and the Island model to Gibbs sampling. We successfully extracted similar subsequences that have 1/5 the original sequence database and 99% or greater Recall without specifying the length of the extracted subsequence. However, neither insertion nor deletion in the sequence was considered in the present study. Therefore, to improve conformity rates, further studies considering insertion and deletion to the sequence should be undertaken.

Acknowledgement This work was made possible in part by a Grant-in-Aid for Scientific Research (C) (2) (No.20500137) from the Japanese Society for the Promotion of Science and a Hiroshima City University Traveling Grant for Overseas Academic Meetings.

Reference

1) PROSITE. http:// kr.expasy.org/prosite.

2) Pfam. http://www.sanger.ac.uk/Software/Pfam.

3) Sonnhamer, E.L.L., Eddy, S.R., and Durbin, R. Pfam: A Comprehensive Database of Protein Domain Families Based on Seed Alignments. Proteins, Vol. 28, pp. 405-420, 1997.

4) Kato, T., Kitakami, H., Takaki, M., Tamura, K., Mori, Y., and Kuroki, S. Extraction for Frequent Sequential Patterns with Minimum Variable-Wildcard Regions, Proceedings of the 2006 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'06 & RTCOMP'06), Vol.Ⅱ, Las Vegas, Nevada, USA, pp. 825-831, June 26-30, 2006.

5) Kato, T., Kitakami, H., Mori, Y., Tamura, K., and Kuroki, S. Extraction of Non-redundant Frequent Sequence Patterns with Minimum Cover on Variable-Length Wildcard Regions, The IEICE Transaction on Information and Systems (Japanese Edition), Vol. J90-D, No. 2, pp. 281-291, February 2007.

6) Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.N., and Wotton, J. Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignment, Science, Vol. 263, pp. 208-214, 1993.

7) Satoh, H., Yamamura, M., and Kobayashi, S. Minimal Generation Gap Model for GAs Considering both Exploration and Exploitation. In Proceedings of the 4th International Conference on Soft Computing, Vol. 2, pp. 494-497, Fukuoka, Japan, 30 Sep-5 Oct 1996.

8) Yoshimura, J., Shimonobou, T., Sekiguchi, T., and Okamoto, M. Development of the Parameter-fitting Module for Web-based Biochemical Reaction Simulator BEST-KIT, Chem-Bio Informatics Journal, Vol. 3, No. 3, pp.114-129, 2003.

9) Cantu-Paz, E. Efficient and Accurate Parallel Genetic Algorithms, Springer, 2000.

10) Hiroyasu, T., Miki, M., and Kamiura, J. A Presumption of Parameter Settings for Distributed Genetic Algorithms by Using Design of Experiments, Transactions of Information Processing Society of Japan (Japanese Edition), Vol. 43, No. SIG10(TOM7), pp. 199-217, 2002.

11) Liu, J.S., Neuwald, A.N., and Lawrence, C.E. Bayesian Models for Multiple Local Sequence Alignment and Gibbs Sampling Strategies, JASA, Vol. 90, pp. 1156-1170, 1995.

12) Liu, L., Jiao, L., and Huo, H. A Greedy Two-stage Gibbs Sampling Method for Motif Discovery in Biological Sequences, 2008 International Conference on BioMedical Engineering and Informatics, Vol. 1, pp. 13-17, IEEE Computer Society Press, 2008.

13) Shida, K. Hybrid Gibbs-Sampling Algorithm for Challenging Motif Discovery: GibbsDST, Genome Informatics, Vol. 17, No. 2, pp. 3-13, 2006.