

Local Multiple Alignment of Numerical Sequences: Detection of Subtle Motifs from Protein Sequences and Structures

Tatsuya Akutsu¹

takutsu@kuicr.kyoto-u.ac.jp

Katsuhisa Horimoto²

horimoto@post.saga-med.ac.jp

¹ Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji, Kyoto 611-0011, Japan

² Laboratory of Mathematics, Saga Medical School, 5-1-1 Nabeshima, Saga, Saga 849-8501, Japan

Abstract

This paper presents a new method to find motifs from multiple protein sequences and multiple protein structures. The method consists of two parts: quantification and local multiple alignment. In the former part, protein sequences and protein structures are transformed into sequences of real numbers and real vectors respectively. In the latter part, fixed length regions having similar shapes are located. A Gibbs sampling algorithm for sequences of real numbers/vectors is newly developed for finding common regions. The results of the comparison with a standard Gibbs sampling program show that the method is particularly useful when structural information is available.

Keywords: Gibbs sampling, motif detection, protein structure, local multiple alignment

1 Introduction

Motif extraction is one of the well studied problems in Bioinformatics. Many methods have been proposed for rapid extraction of subtle motifs. Most of them are based on *local multiple alignment*. Local multiple alignment is a problem of locating a region of fixed length from each input sequence so that the *score* determined from the set of regions is optimized.

Stormo and Hartzell adopted the *relative entropy score* and developed a heuristic iterative algorithm for finding an optimal score [16]. This scoring scheme is widely used along with variants [5, 11]. Although local multiple alignment under the relative entropy scoring is NP-hard [1], several practical algorithms have been developed. EM (*expectation maximization*) algorithms [3, 10] and a *Gibbs sampling* algorithm [11] are widely used.

However, most previous methods are based on residue identity only and physico-chemical properties of residues or 3D structural information are not taken into account. In general, amino acid residues aligned at the same site are not necessarily identical even in the same protein family because of neutral changes under the same functional constraint. Of course, distinct residues can be aligned at the same site using a position specific score matrix [5]. However, similarities of residues are not taken into account by a position specific score matrix. Using a pairwise score matrix, physico-chemical properties can be taken into account. But, in such a case, we can no longer use the relative entropy score in a reasonable way. Of course, we can use the *SP (sum of pairs)* score. However, the SP score has some drawbacks [5]. Therefore, motif extraction methods based on physico-chemical properties and/or three dimensional properties should be studied.

In a previous paper [7], we developed a simple *quantification* method for transforming an amino acid sequence to a *numerical sequence* (i.e., a sequence of real numbers/vectors) so that physico-chemical

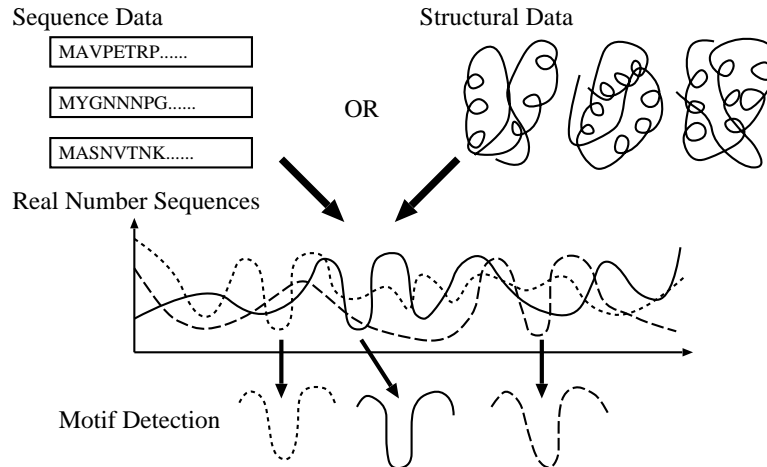


Figure 1: Motif detection using numerical sequences.

properties could be taken into account. The method was applied to extraction of basic/helix-loop-helix (bHLH) motifs and was compared with the results by sequence based motif extraction methods. The comparison results showed the effectiveness of the quantification method. However, extraction of motifs from numerical sequences was not automatic.

Many algorithms have been proposed for comparison of two protein structures [6, 14]. However, to our knowledge, there is no established algorithm for comparing *multiple* protein structures *simultaneously*. Indeed, we proved that finding a maximum common substructure from multiple structures is NP-hard under a simple and geometric definition of the problem [2].

This paper presents a new method (see Fig. 1) in which physico-chemical properties and structural information are taken into account. In this method, protein sequences and protein structures are transformed into sequences of real numbers and real vectors respectively. Then, common regions are located from these sequences. We develop a new Gibbs sampling algorithm to identify common regions. We also develop a simple *local search* algorithm. These two algorithms are compared using real protein sequence data. The present results show that the Gibbs sampling algorithm is much better than the simple local search algorithm. The Gibbs sampling algorithm for numerical sequences is also compared with a standard Gibbs sampling program [11, 13]. The new algorithm is better than the standard Gibbs sampling algorithm when structural information is available.

2 Local Multiple Alignment for Numerical Sequences

We developed two algorithms (LS-R and GIBBS-R) for local multiple alignment for numerical sequences based on previous algorithms for sequences of letters. Though GIBBS-R is better than LS-R, we describe both algorithms because GIBBS-R is developed based on LS-R, GIBBS-R must be compared with other algorithms, and LS-R has some theoretical property.

2.1 Local Search and Gibbs Sampling

In this subsection, we briefly review two algorithms developed for local multiple alignment of sequences of letters. Let Σ be an alphabet ($|\Sigma| = 20$ for protein sequences). For a sequence s over Σ , $|s|$ denotes the length of s and $s[i]$ denotes the i -th character of s . Thus, $s = s[1]s[2] \dots s[|s|]$.

Local multiple alignment for sequences is defined as follows: given a set $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ of sequences, and an integer L , find a subsequence t_i of length L from each s_i , maximizing the score of (t_1, \dots, t_n) , where gaps are not allowed in each t_i .

In this work, we adopted *relative entropy score* (i.e., average information content) since this score is widely used in sequence analysis. Let (t_1, \dots, t_n) be a set of subsequences of the same length L . Let $\#_j(a)$ be the number of the appearances of letter a in the j -th column of these sequences (i.e., $\#_j(a) = |\{t_i | t_i[j] = a\}|$). Let $f_j(a)$ be the frequency of letter a in the j -th column (i.e., $f_j(a) = \frac{\#_j(a)}{n}$). Let $p(a)$ denote the background probability of a . For example, we can let $p(a)$ be the frequency of letter a in the input sequences. Then, the relative entropy score is defined by

$$\text{score}(t_1, \dots, t_n) = \frac{1}{L} \sum_{j=1}^L \sum_{a \in \Sigma} f_j(a) \log \frac{f_j(a)}{p(a)}. \quad 1$$

Next, we present a local search algorithm LS [1]. LS can be seen as a simplified version of the EM algorithm, where similarities and differences are discussed in [8]. It should be noted that each iteration takes $O(mnL)$ time, where m is the average length of input sequences.

1. Select a subsequence t_i from each s_i at uniformly random.
2. Let $f_j(a)$ be the frequency of letter a in the j -th column of the subsequences.
3. For each i , find a subsequence t'_i of s_i maximizing $\sum_{j=1}^L \log \frac{f_j(t'_i[j])}{p(t'_i[j])}$.
4. Replace (t_1, \dots, t_n) with (t'_1, \dots, t'_n) .
5. Repeat steps 2–4 until the score does not increase (i.e., until reaching a local optimum).

LS: Simple Local Search Algorithm for Sequences of Letters

LS is a local search algorithm in a sense that the relative entropy score increases monotonically as the iteration steps proceed. Here, we briefly show this fact. Let (t_1, \dots, t_n) be subsequences appeared in STEP 2. Let (t'_1, \dots, t'_n) be subsequences selected in STEP 3. Let $f_j(a)$ and $f'_j(a)$ be the frequencies of letter a in the j -th column of t_i and t'_i ($i = 1, \dots, n$), respectively. Then, we have the following:

$$\sum_{j=1}^L \sum_{a \in \Sigma} f_j(a) \log \frac{f_j(a)}{p(a)} = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^L \log \frac{f_j(t_i[j])}{p(t_i[j])} \leq \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^L \log \frac{f_j(t'_i[j])}{p(t'_i[j])} = \sum_{j=1}^L \sum_{a \in \Sigma} f'_j(a) \log \frac{f_j(a)}{p(a)} \leq \sum_{j=1}^L \sum_{a \in \Sigma} f'_j(a) \log \frac{f'_j(a)}{p(a)},$$

where the latter inequality comes from the fact that $\sum p_i \log q_i \leq \sum p_i \log p_i$.

It should be noted that we can assume $f_j(a) > 0$ for all j, a by introducing a *pseudocount* [5].

Computational experiments show that LS converges to a local optimal very quickly (within several iterations) [1]. However, LS likely converges to local optima. The Gibbs sampling algorithm (GIBBS, in short) uses a stochastic search procedure to avoid getting trapped into local optima. GIBBS takes $O(mL + nL)$ time per iteration.

1. Select a subsequence t_i from each s_i at uniformly random.
2. Select a sequence s_i at uniformly random (or cyclically).
3. Let $f_j(a)$ be the frequency of letter a in the j -th column of the subsequences except t_i .
4. Select a subsequence t'_i of s_i with the probability proportional to $\prod_{j=1}^L \frac{f_j(t'_i[j])}{p(t'_i[j])}$.
5. Replace t_i by t'_i .
6. Repeat steps 2–5 for sufficiently many times.

GIBBS: Gibbs Sampling Algorithm for Sequences of Letters

Different from LS, the relative entropy score does not monotonically increase in GIBBS. However, GIBBS will reach the global optimal after a very long iteration.

¹ $\log x$ means $\log_2 x$ in this paper.

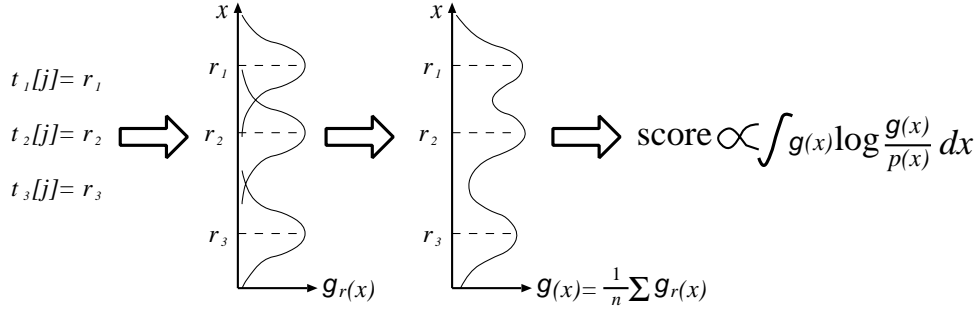


Figure 2: Relative entropy score for real number sequences.

2.2 Algorithms for Sequences of Real Numbers

We modify LS and GIBBS so that they can be applied to numerical sequences. In this subsection, we describe the algorithms for real number sequences.

In LS and GIBBS, we use frequencies of letters in each column since we only mind whether two letters are the same or not. In the case of real number sequences, the distance between two numbers must be taken into account. For that purpose, we associate the distribution $g_r(x)$ with each real number r . In this work, we use the normal distribution: $g_r(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-r)^2}{2\sigma^2}\right)$, though there is no concrete reason. Let (t_1, \dots, t_n) be sequences of real numbers of length L . Then, we define the relative entropy score (see Fig. 2) by

$$score(t_1, \dots, t_n) = \frac{1}{L} \sum_{j=1}^L \int_{-\infty}^{+\infty} \left\{ \frac{1}{n} \cdot \left(\sum_{i=1}^n g_{t_i[j]}(x) \right) \cdot \log \frac{\frac{1}{n} \cdot \sum_{i=1}^n g_{t_i[j]}(x)}{p(x)} \right\} dx.$$

It should be noted that “ \sum_a ” is replaced with “ $\int dx$ ” in the above. Using this scoring, we obtain a local search algorithm LS-R.

1. Select a subsequence t_i from each s_i at uniformly random.
2. For each i , find a subsequence t'_i of s_i maximizing

$$\sum_{j=1}^L \int \left\{ \frac{1}{n} \cdot g_{t'_i[j]}(x) \cdot \log \frac{\frac{1}{n} \cdot \sum_{i=1}^n g_{t_i[j]}(x)}{p(x)} \right\} dx.$$

3. Replace (t_1, \dots, t_n) with (t'_1, \dots, t'_n) .
4. Repeat steps 2–3 until the score does not increase (i.e., until reaching a local optimum).

LS-R: Local Search Algorithm for Sequences of Real Numbers

Here, we prove that the score monotonically increases in this case too.

$$\begin{aligned} & \sum_{j=1}^L \int_{-\infty}^{+\infty} \left\{ \frac{1}{n} \cdot \left(\sum_{i=1}^n g_{t_i[j]}(x) \right) \cdot \log \frac{\frac{1}{n} \cdot \sum_{i=1}^n g_{t_i[j]}(x)}{p(x)} \right\} dx \\ &= \sum_{i=1}^n \sum_{j=1}^L \int_{-\infty}^{+\infty} \left\{ \frac{1}{n} \cdot g_{t_i[j]}(x) \cdot \log \frac{\frac{1}{n} \cdot \sum_{i=1}^n g_{t_i[j]}(x)}{p(x)} \right\} dx \\ &\leq \sum_{i=1}^n \sum_{j=1}^L \int_{-\infty}^{+\infty} \left\{ \frac{1}{n} \cdot g_{t'_i[j]}(x) \cdot \log \frac{\frac{1}{n} \cdot \sum_{i=1}^n g_{t_i[j]}(x)}{p(x)} \right\} dx \end{aligned}$$

$$\begin{aligned}
&\leq \sum_{i=1}^n \sum_{j=1}^L \int_{-\infty}^{+\infty} \left\{ \frac{1}{n} \cdot g_{t'_i[j]}(x) \cdot \log \frac{\frac{1}{n} \cdot \sum_{i=1}^n g_{t'_i[j]}(x)}{p(x)} \right\} dx \\
&= \sum_{j=1}^L \int_{-\infty}^{+\infty} \left\{ \frac{1}{n} \cdot \left(\sum_{i=1}^n g_{t'_i[j]}(x) \right) \cdot \log \frac{\frac{1}{n} \cdot \sum_{i=1}^n g_{t'_i[j]}(x)}{p(x)} \right\} dx.
\end{aligned}$$

In practice, the integral must be replaced by the summation. In the current implementation, input data are normalized so that the values lie in the domain of $[-15\dots 15]$ and the integral is replaced by the summation for 100 segments of the same length. If the domain is divided into K segments, each iteration takes $O(mnLK)$ time.

GIBBS can also be modified for sequences of real numbers as below.

1. Select a subsequence t_i from each s_i at uniformly random.
2. Select a sequence s_i at uniformly random (or cyclically).
3. Select a subsequence t'_i of s_i with the probability proportional to

$$\exp \left(\sum_{j=1}^L \int \left\{ \frac{1}{n} \cdot g_{t'_i[j]}(x) \cdot \ln \frac{\frac{1}{n} \cdot \sum_{k \neq i} g_{t_k[j]}(x)}{p(x)} \right\} dx \right).$$

4. Replace t_i by t'_i .
5. Repeat steps 2–4 for sufficiently many times.

GIBBS-R: Gibbs Sampling Algorithm for Sequences of Real Numbers

GIBBS-R takes $O(mLK + nLK)$ time per iteration. In the current implementation, STEPS 2–4 are iterated 400 times and the best solution is selected as an output. Though we did not prove any theoretical property on the scores of GIBBS-R, GIBBS-R worked well for real number sequences.

It should also be noted that GIBBS-R (resp. LS-R) is almost identical to GIBBS (resp. LS) if the integral is replaced by the summation and σ is very small.

2.3 Algorithms for Sequences of Real Vectors

Modification of LS-R and GIBBS-R for sequences of real vectors is very easy. Each coordinate of vectors is processed independently and the score is defined to be the sum of the scores for all coordinates. Let $\mathbf{t}_i[j]$ be a D -dimensional vector assigned to the j -th residue of the i -th sequence. Let $\mathbf{t}_{i,k}[j]$ denote the k -th coordinate of a vector $\mathbf{t}_i[j]$ (i.e., $\mathbf{t}_i[j] = (\mathbf{t}_{i,1}[j], \mathbf{t}_{i,2}[j], \dots, \mathbf{t}_{i,D}[j])$). Then, we define the relative entropy score by

$$score(t_1, \dots, t_n) = \frac{1}{DL} \sum_{k=1}^D \sum_{j=1}^L \int_{-\infty}^{+\infty} \left\{ \frac{1}{n} \cdot \left(\sum_{i=1}^n g_{\mathbf{t}_{i,k}[j]}(x) \right) \cdot \log \frac{\frac{1}{n} \cdot \sum_{i=1}^n g_{\mathbf{t}_{i,k}[j]}(x)}{p(x)} \right\} dx.$$

Modification of the other parts of LS-R and GIBBS-R is straight-forward.

3 Quantification of Sequences and Structures

Many indices have been proposed for quantification of protein sequences. For example, the *hydropathic index* is a famous one for measuring hydropathic properties of amino acids [9]. In this paper, we use another method which was previously proposed by us [7]. The method was based on PCA (*principal component analysis*). In order to apply PCA to protein sequences, four physico-chemical properties, polarity, hydrophobicity, volume and pK_a , were adopted as the variables that characterized amino acid

residues. Using this method, each protein sequence is transformed into a sequence of real numbers of the same length.

For quantification of protein structures, we adopt a very simple method in which each protein structure is transformed into a sequences of *real vectors*. Let $\mathbf{p}_i[j]$ denote the position of the j -th C α atom of the i -th protein structure. We define the k -th coordinate value of $\mathbf{t}_i[j]$ by

$$\frac{1}{2H+1} \sum_{h=j-H}^{j+H} dist(\mathbf{p}_i[h], \mathbf{p}_i[h+d_k]),$$

where $dist(\mathbf{x}, \mathbf{y})$ is the distance between points \mathbf{x} and \mathbf{y} . That is, each coordinate value of a vector is the (averaged) distance between the j -th C α atom and the $(j+d_k)$ -th C α atom. Currently, we use $H=1$, $D=2$, $d_1=15$, and $d_2=4$. The second coordinate of each vector reflects secondary structure of the residue: it takes a small value if the j -th residue belongs to an α -helix. The first coordinate reflects the relative configuration of secondary structures. It should be noted that, in quantification of protein structures, only structural information are used and information about residue types are not taken into account.

The choice of the background probability $p(x)$ is also an important problem in local multiple alignment. We adopted $p(x)$ defined by $p(x) \propto \sum_{i,j} g_{t_i[j]}(x)$ for sequence data and adopted the flat probability (i.e., $p(x) = const$) for structural data based on a few trials. *Pseudocount* must be used in order to avoid ‘log(0) error’ [5]. We adopted a simple pseudocount (adding the same constants) in the current implementation.

4 Results

We compared GIBBS-R with LS-R and compared GIBBS-R with the widely used Gibbs sampling program `gibbs9_95` [11, 13]. We used a standard PC with Pentium-III 1GHz CPU and 256MB main memory, where it was working under the LINUX operating system. LS-R and GIBBS-R were implemented using C language.

It should be noted that several additional techniques were introduced in `gibbs9_95`, whereas a simple implementation of GIBBS-R was used. Since `gibbs9_95` is a very powerful tool and can identify most motifs if a large number of sequences are input, we do not aim to compare GIBBS-R with `gibbs9_95` directly. For comparison with `gibbs9_95`, we only examined cases where the numbers of input sequences were small.

4.1 Detection of 1D Motifs

We applied GIBBS-R and LS-R to the following data sets of protein sequences, for each of which `gibbs9_95` identified the correct regions of the motifs [11].

- (i) Helix-turn-helix motif: RP32_ECOL, TER2_ECOL, RCRO_LAMB, DEOR_ECOL, RPC2_LAMB, LACLECO, NAHR_PSEP, NIFA_KLEP, DICA_ECOL, FIS_ECOLI, MERD_SERM, HMAN_DROM, LEXA_ECOL, ARAC_ECOL.
- (ii) Lipocalins: ICYA_MANSE, LACB_BOVIN, BBP_PIEBR, RETB_BOVIN, MUP2_MOUSE.

The outputs of both GIBBS-R and LS-R depend on the sets of randomly selected initial positions. Thus, we executed GIBBS-R (resp. LS-R) multiple times and chose a set of regions with the highest score. Since LS-R converged very quickly to local optima, LS-R was executed 500 times for each data set whereas GIBBS-R was executed 50 times. $L=15$ and $\sigma=0.1$ were used in all cases. Table 1 summarizes the results. For each data set and for each algorithm, the highest score, the average score and the CPU time (sec.) per execution are shown.

It is seen that both the best score and the average score of GIBBS-R are much better than those of LS-R. It is also seen that LS-R quickly converges to local optima, but which are far from the global

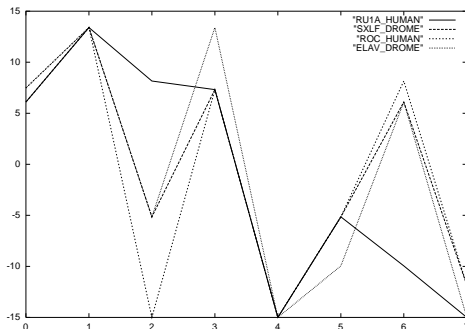


Figure 3: Identified RNP-1 regions in numerical sequences.

optimal. Indeed, GIBBS-R identified almost correct regions in all cases (GIBBS-R can identify both motifs A and B of lipocalins though the scores and the time for motif A are only shown in the above), whereas LS-R failed to identify for all cases. From this result, it is confirmed that the Gibbs sampling technique is useful for multiple local alignment of real number sequences.

Table 1: Comparison of LS-R and GIBBS-R.

| | (i)BEST | (i)AVE. | (i)TIME | (ii)BEST | (ii)AVE. | (ii)TIME |
|---------|---------|---------|---------|----------|----------|----------|
| LS-R | 1.06 | 0.88 | 0.15 | 1.99 | 1.74 | 0.03 |
| GIBBS-R | 1.57 | 1.42 | 1.63 | 2.59 | 2.32 | 1.03 |

In general, the current implementation of GIBBS-R was no better than `gibbs9_95` for sequence data. However, there was a case where GIBBS-R worked better than `gibbs9_95`. Here, we show that case because it is an illustrative example. We applied GIBBS-R and `gibbs9_95` to the identification of the RNP-1 motif in a major family of RNA binding proteins [5]. The motif consists of 8 residue positions and is shown on the left hand side of the figure below. GIBBS-R output the correct motif regions as a second candidate of the motif (the second best solution among multiple trials). On the other hand, `gibbs9_95` could not identify correct regions though we executed `gibbs9_95` 100 times. The best result (not necessarily the best score) of 100 trials was shown on the right hand side.

| | MOTIF | <code>gibbs9_95</code> |
|------------|-----------------------------|-----------------------------|
| RU1A_HUMAN | ...srslkmRGQAFVIFkevssat... | ...srslkmrGQAFVIFKEVSSAT... |
| SXLF_DROME | ...kltgrprGVAFVRYnkreeaq... | ...kltgrprGVAFVRYNKREEAQ... |
| ROC_HUMAN | ...vgcsvhKGFAFVQYvnernar... | ...vgcsvhkGFAFVQYVNERNAR... |
| ELAV_DROME | ...gndtqtKGVGFIRFdkreeat... | ...pskgqs1GYGFVNYVRPQDAE... |
| | ***** | |

The reason why GIBBS-R identified the correct regions is that residues R and K are transformed into very close values (see Fig. 3), whereas `gibbs9_95` treated these residues as distinct residues. It is highly expected that similar situations will occur for other motifs. Thus, GIBBS-R may be useful for identification of motifs from small sets of protein sequences.

4.2 Detection of 3D Motifs

GIBBS-R is particularly useful when it is applied to structural data. We examined the following three cases from Ref. [12]: (i) the acyltransferase family, (ii) the ADP-binding $\beta\alpha\beta$ -fold, and (iii) the helix-turn-helix motif. For each case, `gibbs9_95` found the correct motif regions when the number of

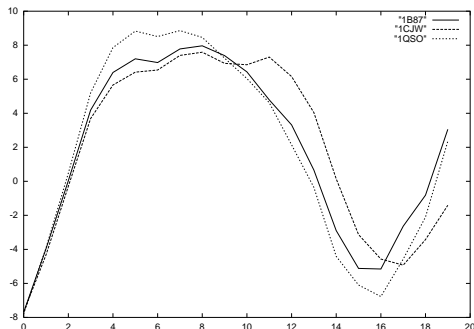


Figure 4: Identified motif regions in real vector sequences (1st coordinate).

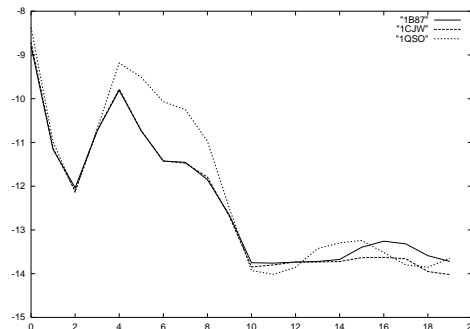


Figure 5: Identified motif regions in real vector sequences (2nd coordinate).

input sequences was not small (e.g., ≥ 8). However, `gibbs9_95` failed to find the correct regions when the number of input sequences was small. Thus, we examined such cases to verify the effectiveness of GIBBS-R.

We executed GIBBS-R (resp. `gibbs9_95`) 50 times for each case and selected the solution with the best score. We used $\sigma = 3.0$ in GIBBS-R. The total execution time was less than a minute in each case. In all cases, GIBBS-R identified correct or near correct regions, but `gibbs9_95` identified incorrect regions. It should be noted that GIBBS-R uses structural data only and `gibbs9_95` uses sequence data only. In each case, physico-chemical properties of amino acids are not taken into account.

(i) The acyltransferase family.

It is known that this family has two motifs, where we only focus on the first motif (Motif A in [12]). We used the structural data of 1B87, 1CJW and 1QSO from PDB [4] and used $L = 20$. The following shows the regions identified by GIBBS-R and `gibbs9_95`, where a typical pattern among 50 trials is shown for `gibbs9_95` because the solution with the best score was far from the correct motif. A typical motif pattern is also shown at the bottom on the left hand side.

| | GIBBS-R | <code>gibbs9_95</code> |
|------|---------------------|------------------------|
| 1B87 | VVSSRRKNQIGTRLVNYL | LHPLVVSSRRKNQIGTRLVNY |
| 1CJW | AVHRSFRQQGKGSVLLWRY | LHALAVHRSFRQQGKGSVLLWR |
| 1QSO | YVDNSRVKGGKLIQFV | WAAVAVESSEKIIGMINFFNH |
| | .V.PDHRGKGIG..LI..I | |

The reason why the length of the identified regions by `gibbs9_95` is different from that by GIBBS-R is that the length by `gibbs9_95` depends on the result of execution. It should be noted that `gibbs9_95` identified the correct regions for the first two sequences, but an incorrect region for the third sequence.

The identified regions in the real vector sequences are shown in Fig. 4 and Fig. 5. It is seen that the identified regions have similar patterns.

(ii) The ADP-binding $\beta\alpha\beta$ -fold.

Many FAD-, NAD- and NADP-binding domains share a $\beta\alpha\beta$ -fold that binds the ADP-moiety of dinucleotide. One motif is known for this binding domain [12]. We used structural data of 1AN9, 1GAL and 1PGN. In this case, GIBBS-R identified almost correct regions under $L = 25$ as below, where the position of the first sequence was shifted only by 2 residues. Slight shift is reasonable because the distances between $C\alpha$ atoms change continuously. `gibbs9_95` failed in all trials, where a typical result is shown on the right hand side.

| | GIBBS-R | <code>gibbs9_95</code> |
|------|--------------------------|----------------------------|
| 1AN9 | RVVVIGAGVIGLSTALCIHERYHS | LQAVTLGGTFQVGNWNEINNIQDHNT |
| 1GAL | TVDYIIAGGGLTGLTTAARLTENP | AQVDSWETVFGNEGWNWDNVAAYSLQ |
| 1PGN | QADIALIGLAVMGQNLILNMNDHG | AQADIALIGLAVMGQNLILNMNDHGF |

(iii) The helix-turn-helix motif.

The helix-turn-helix (HTH) motif is famous and found in a diverse family of DNA binding proteins. We used structural data of 1ADR, 1FIA and 6CRO. In this case, GIBBS-R identified the correct regions under $L = 15$ as below. `gibbs9_95` failed in all trials, where a typical result is shown on the right hand side.

| | GIBBS-R | <code>gibbs9_95</code> |
|------|----------------|------------------------|
| 1ADR | RQAALGKMVGVSNV | MGERIRARRKCLKIRQA |
| 1FIA | NQTRAALMMGINRG | MGINRGTLRKKLKKYGM |
| 6CRO | GQTKTAKDLGVYQS | INKAIHAGRKIFLTINA |

5 Conclusion

We proposed a Gibbs sampling algorithm (GIBBS-R) for sequences of real numbers/vectors. This is a simple but non-trivial variant of the Gibbs sampling algorithm for sequences of letters [11, 13].

For detection of 1D motifs (motifs based on sequence information), GIBBS-R was in general no better than the standard Gibbs sampling program `gibbs9_95` [11, 13] while GIBBS-R may work better in some cases, especially for small sets of protein sequences. For detection of 3D motifs (motifs based on structural information), GIBBS-R could find several motifs that could not be found by `gibbs9_95`. The followings are considered as major reasons why GIBBS-R was more successful for the detection of 3D motifs than the detection of 1D motifs: (i) Motifs are usually defined based on its function, and function is closely related with its structure. Thus, it is reasonable that structural information contributes to the detection of motifs more than sequence information. (ii) Distances between $C\alpha$ atoms change continuously, whereas quantified values of residues change non-continuously (compare Fig. 3, Fig. 4 and Fig. 5). Thus, GIBBS-R is robust for small insertions/deletions of $C\alpha$ atoms.

It should be noted that GIBBS-R was not optimized. There are many rooms of improvements: choice of $g_r(x)$, choice of $p(x)$, introduction of Dirichlet priors, introduction of weighted prior for segmentation, \dots . Therefore, if GIBBS-R is optimized using these, the performance of GIBBS-R for 1D sequences might be significantly improved.

Gaps (insertions/deletions) are not allowed in the current implementation. However, the original Gibbs sampling algorithm was already modified for finding gapped motifs from sequence data [15]. Modification of GIBBS-R for finding gapped motifs should also be studied.

Though we have examined only one quantification method [7], other amino acid indices can be used for quantification. For example, if the hydrophobic index is used, motifs reflecting hydrophobic properties might be detected. Moreover, multiple indices can be used since sequences of real vectors can be treated by GIBBS-R.

We applied GIBBS-R to protein sequences and protein structures. In general, GIBBS-R can be applied to any sequences if the sequences can be transformed into sequences of real numbers/vectors. For example, it might be applied to motif detection of DNA sequences, and detection of similar genes from time series data of gene expression profiles. Especially, if appropriate indices or scores are defined for DNA fragments of fixed size, GIBBS-R may be applied to the detection of regulatory sites of DNA sequences. This topic should be explored since understanding of regulatory mechanisms of DNA sequences is a very important problem.

Acknowledgements

This work was partially supported by Grant-in-Aid for Scientific Research on Priority Areas (C) ‘‘Genome Information Science’’, Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan. Tatsuya Akutsu was also partially supported by HITOCC (Hyper Information Technology Oriented Corporation Club) and Grant-in-Aid #13680394 from MEXT, Japan.

References

- [1] Akutsu, T., Arimura, H., and Shimozone, S., On approximation algorithms for local multiple alignment, *Proc. 4th Int. Conf. Computational Molecular Biology (RECOMB-2000)*, 1–7, 2000.
- [2] Akutsu, T. and Halldórson, M.M., On the approximation of largest common subtrees and largest common point sets, *Theoretical Computer Science*, 233:33–50, 2000.
- [3] Bailey, T.L. and Elkan, C., Unsupervised learning of multiple motifs in biopolymers using expectation maximization, *Machine Learning*, 21:51–80, 1995.
- [4] Bernstein, F.C. *et al.*, The Protein Data Bank: A computer-based archival file for macromolecular structures, *J. Molecular Biology*, 112:535–542, 1977.
- [5] Durbin, R., Eddy, S., Krogh, A., and Mitchison, G., *Biological Sequence Analysis. Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, 1998.
- [6] Holm, L. and Sander, C., Protein structure comparison by alignment of distance matrices, *J. Molecular Biology*, 233:123–138, 1993.
- [7] Horimoto, K., Yamamoto, H., Yanagi, K., Ohshima, K., and Otsuka, J., A simple procedure for assigning a sequence motif with an observed pattern: application to the basic/helix-loop-helix motif, *Protein Engineering*, 7:1433–1440, 1994.
- [8] Horton, P., Alignment vs. sum of all alignments scoring for motif extraction, *Proc. 6th SIGMPS Symposium*, Information Processing Society of Japan, 2000.
- [9] Kyte, J. and Doolittle, R.F., A simple method for displaying the hydropathic character of a protein, *J. Molecular Biology*, 157:105–32, 1982.
- [10] Lawrence, C.E. and Reilly, A.A., An expectation maximization (EM) algorithm for identification and characterization of common sites in unaligned biopolymer sequences, *PROTEINS: Structure, Function, and Genetics*, 7:41–51, 1990.
- [11] Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.F., and Wootton, J.C., Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment, *Science*, 262:208–214, 1993.
- [12] Neuwald, A.F. and Green, P., Detecting patterns in protein sequences, *J. Molecular Biology*, 239:698–712, 1994.
- [13] Neuwald, A.F., Liu, J.S., and Lawrence, C.E., Gibbs motif sampling: detection of bacterial outer membrane protein repeats, *Protein Science*, 4:1618–1632, 1995.
- [14] Orengo, C.A. and Taylor, W.R., A local alignment method for protein structure motifs, *J. Molecular Biology*, 233:488–497, 1993.
- [15] Roche, E. and Tompa, M., An algorithm for finding novel gapped motifs in DNA sequences, *Proc. 2nd Int. Conf. Computational Molecular Biology (RECOMB-98)*, 228–233, 1998.
- [16] Stormo, G. and Hartzell, G.W., Identifying protein-binding sites from unaligned DNA fragments, *Nucleic Acids Research*, 86:1183–1187, 1989.