ICOT Technical Report: TR-0793

TR-0793

Protein Multiple Sequence Alignment using Knowledge

by
M. Hirosawa, M. Hoshida & M. Ishikawa

August, 1992

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
(03)3456-3191～5
Telex ICOT J32964
Minato-ku Tokyo 108 Japan

Institute for New Generation Computer Technology

Protein Multiple Sequence Alignment using Knowledge

Makoto Hirosawa, Masaki Hoshida, Masato Ishikawa
(Institute for New Generation Computer Technology (ICOT),
1-4-28 Mita,Minato-ku, Tokyo 108, Japan)

Key Words : Multiple Alignment, Knowledge, Motif

Abstract

Protein sequence alignment is an important technique for drawing phylogenic trees and predicting the structure and function of proteins. So far, many alignment systems have been developed. These systems produce alignment that is optimal or near-optimal according to an artificially defined evaluation function. However, these computationally optimal alignments don't necessarily correspond to the biologically optimal alignment that biologists desire.

We interviewed experts on alignment and extracted the knowledge they use to align sequences. The alignment system we will introduce in this paper produces biologically meaningful (sometimes optimal) alignment by using the knowledge we extracted.

1 Introduction

Similarity analysis of protein sequences by the use of multiple alignment is an important technique for drawing the phylogenic trees of creatures and for predicting the function and structure of proteins. By simultaneously aligning sequences of similar proteins, we can identify regions of protein sequences that may have important functions when these sequences are folded into proteins. Also, by simultaneously aligning the same proteins (myoglobin, for example) from different species of creatures, we can analyze the similarity between sequences belonging to creatures and draw phylogenic trees of creatures. Because biological expertise is necessary for multiple alignment, multiple alignment has been produced by hand by biologists up until recently.

However with the increasing rate of determination of protein sequences, the number of multiple alignments that biologists must handle are has also increased remarkably. The difficulty of each multiple alignment has become harder because the number of sequences that should be aligned and the length of their sequences has increased. This situation has come to impose more burden on biologists and the introduction of computers into multiple alignment has become indispensable. Therefore, research has been done to facilitate multiple alignment by computer.

When computer scientists began to explore how to use computers to make alignment, we may guess that they had a desire to invent an algorithm that produces the optimal alignment. However, what is the optimal alignment? Without defining the optimal alignment, how could they attempt to find the optimal alignment?. Of course, they might ask biologists what the optimal alignment is. However, what they found was the reality that the optimal alignment varies from biologist to biologist.

Computer scientists circumvented this difficulty by defining the computationally optimal alignment instead of searching for the biological optimal or biological meaningful alignment. They introduced a similarity index between amino acids. Then, they defined the optimal multiple alignment

based on the similarity index. One of the similarity indices generally employed is the Dayhoff score index [Dayhoff 1978].

Needleman and Wunsch [Needleman and Wunsch 1970] showed that a computationally optimal alignment of n sequences can be produced by Dynamic Programming (DP). However, the problem with DP is the incredible computational time it requires. Therefore, other algorithms that can produce semi-optimal alignment within a limited time have been developed.

Today, biologists use these programs to produce a temporary alignment, which biologists have to refine to produce the final alignment . They must refine by themselves the alignment into a biological meaningful alignment because the alignments that these programs produce are just computational optimal or semi-optimal alignments instead of biologically optimal alignments.

As was mentioned before, the number of alignments that biologists must handle is increasing, and the difficulty of each alignment is increasing. Therefore, biologists are now feeling the need for computer assistance even for refining temporary alignments. Also, with the accumulation of protein sequence data in databases, researchers other than biology experts can have the opportunity to make biological discoveries by just analyzing sequence data. Automatic refinement of alignments using biological know-how and knowledge is also necessary for them.

As knowledge to be used in the refinement phase, the heuristics that biologists rely on are rather ambiguous. However, knowledge on the biological meaningful portion of sequences has been accumulated in databases, one of which is Prosite [Bairoch 1991], and another information has been published in various papers.

We have developed an alignment system composed of two modules, namely, *the aligner* and *the intelligent refiner*. The aligner produces a computationally semi-optimal alignment. Then, the intelligent refiner refines the product of the aligner to produce the biologically optimal alignment.

To design an intelligent refiner, we interviewed experts on multiple alignment. We base the framework of the intelligent refiner on analysis of the way they align sequences and the knowledge

they use for alignment.

In this paper, we explain the alignment system with the intelligent refiner. In the second section, we explain why the intelligent refiner is important. In the third section, the alignment system with the intelligent refiner is introduced. Then, in the fourth section, examples of application of the intelligent refiner are shown. Finally, discussion is delivered in the fifth section.

2 Why is an intelligent refiner necessary?

In this section, firstly, computationally optimal alignment is defined, and previous works on finding computationally optimal alignment are reviewed in subsection 2.1. Then, in subsection 2.2, it is shown that computationally optimal alignment does not always corresponds to biologically optimal alignment. Finally, in subsection 2.3, the possibility that we can make biologically optimal alignments from computationally optimal alignments by using biological knowledge will be indicated.

2.1 Computationally optimal alignment

Computer scientists have been engaged in finding an algorithm that produces an optimal alignment in the smallest time possible. They defined the evaluation function of the alignment to be optimized. They have strived to produce a computationally optimal alignment.

Definition of computationally optimal alignment

Generally, the evaluation function is defined as below. Firstly, we must define the similarity index between amino acids, `sim(A1,A2)`. A1 and A2 signify one of twenty kinds amino acids. This index can be represented by a two dimensional matrix with 20 columns and 20 rows. We select the Dayhoff similarity matrix [Dayhoff 1978] for producing an optimal alignment.

Secondly, the alignment of each column is evaluated. When the number of sequences is two, the evaluation value of each column is the similarity index between two amino acids which belong to the column.

When there are more than two sequences to be aligned, ideally a similarity index of more than three sequences, e.g. `sim(A1,A2,A3)`, should be employed. However, because there is no well-defined similarity index for more than three sequences, a heuristic similarity index is usually employed. As heuristic evaluation of each column, the summation of the similarity index between any pair of amino acids belonging to the column is usually used. In the case of three sequences, `sim(A1,A2) + sim(A2,A3) + sim(A3,A1)` is used for `sim(A1,A2,A3)`.

Finally, the evaluation value of each column is summed up through the sequences to generate an overall evaluation value of alignment.

Review of algorithms making computationally optimal alignment

This fixation on the evaluation function of alignment encouraged computer scientists to conduct research into multiple alignment. This resulted in a proliferation of algorithms capable of producing alignment.

Needleman and Wunsch [Needleman and Wunsch 1970] introduced Dynamic Programming(DP) into multiple alignment. With n way DP, the computationally optimal alignment can be produced theoretically. However, the problem with DP is the incredible computational time it requires. N-way DP takes computational time in the order of the n-th power of the sequence length. If we align sequences whose length are short, by resricting solution space we can align the sequences within manageable time[Carrillo and Lipman 1988]. However, if we apply the program to practical problem, the program requires extremely long time.

To keep this expansion of computational time manageable, a variety of multiple alignment algorithms have been developed that can produce a semi-optimal alignment within a limited time. As algorithms with 2 way DP, tree based algorithms [Johnson and Dolittle 1986] [Barton 1990] and the iterative improving algorithm[Berger and Manson 1991][Ishikawa *et al.* 1992] were developed. As an iterative improving alignment system without 2 way DP, an alignment using simulated annealing

was developed[Ishikawa *et al.*].

Recently, with the increasing power of the computer, 2 way DP as well as 3 way DP has become available [Murata 1985] and alignment system based on 3 way DP has been developed [Hirosawa *et al.* 1991].

2.2 Biological reliability of computationally optimal alignment

It is hard to define biologically optimal alignment because biologically optimal alignment depends on the sequences to be aligned. We select some sequences as an example, and we define biologically optimal alignment of the sequences. The definition of the biologically optimal alignment for this case is done in 2.2.1. Then, the biological reliability of the computationally optimal alignment is investigated in 2.2.2.

2.2.1 Definition of biologically optimal alignment

The sequences we will use to define the biologically optimal alignment are six sequences of endonuclease (that cuts nuculeotide) from Retro-virus (which the AIDS virus belongs to) and its relatives. This is shown in Figure 1, and one of its biological optimal alignments is shown in Figure 2.

```
17.6    : ILDFHEKLLHPGIQKTTKLFGETYYFPNSQLLIQNIINECSICNLAKTEHRNTDMPTKTT
M-MuLV  : LLDFLHQLTHLSFSKMKALLERSHSPYYMLNRDRTLKNITETCKACAQVNASKSAVKQGTR
HTLV    : LTDALLITPVLQLSPAELHSFTHCGQTALTLQGATTTEASNILRSCHACRGGNPQHQMPRGHI
RSV     : VADSQATFQAYPLREAKDLHTALHIGPRALSKACNISMQQAREVVQTCPHCNSAPALEAGVN
MMTV    : ISDPIHEATQAHTLHHLNAHTLRLLYKITREQARDIVKACKQCVVATPVPHLGVN
SMRV    : ILTALESAQESHALHHQNAAALRFQFHITREQAREIVKLCPNCPDWGSAPQLGVN
```

Figure 1 Sequences to be aligned
This set of sequences is from endonuclease of retrovirus and its relatives

```
17.6    : -------ILD--F-------HEKLLHPGIQKTTK-LF--GET-YY-FPNSQLLIQNIINECSICNL-AKT-EHR--N-TDMPTKT
M-MuLV  : -------LLD-FL-------HQ-LTHLSFSKM-KALLERSHSPYYMLNRDRTL-KNITETCKACAQ-VNA-SKS--A-VKQGTR-
HTLV    : LTDALL-ITP-VLQLSPAELHS-FTHCGQTAL-T-LQ--------GATTTEA--SNILRSCHACRG-GNPQHQMPRGHI------
RSV     : VADSQATFQAYPLR-EAKDLHT-ALHIGPRAL-S-KA-------CNISMQQA--REVVQTCPHC---NSA-PALEAG-VN-----
MMTV    : -------ISD-PIH-EATQAHT-LHHLNAHTL-R-LL-------YKITREQA--RDIVKACKQCVV-ATPVPHL--G-VN-----
SMRV    : -------ILT-ALE-SAQESHA-LHHQNAAAL-R-FQ-------FHITREQA--REIVKLCPNCPDWGSA-PQL--G-VN-----
                          *    *                                  *  *
```

Figure 2 Aligned Sequences
The alignment identifies the reverse pattern of the prototype zinc finger

The column whose elements are identical is marked by "*". These columns are called conserved columns. There are four conserved column "H"s and "C"s. Some patterns of conserved columns are known to have biological meaning and are called motifs. The motif consisting of conserved patterns of "HXXXH" and "CXXC" is a reverse prototype motif of zinc finger. "X" in the pattern means any amino acid. The protein with the zinc finger is one of the proteins known to have the capability to bind to DNA/RNA sequences.

We define the biologically optimal alignment as the alignment that identifies the zinc finger pattern. The homology score (the index that measures the similarity between sequences) of sequences in the alignment is very low at 26 percent. It indicates that these sequences are difficult to align.

2.2.2 Biological reliability of computationally optimal alignment

Alignment of three sequences

Because the computationally optimal alignment of three sequences is obtained by 3-way DP, we investigated the biological reliability of alignments produced by 3-way DP.

We selected three sequences from these six sequences and aligned the three sequences by 3-way DP. There can be 20 triplets of sequences and a corresponding number of alignment were produced. Then, we investigated whether the alignment corresponds to the biologically optimal alignment.

Four alignments of triplet sequences selected from {HTLV,RSV,MMTV,SMRV} catch the zinc finger motif. It can be explained that these alignments can catch the zinc finger because the homology score between the four sequences is rather high (32 percent). However, the quality of the other sixteen alignments is low. Only alignment of {17.6, M-MuLV, MMTV} and that of {M-MuLV, MMTV, SMRV} catch the zinc finger motif.

Optimal alignment of more than 4 sequences

It is theoretically possible to implement more than 4-way DP. However, because it is hard to implement more than 4-way DP, usually the computationally near-optimal alignment is produced by the methods reviewed in 2.1.

By using these alignment methods, we have experimented on alignment of various kinds of protein. From our experience with multiple alignment, it can be said that computationally near-optimal alignment of more than 4 sequences doesn't necessarily correspond to the biologically optimal alignment.

2.3 Possibility of Inference of biologically optimal alignment from computationally optimal alignment

As was shown in the preceding two subsections, the computationally optimal or semi-optimal alignment doesn't necessarily correspond to biological optimal alignment.

However, biologists and even computer scientists with biological knowledge can make biologically optimal alignment by refining the computationally optimal or near-optimal alignment.

We will show you examples in which we can make the biologically optimal alignment by refinement if we use biological knowledge. Two example alignments are produced 3 way DP. So, the alignments are computationally optimal alignments. The sequences to be aligned are the sequences whose biological optimal alignment is defined in 2.2.1. The biologically optimal alignment must identify zinc finger pattern.

Example alignment 1

The evaluation value of example alignment 1 is 157. In the figure, the conserved columns are marked by "*", as before. Columns whose elements are not completely identical are marked by "+". We call these half conserved columns. In this case, the half conserved column contains two identical amino acids.

Because the sequences contains the zinc finger, we must refine the alignment to identify the zinc finger. As explained before, the zinc finger motif consists of "HXXXH" and "CXXC". In the alignment, "CXXC" is identified. And a consensus column of "H", which is the last letter of "HXXXH", is identified. However, the amino acids which should correspond to the first letter of "HXXXH" are scattered on both sides of conserved column of "E".

To make conserved column of "H", conserved column of "E" must be broken. However, because the general strategy of the refinement of alignments is to fix conserved columns and to modify the region between conserved columns (this strategy was employed when [butler *et al.* 1990] wrote multiple alignment program), we cannot break the conserved column. The result is that we cannot identify the zinc finger.

However, actually we can refine the alignment to identify the zinc finger motif if we know zinc finger motif. However, because the evaluation value of the alignment is reduced form 145 to 134 when we refine the alignment, we cannot identify the zinc finger without biological knowledge.

```
17.6 : ---------IL-DF---HE--KLLHPGIQKTTKL-FGETYYFPNSQLLIQNIINECSICNLAKTEHRNTDMPTKTT-
HTLV : LTDALLITPVL-QL-SPAELHSFTHCG-Q--TALTLQGA---TTTE--ASNILRSCHAC---RGGNPQHQMP-RGHI
SMRV : ---------ILTALESAQESHALHH---QNAAALRFQ--FHITREQ--AREIVKLCPNCP-DWGSAPQLGVN-----
                ++  + +  * + + * + *  +** ++     +  +  + **   *  *    +  ++  ++
  (Evaluation value = 145)
```

Figure 3 Example Alignment 1

```
17.6 : ---------IL-DF------HEKLLHPGIQKTTKL-FGETYYFPNSQLLIQNIINECSICNLAKTEHRNTDMPTKTT-
HTLV : LTDALLITPVL-QL-SPAELHS-FTHCG-Q--TALTLQGA---TTTE--ASNILRSCHAC---RGGNPQHQMP-RGHI
SMRV : ---------ILTALESAQESHA-LHH---QNAAALRFQ--FHITREQ--AREIVKLCPNCP-DWGSAPQLGVN-----
                ++  + +  * *    * * *  +** ++     +  +  + **   *  *    +  ++  ++
  (Evaluation value = 134)
```

Figure 4 Biologically more optimal alignment of Example 1

Example Alignment 2

Example alignment 2 is another example that can be refined into a biologically more optimal alignment when we have biological knowledge. In alignment example 2, in the part of the alignment that corresponds to "HXXH", "H" is not properly aligned but is mis-bridged. Mis-bridged, in this case,

means that "H"s corresponding to the first letter in "HXXH"(in the third sequence) are aligned with "H"s corresponding to the last letters "HXXH"(in the first and second sequences) .

If we fix the mis-bridged "H" and refine the alignment, we cannot identify the zinc finger. However, if we have experience in aligning the sequences of some protein that contains the zinc finger, we can guess the possibility of a mis-bridged "H". We can refine the alignment into an alignment in which the zinc finger is identified. However, because the evaluation value of the alignment is reduced from 161 to 156 when we refine the alignment, we cannot identify the zinc finger without biological knowledge.

```
17.6    : -------ILDF--HE-KLL-HPGIQKTTKLF--GET-YY-FPNSQLLIQNIINECSICNLAKTEHRNTDMPTKTT
M-MULV  : -------LLDF-LHQ---LTHLSFSKMKALLERSHSPYYMLNRDRTL-KNITETCKACAQVNASKSAVKQGTR--
RSV     : VADSQATFQAYPLREAKDL-HTALHIGPRAL--SKA-CN-ISMQQA--REVVQTCPHCNSAPALEAGVN------
                +++ +++ + * *    +   ++  +   ++       +  ++  ++  ** + +    +   +
   (Evaluation value = 161)
```

Figure 5 Example Alignment 2

```
17.6    : -------ILDF--------HEKLLHPGIQKTTKLF--GET-YY-FPNSQLLIQNIINECSICNLAKTEHRNTDMPTKTT
M-MULV  : -------LLDF-------LHQ-LTHLSFSKMKALLERSHSPYYMLNRDRTL-KNITETCKACAQVNASKSAVKQGTR--
RSV     : VADSQATFQAYPLREAKDLHT ALHIGPRAL--SKA-----CN-ISMQQA--REVVQTCPHCNSAPALEAGVN------
                +++       ++  ++++       +   +   ++       +  ++  ++  ** + +    +   +
   (Evaluation value = 156)
```

Figure 6 Biologically more optimal alignment of Example 2

By investigating the two examples above, it is indicated that we can infer the biologically optimal alignment from the biologically optimal alignment. It is also found that if we use only the evaluation value of the alignment as a measure of the alignments, we cannot make the biologically optimal alignment.

3 Multiple alignment system with Intelligent Refiner

3.1 Framework of the alignment system with Intelligent Refiner

Our system consists of two modules, *the aligner* and *the intelligent refiner* (Figure 7). The aligner produces a computationally optimal or near-optimal alignment. The alignment is made without using biological knowledge.

By analyzing the alignment produced by aligner and consulting biological knowledge, the intelligent refiner can roughly understand where conserved column regions are and where another conserved column region may be found in the alignment. The intelligent refiner modifies the alignment in such a way as to increase the conserved column region. The modified alignment in one cycle becomes the input to the next iteration. In each iteration, the intelligent refiner can understand more precisely where conserved regions are and, thus, can gradually identify the conserved column regions. Finally, the biologically near-optimal alignment is produced, if not optimal.

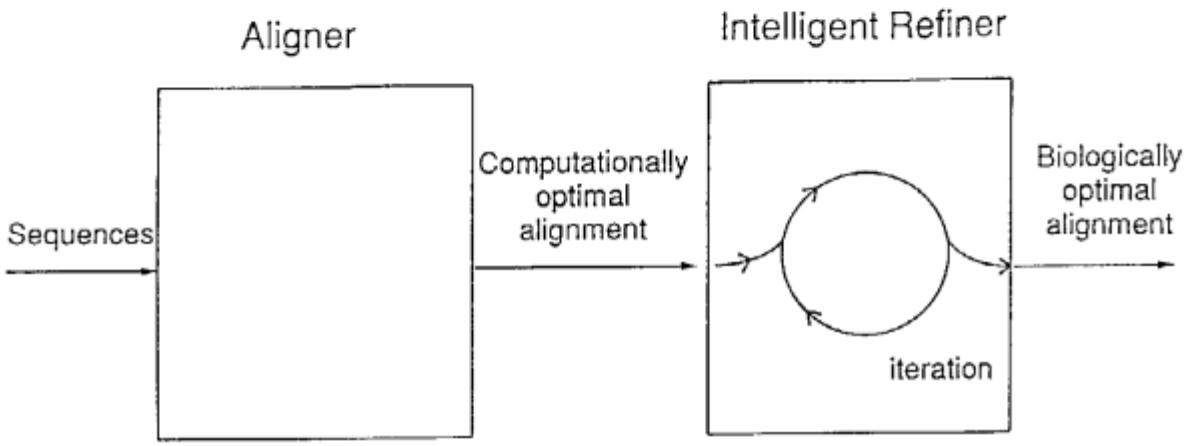

Figure 7 Multiple alignment system with intelligent refiner

Because we can use any alignment system that can produce a computationally optimal or near-optimal alignment as the aligner module, we don't describe the aligner. In the following subsection, the intelligent refiner is explained.

3.2 Intelligent refiner

We designed the intelligent refiner by analyzing the knowledge that experts on multiple alignment use. The program of intelligent refiner is written in Prolog. The structure and function of the intelligent aligner is explained below.

3.2.1 Framework of intelligent refiner

The overall framework of the intelligent refiner is shown in Figure 8. The intelligent refiner is composed of *a control module*, *refinement rule base* and *biological knowledge base*. The refinement rule base is composed of two kinds of rules, *general rules*, and *specific rules*.

The control module iteratively modifies the alignment by calling refinement rules in the refinement rule base to produce the biologically optimal alignment. When specific rules are used, biological knowledge is consulted. In the following, the biological knowledge base and the refinement rule base are explained.

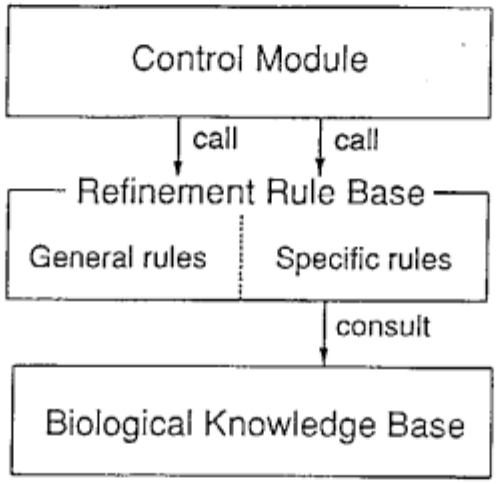

Figure 8 Framework of Intelligent Refiner

Biological knowledge base

In our system, biological knowledge is written in Prolog. In the biological knowledge base, the biological knowledge we extracted from the database, Prosite [Bairoch 1991], and the knowledge we extracted from other materials are contained. Prosite is the database in which knowledge on motifs and related knowledge is written in natural language. We translated a portion of the knowlege in Prosite into Prolog.

In addition to the knowledge we store, biologists can easily input their own knowledge into the biological database, because Prolog has syntax that is similar to natural language.

We show a portion of biological knowledge in Figure 9. The syntax and meanings are explained below. `motif` is a predicate that tells us the meaning of the motif in the third argument (the expression of the motif is explained later). When its first argument is name, the second argument means the name of the motif, for example, zinc finger[(1),(2)]. When its first argument is protein, the second argument means the name of the protein that contains the motif, for example, kinase[(3),(4),(5)].

When we express motif, words joined by "-" are used. The syntax of words are similar to that used in Prosite and defined as follows.

- A word with a single letter signifies an amino acid as usual convention of biology. For example, `C` signifies cysteine[(1),(2)].
- A word with a single letter `X` signifies any amino acid[(1)~(5)].
- When plural amino acids of the same type continue, we specify the length of the amino acids in the argument. For example, `C(4)` is the string of `C` whose length is four. When its length is variable, the minimum length is specified in the first argument and and the maximum length is specified in the second argumentand. For example, `X(3,5)` is the string of `X` whose length is from three to five.
- The letters in parentheses (`[` and `]`) means any amino amino signified by any of the letters. For example `[LIV]` signifies an amino acid signified by `L` or `I` or `V` (`L` signifies leucine, `I` signifies isoleucine and `V` signifies valine)[(3)].

The predicate `function` expresses the function of the specified motif. The second argument of the `function` means the function of the motif in the first argument. The first argument is the motif name or protein name. For example, the function of the motif named the zinc finger is DNA binding[(6)] and the function of the protein named kinase is adding phosphoric acid to other protein[(7)].

The predicate upper_concept expresses the hierarchical relationship of protein. For example, serine_threonine kinase is one class of kinase[(8)], and tyrosine kinase is another class of kinase[(9)].

Expression (10) is the rule that means that even if we don't know the function of a protein, we can infer it if we know the function of the protein which is the upper concept of the protein we are focusing on. Expression (11) is the rule that means that even if we don't know the motif of a protein, we can infer it if we know the motif of the protein which is the upper concept of the protein we are focusing on.

```
motif(name, zinc_finger(normal),  ''C-X(3,5)-C-X(10,25)-H-X(3,5)-H''). (1)
motif(name, zinc_finger(reverse), ''H-X(3,5)-H-X(10,25)-C-X(3,5)-C''). (2)
motif(protein, kinase,''[LIV]-G-X-G-[FY]-[SG]-X-[LIV]''). (3)
motif(protein, kinase(tyrosine),
            ''[LIVMFYC]-X-[HY]-X-D-[LIVMFY]-K-X(2)-N-[LIVMFC](3)''). (4)
motif(protein, kinase(serine,threonine),
          ''[LIVMFYC]-X-[HY]-X-D-[LIVMFY]-RSTA-X(2)-N-[LIVMFC](3)''). (5)

function(zinc_finger, binding(dna)). (6)
function(kinase,add(phosphoric_acid)). (7)

upper_concept(kinase(serine,threonine), kinase). (8)
upper_concept(kinase(tyrosine), kinase). (9)

function(Protein,Function)  :- upper_concept(Protein,UpperProtein),
                               fuction(UpperProtein,Function). (10)

motif(protein,Protein,Motif)  :- upper_concept(Protein,UpperProtein),
                                 motif(protein,UpperProtein,Motif). (11)
```

Figure 9 Biological knowledge base

3.2.2 Refinement rule base

As mentioned before, the refinement rule base is composed of two kinds of rules, the general rules and the specific rules. Rules are expressed using IF-THEN rule. While the general rules don't consult the biological knowledge base, the specific rules consult the biological knowledge base.

We extracted more than twenty rules from alignment experts. Of these, fifteen rules are currently used in the intelligent refiner. Here we show representative rules only. Four representative general rules are shown in Figure 10. Four representative specific rules are shown in Figure 11. The rules will be explained using examples later.

In the rules, several routines are called. However, because their functions are clear from the context, we will not explain the routine further.

General rule 1 (see Figure 12)

IF An half conserved column c_i , in which more than specified percentage(e.g. 80 %) of whose elements are identical amino acids (x_s), is found. AND

In the sequence that doesn't have x_s in the column, x_s are sought within specified distance from c_i (checked by `search routine 1`).

THEN The modified alignment is produced in the constraint that the found x_s is aligned in the column c_i (done by `modification routine`). When plural alternatives are generated, the modified alignment whose evaluation value is the highest is selected.

General rule 2 (see Figure 13)

IF Sequences in the alignment are grouped in to two groups, g_i and g_j according to similarities between the sequences (checked by `grouping routine`). AND

Patterns of conserved columns (p_i and p_j) in each group of alignment (a_i and a_j) are found AND

Common sub-pattern (p_{ij}) is found in the both patterns (p_i and p_j) (done by `search routine 2`)

THEN a_i and a_j are aligned in the constraint that p_{ij} in a_i and p_{ij} in a_j are aligned (done by `modification routine`).

General rule 3 (see Figure 14)

IF p (conserved column pattern composed of more than a specified length of columns(e.g. length of three)) is identified.

THEN The portion of alignment corresponding to p is temporarily fixed, and the alignment is divided into two parts a_s and a_t, a_s is the left side of p and a_t is the right side of p (`refinement routine` will be called respectively, and the resultant alignments and alignment corresponding to p are integrated by `integration routine 1`).

General rule 4 (see Figure 15)

IF The `refinement routine` cannot identify any conserved column in alignment a any more AND

Sequences in the alignment a are grouped in to two groups, g_i and g_j, according similarities between sequences (checked by `grouping routine`).

THEN The alignment a is divided into two groups, a_i and a_j, each of which corresponds to g_i and g_j (And `refinement routine` will be called respectively and resultant alignments are integrated into one alignment by `integration routine 2`)

Figure 10 Representative rules of general rules

Specific rule 1 (see Figure 16)

IF Discovered conserved column pattern p contains some part of an motif m_i stored in the biological knowledge base (done by `motif_check routine`).

THEN the `Motif_finding routine` is called to identify the other part of the motif. (`motif_finding routine` will call (`general rule 1`, `general rule 2` or `specific rule 2` and so on)).

Specific rule 2 (see Figure 16)

IF An motif (that `motif_finding routine` trys to identify) have two amino acids, x_i and x_j, of same kind of amino acid x AND

The motif has no other conserved amino acids between x_i and x_j AND

There are a half conserved column of c_i of x and a conserved column of c_j of x (done by `motif_finding routine`).

THEN `Modification routine` is called to produce alignment in the constraint that $x_{j,t}$ (belonging to the the sequence s_t that doesn't have x in the half conserved column c_i) is aligned in the half conserved column c_i.

specific rule 3

IF A motif (m_i) is identified in the alignment AND

The protein that has the motif (m_i) have another motif m_j (checked by `knowledge_consulting routine`)

THEN the `motif_finding routine` is called to identify the motif m_i

Specific rule 4

IF An motif m_i is identified.

THEN The portion of alignment corresponding to m_i is temporarily fixed. And the alignment is divided into two parts a_s and a_t, a_s is the left sideof m_i and a_t is the right side of m_i (And `refinement routine` will are called in respectively and resultant alignments and the alignment corresponding to m_i are integrated by `integration routine 1`).

Figure 11 Representative rules of specific rules

Example application of general rules

Four general rules are explained using examples. Figure 12 is an example of application of general rule 1. By `search routine 1` a half conserved column of "A" (there is an exception in the first sequence) is found ("_" in Figure 12 signifies any character) and in the first sequence, an "A" is found in the neighborhood of the half conserved column. Then, general rule 1 is fired. The `modification routine` is called to produce a conserved column of "A".

Figure 12 Example application of general rule 1
"_" means any character

Figure 13 is an example of application of general rule 2. In the example, the sequences are decomposed into two groups, the first three sequences and the last three sequences according to the similarities of the sequences (checked by grouping routine)

By search routine 1, it is found that a conserved column pattern "AP" is found in both groups. Then, general rule 2 is fired. The modification routine is called to produce a conserved column of "AP" that extends all sequences.

Figure 13 Example application of general rule 2
The sequences can be decomposed into two groups, the first three sequences and the last three sequences, according to the similarities between the sequences

The temporary alignment produced by general rule 1 or general rule 2 is sent to the evaluation routine with the current alignment (the most biologically optimal alignment at the time). In the routine, the current alignment and the temporary alignment are evaluated by consulting the biological knowledge base. Then, the new current alignment is selected.

Figure 14 is an example of application of general rule 3. In the alignment, there is conserved column pattern "HSPA" and the length of conserved column pattern is more than the specified length (e.g. 3) in the rule. Then, general rule 3 is fired. The the sub-alignment corresponds to "HSPA" is temporarily fixed (the refinement routine will be called in sub-alignment 1 and 2, and the refined sub-alignments and the sub-alignment corresponding to "HSPA" will be integrated by integration routine 1

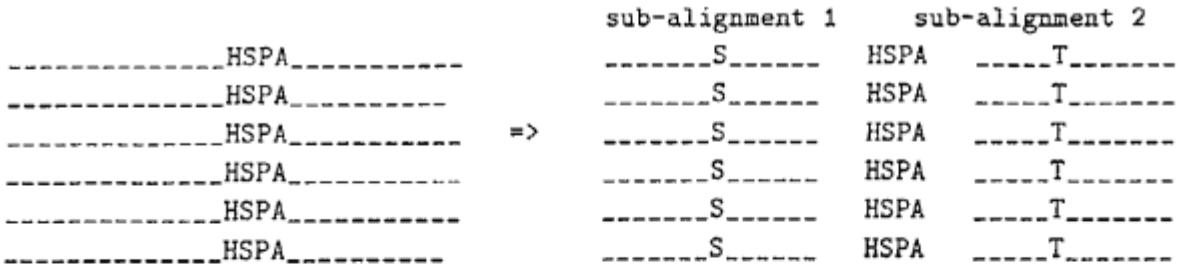

Figure 14 Example application of general rule 3

Figure 15 is an example of application of general rule 4. General rule 4 is used when no other rules can be applied. The sequences are decomposed into two groups, the first three sequences and the last three sequences according to the similarities of the sequences (checked by grouping routine). In the sub-alignment between "MMM" and "III", the refiner routine cannot find any conserved column extending all sequences .

Then, general rule 4 is fired. The sub-alignment is decomposed into two sub-alignments, the sub-alignment belonging to group 1 and that belonging to group 2 (each sub-alignment will be refined by the refinement routine and the two refined alignments are integrated by the integration routine 2).

Procedures corresponding to general rule 1 and general rule 3 were incorporated in the alignment program written by [Butler *et al.* 1990]

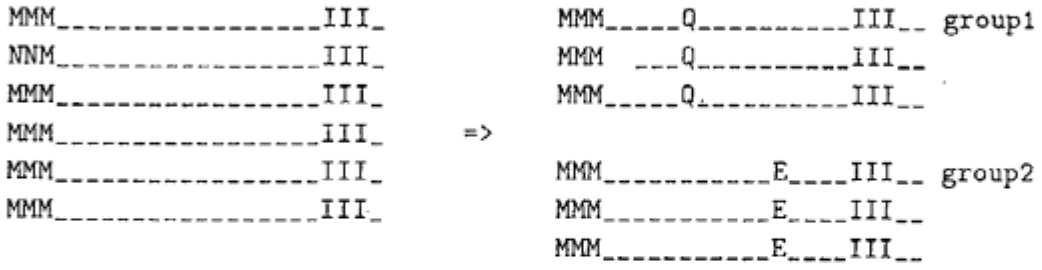

Figure 15 Example application of general rule 4

Example application of specific rules

Example application of specific rule 1, specific rule 2 and specific rule 3 will be explained using examples. Because specific rule 4 is similar to general rule 4, the rule will not be explained here.

Specific rule 1 and specific rule 2

An example application of specific rule 1 and specific rule 2 is explained by using Figure 16. The `motif_check routine` identifies that the conserved column of "CXXXC" is the latter part of zinc finger (reverse type) motif "H-X(3,5)-H-X(10,25)-C-X(3,5)-C" by consulting the `knowledge (2)` in the biological database. Then, specific rule 1 is fired. The `motif_finding routine` finds the half conserved pattern of "HXXXH" in which the the latter column of "H" is half conserved. Then, specific rule 2 is fired. The `modification routine` makes the alignment which has a conserved column pattern of "H-X(3)-H-X(15)-C-X(3)-C"

Figure 16 Example application of specific rule 1 and specific rule 2

Specific rule 3

The example application of specific rule 3 is explained by using the sequences belonging to a protein called tyrosine kinase. When the rule is applied, the `knowledge (3)(4)(9)(11)` in the biological knowledge base are consulted.

If motif "[LIVMFYC]-X-[HY]-X-D-[LIVMFY]-K-X(2)-N-[LIVMFC](3)" is identified in the alignment, the `knowledge_consulting routine` finds that it is the motif of tyrosine kinase (`expression (4)` in the biological database). Then, other motifs belonging to tyrosine kinase are sought in the biological knowledge base. The corresponding motifs are not stored explicitly in the biological database in this case. However, using inference rule (`expression (11)`), and the knowledge that kinase is the upper concept of tyrosine kinase (`expression (9)`), it is derived that tyrosine kinase also has a motif "[LIV]-G-X-G-[FY]-[SG]-X-[LIV]".

Then, specific rule 3 is fired, and the `Motif_finding routine` is called to identify "'[LIV]-G-X-G-[FY]-[SG]-X-[LIV]".

4 Example of Application

Two simple examples of application of the intelligent refiner to near-optimal alignment are shown in this section. The alignment to be aligned was produced by the iterative improvement algorithm [Ishikawa 1992].

Example 1

Application example 1 is the alignment of sequences from protein kinase (Figure 16). The upper three sequences are similar and the lower five sequences are similar. So, sequences can be grouped into two groups.

Firstly, in conserved columns are investigated and three `G`s, single `V` and single `E` are identified. From these conserved columns, "GXGXXGXV" is recognized as the conserved column pattern.

Secondly, by consulting the biological knowledge base, the pattern is identified as the motif of kinase. Then, specific rule 4 is fired. The alignment corresponding to the pattern is temporarily fixed. In sub-alignment on the right side of the pattern, conserved columns are sought (Figure 17).

Thirdly, conserved column patterns are sought in the two groups of sequences (the upper three and the lower three sequences). Then, the patterns of "VAXKXLK"(the upper group) and "AXK"(the lower group) are identified. Because the both groups have "AXK" as a sub-pattern of conserved columns, general rule 2 is fired and the alignment that has a conserved column of "AXK" is produced(Figure 18).

Finally, the right side of the alignment in Figure 16 is replaced by the alignment in Figure 18. Then, refinement of the alignment is completed (Figure 19).

Although, the alignment in Figure 19 is a computationally less optimal alignment because the evaluation value is 1582 which is worse than 1593 (the evaluation value before refinement). However,

because the refined alignment identifies the conserved pattern "AXK", the alignment is biologically more optimal.

```
Sequence1:  -IGEGEFGEVYRGT----LR---LPSQDCKTVAIKTLKDTSPGGQWWNFLREATIMGQF---SHPHI
Sequence2:  --GEGCFGQVVLAEAIGLDK---DKPNRVTKVAVKMLKSDATEKDLSDLISEMEMMKMI--GKHKNI
Sequence3:  --GEGEFGKVVKATA--FHL---KGRAGYTTVAVKMLKENASPSELRDLLSEFNVLKQV---NHPHV
Sequence4:  VIGKGSFGKVMQVR----KK---DTQKVYALKAIRK-SYIVSKSEVTHTLAERTVLARV---DCPFI
Sequence5:  -LGKGGYGKVFQVR----KVTGANTGKIFAMKVLKKAMIVRNAKDTAHTKAERNILEEV---KHPFI
Sequence6:  TLGTGSFGRVMLVK----HK---ETGNHYAMKILDK-QKVVKLKQIEHTLNEKRILQAV---NFPFL
Sequence7:  IIGRGGFGEVYGCR----KA---DTGKMYAMKCLDK-KRIKMKQGETLALNERIMLSLVSTGDCPFI
Sequence8:  ELGKGAFGVVRRCV----KV---LAGQEYAAKIINT-KKL-SARDHQKLEREARICRLL
              G G  G V                                          E
(Evaluation Value = 1593)
```

Figure 16 Input alignment to the intelligent refiner

```
                                va k lk              E
Sequence1:  YRGT----LR---LPSQDCKTVAIKTLKDTSPGGQWWNFLREATIMGQF---SHPHI  Upper
Sequence2:  VLAEAIGLDK---DKPNRVTKVAVKMLKSDATEKDLSDLISEMEMMKMI--GKHKNI  group
Sequence3:  VKATA--FHL---KGRAGYTTVAVKMLKENASPSELRDLLSEFNVLKQV---NHPHV  --------
Sequence4:  MQVR----KK---DTQKVYALKAIRK-SYIVSKSEVTHTLAERTVLARV---DCPFI
Sequence5:  FQVR----KVTGANTGKIFAMKVLKKAMIVRNAKDTAHTKAERNILEEV---KHPFI  Lower
Sequence6:  MLVK----HK---ETGNHYAMKILDK-QKVVKLKQIEHTLNEKRILQAV---NFPFL  group
Sequence7:  YGCR----KA---DTGKMYAMKCLDK-KRIKMKQGETLALNERIMLSLVSTGDCPFI
Sequence8:  RRCV----KV---LAGQEYAAKIINT-KKL-SARDHQKLEREARICRLL
                              a k                  E
(Evaluation Value = 719)
```

Figure 17 Sub-alignment to refine

```
Sequence1:  YRGT----LRLPS---QDCKTVAIKTLKD-TS--PGGQWWNFLREATIMGQF---SHPHI
Sequence2:  VLAEAIGLDKDKP---NRVTKVAVKMLKS-DA--TEKDLSDLISEMEMMKMI--GKHKNI
Sequence3:  VKATA--FHLKGR---AGYTTVAVKMLKE-NA--SPSELRDLLSEFNVLKQV---NHPHV
Sequence4:  MQV-------RKK---DTQKVYALKAIRK-SYIVSKSEVTHTLAERTVLARV---DCPFI
Sequence5:  FQV-------RKVTGANTGKIFAMKVLKKAMIVRNAKDTAHTKAERNILEEV---KHPFI
Sequence6:  MLV-------KHK---ETGNHYAMKILDK-QKVVKLKQIEHTLNEKRILQAV---NFPFL
Sequence7:  YGC-------RKA---DTGKMYAMKCLDK-KRIKMKQGETLALNERIMLSLVSTGDCPFI
Sequence8:  RRC-------VKV---LAGQEYAAKIINT-KKL-SARDHQKLEREARICRLL---KHPNI
                              A K                  E
(Evaluation Value = 708)
```

Figure 18 Sub-alignment after refinement

```
Sequence1:  -IGEGEFGEVYRGT----LRLPS---QDCKTVAIKTLKD-TS--PGGQWWNFLREATIMGQF---SHPHI
Sequence2:  --GEGCFGQVVLAEAIGLDKDKP---NRVTKVAVKMLKS-DA--TEKDLSDLISEMEMMKMI--GKHKNI
Sequence3:  --GEGEFGKVVKATA--FHLKGR---AGYTTVAVKMLKE-NA--SPSELRDLLSEFNVLKQV---NHPHV
Sequence4:  VIGKGSFGKVMQV-------RKK---DTQKVYALKAIRK-SYIVSKSEVTHTLAERTVLARV---DCPFI
Sequence5:  -LGKGGYGKVFQV-------RKVTGANTGKIFAMKVLKKAMIVRNAKDTAHTKAERNILEEV---KHPFI
Sequence6:  TLGTGSFGRVMLV-------KHK---ETGNHYAMKILDK-QKVVKLKQIEHTLNEKRILQAV---NFPFL
Sequence7:  IIGRGGFGEVYGC-------RKA---DTGKMYAMKCLDK-KRIKMKQGETLALNERIMLSLVSTGDCPFI
Sequence8:  ELGKGAFGVVRRC-------VKV---LAGQEYAAKIINT-KKL-SARDHQKLEREARICRLL---KHPNI
              G G  G V                          A K
(Evaluation Value = 1582)
```

Figure 19 Result of refinement

Example 2

The refinement of an alignment whose sequences are from retro-virus is briefly shown below. In the alignment (Figure 20), the half conserved column pattern of "HXXXH" and conserved column pattern of "CXXC" are found. Both patterns constitute the zinc finger motif. This is the similar case with the alignment in Figure 16. Similarly, specific rule 1 and specific rule 2 are applied to this case. Then, a refined alignment (Figure 21) is produced. The refined alignment has the conserved column pattern of the zinc finger.

The evaluation value is reduced from 654 to 619. This means that biologically more optimal alignment (after the refinement) is computationally less optimal than the alignment before refinement.

```
Sequence1 : ILDFHEKLLHPGIQKTTKLFGET--------YYFPNSQLLIQNIINECSIC-NLAKTEHRNTDMPTK
Sequence2 : LLDF----LHQLTHLSFSKMKALLERSHSPYYMLNRDRTL-KNITETCKAC-AQVNASKSAVKQGTR
Sequence3 : VLQLSPAELHSFTHCGQTALTLQ---------GATTTEA--SNILRSCHAC--RGGNPQHQMPRGHI
Sequence4 : PLR-EAKDLHTALHIGPRALSKA--------CNISMQQA--REVVQTCPHC-NSAPALEAGVN----
Sequence5 : PIH-EATQAHTLHHLNAHTLRLL--------YKITREQA--RDIVKACKQCVVATPVPHLGVN----
Sequence6 : ALE-SAQESHALHHQNAAALRFQ--------FHITREQA--REIVKLCPNCPDWGSAPQLGVN----
(Evaluation value = 654)
```

Figure 20 Example of refinement [before refinement]

```
Sequence1 : -----ILD-FHEKLLHPGIQKTTKLFG---ETYY-FPNSQLLIQNIINECSICNLAKTEHR-NTDMPTK
Sequence2 : -----LLDFLHQ-LTHLSFSKMKALLERSHSPYYMLNRDRTL-KNITETCKACAQVNASKS-AVKQGTR
Sequence3 : VLQLSPAE-LHS-FTHCGQTALTLQ-G--------ATTTEA--SNILRSCHACRGGNPQHQMPRGHI--
Sequence4 : PLR-EAKD-LHT-ALHIGPRALSKACN--------ISMQQA--REVVQTCPHCNSAPALEA-GVN----
Sequence5 : PIH-EATQ-AHT-LHHLNAHTLRLLYK--------ITREQA--RDIVKACKQCVVATPVPHLGVN----
Sequence6 : ALE-SAQE-SHA-LHHQNAAALRFQFH--------ITREQA--REIVKLCPNCPDWGSAPQLGVN----
(Ealuation value = 619)
```

Figure 21 Example of refinement(retro-virus sequences)[after refinement]

5 Discussion

1. [Butler *et al.* 1990] employed Strand, a logic programming language, to write a multiple alignment program. Their program is applied to sequences that are not aligned at all. The program use procedures like general rule 1 and general rule 3 to identify the conserved column iteratively.

The program has two weak points. One is that, because the program is applied to the sequences that are not aligned, it often produces spurious conserved columns. The other is that once some spurious conserved column is produced, the spurious conserved column is fixed and is never revised.

The alignment system with intelligent refiner solves the problem as follows and the reliability of the alignment the system produces is higher.

- Our system, firstly, generate computationally near-optimal alignment and the system refines the computationally near-optimal alignment using knowledge. The computationally near-optimal alignment doesn't necessarily correspond to the biological optimal alignment. However, we can get information on where possible conserved columns are. Then, we can gradually increase the reliability of the information by iteratively refining the alignment.
- Because we have knowledge on the motif in the biological knowledge base, our system can identify the motif with high reliability. For example, with biological knowledge, the spurious column conserved of "E"(that blocks the alignment of "H") in the alignment in Figure 3 is remedied to securely identify the motif.
- Because we have alignment rules that break spurious conserved column (e.g. specific rule 2), the risk that the alignment is trapped by spurious conserved columns is reduced.

2. Conceptually, it is better to input computationally optimal or near-optimal alignments into the intelligent refiner. However, it is possible for biologists to input the alignment roughly made by the hand or the alignment produced by the tree base algorithm. Although, the quality of the alignment that the intelligent refiner produces when the biologists use these alignments as input is worse than the case when the computationally optimal alignment is used as input, the quality of the resultant alignment is tolerable for practical use.

3. The alignment system with intelligent refiner is not only for biologists but also for computer

scientists. By analyzing the resultant alignment produced by intelligent refiner which contains plenty of biological knowledge and refinement rules. the possibility of making biological discoveries will be opened.

4. Because the intelligent refiner is still at a primitive level, we must continue research to improve the power of the system. The quality of the produced alignment that is determined by the amount and quality of biological knowledge and refinement rules. We must increase the biological knowledge and extract effective rules more form experts on multiple alignment to improve intelligent refiner.

5. There are refinement rules that we extracted from biologists but we haven't incorporated into the intelligent refiner yet. One of the knowledge is on the alignments that biologists don't favor. One class of these unfavored alignments is the alignment with islands phenomenon. The phenomenon is shown in Figure 21 (left), every amino acid is expressed by "X" or "O". There is an island composed of amino acids in an ocean composed of gap letters "-". The island is composed of amino acids from the sequence $1 \sim 4$. Alignment experts think that the ocean of gaps should be reduced to make compact alignment like alignment on the right of the figure. We are now investigating how to recognize the phenomenon and how to remedy it.

```
Sequence1:   XXXXXXXXXX---O---XXXXXXXX          XXXXXXXXXXOXXXXXXXX
Sequence2:   XXXXXXXXXX--OOO-----XXXXX          XXXXXXXXXXOOO-XXXXX
Sequence3:   XXXXXXXXXX--OOOO----XXXXX    =>    XXXXXXXXXXOOOOXXXXX
Sequence4:   XXXXXXXXXX-----OOO--XXXXX          XXXXXXXXXX-OOOXXXXX
Sequence5:   XXXXXXXXXXXXX-------XXXXX          XXXXXXXXXXXXX-XXXXX
```

Figure 21 An example of unfavored alignment : Island phenomenon and its remedied alignment.

Acknowledgment

The authors acknowledge R.Tanaka and Y.Totoki of IMS for their programming and experimental effort.

We would like to especially thank K.Kuma, N.Iwabe of Kyoto Univ. for their willingness to answer our questions when they showed us their actual alignment process. Without their encouragement, our research on multiple alignment couldn't have done. We also acknowledge T.Miyata and H.Hayashida of Kyoto University, H.Toh of PERI and George Michaels of NIH for their discussion on multiple alignment.

References

[Bairoch 1991] Bairoch,A. Prosite : A dictionary of protein site ans pattern : User manual Release 7.00, May 1991.

[Berger and Manson 1991] Berger,M. and Manson,P.(1991) A novel randomized iterative strategy for aligning multiple protein sequences. *Computer Application in the Biosciences, 7*, 1991. pp.479-484.

[Barton 1990] Barton,J.G. (1990) Protein Multiple Alignment and Flexible Pattern Matching. in *Methods in Enzymology Vol.183*, Academic Press, 626-645.

[Butler *et al.* 1990] Butler,R., ,Butler,T., Foster,I., Karonis,N., Olson,R., Overbeek,R., Pfluger,N., Price.M. and Tuecke,S(1990). Aligning Genetic Sequences *in Foster,I. and Taylor,S. Strand – New concept in parallel programming.* Prentice Hall.

[Carrillo and Lipman 1988] H. Carrillo and D. Lipman. The multiple sequence alignment problem in biology. *SIAM J. Appl. Math.*, 48, 1988, pp.1073–1082.

[Dayhoff,O. *et al.* 1978] Dayhoff,M.O., Schwatz,R.M. and Orcutt,B.C. (1978) A model of evolutionary change in proteins. In Dayhoff,M.O.(ed), *Atlas of Protein Sequence and Structure Vol.5, Suppl.3,* Nat. Biomed. Res. Found., Washington, D. C., 363–373.

[Hirosawa *et al.* 1991] Hirosawa,M., Hoshida,M., Ishikawa,M. and Toya,T. (1991) Multiple Alignment System for Protein Sequences employing 3-dimensional Dynamic Programming. *Genome Informatics Workshop II,* (in Japanese).

[Ishikawa *et al.* 1991a] Ishikawa,M., Hoshida,M., Hirosawa,M., Toya,T., Onizuka,K. and Nitta,K. (1991a) Protein Sequence Analysis by Parallel Inference Machine. *Information Processing Society of Japan, TR-FI-23-2,* (in Japanese).

[Johnson and Dolittle 1986] M. S. Johnson and R. F. Doolittle. A method for the simultaneous alignment of three or more amino acids sequences. *J. of Mol. Evol.,* 23, 1986, pp.267–278.

[Ishikawa *et al.* 1991] Ishikawa,M., Toya,T., Hoshida,M., Nitta,K., Ogiwara,A. and Kanehisa,M. (1991b) Multiple Alignment by Parallel Simulated Annealing. *Genome Informatics Workshop II,* (in Japanese).

[Ishikawa *et al.* 1992] Ishikawa,M., Hoshida,M., Hirosawa,M., Toya,T. and Nitta,K. (1992) Protein Sequence Analysis by Parallel Inferrence Machine. *Proc. Int. Conf. on Fifth Generation Computer Systems 1992.*

[Murata 1985] Murata,M. (1985) Simulteneous comparison of three protein sequences *Proc. Natl. Acad. Sci. USA Vol. 32,* 1985, pp.3073-3077.

[Needleman and Wunsch 1970] Needleman,S.B. and Wunsch,C.D. (1970) A General Method Applicable to the Search for Similarities in the Amino Acid Sequences of Two Proteins. *J. of Mol. Biol.,* 48, 443-453.