A Linear Programming Based Algorithm for Multiple Sequence Alignment by Using Markov Decision Process

Department of Computer Science
University of Maryland, College Park

Shirin Mehraban
Advisors: Dr. Fern Hunt and Dr. Chau-Wen Tseng
May 12, 2004

Table of Contents

1. Abstract

Bioinformatics is a scientific field that combines biology and information technology for the purpose of developing tools that identify genes and proteins for medical and biotechnology applications. In the last decade, there has been an enormous increase in the study of biological sequences. Analyzing biological sequences requires matching a set of protein sequences composed of unknown properties with the protein sequences whose properties are well known. This project proposes a promising algorithm that offers an alternative approach to the problematic process of aligning and matching longer biological sequences.

2. Introduction

Bioinformatics is “the field of science in which biology, computer science, and information technology merge to form a single discipline. The ultimate goal of the field is to enable the discovery of new biological insights as well as to create a global perspective from which unifying principles in biology can be discerned.” [1] There are several objects of interest in bioinformatics such as the Genome sequences (DNA and protein sequences) and the Macromolecular structures (a structure in which all of the atoms are linked by chemical bonds as a unit). One of the important tools in bioinformatics is sequence alignments. A variation of the sequence alignments is called the multiple sequence alignments. The multiple sequence alignments identify the conserved regions of the biological sequence that can have major medical or biological importance. The theory of evolution shows that all living creatures are descended from a common ancestor by a process known as mutation. The principal non-experimental method of tracing biological history is an example of multiple sequence alignment. Sequence alignment is a method of writing one sequence on top of another in a way that shows how mutation occurs during evolution. These changes are known as *insertions* (inserting amino acids to the second sequence) and *deletions* (the process in which nucleotide pairs are removed from a gene which is denoted with a gap). In our work different combinations of insertions and deletions are called *actions*.

3. Background

The methods used to align biological sequences are based on solving a minimization problem (cost) or a maximization problem (score). In this project, a Scoring Markov decision model is used to calculate the probability matrix of the alignment. The resulting matrix is used to solve a minimization or a maximization problem. A Markov chain is a random process consisting of states that can change with time. One of the Markov Chain's properties is that it is memory-less, meaning at time n+1 it has no memory of the previous states at time 0, 1, ..., n-1. In a Markov chain, the probability of traveling from one state to another is important. Thus, a matrix that consists of these probabilities is called Markov chain matrix where each row and column is labeled with state numbers.

Two properties are extremely important in building the Markov chain. The first is that the probability of traveling from one state to another should be greater or equal to zero. The second is that the summation of probabilities of traveling from one state to all other states should be equal to one. These properties can be formulated as shown in below.

Markov Chain properties:

- $P_{ij} \geq 0$ (1)
- $\sum_{j=1}^{n} P_{ij} = 1 \qquad i = 1, \ldots, m$

The method of aligning sequences used in the algorithm uses Linear Programming. Linear Programming is a method to minimize or maximize a linear function of one or more variables. The solution must satisfy certain equations or inequalities called constraints. We obtain alignments by solving a linear programming problem derived from the Markov chain decision model. The Markov decision model is a process used to define the next state's *action* in a particular time. After the observation of the state of the process, an *action* must be chosen for the next state. Moreover, if the process is in state i at time n and *action a* is chosen, then the next state of the process is determined by the probability of the present state and subsequent action as shown in equation (2).

$$P_{ij}(a) = \text{Pr ob}\{ \mathrm{X}_{\mathrm{n}+1} = j \text{ given } \mathrm{X}_{\mathrm{n}} = i \text{ and } \mathrm{a}_{\mathrm{n}} = a \} \tag{2}$$

Furthermore for each action *a*, the two properties stated in equation (3) and (4) should hold.

$$P_{ij}(a) \geq 0 \tag{3}$$

$$\sum_{j=1}^{m} P_{ij}(a) = 1 \tag{4}$$

4. Algorithm and Implementation

The algorithm proposed in this paper is based on the Markov decision model which creates probability matrices used to set up the linear programming problem. Because we focus on aligning three sequences, there are eight different possibilities of combining gaps and letters. A gap is represented as ‘0’ and letter is represented as ‘1’. Moreover, these different possibilities are numbered for ease of reference to the *actions* as shown in Figure 1.

$$\begin{array}{cccccccccc}
 & & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
 & & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
 & & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
a & = & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8
\end{array}$$

Figure 1

Each column of the aligned sequence represents a state and is defined by a number as shown in Figure 2.

$$\begin{array}{cccccccccccccccc}
 & - & M & S & Q & L & L & L & - & W & L & G & L & W & S & - \\
 & - & L & G & H & - & G & L & V & T & P & T & E & Q & G & V \\
 & - & L & G & H & - & G & L & V & T & - & T & E & Q & G & V \\
\text{State Number :} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 3 & 8
\end{array}$$

Figure 2

For example, assume that we want to align the sequences stated in figure 3.

$$\begin{array}{ccccccccc}
- & M & P & L & L & Q & G & V & L \\
- & - & - & - & - & - & - & M & P \\
- & - & - & - & - & - & - & - & M
\end{array}$$

Figure 3

The first step is to assign the state number to each column of our sequence. Figure 4 shows the assigned states for the sequences in Figure 3.

$$\begin{array}{ccccccccc}
- & M & P & L & L & Q & G & V & L \\
- & - & - & - & - & - & - & M & P \\
- & - & - & - & - & - & - & - & M \\
1 & 2 & 3 & 4 & 4 & 5 & 6 & 7 & 8
\end{array}$$

Figure 4

Each state represents an action, which is defined by the algorithm. For the example mentioned in Figure 3 the algorithm defines the *actions* of each state as described in Figure 5.

$$\begin{array}{ccccccccc}
- & M & P & L & L & Q & G & V & L \\
- & - & - & - & - & - & - & M & P \\
- & - & - & - & - & - & - & - & M \\
1 & 2 & 2 & 2 & 2 & 2 & 2 & 4 & 8
\end{array}$$

Figure 5

The following formula calculates the frequency of transition from a particular state to other *actions*.

$P_{ij}(a) = \dfrac{n(i,j,a)}{n(i,a)}$, where a denotes the *action*, i represents the initial state, and j is defined as the final state.

In case of the example shown in Figure 3 above, the algorithm creates a matrix shown in Figure 6, where includes i, j, a, P_{ij}, and $P_{ij}(a)$ which is calculated using the above formula..

i	j	P_{ij}	$P_{ij}(a)$	a
1	2	1	1	2
2	3	1	1	2
3	4	1	1	2
4	4	1	0.5	2
4	5	1	0.5	2
5	6	1	1	2
6	7	1	1	4
7	8	1	1	8

Figure 6

These frequencies are stored in a matrix and will be used to solve a maximization linear function as shown in equations (5) and (6) below.

$$\text{maximize} \quad \sum_j \sum_a y_{ja} R(j, a) \tag{5}$$

Constraints :

$$\begin{aligned} &1.\quad \sum_j \sum_a y_{ja} = \frac{1}{1-\alpha} \\ &2.\quad \sum_j y_{ja} = b_j + \alpha \sum_i \sum_a y_{ia} P_{ij}(a) \end{aligned} \tag{6}$$

$$y_{ja} \geq 0, \qquad all\ j, a$$

y_{ja} measures how often from state j one should choose action a when there is a maximum total scoreMI b_j MA the initial distribution of states in the Markov Chain is set to

1/m where m is the number of states in the sequence. $P_{ij}(a)$ is the calculated frequency in Markov decision model associated with action a. $R(j,a)$ represents the score of transition from state *j* to another state with action a.

As observed in the equations (5) and (6) above, the linear programming function consists of two constraints. However, the first constraint can be derived from the second constraint. Thus, it can be assumed that the problem has only one independent constraint.

To solve this linear programming, we are required to produce a matrix for $P_{ij}(a)$ MI This matrix can also be denoted as accumulation of different frequencies for each action as shown in Figure 7.

$$P_{ij}(a) = \begin{bmatrix} P_{ij}(1) \\ P_{ij}(2) \\ P_{ij}(3) \\ P_{ij}(4) \\ P_{ij}(5) \\ P_{ij}(6) \\ P_{ij}(7) \\ P_{ij}(8) \end{bmatrix}$$

Figure 7

Therefore, in the above example, the algorithm creates a $P_{ij}(a)$ matrix with the result shown in figure 6. Figure 8 represents part of the $P_{ij}(a)$ for the previous example.

$$P_{ij}(2) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 8

As stated above, b_j MA the initial distribution of states in the Markov Chain is set to 1/m where m is the number of states in the sequence. In the above example the number of states is 20. Figure 9 shows the initial distribution.

$$b_j = \begin{bmatrix} 0.125 \\ 0.125 \\ 0.125 \\ 0.125 \\ 0.125 \\ 0.125 \\ 0.125 \\ 0.125 \end{bmatrix}$$

Figure 9

There are two different types of sequences used for testing the algorithm. The first sequences uses letters and gaps as shown in Figure 10.

$$\begin{array}{ccccccccccccccccccccccccccccccc} - & - & - & - & - & - & - & - & - & M & E & L & T & E & A & F & L & T & L & H & W & G & L & P & R & L & H & H & L & L & A \\ - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & M & W & N & A & L & V & W & Q & Q & V & A & A \\ - & - & - & - & - & - & - & - & - & - & - & M & A & L & S & F & L & S & P & S & L & S & R & L & G & L & W & - & - & A & S \end{array}$$

Figure 10

According to the type of sequence that is being processed, different score values are calculated. The score of a state is not permanent; however, it depends on a transition to different actions.

In the set of data shown in Figure 10, a pair of sequences is selected and then the score from Blosum62 (which is a generally accepted table for scoring pairs of amino acids) is selected. According to the action, if there is a deletion, a deletion penalty, d would be deducted, and if there is an extension with gaps, an extension penalty, e, would be deducted. The following procedure is repeated for each sequence, and the result of the following calculation is divided by the number of sequences that are being aligned as shown in equation (7) and (8).

Example: Alignment of three sequences

$$R\left(\begin{pmatrix} - \\ - \\ M \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}\right) = [\mathrm{Bl}\,(-, -) - e - e + \mathrm{Bl}\,(-, M) - e - d + \mathrm{Bl}\,(-, M) - e - d\,]/3 \tag{7}$$

$$R\left(\begin{pmatrix} M \\ M \\ M \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}\right) = [\,\mathrm{Bl}\,(M, M) - d + \mathrm{Bl}\,(M, M) + \mathrm{Bl}\,(M, M) - d]/3 \tag{8}$$

The second set of data contains letters, gaps, and question marks as shown in Figure 11. The question mark symbol in a sequence represents the effect of an insertion or deletion at the beginning of a DNA sequence. This effects the translation into protein at those positions.

$$\begin{array}{cccccccccccccccccc}
- & - & V & T & K & A & A & Q & K & A & A & K & K & ? & ? & ? & ? & ? \\
- & - & V & T & K & S & A & Q & K & A & Q & K & A & G & K & ? & ? & ? \\
- & - & V & T & K & S & A & V & K & A & G & K & K & ? & ? & ? & ? & ?
\end{array}$$

Figure 11

The score for data with a question mark is calculated with equations (9), (10), and (11), where f is the penalty of having question mark.

$$\begin{aligned}
&C(\text{ letter}, ?) = Bl(\text{ letter}, -) - f \\
&C(-, ?) = -(f + d) \qquad (9) \\
&C(?, ?) = -2f
\end{aligned}$$

Example:

$$R(\begin{pmatrix} M \\ ? \\ M \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}) = [\, Bl(M, -) - f - e + Bl(M, M) + Bl(-, M) - f - e \,]/3 \qquad (10)$$

$$R(\begin{pmatrix} ? \\ - \\ ? \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}) = [\, C(?, ?) - d - d + C(-, ?) - d - e + C(-, ?) - d - e \,]/3 \qquad (11)$$

This algorithm uses equations (7), and (8) to calculate the scores of each state. The score results for *action* 1 using the sequences in Figure 3 are represented in Figure 12 below.

$$R(j,1) = \begin{bmatrix} -36.333 \\ -38.667 \\ -38.667 \\ -38.667 \\ -38.667 \\ -38.667 \\ -37.667 \\ -35.333 \end{bmatrix}$$

Figure 12

Since Matlab's solves minimization problems, we insert a negative sign in front of the maximization function to solve for minimization problem.

The goal to be achieved is to determine what insertion/deletion pattern should be chosen for the next column given state *i*. Therefore, the probability of choosing *action* a in state i is calculated as shown in equation (12), and is denoted as $\beta_i(a)$.

$$\beta_i(a) \;=\; \frac{y_{ia}}{\sum_{a'} y_{ia'}} \tag{12}$$

$$\textit{where } \{y_{ia}\} \text{ are solution of LP}$$

For example Figure 13 shows a section part of the y_{i2}, which is a part of the linear programming solutions for *action* 2 of sequences shown in Figure 3.

$$y_{i2} = \begin{bmatrix} 0.125 \\ 0.13125 \\ 0.13156 \\ 0.13495 \\ 0.12837 \\ 4.8894e-016 \\ 5.068e-014 \\ 1.8475e-014 \end{bmatrix}$$

Figure 13

$\sum_{a'} y_{ia'}$ is the summation of Linear Programming solutions on state *i* over all possible actions a'. The set of all $\beta_i(a)$ is called a *policy*.

By using theses *policies*, the suggested *action* from this algorithm is chosen. The results of these *policies* are stored in a matrix where the columns represent the *actions* and the rows represent the states. Furthermore, for each state if $\beta_i(a)$ = 1 for some action a, then action a would be chosen as the suggested action. If 0 < $\beta_i(a)$ < 1, then an interval method would be used to select action *a* with probability $\beta_i(a)$. A random number would be chosen in the interval method. Furthermore, the action whose number is greater and is closest to the random number would be selected for the next action.

Figure 14 demonstrates the policy result from the algorithm with figure 3 sample set.

State	*Policy*
1	2
2	2
3	2
4	2
4	2
5	2
6	4
7	8
8	8

Figure 14

Subsequently, the alignments that result from the suggested actions were compared with the alignments obtained by using the CLUSTALW method. These comparisons are done on the same set of sequences so this is just a test of how consistent the algorithm is with CLUSTALW. The purpose of this project is to discover a new algorithm which consists of the same result as CLUSTALW to be used in a longer set.

5. Result

The algorithm has been tested with Honghui Wan data set on the first 40 sequences. The sequences have been aligned with CLUSTALW with a length of 100. This set of data is derived from protein cytochrome p450. The range of extension penalty, *e*, for the test was as 2 to 10 insteps of 2. The *deletion* penalty, *d*, had the same range and $0 \leq \alpha \leq .9$ in steps of .1. A few of examples are described below.

The following is the result observed from the data set with $e = 2$ (extension penalty), $d = 4$ (deletion penalty), and $\alpha = 0.5$. The final result of this algorithm is a matrix that includes the state number with the corresponding suggested and actual action for each of the states.

Figure 15 shows the result of this algorithm for sample data shown in Figure 3.

State	*Suggested Action*	*Actual Action*
1	2	2
2	2	2
3	2	2
4	2	2
4	2	2
5	2	2
6	4	4
7	8	8
8	8	8

Figure 15

The results of the algorithm are represented with symbols for ease of reference. In Figure 16, '*' represents the discrepancies and '–' represents that the suggested and actual sequences matched each other. This means that the result from the algorithm was consistent with the result achieved with CLUSTALW. In this example the discrepancy is equal to 2 since one of the states have been repeated two times.

```
– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –
– M P L L Q G V L S H I L S W T G L C Q V L F V V W T L L G A V V V V W T A K L L V R H V W Y T H –
– – – – – – – – M P Q L S L S S L G L W P M A A S P W L L L L L V G A S W L L A R I L A W T Y T F Y D

* — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — —


– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – I T P T E E G L K N S T Q M S A T Y
– – – R L S C F N K P H A N S W L – – – – – – – – – F G H L G Q M Q S T E E G L Q H V D E L V Q T F
N C C R L R C F P Q P P K R N W F – – – – – – – – – L G H L G L I H S S E E G L L Y T Q S L A C T F

— * — — — — — — — — — — — — — — — — — — — — — — — * — — — — — — — — — — — — — — — — — — — — — — — —
```

Figure 16

Another example of the sample data with the same extension and deletion penalties is shown below. The discrepancies is also 2 in this example.

```
– – – – – – – – M P Q L S L S S L G L W P M A A S P W L L L L L V G A S W L L A R I L A W T Y T F Y D
– – – – – – – –  – M E L T E A F L T L H W G L P R L H H L L A L L C L V A V V Y K L A T L L A K R R D
– – – – – – – –  –  – – – – – – – – – – M W N A L V W Q Q V A A L L C L L A V L L K A T Q I Y L S K K R

— — — — — — * — — — * — — — — * — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — — —


N C C R L R C F P Q P P K R N W F – – – – – – – – – L G H L G L I H S S E E G L L Y T Q S L A C T F
V F R S Y E D F P G P P T – H W L – – – – – – – – – F G H V L E F K Q D G T D F D T L M A W T K Q Y
Q E R I L E Q F P G P P R – H E L – – – – – – – – – L G N V D Q I R R D G K D L D L L V N W T Q S H

— — — — — — — — — — — — — — — — — — — — — — — — — * — — — — — — — — — — — — — — — — — — — — — — — —
```

The second set of data that the algorithm has been tested upon is the data supplied by Dr. Arlin Stoltzfuss of the Center for Advanced Research in Biotechnology at the National Institute of Standards and Technology. This set of consists of aligned proteins of species used in a study of protein evolution. The data set is aligned according to CLUSTALW. The range of extension penalty, *e*, for the test were 2 to 10 with insteps of

2. The *deletion* penalty, *d*, had the same range and $0 \le \alpha \le .9$ in steps of .1 as describe in the sample data above.

The result of three of the sequences from the data set with $e = 2$, $d = 6$, and $\alpha = .1$ with discrepancies = 3 is shown below.

```
K F A E I E S K I D R R S G K E L E K – N P K S I K S G D A A M V R M V P Q K P M C V E V F N D Y A
K F E K I M S E M D K R T G K V L R E – N P D I V K N G K S M M A E L V P S K P L C V E S F Q D Y P
K F A D I K E K C D R R N G K T T E E – N P K S I K S G D A A I V M L V P S K P M C A E A F Q E F P

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _


P L G R F A V R D M R Q T V A V G I I K A V T K K D G – G A G K – – V T K A A A K A A K K ? ? ? ? ?
P L G R F A V R D M R Q T V A V G I I K S T V R A K ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
P L G R F A V R D M R Q T V A V G V I K A V T F K D – – A A G K – – V T K A A E K A Q K K K ? ? ? ?

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ * _ * _ _ _ *_ * _ _ _ _ _ _ _ _ _ _ _ * _ _ _ _ _
```

Another result from the same data set with $e = 2$, $d = 6$, and $\alpha = .1$ with discrepancy of 2 is shown below.

```
K F A E I L T K I D R R S G K E L E K – E P K F L K N G D A G F V K M I P T K P M V V E T F S A Y P
K F Q E I L S K N D R R T G K V I E E – E P K F V K S G D A A M V K L I P T K P M C V E T F S E Y P
K F S E I K E K C D R R T G K T T E T – E P K A I K S G D A A I T V L V P S K P L C V E S F Q E F P

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ * _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _


P L G R F A V R D M R Q T V A V G V I K S V E K K D P – S G A K – – V T K S A A K K G G K ? ? ? ? ?
P L G R F A V R D M K Q T V A V G V I K V V E K K E – – – – – – – – – – – – – – L K K K ? ? ? ? ? ? ?
P L G R F A V R D M R Q T V A V G V I K S V N F K E T – T S G K – – V T K A A E K A Q K K K ? ? ? ?


_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ * _ _ _ _ _ _ * _ _ _* _ * _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```

6. Related Works

As mentioned in the background section on page [4], the methods used to align biological sequences are based on solving a minimization problem (cost) or a maximization problem (score). The propose algorithm is based on solving the maximization problem, also known as dual. However, there exists another method which is called primal where the linear function solves a minimization (cost) problem as shown in (13) below.

Minimization

$$b_1u_1 + b_2u_2 + \dots + b_nu_n$$

Constrain (13)

$$-u_i + \alpha\sum_j P_{ij}(a)u_j >= -C(i,a)$$

$$u_i >= 0$$

The result of multiple sequence alignments using the dual method on the 40 sequences from the Honghui Wan set, each with a length of 100, shows improvement.

For example, Figure 17 shows the result of the discrepancies of the algorithm for the data set. The discrepancy is 2. However, with the same set of data, the result of discrepancies from primal problem is 3, as demonstrated in Figure 18.

```
– – – – – – – – M E L T E A F L T L H W G L P R L H H L L A
– – – – – – – – – – – – – – – – – – M W N A L V W Q Q V A A
– – – – – – – – – – M A L S F L S P S L S R L G L W – – A S

_ _ _ _ _ _ _ * _ * _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```

Figure 17

```
− − − − − − − − M E L T E A F L T L H W G L P R L H H L L A
− − − − − − − − − − − − − − − − − − M W N A L V W Q Q V A A
− − − − − − − − − − M A L S F L S P S L S R L G L W − − A S

* * * * * * * * _ * _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```

Figure 18

Figure 19 demonstrates another example of the result of our algorithm and Figure 20 demonstrates the result of the primal problem.

```
− M P L L Q G V L S H I L S W T G L C Q V L F V V W T L L G
− − − − − − − M P Q L S L S S L G L W P M A A S P W L L L L
− − − − − − − − M E L T E A F L T L H M G L P R L H H L L A

* * _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```

Figure 19

```
− M P L L Q G V L S H I L S W T G L C Q V L F V V W T L L G
− − − − − − − M P Q L S L S S L G L W P M A A S P W L L L L
− − − − − − − − M E L T E A F L T L H M G L P R L H H L L A

* * * * * _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```

Figure 20

7. Conclusion

Sequence alignment is based on biological evaluation. Alignment is a tool for discovering the biological function of genes and proteins, and is important for discovering genetic and DNA related diseases. The proposed algorithm is a new approach in the multiple sequence alignment that uses Markov decision model to solve a linear maximization problem. This algorithm establishes more improved results than primal problem since it generates fewer discrepancies. The algorithm can still be improved to produce more improved result which are consistent with CLUSTALW.

8. References

[1] http://www.ncbi.nlm.nih.gov/About/primer/bioinformatics.html.

[2] http://evotutor.org/Author/Research/Paper-CW.htm.

[3] Dubrin R., Eddy S., Krogh A., Mitchison G. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, 1999.

[4] Hunt F., Kearsley A., O'gallagher A. "Linear Programming Based Algorithm for Multiple Sequence Alignment," Proceedings of the 2003 IEEE Bioinformatics Conference CSB 2003, IEEE Computer Society 2003.

[5] Hunt F., Kearsley A. "An Optimization Approach to Multiple Sequences Alignment," National Institution of Standard and Technology, March 2002.

[6] Jeanmougin, F., Thompson J., Gouy M., Higgins D., and Gibson T. "Multiple sequence alignment with clustalx." Trends in Biochemical Sience, October 1998.

[7] Ross, Sheldon. Intro to Probability Model, 7th edition. Harcourt, Academic Press, Burlington, 2000.

[8] Ross, Sheldon. Applied Probability Models with Optimization Applications. Holden Day, San Francisco, 1970.