DNA Sequences Similarity Using Dynamic Programming

محاذاة سلاسل الدي ان أي باستخدام البرمجة الديناميكية

Inas. R.Shareef
Karbala University/College of Science
Computer Department

Abstract

A DNA sequence is a representation of the genetic code contained within an organism. Molecular biology researchers have great need to compare portions of DNA sequences. The aim of this paper is to find the similarity between the DNA of human and other animals., Dynamic Programming and Dot Plot used to show the result. The relationship between edit distance and string alignment (with prevision for DNA sequence alignment) is demonstrated. The results show that the Dynamic Programming have great facilities to get the similarity between DNA sequences compared with Dot Plot method .

الملخص

تمثل سلسلة دي إن أي DNA الشفرة الجينيةِ للكائن الحي التي تحمل الصفات الوراثية وباحثو عِلْمِ الأحياء في حاجةُ عظيمةُ لمُقَارَنَة أجزاءِ سلاسلِ دي إن أي.

يهدف البحث الى إيَجادَ التشابهَ بين سلاسل دي إن أي للأنسان مع سلاسل الحيوانات، حيث تم تنفيذ طريقتي (البرمجة الديناميكيةDynamic Programming , دوت بلوت Dot Plot) لأيجاد النتائج.

تبين وجود علاقةَ بين تُحريّرُ المسافةً(Edit Distance) ومحاذاة السلاسل (Sequence Alignment) ساعدت في الحصول على نتائج جيدة.

اوضحت النتائج بان البرمجة الديناميكية تملك مرونة وتسهيلات كبيرة لايجاد التشابه بين سلاسل الدي ان أي مقارنة مع طريقة الدوت بلوت .

1. Introduction

Through biological experiments they discover that study of DNA is very important because it have important rule in limitation the characters of organisms, to implement accurate study for genetic sequence of DNA they breake it into shortest subsequence to be studied more easily. A genetic sequence is a string formed from four letters alphabet Adenine(A), Thymine(T), Guanine(G), Cytosine(C) of biological macromolecules referred together as the DNA bases[1].

A gene is a genetic sequence that contains the information needed to construct a protein. All of your genes taken together are referred to as the human genome, a blue print for the parts needed to construct the proteins that form your cells and, by extension, your body. Each new cell produced by your body receives a copy of the genome this copying process as well as natural wear and tear, introduces a small number of changes into the sequences of many genes[1].

Among the most common changes are the substitution of one base to another and the deletion of substring of bases, such changes are generally referred to as point mutations. As a result of these point mutations, the same gene sequenced from closely related organisms will have slight differences [1].

In this paper two methods of DNA sequences was implementation to show the similarity between DNA of human with other mammal animals . The rest of paper organized as follow, definition of DNA explained in section 2, motivation of this subject explained in section 3 , section 4 contains the principle of sequence comparison , description of computational methods of alignment shown in section 5

, Implementation and Results in section 6,finally , Conclusion was demonstrated in section 7 .

2. Definition of DNA

DNA is an acronym for the molecule Deoxyribo Neocleic Acid. DNA is contained in each living cell of an organism and it is the carrier of that organism's genetic code. The genetic code is a set of sequence which defines what proteins to build with in the organism, since organisms must replicate and/or reproduce tissue for contained life, there must be some means of encoding the unique genetic code for the proteins used in making that tissue. The genetic code is information which will be needed for biology growth and reproductive inheritance[2] .

2.1 Components of DNA sequence

In DNA sequence we deal with complex information in department that represents this sequence, it contains four nucleotide molecules which are identical in all represents excepting a nitrogen base. The four nucleotides are thus named after these different bases: adenine, guanine, cytosine, and thymine. The nucleotides are often denoted by the letters A, G, C, and T. A nucleotide will bond may only follow the pairing A-T or G-C. This pairing is called a base pair (bp) thus; the two strands composing the DNA molecule are exactly complementary. [2]

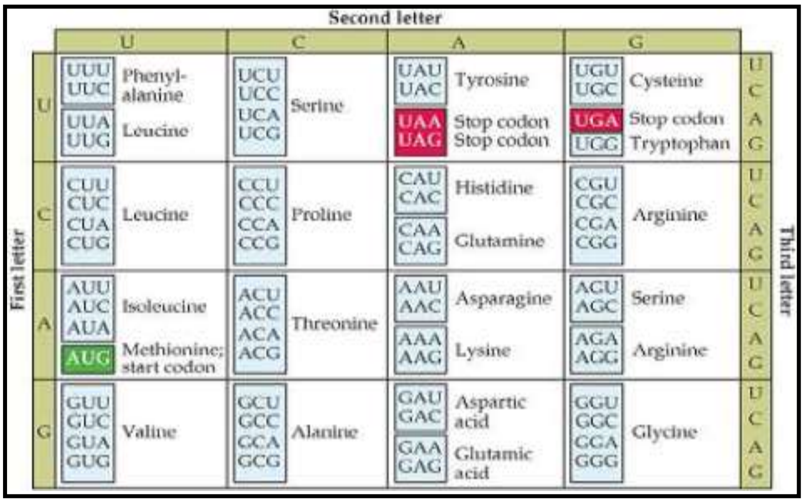
2.2 Codons

Although the genetic code is composed of DNA sequences, proteins are not built directly from them. There are intermediary chemicals called amino acids which when combined in a certain order, lead to proteins. The DNA sequence is split into triplets of nucleotides which code for these amino acids. The triplet is called a 'Codons'.

There are approximately 20 amino acids used in proteins. An amino acid is en coded by a sequence of three nucleotides, called a codon, for example, the amino acid methionine is represented by the codon ATG.

If we had other amino acids represented by the codons CGA, TAC, AAG, TTG, and TGA, then we could string them together to produce a sequence: ATGCGATACAAGTTGTGA which may represent the encoding for particular protein comprised of 6 amino acids. (We have 18 nucleotides here, and each amino acid is encoded by 3 nucleotides).

Since we have 4 possible nucleotides and a codon is composed of 3 nucleotide, there are 4^3=64 possible codon triples, but since in the whole of life on earth there are only 20 amino acids used to compose proteins ,most amino acids are specified by more than one codon [2] .



	Second letter				
First letter	U	C	A	G	Third letter
U	UUU UUC Phenylalanine UUA UUG Leucine	UCU UCC UCA UCG Serine	UAU UAC Tyrosine UAA Stop codon UAG Stop codon	UGU UGC Cysteine UGA Stop codon UGG Tryptophan	U C A G
C	CUU CUC CUA CUG Leucine	CCU CCC CCA CCG Proline	CAU CAC Histidine CAA CAG Glutamine	CGU CGC CGA CGG Arginine	U C A G
A	AUU AUC AUA Isoleucine AUG Methionine; start codon	ACU ACC ACA ACG Threonine	AAU AAC Asparagine AAA AAG Lysine	AGU AGC Serine AGA AGG Arginine	U C A G
G	GUU GUC GUA GUG Valine	GCU GCC GCA GCG Alanine	GAU GAC Aspartic acid GAA GAG Glutamic acid	GGU GGC GGA GGG Glycine	U C A G

Figure1: List of Codons

3. Motivations

Some of the most common users of DNA sequence alignment and similarity are in determining the function of new sequences, and in medical applications.

3.1 Determine function of new sequences

As new sequences are discovered and catalogued, it becomes necessary to hypothesize their function. One way is to look for matches in a database of sequences for which we already know the protein encoding. If we can find the match (or close match) in this database, we then have a clue as to the new sequence's function[2].

3.2 Medical Application

Multiple sclerosis is a disease in which the immune system T-cells attack the body's myelin sheath around nerves. It was hypothesized that myelin sheath proteins are similar to viral or bacterial sheath proteins from an earlier infection. So, researchers:

1. Sequenced myelin sheath proteins.
2. Searched a protein database for similar bacterial and viral sequences.
3. Performed lab tests to determine if T-cells attacked these same proteins.

They discovered that the immune system was indeed confusing bacterial and viral proteins with the body's own myelin sheath proteins. This was a vital step in the progress toward treating multiple sclerosis[2] .

4. Sequence Comparison

Sequence comparison can be defined as the problem of finding which parts of the sequences are similar and which parts are different. It is regarded as the building block for many other, more complex problems such as multiple alignments (the comparison of a group of related sequences) and the construction of phylo genetic trees that explain the evolutionary relationship among species. Sequence comparison is actually a well know problem in computer science. For the computer scientist, bio molecular sequences are just another source of data. Indeed, one that has experienced a tremendous growth in interest to the point that it has spawned an interdisciplinary field of its own, generally know as bio informatics, computational molecular biology or just computational biology. As biological databases grow in size, faster algorithms and tools are needed. This work will concentrate on efficient algorithms for comparing two sequences.[4]

4.1 Sequences Alignment And Similarity

Sequence aligning is used for sequence comparisons in molecular biology. By aligning sequences we can see how similar the sequences are, and from that information draw some conclusions on how related the sequences are.

Biologists use the comparison to discover evolutionary divergence, the origins of disease, and ways to apply genetic codes from one organism in to another. If the two genetic sequences are similar functions. We would like a way to quantify "similar enough" [2].

The idea of aligning two sequences (of possibly different sizes) is to write one on top of the other, and break them into smaller pieces by inserting spaces in one or the other so that identical subsequences are eventually aligned in a one-to-one correspondence . Naturally, spaces are not inserted in both sequences at the same position. In the end, the sequences end up with the same size. The figure2 example illustrates an alignment between the sequences

```
A = ACAAGACAG - CGT
    | || | || |||
B = AGAACA - AGGCGT
```

Figure2 . Alignment of Two Sequences

The objective is to match identical subsequences as far as possible. In the example, nine matches are highlighted with vertical bars. However, if the sequences are not identical, mismatches are likely to occur as different letters are aligned together. Two mismatches can be identified in the example: a .C. of A aligned with a .G. of B, and a .G. of A aligned with a .C. of B. The insertion of spaces produced gaps in the sequences. They were important to allow a good alignment between the last three characters of both sequences . An alignment can be seen as a way of transforming one sequence into the other. From this point of view, a mismatch is regarded as a substitution of characters. A gap in the first sequence is considered an insertion of a character from the second sequence into the first one, whereas a gap in the second sequence is considered a deletion of a character of the first sequence.

In the previous example, A can be converted into B in four steps:

1. substitute the first .C. for a .G.
2. substitute the first .G. for a .C.
3. delete the second .C.
4. insert a .G. before the last three characters.

Once the alignment is produced,a score can be assigned to each pair of aligned letters,called aligned pair,according to a chosen scoring scheme.We usually reward matches and penalize mismatches and gaps. The overall score of the alignment can then be computed by adding up the score of each pair of letters.For instance, using a scoring scheme that gives a+1value to matches and −1to mismatches and gaps -2 ,the alignment of above example scores $9\cdot(1) + 2\cdot(-1)+ 2\cdot(-2) = 3$.

The similarity of two sequences can be defined as the best score among all possible alignments between them. Note that it depends on the choice of scoring scheme. In the next sections, the Computational methods of finding the best alignment of two sequences (an alignment that gives the highest score) will be addressed. A related notion is that of distance. However, this work will focus on similarity, as it is the preferred choice for biological applications. In general, our scoring could be (+m) for a match, (-s) for a mismatch, and (-d) for a gap. The score of alignment will there for is[3]: +m (#matches)-s (#mismatches)-d (#gaps) .

5.Computational Methods Of Alignment

Two methods used to satisfied the aim of this paper:

5.1 Dot Plot

A simple way to visualize possible alignments between two sequences is to look at a dot plot. Regions of similarity between two DNA sequences can be plotted by hand. The simplest dot plot between a sequence of length m and a sequence of length n is an m×n matrix with dots placed in locations that correspond to matching elements (i.e. a dot is placed in location (i,j) if x(i)=y(j)).

First sequence along the columns of the matrix , Second sequence along the rows of the matrix ,then assign a dot to a matrix cell, whenever the column element is identical to a row element Diagonal lines, denote identical regions in both sequences Example : if x =TESTSEQ, y=NEWTESTS [5] .

	N	E	W	T	E	S	T	S
T				*			*	
E		*			*			
S						*		*
T				*			*	
S						*		*
E		*			*			
Q								

Figure3: .Explain the Dot Plot Method

The quality of this method judged by eye ,and the Dot Plot is fine for small sequences, but is not adequate for very long ones, needless to say, techniques must be developed to address this huge space, and also to perform calculations with in this space more optimally than brute force [2].

5.2 Dynamic Programming

The goal of Dynamic Programming is to find the optimal solution to a complicated problem ,the approach is break the problem down into smaller, tractable problems, and solve those in a way that is optimal for the final solution ,"dynamic" because the program is constantly making decisions and executing bits of code in ways that could not be programmed linearly [6] .

In this paper, Dynamic programming was used to find best similarity of DNA sequences which is a powerful algorithmic paradigm, first introduced be Bellman in the context of operations research and then applied to the alignment of biological sequences by Needleman and Wunsch. Dynamic programming now plays the leading role in many computational problems, including control theory, financial engineering, and bio informatics. The key idea of dynamic programming is to break up a large computational problem into smaller sub problems, and eventually use the stored answers to solve the original problem[2].

Dynamic programming uses the concept of 'memorization' to eliminate calculating values more than once. Memorization is simply the process of storing a calculated value which we know will be used to calculate other values later in computation.

In the case of sequence alignment, we take a cue from the use of dot plots (See Figure 2). Similar to a dot plot, we create a table arranging one string on a vertical axis and another string on a horizontal axis. Unlike a dot plot, however, we populate the cells not with dots but rather with numerical values. The numerical values represent an edit distance (or some other score, as will be shown later in the case of DNA sequences)[2].

Edit distance

Edit distance can be thought of as the "difference" between two strings. The difference between two strings is measured by counting the number of edit operations which must be performed character by character, to transform one string into another. These edit distance operations are[2]:

R =replace	I =insert
D =delete	M =match

For example, to transform the string "cat" to string "chat" we can insert (I) the character 'h' between the 'c' and 'a' of "cat", yielding the string "chat". There are many possible edit distances between "cat" and "chat", but the minimal edit distance is one (I)—just one insertion. Our objective is to discover a minimal edit distance which will tell us, the minimum number of edit operations which may be used to transform one string into another. This number is the most interesting because we are usually searching for strings or proteins of strings with the most similarity. How can we derive or calculate a value for the edit distance between two strings? We must use a general description of the problem. To illustrate, Say that we have two strings S and T, of length n and m, respectively ,|S| =n, |T| =m, We can define D(i,j) as the value of the minimal edit distance between strings S[1]…S[i] and T[1]…T[j] ,so that the minimal edit distance between S and T is D(n,m).[2]
In order to calculate D(i,j) ,we must establish a base condition based on two cases:

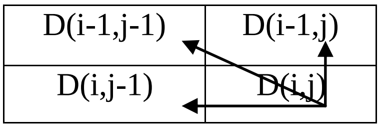
$$D(i,0)= i , \quad D(0,j)=j$$

Note: D(0,0)=0

And establish a relation, used in all other cases:

$$D(i,j) = \text{minimum} \begin{cases} D(i-1,j)+1 \\ D(i,j-1)+1 \\ D(i-1,j-1)+p(i,j) \end{cases} \qquad \ldots(1)$$

Where : $p(i,j)=1$ if $S[i] \neq T[i]$ and
$p(i,j)=0$ if $S[i] =T[i]$



For example, consider the edit distance between the strings "cat" and "chat",
S = "cat", T ="chat", n =|S| =3, m =|T| =4.
Thus, the minimal edit distance between "cat" and "chat" is defined as:
D(n,m) = D(3,4).
Since, there is only one edit operation required to transform "cat" to "chat" (inserting the 'h' character), the value of D(3,4) is 1.
We can think of this function as string that the minimal effort required to transform the first three (3) characters of "cat" into the first four (4) characters of "chat" is one (1) edit operation. Applying this generalization to other transformations, we can see, for example, that D(3,0) represents the minimal effort required to transform the string "cat" into null string, (the edit operations are: delete 'c', delete 'a', delete 't').
Thus, D(3,0) has value of 3.
The series of edit operations is called an "edit transcript ", and is represented in the following way:

I (the edit transcript)
C AT (original string)
CHAT (transformed string)
(edit distance =1).

Note that there may be more than one edit distance transcript to active the same alignment:

MRIM (the edit transcript)
CA T (original string)
CHAT (transform string)
(edit distance =2).

We can see that the edit distance in the sum of insertions, deletions, and replacements in an alignment of strings. The minimal edit distance found in an optimal alignment of strings. [2]

	0	C	H	A	T
0	0	1	2	3	4
C	1	0	1	2	3
A	2	1	1	1	2
T	3	2	2	2	1

Figure4: Edit Distance Example

This visual alignment is all fine and well, and the minimal edit distance may be a good way to quantitatively compare strings, but what about DNA sequences? It turns out that gaps are important in DNA sequences because they represent significant biological events. A run of gaps must be accounted for, and it would be better to score insertions, deletions, and replacements differently. We must have some way of scoring our alignments.

Gaps are an important concept in biological applications, because a stream of gaps in a DNA sequence may represent a significant biological characteristic. Gaps usually incur a ‘penalty’ to the potential alignment score between two sequences, depending on the length of the gap. The way to achieve a more useful score is to turn around our definition a bit. Instead of computing the minimal edit distance between strings or sequences, it will help to compute the maximum similarity between them. To do this, we need a scoring function.
Say that we have two strings S and T of differing lengths. Alignment A maps S and T into strings S’ and T’, such that |S’| = |T’|. If we remove the gaps from S’ and T’, we will have restored S and T.

The value of alignment A is defined as[2]:

$$\sum_{i=1}^{l} \square \ \sigma (S'[i], T'[i]) \qquad \ldots(2)$$

Where l =|S'|=|T'| , σ (x,y) is a *scoring function*.

An *optimal alignment* is an alignment that has maximum possible value for these two strings. The scoring function produces a value dependent upon the matching qualities of two characters in our strings. For example, with two distinct characters 'A' and 'G', we can employ a scoring function defined as such:
s(A,A) = +2, match
s(G,A) = -1, substitution of 'A' for 'G'
s(G,-) = -1, deletion of character 'G'
s(-,G) = -1 , insertion of character 'G'
Using a scoring function, we can apply different scoring tables to our algorithm to achieve results which are biologically interesting.
Say that we have two strings S and T, of length n and m, respectively.
|S|=n, |T|=m

We can define V(i,j) As the value of the maximum score of the string alignments S[1]…S[i] and T[1]…T[i]
So that the maximum score of an alignment between S and T is V(n,m)
In order to calculate V(i,j), we must:
-Establish base conditions:

$$V(0,0) = 0$$
$$V(i,0) = V(i-1,0)+\text{score}\ (S[i],\_) \qquad \text{for } i>0$$
$$V(0,j)=V(0,j-1)+\text{score}\ (\_,T[j]) \qquad \text{for } j>0$$

-And establish a recurrence relation:

$$V(i,j) = \text{maximum} \begin{cases} V(i-1,j)+\text{score}(S[i],\_) \\ V(i,j-1)+\text{score}(\_,T[j]) \\ V(i-1,j-1)+\text{score}(S[i],T[j]) \end{cases} \qquad \ldots(3)$$

For 0 < I >=n and 0 < j >=m.

The algorithms of this paper explained in the following to show the similarity between two sequences,

```
Dot Plot Algorithm
Inputs:    Two Sequences S,T .
Outputs:  Find similarity between  S,T .
Begin Of Algorithm
   Step1: Create  an m×n  matrix, where m is a length of S , n is a length of T.
   Step2:  Put sequences  S & T on Horizontal & Vertical axes .
  Step3: Mark " * " for nucleotide match Self-sequence comparison .
End Of Algorithm
```

```
Dynamic Programming Algorithm For Edit Distance
Inputs:    Two Sequences S,T .
Outputs:  Find minimal edit distance between  S,T .
Begin Of Algorithm
    -Initialization
        $D(0, 0) = 0$
        $D(i, 0) = I$ for $i = 1…m$
        $D(0, j) = j$ for $j = 1…n$
    - Main Iteration (Aligning prefixes)
        for each $i = 1…m$
        for each $j = 1…n$
        $D(i, j) = \max [ D(i-1, j) +1,$
                          $D(i, j-1) +1,$
                          $D(i-1, j-1) + t( S_i ,T_i) ]$
End Of Algorithm.
```

```
Dynamic Programming Algorithm For Scoring Function
Inputs:    Two Sequences S,T .
Outputs:  Find maximum similarity between  S,T .
Begin Of Algorithm
    -Initialization
        $V(0, 0) = 0$
        $V(i,0) = V(i-1,0)+ \sigma (S[i],\_)$        for $i =1…m$
        $V(0,j)=V(0,j-1)+ \sigma (\_,T[j])$        for $j =1…n$
    - Main Iteration (Aligning prefixes)
        for each $i = 1…m$
        for each $j = 1…n$
        $V(i, j) = \max [ V(i-1,j)+ \sigma (S_i ,\_ ) ,$
                          $V(i,j-1)+ \sigma (\_, T_i ) ,$
                          $V(i-1,j-1)+ \sigma ( S_i ,T_i ) ]$
End Of Algorithm.
```

6.Implementation and Results

In this section , more details can be founded to explain the stages of running the designed software, Real DNA sequences of human and other animals(Mouse, Chicken, Rat, Dog) were taken from web site in [7], to compare the genetic characteristic between them .The interface of designed software shown in Figure 5.

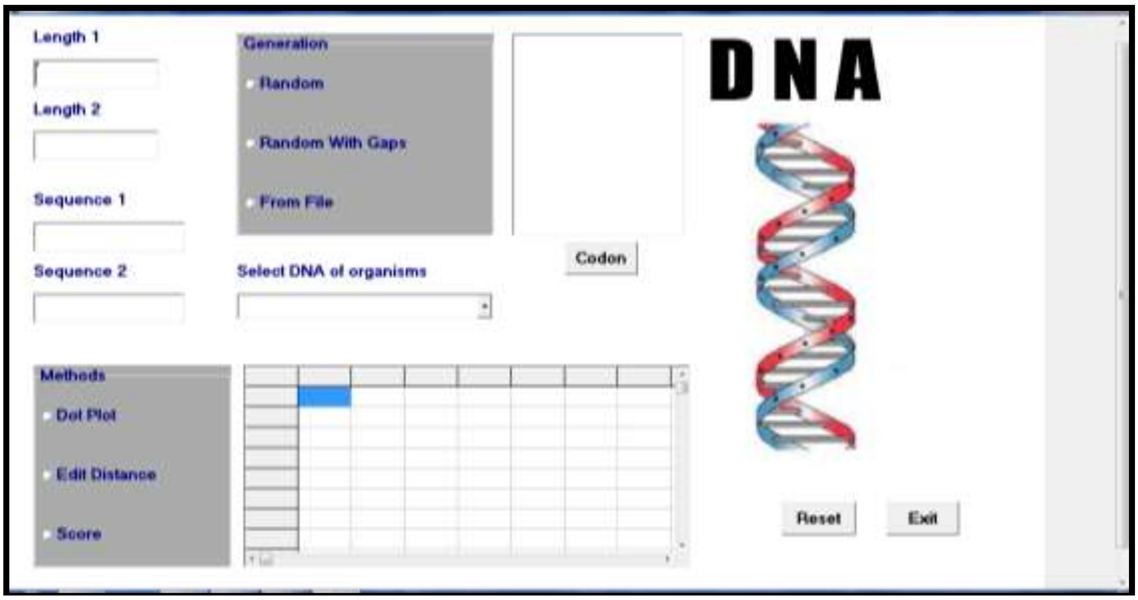


Figure5: Software Interface

More facilities can be shown in the interface , the steps or stages of running explained in the following diagram :

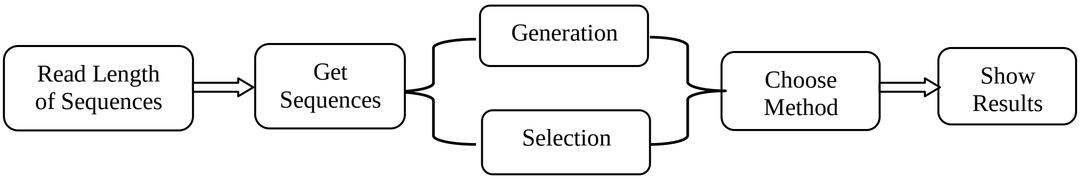
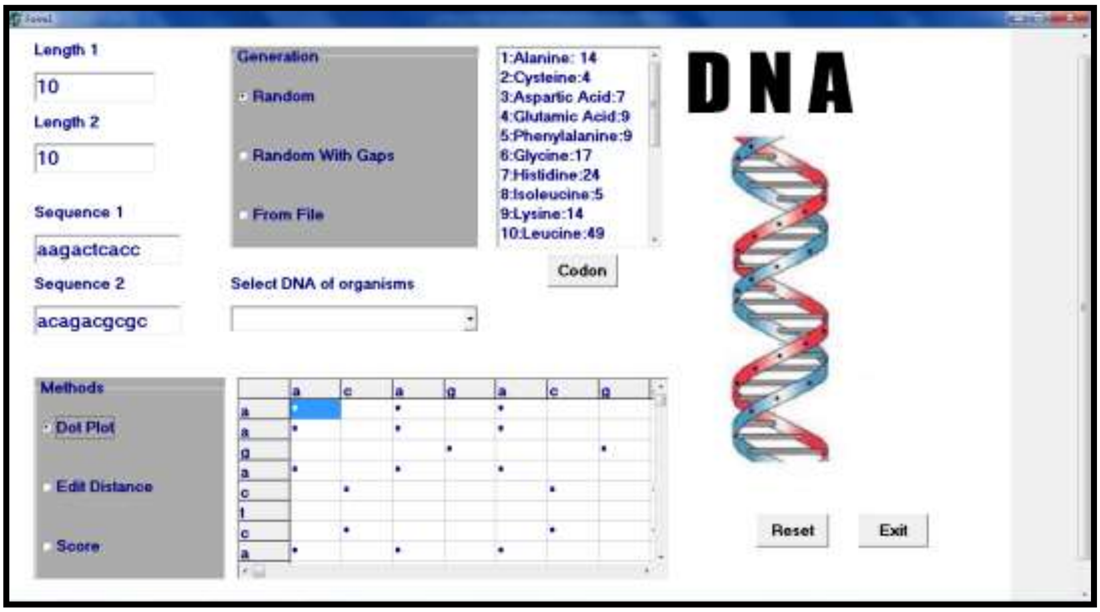


Figure 6. Diagram of Running Stages

Software beginning by read length of sequences , then , can generate two sequences by one way of generating methods (Random, Random with gaps ,or Take it from file) ,Or the user can input the DNA sequences directly, or may be selected from real sequences of human or animals obtained from web site in [7] .

After we having two DNA sequences, any method can be chosen to find the similarity between them such as (dot plot, edit distant, score function) and to show the results. In The dot plot button will display the result on matrix which represent the similarity between the two DNA sequences as dots in cross of coordinate of two similar nucleotides, see figure 7.



Length 1: 10

Length 2: 10

Sequence 1: aagactcacc

Sequence 2: acagacgcgc

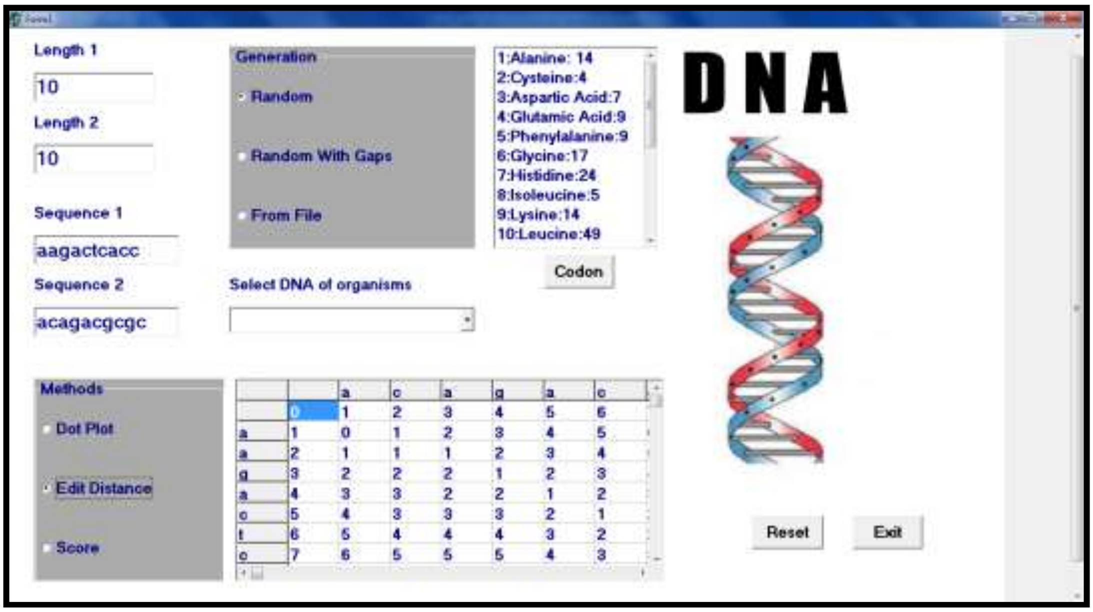
Select DNA of organisms:

1:Alanine: 14
2:Cysteine:4
3:Aspartic Acid:7
4:Glutamic Acid:9
5:Phenylalanine:9
6:Glycine:17
7:Histidine:24
8:Isoleucine:5
9:Lysine:14
10:Leucine:49

	a	c	a	g	a	c	g
a	•		•		•		
a	•		•		•		
g				•			•
a	•		•		•		
c		•				•	
t							
c		•				•	
a	•		•		•		

Figure 7: Result of implement Dot Plot Method

In the *edit distance method* ,the resulted values represent the minimal number of edit operation required to transform the first sequence to the second sequence, DNA sequence of human with DNA sequence of other organism was compared to find minimal different between them and know nearest organism to human in characteristics, in this study real part of DNA sequences of organisms in the same index (sixth chromosome) were taken, see figure 8 .



Length 1: 10

Length 2: 10

Sequence 1: aagactcacc

Sequence 2: acagacgcgc

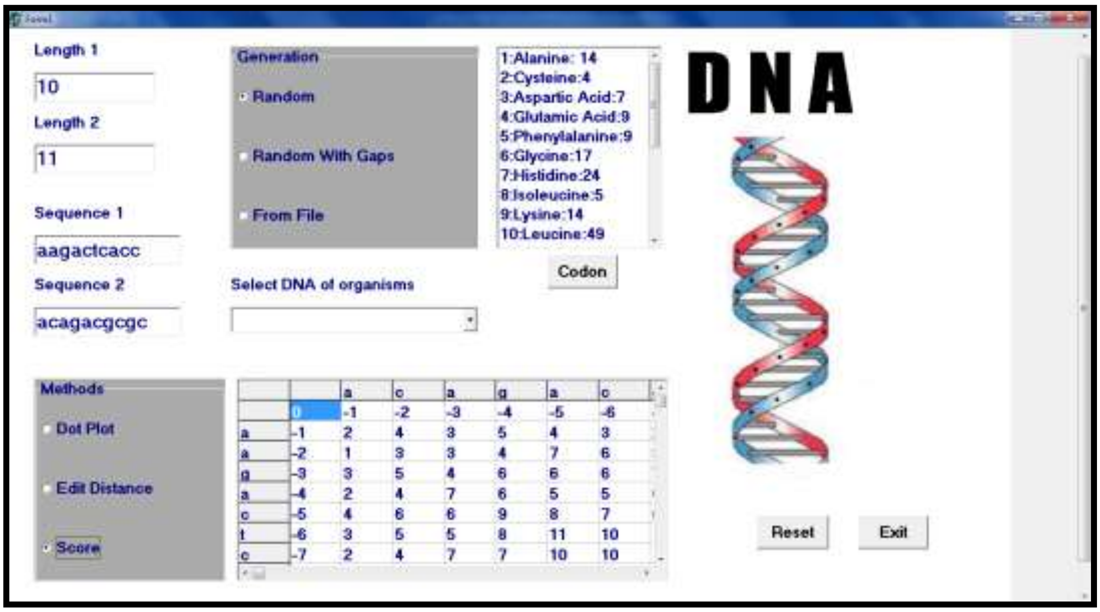
Select DNA of organisms:

1:Alanine: 14
2:Cysteine:4
3:Aspartic Acid:7
4:Glutamic Acid:9
5:Phenylalanine:9
6:Glycine:17
7:Histidine:24
8:Isoleucine:5
9:Lysine:14
10:Leucine:49

		a	c	a	g	a	c
	0	1	2	3	4	5	6
a	1	0	1	2	3	4	5
a	2	1	1	1	2	3	4
g	3	2	2	2	1	2	3
a	4	3	3	2	2	1	2
c	5	4	3	3	3	2	1
t	6	5	4	4	4	3	2
c	7	6	5	5	5	4	3

Figure 8: Result of implement Edit Distance Method

In the *score method*, which used to find the maximum similarity between two DNA sequences with gaps(when the sequences are not equal in length), the resulted values represent the maximum number of alignment, show figure 9.



Length 1: 10

Length 2: 11

Sequence 1: aagactcacc

Sequence 2: acagacgcgc

Generation: ● Random ○ Random With Gaps ○ From File

Select DNA of organisms

1:Alanine: 14
2:Cysteine:4
3:Aspartic Acid:7
4:Glutamic Acid:9
5:Phenylalanine:9
6:Glycine:17
7:Histidine:24
8:Isoleucine:5
9:Lysine:14
10:Leucine:49

Methods: ○ Dot Plot ○ Edit Distance ● Score

		a	c	a	g	a	c
	0	-1	-2	-3	-4	-5	-6
a	-1	2	4	3	5	4	3
a	-2	1	3	3	4	7	6
g	-3	3	5	4	6	6	6
a	-4	2	4	7	6	5	5
c	-5	4	6	6	9	8	7
t	-6	3	5	5	8	11	10
c	-7	2	4	7	7	10	10

Figure 9: Result of implement Score Method

The DNA sequences consist of 20 types of codon, to know the name of each codon and the number of its appear, all this will display on the text field after press the *codon's button*, codones can be displayed to show their organization.

7. Conclusion

Real DNA sequences of human and other animals(Mouse, Chicken, Rat, Dog) were taken from web site in [7] to compare the genetic characteristic between them. This study contains two sides which make our conclusions:

In biology side (by help biological experimenters):

1. Difficult deal with these sequences that these sequences carry the genetic characteristics, this sequencing of DNA alphabet determine the organism's characteristics, because of difficult in dealing with map of DNA sequence for any organism, so it will deal with small part of chromosome which carry enzymes that represents special characteristics .

   By this study found the sixth chromosomes that contain prolactin hormones that responsible for milk in the mammals.
2. From application this project can conclusion the closest animals to human from the genomic characteristics is (Rat) which also conclude by biology experiment.

In programming side

1. By using the dynamic programming to find the similarity between DNA sequences which is the best way for dealing with long DNA sequences represent tool that facilities biology's work.
2.The Dynamic programming algorithm is best from Dot Plot algorithm because Dynamic programming use the concept of "memorization" is simply the process of storing a calculate value which we know will be used to calculate other values later in computation.
3. The gaps are important in DNA sequences , effect on finding similarity between two DNA sequences by score function.

References

[1] Thomas Clark, Robert Sedgewick ,Global Sequence alignment ,2002.
[2] M.Lehman,Experiments with algorithms for DNA sequence Alignment , 2002 .
[3] Sérgio Anibal, Sequence alignment algorithms ,London University ,2003.
[4] Ellen Sherwood, Methods and applications of DNA sequences alignment ,Stockholm,2007.
[5] Zivanovic *et al*., Dot plot matrix, 2002 .
[6] Saad Mneimneh, Computational biology, 2000
[7]www. ncbi.nlm.nih.gov