Finding the Best Biological Pairwise Alignment Through Genetic Algorithm

Determinando o Melhor Alinhamento Biológico Através do Algoritmo Genético

Paulo Mologni [1], Ailton Akira Shinoda[2], Carlos Dias Maciel[3]

Resumo

Este artigo propõe um método alternativo baseado em algoritmo genético para determinar o alinhamento ótimo de pares de seqüências biológicas. O artigo descreve o alinhamento de pares de seqüências levando em consideração o cromossomo. O cromossomo está associado a uma função objetivo dependente da matriz de substituição BLOSUM50.
Palavras-chave: alinhamento genético, programação dinâmica, algoritmo genético

Abstract

This article describes an alternative method based on Genetic Algorithm (GA) to find the optimal pairwise alignment. It describes the alignment sequence taking into account the chromosome. Each chromosome is associated with a fitness function based on BLOSUM50 substitution matrix.
Key words: biological alignment, dynamic programming, genetic algorithm

Introduction

As a segment of genetic material is passed on through the generations in some line of descent in a population, the sequence constituting this material will change through the process of mutation. The simplest mutations are of the form of a switch from one nucleotide to another, or in the form of an insertion or a deletion. Mutations can spread to an entire species, or nearly so, through the process of natural selection or random drift. When a switch in nucleotides spreads throughout most of a species we call it a substitution. As substitution, insertions, and deletions get passed along through two independent lines of descent, the two sequences will slowly diverge from each other. For example, the original sequence may have been

cggtatcca,

whereas the two descendents might be

cgggtatccaa

and

ccctaggtccca.

This divergence will happen at varying states, depending on the function of the piece of DNA in question and how well that function tolerates substitutions and other changes. For protein coding DNA, the corresponding protein sequences also

[1] M.Sc. Student at the Department of Electrical Engineering, Londrina State University
[2] Adjunct Professor with the Department of Electrical Engineering, Londrina State University – e-mail: shinoda@uel.br.
[3] Adjunct Professor with the Department of Electrical Engineering, Londrina State University – e-mail: maciel@uel.br.

evolve through time as result of DNA sequence evolution. Individual genes typically contain stretches that change rapidly and other stretches that remain relatively constant. The latter regions are called function domains since their low tolerance to change suggests that they have a critical function role in the viability of the organism.

Many problems in bioinformatics relate to the comparison of two (or more) DNA or protein sequences. In order to compare sequences of nucleotides or amino acids, it is usual to use alignments. The following is an example of an alignment of the above two descendent sequences:

c g g g t a - - t c c a a

c c c - t a g g t c c c a

The symbol "-" is called an *indel* : it represents an assumed insertion or deletion at some point in the evolutionary history leading to the two sequences. A sequence of *l* consecutive indels is called a gap of length *l*. In the above alignment there are two gaps, one of length 1 and one of length 2.

There are many types of alignments. There are global alignments, in which the entire lengths of sequences are aligned, and there are local alignments, which align only subsequences of each sequence. There are gapped alignments, in which indels are allowed, and there are ungapped alignments, in which indels are not allowed. There are pairwise alignments, which are alignments of two sequences, and there are multiple alignments, which align more than two sequences.

For a particular type of pairwise alignment (for instance ungapped global), there are many possible such alignments between any two sequences. Good alignments of related sequences are ones that better reflect the evolutionary relationship between them. There are several ways to discriminate between good and bad alignments.

This article proposes an alternative method based on Genetic Algorithm (GA) (HOLLAND, 1975), to find the optimal pairwise alignment. The alignment sequence is described in the chromosome and each chromosome is associated with a fitness function based on BLOSUM50 (ALTSCHUL, 1991) substitution matrix. For each generation a few pairs of chromosome are randomly chosen for crossover. The next step is the mutation of these offspring taking into account the mutation rate. The last step is the selection of chromosomes with high fitness function. This cycle is repeated until a desired termination criterion is reached. This criterion can also be set by the number of evolution cycles, or the amount of variation of individuals between different generations, or a pre-defined value of fitness. In this work the number of evolution cycles (computational runs) was chosen. Section 2 describes some types of alignment, section 3 introduces GA algorithm, section 4 presents the GA approach in pairwise alignment, and section 5 shows the result employing the GA approach.

Types of Alignment

The global alignment Needleman-Wunsch algorithm (NEEDLEMAN; WUNSCH, 1970) uses dynamic programming, which is a general computational technique used in many fields of study. It is applicable when large searches can be divided into a succession of small stages so that the solution of the initial search stage is trivial, each partial solution in a latter stage can be calculated by reference to only a small number of solutions in an earlier stage, and the final stage contains the overall solution. Dynamic programming can be applied to alignment problems because similarity indices obey the following rule:

$$S_{1\to x,1\to y} = \max S_{1\to x-1,1\to y-1} + S_{x,y} \qquad (1)$$

in which $S_{1\to x,1\to y}$ is the similarity index for the two sequences up to residue *x* in the first sequence and residue *y* in the second sequence, $\max S_{1\to x-1\to y-1}$ is the similarity index for the best alignment up to

residues *x*-1 in the first sequence and *y*-1 in the second sequence, and $S_{x,y}$ is the similarity score for aligning residues *x* and *y*.

An alignment is calculated in two stages. First, the two sequences are arranged in the same way as a dot in a matrix. For each element in the matrix, say *x* and *y*, the similarity index, $S_{1\to x,1\to y}$, is calculated. At the same time, the position of the best alignment score in the previous row or column is stored. This stored value is called a pointer. The relationship between the new $S_{1\to x,1\to y}$ value and the pointer is represented by an arrow. In the second stage the alignment is produced by starting at the highest similarity score in either the rightmost column or the bottom row, and proceeding form right to left by following the best pointers. This stage is called the traceback. The graph of pointers in the traceback is also referred to as the path graph because it defines the paths through the matrix that correspond to the optimal alignment.

A much more common situation is where we are looking for the best alignment between subsequences of *x* and *y*. This arises for example when it is suspected that two protein sequences may share a common domain, or when comparing extended sections of genomic DNA sequence. It is also usually the most sensitive way to detect similarity when comparing two very highly diverged sequences, even when they may have a shared evolutionary origin along their entire length. This is because usually in such cases only part of the sequence has been under strong enough selection to preserve detectable similarity. The rest of the sequence will have accumulated so much noise through mutation that it is no longer alignable. The highest scoring alignment of subsequences of *x* and *y* is called the best local alignment. An algorithm was proposed by Smith and Watermann and local alignment is sometimes referred to as Smith Watermann alignment (SMITH; WATERMAN 1981).

Genetic Algorithm (GA)

Throughout a genetic evolution, the fitter chromosome has a tendency to yield good quality offspring that means a better solution to any problem. In a practical GA application, a population pool of chromosomes has to be installed and these can be randomly set initially. The size of this population varies from one problem to another. In each cycle of genetic operation, termed as an evolving process, a subsequent generation is created from the chromosomes in the current population. This can only succeed if a group of these chromosomes, generally called *parents* or a collection term *mating pool* is selected via a specific selection routine. The genes of the parents are mixed and recombined for the production of offspring in the next generation. It is expected that from this process of evolution (manipulation of genes), the *better* chromosome will create a larger number of offspring, and thus has a higher chance of surviving in the subsequent generation, emulating the survival-of-the-fittest mechanism in nature.

The basic principles of GA were first proposed by Holland (HOLLAND, 1975). GA is inspired by the mechanism of natural selection where stronger individuals are like the winners in a competition environment. Through the genetic evolution method, an optimal solution can be found and represented by the final winner of the genetic game.

GA presumes that the potential solution of any problem is an individual and can be represented by a set of parameters. These parameters are regarded as the genes of a chromosome and can be structured by a string of values in binary form. A positive value, generally known as a fitness value, is used to reflect the degree of quality of the chromosome for the problem, which would be highly related with its objective value.

In order to facilitate the GA evolution cycle, two fundamental operators: crossover and mutation operator are required, although the selection routine can be termed as the other operator. To further

illustrate the operational procedure, one-point crossover mechanism is depicted on Figure 1. A crossover point is randomly set. The portions of the two chromosomes beyond this cut-off point to the right are to be exchanged to form the offspring. An operation rate (p_C) with a typical value between 0.5 and 1.0 is normally used as the probability of crossover.

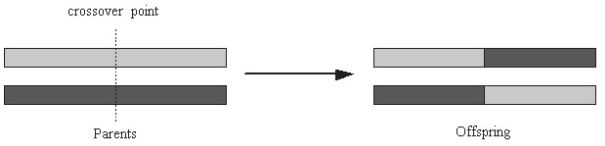

Figure 1- One-point crossover

However, for mutation (Figure 2), this is applied to each offspring individually after the crossover exercise. It alters each bit randomly with a probability p_m .

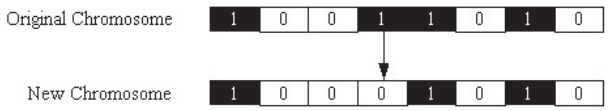

Figure 2- Bit mutation on the fourth bit

The choice of p_m e p_C as the control parameters can be a complex nonlinear optimization problem to solve. Furthermore, their settings are critically dependent upon the nature of the objective function.

Figure 3 is a flow chart of a genetic algorithm. The algorithm begins by defining a chromosome as an array of parameter values to be optimized.

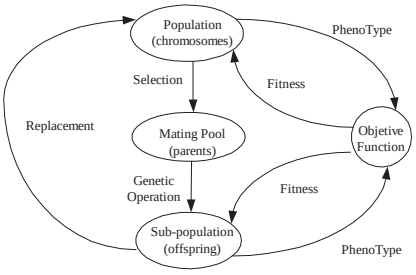

Figure 3- A Genetic Algorithm cycle

GA Approach

We use two short amino sequences to illustrate the GA approach, S_1= *HEAGAWGHEE* and S_2= *PAWHEAE*, where each amino acid is described in Table 1.

Table 1- The genetic code

Amino Acid	Symbol
alanine	A
arginine	R
aspartic acid	D
asparginine	N
cysteine	C
glutamic acid	E
glutamine	Q
glycine	G
histine	H
isoleucine	I
leucine	L
lysine	K
methionine	M
phenylalanine	F
proline	P
serine	S
threonine	T
tryptophan	W
tyrosine	Y
valine	V

A possible alignment can be

S_1 H E A G A W G H E E - - - - - - -

S_2 - - - - - - - - - - P A W H E A E

Thus the chromosome is formed by a 34-bit binary, the first seventeen-bit representing S_1 and the last seventeen-bit expressing S_2 sequence. This number is obtained due to the quantity of amino acid in S_1 (10) and S_2 (7). A typical chromosome is described in Figure 4.

H	-	E	-	A	-	-	G	A	-	W	G	H	E	-	-	E	P	-	-	-	A	W	H	-	E	-	A	E	-	-	-	-	-
1	0	1	0	1	0	0	1	1	0	1	1	1	1	0	0	1	1	0	0	0	1	1	1	0	1	0	1	1	0	0	0	0	0

Figure 4- GA chromosome description

The fitness function is evaluated based on subset BLOSUM50 substitution matrix (ALTSCHUL, 1991) shown in Table 2.

Table 2- Subset BLOSUM50

	H	E	A	G	A	W	G	H	E	E
P	-2	-1	-1	-2	-1	-4	-2	-2	-1	-1
A	-2	-1	**5**	0	**5**	-3	0	-2	-1	-1
W	-3	-3	-3	-3	-3	**15**	-3	-3	-3	-3
H	**10**	0	-2	-2	-2	-3	-2	**10**	0	0
E	0	**6**	-1	-3	-1	-3	-3	0	**6**	**6**
A	-2	-1	**5**	0	**5**	-3	0	-2	-1	-1
E	0	**6**	-1	-3	-1	-3	-3	0	**6**	**6**

Table 3 illustrates how the S_1 and S_2 fitness sequences are determined taking into account each half of chromosome in GA approach, the BLOSUM50 substitution matrix, and gap. The fitness function evaluated in this case is -2-8+5-8-8-1+0+0-8-8= -54

Table 3- Fitness function

H	E	A	-	-	G	A	W	G	H	E	E
P	-	A	W	H	-	E	-	A	E	-	-
-2	-8	5	-8	-8	-8	-1	0	0	0	-8	-8

Taking into account two chromosomes (Figure 5).

10101001101111001.00110001011000101
01010110110101011.10001110100110000

Figure 5 - Chromosome

The crossover operation mixes both chromosomes (half/half) generating two offspring. Figure 6 describes this process.

10101001101111001.10001110100110000
01010110110101011.00110001011000101

Figure 6 - Crossover

There is a little difference in the GA mutation operation due to the number of amino acids in the chromosome which should always be the same. Thus if mutation occurs in the bit 1 to change to bit 0 the next bit 0 will be modified to bit 1 within each half of the chromosome. The same procedure is applied for mutation in bit 0. This is illustrated in Figure 7 in 3 steps.

10101001101111001.00110001011000101
H-E-A--GA-WGHE--E.--PA---W-HE---A-E

10**0**0100110111100 1.00111**1**001011000101
H--A--GA-WGHE--E.--PAW--W-HE---A-E

10**01**1001101111001.00111**1**00**0**011000101
H--EA--GA-WGHE--E.--PAW----HE---A-E

Figure 7- Mutation

In GA operation a subsequent selection is created from the chromosomes in the current population. The parents (50%) are selected randomly and the genes of the parents are mixed, recombined, and changed (mutation) for the production of offspring in the next generation. The choice of survival members is based on the strongest fitness function to maintain the same population size. This process is repeated in each cycle or generation to hit the best fitness (pairwise alignment).

Results

The population size is kept constant (four members or chromosomes) during the generations and the initial population is shown in Figure 8.

1111111110000000|00000000001111111
HEAGAWGHEE-------
----------PAWHEAE

1111111101000000|00000000010111111
HEAGAWGHE-E------
---------P-AWHEAE

1111111100100000|00000000011011111
HEAGAWGHE--E-----
---------PA-WHEAE

1111111100010000|00000000011011111
HEAGAWGHE---E----
---------PAW-HEAE

Figure 8 - Initial Population

Note that the initial fitness functions for the population are the same for all the members with seventeen gaps (-8*17= -136).

Figure 9 through Figure 11 shows the results from GA approach to get the better alignment (fitness function +1) described in Table 4.

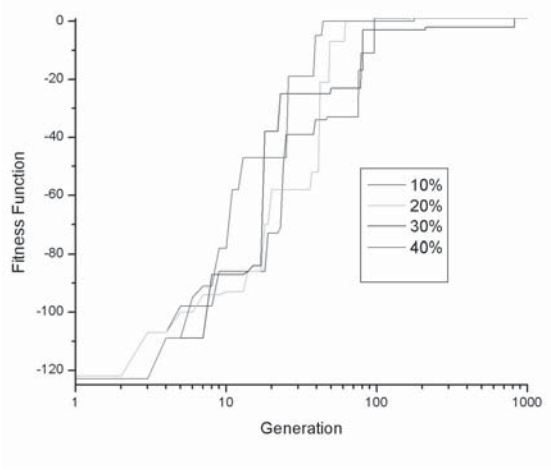

Figure 9- GA pairwise – sample 1

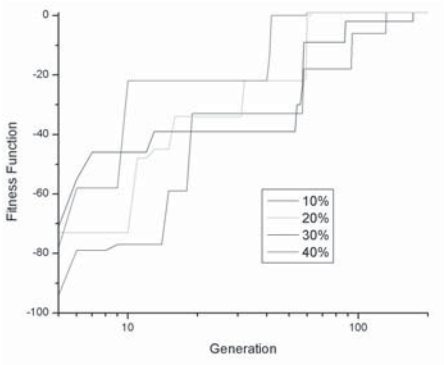

Figure 10- GA Pairwise – Sample 2

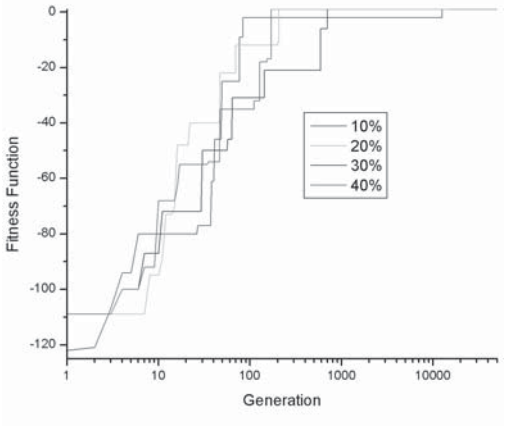

Figure 11- GA pairwise – sample 3

Table 4 - Better Alignment

-	H	E	A	-	G	A	-	W	G	H	-	E	-	E	-	-
-	-	P	A	-	-	-	-	W	-	H	-	E	A	E	-	-

In each sample (1,2 or 3) a different seed is applied to generate a random number in the GA process. Four mutation rates are employed and the best alignment is the same for all mutation rate. Figure 9 presents the worst performance taking into account the number of generations or convergence, near 10000 for rate mutation 10%. Figure 10 requires a lower number of generations, near 1000 for rate mutation 30%. Figure 11 describes the best performance for generation size, in the range of 100 for rate mutation 30%. These behaviors are associated with biology diversity, that is, there is high population homogeneity for rate mutation 10% in sample 1 (Figure 9) while the population diversity in sample 3 (Figure 11) is significant for all mutation rates.

Conclusion

This article described an alternative method based on Genetic Algorithm (GA) to find the optimal pairwise alignment. It performed the alignment sequence taking into account the chromosome and it was associated with a fitness function from BLOSUM50 substitution matrix. The results showed that the convergence in the best pairwise alignment is associated with the population diversity.

References

ALTSCHUL S. F**.** Amino acid substitution matrices from an information theoretic perspective. *Journal of Molecular Biology*, London, v.219, n.3, p.555-565, jun. 1991

HOLLAND J. H. *Adaptation in natural and artificial systems*. Cambridge: Massachussets Institute of Tecnology, 1975.

NEEDLEMAN S. B.; WUNSCH C. D. A general method applicable to search for similarities in amino acid sequence of 2 proteins. *Journal of Molecular Biology*, London, v.48, n.3, p.443-453, 1970.

SMITH T. F.; WATERMAN**,** M. S. Identification of common molecular subsequences. *Journal of Molecular Biology*, London, v.147, n.1, p.195-197,1981.