Multiple Sequence Alignment using a GLOCSA Guided Genetic Algorithm

Edgar D. Arenas-Díaz
IIMAS - UNAM
Circuito exterior s/n, C.U.
Mexico D.F.
earenasd@uxmcc2.iimas.unam.mx

Helga Ochoterena-Booth
Instituto de Biología - UNAM
Circuito exterior s/n, C.U.
Mexico D.F., Mexico
helga@ibiologia.unam.mx

Katya Rodríguez-Vázquez
IIMAS - UNAM
Circuito exterior s/n, C.U.
Mexico D.F., Mexico
katya@uxdea4.iimas.unam.mx

ABSTRACT

This paper introduces GLOCSA as a new scoring function to rate multiple sequence alignments. It is intended to be simple, considering the whole alignment at once and reflecting the parsimony of an alignment. Then, a GLOCSA Guided Genetic Algorithm is proposed in order to refine alignments previously generated by MUSCLE. The results so far are depicted in this paper.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and genetics

General Terms

Algorithms

Keywords

genetic algorithms, biological applications, sequence alignment, autoadaptation, GLOCSA

1. INTRODUCTION

1.1 Sequence Alignment

DNA sequences, RNA sequences and the protein sequences encoded change through time, evolving, mainly under the action of mutation. The simplest types of mutation are point mutations, which are *substitutions* of nucleotides or aminoacids, and insertions/deletions, also known as *indels*. To align two or more sequences, the sequences are put together in a matrix. With a sequence in each line of the matrix, the process of aligning them, represents the insertion of "−" instances in the sequences (see table 2). In order to choose the *best* alignment, it is considered that, in biological terms, the process of alignment has the objective to align homologous residues (having the same evolutionary origin). Assuming that evolution is parsimonious when performing an alignment, it is also sought to minimize the number of evolutionary changes (events of substitutions or indels) that the alignment implies.

To address the problem of sequence alignement different approaches have been developed, starting from dynamic programming algorithms such as Needleman-Wunsch [4] and Smith-Waterman algorithms [6] which are only used in pairwise alignments due to their scalability constrains. Using these *exact* algorithms the heuristics known as progressive alignments were devised. MUSCLE [2] is a very efficient progressive alignment method highly used in the community. Other approaches have been used, such as Hidden Markov Models [3] and Simulated Annealing. Evolutionary Computation has also been used, from Genetic Algorithms [5] to Evolutionary Programming [1].

2. GLOCSA

The Global Criterion for Sequence Alignment (GLOCSA) is a new proposed function to assess the quality of multiple sequence alignments of DNA. It has been built from the ground up with simplicity in mind and rating the alignment as a whole, not taking pairs of sequences to score their corresponding alignment separately. It also takes into account gaps, seeking to favor parsimony.

GLOCSA is composed of three individual criteria, *Mean Column Homogeneity (MCH)* , *Gap Concentration (GC)* and *Columns Increment (CI)*. These are combined in a polynomial with a set of corresponding weights (w_{mch}, w_{gc} and w_{ci}, respectively).

$$GLOCSA = w_{mch}MCH + w_{gc}GC + w_{ci}CI \qquad (1)$$

These weights are set by default to the following values: $w_{mch} = 1000$, $w_{gc} = 20$ and $w_{ci} = -20$. These default values were determined empirically, adjusting them to assing a better scores to better alignments.

At the moment it is intended to rate only multiple sequences of DNA composed of the standard IUB/IUPAC codifications for nucleic acids (including polymorphisms, which are ambiguous codifications for two or three bases used when it cannot be precisely determined which base is there) with the addition of "−" to indicate gaps and "?" to signal that no reading was done. These are shown in table 1.

Table 1: Nucleic Acid Codifications Supported

A	Adenosine	K	G or T	H	A or C or T
C	Cytosine	M	A or C	V	G or C or A
G	Guanine	S	G or C	N	any base
T	Thymine	W	A or T	-	gap
R	G or A	B	G or T or C	?	any base
Y	T or C	D	G or A or T		or gap

To score an alignment of multiple sequences it is considered as a matrix with C columns, being C the maximum number of positions in a sequence, and S lines, where S is the number of sequences in the alignment. At the end of the sequences shorter than the longest one, gap positions are appended ("$-$") to fit all the sequences perfectly in the matrix.

2.1 Mean Column Homogeneity

In the alignment matrix each position is represented in a column, and the column homogeneity has the purpose of rating the grade of diversity in the elements of a given position scoring higher the more homogeneous columns.

The basic idea is, first, counting the occurrences of each of the four bases. A,C,G and T are counted with a weigth of 1.0 while polymorphisms are counted as an equal fraction of a unit for each base they represent (e.g. A counts 1.0 for A while R is either G or A, so it counts 0.50 for G and 0.50 for A). Gaps are also counted, with a slightly smaller weight (i.e. 0.7), in order to favor columns with less gaps. After counting, the column homogeneity of a given column is computed using the following formula,

$$CH_j = \frac{\sum_{t=0}^{4}(wc_{jt})^2}{\left(\sum_{t=0}^{4} wc_{jt}\right)^2} \qquad (2)$$

where wc_{jt} is the (weighted) count of the base t at the column j, and $t = \{0, 1, 2, 3, 4\}$ being $0 = “A''$, $1 = “C''$,$2 = “G''$,$3 = “T''$,$4 = “-''$.

In the case that a position in a sequence has the "?" codification, that sequence is discarded (as it was not there) for the computing of that column homogeneity value. This because a ? implies that in that position the sequence has no information. An special consideration is taken when all the elements in a column are gap codifications ($-$), in that case the column homogeneity is given a value of zero, to penalize the existence of such columns.

When the column homogeneity value for all the columns has been computed , the mean value is obtained and that is the *Mean Column Homogeneity*. This criterion gives higher scores to more homogeneous columns, penalizing diversity of bases in a column. Therefore it is intended to favor more aligned matrices.

2.2 Gap Concentration

Contiguous gap codifications are grouped into gap blocks; as they are in the representation of individuals. Gap Concentration is the mean size of the gap blocks divided by the total number of single gap codifications: $GC = \frac{\overline{S_{GB}}}{GP}$, where $\overline{S_{GB}}$ is the mean size of the gap blocks, and GP is the number of gap positions in the alignment.

This criterion serves the purpose of rewarding the alignments where the gap codifications are located in a more concentrated manner, i.e. where there are fewer larger blocks of gap codifications rather than more blocks of smaller length.

2.3 Columns Increment

It is common that the number of columns in an alignment increases while inserting gaps. *Columns Increment* is the ratio of this increment, defined by $CI = \frac{C}{C_0} - 1$, where C is the number of columns after aligning, and C_0 the number of columns before aligning (the number of nucleotides of the longest sequence). Larger alignments are not generally prefered, and this criterion is intended to penalize them.

Table 2: Alignment Matrix Example

sequence-#								
0	A	A	-	-	A	A	A	A
1	A	A	-	-	A	-	A	A
2	-	-	A	A	A	A	-	-

Table 3: GA Representation Example

sequence-#	
0	$[2, 2]$
1	$[2, 2]$, $[3, 1]$
2	$[0, 2]$

3. GGGA - GA IMPLEMENTATION

GGGA, *GLOCSA Guided Genetic Algorithm* is the Genetic Algorithm implemented to optimize the GLOCSA score. GGGA is a genetic algorithm where a custom representation is proposed, along with a specific mutation operator. There is no crossover operator, selection is performed by tournament and elitism is used. The initialization of the population is done using the mutation operator and a seed alignment (generated with MUSCLE [2] 3.6) which is an input to the algorithm. To produce each individual of the next generation, an individual is selected from the previous generation, using the tournament selection operator, and then submitted to the mutation operator to generate the new individual (under a mutation probability).

3.1 Representation of Individuals

Each individual in the population represents a possible alignment for the sequences. The alignment matrix (described in 2, e.g. in table 2) used to rate an alignment with GLOCSA is the base for the representation of individuals.

But not everything in the matrix is necessary to reconstruct any given alignment of a set of sequences. The only information needed are the position and size of the gap blocks (contiguous gap codifications) in the alignment, because the other sequence information (base codifications) do not change with the alignments. To determine the position of a gap block the following consideration is made: if the bases in every sequence of the alignment are indexed with consecutive numbers, starting from 0 for the first base to ease its implementation, the position of the gap blocks can be determined by the base index it precedes.

Thus, the alignment can be represented by having for each sequence a list of the positions and sizes of every gap block in them, i.e.: each gap block represented as two non-negative integers (position and size).

As a simple illustrative example the alignment matrix of table 2 is transformed to its corresponding representation in table 3. In this example, the sequence 0 has only one gap block of size 2, before the A with index 2 (the third one), hence the list of gap blocks for this sequence only has one element which is $[2, 2]$; sequence 1 has two gap blocks $[2, 2]$, $[3, 1]$; and sequence 2 has only one $[0, 2]$, the two gap codifications at the end of the sequence were appended to fit it in the alignment matrix, so there is no need to include them in the representation (trailing gaps are a consequence of the different lengths of the sequences).

3.2 Mutation Operator and Suboperators

The mutation operator is basically in charge of changing the gap codification appearances in the alignment represented by an individual, in order to explore the solution space. It works with a mutation probability, which determines the number of mutations per individual that will be performed.

For each mutation operation five types of changes to the gap codification appearances are proposed: insertion of new gap blocks, increment of the size of a gap block, decrease of the size of a gap block, shift of positions of gap blocks and deletion of a gap block. These five types of changes are denominated *suboperators*, and the selection of which one will be applied is determined by its probability, which is dynamically adapted throughout the generations. These suboperators were selected because, in the opinion of the autors, they make the algorithm capable of searching the solution space in a relatively efficient way. A crossover operator was also considered but was discarded in early stages because it gave no apparent advantage to the algorithm.

3.2.1 *Insertion Suboperator*

This suboperator chooses randomly a taxa, and inserts a gap block in it. The size of the new gap block is also random, but in a certain range and biased towards smaller sizes. The position of this new gap is determined at random. The size of the new gap blocks to insert is biased towards small sizes because large gap blocks are not very common, but still exist. The method to determine the size of the new gap block is not discussed here due to space constraints.

3.2.2 *Increment Suboperator*

The Increment Suboperator chooses a taxa at random, and increases the size of an existing gap block in one unit. If the selected taxa does not have any gap block at all, this operator does nothing.

3.2.3 *Decrease Suboperator*

As the previous operator, it chooses randomly a taxa, and a gap block from it, whose size will decrease by one; if the size is 1 gap codification, this operator deletes the gap block totally. Again if the selected taxa does not have any gap block at all, this operator does nothing.

3.2.4 *Shift Suboperator*

In a taxa chosen at random, this operator selects first a gap block in it, then a position is selected randomly in that taxa, if a gap block exists in that position,the sizes of them are interchanged. If there is not a gap in the selected position, the position of first gap selected is set to the other position. If the selected taxa does not have any gap block at all, this operator does nothing.

3.2.5 *Deletion Suboperator*

This operator selects randomly a taxa, and then a gap block. This gap block is completely deleted from the list of gap blocks. If no gap block exist in the selected taxa, no operation is done.

3.2.6 *Adaptation of Mutation Suboperators Probability*

The probability of applying each of the subopertors is dynamically adapted as the algorithm evolves (starting all with equal probabilities). It is changed accordingly to their effect in the GLOCSA score of the alignments represented by the individuals, giving more probability to the more benefical ones (or less damaging ones if it is the case). This adaptation is done once at the end of every generation. Due to space constraints the details are not explained in this paper.

Table 4: Test Bench

	# of seq.	max. # of pos.	total # of bases
exmpl19	19	649	10908
exmpl17	17	649	10149
exmpl29	29	245	6150

3.3 Population Initialization

To initialize the population a given alignment is used as a starting point. The individuals of the initial generation are mutations of it, obtained by applying the mutation operator. The mutation operator is applied discarding the adaptation stage, therefore the five suboperators have the same probability while initializing the population.

4. TESTS WITH REAL DATA

4.1 Test Bench

To test the ability of GGGA to optimize the GLOCSA scoring function, three multiple sequence alignment problems were proposed, which are shown in table 4 along with relevant information. The set of sequences *exmpl17* is a subset of *exmpl19*, the two shortest sequences were eliminated, thus presumably reducing the complexity of the alignment.

4.2 GA Test Parameters

Each set of sequences was first aligned with MUSCLE [2] 3.6, a popular progressive alignment tool. The resulting alignment was seeded as a starting point for the initialization of the population, thus the aim of the test is to see if further improvements to the alignment of MUSCLE can be performed, guided by the GLOCSA scoring function. The genetic algorithm for all the experiments was run over 1000 generations with a population of 100, with 5 individuals of elitism. Selection is performed using a tournament of 5 individuals. GLOCSA, the objective function, uses the default weights defined in section 2. The rate of the mutation was in the range of $[0.1, 3]$ with increments of 0.1, the rate of mutation is in units that represent mutations per alignment. For each of these combination of values (the previously mentioned parameters and the mutation rate) 30 experiments were performed.

4.3 Experiments results

Results of these experiments are shown in figures 1, 2 and 3. It was observed that the GLOCSA Guided Genetic Algorithm always improved (at least slightly) the solution previously found by MUSCLE, and as expected the amount of improvement is strongly related with the *mutation rate*; lower (near zero) and higher (close and beyond three mutations per alignment) mutation rates produce less improvements while values in or in the vicinity of the range of $[0.5, 1.0]$, produce the higher optimization values. This trend can be observed more clearly in the mean value of the scores.

Figure 1:

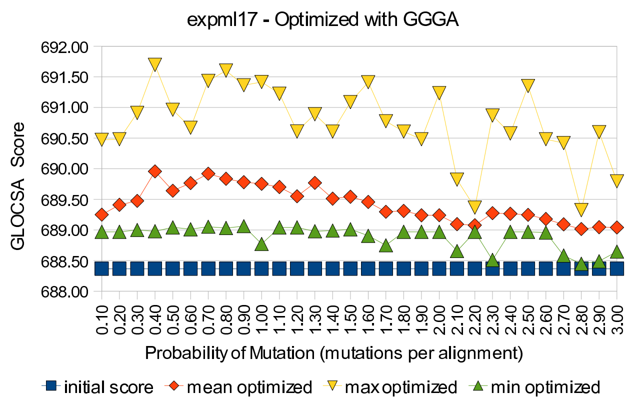


Figure 2:

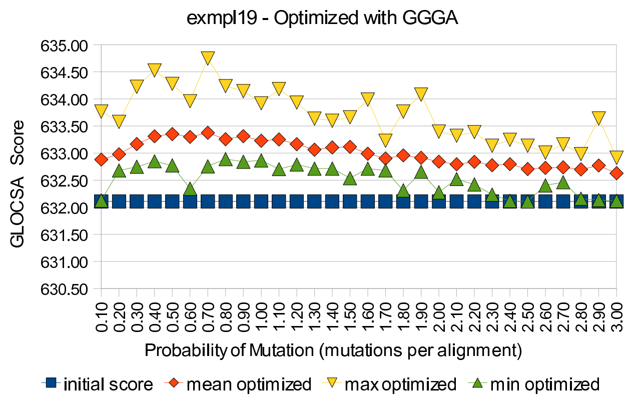
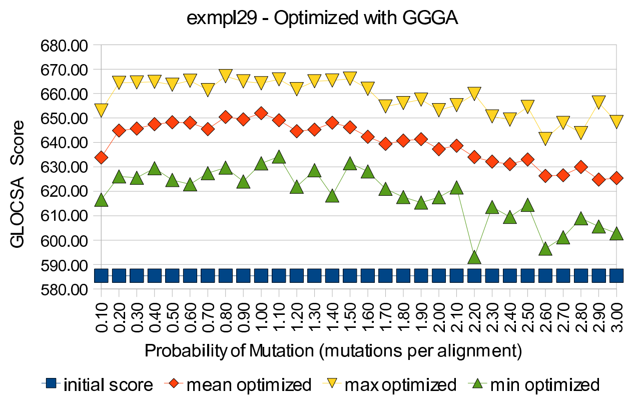


Figure 3:



5. CONCLUSIONS

For the assessment of the quality of multiple sequence alignments, scoring functions have been previously defined, but in the opinion of the authors, the results obtained so far are not satisfactory enough, and therefore the GLOCSA measure was devised. It aims to be considered an alternative scoring function for multiple sequence alignments, one which is simple, rates the whole alignment at once, and is parsimonious.

Given the complexity of the problem of multiple sequence alignment, the techniques of Evolutionary Computation - Genetic Algorithms in particular - seem useful for optimizing this new proposed scoring function. Although it is not efficient in terms of computing time, compared with the fast progressive alignment heuristics, the GGGA has the ability to optimize GLOCSA as the objective function. In the light of performing it as a refinement over previously aligned data with more efficient methods, as in the test experiments where MUSCLE alignments were inserted as starting points, is a promising application.

6. FUTURE WORK

Currently GLOCSA only rates DNA sequence alignments, the next step would be to extend its application scope to protein sequences.

GLOCSA as a quality measure has been validated empirically, but tests to assess its reliability are still pending. This will be done with the aid of defined sets of reference alignments such as BALiBASE [7] and GGGA. Thus resulting in the assessment of both, the scoring function and the genetic algorithm implementation.

7. ACKNOWLEDGMENTS

The authors would like to thank CONACYT, SIBA-UNIBIO (UNAM) and the PAPIIT project IN115806-3 for the funding that made this work possible.

8. REFERENCES

[1] L. Cai, D. Juedes, and E. Liaknovitch. Evolutionary computation techniques for multiple sequence alignment. In *Proceedings of the IEEE Congress on Evolutionary Computation 2000*, 2000.

[2] R. C. Edgar. Muscle: multiple sequence alignment with high accurracy and high throughput. *Nucleic Acids Reseach*, 32(5):1792–1797, 2004.

[3] A. Krogh, M. Brown, I. Mian, Sjölander, and D. Haussler. Hidden markov models in computational biology: applications to protein modeling. *Journal of Molecular Biology*, 235:1501–1531, 1994.

[4] S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

[5] C. Notredame and D. Higgins. Saga: sequence alignment by genetic algorithm. *Nucleic Acids Research*, 1996.

[6] T. Smith and M. Waterman. Comparision of biosequences. *Adv. Appl. Math.*, 2:483–489, 1981.

[7] J. Thompson, F. Plewniak, and O. Poch. Balibase: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 15:87–88, 1999.