

VOLUME 10, NUMBER 1
MARCH, 2009

ISSN 1595-6881

African Scientist



**Journal of the Nigerian Society for
Experimental Biology**

KLOBEX

AfricanScientist

Published by Klobex Academic Publishers for the Nigerian Society for Experimental Biology

EDITOR

Professor C. O. Bewaji, Department of Biochemistry, University of Ilorin, Nigeria

EDITORIAL ADVISORY BOARD

J. I. Anetor (Ibadan)
E. O. Ayalogu (PH)
E. A. Bababunmi (N.Y.)
A. A. Bakare (Ibadan)
E. A. Balogun (Ilorin)

A. A. Biu (Maiduguri)
J. Kayode (Ado-Ekiti)
A. A. Odewumi (Zaria)
J. A. Ogunwale (Ilorin)
Z. S. C. Okoye (Jos)

F. A. Oladele (Ilorin)
G. Olaoye (Ilorin)
J. O. Olowookere (Ikenne)
E. E. A. Oyedunmade (Ilorin)
L. T. Zaria (Maiduguri)

AIMS AND SCOPE

The journal will publish original research papers and review articles which are of interest to scientists and the general public on the continent of Africa in particular and the rest of the world in general. It will also highlight the work of African Scientists, past and present, in various parts of the world.

SUBSCRIPTION INFORMATION

African Scientist is published quarterly by Klobex Academic Publishers, Ilorin, Nigeria. For libraries, University Departments, Research Institutes and other multi-reader Institutions, the subscription rate is as follows:

Volume 10 (2009) 4 issues: £35/\$50/₦6,000

(The quoted price does not include the cost of postage, packing and handling)

Specially reduced subscription rates: In order to maximize the dissemination of the scholarly articles published in this journal, individuals are encouraged to take a personal subscription to the journal at a specially reduced rate of ₦1,200 or equivalent per copy + ₦600 for postage and handling.

Subscription enquiries should be addressed to:

Klobex Academic Publishers, 7 Fasada Shopping Complex, Tanke Tipper Garage,
University Road, P. O. Box 4216, Ilorin, Nigeria.

© 2009 Klobex Academic Publishers

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, electrostatic, magnetic tape, mechanical, photocopying, recording or otherwise, without the prior written permission of the publisher, Klobex Academic Publishers, University Road, P. O. Box 4216, Ilorin, Nigeria.

It is a condition of publication that manuscripts submitted to this journal have not been published and will not be simultaneously submitted or published elsewhere. Upon acceptance of the article, the author(s) will be requested to transfer copyright of the article to the publisher. Such transfer will encourage the widest possible dissemination of information.

AFS 2009002/10101	
Evaluation of solid waste management policy in Benin metropolis, Edo State, Nigeria	
C. Kalu, W. W. Modugu and I. Ubochi.....	1
AFS 20080772/10102	
On efficiency of sequence alignment algorithms	
T. O. Oladele, O. M. Bamigbola and C. O. Bewaji.....	9
AFS 2009004/10103	
Growth performance and survival of <i>Heterobranchus longifilis</i> , <i>Heterobranchus bidorsalis</i> and their reciprocal hybrids	
A. A. Akinwande, F. O. Moody and S. O. Umar.....	15
AFS 2009007/10104	
Land snail populations in shade and full-sun cocoa plantations in South Western Nigeria, West Africa	
O. C. Oke and J. U. Chokor.....	19
AFS 2009010/10105	
<i>In vitro</i> antistrongyline efficacy of neem (<i>Azadirachta indica</i> Juss) leaf aqueous extract in Maiduguri, Nigeria	
A. A. Biu and O. Anthonia.....	31
AFS 2009011/10106	
Pediculosis among residents of Maiduguri, Borno State, Nigeria	
A. A. Biu, I. A. Gulani, and M. Kaka.....	35
AFS 2009022/10107	
<i>Eichhornia crassipes</i> invasion and physicochemical characteristics of a creek flowing through an urban area in Southern Nigeria	
J. O. Olomukoro, G. A. Osunde and C. N. Azubuike.....	39
AFS 2009023/10108	
Macroinvertebrate colonisation of artificial substrates in a Nigerian river III: Cement bricks, ceramic tiles and macrophytes	
J. O. Olomukoro and O. Eloghosa.....	53

AFS 2008077/10102

On efficiency of sequence alignment algorithms

T. O. Oladele¹, O. M. Bamigbola² and C. O. Bewaji³

¹Department of Computer Science, University of Ilorin, Ilorin, Nigeria.

²Department of Mathematics, University of Ilorin, Ilorin, Nigeria.

³Department of Biochemistry, University of Ilorin, Ilorin, Nigeria.

(Received December 31, 2008)

ABSTRACT: The alignment of sequences is a mutual arrangement of two or more sequences which exhibit their similarities and where they differ.

Sequence alignment is usually used to study the evolution of the sequences from a common ancestor, especially biological sequences such as protein sequences or deoxyribonucleic acid (DNA) sequences. Sequence alignment can also be used to study the evolution of languages and the similarity between texts.

In this paper, we discuss sequence alignment algorithms for finding similarities between sequences, homologues (relatives) on a gene or gene-product in genomic databases. This information is useful for answering a variety of biologically related questions.

Key Words: Sequence algorithms; Bioinformatics; Protein sequences; DNA sequences

1. INTRODUCTION

Sequence alignment has become the central tool for sequence comparison in molecular biology. The typical assumption in the use of alignment is the mechanism of molecular evolution. DNA carries over genetic material from generation to generation, by virtue of its semi-conservation duplication mechanism. Change in the material are introduced by occasional errors and mutations in the duplication, and by viruses and other mechanisms which sometimes move sub-sequences within the chromosome and between individuals.

In bioinformatics, a sequence alignment is a way of arranging the primary sequences of DNA, RNA or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences (Lesk, 2002). Aligned sequences of nucleotide or amino acid residues are represented as rows within a matrix. Gaps are inserted between the residues so that residues with identical or similar characters are aligned in successive columns.

Very short or similar sequences can be aligned by hand. Most interesting problems require the alignment of lengthy, highly variable or extremely numerous sequences that cannot be aligned solely by human effort. Instead, human knowledge is primarily applied in constructing algorithms to produce high-quality sequence alignments, and occasionally, in adjusting the final results to reflect patterns that are difficult to represent algorithmically. A lot of computational algorithms, such as Basic Local Alignment Search Tool (BLAST) and Fast Alignment (FASTA) have been applied to sequence alignment problem,

including slow but formally optimizing methods like dynamic programming and efficient heuristic or probabilistic methods designed for large-scale database searches.

People wonder when the terms similarity and homology are used indiscriminately. Similarity simply means that sequences are in some sense similar and have no evolutionary connotations. Homology refers to evolutionary related sequences stemming from a common ancestor.

The ability to calculate the correct alignment is crucial to many types of studies. Consequently, an alignment between sequences may be employed in tracing the common ancestor which contained the matching subsequences.

In life sciences vast quantities of data including nucleotide and amino acid sequences are stored in a database. These sequence data help biologists to determine the chemical structure, biological function, and evolutionary history of organisms. The main feature of managing the exponential growth in sequence data sources is the availability of fast, sensitive and statistically rigorous techniques for detecting similarities between these sequences.

As the amount of nucleotide and amino acid sequence data continues to grow, the data becomes increasingly useful in the analysis of newly sequenced genes and proteins because of the greater chance of finding such sequence similarity. The National Center for Biotechnology Information (NCBI) developed one of the commonly used versions of Basic Local Alignment Search Tool (BLAST).

2. PREAMBLE

For ease of comprehension we define the basic terms and concept involved with the topic.

A sequence is an arrangement of objects, letters, or numbers, for examples

a e i o u	-	a sequence of vowels
1 3 5 7 9	-	a sequence of one-digit odd numbers
A G G T C C T	-	a DNA sequence
L C V L R	-	a protein sequence.

Each of the above examples constitutes a string. As such, a string is a sequence of characters with the first character being a letter. Examples are:

a e i o u, G 3 5 7 9, A G G T C C T

Blank spaces (-) may be included in string formation, such as

G - T C G - C T

More formally, a string is an ordered list of characters written contiguously (as a sequence) from left to right. Note that the position or location of each character in the strings is important. For the string:

a e i o u, the 1st character is a, the second e, and the third i, etc.

Sequence Alignment has to do with comparing and contrasting the arrangement of characters of strings for the purpose of finding patterns of matches and mismatches.

FASTA and BLAST are two related programs that are nearly universally used to approximate local alignment and local similarity. Both of these programs are suites of programs which contain variants used in different problem domains (such as searching DNA versus Searching protein, searching a database versus aligning two strings etc). Both FASTA and BLAST are based on the same kinds of ideas. They are both based on Smith-Waterman algorithm.

Of all the sequence alignment algorithms, BLAST is most widely used. It is typically used to compare one query nucleotide or protein sequence against a database of sequences, and show similarities and sequence matches. Its success and popularity comes from its combination of speed, sensitivity, and statistical assessment of results.

FASTA is just a fast approximation to the Smith-Waterman algorithm, it is slower and more sensitive than the BLAST algorithm because FASTA tolerates gaps (dash) in the aligned sequences.

This paper presents a general review of the strengths and weaknesses of BLAST and FASTA programs which are popularly used for sequence alignment. Work is in progress to test the efficiency of both algorithms with respect to their sensitivity and the statistical significance of their respective outputs.

3. TYPES OF ALIGNMENTS

There are three types of alignment in existence namely: pairwise, structural and multiple alignments.

3.1 PAIRWISE ALIGNMENT

Pairwise sequence alignment methods are concerned with finding the best – matching piecewise (local) or global alignments of protein (amino acid) or DNA (nucleic acid) sequences.

The purpose of pairwise alignment is to find homologues (relatives) of a gene or gene-product in a database of known examples. The most important application of pairwise alignment is identification of sequences of unknown structure or function and also used in the study of molecular evolution. There are two types of pairwise alignment; they are Local and Global alignment.

3.1.1 LOCAL ALIGNMENT

This method produce alignments of only segments of two sequences. It finds related regions within sequences - in other words they can consist of a subset of the characters within each sequence. For example, positions 20 - 40 of sequence A might be aligned with positions 50 - 70 of sequence B. This is a more flexible technique than global alignment and has the advantage that related regions which appear in a different order in two proteins sequences (which is known as domain shuffling) can be identified as being related. This is not possible with global alignment methods.

3.1.2 GLOBAL ALIGNMENT

A global alignment between two sequences is an alignment in which all the characters in both sequences participate in the alignment. Global alignments are useful mostly for finding closely-related sequences.

3.2 STRUCTURAL ALIGNMENT

One potential way to validate sequence alignments is to use structural alignments where possible. Because protein structure is more conserved through evolution than the sequence, structural alignments can be more reliable over long evolutionary distances, when the sequences have diverged so much that simple sequence comparison cannot detect their similarity. However, although structural alignments are useful, they are completely artifactual.

3.3 MULTIPLE ALIGNMENT

This is an extension of pairwise alignment which incorporates more than two sequences into an alignment. Multiple alignment method aligns all the sequences in a specified set.

4. MEASURES OF SEQUENCES ALIGNMENT

It is pertinent to state at this junction that there are basically two approaches to aligning sequences. One approach deterministic process while the order hinge on stochastic or probabilistic measures.

In this paper sequence alignment is discussed using the first approach since another paper centered on the use of suffix trees is scheduled for publication elsewhere (manuscript in preparation). We now define some quantitative measures of sequence similarity and difference.

The distance between any two strings S1 and S2 - can be measured in terms of Hamming distance and/or Levenshtein distance (Gusfield, 1997).

1. Hamming distance: This is the number of positions with mismatching characters in the strings S1 and S2. E.g. with S1 = a g t c t and S2 = c g t a t, Hamming distance = 2. The characters in the 2nd, 3rd and 5th positions correspondingly match in the two strings while those in the 1st and 4th positions are not the same letters (mismatch). Hamming distance for S1 and S2 = 2.

2. Levenshtein or edit distance: The Levenshtein or edit distance between strings S1 and S2 (not necessarily of the same length) is the minimal number of edit operations required to change string S1 into S2. An edit operation is a deletion, insertion or substitution (replacement) of a single character in either string. e.g.

Let S1 = a, g - t c c
S2 = c g c t - a

To transform S1 and S2, replace the 1st character in S1 by c, then insert c into the 3rd character in S1 (the character - is known as the gap), deletion of the 5th character in S1 followed by substitution of the 6th character in S1 by a. Thus, resulting in the edit operations shown below:

String S1: a g - t c c
String S2: c g c t - a

Edit operations: R M I M D R

∴ Levenshtein distance = 3

This sequence of edit operations for the transformation is referred to as an edit transcript. Essentially, Hamming and Levenshtein distances are measures of dissimilarity between strings. Relatively similar sequences are expected to give smaller values for Hamming and Levenshtein distances than widely dissimilar sequences.

For two strings S1 and S2, let $d(i, j)$ denote the minimum number of edit operations required to transform the first i characters of S1, i.e. $S1[1 \dots i]$, into the first j character of S2, i.e. $S2[1 \dots j]$. If indeed, S1 and S2 have n and m characters respectively, then the edit distance between S1 and S2 is precisely the value $d(n, m)$.

The issue in sequence alignment, which is known as the edit distance problem, thus boils down to computing the edit distance between two given strings together with an optimal edit transcript that describes the transformation. An optimal edit transcript is an edit transcript that utilizes the minimum number of edit operations.

It is easy to use pen and paper method to align short sequences as is exhibited in the examples above. Knowing very well that the length of the genome of most organisms run into several millions of DNA and protein sequences, more efficient algorithms are required in order to carry out global pairwise alignment of the sequences.

Most of the algorithms in existence for determining global optimal sequence alignment are based on the mathematical technique called Dynamic programming.

5. SEQUENCE ALIGNMENT ALGORITHMS

An algorithm is a set of logical steps for solving a problem. Below is stated the basic features of some of the algorithms in existence for sequence alignment. It is pertinent to repeat that, in general, they incorporate the dynamic programming approach.

5.1 NEEDLEMAN-WUNSCH ALGORITHM

This is a pairwise global alignment algorithm. Needleman-Wunch uses a **global alignment algorithm**, which mean that the full sequence must be aligned and gaps at the end of the sequence are penalized as much as internal gaps. It is a rigorous algorithm which is based on dynamic programming. It is simple to implement but slow and not used for searching.

5.2 SMITH-WATERMAN ALGORITHM

Smith-Waterman uses a local alignment algorithm, which mean that it searches for both full and partial sequence matches. Gaps at the end of the sequence essentially have a penalty of zero. It is a rigorous algorithm which is based on dynamic programming. It is fairly simple to implement but slow and not used for searching.

5.3 FRAMESEARCH ALGORITHM

The Framesearch algorithm is an extension of Smith-Waterman method, for pairwise alignment between a protein sequence and a nucleotide sequence.

6. SOFTWARE FOR SEQUENCE ALIGNMENT

The following are some of the standard softwares available for sequence alignment. Mostly, they implement the above-listed algorithms. The codes of the package are generously made available on the internet.

6.1 SSEARCH

This is based on the dynamic programming technique. It implements the standard Smith-Waterman algorithm. It is considerably slower than the more modern BLAST and FASTA methods. However, Smith Waterman remains the golden standard for protein – protein or nucleotide – nucleotide pairwise alignment; its speed can be improved by using specialized hardware or a computer cluster.

6.2 BLAST (Basic Local Alignment Search Tool)

BLAST (Basic Local Alignment Search Tool) employs a heuristic algorithm to search a sequence database for the closest matches to a test sequence (Altschul et al. 1990).

This method uses a pre-computed hash table to serve as an index for short sequences. Given a query sequence, the subsequences are looked up in the index to reduce the amount of time and searching involved. Several parameters need to be provided to make this method faster or more accurate. Once, patterns that match the search sequence are found, more accurate and intensive algorithms may be applied.

BLAST uses a pairwise local search and uses a number of methods to increase the speed of the original Smith-Waterman algorithm.

Due to the high speed and sensitivity of the algorithm and the concurrent availability of up-to-date sequence database, BLAST is the most widely used computer program for database searches that has been developed. When a new gene sequence of unknown function has been obtained in the laboratory, BLAST is usually first used to determine it.

If this sequence is related to any other sequences in the existing databases. BLAST will find related sequences with a very high degree of reliability.

6.3 FASTA (Fast Alignment)

This method is based on pairwise local search. It is much slower but more sensitive to finding sequence similarities than BLAST (Henikoff and Henikoff 1992) and it is more suitable for nucleic acid sequence searches. Recent versions of the FASTA suite include specialized algorithms for translated searches that can align low-quality nucleotide sequence to protein sequence (Pearson and Lipman, 1988; Pearson, 1990; Pearson, 1991).

6.4 CLUSTAL

Clustal algorithms are based on progressive multiple alignment methods. They come in several varieties such as Clustal W, Clustal X etc.

7. COMPARISON BETWEEN BLAST AND FASTA SEARCHES

Almost immediately upon its release in 1990, BLAST became the dominant searching engine for biological sequence databases. The initial reasons for its success were speed, it outputs a range of solutions, and that each match is accompanied by an estimate of statistical significance. Since the introduction of BLAST, FASTA has evolved and now also outputs multiple solutions and estimates of statistical significance. Nonetheless, BLAST remains dominant. We are currently working on a quantitative comparison between the two sequence search engines.

REFERENCES

- Altschul S. F., Gish W., Myers E. W., Lipman D. J. (1990), Basic local alignment search tool. *Journal of Molecular Biology*, 215:403-410.
- Gusfield, D. (1997), *Algorithms on strings, Trees and sequences*. Cambridge University press, New York. Pp. 376-379.
- Henikoff, S. and Henikoff, J. G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*. 89: 10915 – 10919.
- Lesk, A.M. (2002), *Introduction to Bioinformatics*. Oxford University Press, Oxford.
- Pearson, W. R. (1990) Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods in Enzymology* 183: 63 – 98.
- Pearson, W. R. (1991) Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithm.
- Pearson, W. R. and Lipman, D. J. (1988) Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*. 85: 2444 – 2448.