

Software

Open Access

MBEToolbox: a Matlab toolbox for sequence data analysis in molecular biology and evolution

James J Cai*¹, David K Smith², Xuhua Xia³ and Kwok-yung Yuen¹

Address: ¹Department of Microbiology, University of Hong Kong, Pokfulam, Hong Kong, China, ²Department of Biochemistry, University of Hong Kong, Pokfulam, Hong Kong, China and ³Department of Biology, University of Ottawa, Canada

Email: James J Cai* - jamescai@hkusua.hku.hk; David K Smith - dsmith@hkucc.hku.hk; Xuhua Xia - xxia@uottawa.ca; Kwok-yung Yuen - kyyuen@hkucc.hku.hk

* Corresponding author

Published: 22 March 2005

Received: 03 December 2004

BMC Bioinformatics 2005, 6:64 doi:10.1186/1471-2105-6-64

Accepted: 22 March 2005

This article is available from: <http://www.biomedcentral.com/1471-2105/6/64>

© 2005 Cai et al; licensee BioMed Central Ltd.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/2.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Background: MATLAB is a high-performance language for technical computing, integrating computation, visualization, and programming in an easy-to-use environment. It has been widely used in many areas, such as mathematics and computation, algorithm development, data acquisition, modeling, simulation, and scientific and engineering graphics. However, few functions are freely available in MATLAB to perform the sequence data analyses specifically required for molecular biology and evolution.

Results: We have developed a MATLAB toolbox, called MBEToolbox, aimed at filling this gap by offering efficient implementations of the most needed functions in molecular biology and evolution. It can be used to manipulate aligned sequences, calculate evolutionary distances, estimate synonymous and nonsynonymous substitution rates, and infer phylogenetic trees. Moreover, it provides an extensible, functional framework for users with more specialized requirements to explore and analyze aligned nucleotide or protein sequences from an evolutionary perspective. The full functions in the toolbox are accessible through the command-line for seasoned MATLAB users. A graphical user interface, that may be especially useful for non-specialist end users, is also provided.

Conclusion: MBEToolbox is a useful tool that can aid in the exploration, interpretation and visualization of data in molecular biology and evolution. The software is publicly available at <http://web.hku.hk/~jamescai/mbetoolbox/> and http://bioinformatics.org/project/?group_id=454.

Background

MATLAB integrates programming, visualization and computation in an easy-to-use environment and is widely used in scientific and engineering studies. One of the most attractive features of MATLAB is that the basic data element of the system is a matrix that does not require dimensioning. This allows users to solve many technical computing problems, especially those with matrix and

vector formulations, in a very effective way. The MATLAB environment itself offers a comprehensive set of built-in functions and many toolboxes have been developed, and are often freely available, for more specialized needs.

However, to our knowledge, these advantages in the MATLAB environment have not been fully utilized in the area of molecular biology and evolution. Only a few MATLAB

toolboxes or functions are freely available for data analysis, exploration, and visualization of nucleotide and protein sequences. MATHWORKS has recently provided a bioinformatics toolbox, however this toolbox has relatively limited functions for molecular evolutionary studies. MBEToolbox, is presented here to fulfil the most obvious needs in sequence manipulation, genetic distance estimation and phylogeny inference under the MATLAB environment. Moreover, this toolbox provides an extensible, functional framework to formulate and solve problems in evolutionary data analysis. It facilitates the rapid construction of both general applications, as well as special-purpose tools for evolutionary biologists, in a fraction of the time it would take to write a program in a scalar, noninteractive language such as C or FORTRAN.

Implementation

MBEToolbox is written in the MATLAB language and has been tested on the WINDOWS platform with MATLAB version 6.1.0. The main functions implemented are: sequence manipulation, computation of evolutionary distances derived from nucleotide-, amino acid- or codon-based substitution models, phylogenetic tree construction, sequence statistics and graphics functions to visualize the results of analyses. Although it implements only a small fraction of the multiplicity of existing methods used in molecular evolutionary analyses, interested users can easily extend the toolbox.

Input data and formats

MBEToolbox requires a single ASCII file containing the nucleotide or amino acid sequence alignment in either PHYLIP [1], CLUSTALW [2] or fasta format. The toolbox does provide a built-in CLUSTALW [2] interface if an unaligned sequence file is provided. Protein-coding DNA sequences can be automatically aligned based on the corresponding protein alignment with the command `alignseqfile`.

After input, in common with the MATHWORKS bioinformatics toolbox, MBEToolbox represents the alignment as a numeric matrix with every element standing for a nucleic or amino acid character. Nucleotides A, C, G and T are converted to integers 1 to 4, and the 20 amino acids are converted to integers 1 to 20. A header, containing information about the names and type of the sequences as well as the relevant genetic code for protein-coding nucleotides, is attached to the alignment matrix to form a MATLAB *structure*. An example alignment structure, `aln`, in MATLAB code follows:

```
aln =
```

```
seqtype: 2
```

```
geneticcode: 1
```

```
seqnames: {1 × n cell}
```

```
seq: [n × m double]
```

where n is the number of sequences and m is the length of the aligned sequences. The type of sequence is denoted by 1, 2 or 3 for sequences of non-coding nucleotides, protein coding nucleotides and amino acids, respectively.

Sequence manipulation and statistics

The alignment structure, `aln`, can be manipulated using the MATLAB language. For example, `aln.seq(x,:)` will extract the x th sequence from the alignment, while `aln.seq(:, [i: j])` will extract columns i to j from the alignment. Users may easily extract more specific positions by using functions developed in the toolbox, such as `extractpos(aln, 3)` or `extractdegeneratesites` to obtain the third codon positions or fourfold degenerate sites, respectively. For each sequence, some basic statistics such as the nucleotide composition (`ntcomposition`) and GC content, can be reported. Other functions include the calculation of the relative synonymous codon usage (RSCU) and the codon adaptation index (CAI), counts of segregating sites, taking the reverse complement or translating a sequence, and determining the sequence complexity.

Evolutionary distances

The evolutionary distance is one of the important measures in molecular evolutionary studies. It is required to measure the diversity among sequences and to infer distance-based phylogenies. MBEToolbox contains a number of functions to calculate evolutionary distances based on the observed number of differences. The formulae used in these functions are analytical solutions of a variety of Markov substitution models, such as JC69 [3], K2P [4], F84 [1], HKY [5] (see [6] for detail). Given the stationarity condition, the most general form of Markov substitution models is the General Time Reversible (GTR or REV) model [7-10]. There is no analytical formula to calculate the GTR distance directly. A general method, described by Rodriguez et al. [9], has been implemented here. In this method a matrix F , where F_{ij} denotes the proportion of sites for which sequence 1 (s_1) has an i and sequence 2 (s_2) has a j , is formed. The GTR distance between s_1 and s_2 is then given by

$$\hat{d} = -\text{tr}(\Pi \log(\Pi^{-1}F))$$

where Π denotes the diagonal matrix with values of nucleotide equilibrium frequencies on the diagonal, and $\text{tr}(A)$ denotes the trace of matrix A . The above formula can be expressed in MATLAB syntax directly as:

```
>> d = -trace(PI*logm(inv(PI)*F))
```

MBEToolbox also calculates the gamma distribution distance and the LogDet distance [11] (i.e., Lake's paralinear distance [12]).

For alignments of codons, the toolbox provides calculation or estimation of the synonymous (K_s) and non-synonymous (K_a) substitution rates by the counting method of Nei and Gojobori [13], the degenerate methods of Li, Wu and Luo [14] and the method of Li or Pamilo and Bianchi [15,16], as well as the maximum likelihood method through PAML [17]. All these methods for calculating K_s and K_a require that the input sequences are aligned in the appropriate reading frame, which can be performed by the function `alignseqfile`. Unresolved codon sites will be removed automatically. In addition, several quantities, including the number of substitutions per site at only synonymous sites, at only non-synonymous sites, at only four-fold-degenerate sites, or at only zero-fold-degenerate sites can be calculated. The output from these calculations are distance matrices which can be exported into text or excel files, or used directly in further operations.

Phylogeny inference

Two distance-based tree creation algorithms, Unweighted Pair Group Method with Arithmetic mean (UPGMA) and neighbor-joining (NJ) [18] are provided and trees from these methods can be displayed or exported. Maximum parsimony and maximum likelihood algorithms can be applied to nucleotide or amino acid alignments through an interface to the `phylip` package [1]. As properly implemented maximum likelihood methods are the best vehicles for statistical inference of evolutionary relationships among species from sequence data, several maximum likelihood functions have been explicitly implemented in MBEToolbox. These functions allow users to incorporate various evolutionary models, estimate parameters and compare different evolutionary trees.

The simplest case of estimation of the evolutionary distance between two sequences, s_1 and s_2 , can be considered as the estimation of the branch length (the number of substitutions along a branch) separating ancestor and descendent nodes. Branch lengths, relative to a calibrated molecular clock, can reveal the time interval for this separation. A continuous time Markov process is generally used to model evolution along the branch from s_1 to s_2 . A *transition rate matrix*, Q , is used to indicate the rate of changing from one state to another. For a specified time interval or distance, t , the *transition probability matrix* is calculated from $P(t) = e^{Qt}$. If there are N sites, the full likelihood is

$$L = \prod_{i=1}^N \pi_{s_i^1} P(s_i^1 \rightarrow s_i^2, t)$$

In this equation, s_i^1 and s_i^2 are the i th bases of sequences 1 and 2 respectively; $\pi_{s_i^1}$ is the expected frequency of base s_i^1 .

In MBEToolbox, to calculate the likelihood, L , at a given time interval (or distance) t , we have to specify a substitution model by using an appropriate model defining function, such as `modeljc`, `modelk2p` or `modeljtr` for non-coding nucleotides, `modeljtt` or `modeldayhoff` for amino acids, or `modelgy94` for codons. These functions return a model structure composed of an *instantaneous rate matrix*, R , and an equilibrium frequency vector, π which give Q , ($Q = R \cdot \text{diag}(\pi)$). Once the model is specified, the function `likelidist(t, model, s1, s2)` can calculate the log likelihood of the alignment of the two sequences, s_1 and s_2 , with respect to the time or distance, t , under the substitution model, *model*.

In most cases we wish to estimate t instead of calculating L as a function of t , so the function `optimlikelidist(model, s1, s2)` will search for the t that maximises the likelihood by using the Nelder-Mead simplex (direct search) method, while holding the other parameters in the model at fixed values. This constraint can be relaxed by allowing every parameter in the model to be estimated by functions, such as `optimlikelidistk2p`, that can estimate both t and the model's parameters. Figure (1a and 1b) illustrates the estimation of the evolutionary distance between two ribonuclease genes through the fixed- and free-parameter K2P models, respectively. When the K2P model's parameter, *kappa*, is fixed, the result and trace of the optimisation process is illustrated by the graph of L and t (Fig. 1a). When *kappa* is a free parameter, a surface shows the result and trace of the optimisation process (Fig. 1b).

When calculating the likelihood of a phylogenetic tree, where s_1 and s_2 are two (descendant) nodes in a tree joined to an internal (ancestor) node, s_a , we must sum over all possible assignments of nucleotides to s_a to get the likelihood of the distance between s_1 and s_2 . Consequently, the number of possible combinations of nucleotides becomes too large to be enumerated for even moderately sized trees. The *pruning algorithm* introduced by Felsenstein [19] takes advantage of the tree topology to evaluate the summation in a computationally efficient (but mathematically equivalent) manner. This and a simple and elegant mapping from a 'parentheses' encoding of a tree to the matrix equation for calculating the likelihood

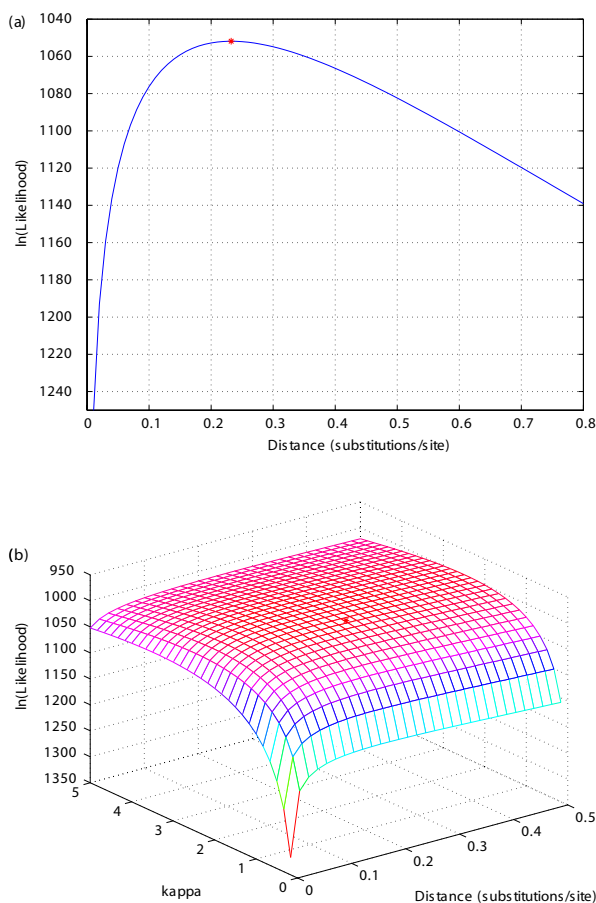


Figure 1
Log-likelihood of evolutionary distance. (a) Likelihood as function of K2P distance. The distance is estimated by maximising the likelihood of the alignment with the bias of transitions to transversions, κ , held fixed. (b) Likelihood as a function of distance and κ . Both the distance and κ are optimised simultaneously. The maximum likelihood peaks are marked with *. The two sequences used are the coding regions of Tamarin eosinophil-derived neurotoxin (Acc. No.: U24099) and human eosinophil cationic gene (Acc. No: NM_002935).

of a tree, developed in the MATLAB software, PHYLLAB [20], have been adopted in likelitree.

Combination of functions

Basic operations can be combined to give more complicated functions. A simple combination of the function to extract the fourfold degenerate sites with the function to calculate GC content produces a new function (countgc4) that determines the GC content at 4-fold degenerate sites

(GC4). A subfunction for calculating synonymous and nonsynonymous differences between two codons, getsyn-nonsyndiff, can be converted into a program for calculating codon volatility [21] with trivial effort. Similarly, karlinsig which returns Karlin's genomic signature (the dinucleotide relative abundance or bias) for a given sequence can be easily re-formulated to estimate relative *di-codon* frequencies, which may be a new index of biological signals in a coding sequence. In addition, the menu-driven user interface, MBEGUI, is also a good example illustrating the power of combination of basic MBEToolbox functions.

Graphics and GUI

Good visualisation is essential for successful numerical model building. Leveraging the rich graphics functionality of MATLAB, MBEToolbox provides a number of functions that can be used to create graphic output, such as scatterplots of K_s vs K_a , plots of the number of transitions and transversions against genetic distance, sliding window analyses on a nucleotide sequence and the Z-curve (a 3-dimensional curve representation of a DNA sequence [22]). A simple menu-driven graphical user interface (GUI) has been developed by using GUIDE in MATLAB. The top menu includes File, Sequences, Distances, Phylogeny, Graph, Polymorphism and Help submenus (Fig. 2). It aids the usage of the most frequently required functions so that users do not have to run any scripts or functions from the MATLAB command line in most cases.

Results and discussion

Vectorization simplifies programming

MATLAB is a matrix language, which means it is designed for vector and matrix operations. Programming can be simplified and made more efficient by using algorithms that take advantage of *vectorization* (converting for and while loops to the equivalent vector or matrix operations). The MATLAB compiler in version 7.0 will automatically recognize and vectorize loops without recursion. An example of vectorization is the calculation of Z-scores [23] for Smith-Waterman alignments [24] to give a measure of the significance of an alignment score against a background of scores from randomly generated sequences with the same composition and length. Hence, Z-scores are designed to overcome the bias due to the composition of the alignment and are usually calculated by comparing an actual alignment score with the scores obtained on a set of random sequences generated by a Monte-Carlo process. The Z-score is defined as:

$$Z(A, B) = (S(A, B) - \text{mean}) / \text{standard deviation}$$

where $S(A, B)$ is the Smith-Waterman (S-W) score between two sequences A and B . The mean and standard deviation are taken from realignments of the permuted

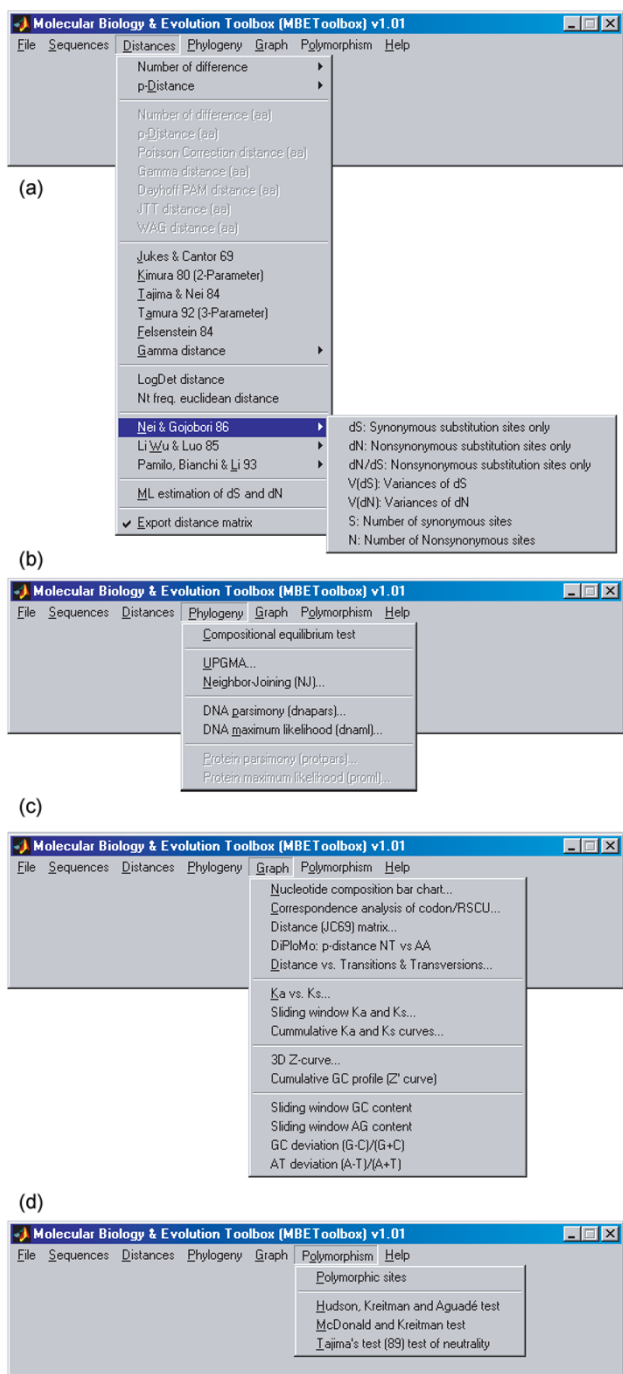


Figure 2
MBETToolbox GUI. (a) Distances submenu; (b) Phylogeny submenu; (c) Graph submenu; and (d) Polymorphism submenu.

sequences. The algorithm is implemented as follows in MATLAB with as few as 15 lines of code:

```

function [z,z_raw] = zscores(s1,s2,nboot)

m1 = length(s1);
m2 = length(s2);

% Initialise two vectors holding Z-score of
% s1_rep and s2_rep, i.e., replicate samples
% of sequences s1 and s2.

v_z1 = zeros(1,nboot);
v_z2 = zeros(1,nboot);

z_raw = smithwaterman(s1,s2);

for (k = 1:nboot),

    s1_rep = s1(:,randperm(m1));

    v_z1(1,k) = smithwaterman(s1_rep, s2) ;

    s2_rep = s2(:,randperm(m2));

    v_z2(1,k) = smithwaterman(s1, s2_rep);

end

z1 = (z_raw-mean(v_z1))./std(v_z1);
z2 = (z_raw-mean(v_z2))./std(v_z2);

z = min(z1,z2);
    
```

where randperm(n) is a vector function returning a random permutation of the integers from 1 to n and smithwaterman performs local alignment by the standard dynamic programming technique.

Extensibility

An important distinction between compiled languages with subroutine libraries and interactive environments like MATLAB is the ease with which problems can be specified and solved in the latter. Moreover, MATLAB toolboxes are traditionally organised in a less object-oriented mode and, consequently, functions are more independent of each other and easier to combine and extend. Several examples were given in the Implementation section.

Comparison with other toolboxes

Some other toolboxes have been developed in MATLAB for bioinformatics related analyses. These include PHYL-LAB [20] and MATARRAY [25] as well as the

bioinformatics toolbox developed by MATHWORKS. Other examples can be found at the link and file exchange maintained at MATLAB CENTRAL [26]. PHYLLAB is a molecular phylogeny toolbox which also provides some functions for sequence and tree input and manipulation. Its main focus is on creating a maximum likelihood tree based on Bayesian principles using a Markov chain Monte Carlo method to compute posterior parameter distributions. MATARRAY is focussed on the analysis of gene expression data from microarrays and provides normalization and clustering functions but does not address molecular evolution. The bioinformatics toolbox from MATHWORKS provides a range of bioinformatics functions, including some related to molecular evolution.

MBEToolbox provides a much broader range of molecular evolution related functions and phylogenetic methods than either the more specialized PHYLLAB project or the more general bioinformatics toolbox from MATHWORKS. These extra functions include IO in PHYLIP format, statistical and sequence manipulation functions relevant to molecular evolution (e.g. count segregating sites), evolutionary distance calculation for nucleic and amino acid sequences, phylogeny inference functions and graphic plots relevant to molecular evolution (e.g. K_a vs K_s). As such it makes an important contribution to the bioinformatics analyses that can be performed in the MATLAB environment.

A novel enhanced window analysis

To test for the selective pressures in the different lineages of a phylogenetic tree, the nonsynonymous to synonymous rate ratio (K_a/K_s) is normally estimated [27-29]. Values of $K_a/K_s = 1, > 1$, or < 1 indicate neutrality, positive selection, or purifying selection, respectively. However, K_s and K_a are measurements of average synonymous and nonsynonymous substitutions per site along the whole length of the sequences. Average K_s and K_a values give neither the pattern of intragenic fluctuation of selective constraints, nor region- or site-specific information. A sliding window method is usually adopted to examine the *intra-genic* pattern of the substitution rates and to test for the occurrence of significant clusters of variant regions [30-33]. Significant heterogeneity in K_s would indicate that the neutral substitution rate varies across the gene, whereas heterogeneity in K_a may indicate that selective constraints vary along the gene. The results and accuracy of sliding window methods, either overlapping or non-overlapping, depend on both the size of the window and the moving distance adopted. Large window lengths may obliterate the details of patterns in K_s or K_a , whereas small window lengths usually result in larger statistical fluctuations. Hence, the resolution of a sliding window is usually limited.

A mathematical formalism, similar to the Z'-curve [34], is introduced here to solve this problem. Consider a subsequence based analysis of K_s or K_a . In the n -th step, count the *cumulative* numbers of K_s or K_a occurring from the first to the n -th nucleotide position in the gene sequences being inspected. Let \mathbb{K} denote either K_s or K_a and $\mathbb{K}^{(n)}$ denote the cumulative \mathbb{K} at the n -th sequence position. $\mathbb{K}^{(n)}$ is usually an approximately mono-increasing linear function of n . The points $(\mathbb{K}^{(n)}, n)$, $n = 1, 2, \dots, N$ are fit by a least square method to a linear function, $f(\mathbb{K}^{(n)}) = \beta n$, to give a straight line with β being its slope. We define

$$\mathbb{K}'^{(n)} = \mathbb{K}^{(n)} - \beta n$$

The two-dimensional curve of $(\mathbb{K}'^{(n)} \sim n)$ gives an alternative representation of the normal sliding window curve.

To compare these two curve representations, the example dataset of Suzuki and Gojobori [35], which contains the coding regions of two hepatitis C virus strains (HCV-JS – Genbank Acc.: D85516 and HCV-JT – Genbank Acc.: D11168), was used. The entire coding sequence is divided into eight regions (C, E1, E2, NS2, NS3, NS4, NS5A, NS5B). Some of the coding regions have been combined as these short ORFs are unlikely to yield meaningful K_s and K_a values. The reduction of K_s in the C, E1 and NS5B regions, as well as its elevation in NS3, which have been shown in previous studies [35], are not clear in a standard sliding window representation (Fig. 3a). In contrast a sharp increase in the $(\mathbb{K}'^{(n)} \sim n)$ curve (Fig. 3b), indicates an increase in \mathbb{K} , while a drop in the curve indicates a decrease in \mathbb{K} . This new method has been implemented in the function plotSlidingKaKs. Since it is derived from the sliding window method, it is called the enhanced sliding window method.

Limitations

The current version of this toolbox lacks novel algorithms yet it implements a variety of existing algorithms. There are some limitations in the practical use of MBEToolbox. First, though the toolbox provides many methods to infer and handle sequence and evolutionary analyses, the full range of these features can only be accessed through the MATLAB command line interface, as in the majority of MATLAB packages. Second, some of the functions cannot handle ambiguous nucleotide or amino acid codes in the sequences. The future development of MBEToolbox will overcome these present limitations.

Conclusion

The MBEToolbox project is an ongoing effort to provide an easy-to-use, yet powerful, analysis environment for

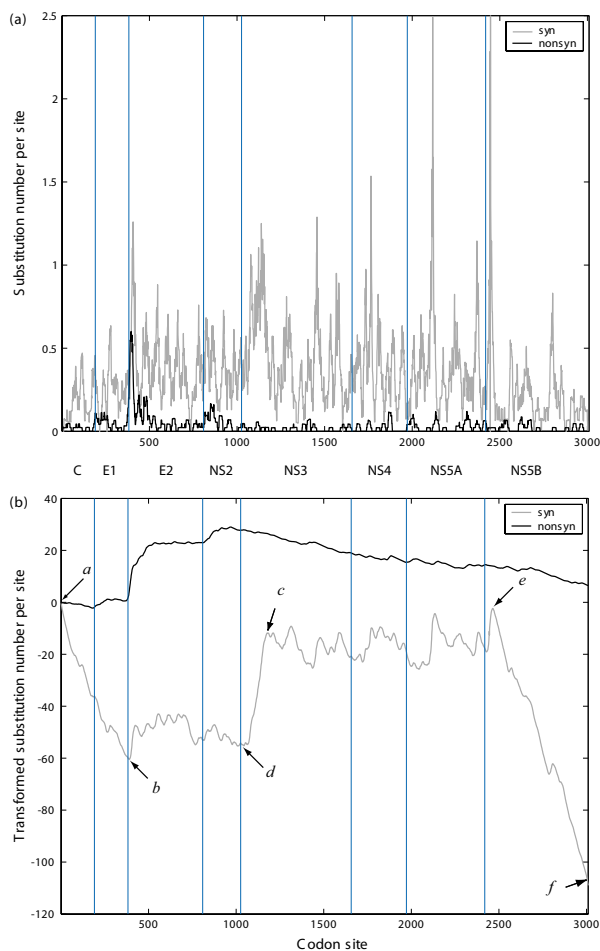


Figure 3

A comparison between sliding window and enhanced sliding window methods. Sliding window analysis of K_s and K_a for the concatenated coding regions of two hepatitis C virus strains, HCV-JS and HCV-JT. The number of codons for the C, E1, E2, NS2, NS3, NS4, NS5A, and NS5B genes are 191, 192, 426, 217, 631, 315, 447, and 591, respectively. The different coding regions are separated by vertical lines. (a) illustrates the result of a normal sliding window analysis; (b) illustrates the result of the enhanced sliding window analysis. Beginnings and ends of regions poor in synonymous substitutions ($slope < 0$) are indicated by the arrows **a** and **b** (genes C and E1) and **e** and **f** (gene NS5B). A region rich in synonymous substitutions ($slope > 0$) in gene NS3 is indicated by arrows **c** and **d**.

molecular biology and evolution. Currently, it offers a substantial set of frequently used functions to manipulate sequences, to calculate genetic distances, to infer phylogenetic trees and related analyses. MBEToolbox is a useful

tool which should inspire evolutionary biologists to take advantage of the MATLAB environment.

Availability and requirements

Project name: MBEToolbox

Project web page: http://web.hku.hk/~jamescai/mbe_toolbox/

http://bioinformatics.org/project/?group_id=454

Operating system: WINDOWS 95/98/2000/XP

Programming language: MATLAB 6.0 or higher

Other requirements: Statistics Toolbox

License: GPL

Any restrictions on use by non-academics: License needed

Authors' contributions

JJC designed and implemented the software and wrote the draft of the manuscript. DKS participated in the design and revised the manuscript. XX participated in the design and provided suggestions for future development. KYY supervised and participated in the design of the study. All authors read and approved the final version of the manuscript.

Acknowledgements

This work was supported by the AIDS Trust Fund (MSS 083), Research Grant Council Grant (HKU 7363/03M), and University Development Fund, University of Hong Kong. JJC would like to thank Dr. Nam-Kiu Tsing (Department of Mathematics, University of Hong Kong) for valuable discussions, and Dr. Richard E. Strauss (Department of Biological Sciences, Texas Tech University) for allowing his NJ routine to be adapted for MBEToolbox and for releasing his MATLAB library to the public.

References

1. Felsenstein J: **PHYLIP – Phylogeny Inference Package (Version 3.2)**. *Cladistics* 1989, **5**:164-166.
2. Thompson J, Higgins D, Gibson T: **CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice**. *Nucl Acids Res* 1994, **22**:4673-4680.
3. Jukes TH, Cantor C: **Evolution of protein molecules**. In *Mammalian Protein Metabolism* Edited by: Munro HN. New York: Academic Press; 1969:21-132.
4. Kimura M: **A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences**. *J Mol Evol* 1980, **16**:111-120.
5. Hasegawa M, Kishino H, Yano T: **Dating of the human-ape splitting by a molecular clock of mitochondrial DNA**. *J Mol Evol* 1985, **22**:160-174.
6. Nei M, Kumar S: *Molecular evolution and phylogenetics* Oxford, UK: Oxford University Press; 2000.
7. Lanave C, Preparata G, Saccone C, Serio G: **A new method for calculating evolutionary substitution rates**. *J Mol Evol* 1984, **20**:86-93.

8. Tavaré S: **Some probabilistic and statistical problems in the analysis of DNA sequences.** *Lectures on Mathematics in the Life Sciences* 1986, **17**:57-86.
9. Rodríguez F, Oliver JL, Marin A, Medina JR: **The general stochastic model of nucleotide substitution.** *J Theor Biol* 1990, **142**:485-501.
10. Yang Z: **Estimating the pattern of nucleotide substitution.** *J Mol Evol* 1994, **39**:105-111.
11. Steel MA: **Recovering a tree from the leaf colourations it generates under a Markov model.** *Appl Math Lett* 1994, **7**:19-32.
12. Lake JA: **Reconstructing evolutionary trees from DNA and protein sequences: paralinear distances.** *Proc Natl Acad Sci USA* 1994, **91**:1455-1459.
13. Nei M, Gojobori T: **Simple methods for estimating the numbers of synonymous and nonsynonymous nucleotide substitutions.** *Mol Biol Evol* 1986, **3**:418-426.
14. Li WH, Wu CI, Luo CC: **A new method for estimating synonymous and nonsynonymous rates of nucleotide substitution considering the relative likelihood of nucleotide and codon changes.** *Mol Biol Evol* 1985, **2**:150-174.
15. Li WH: **Unbiased estimation of the rates of synonymous and nonsynonymous substitution.** *J Mol Evol* 1993, **36**:96-99.
16. Pamilo P, Bianchi NO: **Evolution of the Zfx and Zfy genes: rates and interdependence between the genes.** *Mol Biol Evol* 1993, **10**:271-281.
17. Yang Z: *Phylogenetic Analysis by Maximum Likelihood (PAML). Version 3.0* London: University College; 2000.
18. Saitou N, Nei M: **The neighbor-joining method: a new method for reconstructing phylogenetic trees.** *Mol Biol Evol* 1987, **4**:406-425.
19. Felsenstein J: **Evolutionary trees from DNA sequences: a maximum likelihood approach.** *J Mol Evol* 1981, **17**:368-376.
20. Rzhetsky A, Morozov P: **Markov chain Monte Carlo computation of confidence intervals for substitution-rate variation in proteins.** *Pac Symp Biocomput* 2001, **6**:203-214.
21. Plotkin JB, Dushoff J, Fraser HB: **Detecting selection using a single genome sequence of M. tuberculosis and P. falciparum.** *Nature* 2004, **428**:942-945.
22. Zhang R, Zhang CT: **Z curves, an intuitive tool for visualizing and analyzing the DNA sequences.** *J Biomol Struct Dyn* 1994, **11**:767-782.
23. Pearson WR, Lipman DJ: **Improved tools for biological sequence comparison.** *Proc Natl Acad Sci U S A* 1988, **85**:2444-2448.
24. Smith TF, Waterman MS: **Identification of common molecular subsequences.** *J Mol Biol* 1981, **147**:195-197.
25. Venet D: **MatArray: a Matlab toolbox for microarray data.** *Bioinformatics* 2003, **19**:659-660.
26. **MATLAB Central** [<http://www.mathworks.com/matlabcentral/>]
27. Sharp PM: **In search of molecular darwinism.** *Nature* 1997, **385**:111-112.
28. Akashi H: **Within- and between-species DNA sequence variation and the 'footprint' of natural selection.** *Gene* 1999, **238**:39-51.
29. Crandall K, Kelsey C, Imamichi H, Lane H, Salzman N: **Parallel evolution of drug resistance in HIV: failure of nonsynonymous/synonymous substitution rate ratio to detect selection.** *Mol Biol Evol* 1999, **16**:372-382.
30. Clark AG, Kao T: **Nonsynonymous Substitution at Shared Polymorphic Sites Among Self-Incompatibility Alleles of Solanaceae.** *Proc Natl Acad Sci USA* 1991, **88**:9823-9827.
31. Ina Y: **ODEN: a program package for molecular evolutionary analysis and database search of DNA and amino acid sequences.** *Comput Appl Biosci* 1994, **10**:11-12.
32. Endo T, Ikeo K, Gojobori T: **Large-scale search for genes on which positive selection may operate.** *Mol Biol Evol* 1996, **13**:685-690.
33. Choi SS, Lahn BT: **Adaptive evolution of MRG, a neuron-specific gene family implicated in nociception.** *Genome Res* 2003, **13**:2252-2259.
34. Zhang CT, Wang J, Zhang R: **A novel method to calculate the G+C content of genomic DNA sequences.** *J Biomol Struct Dyn* 2001, **19**:333-341.
35. Suzuki Y, Gojobori T: **Analysis of coding sequences.** In *The phylogenetic handbook: a practical approach to DNA and protein phylogeny* Edited by: Salemi M, Vandamme A. Cambridge, UK: Cambridge University Press; 2003:283-311.

Publish with **BioMed Central** and every scientist can read your work free of charge

"BioMed Central will be the most significant development for disseminating the results of biomedical research in our lifetime."

Sir Paul Nurse, Cancer Research UK

Your research papers will be:

- available free of charge to the entire biomedical community
- peer reviewed and published immediately upon acceptance
- cited in PubMed and archived on PubMed Central
- yours — you keep the copyright

Submit your manuscript here:
http://www.biomedcentral.com/info/publishing_adv.asp

