ELSEVIER Computational Statistics & Data Analysis 40 (2002) 285–291

COMPUTATIONAL STATISTICS & DATA ANALYSIS

www.elsevier.com/locate/csda

Fast computation of maximum likelihood trees by numerical approximation of amino acid replacement probabilities

T. Pupko[a,*], D. Graur[b]

[a] *The Institute of Statistical Mathematics, 4-6-7 Minami Azabu, Minato ku, Tokyo, Japan*
[b] *Department of Zoology, Faculty of Life Sciences, Tel-Aviv University, Tel-Aviv, Israel*

Received 1 June 2001; received in revised form 1 December 2001; accepted 1 December 2001

Abstract

Phylogenetic reconstruction methods seek to find the tree topology that best describes the evolutionary history of a set of DNA or protein sequences. Maximum likelihood (ML) based models were shown to be superior to distance and parsimony methods. However, applications of ML models are severely limited in the number of sequences they can handle because of computational limitations. A speedup for calculating the replacement probabilities between any two amino acid states is presented. An empirical 29-fold speedup is achieved. The speedup is based on the Chebyshev polynomial series. In theory, the likelihood scores of the tree are only approximated, but in practice the likelihood values computed are essentially identical to those estimated by exact computation.

Keywords: Maximum likelihood; Chebyshev polynomial; Molecular evolution

1. Introduction

Maximum likelihood (ML) models based on amino acid sequences are frequently used in evolutionary studies. Computer programs, such as MOLPHY (Adachi and Hasegawa, 1995) PUZZLE (Strimmer and von Haeseler, 1996) and PAML (Yang, 1997), implement these models but are limited in the number of taxa they can handle because of computational limitations. A major component of the ML computational effort involves the estimation of branch lengths—the average number of character

* Corresponding author.
E-mail address: tai@ism.ac.jp (T. Pupko).

0167-9473/02/$ - see front matter
PII: S0167-9473(02)00008-7

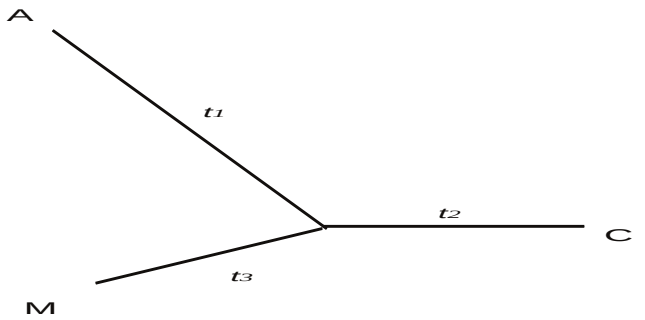

Fig. 1. Unrooted phylogenetic tree for 3 taxa. Letters denote amino acids: A, alanine; C, cysteine; M, methionine. Branch lengths correspond to an average number of amino acid replacements per amino acid site.

changes between two adjacent nodes of the tree. Previously published methods for reducing the computation costs were mainly concerned with nucleotide sequences, and were focused on achieving faster convergence of the maximum likelihood function (Olsen et al., 1994; Rogers and Swofford, 1998). However, stochastic processes for nucleotide sequences are based on 4×4 substitution matrices. When amino acid sequences are concerned, the computations involve 20×20 replacement matrices (between any two amino acids), which substantially increase the computation time.

We implement an approximation method for the calculation of the probability that an amino acid will be replaced by another along a branch of a given length. This numerical approximation saves considerable computation time while at the same time it yields likelihood values that are essentially identical to those estimated by the exact method. The detailed procedures used to evaluate a likelihood of a given phylogenetic tree are given for example in (Adachi and Hasegawa, 1995). For example, the likelihood (L) of the tree in Fig. 1 is

$$L = \sum_{k \in AA} [\pi_k \times p(k \mapsto A \,|\, t_1) \times p(k \mapsto C \,|\, t_2) \times p(k \mapsto M \,|\, t_3)],$$

where k is any of the 20 possible amino acid states (AA), π_k is the frequency of amino acid k, and $p(i \mapsto j \,|\, t)$ is the probability that amino acid i will change into amino acid j along a branch of length t.

The formula for $p(i \mapsto j \,|\, t)$ is derived from the theory of stochastic processes. We assume that X_t is the state of a time-continuous, time homogeneous Markov chain with state space equal to the 20 amino acid states $1, 2, \ldots, 20$ and that the transition probabilities are constant.

$$p(i \mapsto j \,|\, t) = p(X_{t+s} = j \,|\, X_s = i).$$

From the Chapman–Kolmogorov relation

$$p(i \mapsto j \,|\, s + t) = \sum_{k=1}^{20} p(i \mapsto k \,|\, s) \times p(k \mapsto j \,|\, t)$$

an explicit expression for $p(i \mapsto j \mid t)$ can be derived (e.g., Karlin and Taylor, 1975):

$$P(t) = e^{Qt},$$

where $P(t)$ is a 20×20 replacement matrix such as

$$p(i \mapsto j \mid t) = [P(t)]_{i,j}$$

and Q is the rate matrix.

To calculate the matrix exponent, the eigenvectors and eigenvalues of Q are calculated:

$$p(i \mapsto j \mid t) = \sum_{k=1}^{20} U_{ik} U_{kj}^{-1} e^{t\lambda_k},$$

where U is the 20×20 matrix of the right-hand eigenvectors of Q, and λ is the corresponding eigenvalue vector.

It should be noted that it is assumed that the following reversibility condition holds:

$$\pi_i \times p(i \mapsto j \mid t) = \pi_j \times p(j \mapsto i \mid t).$$

It is easy to show that this condition is sufficient for U to be a real matrix. Furthermore, from this assumption it can be shown that the likelihood of the tree is independent of the position of the root. It is this property that makes the dynamic program computation feasible (Felsenstein, 1981).

From the above equations it is obvious that the total time needed to compute amino acid likelihood functions strongly depends on the time required for the computation of the $p(i \mapsto j \mid t)$ values. Thus, reducing the time of likelihood evaluation for each of the many trees that need to be examined in phylogenetic studies is of considerable practical value.

2. Motivation for using the Chebyshev polynomials

Our goal is to develop an approximation to the function $f(t) = P_{ij}(t) = [P(t)]_{i,j}$ in a fixed interval $[0, t_0]$. This problem is approached by using polynomials approximation. Thus, we look for a polynomial that in the interval $[0, t_0]$ has the property that among all other polynomial approximations with the same degree has the minimum maximum error. The polynomial with such a property is called the minimax polynomial, and is generally very difficult to find (e.g., Ralston, 1965). The Chebyshev approximating polynomial is almost identical and is very easy to compute.

The Chebyshev polynomials approximate a function $f(t)$ in terms of interpolation: the polynomial $p(t)$ would have the same values as $f(t)$ in several points $[x_0, \ldots, x_m]$ while in the other points in $[0, t_0]$, $e(t) = |f(t) - p(t)|$ should be minimum. The Taylor approximation, for example, approximates the function in a specific point x_0. The problem of using Taylor series to approximate a function in an interval, is that $e(t)$ for the Taylor polynomial is extremely nonuniform—very small near x_0, but growing very rapidly near the end points. It would seem more reasonable to use as approximation functions, whose behavior over an interval would be in some sense uniform. The Chebyshev polynomials have ideal properties for these aims.

3. The Chebyshev polynomials

Let $x = \cos a$, thus, $a = \arccos x$. Defining $T_n(x) = \cos(n \arccos x)$ we obtain the Chebyshev polynomials. Explicitly, $T_0(x)=1$; $T_1(x)=x$; $T_2(x)=2x^2-1$. It follows that $T_{n+1}(x) = 2 \times T_n(x) - T_{n-1}(x)$, and the leading coefficient is $2n-1$ for $n>0$ and 1 for $n=0$. For a more general description see (Clenshaw, 1962). When we interpolate a function $f(x)$ at the points $(x_0,\ldots,x_m)$ the remainder term has the form

$$\frac{(x-x_0)(x-x_1)\cdots(x-x_m)f^{m+1}(\xi)}{(m+1)!}.$$

This is a special case of the Taylor polynomial when $x_0=x_1=\cdots=x_m$. We search for the best choice of $x_0,\ldots,x_m$ for this interpolation. This is equivalent to asking which $x_0,\ldots,x_m$ will minimize $(x-x_0)(x-x_1)\cdots(x-x_m)$. That this expression is invariant to the choice of $f(x)$ is what makes the Chebyshev polynomial approximation general. It can be shown (e.g., Ralston, 1965) that choosing $x_0,\ldots,x_m$ to be the m root of $T_{m+1}(x)$ has the desired property.

Instead of writing the interpolation polynomial as $\sum_{i=0}^{m} a_i x^i$ we write it as $\sum_{j=0}^{m} c_j T_j(X)$. In this application Chebyshev coefficients are stored in the form of c_j. Using this form usually results in less number of coefficients (and hence, less memory and more speed).

4. Finding the Chebyshev coefficients

Using the orthogonality of the Chebyshev polynomials, it can be shown that the formula for c_j is defined by

$$c_j = \frac{2}{n}\sum_{j=0}^{n} f\left(\cos\frac{\pi j}{n}\right)\cos\frac{\pi r j}{n}.$$

Clenshaw (1962). These Chebyshev polynomials are used to interpolate the function $P_{ij}(t)$. In practice, the c_j's are computed for $n=60$ and a truncated polynomial is used with the desired number of coefficients. Each function from the set $\{P_{ij}(t)\}$ is interpolated separately, resulting in 400 different polynomials. Furthermore, each $P_{ij}(t)$ function is analytic, i.e., these functions have continuous derivatives of order m for all m. For these functions the c_j converge at least as rapidly as a geometric progression (Clenshaw, 1962).

5. Computer implementation

The Chebyshev polynomials approximate the replacement functions in an interval. In which interval should the approximation be valid? Practical values for the t are in the range $[0,1]$. However, not all sites evolve at the same rate, and models that assume among site rate variation are considered to be superior (Yang, 1994). In these models, evolution of a site with rate r is equivalent to evolution along a tree in which all branch

lengths are multiplied by r (Yang, 1994). Thus, even larger range might be appropriate. On the other hand—the bigger the range—the more coefficients are needed in order to achieve the same approximation. Thus, a more efficient approach is to approximate the replacement probabilities in a smaller interval, and when t is outside this interval, to invoke a call to the nonapproximated version of the $p(i \mapsto j \mid t)$ function. The optimum interval might change from one dataset to another, but we found out that a default interval of $[0,1]$ is generally close to the optimal.

A possible problem with the Chebyshev approximation is that approximated probability $p(i \mapsto j \mid t)$ values might not be in the interval $[0,1]$. Hence, the approximated probability values are checked, and if the values are outside the $[0,1]$ interval the nonapproximate version of the $p(i \mapsto j \mid t)$ function is called.

This fast implementation of the Chebyshev approximation was combined in a program for evaluating the likelihood of trees, and finding the ML branch length estimation assuming the rates are gamma distributed among sites (Yang, 1994). Finding the ML branch lengths is computationally expensive, and is an essential step in exact and heuristic searches for the ML tree. Our computation is based on any one of the three widely used substitution matrices: DAY (Dayhoff, 1978), JTT (Jones et al., 1992) and REV24 (Adachi and Hasegawa, 1995). This approximation was also applied in a program for inferring the ML-ancestral amino acid sequences, the amino acid sequences in the internal nodes of the phylogenetic tree (Pupko et al., 2000).

6. An empirical example

As a sample data 71 lysozyme c amino acid sequences were collected from a genebank. The sequences were aligned using ClustalW (Thompson et al., 1974). All positions containing gaps were removed. Lysozyme c sequences were chosen since it has been postulated that some amino acid replacements in the evolutionary history of these sequences provide evidence for positive selection in the lineages leading to foregut-fermentering taxa (Stewart and Wilson, 1987; Zhang and Kumar, 1997).

A phylogenetic tree was constructed using the Molphy software (Adachi and Hasegawa, 1995). To estimate the speedup factor we compared 100 repeats of the likelihood evaluations. Likelihood evaluations were done with the JTT replacement matrix, and assuming among-site rate variation with the gamma rate parameter equal to 0.9335 (ML estimation of the rate parameter).

7. Results

The $p(i \mapsto j \mid t)$ function is used in many likelihood-based computations, e.g., finding the most likely tree topology and branch length (Adachi and Hasegawa, 1995; Strimmer and von Haeseler, 1996; Yang, 1997) ML estimation of ancestral sequences (Yang, 1997; Pupko et al., 2000) and ML-based test for positive selection (Pupko et al., 2001). When evaluating the performance of the polynomial approximation one has to evaluate both the degree of approximation and the speedup factor achieved. As shown in the example in Table 1, the differences between the exact log-likelihood value and the approximated values for approximations with more than six coefficients are trivial

Table 1
Percent deviation from exact log-likelihood values as a function of the number of Chebyshev coefficients. To estimate the speedup factor we compared the computation time of 100 repeats of the likelihood evaluations. Data for likelihood evaluations consist of 71 lysozyme *c* sequences collected from a genebank. Likelihood evaluations were done with the JTT replacement matrix (Jones et al., 1992) assuming among-site rate variation (see text). Running times were computed on a 600 MHz Pentium machine with 256 MB RAM

Number of coefficients	Log likelihood	Percent deviation	Speedup factor
2	−3.685560E + 03	4.62E − 01	49.00
3	−3.668440E + 03	−4.46E − 03	45.00
4	−3.668556E + 03	−1.29E − 03	42.00
5	−3.668601E + 03	−6.71E − 05	39.00
6	−3.668603E + 03	−1.53E − 06	33.00
7	−3.668603E + 03	7.19E − 08	31.00
8	−3.668603E + 03	1.26E − 08	29.00
9	−3.668603E + 03	1.08E − 09	28.00
10	−3.668603E + 03	6.22E − 11	25.00
11	−3.668603E + 03	2.83E − 12	25.00
12	−3.668603E + 03	1.49E − 13	24.00
13	−3.668603E + 03	6.20E − 14	23.00
14	−3.668603E + 03	7.44E − 14	22.00
15	−3.668603E + 03	9.92E − 14	20.00
No.	−3.668603E + 03	0	1

as far as trees with log likelihoods of −3.668E+03 or more are concerned. For practical purposes, we recommend using eight coefficients, which results in a 29-fold speedup in this case.

The degree of speedup depends on the portion of the computation, which is based on $p(i \mapsto j \mid t)$ estimations. In some cases memory is interchangeable with speed: it might be more efficient to store a 20×20 matrix $[P(t)]_{i,j}$ for each branch length of the given tree prior to computing the tree likelihood, thus, avoiding repeated calls for the $p(i \mapsto j \mid t)$ function with the same arguments. However, when memory is scare, or when different rates are assigned to each position, this is inapplicable.

This speedup factor can be combined with previously published approximation methods, such as the five speedup factor method for inference of branch lengths suggested by Rogers and Swofford (1998). This method is based on using parsimonious reconstruction of ancestral sequences at the internal nodes of the tree.

The speedup obtained with this method, while substantial (Table 1), may only enable a slight increase in the number of sequences that can be dealt with in exhaustive maximum likelihood searches (Felsenstein, 1988). However, for purposes other than exhaustive searches, e.g., likelihood comparisons among alternative trees, heuristic ML searches or ancestral amino acid sequence reconstructions, the Chebyshev approximation may spell the difference between “can do” and “can’t”.

8. Uncited References

Kishino et al., 1990; Saito and Nei, 1987.

Acknowledgements

We thank Itsik Pe'er and an anonymous referee for helpful discussions and comments and Giora Unger for his help with the computer program. The first author was supported by a JSPS fellowship. This study was also supported by the Magnet Da'at Consortium of the Israel Ministry of Industry and Trade and a grant from Tel Aviv University.

References

Adachi, J., Hasegawa, M., 1995. MOLPHY: Programs for Molecular Phylogenetics, version 2.3. The Institute of Statistical Mathematics, Tokyo.

Clenshaw, C.W., 1962. Mathematical Tables, Vol. 5, Chebyshev Series for Mathematical Functions. National physical laboratory, London.

Dayhoff, M.O., 1978. Atlas of Protein Sequence and Structure, Vol. 5, Suppl. 3. National Biomedical Research Foundation, Washington, DC.

Felsenstein, J., 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. 17, 368–376.

Felsenstein, J., 1988. Phylogenies from molecular sequences: inference and reliability. Ann. Rev. Genet. 22, 521–565.

Jones, D.T., Taylor, W.R., Thornton, J.M., 1992. The rapid enumeration of mutation data matrices from protein sequences. Comput. Appl. Biosci. 8, 275–282.

Karlin, S., Taylor, H.M., 1975. A First Course in Stochastic Processes, Second Edition. Academic Press, CA.

Kishino, H., Miyata, T., Hasegawa, M., 1990. Maximum likelihood inference of protein phylogeny and the origin of chloroplasts. J. Mol. Evol. 151–180.

Saito, N., Nei, M., 1987. The neighbor joining method: a new method for reconstructing phylogenetic tree. Mol. Biol. Evol. 4, 406–425.

Olsen, G.J., Matsuda, H., Hagstrom, R., Overbeek, R., 1994. FastDNAml: a tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. Comput. Appl. Biosci. 10, 41–48.

Pupko, T., Pe'er, I., Shamir, R., Graur, D., 2000. A fast algorithm for joint reconstruction of ancestral amino acid sequences. Mol. Biol. Evol. 17, 890–896.

Pupko, T., Sharan, R., Hasegawa, M., Shamir, R., Graur, D., 2001. In: Gascuel, O., Moret, B.M.E. (Eds.), A chemical-distance based test for positive Darwinian selection. WABI 2001, Lecture Notes in Computer Science, Vol. 2149, pp. 142–155.

Ralston, A., 1965. A First Course in Numerical Analysis. McGraw-Hill, New York.

Rogers, J.S., Swofford, D., 1998. A fast method for approximating maximum likelihoods of phylogenetic trees from nucleotide sequences. Syst. Biol. 47, 77–89.

Stewart, C.B., Wilson, A.C., 1987. Sequence convergence and functional adaptation of stomach lysozymes from foregut fermenters. Cold Spring Harbor Symp. Quant. Biol. 52, 891–899.

Strimmer, K., von Haeseler, A., 1996. Quartet puzzling: a quartet maximum likelihood method for reconstructing tree topologies. Mol. Biol. Evol. 13, 964–969.

Thompson, J.D., Higgins, D.G., Gibson, T.J., 1974. CLUSTALW: improving the sensitivity of progressive multiple sequences alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. 22, 4673–4680.

Yang, Z., 1994. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. J. Mol. Evol. 39, 306–314.

Yang, Z., 1997. PAML: a program package for phylogenetic analysis by maximum likelihood. CABIOS. 13, 555–556.

Zhang, J., Kumar, S., 1997. Detection of convergent and parallel evolution at the amino acid sequence level. Mol. Biol. Evol. 14, 527–536.