Dang et al. BMC Bioinformatics 2014, 15:341
http://www.biomedcentral.com/1471-2105/15/341

BMC Bioinformatics

RESEARCH ARTICLE Open Access

FastMG: a simple, fast, and accurate maximum likelihood procedure to estimate amino acid replacement rate matrices from large data sets

Cuong Cao Dang[1], Vinh Sy Le[1], Olivier Gascuel[2], Bart Hazes[3] and Quang Si Le[4*]

Abstract

Background: Amino acid replacement rate matrices are a crucial component of many protein analysis systems such as sequence similarity search, sequence alignment, and phylogenetic inference. Ideally, the rate matrix reflects the mutational behavior of the actual data under study; however, estimating amino acid replacement rate matrices requires large protein alignments and is computationally expensive and complex. As a compromise, sub-optimal pre-calculated generic matrices are typically used for protein-based phylogeny. Sequence availability has now grown to a point where problem-specific rate matrices can often be calculated if the computational cost can be controlled.

Results: The most time consuming step in estimating rate matrices by maximum likelihood is building maximum likelihood phylogenetic trees from protein alignments. We propose a new procedure, called FastMG, to overcome this obstacle. The key innovation is the alignment-splitting algorithm that splits alignments with many sequences into non-overlapping sub-alignments prior to estimating amino acid replacement rates. Experiments with different large data sets showed that the FastMG procedure was an order of magnitude faster than without splitting. Importantly, there was no apparent loss in matrix quality if an appropriate splitting procedure is used.

Conclusions: FastMG is a simple, fast and accurate procedure to estimate amino acid replacement rate matrices from large data sets. It enables researchers to study the evolutionary relationships for specific groups of proteins or taxa with optimized, data-specific amino acid replacement rate matrices. The programs, data sets, and the new mammalian mitochondrial protein rate matrix are available at http://fastmg.codeplex.com.

Keywords: Amino acid replacement rate matrices, Maximum likelihood methods, Phylogenetic trees, Protein alignments, Large data sets

Background

Amino acid replacement rate matrices represent the estimates of instantaneous substitution rates between amino acids. The rates simultaneously capture aspects of DNA-level mutation, the genetic code and protein-level selection strength, which varies based on similarity in chemical and physical properties. For example, we usually observe a high substitution rate between lysine and arginine (both positively charged and polar) and a low substitution rate between lysine and cysteine (neutral and nonpolar). Ideally, the replacement rate matrix parameters are optimized against the data under study, but in practice information content in typical sequence alignments is insufficient to do so. Instead, a small number of generic matrices are made available to researchers.

Amino acid replacement rate matrices are essential for many protein analyses, including estimating pairwise distances between protein sequences, or reconstructing protein phylogenetic trees using maximum likelihood or Bayesian frameworks [1,2] and references therein. Amino acid replacement rate matrices can also be converted into score matrices for protein sequence alignment. Roles and applications of amino acid replacement rate matrices were summarized by Thorne [3].

A number of methods have been proposed to estimate the matrices from protein alignments since the time of

* Correspondence: quang@well.ox.ac.uk
[4]The Wellcome Trust Center for Human Genetics, Oxford University, Oxford, UK
Full list of author information is available at the end of the article

Dayhoff [4]. These methods belong to either counting or maximum likelihood approaches. The counting methods are fast, but they are limited to only pairwise protein alignments and closely related amino acid sequences [4,5]. The maximum likelihood methods have been designed to fully utilize the information contained in multiple protein alignments and the corresponding phylogenetic trees which must be estimated from the data [6-8]. This assumes that the trees are correct, which can be ameliorated by a Bayesian analysis over a set of plausible trees but this would increase an already large computational burden.

With the rapid rise in whole genome sequencing it is now increasingly common to have access to both large numbers of taxa and long concatenated sequence alignments. This creates the opportunity to estimate data-specific amino acid replacement matrices but also requires efficient computational methods because estimating amino acid replacement rate matrices from protein alignments by maximum likelihood methods is a complex and time-consuming process [7,9,10] and references therein.

Recently, a fully automated maximum likelihood estimation procedure was proposed and used to estimate matrices from different data sets [8,10,11]. It consists of two main steps: building maximum likelihood phylogenetic trees and estimating parameters based on the information contained in multiple protein alignments and the corresponding phylogenetic trees. Building maximum likelihood trees is itself a difficult problem because the number of possible trees increases exponentially with the number of sequences in the alignment [1,12]. A number of maximum likelihood tree search heuristics have been developed to reduce the computational burden [13-16]; however, building maximum likelihood trees is still the most time consuming step in the estimation process. For example, in this study it took 98% (319 out of 324 hours) when estimating the amino acid replacement rate matrix from 100 large alignments in the HSSP (homology-derived structures of proteins) database [17].

In this paper, we propose a new maximum likelihood estimation procedure, FastMG, to work with large data sets. The key idea is to split large alignments into multiple non-overlapping sub-alignments with fewer sequences (each sub-alignment contains at most k sequences) in order to substantially reduce the computational burden of building maximum likelihood trees. The matrices are then estimated from the joint maximum likelihood analysis of the smaller sub-alignments instead of from the large original alignments.

A preliminary study showed that the splitting strategy significantly decreased the running time of the estimation procedure [9]. Here we demonstrate that a naïve random splitting method compromises the quality of the results. In contrast, our "tree-based splitting method" selects sub-trees that retain enough information to estimate accurate amino acid replacement rates. Experiments with different large data sets showed that the FastMG procedure yields similar quality matrices in much less time than the standard estimation procedure.

Results and discussion

We assessed the performance of the FastMG procedure on three large data sets: HSSP data set [17], Pfam data set [18], and our concatenated protein alignment of mitochondrial proteins from 299 mammalian species with 3062 amino acid sites. The FastMG procedure was examined with the random splitting algorithm, the tree-based splitting algorithm, and different k values targeting sub-alignment sizes of 8, 16, 24, 32, and 64 sequences. All matrices were estimated on a personal computer (Intel 2.66 GHz, 8 GB RAM). The PhyML software version 3.0 [14] was used to build maximum likelihood trees from alignments with options: 4 gamma categories model, no invariant sites, no bootstrap, SPR tree improvement, and JTT model. The XRATE software version 0.2 [19] was used to estimate model parameters using information in the alignments and corresponding phylogenetic trees. Let us denote:

- FastMG^{R}: The FastMG estimation procedure with the random splitting algorithm.
- FastMG^{T}: The FastMG estimation procedure with the tree-based splitting algorithm.
- M_k^{R} : Replacement rate matrix estimated from data set M using the FastMG^{R} procedure and threshold k (e.g., HSSP_8^{R} is the matrix estimated by the FastMG^{R} procedure with $k = 8$).
- M_k^{T} : Replacement rate matrix estimated from data set M using the FastMG^{T} procedure and threshold k (e.g., HSSP_8^{T} is the matrix estimated by the FastMG^{T} procedure with $k = 8$).
- M^{s}: Replacement rate matrix estimated from data set M using the standard maximum likelihood estimation procedure (e.g., HSSP^{S} is the matrix estimated by the HSSP data set using the standard estimation procedure).

We compared $\text{M}_{\text{k}}^{\text{R}}$, $\text{M}_{\text{k}}^{\text{T}}$, and M^{s} matrices in terms of both quality and running time. To avoid model bias due to over-fitting, each data set consisted of two alignment sets: the training alignment set and the testing alignment set. The matrices were estimated from alignments in the training set and subsequently used to build maximum likelihood trees for alignments in the testing set. Likelihood scores for test set alignments were used to compare the quality of different matrices as used in other studies [7,8,11]. Moreover, we used the Kishino-Hasegawa test [20] to assess the statistical significance of the difference between two matrices as used in previous studies [8,11].

HSSP data set

We selected 400 alignments from the HSSP database to assess the performance of estimation procedures. The 100 alignments with the largest number of sequences were used as the training alignment set for estimating matrices while the other 300 alignments were used as the testing alignment set. The 100 training alignments contained between 140 and 285 sequences (the mean and max pairwise distances between sequences were 0.481 and 1.286 respectively), for a total of 18325 sequences. The 300 testing alignments contained between 10 and 100 sequences (the mean and max pairwise distances between sequences were 0.635 and 1.846, respectively), for a total of 12854 sequences.

We examined the correlation between matrices estimated from the standard and the FastMG procedures. Table 1 shows high correlations between the 190 exchangeability coefficients of these matrices. As expected, the correlations increase with the increase of k. The results also show that tree-based splitting gives exchangeability parameters that are closer to the standard procedure. The opposite is true for the 20 frequency parameters of the matrices, likely because sub-alignments created by random splitting tend to represent the residue composition of the entire data set, whereas tree-based splitting gives sub-alignments that reflect the residue composition of individual clades. Another reason is likely because the random splitting algorithm selects taxa that are more distantly related and for which there was more time for the substitution process to reach equilibrium.

The more important impact on tree likelihood values in pairwise comparisons between $HSSP^S$ and other matrices estimated by $FastMG^R$ and $FastMG^T$ procedures are represented in Table 2. The matrices estimated from the $FastMG^R$ procedure were not as good as the $HSSP^S$ matrix for any k value. In contrast, the $FastMG^T$ procedure generated high quality matrices and $FastMG^T$ with $k \geq 16$ was at least as good as the standard estimation procedure.

Table 3 shows the running time of the standard estimation procedure and the FastMG procedure with different splitting algorithms and k values for the HSSP data set. The estimation time of the FastMG procedure increased linearly with the increase of k (e.g., it took 11.0 hours and 22.9 hours to estimate $HSSP_8^T$ and $HSSP_{16}^T$, respectively). This was an order of magnitude faster than the 323.7 hours needed to estimate $HSSP^S$. Interestingly, $FastMG^T$ was noticeably and consistently faster than $FastMG^R$. The difference is not as large but suggests faster convergence for tree-based splitting.

Table 1 The Pearson correlations between the $HSSP^S$ matrix and other matrices estimated by the FastMG procedure

Matrices	Frequencies	Exchangeability matrix
$HSSP_8^R/HSSP_8^T$	0.992/0.986	0.989/0.991
$HSSP_{16}^R/HSSP_{16}^T$	0.996/0.993	0.992/0.996
$HSSP_{24}^R/HSSP_{24}^T$	0.998/0.995	0.994/0.997
$HSSP_{32}^R/HSSP_{32}^T$	0.998/0.997	0.995/0.998
$HSSP_{64}^R/HSSP_{64}^T$	0.999/0.999	0.997/0.999

Pfam data set

We also examined different estimation procedures on alignments from the Pfam database. The training alignment set contained the 100 largest alignments from the Pfam database with in total 7640 sequences and a range of 46 to 202 sequences per alignment (the mean and max pairwise distances between sequences were 1.341 and 62.428, respectively). The testing alignment set consisted of 480 other alignments with in total 5434 sequences and a range of 5 to 41 sequences per alignment (the mean and max pairwise distances between sequences were 1.174 and 63.096, respectively). Note that the Pfam alignments tended to contain fewer sequences than the HSSP alignments.

We observed similar values and trends as for the HSSP data set, with very high correlations between the $Pfam^S$ matrix and matrices estimated from the FastMG procedure (see Table 4).

The pairwise comparisons of tree likelihood values between the $Pfam^S$ matrix and matrices estimated from the FastMG procedure are represented in Table 5. Again, we observed similar trends as for the HSSP data set. The quality of matrices increased with an increase of k, however, $FastMG^R$ matrices were never as good as the $Pfam^S$ matrix. In contrast, $FastMG^T$ produced matrices that were in a majority of cases slightly better than the standard estimation procedure for all k values. For example, the $Pfam_{16}^T$ matrix was better than the $Pfam^S$ matrix on 290 out of 480 testing alignments. Moreover, the Kishino-Hasegawa test showed that the $Pfam_{16}^T$ matrix was significantly better than the $Pfam^S$ matrix on 104 testing alignments. The opposite was true only 41 times.

The running time of the standard estimation procedure and the FastMG procedure with different splitting algorithms and k values for the Pfam data set are represented in Table 6. We observed similar running time patterns as for the HSSP data set, however, the magnitude of the speed gain was less in this case (e.g. ~4 times for $Pfam_{16}^T$ compared to $Pfam^S$) because of the typically lower number of sequences in the Pfam alignments.

Mammalian protein alignment

Our mammalian protein alignment (Mam) consists of the concatenated 12 mitochondrial proteins (all except ND6)

Table 2 Pairwise comparisons between $HSSP^S$ and other matrices estimated from the FastMG procedure

M1	M2	LogLK (M1-M2)	M1 > M2 (#TP)	M1 < M2 (#TP)	#M1 > M2 (#TP)	#M1 < M2 (#TP)
$HSSP^S$	$HSSP^R_8$	0.02	201 (130)	99 (65)	66 (27)	12 (4)
$HSSP^S$	$HSSP^R_{16}$	0.01	208 (126)	92 (64)	69 (26)	10 (4)
$HSSP^S$	$HSSP^R_{24}$	0.01	203 (114)	97 (72)	72 (26)	9 (5)
$HSSP^S$	$HSSP^R_{32}$	0.01	206 (119)	94 (59)	69 (24)	11 (4)
$HSSP^S$	$HSSP^R_{64}$	0.01	200 (101)	100 (63)	73 (16)	11 (4)
$HSSP^S$	$HSSP^T_8$	0.01	191 (124)	109 (75)	43 (19)	33 (15)
$HSSP^S$	$HSSP^T_{16}$	0.00	152 (95)	148 (81)	26 (5)	34 (9)
$HSSP^S$	$HSSP^T_{24}$	0.00	142 (88)	158 (89)	19 (4)	36 (11)
$HSSP^S$	$HSSP^T_{32}$	0.00	132 (78)	168 (87)	18 (3)	32 (6)
$HSSP^S$	$HSSP^T_{64}$	0.00	131 (72)	169 (80)	15 (4)	40 (4)

LogLK: the log likelihood difference per site between trees inferred using M1 and M2; a positive (negative) value means M1 is better (worse) than M2. M1 > M2: the number of alignments where M1 results in better likelihood value than M2; #TP: the number of alignments where tree topologies inferred using M1 and M2 are different. #M1 > M2 (p <0.05): the number of alignments where the Kishino-Hasegawa test indicates that M1 is significantly better than M2. #M1 < M2 (p <0.05): the same as #M1 > M2, but now M2 is significantly better than M1.

of 299 mammalian species and 3602 amino acid sites. The mean and max pairwise distances between sequences were 0.256 and 0.441, respectively. Because it is a single alignment, we used 10-fold cross validation to examine the performance of different estimation procedures [21]. In particular, the 3602 amino acid sites were randomly distributed across 10 non-overlapping partitions $P_1, \ldots, P_{10}$ each consisting of 360 or 361 sites. The validation was repeated 10 times. Let V_t ($t = 1 \ldots 10$) denote the t^{th} validation (i.e. the part P_t was used as the testing data and the other 9 parts were used as the training data).

Table 7 shows the likelihood comparisons between the Mam^S matrix and the matrices estimated from the FastMG procedure at the 10 validations. Again, the matrices estimated from the $FastMG^R$ procedure were never as good as the Mam^S matrix, while the matrices estimated by $FastMG^T$ with $k \geq 16$ were similar or slightly better than the Mam^S matrix. For example, $FastMG^T$ with $k = 16$ gave slightly better likelihood scores than the standard estimation procedure in all 10 validations and significantly better scores in 4 validations.

Table 3 The running time of the standard estimation procedure and the FastMG procedure with different splitting algorithms and *k* values for the HSSP data set

Matrices	Building trees (hours)	Estimating parameters (hours)	Total time (hours)
$HSSP^R_8/HSSP^T_8$	10.7/7.4	6.3/3.6	17.0/11.0
$HSSP^R_{16}/HSSP^T_{16}$	22.9/19.0	6.0/3.9	28.9/22.9
$HSSP^R_{24}/HSSP^T_{24}$	32.6/29.9	5.7/3.9	38.3/33.8
$HSSP^R_{32}/HSSP^T_{32}$	42.3/40.0	5.5/3.9	47.8/43.9
$HSSP^R_{64}/HSSP^T_{64}$	73.7/71.7	5.2/4.1	78.9/75.8
$HSSP^S$	319.5	4.2	323.7

We also examined the performance of the original MtMam matrix [22] and matrices estimated from the FastMG procedure. Table 8 shows that the matrices estimated by $FastMG^T$ were significantly better than the original MtMam for all 10 validations.

The estimation time of different matrices is presented in Table 9. For this alignment with 299 species the FastMG procedure was an order of magnitude faster than the standard estimation procedure (e.g. $FastMG^T$ with $k = 16$ was about 24 times faster than the standard estimation procedure). It also repeats the trend that the tree-splitting method is both faster and yields more accurate matrices.

Trends and special considerations

At the start of our studies we anticipated that alignment splitting would result in only minimal deterioration of matrix quality. Instead we observed, on balance, a small improvement in the performance of $FastMG^T$ matrices compared to those obtained without splitting. Although the effect is small it is consistent and suggests that a systematic effect is at play. Although further studies are needed to better understand these effects, it is plausible that the specific elimination of deeper and often less well defined branches contributes to the effect. In addition, tree-based splitting results in alignments of more closely related sequences with lower risk of sequence alignment errors.

There were a number of alignments where the matrices inferred from the standard and $FastMG^T$ procedures resulted in different tree topologies. This occurred in about 50-75% of cases for HSSP alignments and 10-25% of cases for the Pfam alignment (Tables 2 and 5, columns 4 and 5). This different rate of occurrences is expected because the HSSP alignments have more sequences and are therefore

Table 4 The Pearson correlations between the $Pfam^S$ matrix and other matrices estimated from the FastMG procedure

Matrices	Frequencies	Exchangeability matrix
$Pfam_8^R/Pfam_8^T$	0.994/0.990	0.993/0.993
$Pfam_{16}^R/Pfam_{16}^T$	0.997/0.996	0.995/0.997
$Pfam_{24}^R/Pfam_{24}^T$	0.998/0.999	0.995/0.999
$Pfam_{32}^R/Pfam_{32}^T$	0.999/0.999	0.998/0.999
$Pfam_{64}^R/Pfam_{64}^T$	1.000/1.000	0.999/1.000

more prone to topology differences. We considered the possibility that achieving a significantly better likelihood score was due to finding a different tree topology. However, in cases where the likelihood score is significantly better only a minority of alignments has a different tree topology.

Our three test cases all include a considerable amount of sequence divergence and all show that tree-splitting is superior to random splitting. A preliminary study on closely related influenza sequences showed that tree-splitting is still optimal [9]. However, in extreme cases the number of observed substitutions upon tree-splitting may become too small to be informative. In such a case random splitting may be preferred.

Another consideration is that our current FastMG procedure uses neighbor-joining to create the tree needed for the tree-based splitting algorithm. This scales well for alignments with hundreds or even thousands of sequences, but becomes inefficient for extremely large alignments. The performance of the FastMG procedure for such huge alignments will likely benefit from faster alignment splitting methods [23] and faster tree building methods (e.g. FastTree [24]) and this deserves further study if such cases become more common.

Conclusions

Amino acid replacement matrices are essential for many statistical methods to analyze protein sequences. Maximum likelihood methods typically generate the best replacement matrices because they can fully utilize information in the multiple protein alignments. However, for this application maximum likelihood analysis is complex and computationally expensive. Here we propose a modified maximum likelihood procedure to estimate amino acid replacement rate matrices from large data sets. The key component of the modified estimation procedure is the splitting algorithm that divides large alignments into multiple sub-alignments with fewer sequences that are subsequently used to estimate matrices. The extensive experiments showed that the $FastMG^T$ procedure performed well with large data sets and reduces the running time as function of the number of sequences from approximately quadratic to linear, as we expect based on the time complexity of tree inference (see Methods section). $FastMG^T$ with $k \geq 16$ was about an order of magnitude faster than the standard estimation procedure while it did not reduce the quality of estimated matrices.

Experiments strongly suggest that $k = 16$ is a good choice for the $FastMG^T$ estimation procedure. Even the analysis of the 100 largest alignments of the HSSP database took less than a day with $FastMG^T$ and $k = 16$ on a typical desktop computer. Thus, our method now enables researchers to avoid generic pre-calculated matrices and instead estimate optimal amino acid replacement matrices for their particular needs from large data sets on their personal computers.

Methods

Model

As usual, the amino acid substitution process is assumed to be independent among amino acid sites. Although the data

Table 5 Pairwise comparisons between the $Pfam^S$ matrix and other matrices estimated by the FastMG procedure

M1	M2	LogLK (M1-M2)	M1 > M2 (#TP)	M1 < M2 (#TP)	#M1 > M2 (#TP)	#M1 < M2 (#TP)
$Pfam^S$	$Pfam_8^R$	0.01	299 (67)	181 (41)	119 (8)	55 (8)
$Pfam^S$	$Pfam_{16}^R$	0.01	294 (54)	186 (35)	132 (6)	55 (3)
$Pfam^S$	$Pfam_{24}^R$	0.01	309 (57)	171 (38)	142 (10)	51 (3)
$Pfam^S$	$Pfam_{32}^R$	0.00	275 (40)	205 (35)	116 (6)	65 (2)
$Pfam^S$	$Pfam_{64}^R$	0.00	279 (38)	201 (28)	117 (4)	64 (3)
$Pfam^S$	$Pfam_8^T$	0.00	218 (54)	262 (64)	51 (3)	80 (8)
$Pfam^S$	$Pfam_{16}^T$	0.00	190 (33)	290 (51)	41 (2)	104 (6)
$Pfam^S$	$Pfam_{24}^T$	0.00	212 (33)	268 (39)	50 (2)	82 (1)
$Pfam^S$	$Pfam_{32}^T$	0.00	233 (36)	247 (36)	58 (1)	72 (0)
$Pfam^S$	$Pfam_{64}^T$	0.00	166 (21)	314 (31)	27 (0)	91 (2)

LogLK: the log likelihood difference per site between trees inferred using M1 and M2; a positive (negative) value means M1 is better (worse) than M2. M1 > M2: the number of alignments where M1 results in better likelihood value than M2; #TP: the number of alignments where tree topologies inferred using M1 and M2 are different. #M1 > M2 (p <0.05): the number of alignments where the Kishino-Hasegawa test indicates that M1 is significantly better than M2. #M2 > M1 (p <0.05): the same as #M1 > M2, but now M2 is significantly better than M1.

Table 6 The running time of the standard estimation procedure and the FastMG procedure with different splitting algorithms and *k* values for the Pfam data set

Matrices	Building trees (hours)	Estimating parameters (hours)	Total time (hours)
$Pfam_8^R/Pfam_8^T$	2.8/2.4	5.5/3.1	8.3/5.5
$Pfam_{16}^R/Pfam_{16}^T$	6.6/6.2	4.7/3.0	11.3/9.3
$Pfam_{24}^R/Pfam_{24}^T$	9.9/9.4	4.2/2.9	14.1/12.3
$Pfam_{32}^R/Pfam_{32}^T$	12.4/12.6	4.0/2.9	16.4/15.5
$Pfam_{64}^R/Pfam_{64}^T$	20.8/22.4	3.4/2.8	24.2/25.2
$Pfam^S$	35.8	2.9	38.7

might not be compositionally homogeneous across the sequences in the alignment (e.g., different amino acid compositions among the clades), we assume that the standard model for the amino acid substitution process over the tree is a Markov process with time-homogeneous, time-continuous, and time-reversible properties [6-8] and references therein. The model is represented by a 20×20 instantaneous substitution rate matrix $Q = \{q_{xy}\}$ where $q_{xy} (x \neq y)$ represents the number of substitutions from amino acid x to amino acid y per time unit. The diagonal elements q_{xx} are assigned such that the row sums are all zero. Since the model is time reversible, the matrix Q can be decomposed into a symmetric exchangeability rate matrix $R = \{r_{xy}\}$ and an amino acid equilibrium frequency vector $\Pi = \{\pi_x\}$ such that $q_{xy} = r_{xy}\pi_y$ and $q_{xx} = -\sum_{x \neq y} q_{xy}$.

Model estimation procedure

Given a set of c protein alignments $D = \{D_1, \ldots, D_c\}$, the substitution model Q can be estimated by maximizing the likelihood $L(D)$ using equation 1 as follows

$$L(\boldsymbol{D}) = \prod_{i=1}^{c} L\left(T_i, \rho_i, Q; D_i\right) \quad (1)$$

where $L(T_i, \rho_i, Q; D_i)$ is the likelihood of protein alignment D_i given phylogenetic tree T_i; the rate variation model ρ_i; and substitution model Q [7,8].

Optimizing $L(D)$ is a difficult problem because we have to simultaneously optimize parameters of T, Q, and ρ. Previous studies indicated that a good approximation of Q *(Q')* can be obtained with near-optimal trees *(T')* and rate variation model (ρ') [7,8,10] and references therein. Because trees and rate variation models are computed a priori, the likelihood $L(D)$ can thus be approximated by equation 2 as follows

$$L(\boldsymbol{D}) = \prod_{i=1}^{c} L\left(Q'; T_i', \rho_i', D_i\right) \quad (2)$$

A better model Q can be estimated from alignments of D using an iterative approach as detailed in the 4-step standard estimation procedure as follows [8]:

Standard estimation procedure

- Step 0: Input a set of multiple alignments D and a starting matrix Q (typically input only exchangeability rate matrix R, the frequency vector Π is estimated from the data D).
- Step 1: Build phylogenetic tree T_i and rate across site model ρ_i for each alignment D_i using maximum likelihood tree construction programs such as PhyML [14].
- Step 2: Estimate a new exchangeability matrix Q' using the approach described by Le and Gascuel [8] and the XRate software [19].
- Step 3: Compare Q' and Q, if they are nearly identical, return Q' as the optimal model. Otherwise, assign $Q \leftarrow Q$' and goto Step 1.

Previous studies have showed that this estimation procedure usually stops after three iterations. The

Table 7 The log likelihood per site comparisons between the Mam^S matrix and matrices estimated by the FastMG procedure

	Mam_8^R/Mam_8^T	Mam_{16}^R/Mam_{16}^T	Mam_{24}^R/Mam_{24}^T	Mam_{32}^R/Mam_{32}^T	Mam_{64}^R/Mam_{64}^T
LogLK per site	0.72/-0.04	0.58/-0.06	0.49/-0.05	0.42/-0.05	0.26/-0.02
# Significantly better	10/0	10/0	10/0	10/0	10/0
# Significantly worse	0/1	0/4	0/4	0/5	0/3

LogLK per site: average log-likelihood per site difference between Mam^S and the other matrices, positive/negative values indicate that the Mam^S matrix was better/worse than the other matrices. #Significantly better: number of tests where Mam^S is significantly better than the Mam^R/Mam^T matrix (based on Kishino-Hasegawa test). #Significantly worse: number of tests where Mam^S is significantly worse than the Mam^R/Mam^T matrix (based on Kishino-Hasegawa test).

procedure is called "standard maximum likelihood estimation procedure".

Building maximum likelihood trees in Step 1 is a computationally expensive problem [12]. Although a number of heuristics have been proposed for searching maximum likelihood trees [13-16], Step 1 is still the bottleneck of the estimation process. It is roughly estimated [25] that the computing time of PhyML (and of other similar maximum likelihood approaches) is in $O(n^2 s)$, where n is the number of taxa and s the number of sites. Experiments with a number of different data sets have confirmed this approximation. Thus, it is expected that splitting the original alignments into two equally-sized sub-alignments divides the total computing time by a factor two. This property explains why the computing time to build trees displayed in Tables 3, 6 and 9 is approximately linear as a function of k (size of sub-alignments).

Alignment-splitting algorithms

Consider a multiple alignment D_i of n sequences $(d_i^1, ..., d_i^n)$, splitting alignment D_i is a process to divide alignment D_i into non-overlapping smaller sub-alignments D_i^1, ..., D_i^m such that each sequence $d_i^{j=1...n} \in D_i$ belongs to exactly one sub-alignment $D_i^t (t = 1...m)$. For example, alignment D_i of 8 sequences $(d_i^1, ..., d_i^8)$ can be split into three sub-alignments $D_i^1 = (d_i^1, d_i^2, d_i^8)$, $D_i^2 = (d_i^3, d_i^4, d_i^5)$, $D_i^3 = (d_i^6, d_i^7)$. Let k be the maximum number of sequences in a sub-alignment; we seek a method of alignment splitting that allows us to minimize k, to maximize computational efficiency, while retaining adequate amounts of information in sub-alignments for estimating the amino acid replacement rates.

Random splitting algorithm

Given a multiple alignment D_i of n sequences $(d_i^1, ..., d_i^n)$ and a threshold k, we describe a naïve splitting algorithm, called "Random splitting algorithm". The general idea of the algorithm is to randomly split sequences $(d_i^1, ..., d_i^n)$ into sub-alignments such that each sub-alignment contains at most k sequences. To prevent creating too small sub-alignments that might not contain enough information for estimating the replacement rates, the algorithm also requires that each sub-alignment needs to contain at least $k/2$ sequences. Thus, each sub-alignment contains from $k/2$ to k sequences. The random splitting algorithm is fully described in Algorithm 1.

Algorithm 1 Random splitting algorithm

Input: A multiple alignment D_i of n sequences $\left(d_i^1, \ldots, d_i^n\right)$ and an integer number k.

Output: A set of sub-alignments of D_i each containing from $k/2$ to k sequences.

Begin

$m \leftarrow 0$; //the number of sub-alignments

$n \leftarrow | D_i |$; //the number of sequences in D_i

while $(n > k)$

Generate a random number s from $k/2$ to $\min(k, n)$;

$m \leftarrow m + 1$;

Select randomly s sequences from D_i to create a new sub-alignment D_i^m;

Remove selected sequences out of D_i;

$n \leftarrow n - s$;

endwhile;

if $(n \geq 3)$

$m \leftarrow m + 1$;

$D_i^m \leftarrow D_i$; *//Note that* D_i^m *has fewer than* $k/2$ *sequences.*

else

Add remaining sequences in D_i to sub-alignment D_i^m;

//Note that D_i^m *might contain more than* k *sequences.*

endif;

Return sub-alignments $D_i^1 \ldots D_i^m$

End;

The random splitting algorithm is very simple; however, its main drawback is that sequences of the same sub-alignment might be very distantly related. This could compromise estimation of amino acid replacement rates.

Tree-based splitting algorithm

We designed a new splitting algorithm, called tree-based splitting algorithm, to maximize homology between sequences in each sub-alignment. Let T_i denote the phylogenetic tree relating sequences $(d_i^1, ..., d_i^n)$ of D_i. The key idea of the tree-based splitting algorithm is that sequences in the same sub-tree of T_i should be split into the same sub-alignment. Figure 1 shows an example of 9 sequences related by a tree. The sequences can be split into 2 sub-alignments (s_1, s_2, s_3, s_8) and $(s_4, s_5, s_6, s_7, s_9)$ where (s_1, s_2, s_3, s_8) sequences belong to the left sub-tree while $(s_4, s_5, s_6, s_7, s_9)$ sequences belong to the right sub-tree.

The tree-based splitting algorithm follows the Neighbor-joining algorithm schema [26] to step-by-step group sequences into sub-alignments. The algorithm also requires that each sub-alignment contains at least $k/2$ sequences and at most k sequences. The tree-based splitting algorithm is fully described in Algorithm 2.

Algorithm 2 Tree-based splitting algorithm

Input: A multiple alignment D_i of n sequences $\left(d_i^1,\ldots,d_i^n\right)$ and a number k.

Output: A set of sub-alignments of D_i each containing from $k/2$ to k sequences.

Begin

$m \leftarrow 0$; //the number of sub-alignments

Consider n sequences of D_i as n different nodes (node G_j contains sequence d_i^j, $j = 1 \ldots n$);

repeat

Find two neighbor nodes G_i, G_j following the Neighbor-joining schema [25] ;

Let n_i and n_j be the number of sequences in nodes G_i and G_j, respectively (assume that $n_i \geq n_j$);

if $(n_i + n_j \leq k)$

Join neighbor nodes G_i, G_j into a new node G_{ij} (node G_{ij} contains all sequences of G_i, G_j); Replace nodes G_i, G_j by new node G_{ij};

else // $n_i + n_j > k$ and $n_i > k/2$

$m \leftarrow m+1$;

Create sub-alignment D_i^m from all sequences of G_i;

Remove node G_i;

endif

until (only one node remains);

Let G_0 be the remaining node and n_0 be the number of sequences in G_0;

if $(n_0 \geq 3)$

$m \leftarrow m+1$;

Create sub-alignment D_i^m from all sequences of G_0;

//Note that D_i^m *contains less than* $k/2$ *sequences.*

else

Add sequences of G_0 to sub-alignment D_i^m;

// Note that D_i^m *might contain more than* k *sequences.*

endif;

Return sub-alignments $D_i^1 \ldots . D_i^m$;

End;

Table 8 The log likelihood per site comparisons between the original MtMam matrix and matrices estimated by the FastMG procedure

	Mam_8^R/Mam_8^T	Mam_{16}^R/Mam_{16}^T	Mam_{24}^R/Mam_{24}^T	Mam_{32}^R/Mam_{32}^T	Mam_{64}^R/Mam_{64}^T
LogLK per site	0.37/-0.39	0.23/-0.40	0.14/-0.40	0.07/-0.40	−0.09/-0.37
# Significantly better	0/0	0/0	0/0	0/0	0/0
# Significantly worse	0/10	0/10	1/10	4/10	7/10

LogLK per site: average log-likelihood per site difference between the original MtMam and the other matrix, positive/negative values indicate that the original MtMam matrix was better/worse than the other matrix. #Significantly better: number of tests where the original MtMam is significantly better than the Mam^R/Mam^T matrix (based on Kishino-Hasegawa test). #Significantly worse: number of tests where the original MtMam is significantly worse than the Mam^R/Mam^T matrix (based on Kishino-Hasegawa test).

Table 9 The running time (hours) of different estimation procedures

	Mam^S	Mam_8^R/Mam_8^T	Mam_{16}^R/Mam_{16}^T	Mam_{24}^R/Mam_{24}^T	Mam_{32}^R/Mam_{32}^T	Mam_{64}^R/Mam_{64}^T
Avg. time	22.2	0.5/0.4	1.5/0.9	2.2/1.4	2.7/1.9	4.3/3.6
Speed up		42/61.4	14.8/24.5	10.2/16	8.2/11.6	5.1/6.1

Note that the distance matrix between sequences used in the Neighbor-joining algorithm is estimated by the maximum likelihood method using the LG matrix [8]. Technically, we used BIONJ [27] (an improved version of the Neighbor-joining algorithm) to split large alignments.

Fast maximum likelihood estimation procedure (FastMG)

The FastMG procedure consists of two phases: first, the large original alignments are split into non-overlapping sub-alignments by one of the alignment splitting algorithms; then the matrix is estimated by joint maximum likelihood analysis of the smaller sub-alignments instead of the large original alignments. The FastMG procedure is described by the following 5-steps

Fast maximum likelihood estimation procedure (FastMG)

- Step 0: Input a set of multiple alignments D; a starting matrix Q (typically input only exchangeability rate matrix R, the frequency vector Π is estimated from the data D); and a threshold k.
- Step 1: For each alignment $D_i \in D$, split D_i into sub-alignment $D_i^1 \ldots D_i^m$ using either the random splitting algorithm or the tree-based splitting algorithm.

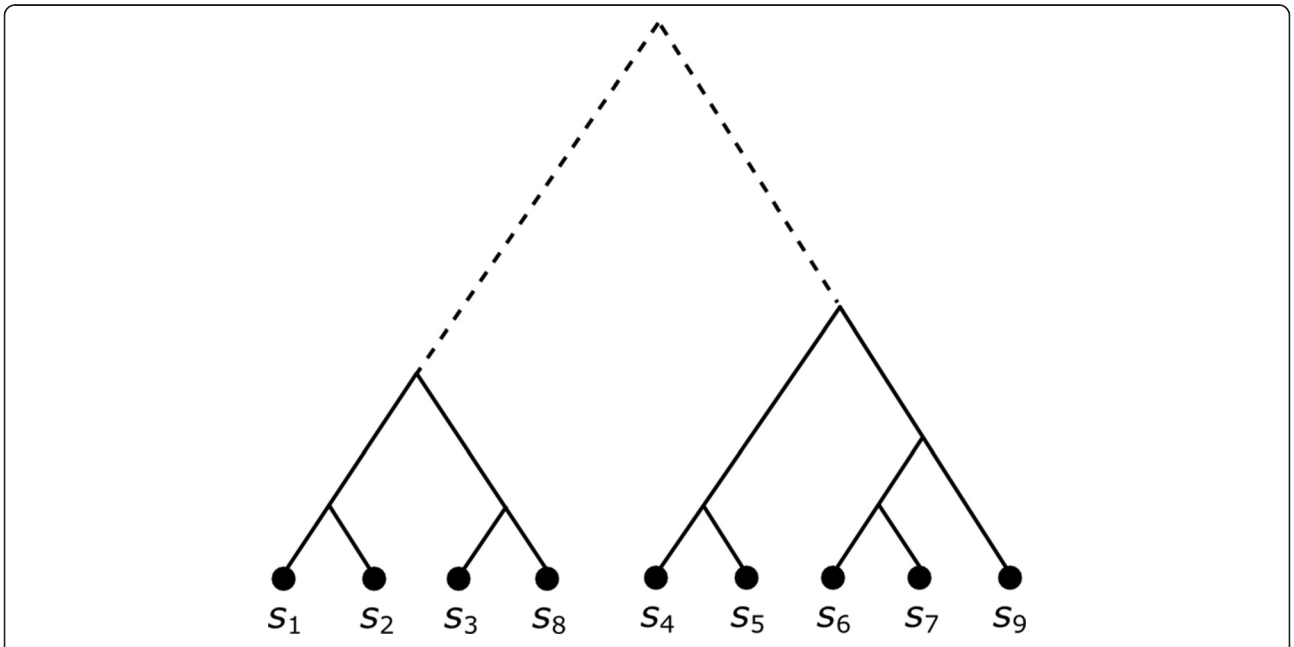

Figure 1 Tree-based splitting example. The tree-based splitting algorithm would divide this hypothetical tree for a 9-sequence alignment into two sub-alignments (s_1, s_2, s_3, s_8) and (s_4, s_5, s_6, s_7, s_9), corresponding to the left and right sub-trees, respectively.

- Step 2: Build phylogenetic tree $T_i^j$and rate across site model ρ_i^j for each sub-alignment D_i^j using maximum likelihood tree construction programs such as PhyML [14].
- Step 3: Estimate a new matrix Q' from sub-alignments using the approach described by Le and Gascuel [8] and the XRate software [19].
- Step 4: Compare Q' and Q, if they are nearly identical, return Q' as the optimal model. Otherwise, assign $Q \leftarrow Q$' and go to Step 2.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
LSV, LSQ and OG discussed ideas. DCC and LSV designed algorithms and experiments. BH created the concatenated mammalian protein alignment. DCC implemented the algorithms, conducted experiments, and wrote the draft manuscript. All authors analyzed experiment results. BH and OG revised the manuscript. All authors read and approved the final manuscript.

Acknowledgements
This work is financially supported by Vietnam National Foundation for Science and Technology (102.01-2013.04). BH and OG were supported by LABEX NUMEV to work on this study.

Author details
[1]VNU University of Engineering and Technology, Hanoi, Vietnam. [2]Institut de Biologie Computationnelle, LIRMM, CNRS – Université Montpellier 2, Montpellier, France. [3]Department of Medical Microbiology & Immunology, University of Alberta, Alberta, Canada. [4]The Wellcome Trust Center for Human Genetics, Oxford University, Oxford, UK.

Received: 13 May 2014 Accepted: 29 September 2014
Published: 24 October 2014

References
1. Felsenstein J: *Inferring Phylogenies.* Sunderland, MA, USA: Sinauer Associates; 2004.
2. Yang Z: *Computational Molecular Evolution.* Oxford, UK: Oxford University Press; 2006.
3. Thorne JL: **Models of protein sequence evolution and their applications.** *Curr Opin Genet Dev* 2000, **10:**602–605.
4. Dayhoff M, Schwartz R, Orcutt B: **A model of evolutionary change in proteins.** *Atlas Protein Seq Struct* 1978, **5:**345–351.
5. Jones DT, Taylor WR, Thornton JM: **The rapid generation of mutation data matrices from protein sequences.** *Comput Appl Biosci CABIOS* 1992, **8:**275–282.
6. Adachi J, Hasegawa M: **Model of amino acid substitution in proteins encoded by mitochondrial DNA.** *J Mol Evol* 1996, **42:**459–468.
7. Whelan S, Goldman N: **A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach.** *Mol Biol Evol* 2001, **18:**691–699.
8. Le QS, Gascuel O: **An improved general amino acid replacement matrix.** *Mol Biol Evol* 2008, **25:**1307–1320.
9. Le DV, Dang CC, Le QS, Le VS: **A Fast and Efficient Method for Estimating Amino Acid Substitution Models.** In *Proceedings of The Third International Conference on Knowledge and Systems Engineering.* Edited by Ho TB, McKay RI, Nguyen XH, Bui TD. New York, NY, USA: IEEE Publishing; 2011:85–91.
10. Dang CC, Lefort V, Le VS, Le QS, Gascuel O: **ReplacementMatrix: a web server for maximum-likelihood estimation of amino acid replacement rate matrices.** *Bioinformatics* 2011, **27:**2758–2760.
11. Dang CC, Le QS, Gascuel O, Le VS: **FLU, an amino acid substitution model for influenza proteins.** *BMC Evol Biol* 2010, **10:**99–110.
12. Chor B, Tuller T: **Maximum likelihood of evolutionary trees: hardness and approximation.** *Bioinformatics* 2005, **21:**97–106.
13. Guindon S, Gascuel O: **A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood.** *Syst Biol* 2003, **52:**696–704.
14. Guindon S, Dufayard J-F, Lefort V, Anisimova M, Hordijk W, Gascuel O: **New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0.** *Syst Biol* 2010, **59:**307–321.
15. Le VS, von Haeseler A: **IQPNNI: moving fast through tree space and stopping in time.** *Mol Biol Evol* 2004, **21:**1565–1571.
16. Stamatakis A, Ludwig T, Meier H: **RAxML-III: a fast program for maximum likelihood-based inference of large phylogenetic trees.** *Bioinformatics* 2005, **21:**456–463.
17. Schneider R, de Daruvar A, Sander C: **The HSSP database of protein structure-sequence alignments.** *Nucleic Acids Res* 1997, **25:**226–230.
18. Bateman A, Birney E, Cerruti L, Durbin R, Etwiller L, Eddy SR, Griffiths-Jones S, Howe KL, Marshall M, Sonnhammer ELL: **The Pfam protein families database.** *Nucleic Acids Res* 2002, **30:**276–280.
19. Klosterman PS, Uzilov AV, Bendaña YR, Bradley RK, Chao S, Kosiol C, Goldman N, Holmes I: **XRate: a fast prototyping, training and annotation tool for phylo-grammars.** *BMC Bioinformatics* 2006, **7:**428–453.
20. Kishino H, Hasegawa M: **Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in hominoidea.** *J Mol Evol* 1989, **29:**170–179.
21. Kohavi R: **A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection.** In *Proceedings of the 14th International Joint Conferences on Artificial Intelligence.* Montreal: Morgan Kaufmann Publishers Inc; 1995:1137–1143.
22. Yang Z, Nielsen R, Hasegawa M: **Models of amino acid substitution and applications to mitochondrial protein evolution.** *Mol Biol Evol* 1998, **15:**1600–1611.
23. Blackshields G, Larkin M, Wallace IM, Wilm A, Higgins DG: **Fast embedding methods for clustering tens of thousands of sequences.** *Comput Biol Chem* 2008, **32**(4):282–286.
24. Price MN, Dehal PS, Arkin AP: **FastTree 2 – approximately maximum-likelihood trees for large alignments.** *PLoS ONE* 2010, **5:**e9490.
25. Dereeper A, Guignon V, Blanc G, Audic S, Buffet S, Chevenet F, Dufayard J-F, Guindon S, Lefort V, Lescot M, Claverie J-M, Gascuel O: **Phylogeny.fr: robust phylogenetic analysis for the non-specialist.** *Nucleic Acids Res* 2008, **36**(suppl 2):W465–W469.
26. Saitou N, Nei M: **The neighbor-joining method: a new method for reconstructing phylogenetic trees.** *Mol Biol Evol* 1987, **4:**406–425.
27. Gascuel O: **BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data.** *Mol Biol Evol* 1997, **14:**685–695.

doi:10.1186/1471-2105-15-341
Cite this article as: Dang *et al.*: FastMG: a simple, fast, and accurate maximum likelihood procedure to estimate amino acid replacement rate matrices from large data sets. *BMC Bioinformatics* 2014 **15**:341.