JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 10, Number 6, 2003

Pp. 997–1010

A Transition Probability Model for Amino Acid Substitutions from Blocks

SHALINI VEERASSAMY, ANDREW SMITH, and ELISABETH R.M. TILLIER

ABSTRACT

Substitution matrices have been useful for sequence alignment and protein sequence comparisons. The BLOSUM series of matrices, which had been derived from a database of alignments of protein blocks, improved the accuracy of alignments previously obtained from the PAM-type matrices estimated from only closely related sequences. Although BLOSUM matrices are scoring matrices now widely used for protein sequence alignments, they do not describe an evolutionary model. BLOSUM matrices do not permit the estimation of the actual number of amino acid substitutions between sequences by correcting for multiple hits. The method presented here uses the Blocks database of protein alignments, along with the additivity of evolutionary distances, to approximate the amino acid substitution probabilities as a function of actual evolutionary distance. The PMB (Probability Matrix from Blocks) defines a new evolutionary model for protein evolution that can be used for evolutionary analyses of protein sequences. Our model is directly derived from, and thus compatible with, the BLOSUM matrices. The model has the additional advantage of being easily implemented.

Key words: amino acid substitution, empirical probability model, Blocks database.

1. INTRODUCTION

A MODEL DESCRIBING THE PROBABILITY OF SUBSTITUTION from one amino acid to another is useful for sequence alignment, phylogenetic analysis, the inference of ancestral protein sequences, and computer simulation of protein evolution (see Thorne [2001] for review). Models can be obtained from the ad hoc definition of parameters such as in the codon substitution model, for which probabilities of substitution for the amino acids depend on the number and type of base substitution that are needed at the DNA coding level. Parameters of the model can also be estimated using a rigorous maximum likelihood approach, but this approach is computationally intensive, makes several restrictive assumptions about the parameters, and assumes a known tree topology (Adachi and Hasegawa, 1996; Yang *et al.*, 1998; Adachi *et al.*, 2000; Whelan and Goldman, 2001). The parameters can also be obtained in an empirical manner from alignments of closely related proteins. The Dayhoff PAM matrices (Dayhoff *et al.*, 1978) and related Gonnet (Gonnet *et al.*, 1992) and JTT (Jones *et al.*, 1992) matrices define estimates of transition probabilities from the frequencies of substitution observed in actual proteins. For very closely related proteins, multiple

Ontario Cancer Institute, University Health Network, Toronto, Ontario, Canada.

substitutions at a site are unlikely, so the observed frequencies of substitution accurately reflect the actual substitution probabilities. Substitution models derived from only closely related proteins are not likely to be accurate at higher evolutionary distances. Recently, approaches were developed to estimate an amino acid substitution model from alignments of varying degrees of divergence. Arvestad and Bruno (1997) developed a method applicable to smaller datasets. Muller and Vingron (2000) developed a model using properties of the resolvents of the corresponding transition matrices and the SYSTERS database of aligned protein sequence families (Krause *et al.*, 2000). Devauchelle *et al.* (2002) used principal component analysis to estimate a model from transition matrices.

The BLOSUM series of matrices was derived to increase the accuracy of scoring matrices for larger sequence divergence. In the approach of Henikoff and Henikoff (1992), frequencies of substitution were obtained from a database of Blocks. A series of scoring matrices, applicable to sequences with increasing divergence, was then derived by clustering sequences above a given sequence identity so that their contribution is down-weighed. The BLOSUM matrices give a score for each type of amino acid substitution and were shown to produce superior alignments (as assessed by their performance in database searches) when compared to scoring matrices obtained from PAM matrices (Henikoff and Henikoff, 1992). The advantage of the Blocks database is that the alignments are very reliable; only the parts of protein sequences that have been aligned without any gaps are included. Because there are no gaps in the alignments, the database is ideal to model the process of amino acid substitution without regard for the processes of insertion and deletion. However, the BLOSUM matrices derived from Blocks are scoring matrices and do not define a probability model of substitution.

Here we have used a very simple approach, based on the original clustering approach of Henikoff and Henikoff (1992), to generate BLOSUM matrices from an updated version of the Blocks database. From these BLOSUM matrices, we derived mutability matrices. The mathematical property that evolutionary distances are additive was used to estimate the relationship between the actual substitution frequency and the average observed substitution frequency. The mutability matrices could then be expressed as a function of actual evolutionary distance thus defining an evolutionary model for protein evolution that is consistent with the BLOSUM scoring matrices and that is applicable over the complete range of evolutionary divergence.

2. METHODS

2.1. Blocks and BLOSUM

We obtained the Blocks databases (Henikoff *et al.*, 1999) from the NCBI ftp site.[1] The databases we analyzed excluded blocks that are biased in their composition (usually because of repeats in the sequences when available (the minus.dat files). The most recent Blocks database was Blocks+v13Aug01 (Henikoff *et al.*, 1999). Also from the NCBI ftp site,[2] we obtained the program BLOSUM (Henikoff and Henikoff, 1992) to find the observed frequency of each type of amino acid substitution (the frequency matrices). The program was run for clustering percentages ranging from 30% to 100% in increments of 2%, and for no clustering, yielding a series of frequency matrices, one for each clustering percentage.

2.2. Mutability matrices

Here we define mutability matrices and present some important properties. For the present subsection, subscripts are used to indicate matrix and vector elements. Let F be any substitution frequency count matrix obtained from BLOSUM. Since F is symmetric (i.e., the order of replacement of amino acids is not known) and forward and backward substitutions are combined in a single count, we assume throughout that the diagonal values of F have been doubled. The vector π of observed frequencies for each amino

[1] ftp://ncbi.nlm.nih.gov/repository/blocks/

[2] ftp://ncbi.nlm.nih.gov/repository/blocks/unix/BLOSUM/

acid is obtained by summing each row of F and dividing by the sum of all entries in the matrix:

$$\pi_i = \frac{\sum_{j=1}^{20} F_{ij}}{\sum_{i'=1}^{20}\sum_{j=1}^{20} F_{i'j}}. \tag{1}$$

Each row of the frequency matrix is then divided by the sum of the corresponding row resulting in mutability matrix M:

$$M_{ij} = \frac{F_{ij}}{\sum_{j=1}^{20} F_{ij}}. \tag{2}$$

Mutability matrices describe the frequency of any amino acid being substituted by any other (including itself). Because each element of a mutability matrix falls between 0 and 1 and the sum of the elements in each row is equal to 1, mutability matrices are stochastic matrices. The mutability matrices derived from the frequency matrices are reversible and thus fulfill the detailed balance equation:

$$\pi_i M_{ij} = \pi_j M_{ji}. \tag{3}$$

The average substitution frequency can be calculated from the mutability matrices and the frequency of the amino acids using the formula

$$\mathcal{D}(M) = 1 - \sum_{i=1}^{20} \pi_i M_{ii}. \tag{4}$$

The mutability matrices describe the observed frequencies of substitution (observed distance) expected after an unknown actual evolutionary distance P. Therefore, in addition to the definition given above, a mutability matrix can also be considered a function of P. To reflect the dependence of a mutability matrix upon the (unknown) evolutionary distance of the sequences from which M was derived, we let $M(P)$ denote the mutability matrix M as a function of some evolutionary distance P. The value of P is unknown since sequences ancestral to those in the alignment are unknown and because multiple substitutions may have actually occurred at any site. Since actual evolutionary distance is an additive metric, taking the square (square root) of a mutability matrix will double (halve) the actual evolutionary distance. In general, for any mutability matrix M and any number n,

$$M(P)^n = M(nP). \tag{5}$$

Equation 5 is a special form of the Chapman–Kolmogorov equation for Markov chains. It was this property that allowed the derivation of the PAM series of matrices from 1 PAM (Dayhoff *et al.*, 1978).

As stated in Section 2.1, our method makes use of data obtained from Blocks+ and BLOSUM at a series of clustering percentages. Unless otherwise stated, for the rest of this paper we use subscripts to indicate a clustering percentage; e.g., the matrix M_c is the mutability matrix corresponding to some clustering percentage $c \in \{0, 30, 32, 34, \ldots, 96, 98, 100\}$.

2.3. Derivation of the formula for the actual evolutionary distance

Without knowledge of ancestral sequences, the actual number of substitutions is unknown and will always be larger that the observed substitution frequency. To determine the relationship between observed evolutionary distance and actual evolutionary distance, we can use the additivity of mutability matrices

(see Equation 5). This property allows us to consider the behavior of the observed distance as we linearly increase the actual distance (by taking powers of the mutability matrices) without knowing the value of the actual distance. Our approach is to consider for each clustering the derivative of the observed distance with respect to the actual distance. The derivatives can be estimated numerically by considering small fractional changes of the actual distance P. Because of the additivity property (Equation 5), making small fractional changes to P corresponds to taking powers of the mutability matrices close to 1.

To estimate the derivatives, we used the five-point formula for numerical differentiation (Burden and Faires, 1985):

$$\frac{df(x_0)}{dx} = \frac{1}{12h}\left(f(x_0 - 2h) - 8f(x_0 - h) + 8f(x_0 + h) - f(x_0 + 2h)\right). \tag{6}$$

The derivative of the function f is thus estimated by considering values of f at several points separated by a small interval h. For our purpose, we substituted $x_0 = P$, $f(x_0) = \mathcal{D}(M(P))$, and $h = 0.01P$ into Equation 6. The resulting formula,

$$\begin{aligned}\frac{d\mathcal{D}(M(P))}{dP} = \frac{1}{12(0.01P)}[&\mathcal{D}(M(P - 0.02P)) - 8\mathcal{D}(M(P - 0.01P)) \\ &+ 8\mathcal{D}(M(P + 0.01P)) - \mathcal{D}(M(P + 0.02P))],\end{aligned} \tag{7}$$

describes the derivative of the observed distance with respect to the actual distance. Defining the small interval h as a constant function of P allows us to use the additivity property defined in Equation 5. We now have

$$0.12P\frac{d\mathcal{D}(M(P))}{dP} = \mathcal{D}(M(P)^{0.98}) - 8\mathcal{D}(M(P)^{0.99}) + 8\mathcal{D}(M(P)^{1.01}) - \mathcal{D}(M(P)^{1.02}). \tag{8}$$

We can find the appropriate powers (i.e., 0.98, 0.99, 1.01, and 1.02) of the mutability matrices using MATLAB 6 (release 12) (Mathworks, Natick, Massachusetts). The expected observed distance for M_c and its powers are calculated using Equation 4. The right-hand side of Equation 7 can then be calculated and plotted against the $\mathcal{D}(M(P))$ for each clustering percentage (Fig. 1). For notational convenience, we define

$$D = \mathcal{D}(M(P)) \tag{9}$$

as the expected observed distance for an actual average distance P. The expression on the right side of Equation 8 can be estimated by a polynomial function f in D which will yield a differential equation:

$$\frac{dD}{dP} = f(D) = a_n D^n + a_{n-1} D^{n-1} + \ldots + a_0. \tag{10}$$

We also know initial conditions since at sufficiently low distance there are no overlapping substitutions and the number of actual substitutions is equal to the number of observed substitutions. Thus,

$$\lim_{P \to 0} D = 0 \text{ and } \lim_{P \to 0} \frac{dD}{dP} = 1. \tag{11}$$

If the degree of the polynomial approximation is low enough, Equation 10 can be solved to give the correction formula $\mathcal{C}$ for an estimate of the actual evolutionary distance $\hat{P}$ as a function of the observed evolutionary distance D:

$$\hat{P} = \mathcal{C}(\mathcal{D}). \tag{12}$$

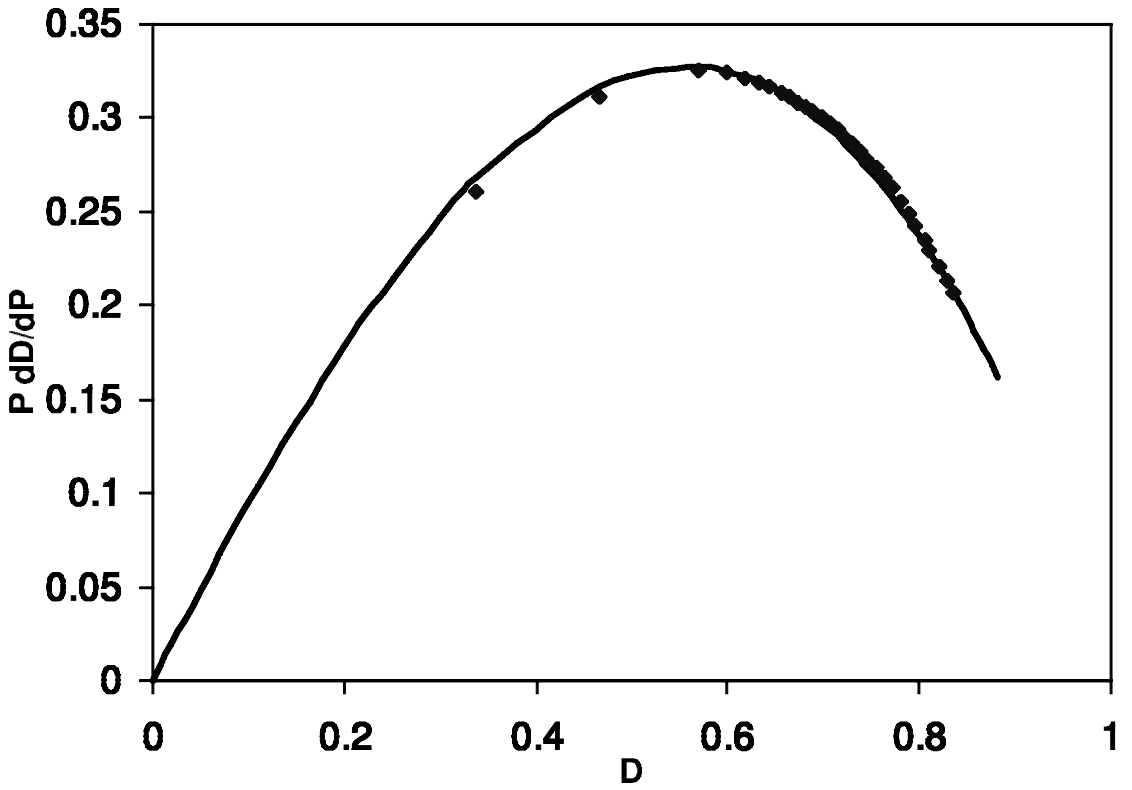

FIG. 1. Plot of the estimate of PdD/dP as a function of the expected distance (D) for each M_c matrix. The fitted line corresponds to the cubic approximation.

2.4. Approximation of the instantaneous rate matrix

After deriving the correction formula for evolutionary distance, for each clustering percentage c, we know the appropriate average level of evolutionary distance $\hat{P}_c$. Another form of the Chapman–Kolmogorov equation for Markov chains allows us to formulate the mutability matrix M as the exponential of some instantaneous rate matrix A multiplied by a divergence time P. Therefore, for each clustering percentage c, there exists an instantaneous rate matrix A_c, such that

$$M_c = \exp(A_c P_c). \tag{13}$$

The mutability matrix M_c and the estimated evolutionary distance $\hat{P}_c$ can provide an estimate of the corresponding instantaneous rate matrix A_c using

$$A_c = \frac{\ln M_c}{\hat{P}_c}, \tag{14}$$

but only when the logarithm exists (Devauchelle *et al.*, 2002). The logarithm of a mutability matrix may not exist when the number of substitutions is high. Additionally, to calculate the logarithm of the matrix, a numerical procedure by Parlett (described in Golub and Van Loan (1983)) may break down when there are repeated eigenvalues and requires the matrices to be close to unity. The accuracy of the estimated logarithm can be verified by checking that the exponential of the resulting matrix is close to the original matrix. The difference in the norm (largest singular value) between the exponential of the estimated log of the matrix and the original matrix is evaluated. We used the criteria[3] (built into Matlab)

[3]The constant 2.22×10^{-13} corresponds to the `tol` variable in the `logm` Matlab function, which is equal to $1000 \times$`eps`. The value of `eps` is equal to the distance from 1.0 to the next floating point number.

that for any clustering percentage c,

$$\frac{\| \exp(A_c \hat{P_c}) - M_c \|}{\| M_c \|} < 2.22 \times 10^{-13}. \tag{15}$$

The instantaneous rate matrix A_c can then be estimated from mutability matrix M_c and evolutionary distance P_c (obtained as described in Section 2.3) as long as the condition given in Equation 15 is satisfied. Each clustering percentage leads to a slightly different estimate of a rate matrix. The corresponding P_c evolutionary distances for each M_c matrix are approximated using the derivatives of the observed distances D_c so as to make the series of matrices "colinear" in that they satisfy the additivity property of a single Markov process as described in Equation 3. Under the assumption that a mutability matrix M_c reflects the result of the substitution process after evolutionary distance P_c, there should be a single, "universal" instantaneous rate matrix U such that for any clustering percentage c,

$$M_c = \exp(UP_c), \tag{16}$$

The universal instantaneous rate matrix U can be obtained from the individual instantaneous rate matrices by finding appropriate weights w_c, $(0 \leq w_c \leq 1)$, and defining

$$U = \sum_c w_c A_c \tag{17}$$

where,

$$\sum_c w_c = 1. \tag{18}$$

The quality of the universal instantaneous rate matrix U, and equivalently the weights w_c, is based on the sum of the relative differences

$$\sum_c \frac{\| \exp(U\hat{P_c}) - M_c \|}{\| M_c \|}, \tag{19}$$

which we seek to minimize.

3. RESULTS

We obtained the BLOSUM clusterings from the latest version of the database Blocks+v.13Aug01 and calculated the mutability matrix M_c for each clustering percentage c. We also calculated each corresponding average observed distance $\mathcal{D}(M_c(P))$ using Equation 4. We derived the correction formula for the evolutionary distance in order to determine the degree of actual evolutionary divergence P_c corresponding to each of these observed distances (Table 1). The estimated derivatives using Equation 8 are plotted in Fig. 1. Using least-squares, we were able to fit the resulting curve to a quadratic polynomial (with correlation coefficient $R^2 = 0.9978$) such that

$$\frac{P}{D}\frac{dD}{dP} = -0.571D^2 - 0.423D + 1 \tag{20}$$

(Fig. 1). Equation 20 is a separable differential equation and thus easily solved. With the initial conditions described above in Equation 11, we obtain the correction formula

$$\hat{P} = \frac{1.228D}{(1.744 + D)^{0.365}(1.004 - D)^{0.635}} \text{ for } D < 0.8813. \tag{21}$$

Equation 21 describes $\hat{P}$, the estimated frequency of actual substitutions, as a function of D, the observed frequency. The condition $D < 0.8813$ is given because our approximation does not apply above the highest range of divergence values we considered (corresponding to the clustering of 30%, see Table 1) and

TABLE 1. THE AVERAGE OBSERVED DISTANCE, AND AVERAGE ACTUAL DISTANCE DERIVED FROM THE OBSERVED FREQUENCY DISTRIBUTION OF AMINO ACID PAIRS FOR EACH CLUSTERING PERCENTAGE IN THE BLOCKS+AUG01 (MINUS.DAT) DATABASE

Cluster %	*Average observed distance* (D)	*PMB distance* (P)	*Relative residual with* $U = A_{68}$
30	0.8813	2.8835	0.0890
32	0.8747	2.7712	0.0790
34	0.8678	2.6619	0.0733
36	0.8632	2.5950	0.0824
38	0.8572	2.5120	0.0768
40	0.8494	2.4112	0.0727
42	0.8422	2.3247	0.0619
44	0.8348	2.2420	0.0813
46	0.8288	2.1786	0.0708
48	0.8202	2.0940	0.0903
50	0.8099	2.0007	0.0917
52	0.8049	1.9575	0.0794
54	0.7959	1.8849	0.0706
56	0.7890	1.8320	0.0552
58	0.7810	1.7739	0.0522
60	0.7711	1.7061	0.0299
62	0.7629	1.6529	0.0245
64	0.7545	1.6019	0.0160
66	0.7458	1.5509	0.0133
68	0.7384	1.5098	0.0000
70	0.7297	1.4638	0.0069
72	0.7224	1.4270	0.0101
74	0.7145	1.3882	0.0109
76	0.7060	1.3483	0.0236
78	0.6986	1.3150	0.0253
80	0.6888	1.2727	0.0276
82	0.6818	1.2436	0.0290
84	0.6727	1.2074	0.0442
86	0.6637	1.1726	0.0481
88	0.6552	1.1412	0.0548
90	0.6440	1.1012	0.0491
92	0.6327	1.0628	0.0620
94	0.6185	1.0166	0.0702
96	0.5987	0.9561	0.0838
98	0.5692	0.8734	0.1061
100	0.4671	0.6373	0.1411
n	0.3385	0.4119	0.1553

extrapolation to larger values of D will most likely not be accurate. Since the limit of applicability corresponds to a very large degree of sequence divergence in the range where alignments become questionable, extrapolation should not be necessary in most applications. The condition should therefore not restrict the applicability of the formula for actual protein alignments. The correction formula for multiple hits was compared to the Jukes and Cantor (1969) formula for amino acids,

$$P = \frac{19}{20} \ln\left(1 - \frac{20}{19} D\right), \tag{22}$$

which assumes an equal rate of substitution between amino acids, each with the same frequency. We also compared this correction with that of the original PAM matrices (Dayhoff *et al.*, 1978) as shown in Fig. 2.

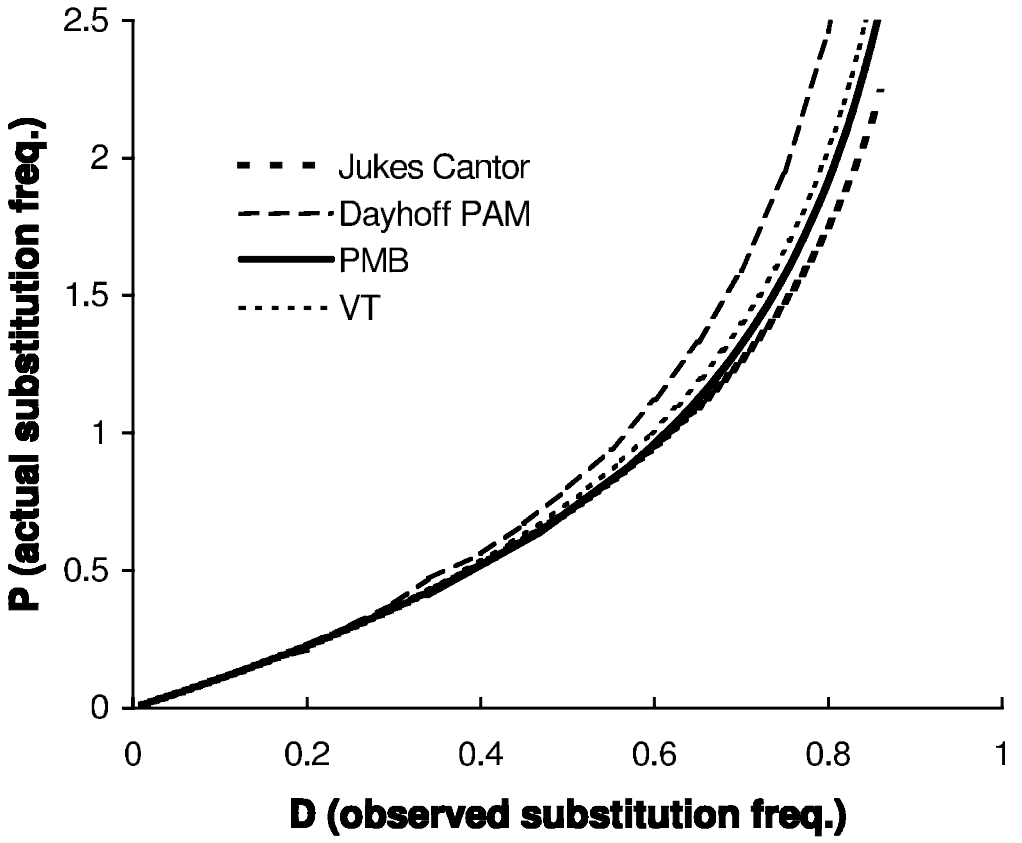

FIG. 2. Plot of the actual substitution frequency versus the observed substitution frequency. Shown are the estimates using the Dayhoff PAM matrix (long dashes), the Jukes and Cantor formula (short dashes), The VT matrix (dots) and our new approximation (solid line).

We see that the correction is more severe than that by Jukes and Cantor (1969) but not as severe as the Dayhoff *et al.* (1978) PAM approximation. We also calculated the observed expected distance for powers of the exponential of the (Muller and Vingron, 2000) VT rate matrix (Fig. 2) divergence. Surprisingly, the curve for our correction formula is extremely close to that of the VT matrix, which was obtained with a different method and with a different database.

The mutability matrices M_c we derived from the Blocks database all had positive eigenvalues, and all gave accurate logarithms that could be used for the approximation of a rate matrix. The relative error of the estimate of the log matrix could be estimated (Equation 15). This error was extremely small for all clusterings, well within tolerance, even for matrices with the highest levels of sequence divergence which are far from unity.

Figure 3 shows the relative residual between specified U matrices and each of the A_c matrices. We call these the cluster residuals. We found that the matrix with the smallest average cluster residual is given by the rate matrix A_{68}, which is given in Fig. 4. It defines the matrix for PMB(P) (Probability Matrix from Blocks) such that

$$\mathrm{PMB}(P) = \exp(0.01PU), \tag{23}$$

where P is the evolutionary distance in PAM units. We implemented the PMB matrix into the PROML and PROTDIST v3.6 programs of the PHYLIP package (Felsenstein, 2002). We named those versions of the programs PMBML, PMBMLK, and PMBDIST. They are available for download from the web site *www.uhnres.utoronto.ca/tillier/pmb/pmb.html.* We compared *lod* scoring matrices obtained from the PMB matrix with those for the VT matrix (Muller and Vingron, 2000). We found that they are quite similar,

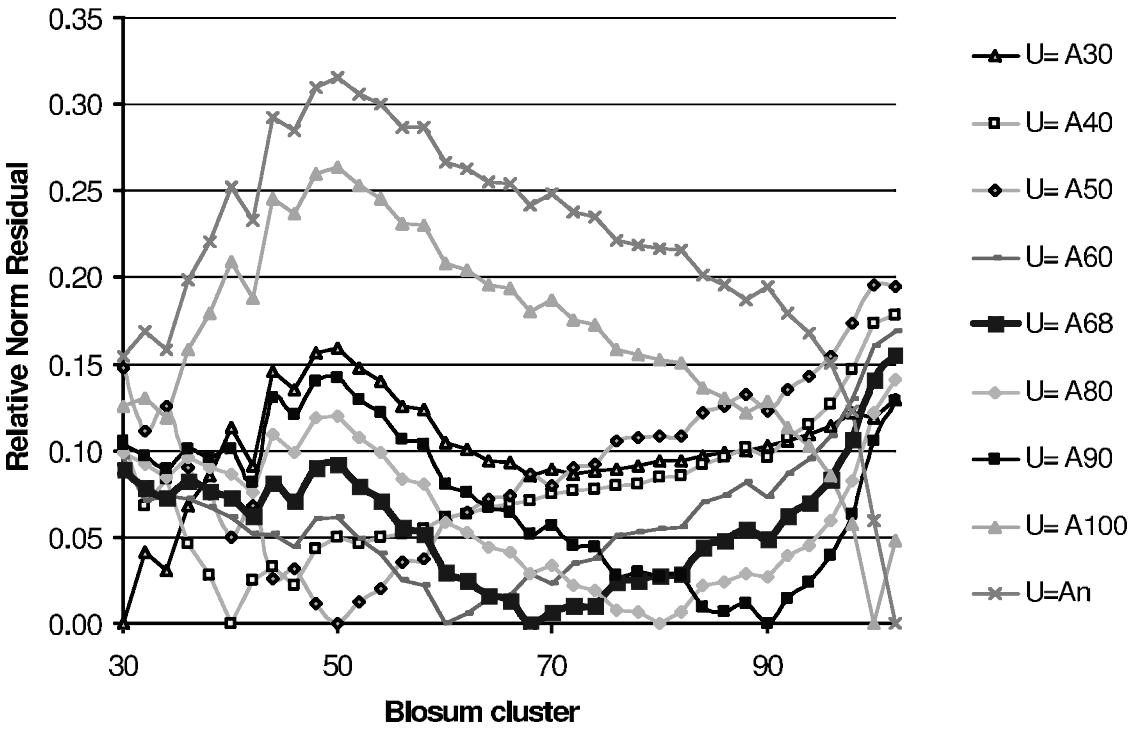

FIG. 3. Relative difference between Universal rate matrix and rate matrices from the mutability matrices. $U = A_c$ for $c \in \{30, 40, 50, 60, 72, 80, 90, 100, n\}$ or U taken as the average, and as the weighted average of all mutability matrices.

and an example is shown in Fig. 5 for the 250 PAM level. This is not surprising because although the VT matrix was obtained from a different database and with a much more complicated and computationally intensive approach, it was also found to give scoring matrices very similar to the BLOSUM matrices. This shows the strong influence of the matrix used to align protein sequences because the BLOSUM matrix was used to obtain the alignments upon which the VT matrix was derived (Muller and Vingron, 2000).

Replacement Amino Acid

Original Amino Acid	AA freq.	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	7.56%	-	444	216	218	249	380	544	1292	164	375	766	505	380	236	502	2196	759	45	210	1499
R	5.38%	624	-	530	311	84	1053	831	452	357	250	559	2675	201	135	274	733	637	118	264	354
N	3.77%	434	757	-	1468	124	726	735	1092	656	266	336	961	178	211	342	1570	946	51	296	418
D	4.47%	368	374	1237	-	87	545	1914	737	230	130	276	574	43	132	310	992	496	46	166	228
C	2.85%	660	158	164	136	-	110	92	284	150	297	476	168	132	307	112	545	366	48	184	540
Q	3.39%	846	1670	807	718	92	-	2250	673	508	211	739	1782	515	183	391	1155	914	135	335	505
E	5.35%	769	835	518	1600	49	1426	-	433	312	237	398	1222	130	117	525	799	650	72	190	448
G	7.80%	1251	311	527	422	104	293	296	-	161	109	291	325	89	189	261	886	291	55	126	201
H	3.00%	412	639	823	343	142	573	555	418	-	164	337	551	112	240	184	614	618	93	703	272
I	5.99%	473	224	167	97	141	120	212	142	82	-	3432	258	804	609	143	288	590	69	239	5186
L	9.58%	604	314	132	129	142	261	222	237	106	2145	-	257	1206	1015	184	242	418	192	271	1403
K	5.20%	734	2766	697	493	92	1162	1257	488	319	298	473	-	218	133	424	846	724	77	232	388
M	2.19%	1309	494	306	89	172	797	317	317	154	2197	5271	518	-	941	195	593	852	127	366	1509
F	4.50%	396	161	177	131	194	138	139	327	160	810	2160	153	458	-	214	447	327	354	1859	658
P	4.20%	903	350	307	329	76	316	667	485	131	203	419	525	101	229	-	940	589	87	193	489
S	6.82%	2432	578	868	650	228	574	626	1013	271	253	340	645	191	295	579	-	2750	99	253	462
T	5.64%	1016	608	632	393	185	549	616	402	329	626	710	667	331	261	439	3325	-	137	238	1416
W	1.57%	215	404	123	131	87	292	243	272	177	263	1167	256	177	1012	233	430	490	-	1381	445
Y	3.60%	441	394	311	207	146	316	282	274	587	398	722	336	223	2326	226	479	373	604	-	435
V	7.15%	1585	267	221	143	215	240	335	220	114	4345	1881	282	463	415	287	441	1118	98	219	-

FIG. 4. The instantaneous rate matrix for PMB. The first column gives the frequencies for each of the amino acids. Entries are multiplied by 10,000. Diagonal elements are such that the rows of the matrix add up to zero.

Difference in lod scores of VT(120) and PMB(120)

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	
	0	0	0	1	1	0	1	0	0	0	-1	0	-1	-1	0	0	1	-1	-2	0	A
		0	-1	-1	2	1	-2	0	2	-1	-1	0	-1	-1	0	-1	-1	-1	-2	-1	R
A	0		-1	0	1	0	0	0	0	0	-1	1	-1	-1	-1	1	0	-3	-1	-1	N
R	0	0		-1	-1	-1	1	0	1	-1	-1	0	-1	-2	-1	0	0	-2	0	-1	D
N	0	-1	0			2	0	1	1	2	0	1	0	0	1	0	2	2	3	2	1	C
D	0	0	0	0			0	0	-1	2	-2	-1	0	-1	-1	1	-1	0	-2	-1	-1	Q
C	1	0	0	0	0			-1	1	0	-1	-1	0	-1	-2	-1	0	-1	-2	-2	-1	E
Q	0	0	0	0	0	0			1	0	-1	-1	0	-1	-2	0	1	0	0	-2	0	G
E	1	0	0	0	1	1	-1			0	-1	1	0	0	1	1	1	-1	-1	3	0	H
G	0	0	0	-1	0	-1	0	0			0	0	-2	0	-1	-1	-1	0	-1	-1	0	I
H	1	1	0	0	0	1	1	0	0			1	-2	0	0	0	0	-1	-1	0	0	L
I	0	-1	-1	0	1	-1	-1	-1	-1	-1			0	-1	-2	-1	0	0	-2	-1	-1	K
L	0	0	-1	0	1	-1	0	0	0	0	0			1	0	-1	-1	0	0	-2	0	M
K	1	0	0	0	0	0	0	1	0	-1	-1	0			1	-3	-1	-1	0	1	0	F
M	0	0	0	-1	1	0	0	0	1	0	0	0	-1			0	1	0	-3	-2	-1	P
F	0	0	-1	0	-1	-1	0	-1	1	0	0	-1	0	0			0	0	-2	-1	-1	S
P	0	-1	-1	0	1	0	-1	0	0	0	0	-1	-1	-2	1			0	-3	-1	0	T
S	1	0	1	1	1	0	1	1	1	-1	0	1	0	-1	0	0			2	-1	-2	W
T	0	-1	0	0	2	0	0	1	-2	0	0	0	0	0	-1	0	1			1	0	Y
W	0	-2	-2	-1	2	-1	0	0	-1	-1	-1	-1	0	0	-3	-1	-2	2			1	V
Y	0	-1	-1	-1	-1	0	0	-1	3	0	0	0	0	0	-1	0	1	0	0			
V	0	-1	-1	-1	1	0	-1	-1	-1	0	0	0	0	0	-1	0	0	-1	1	0		
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	

Difference in lod scores of Blosum62 and PMB(120)

FIG. 5. Comparison of log odds (lod) score tables for the PMB matrix. The upper triangular matrix shows the difference between the 10 log 10 scoring matrix for VT(120) minus the 10 log 10 scoring matrix for PMB(120). The lower triangular matrix shows the difference between the 10 log 10 scoring matrix for BLOSUM62 minus the 10 log 10 scoring matrix for PMB(120) matrix. Absolute differences ≥ 2 are shaded.

We used bootstrap resampling (Efron, 1979) to assess the variability in our estimate of the PMB matrix. The bootstrap samples were created by randomly sampling blocks with replacement from the database. The overall size of the resulting database was unchanged from that of the original, but some blocks were represented more than once and some blocks not at all. For each of these samples, an estimated instantaneous rate matrix was obtained with the procedure described above, and its relative norm residual with PMB was calculated.

We also investigated how the estimate of the rate matrix varied with the growth of the Blocks database. Universal rate matrices were derived from several earlier versions (Blocks v5.0, v8.0, v9.0, v10.1, v11.0, and +v12Nov00). We performed bootstrap resampling of these databases to estimate the variability in the estimates. In Fig. 6, the mean residuals and the standard deviation over 100 bootstrap replicates are given for each of the database versions. The bootstrap residuals are generally only slightly higher than the cluster residuals (also shown in Fig. 6), indicating a small variability in the estimate. This variability does increase slightly with the earlier, smaller databases, but even the earliest database was already large enough to obtain an accurate estimate of the rate matrix. The Blocks v.5.0 database consists of 2,106 blocks and was the smallest available, so to investigate the behavior of the matrix estimates on even smaller databases, we took 20 random samples of this database to generate databases 50% and 25% its original size (in blocks). Each of these was bootstrapped, and the results are shown in Fig. 6. Only for those very small databases were the boostrap residuals significantly increased.

The relative residual between the universal matrices obtained for each version of the Blocks database and PMB (from Blocks+v.13Aug01) was also calculated and compared to the residuals with the VT matrix

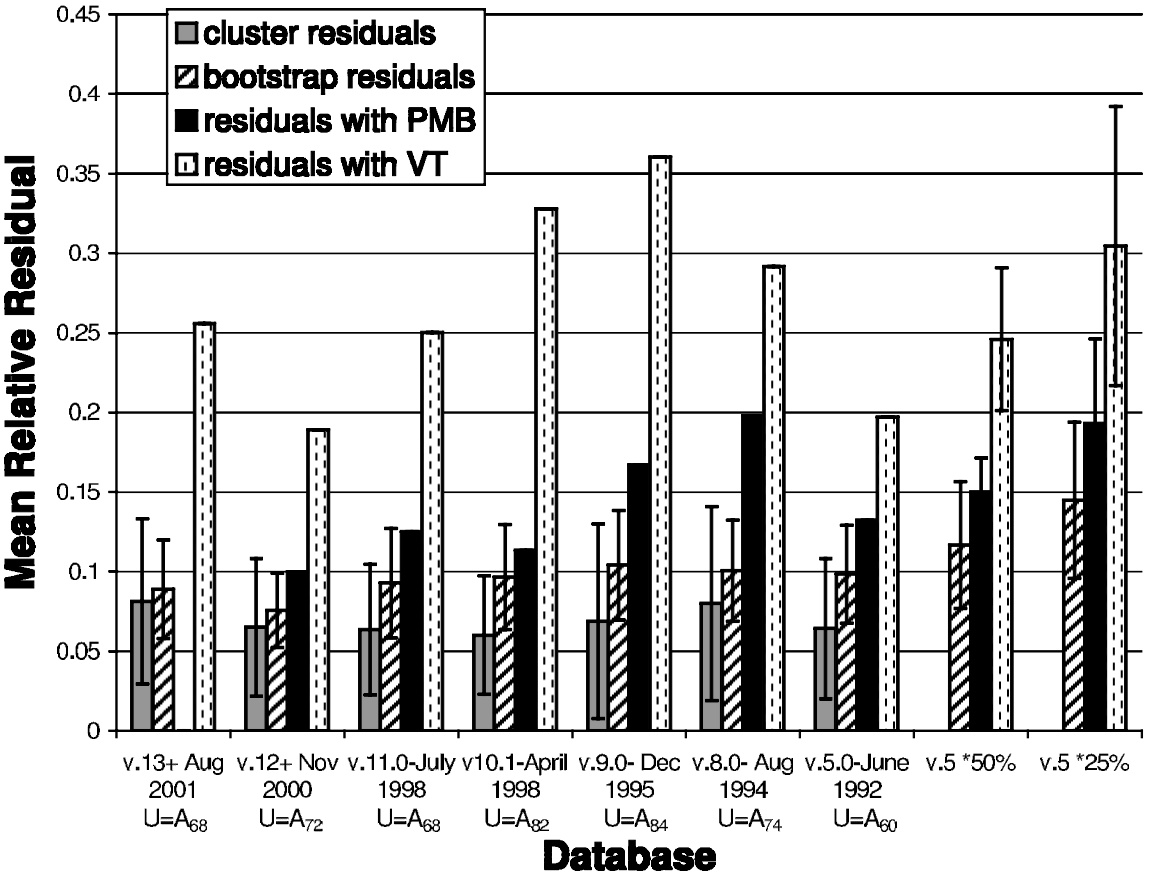

FIG. 6. Relative residuals between matrices are shown calculated from the different versions of the Blocks database. Grey bars indicate the mean residuals between the Universal matrix obtained for the database and of the matrices obtained with each clustering of the database. Hashed bars indicate the residuals with the Universal matrix obtained with that database and that obtained in a 100 bootstrap samples. Black bars indicate the residuals of PMB (the universal matrix obtained with the Blocks+v13Aug01 database) and the Universal matrix from earlier versions of Blocks. The dashed bars indicate the residuals of the VT matrix with the Universal matrices. Error bars indicate plus/minus one standard deviation. The v.5*50% and v.5*25% are databases obtained by randomly sampling one half and one quarter of the blocks in the the v.5.0 June 1992 database; this was done 20 times, and the mean and standard deviation of the residuals are plotted.

(Fig. 6). These comparisons show that the estimate of the universal rate matrix between the versions of the database is much smaller than the difference with VT confirming the result of Fig. 5 showing the differences between the lods scoring matrices obtained from PMB, BLOSUM, and VT.

4. DISCUSSION

The method of clustering sequences of given sequence identity percentages used in BLOSUM allows the derivation of substitution matrices that are applicable at larger average distances. These matrices are useful for determining relative weights for the different types of substitution and have been widely accepted for sequence alignments. For the matrices to be considered as an evolutionary model, however, we need to know the level of actual sequence divergence to which the relative weighing of substitutions correspond. We were able to derive a function for the actual evolutionary distance that rendered the BLOSUM matrices of observed substitution frequencies additive with respect to that overall average distance. The substitution matrices expressed as a function of the actual evolutionary distance then describe an evolutionary model. The database of Blocks contains an undefined mixture of sequence divergence values. The BLOSUM

approach makes inclusive subsets of the database by clustering sequences with an increasing level of sequence identity, and thus down-weighing the contribution from closely related sequences. The matrices are thus derived from sequence alignments that are completely inhomogeneous with respect to evolutionary time. From an evolutionary perspective, the clustering of the data should have a lower bound of sequence identity as well as an upper bound. A more accurate frequency matrix and average distance could be calculated given a more defined evolutionary period. The maximum likelihood methods and the resolvent method attempt to get around the problem of time inhomogeneity in the datasets either by assuming the phylogeny or by estimating evolutionary distances in pairwise sequence alignments. But all alignments are subject to inhomogeneity due to variable rates of substitution at different sites. We kept to the clustering method of Henikoff and Henikoff (1992) in order to derive a matrix compatible with the BLOSUM series. Regardless of how the data is partitioned, a substitution probability matrix can be derived from each subset. From these matrices, we can estimate the average observed distances and their derivatives in order to determine the average level of actual evolutionary divergence for the substitution probability matrices. Once the actual evolutionary distance corresponding to each clustering is known, then rate matrices can be estimated from the logarithms of probability matrices. The PMB is a single universal rate matrix that most closely describes the substitution process over all clustering of the BLOCKS database.

In the derivation of the PMB, we made several approximations. First, the derivative of the expected evolutionary distance was approximated using the five-point formula of Equation 6, and a function of the derivative was approximated again as a polynomial in Equation 20. These two approximations allowed us to describe the relationship between the actual substitution rate and the observed substitution rate as a differential equation with a real-valued solution. The five-point estimate is a very good estimate and justifiable since it is not significantly different from the three-point estimate (indicating convergence, data not shown). The quadratic least-squares approximation was almost a perfect fit and had the advantage of yielding a soluble differential equation for the actual evolutionary distance. The third approximation was made to estimate the logarithms of the mutability matrices. These approximations fell well within tolerance, even for mutability matrices that were not close to the unit matrix (i.e., for matrices corresponding to high evolutionary distances). The last approximation occurs when a single mutability matrix was used to approximate the universal rate matrix. We chose the rate matrix derived from the BLOSUM clustering percentage of 68 because it yielded the smallest deviation between estimated and actual mutability frequencies over the entire range of sequence divergence. That the matrices for the lowest and highest clustering levels were the least well estimated can be partly attributed to sampling error in these matrices due to the fact that entries outside of the diagonal are sparse for the high clustering percentages and the entries on the diagonal are sparse for the lower clustering percentages. The best approximation is, of course, to the matrix from which the universal matrix was derived, and the error increases for the matrices away from it. Overall, the average relative difference was less than 5%.

The accuracy of our model for the substitution process depends on the variability in that substitution process and the amount of data available to estimate it. To assess the variability in PMB, we obtained instantaneous rate estimates from bootstrap samples of the original database and those from earlier versions of the database. We found that the variability in the bootstrap estimates was only slightly higher than the variability of matrix estimates obtained from the different clusterings of the database. The results also showed that the estimates of the universal instantaneous rate matrices from earlier versions of the Blocks database were not statistically different from the PMB matrix. This explained our finding that there was only a small difference between the Blosum 62 scoring matrix derived from Blocks 5.0 in 1992 and the equivalent scoring matrix obtained from PMB (Fig. 5). Versions of the Blocks database pre-1992 were not available, so by sampling 50% and 25% of the blocks from the earliest available database, we created smaller databases from which to obtain estimates and to evaluate the relationship between the estimate reliability and the database size. Bootstrap resampling did show increased variability with estimates from these small databases, and the average residual with PMB was increased, but it was still smaller than the difference between PMB and VT. The estimate of the rate matrix from Blocks has been quite robust even though the size of the database has increased to over 20 times (in bytes) its size since 1992.

The PMB matrix was derived very simply and directly in the same manner as the BLOSUM matrix and with the additional estimation of corrected evolutionary distances to define an evolutionary model. The PMB model we derived is meant as an approximation that describes the average evolutionary behavior of the average amino acid in the average protein, but few amino acids in real proteins will actually

conform exactly to the model. Popular protein alignment applications such as Blast (Atschul *et al.*, 1990), Fasta (Pearson and Lipman, 1988), and ClustalW (Thompson *et al.*, 1994) all currently use the BLOSUM matrices. Since the alignment and the evolutionary model are closely linked (Mitchison, 1999), it makes sense to use an evolutionary model compatible with the alignment-scoring matrix. The PMB matrix has been derived using sequences in the complete range of sequence divergence and should be more accurate than the PAM matrix (Dayhoff *et al.*, 1978) or the JTT matrix (Jones *et al.*, 1992). The PMB matrix is easily incorporated into applications that use such matrices, including those applications in the popular Phylip package for phylogenetic analysis (Felsenstein, 2002). The improvement for the analysis of any specific protein family will depend on how closely the evolution of the sequences follows the average for the proteins in the Blocks database upon which the PMB model is based. The methods that are described here are fast and easily implemented and can be used to develop more specific substitution matrices applicable to protein families and protein domains.

ACKNOWLEDGMENTS

Many thanks to Franz Lang (Université de Montréal) and Andrew Roger (Dalhousie University, Halifax) for testing the implementations of the PMB matrix in the Phylip package and our eternal thanks to Joe Felsenstein for providing these programs. Thank you to several reviewers for your suggestions on improving the manuscript.

REFERENCES

Adachi, J., and Hasegawa, M. 1996. Model of amino acid substitution in protein encoded by mitochondrial DNA. *J. Mol. Evol.* 42, 459–468.

Adachi, J., Wadell, P.J., Martin, W., and Hasegawa, M. 2000. Plastid genome phylogeny and a model of amino acid substitution for proteins encoded by chloroplast DNA. *J. Mol. Evol.* 50, 348–358.

Altschul, S.F. 1991. Amino acid substitution matrices from an information theoretic perspective. *J. Mol. Biol.* 219, 555–565.

Altschul. S.F., Gish, W., Miller, W., Myers, E.W., and Lipman, J.D. 1990. Basic local alignment search tool. *J. Mol. Biol.* 215, 403–410.

Arvestad, L., and Bruno, W.J. 1997. Estimation of reversible substitution matrices from multiple pairs of sequences. *J. Mol. Evol.* 45, 696–703.

Burden, R.L., and Faires, J.D. 1985. *Numerical Analysis*, PWS, Boston, MA.

Dayhoff, M.R., Schwartz, R., and Orcutt, B. 1978. A model of evolutionary change in protein. *Atlas of Protein Sequences and Structure* 5, 345–352.

Devauchelle, C., Grossmann, A., Hénaut, A., Holschneider, M., Monnerot, M., Risler, J.L., and Torrésani, B. 2002. Rate matrices for analyzing large families of protein sequences. *J. Comp. Biol.* 8, 381–399.

Efron, B. 1979. Bootstrap methods: Another look at the jackknife. *Ann. Statist.* 7, 1–26.

Felsenstein, J. 2002. Phylip (phylogeny inference package) version 3.6.3. Distributed by the author.

Golub, G.H., and Van Loan, C.F. 1983. *Matrix Computation*, p. 384, Johns Hopkins University Press.

Gonnet, G., Cohen, M., and Benner, S. 1992. Exhaustive matching of the entire protein sequence database. *Science* 256, 1443–1445.

Henikoff, S., and Henikoff, J. 1992. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA* 89, 10915–10919.

Henikoff, S., Henikoff, J., and Pietrokovski, S. 1999. Blocks+: A non-redundant database of protein alignment blocks derived from multiple compilations. *Bioinformatics* 15(6), 471–479.

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

Jukes, T.H., and Cantor, C.R. 1969. *Evolution of Protein Molecules*, 21–132, Academic Press, New York.

Krause, A.J., Stoye, J., and Vingron, M. 2000. The SYSTERS protein sequence cluster set. *Proc. Natl. Acad. Sci. USA* 28, 270–272.

Mitchison, G.J. 1999. A probabilistic treatment of phylogeny and sequence alignment. *J. Mol. Evol.* 49, 11–22.

Muller, T., and Vingron, M. 2000. Modeling amino acid replacement. *J. Comp. Biol.* 7, 761–776.

Pearson, W.R., and Lipman, D.J. 1988. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA* 85, 2444–2448.

Thompson, J.D., Higgins, D.G., and Gibson, T.J. 1994. CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting position-specific gap penalties and weight matrix choice. *Nucl. Acids Res.* 22, 4673–4680.

Thorne, J.L. 2001. Models of protein sequence evolution and their applications. *Curr. Opin. Gen. Dev.* 10, 602–605.

Whelan, S., and Goldman, N. 2001. A general empirical model of protein evolution derived from multiple protein families using a maximum likelihood approach. *J. Mol. Evol.* 18, 691–699.

Yang, S., Nielsen, R., and Hasegawa, M. 1998. Models of amino acid substitution and applications to mitochondrial protein evolution. *Mol. Biol. Evol.* 15, 1600–1611.

Address correspondence to:
Elisabeth R. M. Tillier
Ontario Cancer Institute
University Health Network
620 University Avenue, Suite 703
Toronto, Ontario
M5G 2M9 Canada

E-mail: e.tillier@utoronto.ca