Bioinformatics, 35(15), 2019, 2562–2568
doi: 10.1093/bioinformatics/bty1031
Advance Access Publication Date: 24 December 2018
Original Paper

Phylogenetics

Ancestral sequence reconstruction: accounting for structural information by averaging over replacement matrices

Asher Moshe and Tal Pupko*

Department of Cell Research and Immunology, School of Molecular Cell Biology and Biotechnology, George S. Wise Faculty of Life Sciences, Tel Aviv University, Tel Aviv 6997801, Israel

*To whom correspondence should be addressed.

Associate Editor: Russell Schwartz

Received on July 26, 2018; revised on December 3, 2018; editorial decision on December 8, 2018; accepted on December 16, 2018

Abstract

Motivation: Ancestral sequence reconstruction (ASR) is widely used to understand protein evolution, structure and function. Current ASR methodologies do not fully consider differences in evolutionary constraints among positions imposed by the three-dimensional (3D) structure of the protein. Here, we developed an ASR algorithm that allows different protein sites to evolve according to different mixtures of replacement matrices. We show that assigning replacement matrices to protein positions based on their solvent accessibility leads to ASR with higher log-likelihoods compared to naïve models that assume a single replacement matrix for all sites. Improved ASR log-likelihoods are also demonstrated when solvent accessibility is predicted from protein sequences rather than inferred from a known 3D structure. Finally, we show that using such structure-aware mixture models results in substantial differences in the inferred ancestral sequences.
Availability and implementation: http://fastml.tau.ac.il.
Contact: talp@tauex.tau.ac.il
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Understanding how genes and genomes evolve is a major goal in molecular evolution. Ancestral sequences can help elucidate molecular pathways that evolved millions to billions of years ago (Liberles, 2007). Other than the valuable evolutionary knowledge gained from these sequences, ancestral proteins may contain desirable properties that modern proteins lack, such as broader substrate range and higher thermostability. Therefore, they can be used as a good starting point for protein engineering (Gumulya and Gillam, 2017; Ogawa and Shirai, 2014).

Inferring ancestral sequences can be challenging. While it is possible to use molecular paleontology, i.e. the extraction and recovery of DNA information from fossils, this method still has many obstacles to pass before it can be widely used (Zaucha and Heddle, 2017). Thus, a method called ancestral sequence reconstruction (ASR) was developed and so far, it is the best way to deduce the origins of modern proteins (Liberles, 2007). In ASR, the ancestral sequences are inferred by using the extant sequences, a phylogenetic tree and a model of sequence evolution (Pupko *et al.*, 2008).

The evolutionary rate at a specific site in a protein-coding gene dictates the number of substitutions that this site experiences along its evolution. While early modeling approaches to sequence evolution assumed, for simplicity, that all sites evolve at the same rate, this was shown not to be the case more than 50 years ago (Fitch and Margoliash, 1967). Rate variation among sites is affected by several factors, among them are functional and structural constraints (Yang, 1996). Currently, the Gamma distribution is most commonly used to model among site rate variation (Yang, 1996).

Models that account for among sites rate variations and assume a single replacement matrix across all sites are still an oversimplification of the evolutionary dynamics. Specifically, such models ignore the biological intuition and knowledge that sites in a protein are subjected to different evolutionary constraints affected by differences in biochemistry and structure. Amino-acid replacement

propensities substantially vary among different structural parts of the protein and mainly due to solvent accessibility (Goldman *et al.*, 1998).

Several studies have previously suggested models which allow for the replacement matrix to vary over protein sites. Koshi and Goldstein (1995) use maximum-likelihood approaches to compute site-specific replacement matrices for different regions within proteins, either based on secondary structures or based on solvent accessibility. Soyer *et al.* (2003) used multiple matrices to represent different sites in the GPCR protein family. Juritz *et al.* (2013) showed that similar conformations in different proteins are characterized by similar site-specific replacement matrices.

Another approach to model replacement-propensity rate variation was suggested by Le and Gascuel (2010). They proposed using an array of replacement matrices that capture the evolutionary constraints of different regions of proteins. Here, we utilized their approach in order to achieve more accurate ASRs and integrated the methodology in FastML. This is done by using structural data to calculate the most probable amino acid in each position using the most fitting replacement matrix from the given array of matrices. We show that even when structural data are missing, it is better to compute ASR based on structure information prediction rather than to infer ASR ignoring structural data.

2 Materials and methods

2.1 Maximum-likelihood based marginal reconstruction

2.1.1 A single substitution model, no rate variation among sites

The input for the algorithm is a phylogenetic tree and a multiple sequence alignment of the extant sequences (E). The algorithm assumes that alignment positions evolve independently given the phylogenetic tree and therefore we describe our algorithm for ASR for a single amino-acid position. It is also assumed that amino-acid replacement probabilities are determined according to a continuous time Markov process for amino acids such as the AAJC, the Jukes and Cantor (1969) model for amino-acids, JTT (Jones *et al.*, 1992), mtREV (Adachi and Hasegawa, 1996), cpREV (Adachi *et al.*, 2000), WAG (Whelan and Goldman, 2001), DAY (Dayhoff *et al.*, 1978) or LG (Le and Gascuel, 2008). These models provide for each amino acid a stationary probability π_a and for each pair of amino acids a and b, the probability that amino acid a is replaced by amino acid b after an evolutionary time t: $P(a \to b|t)$. Our goal in this algorithm is to infer the most likely assignment in each ancestral node. Note, that in this work, we refer to marginal reconstructions as opposed to joint reconstruction. In marginal reconstruction, the probability of each ancestral character assignment in each node is averaged over all possible assignments in the other internal nodes of the tree (Pupko *et al.*, 2000, 2002; Yang *et al.*, 1995).

To explain the likelihood computations, consider the phylogenetic tree of Figure 1. Here, the probability of the extant characters (the tree likelihood) is:

$$P(E) = \sum_{a_1}\sum_{a_2} P(a_1)P(a_1 \to L|t_3)P(a_1 \to a_2|t_4)P(a_2 \to K|t_1) P(a_2 \to L|t_2)\Big) \quad (1)$$

where a_1 and a_2 are the amino acids assigned to nodes A_1 and A_2, respectively. The first step in estimating ancestral sequences is to compute the posterior probability of each character in each of the tree nodes. These posterior probabilities are computed using Bayes theorem. In the above example, the posterior probability of each character in the root node is computed according to following equation:

$$P(a_1|E) = \frac{P(E|a_1)P(a_1)}{P(E)} = \frac{P(E|a_1)\pi_{a_1}}{P(E)} = \frac{\pi_{a_1}\sum_{a_2} P(a_1 \to L|t_3)P(a_1 \to a_2|t_4)P(a_2 \to K|t_1)P(a_2 \to L|t_2)}{P(E)} \quad (2)$$

The most likely character of the root is computed by $\text{argmax}_a(P(a|E))$.

Using Felsenstein's dynamic programming algorithm (Felsenstein, 1981), the posterior probabilities at the tree root (and thus the most likely ancestor) can be computed in $O(n)$ where n is the number of sequences. We can re-root the tree in each possible node and repeat the above computation, leading to an $O(n^2)$ algorithm to find the ancestral sequences at all internal nodes. In this study, we describe a more efficient dynamic programming algorithm designed to find the ancestors in all the nodes simultaneously in $O(n)$. Similar dynamic algorithms were previously utilized by us for the task of maximum-likelihood tree inference using expectation maximization (Friedman *et al.*, 2002).

We divide the calculations into three parts: a post-order tree traversal we call 'Up', a pre-order traversal we call 'Down' and another tree traversal (for which the order is not important) we call 'Marginal'. The 'Up' computations are those suggested by Felsenstein (1981) to calculate tree likelihoods.

- For each of the extant nodes (i.e. leaves) v we set $\text{UP}[v][i] = \delta ij$, where j is the observed character at extant node v and δij is 1 if $i = j$ and 0 otherwise.
- For ancestor v for which both sons (S^1, S^2) have been calculated and for character i we set:

$$\text{Up}[v][i] = \Big(\sum_j P(i \to j|t_{v \to S^1})\text{Up}[S^1][j]\Big)\Big(\sum_j P(i \to j|t_{v \to S^2})\text{Up}[S^2][j]\Big) \quad (3)$$

While the 'Up' algorithm provides probabilities of a node given all extant taxa that are its descendants, the 'Down' algorithm provides probabilities of a node given all the extant taxa that are not its descendants (see Fig. 2, which shows the 'Up' and 'Down' for a specific node n).

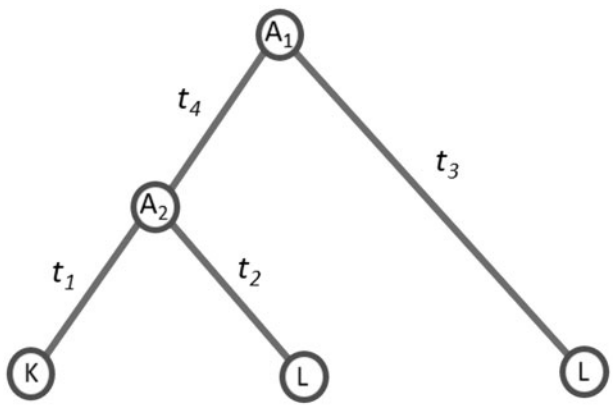

Fig. 1. Example for a phylogenetic tree. The tree shows a single position with three extant sequences that contain K/L and two internal nodes, A_1 and A_2

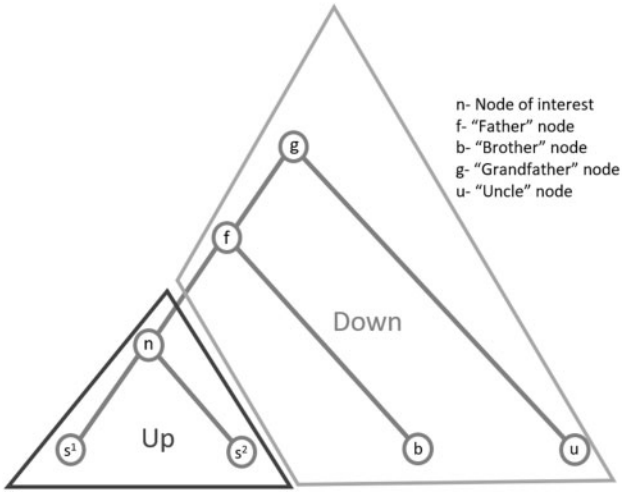

Fig. 2. Tree division to 'Up' and 'Down'

- For the root node, we set Down[root][i] = 1 for all i.
- For a node v, for which the 'Down' component of its father node, f, was already computed we set:

$$\text{Down}[v][i] = \sum_j P(i \to j|t_{g\to f})\text{Down}[f][j] \sum_k P(i \to k|t_{f\to b})\text{Up}[b][k] \quad (4)$$

  where g is the 'grandfather' node, and b is the 'brother' node (Fig. 2)
- For the sons of the root (which do not have a grandfather) the calculation is simply:

$$\text{Down}[v][i] = \sum_k P(i \to k|t_{f\to b})\text{Up}[b][k] \quad (5)$$

Once the 'Up' and 'Down' components are computed, we calculate the 'Marginal' component for each ancestor node:

$$\text{Marg}[v][i] = \pi_i(\text{Up}[v][i]) \sum_j (\text{Down}[v][j])P(i \to j|t) \quad (6)$$

Where t is the branch connecting node i to its father. For the root the marginal is simply $\pi_i(\text{Up}[v][i])$.

Thus, the 'Up' component for a node v refers to the sub-tree below this node, the 'Down' component refers to the remaining of the tree excluding the edge connecting v to its father node and the marginal component, combines these two factors to obtain the probability distribution of characters for a specific node given the entire data. More explicitly, the above 'Marginal' components provide P(a)P(E|a) for each character a in each node. To get P(a|E), we need to divide it by P(E) which is easily computed by $P(E) = \sum_a P(a)P(E|a)$ over all marginal probabilities in that node [as seen in Equation (1)].

2.1.2 Adding rate variation among sites

To add rate variation among sites, we assume a discrete Gamma distribution controlled by a shape parameter α (Uzzell and Corbin, 1971). The gamma distribution is divided to n discrete categories (by default we use $n = 8$). The categories are divided so that each category weight is $1/n$. We denote $R = \{r_1, \ldots, r_n\}$ the set of all possible rates. In the example tree of Figure 1, we use conditional probability to include the rate in the probability computation:

$$P(E|r) = \sum_{a_1 \in A_1} \sum_{a_2 \epsilon A_2} P(a_1)P(a_1 \to L|t_3 r)P(a_1 \to a_2|t_4 r)P(a_2 \to K|t_1 r)P(a_2 \to L|t_2 r) \quad (7)$$

It is similar to the P(E) calculation shown above, but here we multiply each branch length by the rate r. The unconditional probability will therefore be:

$$P(E) = \sum_{r\in R} P(E|r)P(r) \quad (8)$$

And therefore, for the example in Figure 1, we obtain:

$$P(E) = \sum_{r\in R} \sum_{a_1 \in A_1} \sum_{a_2 \epsilon A_2} \pi_{a_1} P(a_1 \to L|t_3 r)P(a_1 \to a_2|t_4 r)P(a_2 \to K|t_1 r)P(a_2 \to L|t_2 r)P(r) \quad (9)$$

The estimations of a specific ancestor a when incorporating rate variation are:

$$P(a|E) = \frac{P(E|a)\pi_a}{P(E)} = \frac{\sum_r P(E|a,r)P(a)P(r)}{P(E)} \quad (10)$$

For each discrete rate r, the algorithm is the same as the algorithm described above except that here, the 'Up', 'Down' and 'Marginal' components are calculated n times (once for each rate category) separately. In each such a computation, instead of using the branch length t, the branch length used is $t \cdot r$. The total marginal probabilities $P(E|a)\pi_a$ for a specific node v are computed by

$$P(E|a)P(a) = \sum_r \text{Marg}[v][a][r]P(r) \quad (11)$$

2.1.3 Using rate variation and multiple replacement matrices

To use multiple matrices, we take the weight of each matrix per position as additional input. The weights of all matrices in each position should sum to a total of 1. We denote the array of matrices $M = \{m_1, \ldots, m_n\}$ and the corresponding weight vectors $W = \{w_1, \ldots, w_n\}$, each entry contains the weight of its respective matrix per position. In the example in Figure 1, we first condition on a specific matrix m in addition to the rate variation:

$$P(E|r,m) = P_m(E|r) = \sum_{a_1 \in A_1} \sum_{a_2 \epsilon A_2} P_m(a_1)P_m(a_1 \to L|t_3 r)P_m(a_1 \to a_2|t_4 r)P_m(a_2 \to K|t_1 r)P_m(a_2 \to L|t_2 r) \quad (12)$$

It is similar to $P(E|r)$ shown above, but here we use the replacement probabilities and stationary probabilities based on a specific m matrix. The unconditional probability will be:

$$P(E) = \sum_{i\in\{1..n\}} \sum_{r\in R} P_{m_i}(E|r)P(r)w_i \quad (13)$$

And therefore, for the example in Figure 1, we obtain:

$$P(E) = \sum_{i\in\{1..n\}} \sum_{r\in R} \sum_{a_1 \in A_1} \sum_{a_2 \epsilon A_2} \pi_{a_1}^{m_i} P_{m_i}(a_1 \to L|t_3 r)P_{m_i}(a_1 \to a_2|t_4 r)P_{m_i}(a_2 \to K|t_1 r)P_{m_i}(a_2 \to L|t_2 r)P(r)w_i \quad (14)$$

For each replacement matrix m the algorithm is the one calculated for the single replacement matrix with discrete rate variation. It means that for every matrix in the model array, both 'Up', 'Down' and 'Marginal' are calculated for all rates. The total marginal probabilities for a specific node v are computed by:

$$P(E|a)\pi_a = \sum_m \sum_r \text{Marg}[v][m][r][a]P(r)w_m \quad (15)$$

Of note, the matrices and their associated weights can be estimated directly from the data analyzed using maximum-likelihood.

However, the matrices and weights can be also obtained based on data external to sequences being analyzed. Here, we apply this latter case, where we use pre-computed matrices for buried and exposed protein regions and the weights are computed based on the protein solvent accessibility values (see below). Note, that using pre-computed empirical amino-acid matrices is the standard in the field of phylogenomics, e.g. when analyzing with the LG, WAG or JTT matrices. As these matrices and weights are not directly estimated from the sequence data, they are not considered as free parameters when comparing different models in model-selection procedures. Of note, for the current work, we use the same test data as that used in Le and Gascuel (2010). Thus, test data and the training data used to estimate the buried and exposed matrices are truly disjoint. This provides further justification for not considering these matrices as additional free parameters.

2.2 Replacement matrices based on solvent accessibility

Solvent accessibility was extracted from structural data using DSSP (Kabsch and Sander, 1983; Touw *et al.*, 2015). Absolute solvent accessibility values were normalized using maximum solvent accessibility values calculated empirically for each amino acid (Tien *et al.*, 2013). When structural data were unavailable (or to simulate such cases), solvent accessibility was predicted using Sable (Adamczak *et al.*, 2004). Predictions were performed on a consensus sequence that contained for each position, the most common character. Positions were dichotomized to either 'Buried' or 'Exposed' using a 10% relative solvent accessibility as threshold (Goldman *et al.*, 1998). It is possible that even when the structural data were available, the solvent accessibility of some positions within the multiple sequence alignment was ambiguous, e.g. due to the introduction of short insertions. In such cases, these positions were assigned 50% weight buried and 50% weight exposed.

2.3 Datasets and trees

We analyzed 148 protein datasets, which were chosen because they included very few gaps and structural data for at least one of the extant sequences were available. These data were previously used by Le and Gascuel (2010). For each dataset, the phylogenetic tree was inferred using PhyML-structure with the EX2 model and the CONF/MIX mode with no among site rate variation (Le and Gascuel, 2010). Notably, the EX2 model is identical to the BE model described below. For each tree topology, the branch lengths were optimized under the maximum-likelihood criterion, under the specified model. The C++ code for branch length optimization under each of these models was added to the FastML program (Ashkenazy *et al.*, 2012).

2.4 Markovian models

A total of eight amino-acid replacement models were tested, among which five have previously been used in ASR: (i) the AAJC, the Jukes and Cantor (Jukes and Cantor, 1969) model for amino-acids; (ii) DAY (Dayhoff *et al.*, 1978); (iii) JTT (Jones *et al.*, 1992); (iv) WAG (Whelan and Goldman, 2001) and (v) LG (Le and Gascuel, 2008). Here, we tested three additional models, all of which are based on combining two amino-acid replacement matrices: the E matrix that models surface exposed sites and the B matrix that models buried sites (Le and Gascuel, 2010). The models differ in how the weights are assigned to each position. The BE model classifies each position as either exposed or buried based on a 10% accessibility threshold (see above). In buried (exposed) position, the weight of the B matrix is set to 1 (0), while the weight of the E matrix is set to 0 (1). In the MIX1 model, we designated solvent accessibility between 10–20% as uncertain and applied 0.5 weights for both the B and the E matrices. Finally, in the MIX2 model, we designated two uncertainty regions. For positions with solvent accessibility between 10–20%, we applied a weight of 0.667 for the B matrix and a weight of 0.333 for the E matrix. For positions where the solvent accessibility ranged from 20% to 30%, we applied weights of 0.333 and 0.667 for the B and E matrices, respectively. As noted above, although the new models use multiple matrices, the number of parameters is the same as for the classic models because the weights are not free parameters estimated as part of the probabilistic evolutionary model.

3 Results

We predicted the ancestral sequences of all datasets using five protein models for nuclear encoded proteins and three structure-aware models, which account for the solvent accessibility of each protein site (listed in Fig. 3). Briefly, the models assign different weights to two replacement matrices: 'Buried' (B) and 'Exposed' (E). The simplest of these models is the BE model, which assigns a 'Buried' replacement matrix to buried positions and an 'Exposed' matrix to exposed positions (see Materials and Methods Section). The BE model was first applied using solvent accessibility data extracted from the three-dimensional (3D) structure of each analyzed protein. The single matrix LG model (Le and Gascuel, 2008) was chosen as a baseline to compare the performance of the BE model to the simpler models that assume a single amino-acid replacement matrix for all sites. Performance was estimated using the log-likelihood score for the most likely marginal reconstruction at the root of each dataset. While all models which comprise of a single replacement matrix scored better than LG in less than 20% of the datasets, the BE model scored higher than LG in more than 90% of the datasets tested (Fig. 3).

Even though the BE model had higher log-likelihoods than LG, it is plausible that the differences in the log-likelihoods are negligible. We thus next compared the log-likelihood differences between all models and LG (Fig. 4). While the other models scored mostly lower than LG (negative difference) the BE model scored substantially higher in most datasets. Of note, log-likelihood differences of

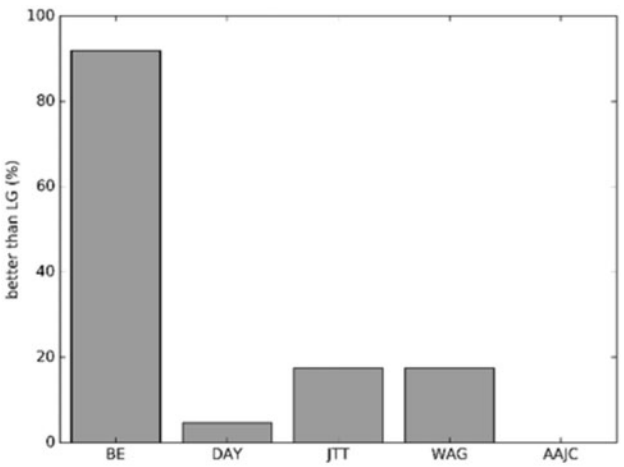

Fig. 3. Comparison of all models to the LG model. The new BE model with the structural data scored higher than LG in over 90% of the datasets tested. For each model the Y axis shows the percentage of datasets which had a higher log-likelihood score compared to the LG model

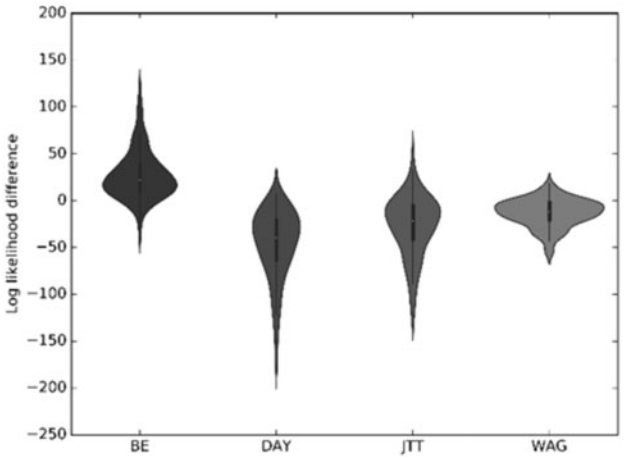

Fig. 4. Comparing log-likelihood scores of all models to the LG model, BE model relies on structure data. The Y axis is the difference in log-likelihood compared to the LG model. The AAJC model scored significantly lower than the other models and is therefore not shown

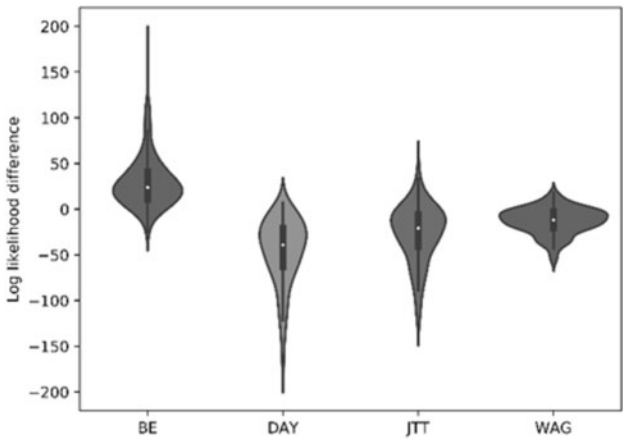

Fig. 5. Comparing log-likelihood scores of all models to the LG model, in which the BE model relies on predicted data. The improved log-likelihood scores are comparable to those achieved when the 3D data are available

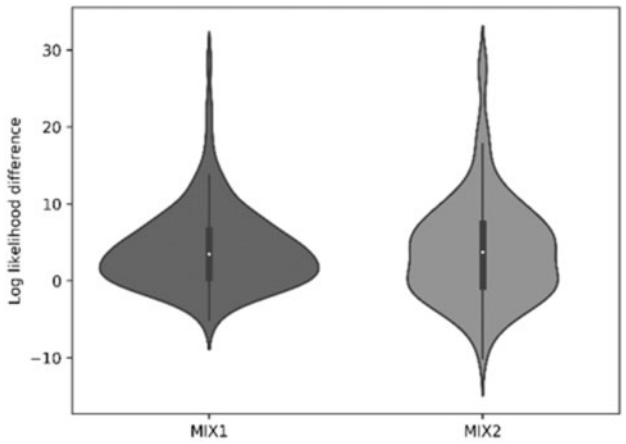

Fig. 6. Comparing log-likelihood scores of the mix models to the BE model. Adding a range of uncertainty and applying mixture of both matrices to it, is mostly beneficial in terms of log-likelihood

10 points or higher are considered highly significant in such cases as the one considered here, in which there is no difference in the number of free parameters between the compared models.

In the above analysis, all tree topologies were estimated using the BE model. It was previously shown that the tree topology may vary depending whether the BE or LG model is used (Le and Gascuel, 2010). To verify that the superiority of the BE model over LG does not stem from the fact that the BE model was used to reconstruct the tree topologies, we repeated the above analysis, this time when all tree topologies were reconstructed using the LG model as implemented in PhyML 3.0 (Guindon *et al.*, 2010). The superiority of the BE model remains even when LG is used to reconstruct tree topologies (Supplementary Fig. S1).

To test the BE model in cases where the 3D structure was unavailable for any of the extant sequences, we analyzed the 148 datasets as above, but this time, their solvent accessibility was predicted from the consensus sequence using Sable (Adamczak *et al.*, 2004). We compared the log-likelihood results to those obtained using the single-matrix models (Fig. 5). Similar to the case in which the structural data are known, the BE model obtained substantially higher log-likelihood scores compared to the single-matrix model. Interestingly, in 84 out of the 148 analyzed protein datasets, the BE model with predicted solvent accessibility had slightly higher log-likelihood score than the BE model for which the solvent accessibility was retrieved from the 3D structure.

The above BE model classifies each position as either exposed or buried based on a strict 10% solvent accessibility cutoff. However, positions that are 20% solvent-exposed are expected to experience different selective constraints compared to positions which are 50% solvent-exposed. It is also expected that some of the positions that have 20% solvent accessibility have accumulated amino-acid replacements similar to buried positions, while others, similar to exposed positions. We thus next tested whether accounting for such uncertainty in classifying positions to either 'Buried' or 'Exposed' can benefit ASR. Specifically, we tested two mixing variants, MIX1 and MIX2, for which positions with intermediate solvent accessibility are modeled according to Equations (12)–(15), i.e. the probability of each ancestral character in these positions is a weighted average over the two matrices B and E (see Materials and Methods Section). When comparing the log-likelihood score to the BE model, a clear significant improvement for most datasets was observed (Fig. 6). Among these two models, MIX1 had a higher average log-likelihood than MIX2, but the differences were insignificant.

We next tested the extent of differences in reconstructed sequences when comparing the LG and MIX1 models. Position differ between models, if the models do not agree for the position in at least one internal node. Out of all protein datasets analyzed, differences were observed in 98% (145/148). The distribution of the number of positions that differ in at least one internal node between the LG and MIX1 models out of the total length is shown in Figure 7. As can be seen, for some datasets, the fraction of affected positions was higher than 30%.

While the overall likelihood score is higher for the BE model compared to the LG model, some insights may be gained by analyzing the results in light of the physico-chemical properties of each position. We first asked whether the log-likelihood gain (i.e. the difference between the log likelihoods of the BE and LG models per position) is the same for positions that differ in their secondary structure. We classified each position as either 'extended' (Beta strand, 'E' in the protein data bank), 'helical' (Alpha helix, 'H' in the protein data bank), 'other' (T, B, S, I, G, '?' or '.' In the protein data bank), as done by Le and Gascuel (2010). Interestingly,

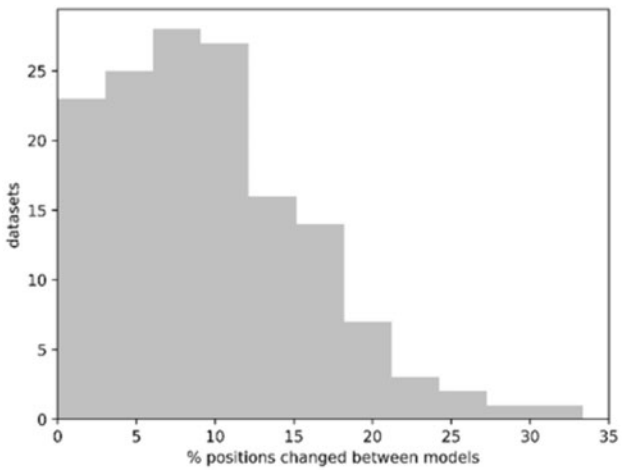

Fig. 7. Percentage of positions that differ between the LG and MIX1 models. For each of the 148 datasets, we computed the percentage of positions that differ between the two models. Shown is the distribution of these percentages

although our model does not take secondary structure into account explicitly, we found that the average log-likelihood gain for 'helical' positions was 0.27 while for 'extended' it was 0.2 and only 0.1 for the 'other' category ($P < 1E–20$; ANOVA). We next compared log-likelihood gains for 'Buried' versus 'Exposed' positions. The average log likelihood difference between the BE and LG model for buried position was 0.26 but only 0.13 for the exposed positions ($P < 1E–20$; Student's *t*-test). To further understand the effect of solvent accessibility on model fit, we classified positions according to their solvent accessibility to six bins: 0%, 0–3% and 3–10%, 10–24%, 24–44% and 44–100%. The first three bins are analyzed using the 'B' model, while the remaining bins are analyzed using the 'E' model. The average log-likelihood gains for each bin were 0.5, 0.25, 0.0001, –0.13, 0.16 and 0.36, respectively ($P < 1E–20$; ANOVA). These results show that relative to the BE model, LG poorly captures the amino acid replacement patterns in extremely 'Buried' or extremely 'Exposed' positions.

4 Discussion

A variety of software for ASR exist (Lartillot *et al.*, 2009; Tamura *et al.*, 2013; Yang *et al.*, 1995). The ever-growing sequence data and the interest in accurate ASR algorithms pose new challenges for such ASR tools. For example, indels were initially treated as unknown characters, which led to ancestral sequences that are longer than all extant sequences. In FastML, we thus reconstruct indel presence/absence in each node prior to sequence reconstruction (Ashkenazy *et al.*, 2012). In this work, we aimed to further improve ASR methodologies by allowing the sequences to evolve according to multiple replacement matrices. Specifically we have shown that fitting a replacement matrix to each position based on structural information can be highly beneficial for ASR.

In a recent study that benchmarked various ASR methodologies, FastML achieved one of the best scores (Randall *et al.*, 2016). Unfortunately, the experiment resulted in relatively easy to reconstruct sequences, and differences among ASR methodologies were minimal. Nevertheless, that research motivates the development of improved experimental benchmarks for ASR.

As we show here, using an array of replacement matrices can be beneficial when taking structural data into account. Such an approach should also be advantageous for studying proteins that contain both trans-membranal and cytosolic domains or to analyze separately different secondary structures. Protein engineering also uses ASR to generate proteins that are more stable than extant proteins and to increase the substrate range of engineered proteins. It is expected that the approach suggested here in which structural information is integrated into the ASR computations should lead to improved engineered proteins.

Our approach to integrate structural information implicitly assumes that both buried, and solvent-exposed positions remain so along the entire course of evolution. In cases where the phylogenetic tree is large and contains dramatic structural changes this assumption might be violated. In addition, the extent to which the protein structure is accounted for in this work is limited to buried and exposed information. More sophisticated models that integrate site-specific structural attributes with amino acid replacement propensities are expected to provide even more accurate estimates of ancestral sequences (see Chi *et al.*, 2018 and references therein).

Acknowledgement

A.M. is a fellow of the Edmond J. Safra Center for Bioinformatics at Tel Aviv University.

Funding

This work was supported by an Israel Science Foundation grant 802/16 to T.P.

Conflict of Interest: none declared.

References

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

Adachi,J. *et al.* (2000) Plastid genome phylogeny and a model of amino acid substitution for proteins encoded by chloroplast DNA. *J. Mol. Evol.*, **50**, 348–358.

Adamczak,R. *et al.* (2004) Accurate prediction of solvent accessibility using neural networks–based regression. *Prot. Struct. Funct. Bioinform.*, **56**, 753–767.

Ashkenazy,H. *et al.* (2012) FastML: a web server for probabilistic reconstruction of ancestral sequences. *Nucleic Acids Res.*, **40**, W580–W584.

Chi,P.B. *et al.* (2018) A new parameter-rich structure-aware mechanistic model for amino acid substitution during evolution. *Prot. Struct. Funct. Bioinform.*, **86**, 218–228.

Dayhoff,M.O. *et al.* (1978) A model of evolutionary change in proteins. In: Dayhoff,M.O. (ed.) *Atlas of Protein Sequence and Structure*. National Biomedical Research Foundation, Washington, DC, Vol. 5, Suppl. 2, pp. 345–352.

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

Fitch,W.M. and Margoliash,E. (1967) A method for estimating the number of invariant amino acid coding positions in a gene using cytochrome c as a model case. *Biochem. Genet.*, **1**, 65–71.

Friedman,N. *et al.* (2002) A structural EM algorithm for phylogenetic inference. *J. Comput. Biol. J. Comput. Mol. Cell Biol.*, **9**, 331–353.

Goldman,N. *et al.* (1998) Assessing the impact of secondary structure and solvent accessibility on protein evolution. *Genetics*, **149**, 445–458.

Guindon,S. *et al.* (2010) New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Syst. Biol.*, **59**, 307–321.

Gumulya,Y. and Gillam,E.M.J. (2017) Exploring the past and the future of protein evolution with ancestral sequence reconstruction: the 'retro' approach to protein engineering. *Biochem. J.*, **474**, 1–19.

Jones,D.T. *et al.* (1992) The rapid generation of mutation data matrices from protein sequences. *Bioinformatics*, **8**, 275–282.

Jukes,T.H. and Cantor,C.R. (1969) Evolution of protein molecules. *Mammalian Prot. Metab.*, **3**, 132.

Juritz,E. *et al.* (2013) Protein conformational diversity modulates sequence divergence. *Mol. Biol. Evol.*, **30**, 79–87.

Kabsch,W. and Sander,C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**, 2577.

Koshi,J.M. and Goldstein,R.A. (1995) Context-dependent optimal substitution matrices. *Prot. Eng. Des. Sel.*, **8**, 641–645.

Lartillot,N. *et al.* (2009) PhyloBayes 3: a Bayesian software package for phylogenetic reconstruction and molecular dating. *Bioinformatics*, **25**, 2286–2288.

Le,S.Q. and Gascuel,O. (2008) An improved general amino acid replacement matrix. *Mol. Biol. Evol.*, **25**, 1307–1320.

Le,S.Q. and Gascuel,O. (2010) Accounting for solvent accessibility and secondary structure in protein phylogenetics is clearly beneficial. *Syst. Biol.*, **59**, 277–287.

Liberles,D.A. (ed.) (2007) *Ancestral Sequence Reconstruction*. Oxford University Press, Oxford, UK.

Ogawa,T. and Shirai,T. (2014) Tracing ancestral specificity of lectins: ancestral sequence reconstruction method as a new approach in protein engineering. *Methods Mol. Biol.*, **1200**, 539–551.

Pupko,T. *et al.* (2008) Probabilistic models and their impact on the accuracy of reconstructed ancestral protein sequences. *Ances. Seq. Reconst.*, **4**, 43–57.

Pupko,T. *et al.* (2002) A branch-and-bound algorithm for the inference of ancestral amino-acid sequences when the replacement rate varies among sites: application to the evolution of five gene families. *Bioinformatics*, **18**, 1116–1123.

Pupko,T. *et al.* (2000) A fast algorithm for joint reconstruction of ancestral amino acid sequences. *Mol. Biol. Evol.*, **17**, 890–896.

Randall,R.N. *et al.* (2016) An experimental phylogeny to benchmark ancestral sequence reconstruction. *Nat. Commun.*, **7**, 12847.

Soyer,O.S. *et al.* (2003) Dimerization in aminergic G-protein-coupled receptors: application of a hidden-site class model of evolution. *Biochemistry*, **42**, 14522–14531.

Tamura,K. *et al.* (2013) MEGA6: molecular evolutionary genetics analysis version 6.0. *Mol. Biol. Evol.*, **30**, 2725–2729.

Tien,M.Z. *et al.* (2013) Maximum allowed solvent accessibilites of residues in proteins. *PLoS One*, **8**, e80635.

Touw,W.G. *et al.* (2015) A series of PDB-related databanks for everyday needs. *Nucleic Acids Res.*, **43**, D364–D368.

Uzzell,T. and Corbin,K.W. (1971) Fitting discrete probability distributions to evolutionary events. *Science*, **172**, 1089–1096.

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

Yang,Z. (1996) Among-site rate variation and its impact on phylogenetic analyses. *Trends Ecol. Evol.*, **11**, 367–372.

Yang,Z. *et al.* (1995) A new method of inference of ancestral nucleotide and amino acid sequences. *Genetics*, **141**, 1641–1650.

Zaucha,J. and Heddle,J.G. (2017) Resurrecting the dead (molecules). *Comput. Struct. Biotechnol. J.*, **15**, 351–358.