Title
A phylogenetic approach for weighting genetic sequences

Journal
BMC Bioinformatics, 22(1)

ISSN
1471-2105

Authors
De Maio, Nicola
Alekseyenko, Alexander V
Coleman-Smith, William J
et al.

Publication Date
2021-12-01

DOI
10.1186/s12859-021-04183-8

Peer reviewed

BMC Bioinformatics

RESEARCH

Open Access

A phylogenetic approach for weighting genetic sequences

Nicola De Maio[1*], Alexander V. Alekseyenko[1,2], William J. Coleman-Smith[1], Fabio Pardi[1,3], Marc A. Suchard[4], Asif U. Tamuri[1,5], Jakub Truszkowski[1,6] and Nick Goldman[1]

*Correspondence: demaio@ebi.ac.uk
[1] European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Genome Campus, Hinxton, UK
Full list of author information is available at the end of the article

Abstract

Background: Many important applications in bioinformatics, including sequence alignment and protein family profiling, employ sequence weighting schemes to mitigate the effects of non-independence of homologous sequences and under- or over-representation of certain taxa in a dataset. These schemes aim to assign high weights to sequences that are 'novel' compared to the others in the same dataset, and low weights to sequences that are over-represented.

Results: We formalise this principle by rigorously defining the evolutionary 'novelty' of a sequence within an alignment. This results in new sequence weights that we call 'phylogenetic novelty scores'. These scores have various desirable properties, and we showcase their use by considering, as an example application, the inference of character frequencies at an alignment column—important, for example, in protein family profiling. We give computationally efficient algorithms for calculating our scores and, using simulations, show that they are versatile and can improve the accuracy of character frequency estimation compared to existing sequence weighting schemes.

Conclusions: Our phylogenetic novelty scores can be useful when an evolutionarily meaningful system for adjusting for uneven taxon sampling is desired. They have numerous possible applications, including estimation of evolutionary conservation scores and sequence logos, identification of targets in conservation biology, and improving and measuring sequence alignment accuracy.

Keywords: Phylogenetics, Sequence weights, Alignment, Protein profile, Conservation scores

Background

Some of the most popular applications in bioinformatics, including multiple sequence alignment [1], sequence database search [2] and protein family profiling [3, 4], employ sequence weighting schemes as a way to mitigate the effects of non-independence of homologous sequences. For example, a database may contain many closely related sequences from one species (like humans) and its close relatives, while other more distantly related species might be under-represented. To address possible problems associated with these biases, several sequence weighting schemes have been proposed over the

years. These methods assign a score to each sequence considered, with the aim of assigning reduced weights to sequences from over-represented clades and larger weights to sequences from under-represented clades. The purpose of sequence weighting schemes is therefore to improve the accuracy of many bioinformatic tasks in a computationally efficient way.

PSI-BLAST [2], for example, employs the Henikoff and Henikoff [5] weighting scheme ("HH94" [5]) where the score of a sequence is the average of the scores of each position of the sequence, the score of a position being $1/rd$, with r the number of different characters at the considered alignment column and d the number of times the character of the considered sequence and position appears in the considered alignment column. The idea of this weighting scheme is to give equal weight to all characters observed at one alignment column, dividing this weight equally among those sequences sharing that character at that position. This method has the advantage of being very fast to calculate, and of giving higher weights to sequences with more rare characters that are, therefore, likely more distantly related.

HMMER [6] and the CLUSTAL family of aligners [1, 7, 8] use the weighting scheme of Gerstein et al. (1994: "GSC94" [9]; similar to [10]), which defines sequence weights iteratively along a phylogeny from tips to root. At each step, the length of the considered tree branch is split proportionally to the current weights of its descendant sequences, and is then added to the weights of the descendant sequences. Here, the idea is that weights are determined by divergence between groups of sequences. The more diverged one group of sequences is from the others, the higher weights it will have. However, the weight of a group is shared among the sequences in the group, so that in a group with many similar sequences each of those sequences will have small individual weight.

Other sequence weighting schemes have also been proposed, although they have seen fewer applications. Maximum discrimination sequence weighting [11] is a complex approach that aims to optimally distinguish homology from chance alignments in database searches. Henikoff and Henikoff [12] proposed a method that splits sequences into clusters based on sequence similarity, and assigns equal weights to sequences in the same cluster and a total weight of 1 to each cluster [12]. Vingron and Argos weighted sequences proportionally to their average distances from all other sequences [13]; Sibbald and Argos proposed an approach in which a sequence receives more weight if it is more isolated in sequence space [14]. Altschul et al. measured evolutionary correlations among sequences using branch lengths in the phylogeny, and then calculated sequence weights using the inverse of the variance-covariance matrix [15]; Gotoh developed a fast approximation of this method [16]. Similar ideas have also been explored within methods aimed at estimating character frequencies at a given position in a protein [17–19], defining tree or alignment informativeness [20–24], and quantifying diversity within a habitat and prioritising conservation efforts [25–31].

The many weighting schemes proposed have rarely been assessed and compared under different scenarios. We show that heuristic approaches can suffer from limitations: for example GSC94, while providing good performance on ultrametric trees, can lead to inaccurate results on non-ultrametric trees. Instead, here we propose a new weighting scheme, the first to be derived from the idea of evolutionary 'novelty'. We quantify the novelty of each sequence compared to the other sequences under consideration by

computing the probability that, at a given position, sequences are 'phylogenetically identical by descent' (PIBD): that is, that they descended from a common ancestor without any substitution occurring. Our terminology highlights the similarity with the concept of "identity by descent" in population genetics (see e.g. [32]), where it applies when two alleles are not only identical ("identity by state"), but also have had no mutation or recombination occurring in the lineages connecting them to their most recent common ancestor. This rigorous approach allows us to quantify the novelty of sequences in very general scenarios (without specific assumptions regarding the phylogeny relating the considered sequences) while being robust to uneven sampling and very elevated or reduced divergence levels, and generally conforming to guiding principles for an acceptable weighting scheme [33]. We present algorithms and scripts to efficiently compute these weights from a phylogeny and from a multiple sequence alignment.

As shown by the examples above, this new weighting scheme has a number of possible applications, from gene family profiles and multiple sequence alignment evaluation to ecology and conservation biology. The aim of our work is to provide a new sequence weighting scheme that is robust to the choice of application and scenario, therefore giving the possibility of improving the accuracy of these bioinformatics tasks, while at the same time being sufficiently computationally efficient to be used on large datasets. As an example, we focus on the task of inferring character frequencies at an alignment column. Inference of character frequencies is not only important for gene family profiling, but also for modeling evolutionary fitness, calculating conservation scores, and visualizing sequence logos [34–38]. We show that our methods result in efficient and accurate inference of character frequencies, with clear advantages compared to previous sequence weighting schemes.

Methods

Phylogenetic novelty scores

We consider a phylogenetic tree ϕ describing the evolutionary relationships of its N tips $s_1, \ldots, s_N$. We want to define weights w_s representing how 'novel' tip s is compared to the other tips of ϕ. Throughout this paper we consider the tips to represent biomolecular sequences comprising amino acid or nucleotide characters, but other possible sets of characters could equally be accommodated. We assume that we have one sequence associated with each tip, conveniently sharing the same names $s_1, \ldots, s_N$, and arranged as the rows of an alignment A. We start by defining weights that are a function of ϕ only, and so depend on the evolutionary history relating the considered sequences and not on the specific sequences themselves. In the next section we extend the definitions to also condition on the observed sequence characters.

As a motivating example, if ϕ consists of only extremely long branches, then we want $w_1 = \cdots = w_N = 1$. In fact, in this case, all sequences represent effectively independent observations, so no weighting correction is needed. This means that, unlike many sequence weighting schemes (e.g. [9, 14, 39, 40]), we want to account for the effect of saturation, so that doubling the length of a long tree branch has negligible effect on the weights.

If instead ϕ has branches all of length 0, we want $w_1 = \cdots = w_N = 1/N$, so that the total alignment score is 1, as in [9, 14, 39, 40]. This is because all the observed sequences

are now just perfectly dependent copies, and so in total they represent just one independent observation of a sequence. At an intermediate level, if ϕ has two tips ($N = 2$), and branch length such that half of the ancestral characters are expected not to have mutated in either branch (they are PIBD with probability 0.5), then we want the total alignment score to be 1.5, and both weights to be 0.75; this is because in this case only half of each sequence will be novel with respect to the other, so in total we observe 1.5 novel sequences, and we want the two sequences to have the same weight.

A simple way to describe how novel s_1 is with respect to s_2 could be to count the number of mismatches between their sequences. However, even if s_1 and s_2 were very divergent from each other, their sequences would still be identical at some alignment column because of chance or of convergent evolution, instead of close relatedness. In our approach, s_1 can be novel with respect to s_2 at a column of A even if they share the same character, as long as they are not PIBD.

We usually cannot know for sure if sequences are PIBD and at which alignment column, so we define $p_s(i)$ as the probability that, at a generic alignment column, the number of tips of ϕ (including s) that are PIBD to s, $i_\phi(s)$, is exactly i. For example, $p_s(N)$ is the probability that, at a generic alignment column, no substitution occurs along ϕ; $p_s(1)$ is the probability that no tip (except s) in ϕ is PIBD to s at some arbitrary alignment column. We then define the weight w_s of s within ϕ as:

$$w_s = \sum_{i=1}^{N} \frac{p_s(i)}{i} = \mathbb{E}_\phi \left[\frac{1}{i_\phi(s)} \right] . \tag{1}$$

In the simplest case of nucleotide sequences evolving under the JC69 substitution model ([41]; all substitution rates are 1/3), the probability that two nodes in ϕ separated by branch length t are PIBD is e^{-t}. So, again in the simple case that $N = 2$ and that the two branches in ϕ have each length $t/2$, s_1 and s_2 each have weight $w_{s_1} = w_{s_2} = p_{s_1}(1) + p_{s_1}(2)/2 = (1 - e^{-t}) + e^{-t}/2 = 1 - e^{-t}/2$, and the sum of the weights is $2 - e^{-t}$. The same is true for any pair of branch lengths with sum t, of course.

We expect the definition of sequence weights given by Eq. 1 to be useful for character frequency inference and many other applications. In fact, in addition to satisfying classical sequence weighting requirements [33], these w_s can also be efficiently calculated from any ϕ and substitution model, as discussed later. We refer to weights w_s as the 'phylogenetic novelty scores' (PNS). We call the sum of all weights in ϕ the 'effective sequence number' (ESN): $T = \sum_{s=1}^{N} w_s$, representing the expected number of evolutionarily distinct character observations at an alignment column. An example graphical representation of the PNS is shown in Fig. 1

Conditioning on observed data

In this section we define weights that are a function not only of phylogeny ϕ, but also of a specific alignment column D of alignment A. These weights refer not to the novelty of a sequence s, but of its specific character D_s observed in row s of column D. The probability that two tips of ϕ are PIBD at a specific alignment column can be strongly affected by the observed characters at that column. Clearly, if the two tips differ at alignment column D then the probability that they are PIBD, conditional on D, is 0. The case that the two tips

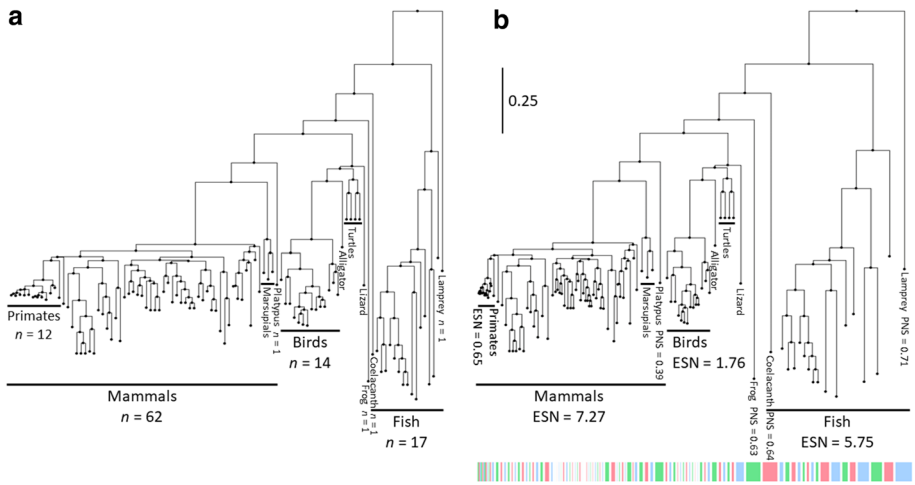


Fig. 1 Example of PNS for a 100-vertebrates tree. Here we show graphically the values of the phylogenetic novelty scores w_s from Eq. 1 for the tips of a tree of 100-vertebrate species. The tree is taken from the UCSC genome browser 100-way alignment of vertebrates to the human genome, downloaded from http://hgdownload.cse.ucsc.edu/goldenPath/hg38/multiz100way/hg38.100way.commonNames.nh. The scale bar indicates 0.25 expected substitutions per site. This tree was also used for simulations in this work. **a** The tree has all tips spaced uniformly on the horizontal axis, representing the case of no weighting scheme being used. **b** Tips are spaced horizontally according to their w_s weight. The weight of each tip can also be seen in the length of the colored bars. Notice how species in regions of the tree with many close relatives (e.g. mammal, primate and bird clades) have low PNSs, and so take up less space individually. This means the horizontal dimension of the plot now gives more equal representation of the novelty of each sequence and clade, instead of emphasising densely sampled clades. More divergent species with few close relatives (e.g. lamprey, coelacanth, frog and platypus) have higher PNSs and are given more horizontal space, representing the greater novelty of their sequences relative to other species in the tree. Cumulative ESN scores (clade-wise sum of PNSs) are also shown for some clades

have the same character in D is less trivial. If we assume that the two tips are separated by a total divergence time t, and for simplicity assuming a JC69 substitution model [41], then the probability that the two tips have the same nucleotide is $(1 + 3e^{-4t/3})/4$ and the probability that the two tips are PIBD is e^{-t}; therefore, the probability that the two tips are PIBD conditional on them having the same nucleotide is $4e^{-t}/(1 + 3e^{-4t/3})$.

We denote by $p_s(i|D)$ the probability that exactly i sequences are PIBD to s at the given, observed alignment column D. The new positional PNSs conditional on D are then defined as:

$$w_s^D = \sum_{i=1}^{N} \frac{p_s(i|D)}{i} \,. \tag{2}$$

Algorithms for calculating the phylogenetic novelty scores

We present several algorithms for calculating PNS. One of these methods ('up-down pruning') is the most computationally efficient, and so is described below, but may also be the most difficult to understand. For this reason, we also mention other approaches and include their full description in the Supplement. In the following we assume that the phylogeny ϕ is rooted; the case of an unrooted topology follows simply by placing an arbitrary root on the tree, as long as the substitution process is reversible and at

equilibrium (as in this case the scores are not affected by the position of the root). In the case of non-reversible or non-stationary character evolution, the position of the root can affect the scores, and so a rooted phylogeny (which could in principle be estimated from sequences in this scenario) is required.

Calculating PNS scores via simulation

We can calculate PNS by simulating sequence evolution along ϕ. If we are interested in weights w_s, at each iteration we start by sampling a root character from the equilibrium distribution. We then sample its descendant characters and the mutation events along the branches of ϕ using standard methods (e.g. [42], "method 2" of [43, 44]) until we reach all the tips, recording each substitution that occurs and hence which tip characters are PIBD. Note that it is not possible to achieve this using software such as *evolver* [45] or "method 1" of *INDELible* [43] that only simulate the start and end state of each branch, and do not distinguish between characters that are PIBD and those that happen to match following multiple substitutions. For each iteration we associate a score of $1/i$ to a tip of ϕ if its observed character is PIBD to the characters of exactly i tips. The final weight of a tip is then obtained by averaging its scores over all iterations. Weights w_s^D can be similarly calculated employing a variant of the up-down approach [46] to sample characters at internal nodes of the phylogeny conditional on D. A straightforward but inefficient way to achieve the same result is to simulate characters without any conditioning, and discard those iterations that do not match D. These approaches are described in more detail in the Supplement.

Calculation of PNS scores via brute-force

We can calculate PNS via brute-force, that is, by enumerating all possible mutational histories on ϕ by considering all possible character assignments at each end of each branch, and for each branch considering whether there is at least one substitution on it or not. Each mutational history results in a score as in the previous method using simulation. By averaging the scores of all mutational histories, while accounting for different probabilities of different histories, we can calculate the w_s or w_s^D weights. The full methods are described in detail in the Supplement.

Pruning method to calculate the ESN

The ESNs $T = \sum_{s=1}^{N} w_s$ or $T^D = \sum_{s=1}^{N} w_s^D$ can be calculated very efficiently (computational cost $\mathcal{O}(N)$) if one is not interested in the weights of the individual tips. The idea is to calculate T iteratively on each subtree of ϕ starting from the tips until we reach the root. We call this the 'pruning ESN' method, due to its similarity with Felsenstein's pruning algorithm [47]. See the Supplement for details.

Up–down pruning approach to calculate PNS

We now present the main, efficient algorithm that we use and recommend for calculating PNS. It can be used for calculating either w_s or w_s^D weights and can be considered an adaptation of Felsenstein's pruning algorithm [47]. The method visits all nodes in ϕ starting from the tips and toward the root ('up' phase) and then again a second time starting from the

root and moving downward to the tips ('down' phase), similar to the up-down approach of [46]. The computational cost of this algorithm is cubic in the number of tips N.

In the following we assume that the substitution rate matrix Q is given. We make no assumptions regarding the state space of the substitution process, which can comprise nucleotide, amino acid or codon states [48], and exclude or include gaps (see e.g. [49]). The probability of having character k at the end of a branch of length t, conditional on having character j at its start, is then $P_t^{j,k}$, the entry in row j and column k of $P_t = \exp(tQ)$. See [50] for a more detailed introduction to these concepts in molecular phylogenetics. Starting with character j at the top node of a branch of length t, we denote the probability that no substitution occurs along the branch, and therefore also that the top and bottom nodes of the branch are PIBD, as:

$$I_t^j = \exp(tQ_{jj}) \ . \tag{3}$$

Note that I_t^j is different from $P_t^{j,j}$, which is the probability that the character a the end of the branch is the same as j, the character at the start of the branch. This is because I_t^j requires not only the two characters to be the same, but also that no substitution occurred on the branch; when substitutions occurred but resulted cumulatively in no change in character at the two ends of a branch (as possible on long branches) we do not consider those two states PIBD, and the two characters are treated as independent observations.

Our objective is to calculate, for each tip s of ϕ, the probability distribution $(p_s(0) = 0, p_s(1), \ldots, p_s(N))$ of having each possible number of PIBD sequences (defined as in Eqs. 1 and 2). To address both the cases of w_s^D and w_s, we present our description as conditioned on data D; for the case that one is interested in w_s, the same equations can be used but setting D as non-informative. (For example, with DNA sequences a non-informative column D will have all entries equal to character "N", representing an unknown nucleotide, so that the partial likelihood for column D at each tip is 1.) As before, we denote the observed character at tip s by D_s; we now represent the observed characters for the leaves in sub-phylogeny ϕ' of ϕ as $D_{\phi'}$. In the particular case that $\phi' = \phi$, we have $D_\phi = D$, so we can represent the final values of interest for tip s also as $p_s(i|D_\phi)$.

For most of the following, we condition probabilities on information from only part of ϕ. Given a node ν of ϕ, and given a sub-phylogeny ϕ' of ϕ, we define $p_\nu^{\phi'}(i)$ to be the probability that there are exactly i tips in ϕ' that are PIBD to ν. We also define $p_\nu^{\phi'}(i,j)$ as the probability of having i tips in ϕ' that are PIBD to ν and to have character j in ν. Similarly, we define $p_\nu^{\phi'}(i|j)$ to be the probability of having i tips in ϕ' that are PIBD to ν, conditional on having character j in ν. Finally, $p_\nu^{\phi'}(i, D_{\phi'}|j)$ is the probability that i tips in ϕ' are PIBD to ν and that the observed data in ϕ' is $D_{\phi'}$, conditional on having character j in ν.

The first step of the up phase is to initialise $p_s^s(i, D_s|j)$ at every tip s of ϕ, for every character j, and for $0 \leqslant i \leqslant N$:

$$p_s^s(i, D_s|j) = \delta(j, D_s)\,\delta(i, 1) \tag{4}$$

where $\delta(x, y)$ is the Kronecker delta function ($\delta(x, y) = 1$ if and only if $x = y$; $\delta(x, y) = 0$ otherwise). In the case that D_s is uninformative, we have $p_s^s(i, D_s|j) = \delta(i, 1)$.

Next, starting from the tips, we move iteratively 'upward', toward the root of ϕ. If branch b with length t connects the two nodes ν_1 (the parent or upper node) and ν_2 (child or lower node), then b splits ϕ into two sub-phylogenies. We call these ϕ_1 and ϕ_2, with ϕ_2 the sub-phylogeny of ϕ containing ν_2 (but not b) and all its descendant nodes and branches, and ϕ_1 the sub-phylogeny of ϕ containing all nodes and branches (except b) not in ϕ_2. Assuming that we have already visited all branches and nodes below b, and therefore know $p_{\nu_2}^{\phi_2}(i, D_{\phi_2}|j)$ for every character j and every $0 \leqslant i \leqslant N$, we can then calculate $p_{\nu_1}^{\phi_2}(i, D_{\phi_2}|j)$ for every character j and every $0 \leqslant i \leqslant N$:

$$\begin{aligned}
p_{\nu_1}^{\phi_2}(0, D_{\phi_2}|j) &= I_t^j p_{\nu_2}^{\phi_2}(0, D_{\phi_2}|j) + \sum_k (P_t^{j,k} - \delta(j,k) I_t^j) \sum_{i=0}^{N} p_{\nu_2}^{\phi_2}(i, D_{\phi_2}|k) \\
p_{\nu_1}^{\phi_2}(1, D_{\phi_2}|j) &= I_t^j p_{\nu_2}^{\phi_2}(1, D_{\phi_2}|j) \\
&\vdots \\
p_{\nu_1}^{\phi_2}(N, D_{\phi_2}|j) &= I_t^j p_{\nu_2}^{\phi_2}(N, D_{\phi_2}|j) \ .
\end{aligned} \tag{5}$$

For the first term $p_{\nu_1}^{\phi_2}(0, D_{\phi_2}|j)$, the first summand $I_t^j p_{\nu_2}^{\phi_2}(0, D_{\phi_2}|j)$ relates to the case in which there are no mutations on the considered branch b, while the second summand relates to the case in which at least one mutation event happens on b. Graphical examples for Eqs. 4 and 5 are given in Additional file 1: Fig. S1. Many of the $p_{\nu_1}^{\phi_2}(i, D_{\phi_2}|j)$ will be 0 (when i is larger than the number of tips in ϕ_2). In practice, we have made use of this to speed up the implementation of the algorithm, but we ignore it here for brevity.

Thanks to Eq. 5 we can 'move' probabilities up along branches, starting from the initialisations at the tips. Next, we show how to 'merge' probabilities when we reach an internal node ν. A given internal node ν splits ϕ into three sub-phylogenies (a parent one, ϕ_P, a left child one ϕ_L, and a right child one ϕ_R), each associated with one of the three branches adjacent to ν (one parent and two child branches). If ν is the root, then for simplicity we consider its parent sub-phylogeny to exist but be empty. Assuming that we have already visited all branches and nodes descendant of ν, and therefore know $p_\nu^{\phi_L}(i, D_{\phi_L}|j)$ and $p_\nu^{\phi_R}(i, D_{\phi_R}|j)$ for every character j and every $0 \leqslant i \leqslant N$, and denoting by $\phi_L \cup \phi_R$ the sub-phylogeny obtained by joining sub-phylogenies ϕ_L and ϕ_R, we can calculate $p_\nu^{\phi_L \cup \phi_R}(i, D_{\phi_L \cup \phi_R}|j)$ for every character j and every $0 \leqslant i \leqslant N$:

$$\begin{aligned}
p_\nu^{\phi_L \cup \phi_R}(0, D_{\phi_L \cup \phi_R}|j) &= p_\nu^{\phi_L}(0, D_{\phi_L}|j) p_\nu^{\phi_R}(0, D_{\phi_R}|j) \\
p_\nu^{\phi_L \cup \phi_R}(1, D_{\phi_L \cup \phi_R}|j) &= p_\nu^{\phi_L}(0, D_{\phi_L}|j) p_\nu^{\phi_R}(1, D_{\phi_R}|j) \\
&\quad + p_\nu^{\phi_L}(1, D_{\phi_L}|j) p_\nu^{\phi_R}(0, D_{\phi_R}|j) \\
&\vdots \\
p_\nu^{\phi_L \cup \phi_R}(N, D_{\phi_L \cup \phi_R}|j) &= \sum_{i=0}^{N} p_\nu^{\phi_L}(i, D_{\phi_L}|j) p_\nu^{\phi_R}(N-i, D_{\phi_R}|j) \ .
\end{aligned} \tag{6}$$

Equation 6 is one of the most computationally demanding steps of the algorithm (jointly with Eq. 10 below) as it has up to quadratic cost in N. Equation 6 is used on each internal node of ϕ, and so causes the algorithm to have a total time complexity in the order of $\mathcal{O}(N^3)$.

Using Eqs. 5 and 6 iteratively, we can calculate $p_\nu^{\phi_L}(i, D_{\phi_L}|j)$, $p_\nu^{\phi_R}(i, D_{\phi_R}|j)$ and $p_\nu^{\phi_L \cup \phi_R}(i, D_{\phi_L \cup \phi_R}|j)$ for each internal node ν, each $0 \leqslant i \leqslant N$, and any character j. We stop once we reach node ρ, the root of ϕ. At ρ we have $p_\rho^{\phi}(i, D_\phi|j) = p_\rho^{\phi_L \cup \phi_R}(i, D_{\phi_L \cup \phi_R}|j)$ for any character j and $0 \leqslant i \leqslant N$. If π are the character frequencies at ρ, we then have the joint probabilities:

$$p_\rho^{\phi}(i, D_\phi, j) = \pi(j) p_\rho^{\phi}(i, D_\phi|j) \ . \tag{7}$$

This concludes the 'up' stage of the method, which is more succinctly described in Eq. 8:

Algorithm stage Up

[initialise]	compute P_t and I_t^j for every branch length t and character j compute $p_s^s(i, D_s \mid j)$ for every tip s, character j and $0 \leqslant i \leqslant N$
[iterate]	visit every internal node ν in post-order traversal; for each ν, j, i calculate $p_\nu^{\phi_L}(i, D_{\phi_L} \mid j)$ and $p_\nu^{\phi_R}(i, D_{\phi_R} \mid j)$ with Eq. 5 $p_\nu^{\phi_L \cup \phi_R}(i, D_{\phi_L \cup \phi_R} \mid j)$ with Eq. 6
[finalise]	at root ρ calculate $p_\rho^{\phi}(i, D_\phi, j)$ for every j, i using Eq. 7

(8)

The 'down' phase is the second and last stage of the algorithm. Starting from root ρ, we move toward the tips, visiting each node and branch in pre-order traversal. Given branch b of length t connecting nodes ν_1 (parent) and ν_2 (child), we assume, as in Eq. 5, that ϕ_1 and ϕ_2 are the two sub-phylogenies induced by b. Assuming that we have already visited iteratively all ancestor branches of b, and therefore know $p_{\nu_1}^{\phi_1}(i, D_{\phi_1}, j)$ for every character j and $0 \leqslant i \leqslant N$, we can calculate $p_{\nu_2}^{\phi_1}(i, D_{\phi_1}, j)$ for every character j and $0 \leqslant i \leqslant N$:

$$\begin{aligned}
p_{\nu_2}^{\phi_1}(0, D_{\phi_1}, j) &= (P_t^{j,j} - I_t^j) \sum_{i=0}^{N} p_{\nu_1}^{\phi_1}(i, D_{\phi_1}, j) + I_t^j p_{\nu_1}^{\phi_1}(0, D_{\phi_1}, j) \\
&\quad + \sum_{k \neq j} P_t^{k,j} \sum_{i=0}^{N} p_{\nu_1}^{\phi_1}(i, D_{\phi_1}, k) \\
p_{\nu_2}^{\phi_1}(1, D_{\phi_1}, j) &= I_t^j p_{\nu_1}^{\phi_1}(1, D_{\phi_1}, j) \\
&\vdots \\
p_{\nu_2}^{\phi_1}(N, D_{\phi_1}, j) &= I_t^j p_{\nu_1}^{\phi_1}(N, D_{\phi_1}, j) \ .
\end{aligned} \tag{9}$$

Equation 9 allows us to 'move' probabilities downward along branches, starting from the root. Next, we show again how to 'merge' probabilities when we reach an internal node ν. Given one left child sub-phylogeny ϕ_L of ν, and given its parent sub-phylogeny ϕ_P, we can calculate $p_\nu^{\phi_P \cup \phi_L}(i, D_{\phi_P \cup \phi_L}, j)$ for every character j and $0 \leqslant i \leqslant N$:

$$\begin{aligned}
p_\nu^{\phi_P \cup \phi_L}(0, D_{\phi_P \cup \phi_L}, j) &= p_\nu^{\phi_L}(0, D_{\phi_L}|j) p_\nu^{\phi_P}(0, D_{\phi_P}, j) \\
p_\nu^{\phi_P \cup \phi_L}(1, D_{\phi_P \cup \phi_L}, j) &= p_\nu^{\phi_L}(0, D_{\phi_L}|j) p_\nu^{\phi_P}(1, D_{\phi_P}, j) \\
&\quad + p_\nu^{\phi_L}(1, D_{\phi_L}|j) p_\nu^{\phi_P}(0, D_{\phi_P}, j) \\
&\vdots \\
p_\nu^{\phi_P \cup \phi_L}(N, D_{\phi_P \cup \phi_L}, j) &= \sum_{i=0}^{N} p_\nu^{\phi_L}(i, D_{\phi_L}|j) p_\nu^{\phi_P}(N - i, D_{\phi_P}, j) \ .
\end{aligned} \tag{10}$$

We use Eq. 10 twice for each internal node v, once with the left child sub-phylogeny ϕ_L and once replacing ϕ_L with the right child sub-phylogeny ϕ_R. Using Eqs. 9 and 10 iteratively, we calculate $p_v^{\phi_P}(i, D_{\phi_P}, j)$, $p_v^{\phi_P \cup \phi_L}(i, D_{\phi_P \cup \phi_L}, j)$ and $p_v^{\phi_P \cup \phi_R}(i, D_{\phi_P \cup \phi_R}, j)$ for each internal node v, each $0 \leqslant i \leqslant N$ and every character j. After we have visited every internal node of ϕ, we reach all tips s using Eq. 9 to obtain $p_s^{\phi \setminus s}(i, D_{\phi \setminus s}, j)$ for all characters j and all $0 \leqslant i \leqslant N$, where $\phi \setminus s$ is the sub-phylogeny obtained by removing s (and itsparent branch) from ϕ. We then combine these probabilities at the tips with the initialisation probabilities $p_s^s(i, D_s|j)$ to obtain, at every tip s, for all characters j and each $1 \leqslant i \leqslant N$:

$$p_s^{\phi}(i, D_\phi, j) = p_s^{\phi \setminus s}(i-1, D_{\phi \setminus s}, j) p_s^s(1, D_s|j)\,. \tag{11}$$

The final probabilities of interest, $p_s^{\phi}(i|D_\phi)$, can be calculated for every $0 \leqslant i \leqslant N$ and every tip s as:

$$p_s^{\phi}(i|D_\phi) = \frac{\sum_j p_s^{\phi}(i, D_\phi, j)}{P(D_\phi)}\,, \tag{12}$$

where $P(D_\phi)$ is the probability of the data (the phylogenetic likelihood of ϕ and the substitution model for D). In the case D is empty, that is, if we want to calculate weights w_s, then $P(D_\phi) = 1$. Otherwise, if we are interested in weights w_s^D, then $P(D_\phi)$ can be calculated as a normalisation factor such that the probabilities $p_s^{\phi}(i|D_\phi)$ at any s sum over i to 1. In either case, the final PNS scores can be easily calculated substituting the results from Eq. 12 into Eqs. 1 or 2.

We summarise the 'down' stage of the algorithm in Eq. 13:

Algorithm stage Down

[initialise]	run algorithm stage Up to calculate $p_\rho^{\phi}(i, D_\phi, j)$ for root ρ, and $p_v^{\phi_L}(i, D_{\phi_L} \mid j)$ and $p_v^{\phi_R}(i, D_{\phi_R} \mid j)$ at every internal node v and every j, i
[iterate]	visit every internal node v in pre-order traversal; for each v, j, i calculate $p_v^{\phi_1}(i, D_{\phi_1} \mid j)$ with Eq. 9 $p_v^{\phi_P \cup \phi_L}(i, D_{\phi_P \cup \phi_L} \mid j)$ and $p_v^{\phi_P \cup \phi_R}(i, D_{\phi_P \cup \phi_R} \mid j)$ with Eq. 10
[finalise]	at each tip s calculate $p_s^{\phi}(i, D_\phi, j)$ for every j, i using Eq. 12 at each tip s calculate $p_s^{\phi}(i \mid D_\phi)$ for every i using Eq. 7

(13)

Fast approximation

The most efficient algorithm above has cubic cost in N. In some circumstances, for example when $N > 10^5$, it becomes important to consider faster solutions. For this reason, we also present an approximate PNS that can be calculated more efficiently. With $i_\phi(s)$ the random variable representing the number of tips in ϕ that are PIBD to s, we have $w_s = \sum_{i=1}^N p_s(i)/i = \mathbb{E}[1/i_\phi(s)]$ (Eq. 1). As an alternative fast approximation we consider:

$$\overline{w}_s = \frac{1}{\mathbb{E}[i_\phi(s)]} = \frac{1}{\sum_{i=1}^N i p_s(i)}\,. \tag{14}$$

The weights $\overline{w}_s$ can be computed very efficiently with an up-down pruning approach, requiring only $\mathcal{O}(N)$ time, so we refer to them as 'fast PNS'. The algorithm to calculate weights $\overline{w}_s$ has many similarities to the one in the previous Section, and is described in detail in the Supplement.

Application to inference of character frequencies

Inference of character frequencies specifically for a single alignment column has broad applications such as modeling selection [34, 35], and creating profile HMMs [3, 4, 6] and sequence logos [36–38]. Here, we assume that the frequencies of interest are the equilibrium frequencies at a given alignment column, i.e. the average character frequencies over long evolutionary times. Such frequencies are typically represented in molecular phylogenetics as π, with $\pi(j)$ being the equilibrium distribution of character j [50]. This definition of frequencies fits well with the assumptions of profile HMMs, and is also reasonable for sequence logos, although we acknowledge that different definitions might be also considered in different settings. In this work, we want to investigate and compare different methods for inferring π.

The simplest inference method is to use the observed frequency $p(j)$ of character j within the given column as an estimate of the true frequency $\pi(j)$. This approach corresponds to assuming that all sequences are independent of each other. This approach might be ideal in some circumstances, for example when the considered sequences are not homologous but only evolutionary convergent, but might be inappropriate in others. As an example, consider an alignment of 1000 homologous human sequences and two mouse sequences (1002 homologous sequences in total). Genetic variation within mice, and variation between mice and humans will have negligible effects on estimates $p(j)$, which will be dominated by within-human genetic variation. However, human sequences are highly correlated, as they have very short divergence time between each other, so within-human allele frequencies will typically not represent evolutionary equilibrium character distributions. The problem here is that using $p(j)$ as an estimate of $\pi(j)$ means treating homologous sequences as independent of each other, while they are often strongly correlated due to shared evolutionary histories.

A traditional way to address this problem is to use sequence weights, for example our w_s, to reduce the contribution of groups of closely related sequences. We can in fact define $p^w(j)$, a new estimate of $\pi(j)$, as:

$$p^w(j) = \frac{\sum_s w_s \delta(j, D_s)}{\sum_s w_s} \tag{15}$$

where, as before, D_s is the observed character for sequence s at the alignment column D under consideration, and δ is again the Kronecker delta function.

We investigate and compare the performance, for character frequency inference, of the three weighting schemes introduced above: w_s, w_s^D and $\overline{w}_s$. We also consider two popular sequence weights: those defined by [5], which we call HH94, and by [9], which we call GSC94. HH94 first calculates, for any s, the score of s at an alignment column D, which we will denote HH_s^D. This score HH_s^D is $1/rd$, with r the number of different characters in D and d the number of times character D_s appears in D. The weight for sequence s, which we denote HH_s, is then defined as the average of HH_s^D over all columns D of alignment A.

GSC94 defines sequence weights iteratively along a phylogeny, by visiting branches in post-order traversal (from the tips to the root). First, all terminal branches (those connected to the tips of ϕ) are visited, and the length of a terminal branch connected to tip s is assigned as the initialisation value of the weight GSC_s of s. Then, every time an internal branch b is visited, its length t is distributed among the weights of its descendant sequences. More precisely, first t is split among the tips, with the part t_s assigned to s being $t_s = t\, GSC_s / \sum_{s' \in S_b} GSC_{s'}$, where S_b is the set of tips descendent from b. Secondly, each GSC_s for $s \in S_b$ is increased by t_s. After the last branches connected to the root have been processed, the GSC_s are the final GSC94 weights.

In addition to the character frequency inference $p(j)$ based on the observed frequencies, and $p^w(j)$ based on one of many weighting schemes studied, we also consider character frequency inference via phylogenetic maximum likelihood (ML). We perform this using PhyML v3.1 by fixing the phylogenetic tree (inferred from the whole alignment A using FastTree v2.1.10 [51]) and the substitution model exchangeabilities, and inferring, one alignment column D at a time, only the equilibrium character frequencies π.

Bayesian approaches to character frequency inference

Above, we introduced point estimate methods for character frequency inference. These methods do not measure inference uncertainty, and this can result in a very limited summary of the available data. For example, observing character j 100 times in an alignment column from 100 distantly related species leads all above methods to infer 100% frequency for j; however, so also does observing j two times within an alignment column of just two closely related sequences. While in the first scenario there should be little uncertainty regarding the inferred frequencies, in the second scenario uncertainty should be elevated. Using a Bayesian method is a natural way to address this issue, and also allows the inclusion of priors over characters frequencies. Here we present a Bayesian variant of the weight-based character frequency inference of Eq. 15.

If A is composed of N independent (non-homologous) sequences, the likelihood of a column D is $P(D|\pi) = \prod_s \pi(D_s)$. It is simple to combine this likelihood with a character frequency prior to obtain a Bayesian posterior distribution, and perform Bayesian character frequency inference. However, we are interested in the general case where the sequences in A are related by a phylogeny ϕ, and therefore are not independent. One possible way to perform Bayesian inference of π in this scenario would be using Bayesian phylogenetic methods such as BEAST [52] or MRBAYES [53], but at often excessive computational cost. Instead, we propose an approximation of the likelihood function $P(D|\pi)$ based on sequence weights w_s:

$$\widehat{P}(D|\pi) = \prod_s \pi^{w_s}(D_s) = \prod_j \left(\pi(j)\right)^{\sum_s w_s \delta(j, D_s)} . \tag{16}$$

Similarly, we can replace weights w_s in Eq. 16 with any other weighting scheme. In the following, we assume a uniform prior $P(\pi)$ on character frequencies, meaning that all possible π are similarly likely a *priori*. Alternative priors are possible, and some might be more realistic, but usually at the cost of introducing more parameters in the model. Our approximation of the posterior probability $P(\pi|D)$ is then:

$$\widehat{P}(\pi|D) = \frac{P(\pi)\widehat{P}(D|\pi)}{P(D)} = \frac{\prod_j \left(\pi(j)\right)^{\sum_s w_s \delta(j,D_s)}}{\int_\xi \prod_j \xi_j^{\sum_s w_s \delta(j,D_s)}} \tag{17}$$

where the integral in the denominator is over all possible character frequencies ξ. Equation 17 is a Dirichlet distribution with parameters $\alpha_j = 1 + \sum_s w_s \delta(j, D_s)$, so in the following we use the properties of Dirichlet distributions [54]. The (approximate) maximum *a posteriori* and ML π are both given by the weighted observed character frequencies $p^w(j)$ in Eq. 15. The approximation of the expectation of $\pi(j)$, $E(\pi(j)|D)$ is however:

$$\widehat{E}(\pi(j)|D) = \frac{1 + \sum_s w_s \delta(j, D_s)}{B + \sum_s w_s} = \frac{\alpha_j}{\alpha_0}\ , \tag{18}$$

where B is the number of possible characters in the considered alphabet, $\alpha_j = 1 + \sum_s w_s \delta(j, D_s)$ and $\alpha_0 = B + \sum_s w_s$. This can be seen as the ML estimate of π in the presence of 1 pseudo-count per character. The posterior variance is then approximated as:

$$\mathrm{Var}(\pi(j)|D) \approx \frac{\alpha_j(\alpha_0 - \alpha_j)}{\alpha_0^2(\alpha_0 + 1)}\ , \tag{19}$$

which can be used as a measure of the uncertainty over character frequencies. However, considering that Eq. 17 has beta-distributed univariate marginals $\widehat{P}(\pi(j)|D) \sim \mathrm{Beta}(\alpha_j, \alpha_0 - \alpha_j)$, in the following we derive approximate 95% posterior probability intervals using the stats.beta.ppf function in *scipy* [55].

Simulations

We use simulations to test and compare computational demands of calculating PNS values as well as for assessing the accuracy of different approaches to infer position-specific character frequencies. In the base simulation scenario, we simulate nucleotide sequence evolution along a 100 vertebrate taxa phylogeny (Fig. 1) using Dendropy [56]. We use a HKY85 substitution model [57] with transition:transversion ratio $\kappa = 3$ both for simulation and inference. We simulate 10 replicates, each replicate consisting of an alignment of 1000 columns. Alignment columns are evolved independently of each other (conditional on the tree and the substitution model). As we do not simulate indel events, so we do not consider gap characters in our inference; when used on real data we would treat gap characters as missing data, as typically done in phylogenetics, but it would also be possible to include the gap character in the substitution model state space (see e.g. [49]).

Specific equilibrium character frequencies π are assigned to each alignment column. For each alignment, 800 columns (80%) are simulated as evolving under the same background equilibrium character frequency distribution, which we set to $\pi(A) = \pi(T) = 0.3$ and $\pi(C) = \pi(G) = 0.2$. The background character frequency distribution represents, in our simulations, the evolutionary dynamics of positions not strongly affected by selective forces; at these positions, the equilibrium character frequency distribution is constant because it is mostly determined by neutral mutational biases, which we assume constant across all alignment columns. The remaining 20% of alignment columns are simulated under position-specific selection, with

position-specific equilibrium character frequency π sampled from a Dirichlet distribution prior with $\alpha = 0.1$ (Additional file 1: Fig. S2A).

Our aim is, for each replicate and each alignment column, to infer π from the simulated sequences alone. For each replicate/alignment, we first infer a phylogenetic tree and alignment-wide HKY85 substitution model parameters using FastTree v2.1.10 [51]. We then consider this tree and the HKY85 κ parameter to be fixed and infer column-specific character equilibrium frequencies. While π is inferred separately at each column, the HKY85 alignment-wide parameters (including nucleotide frequencies) inferred with FastTree are used in some sequence weighting schemes (for example, in Eq. 3). The methods we used to infer equilibrium frequencies are:

- the observed character frequencies in the alignment column (the $p(j)$ described above),
- observed frequencies corrected using the HH94 [5] weights and Eq. 15,
- observed frequencies corrected using the GSC94 [9] weights and Eq. 15,
- observed frequencies corrected using the PNS weights w_s from Eq. 1 combined with Eq. 15,
- observed frequencies corrected using our PNS weights conditional on data, w_s^D from Eq. 2, combined with Eq. 15,
- observed frequencies corrected using our fast approximate PNS $\overline{w}_s$ weights (Eq. 14) combined with Eq. 15,
- Bayesian variants of the methods above, and
- ML phylogenetic inference (only of equilibrium character frequencies) with PhyML v3.1 [58].

All the methods above, except FastTree and PhyML, were implemented in custom Python scripts available from https://bitbucket.org/nicofmay/noveltyscores.

In addition to the basic simulation scenario, we also consider variant scenarios in order to investigate how certain parameters can affect the results:

- We consider alternative tree lengths, which we obtain by multiplying all branch lengths in the tree in Fig. 1 by a constant coefficient, either 0.2 or 5.
- We consider the case of amino acid characters instead of nucleotides. In this case, we simulate under an LG substitution model [59], and when we do inference we assume that the substitution model (including character frequencies) is known. Column-specific character frequencies are inferred as usual. In this case, equilibrium character frequencies for columns under selection are sampled from a Dirichlet distribution prior with $\alpha = 0.02$ (Additional file 1: Fig. S2B).
- To test the effect of very biased taxon sampling in an alignment, we added multiple (either 100 or 1000) human tips to the tree in Fig. 1. The short phylogeny relating the human sequences was randomly sampled at each replicate under a standard coalescent prior [60] with mean coalescent time 0.001 between human sequences. This short human phylogeny was then appended to the human tip of the tree in Fig. 1.

- To test the robustness of methods to the assumption of an ultrametric tree (a tree where all tips have equal distance from the root), we consider the case of a strongly non-ultrametric trees, as is common for some viruses such as influenza.

Results

Sequence weights

We implemented all the considered weighting schemes and all simulations within custom Python scripts (https://bitbucket.org/nicofmay/noveltyscores), making use of the phylogenetic python package dendropy [56]. We implemented all the algorithms presented in the Methods section for calculating weights w_s and w_s^D, and used comparisons of weights from different algorithms to assess the correctness of the implementations.

PNS shows similar trends to previous weighting schemes HH94 and GSC94, assigning higher weights to phylogenetically isolated taxa and smaller weights to taxa within clades with many other closely related taxa (Fig. 2). In particular, PNS seems to show an intermediate 'intensity' compared to the two other schemes. GSC94 weights are the most extreme, assigning the highest weights of any scheme to the most evolutionarily isolated taxa in the tree of Fig. 1, such as Lamprey, Coelacanth and frog *Xenopus tropicalis*. For example, for Lamprey, the GSC94 normalized weight is about 3 times larger than HH94, while PNS is about 2 times larger than HH94. Conversely, for taxa in over-represented clades, such as Human, GSC94 gives the smallest weight, HH94's weight being many times larger, and PNS being intermediate. Rescaling the branch lengths of the tree does not change this overall trend (Additional file 1: Fig. S3).

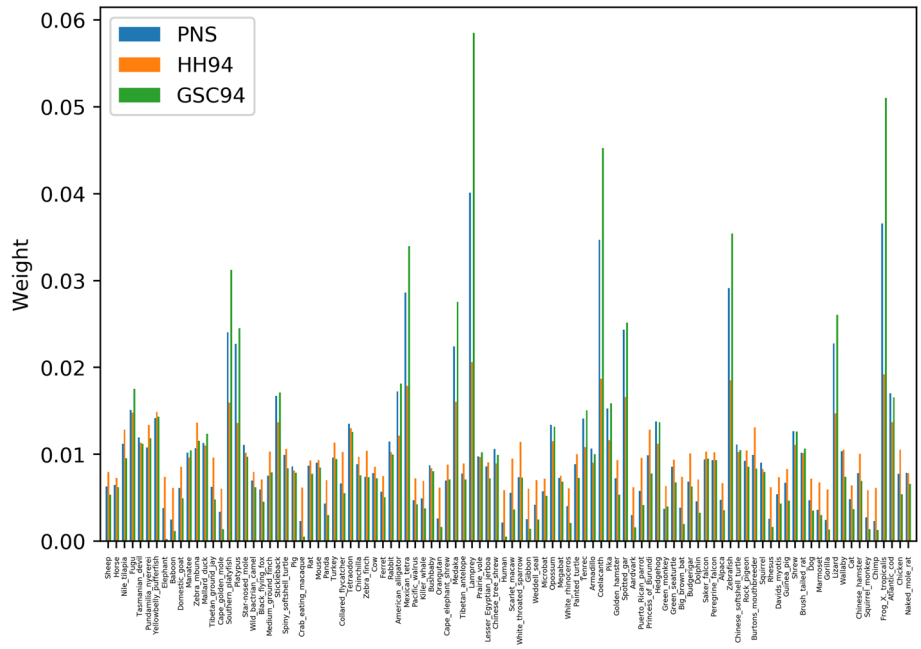


Fig. 2 Comparison of different weighting schemes. Bars show weights assigned to the tips of tree in Fig. 1 (species names on *x*-axis labels) in the scenario of nucleotide data (1 locus of 1kb) by different weighting schemes: PNS (weights w_5), HH94 [5] and GSC94 [9]. Weights from each scheme are normalized so that the sum over taxa is 1

Computational demand

Computational efficiency is one of the main requirements for applicability of weighting schemes, in particular when considering large datasets; for this reason, here we compare the computational demand of different approaches. Calculating sequence weights based on a phylogeny (PNS and GSC94) usually requires limited computational demand, with the dominant cost being the estimation of the phylogeny itself (Fig. 3 and Additional file 1: Fig. S4). One exception to this are the w_s^D weights of Eq. 2: these, being conditional on the data observed at a specific alignment column, need to be re-computed for each position. Calculation of these weights requires time cubic in N, and so it is not surprising that these weights are slower than phylogenetic inference. The other slowest method for character frequency is phylogenetic ML (PhyML), which also needs to be run once for each alignment column. All other approaches are at least one order of magnitude faster than PhyML and w_s^D in estimating character frequencies, and are practical also for larger trees (see e.g. Fig. S4 where we included 1000 closely related taxa). Calculating weights w_s^D and estimating frequencies by ML, instead, becomes infeasible on such larger trees. Estimating frequencies using HH94 weights is the fastest of the methods considered, as it does not require prior estimation of a phylogenetic tree, and it might therefore be one of the few possible choices available for extremely large datasets. The second fastest approach is GSC94, followed by the $\overline{w}_s$ weights, and finally by the w_s weights, although

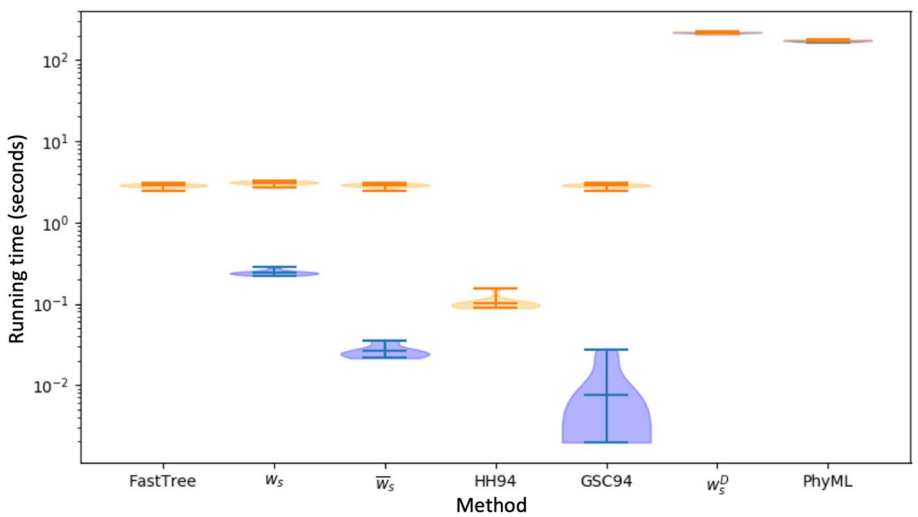


Fig. 3 Computational demand of different approaches to character frequency estimation. Violin plots summarise the running times, in seconds, of different methods. All analyses were run on a MacBook Pro 2017. Each plot contains values for 10 replicates of the scenario of the unscaled tree in Fig. 1 and nucleotide data. Time cost for computing frequencies from un-weigthed observed characters is not shown as it is negligible. Time demand of Bayesian variants of PNS weights is also not shown, as it is the same as for their non-Bayesian variants (Bayesian variants only require the addition of pseudocounts compared to non-Bayesian variants). 'FastTree' represents the cost of running phylogenetic inference with FastTree prior to weight calculation. Orange violin plots show the total cost (including computational cost of phylogenetic inference for methods requiring a phylogeny). Blue violin plots show the cost of calculating the scores without taking into account the cost of phylogenetic tree inference. For w_s^D and 'PhyML', blue and orange plots overlap. Calculating HH94 weights is, overall, the fastest approach among those considered here, as it does not require phylogenetic inference

these methods have very similar computational demand once the cost of inferring a phylogenetic tree is taken into account.

Accuracy of character frequency inference

Here we assess the ability of different weighting schemes, including those derived from our new PNS methods, to facilitate inference of column-specific character frequencies. We measure the accuracy of an approach by calculating, at each alignment column, the Euclidean distance between simulated and inferred character frequencies. ML phylogenetic inference with PhyML is almost always the most accurate method (Figs. 4 and 5). This is perhaps not surprising, given that this approach fully models the effects of varying equilibrium character distributions on character evolution along the phylogeny. However, this approach is also the most computationally demanding, and the advantage of schemes based on sequence weights is that they can be much faster, in particular on datasets with many sequences or many alignment columns. The only case where PhyML seems marginally less accurate than weights-based methods is at high divergence and strong selection (Fig. 5f). This is probably due to the particular implementation in PhyML, which does not allow character frequencies below 1%.

All the weighting schemes considered improve character frequency inference compared to the simplest approach of counting the observed characters at an alignment

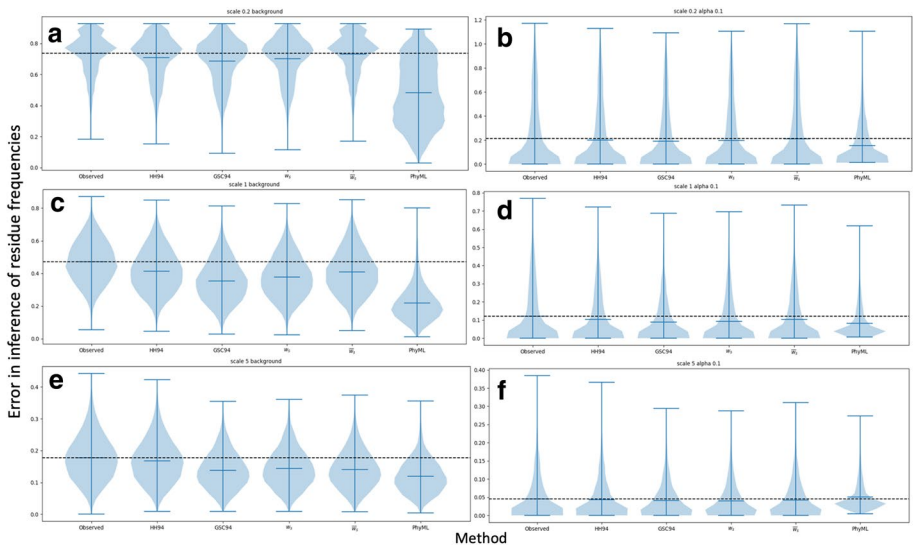


Fig. 4 Equilibrium frequency inference error. Comparison of the accuracy of different methods for reconstructing equilibrium frequencies in the basic simulation scenario (nucleotide characters and tree as in Fig. 1). Violin plots summarise the nucleotide frequency inference error (on the *y*-axis), measured as the Euclidean distance between the vectors of column-specific simulated nucleotide frequencies and inferred ones. Each plot contains 10 replicates, and each replicate contains 800 alignment columns evolved under the background nucleotide frequencies (**a**, **c** and **e**), or 200 alignment columns evolved under equilibrium nucleotide frequencies sampled from a Dirichlet distribution with $\alpha = 0.1$(**b**, **d** and **f**). Horizontal black dashed lines aid comparison by showing the median error of the first method (frequencies extracted from character counts). In **a** and **b** the tree branch lengths were scaled by a factor of 0.2; in **c** and **d** by a factor of 1.0; and in **e** and **f** by a factor of 5.0. Each plot shows results for a particular character frequency inference method, indicated on the *x*-axis. Results from additional methods (e.g. Bayesian approaches) are shown in Additional file 1: Fig. S5

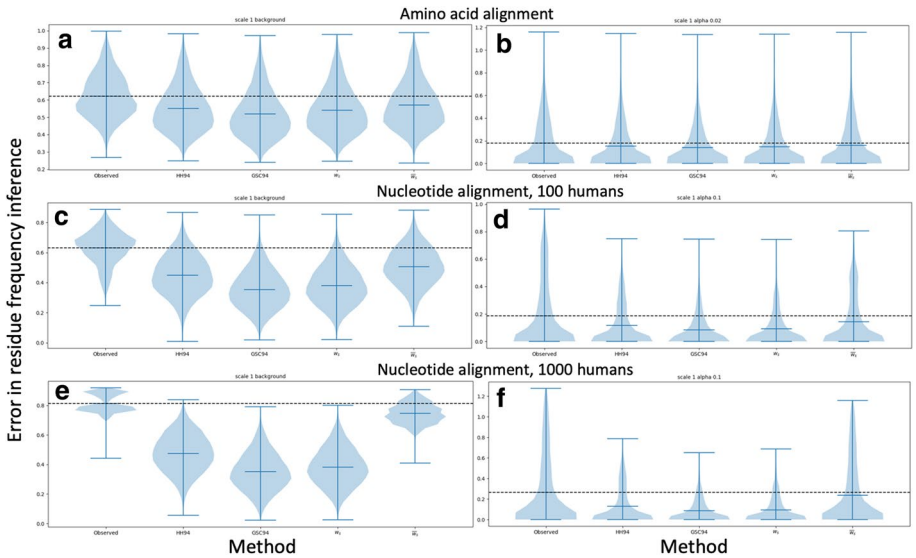


Fig. 5 Equilibrium frequency inference error under different scenarios. Similarly to Fig. 4, we compare the accuracy of different methods for reconstructing equilibrium frequencies. However, here we consider the simulation scenarios of amino acid sequences and modified trees with increased over-representation of human sequences. Values shown are as in Fig. 4. Each plot contains 10 replicates, and each replicate contains 800 alignment columns evolved under the background character frequencies (**a**, **c** and **e**), or 200 alignment columns evolved under equilibrium character frequencies sampled from a Dirichlet distribution with $\alpha = 0.1$ for **d** and **f** and $\alpha = 0.02$ for **b**. In **a** and **b** simulations are under the tree in Fig. 1 and with amino acid sequences. In **c** and **d** we consider nucleotide sequences and the tree in Fig. 1 with 100 added human sequences (see Methods). In **e** and **f** we instead add 1000 human sequences. Results from additional methods (e.g. Bayesian approaches) are shown in Additional file 1: Fig. S5. Results from PhyML are not available, due to excessive computational demand

column (Figs. 4 and 5). GSC94 and w_s weights seem to give more accurate results than HH94 and $\overline{w}_s$ weights, in particular within very biased datasets (Fig. 5c–f). The latter is not too surprising, given that weights $\overline{w}_s$ are an approximation of weights w_s.

We note that in Figs. 4 and 5 the weights w_s and GSC94 give very similar accuracy, with GSC94 sometimes marginally outperforming w_s. In theory, we expect the weights w_s, compared to GSC94, to benefit from the advantages of being based on intuitive mathematical principles and accounting for the effects of saturation. However, saturation probably has very little impact in this scenario (and in many real life scenarios); another important factor at play here might be that PNS gives more uniform weights compared to the more 'extreme' GSC94 weights. The latter might perform better in this case, possibly because PNS counts character observations as independent after one mutation event, when in reality more mutation events might be needed to approach near-independence of character observations. While our weights w_s (and also w_s^D) are calculated exactly (aside from rounding errors), this does not mean that an estimate of character frequencies based on these weights will be exact with respect to phylogenetic maximum likelihood optimization. Rather, they give an approximation, and our weights were not defined specifically in order to optimise character frequency estimation.

A limitation of GSC94 is that it does not work well with trees in which tips have very different distances from root (non-ultrametric trees). This effect has limited impact in our basic simulation scenario, as the tree in Fig. 1 is not far from ultrametric. However,

(See figure on next page.)

Fig. 6 Equilibrium frequency inference error with a strongly non-ultrametric tree. **a**: The strongly non-ultrametric phylogenetic tree under which simulations for this figure are performed. Some tips of the tree (e.g. T10, T20) are close to the root while others (T1, T11) are considerably more evolutionarily distant; in an ultrametric tree, all tips would instead have the same distance from the root. **b** and **c**: Violin plots summarising nucleotide frequency inference error (*y*-axis), measured as the Euclidean distance between the vectors of column-specific simulated nucleotide frequencies and inferred ones. Each plot contains 10 replicates, and each replicate contains (**b**) 800 alignment columns evolved under the background nucleotide frequencies, or (**c**) 200 alignment columns evolved under equilibrium nucleotide frequencies sampled from a Dirichlet distribution with $\alpha = 0.1$. Each plot refers to a particular character frequency inference method, indicated on the *x*-axis

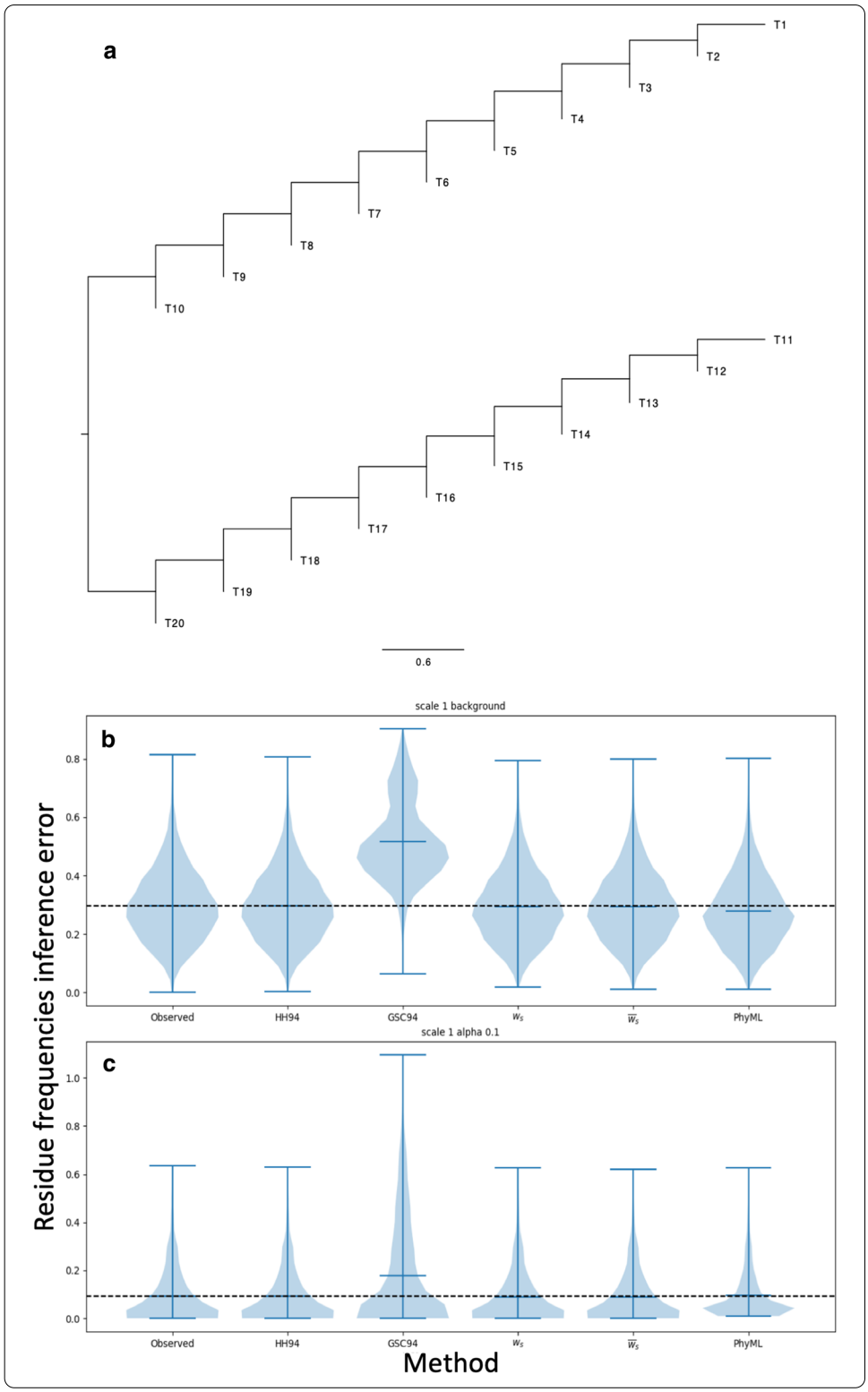
when we consider a strongly non-ultrametric tree (Fig. 6a), as is often observed for some viruses such as influenza [61, 62], we see that the GSC94 weights are strongly impacted, resulting in considerably worse inference than any of the other weighting schemes studied, and worse even than observed character frequencies (Fig. 6b). The reason is that, in such strongly non-ultrametric trees, GSC94 weights at terminal, younger tips tend to be considerably larger than GSC94 weights at older tips closer to internal nodes and in particular those closer to the root. Even in cases when observed characters close to internal nodes can provide useful information regarding equilibrium frequencies, for example when branches are sufficiently long in Fig. 6a, GSC94 weights are still almost exclusively distributed on the latest two phylogenetic tips in this scenario. All other approaches seem to perform similarly well in the scenario of Fig. 6, including simple base counting, and the likely reason is that here no clade is over-represented, and so a weighting scheme is not needed in the first place for the considered application.

Using sequence weights conditional on the data at the specific column, i.e. w_s^D from Eq. 2, unexpectedly does not seem to improve accuracy (Additional file 1: Fig. S5) while, as shown in Fig. 3, it does significantly impact computational demand. For these reasons, we do not generally recommend the use of weights w_s^D.

Using a Bayesian approach to character frequency inference means that the prior on character frequencies can affect the result of the inference. This can have a positive effect if the prior distribution is based on reliable evidence from sources other than the currently considered dataset. However, in our simulations we consider a completely arbitrary prior (corresponding to observing one character of each type at the considered alignment column) and this has the effect of slightly shifting the inferred frequencies closer to a uniform distribution (Additional file 1: Fig. S5). Expectedly, this overall improves character inference at sites evolving under the background frequencies, while it worsens inference at sites evolving under strong selection.

Discussion

We have proposed a new approach for assigning weights to the sequences in an alignment, or, equivalently, to the tips of a phylogenetic tree. First, we define phylogenetic novelty scores (PNS) based on rigorous mathematical principles. These scores summarise how novel is a sequence (respectively, tip), in evolutionary terms, with respect to the rest of the alignment (respectively, tree) and have a number of desirable properties, including meeting the objective criteria of [33].



We have showcased our scores' potential use by considering, as an example application, the inference of position-specific character frequencies. We demonstrate, using simulations, that our scores can improve accuracy of character frequency estimation

compared to some popular sequence weighting schemes, in particular HH94 [5] (see for example Figs. 4E and 5C, E). This however usually comes at the cost of additional computational demand, especially considering that our scores require the availability of an inferred phylogenetic tree, and considering that this might not be feasible for extremely large datasets. PNS and GSC94 [9] weights both require a phylogenetic tree estimate, and both show very similar performance in our main simulation scenario, with GSC94 marginally outperforming PNS. However, we demonstrate that, unlike GSC94 weights, PNS are not affected when the assumption of tree ultrametricity is violated (Fig. 6), and similar patterns are expected with respect to the robustness to the position of the tree. This shows that PNS are particularly versatile in applicability, as one would expect from their formal phylogenetic derivation. Over most scenarios, the most accurate method for position-specific character frequency inference seems to be standard phylogenetic ML inference; however, this approach is also very computationally demanding, and is not suitable for large datasets.

Character frequency inference, our example use-case for PNS, has a number of important applications. Character frequencies are fundamental parameters used in HMM profiling of protein families [3, 6], and our scores could therefore improve approaches to this task. Our scores could also be used to improve character frequency estimates used within alignment column-wise conservation scores [36–38], frequently defined as

$$R = S_{\max} - S_{\text{obs}} = \log_2 B - \left(-\sum_{j=1}^{B} p(j) \log_2 p(j) \right) \tag{20}$$

where $p(j)$ is the frequency of character j at a given alignment column, and B is the number of characters ($B = 4$ for nucleotides and $B = 20$ for amino acids). ($S_{\max}$ is the maximum possible entropy at the considered position, equal to $\log_2 B$, while S_{obs} is the observed value.) Typically, the $p(j)$ are inferred from the observed character frequencies at an alignment column; however, as we have shown, our PNS can significantly improve the inference of these frequencies, and therefore of conservation scores. Our simulations suggest that this is in fact the case (Additional file 1: Fig. S6).

Sequence weights, like our PNS, also have many other applications, for example to aid alignment inference. They have been shown to improve sequence alignment [1] and profile searches [5, 10], and examples of their use include PSI-BLAST [2] (which uses HH94 weights) and the CLUSTAL family of aligners (e.g. [1, 10] use GSC94 weights). Our scores could therefore result in improved alignments.

Sequence weights are also used to measure alignment quality, and our scores could be used for example in the context of the information content score (ICS) [63] or the norMD approaches [64]. Furthermore, our scores could be used in measures of conservation priority in conservation biology, such as phylogenetic diversity *PD* [25], quadratic diversity *Q* [30] and the phylogenetic entropy index H_{P} [31].

Lastly, we note that our scores could be used to improve the definition of phylogenetic effective sample size to be used for AICc [65] and BIC [66]. This is usually defined as the number of alignment columns, but this is not the only reasonable choice [67, 68].

Conclusions

We have proposed new sequence weights that benefit from a number of favourable properties and are derived from rigorous mathematical evolutionary principles. These weights do not enjoy the same level of computational efficiency and simplicity of some other methods, in particular due to the requirement of an input phylogenetic tree relating the considered sequences. However, when applied to the inference of character frequencies, we showed that these sequence weights can be used effectively in a broad range of scenarios, offering considerable computational advantage over full phylogenetic ML estimation, and often leading to more accurate estimates than other sequence weighting schemes. Thanks to their computational efficiency and robustness to phylogenetic assumptions, our phylogenetic novelty scores could have a positive impact in a number of fields, from sequence alignment and protein family profiling to phylogenetics and conservation biology.

Abbreviations

PIBD: Phylogenetically identical by descent; PNS: Phylogenetic novelty scores; ESN: Effective sequence number; HH94: Weighting scheme by Henikoff and Henikoff [5]; GSC94: Weighting scheme by Gerstein et al. [9]; HKY85: Substitution model by Hasegawa et al. [57]; LG: Substitution model by Le and Gascuel [59].

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-021-04183-8.

Additional file 1. Supplementary Methods and Figures.

Acknowledgements

We thank Julia de Beer, Sean Eddy and István Miklós for helpful discussions on these topics.

Author's contribution

NG devised the concept of Phylogenetic Novelty Scores (Eq. 1) as a weighting scheme for evolutionarily related entities. FP performed early computations of the fast approximation (Eq. 14). AVA and MAS contributed ideas toward the pruning methods described here. AUT performed simulations to derive estimates of $\boldsymbol{w_s}$ and $\boldsymbol{w_s^D}$ (the latter both 'inefficiently' and efficiently!). JT created an early implementation of the up-down pruning algorithm. WJC-S created code for plotting phylogenies scaled by PNS values (Fig. 1). NDM combined all these elements, added others, completed all theory and algorithms, implemented the methods, and ran the analyses presented. NDM and NG wrote the manuscript. All authors read and approved the final manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL.

Availability of data and materials

All scripts and data are available from https://bitbucket.org/nicofmay/noveltyscores.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Genome Campus, Hinxton, UK. [2]Present Address: Department of Public Health Sciences, Medical University of South Carolina, Charleston, SC, USA. [3]Present Address: LIRMM, University of Montpellier, CNRS, Montpellier, France. [4]Departments of Biostatistics, Biomathematics and Human Genetics, University of California, Los Angeles, CA, USA. [5]Present Address: Research IT Services, University College London, London, UK. [6]Present Address: RBC Borealis AI, Waterloo, ON, Canada.

Received: 4 December 2020 Accepted: 4 May 2021
Published online: 28 May 2021

References

1. Thompson JD, Higgins DG, Gibson TJ, Clustal W. Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucl Acids Res. 1994;22(22):4673–80.
2. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucl Acids Res. 1997;25(17):3389–402.
3. Eddy SR. Profile hidden Markov models. Bioinformatics. 1998;14(9):755–63.
4. Finn RD, Coggill P, Eberhardt RY, Eddy SR, Mistry J, Mitchell AL, Potter SC, Punta M, Qureshi M, Sangrador-Vegas A, et al. The Pfam protein families database: towards a more sustainable future. Nucl Acids Res. 2015;44(D1):279–85.
5. Henikoff S, Henikoff JG. Position-based sequence weights. J Mol Biol. 1994;243(4):574–8.
6. Finn RD, Clements J, Arndt W, Miller BL, Wheeler TJ, Schreiber F, Bateman A, Eddy SR. HMMER web server: 2015 update. Nucl Acids Res. 2015;43(W1):30–8.
7. Larkin MA, Blackshields G, Brown N, Chenna R, McGettigan PA, McWilliam H, Valentin F, Wallace IM, Wilm A, Lopez R, et al. Clustal W and Clustal X version 2.0. Bioinformatics. 2007;23(21):2947–8.
8. Sievers F, Wilm A, Dineen D, Gibson TJ, Karplus K, Li W, Lopez R, McWilliam H, Remmert M, Söding J, et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. Mol Syst Biol. 2011;7(1):539.
9. Gerstein M, Sonnhammer EL, Chothia C. Volume changes in protein evolution. J Mol Biol. 1994;236(4):1067–78.
10. Thompson JD, Higgins DG, Gibson TJ. Improved sensitivity of profile searches through the use of sequence weights and gap excision. Bioinformatics. 1994;10(1):19–29.
11. Eddy SR, Mitchison G, Durbin R. Maximum discrimination hidden Markov models of sequence consensus. J Comput Biol. 1995;2(1):9–23.
12. Henikoff S, Henikoff JG. Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci USA. 1992;89(22):10915–9.
13. Vingron M, Argos P. A fast and sensitive multiple sequence alignment algorithm. Bioinformatics. 1989;5(2):115–21.
14. Sibbald PR, Argos P. Weighting aligned protein or nucleic acid sequences to correct for unequal representation. J Mol Biol. 1990;216(4):813–8.
15. Altschul SF, Carroll RJ, Lipman DJ. Weights for data related by a tree. J Mol Biol. 1989;207(4):647–53.
16. Gotoh O. A weighting system and algorithm for aligning many phylogenetically related sequences. Bioinformatics. 1995;11(5):543–51.
17. Bruno WJ. Modeling residue usage in aligned protein sequences via maximum likelihood. Mol Biol Evol. 1996;13(10):1368–74.
18. Sunyaev SR, Eisenhaber F, Rodchenkov IV, Eisenhaber B, Tumanyan VG, Kuznetsov EN. PSIC: profile extraction from sequence alignments with position-specific counts of independent observations. Protein Eng. 1999;12(5):387–94.
19. Pei J, Grishin NV. AL2CO: calculation of positional conservation in a protein sequence alignment. Bioinformatics. 2001;17(8):700–12.
20. Cooper GM, Brudno M, N.I.S.C. Comparative Sequencing Program, Green ED, Batzoglou S, Sidow A. Quantitative estimates of sequence divergence for comparative analyses of mammalian genomes. Genome Res 2003;13(5): 813–8204
21. McAuliffe JD, Jordan MI, Pachter L. Subtree power analysis and species selection for comparative genomics. Proc Natl Acad Sci USA. 2005;102(22):7900–5.
22. Eddy SR. A model of the statistical power of comparative genome sequence analysis. PLoS Biol. 2005;3(1):10.
23. Newberg LA, Lawrence CE. Mammalian genomes ease location of human DNA functional segments but not their description. Stat Appl Genet Mol Biol. 2004;3(1):1–12.
24. Newberg LA. Effective species count and motif efficiency: the value of comparative genomics in characterizing conserved sequence positions. Technical Report 07-09, Department of Computer Science, Rensselaer Polytechnic Institute 2007.
25. Faith DP. Conservation evaluation and phylogenetic diversity. Biol Conserv. 1992;61(1):1–10.
26. Crozier R. Preserving the information content of species: genetic diversity, phylogeny, and conservation worth. Annu Rev Ecol Syst. 1997;28(1):243–68.
27. Pardi F, Goldman N. Species choice for comparative genomics: being greedy works. PLoS Genet. 2005;1(6):71.
28. Pardi F, Goldman N. Resource-aware taxon selection for maximizing phylogenetic diversity. Syst Biol. 2007;56(3):431–44.
29. Faller B, Pardi F, Steel M. Distribution of phylogenetic diversity under random extinction. J Theor Biol. 2008;251(2):286–96.
30. Rao CR. Diversity and dissimilarity coefficients: a unified approach. Theor Popul Biol. 1982;21(1):24–43.
31. Allen B, Kon M, Bar-Yam Y. A new phylogenetic diversity measure generalizing the Shannon index and its application to phyllostomid bats. Am Nat. 2009;174(2):236–43.
32. Guo S-W. Proportion of genome shared identical by descent by relatives: concept, computation, and applications. Am J Hum Genet. 1995;56(6):1468.
33. Vingron M, Sibbald PR. Weighting in sequence space: a comparison of methods in terms of generalized sequences. Proc Natl Acad Sci USA. 1993;90(19):8777–81.
34. Halpern AL, Bruno WJ. Evolutionary distances for protein-coding sequences: modeling site-specific residue frequencies. Mol Biol Evol. 1998;15(7):910–7.
35. Tamuri AU, Goldman N, dos Reis M. A penalized-likelihood method to estimate the distribution of selection coefficients from phylogenetic data. Genetics. 2014;197(1):257–71.
36. Schneider TD, Stormo GD, Gold L, Ehrenfeucht A. Information content of binding sites on nucleotide sequences. J Mol Biol. 1986;188(3):415–31.
37. Schneider TD, Stephens RM. Sequence logos: a new way to display consensus sequences. Nucl Acids Res. 1990;18(20):6097–100.

38. Crooks GE, Hon G, Chandonia J-M, Brenner SE. WebLogo: a sequence logo generator. Genome Res. 2004;14(6):1188–90.
39. Felsenstein J. Maximum-likelihood estimation of evolutionary trees from continuous characters. Am J Hum Genet. 1973;25(5):471.
40. Felsenstein J. Phylogenies and the comparative method. Am Nat. 1985;125(1):1–15.
41. Jukes TH, Cantor CR. Evolution of protein molecules. In: Munro HN, editor. Mammalian Protein Metabolism, vol. 3. New York: Academic Press; 1969. p. 21–132.
42. Gillespie DT. Exact stochastic simulation of coupled chemical reactions. J Phys Chem. 1977;81(25):2340–61.
43. Fletcher W, Yang Z. INDELible: a flexible simulator of biological sequence evolution. Mol Biol Evol. 2009;26(8):1879–88.
44. Sipos B, Massingham T, Jordan GE, Goldman N. Phylosim: Monte Carlo simulation of sequence evolution in the R statistical computing environment. BMC Bioinform. 2011;12:104.
45. Yang Z. PAML 4: phylogenetic analysis by maximum likelihood. Mol Biol Evol. 2007;24(8):1586–91.
46. Nielsen R. Mapping mutations on phylogenies. Syst Biol. 2002;51(5):729–39.
47. Felsenstein J. Evolutionary trees from DNA sequences: a maximum likelihood approach. J Mol Evol. 1981;17(6):368–76.
48. Arenas M. Trends in substitution models of molecular evolution. Front Genet. 2015;6:319.
49. Rivas E, Eddy SR. Probabilistic phylogenetic inference with insertions and deletions. PLoS Comput Biol. 2008;4(9):1000172.
50. Yang Z, Rannala B. Molecular phylogenetics: principles and practice. Nat Rev Genet. 2012;13(5):303.
51. Price MN, Dehal PS, Arkin AP. FastTree 2—approximately maximum-likelihood trees for large alignments. PLoS ONE. 2010;5(3):9490.
52. Drummond AJ, Rambaut A. BEAST: Bayesian evolutionary analysis by sampling trees. BMC Evol Biol. 2007;7(1):214.
53. Huelsenbeck JP, Ronquist F. MRBAYES: Bayesian inference of phylogenetic trees. Bioinformatics. 2001;17(8):754–5.
54. Frigyik BA, Kapila A, Gupta MR. Introduction to the Dirichlet distribution and related processes. Department of Electrical Engineering, University of Washington, Technical report UWEETR-2010-0006 2010.
55. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J et al. SciPy 1.0: fundamental algorithms for scientific computing in python. Nature Methods, 1–12 2020.
56. Sukumaran J, Holder MT. DendroPy: a Python library for phylogenetic computing. Bioinformatics. 2010;26(12):1569–71.
57. Hasegawa M, Kishino H, Yano T. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. J Mol Evol. 1985;22(2):160–74.
58. 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(3):307–21.
59. Le SQ, Gascuel O. An improved general amino acid replacement matrix. Mol Biol Evol. 2008;25(7):1307–20.
60. Kingman JFC. The coalescent. Stoch Proc Appl. 1982;13(3):235–48.
61. Nelson MI, Simonsen L, Viboud C, Miller MA, Holmes EC. Phylogenetic analysis reveals the global migration of seasonal influenza A viruses. PLoS Pathogens 2007;3(9).
62. Bedford T, Suchard MA, Lemey P, Dudas G, Gregory V, Hay AJ, McCauley JW, Russell CA, Smith DJ, Rambaut A. Integrating influenza antigenic dynamics with molecular evolution. eLife. 2014;3:01914.
63. Hertz GZ, Stormo GD. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. Bioinformatics. 1999;15(7):563–77.
64. Thompson JD, Plewniak F, Ripp R, Thierry J-C, Poch O. Towards a reliable objective function for multiple sequence alignments. J Mol Biol. 2001;314(4):937–51.
65. Sugiura N. Further analysis of the data by Akaike's information criterion and the finite corrections. Commun Stat Theory Methods. 1978;7(1):13–26.
66. Schwarz G. Estimating the dimension of a model. Ann Stat. 1978;6(2):461–4.
67. Darriba D, Taboada GL, Doallo R, Posada D. jModelTest 2: more models, new heuristics and parallel computing. Nat Methods. 2012;9(8):772.
68. Bartoszek K. Phylogenetic effective sample size. J Theor Biol. 2016;407:371–86.

Publisher's Note