Evaluation of Methods to Detect Shifts in Directional Selection at the Genome Scale

Louis Duchemin ,* Vincent Lanore Philippe Veber , and Bastien Boussau *

Laboratoire de Biométrie et Biologie Evolutive, Univ Lyon, Univ Lyon 1, CNRS, VetAgro Sup, UMR5558, Villeurbanne, France

***Corresponding authors:** E-mails: louis.duchemin@univ-lyon1.fr; bastien.boussau@univ-lyon1.fr.

Associate editor: Tal Pupko

Abstract

Identifying the footprints of selection in coding sequences can inform about the importance and function of individual sites. Analyses of the ratio of nonsynonymous to synonymous substitutions (d_N/d_S) have been widely used to pinpoint changes in the intensity of selection, but cannot distinguish them from changes in the direction of selection, that is, changes in the fitness of specific amino acids at a given position. A few methods that rely on amino-acid profiles to detect changes in directional selection have been designed, but their performances have not been well characterized. In this paper, we investigate the performance of six of these methods. We evaluate them on simulations along empirical phylogenies in which transition events have been annotated and compare their ability to detect sites that have undergone changes in the direction or intensity of selection to that of a widely used d_N/d_S approach, codeml's branch-site model A. We show that all methods have reduced performance in the presence of biased gene conversion but not CpG hypermutability. The best profile method, Pelican, a new implementation of Tamuri AU, Hay AJ, Goldstein RA. (2009. Identifying changes in selective constraints: host shifts in influenza. *PLoS Comput Biol.* 5(11):e1000564), performs as well as codeml in a range of conditions except for detecting relaxations of selection, and performs better when tree length increases, or in the presence of persistent positive selection. It is fast, enabling genome-scale searches for site-wise changes in the direction of selection associated with phenotypic changes.

Key words: **mutsel, profile models, directional selection, relaxation, codon model.**

Introduction

The genomes and phenotypes of extant species bear traces of past adaptations that occurred in their ancestors. A lot of research in molecular evolution has been devoted to detecting and interpreting these traces, both in noncoding and coding sequences (e.g., Moretti et al. 2014; Zhang et al. 2014; Marcovitz et al. 2019; Partha et al. 2019; Merényi et al. 2020). In protein-coding genes in particular, several approaches have been developed to study evolution at the level of whole genes or at the level of single sites (Goldman and Yang 1994; Pupko and Galtier 2002; Abhiman and Sonnhammer 2005; Penn et al. 2008; Yang and Nielsen 2008). Studies have found that amino-acid changes at a single position could create an active site de novo (Risso et al. 2017), that amino acid changes at a few positions could change the affinity of a hormone receptor for its ligand (Bridgham et al. 2006), that changes in rates of evolution accompanied the appearance of new HIV subtypes (Penn et al. 2008), that convergent evolution could be detected at single sites in proteins in mammals (Li et al. 2010), in grasses (Christin et al. 2007), in insects (Zhen et al. 2012), and that amino-acid changes at a single position could alter the dynamic of a worldwide viral epidemic (Korber et al. 2020). Identifying traces of past and current adaptations at the level of single amino-acid sites can thus be very insightful. In this article, we investigate the performance of several methods aiming to do just that. These include one commonly used d_N/d_S method, but also methods that have been more recently developed, based on *amino-acid fitness profiles*. Although other approaches based on the detection of shifts in the rate of sequence evolution have also been used to identify coding sites that have undergone selective pressure changes (Pupko and Galtier 2002; Abhiman and Sonnhammer 2005; Penn et al. 2008; Gu et al. 2013), they have not been evaluated in this study. They have been used less often than d_N/d_S methods, and have not been specifically designed to study the type of changes investigated in this study.

In proteins, amino acids that are never or seldom encountered at a particular site in a group of related species may have been selected against in the past. Those that are frequent may have been favored by selection. One can study these differences in frequency to infer differences in fitness between amino acids. A *fitness profile* is then used to represent the relative fitness of each amino acid at a given site (fig. 1*a*: A, B, C, and C′). When used within models of sequence evolution, a fitness profile determines the fixation probability of arising mutations during the process of evolution through mutation and selection

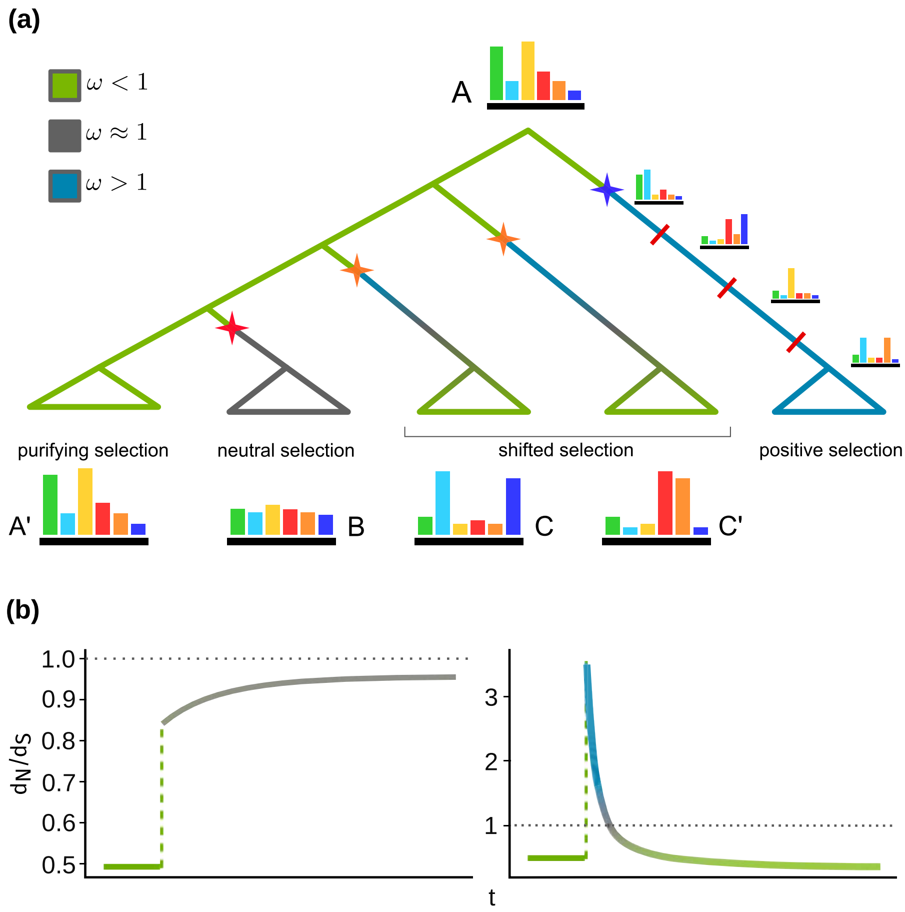

Fig. 1. Schematic representation of various evolution scenarii of a protein site involving profile changes. Colored stars indicate transition events that trigger profile changes. The color gradient along branches shows the variation of d_N/d_S a.k.a. ω values. The left subtree (green) is a case of purifying selection, with fixed profile (A) and $\omega < 1$. Moving right, the gray subtree illustrates relaxed pressure subsequent to the transition in red, resulting in a flattened profile B and $\omega \approx 1$. Further to the right, two cases of shifted selection are represented, each one driven by a different fitness profile (C and C′). In both cases, there is a transient increase in the value of ω, followed by a decrease toward $\omega < 1$, as represented on the right panel in (*b*). The rightmost subtree (blue) is an example of PPS (Tamuri 2021), where the fitness profile rapidly changes along the branch, at intervals marked with red bars. In this case, the value of d_N/d_S remains greater than 1 whereas positive selection continues. (*a*) Cases of selection regime changes, with their representation as fitness profile changes, and their equivalence with the d_N/d_S metric. (*b*) d_N/d_S variations over time. The curve on the left represents the simulated value of d_N/d_S when transition from purifying selection to relaxed selection occurs (transition between profiles A and B above). On the right is the variation of d_N/d_S during a shift in selection direction (transition between profiles A and C or C′ above).

(Halpern and Bruno 1998; Yang and Nielsen 2008; Rodrigue et al. 2010; Tamuri and Goldstein 2012). It also provides a *direction* for selection, which pushes evolution at the site away from low-fitness amino acids and toward high-fitness amino acids.

The shape of a fitness profile derives from selective pressures that operate at a particular site of a protein. These pressures can be related to phenotypic traits or environmental constraints, which could change over time. In such a case, the pressures would change, and so would the fitness profile. Selection may vary in intensity, for instance as a trait becomes more or less important for the global fitness of the organism; and in direction, when changing the value of a trait leads to higher fitness. These different kinds of changes in selective pressure can be captured by variations of the fitness profile: changes in intensity through the pointedness or flatness of the profile (fig. 1*a*, transition from profile A to profile B), and changes in direction through the variation of the overall shape of the profile (fig. 1*a*, transition from profile A to profiles C and C′). In this manuscript, we will focus on trait changes and the associated fitness profile at a site that occur discretely, at once, but progressive, continuous changes certainly occur in nature and would be important to consider.

Approaches to detect variations of selection on single sites of protein-coding sequences all require an annotation of the branches of a phylogeny, whereby each branch is associated to a phenotypic state or environmental condition. Given this annotation, either d_N/d_S or profile methods can be used (fig. 2). Other approaches that do not require such an annotation of branches have been proposed (e.g., Guindon et al. 2004; Dutheil et al. 2012; Murrell et al. 2012, 2015), but they are not evaluated in this work.

Approaches relying on the $\omega = d_N/d_S$ metric have been widely used to capture variations in selective pressure (Kosiol and Anisimova 2019), including in the context of

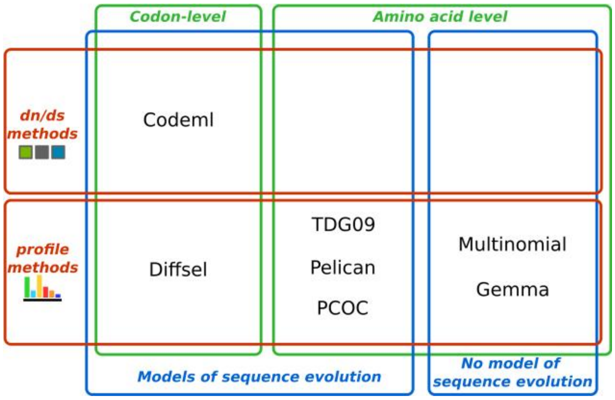

Fig. 2. Methods evaluated in the manuscript. Methods have been positioned based on whether they are based on d_N/d_S or amino-acid profiles, whether they work at the codon or amino-acid level, and whether they rely on a model of sequence evolution running along a phylogeny or not.

genome screening (e.g., Nielsen et al. 2005; Kosiol et al. 2008; Studer et al. 2008; Moretti et al. 2014; Zhang et al. 2014). These methods can show good reliability, either at the level of whole gene sequences or of single sites, when the generating process matches the inference model (e.g., Zhang 2005). The ω metric is defined as the ratio of rates between nonsynonymous (d_N) and synonymous (d_S) substitutions. The underlying assumption is that selection operates at the amino-acid level, so that synonymous codons provide the same fitness, whereas nonsynonymous substitutions induce a variation in fitness. In the popular "branch-site model A" available in PAML, inference is performed at the level of a gene by comparing the likelihood of a model with one set of d_N/d_S values per condition, against a model having one global set of d_N/d_S values through a likelihood ratio test (LRT) (Yang 2007). At the site level, the gene-wise parameter estimates are used to identify sites whose d_N/d_S has changed in a manner correlated with the annotation of the phylogeny (Yang 2005). However, other implementations have been proposed (e.g., Kosakovsky Pond et al. 2011; Murrell et al. 2015). All these d_N/d_S methods should be particularly effective at inferring changes in the intensity of negative selection: weaker (respectively stronger) selection should result in higher (resp. lower) d_N/d_S values. In that sense, d_N/d_S values have been used as a proxy of selection efficiency, even though in some cases this can be misleading (Spielman and Wilke 2015; Jones et al. 2019). In particular, under a constant amino-acid fitness profile, shifts between amino acids with non-0 fitness can result in cases of transient $d_N/d_S > 1$ (Jones et al. 2017), often interpreted as signaling positive selection on a particular branch. In addition, d_N/d_S methods should have good power to detect cases of persistent positive selection (PPS), that is, positive selection operating on all branches of the phylogeny, (rightmost branch, fig. 1*a*), which should result in high d_N/d_S values. However, they might be less effective at detecting changes in the direction of selection (Parto and Lartillot 2018), as they may fail to detect some sites that have undergone episodic changes in directional selection on top of a background of strong purifying selection (see fig. 1 and dos Reis 2015). Furthermore, they do not output estimates of the direction of selection, but only d_N/d_S values.

Profile methods have been developed more recently than d_N/d_S methods and have yet to be used at a genomic scale. They all rely on amino-acid profiles to identify sites that correlate with a phenotype along a phylogeny, but vary in the complexity of their underlying models. Some methods operate at the codon level and can explicitly use amino-acid fitness profiles by distinguishing between the mutation process, operating at the nucleotide level, and the selection process operating at the amino acid level (e.g., Murrell et al. 2012; Parto and Lartillot 2018). These methods build on the *mutsel* framework (Halpern and Bruno 1998; Yang and Nielsen 2008; Rodrigue et al. 2010; Tamuri and Goldstein 2012; Bloom 2014) that provides a better description of coding sequence evolution than d_N/d_S approaches (Bloom 2014; Spielman and Wilke 2016). Other methods operate at the amino-acid level and thus cannot model the mutation process. They use amino-acid *frequency* profiles as a proxy to *fitness* profiles (Tamuri et al. 2009), and may thus be less powerful than methods that operate at the codon level. In both cases, inference can be performed with a LRT at the site level, comparing the likelihood of a model with one profile per condition, against a model having one single profile that applies on all branches of the phylogeny. Such an approach can be problematic (Rodrigue 2013): there is a limited amount of information available in a single site to estimate one parameter per amino acid, or even several parameters if several profiles need to be considered. Unobserved amino acids typically are assigned a fitness or equilibrium frequency of 0, which is unrealistic. Furthermore, they do not contribute to the computation of the degrees of freedom when doing the LRT chi-square test, which can make it anticonservative. To mitigate some of these problems, approaches based on penalized likelihood have been proposed (Tamuri et al. 2014). In addition, Bayesian approaches that treat site-wise amino-acid profiles as a mixture distribution have also been used (Rodrigue et al. 2010; Rodrigue 2013; Rodrigue and Lartillot 2017; Rodrigue et al. 2020), including for a branch-heterogeneous mutsel model (Parto and Lartillot 2018). However, when comparing maximum likelihood, penalized, and Bayesian approaches, it was found (Spielman and Wilke 2016) that these methods often agreed in their estimates of site-wise selective constraints. In this manuscript, we evaluate two maximum likelihood methods (TDG09 and Pelican), and one Bayesian mixture approach (Diffsel).

Both d_N/d_S and profile methods to detect changes in selective pressures could be misled by nonadaptive processes, or by confounding between different selection regimes (Jones et al. 2019). Nonadaptive processes notably include GC-biased gene conversion (gBGC) (Ratnakumar et al. 2010; Bolívar et al. 2019) and CpG hypermutability (Meunier et al. 2005). gBGC occurs during recombination and mimics natural selection by favoring the fixation of G and C alleles. CpG hypermutability increases the mutation rate of CG dinucleotides. These processes can generate patterns in the sequence data that could lead to false positives or false negatives, as has been shown for d_N/d_S methods with respect to both gBGC (Ratnakumar et al. 2010; Guéguen and Duret 2018; Rousselle et al. 2019), and CpG hypermutability (Saunders and Green 2007; Suzuki et al. 2009). Confounding between different selection regimes could happen if a test aiming to find changes in the direction of selection detected sites under PPS. It is unclear how sensitive profile methods would be to these problems.

Genome-scale detection of changes in selective pressure requires a fast method. First, there can be thousands of gene families that each need to be analyzed with the method. Second, using a large number of species can increase the power of an analysis, but also increases its computational cost. In fact, it has been suggested that the high computational cost of d_N/d_S methods may be a hurdle to their more widespread use (Davydov et al. 2019). It is unclear how efficient profile methods could be.

In this article, we evaluate several profile and d_N/d_S methods to detect changes in selective pressures operating on individual positions of a protein-coding gene, on specific branches of a phylogeny. We consider several profile methods that have been published or that we have developed de novo, and compare them to a widely used d_N/d_S method. In particular, we ask whether profile methods can be as powerful as the d_N/d_S method, including in the presence of confounding factors, and pay particular attention to the computational costs of all methods.

Performance measurements are done on simulated data sets, allowing us to characterize the behavior of the methods on a range of tree shapes, branch lengths, and number of transitions along the phylogeny. We also investigate whether the detection methods are sensitive to confounding signal generated by nonadaptive processes of molecular evolution (Meunier et al. 2005; Ratnakumar et al. 2010; Bolívar et al. 2019), or by PPS (Tamuri 2021).

New Approaches

In this article, we introduce Pelican, an improved implementation of the model from Tamuri et al. (2009). This implementation was found to have better sensitivity and specificity than the original and is also faster thanks to optimizations on linear algebra computation.

Multinomial is a fast nonphylogenetic profile method that is also evaluated in this paper. It models observed amino-acid frequency profiles as multinomial distributions and compares the likelihoods at a given site of a single frequency profile versus multiple profiles through a LRT.

Both of these methods are implemented as a single program, that is made available to detect differential selection in protein sequence alignments. In this context, multinomial can be used as a fast filter on the alignment to reduce the amount of candidate sites to be evaluated through Pelican.

Results

We evaluated the performance of detection methods using simulated data sets. The methods that were considered are represented in figure 2 and include:

- codeml, a widely used d_N/d_S method for detecting positive selection, provided in the PAML toolkit (Yang 2007). codeml was configured to use the branch-site model A (Yang 2005; Zhang 2005), and works at the codon level.
- Multinomial, the simplest and fastest profile method, does not rely on a model of sequence evolution and works at the amino acid level. It uses a LRT to compare two models, one in which a single amino-acid profile is used to describe amino-acid frequencies observed at a site across all tip sequences, and one where different amino-acid profiles are used depending on the condition associated to the tip. Multinomial ignores the shape of the phylogeny and could thus be misled by phylogenetic inertia.
- Gemma (Zhou and Stephens 2012), based on a linear mixed model, was originally developed for genome-wide association studies (GWAS). It does not use a model of sequence evolution, but can take into account the structure of the phylogeny, encoded as a correlation matrix, which is introduced as a random effect in the mixed model. We used it at the amino-acid level, by encoding the protein alignment as an alignment of binary characters (see Methods). The phenotypes of species were encoded from the state of each tip taxon.
- TDG09 (Tamuri et al. 2009), a profile method that can be considered as a refinement over the Multinomial method, in that it also works at the amino-acid level but takes into account the phylogeny by relying on a model of sequence evolution. It uses a LRT to compare a model with one profile per condition and a model with one single global profile.
- Pelican, a new implementation of the model underlying TDG09 (Tamuri et al. 2009), originally motivated by the observed discrepancy reported between the performances of Diffsel and TDG09.
- PCOC (Rey et al. 2018), a profile method working at the amino-acid level. It is at its base similar to TDG09 but works with a limited set of pre-existing profiles, and further expects to observe substitutions at every transition between conditions in the phylogeny.
- Diffsel (Parto and Lartillot 2018), a profile method working at the codon level and based on a mutation-selection model in a Bayesian framework. Diffsel has performed significantly better than the other methods in a previous benchmark (Rey et al. 2019).

All simulations were done under a codon-based, time-reversible, mutation-selection model with site-specific amino-acid fitness profiles. The model was run along a phylogeny whose branches are annotated with two conditions that we refer to as *background* and *foreground*. A simulation generates codon and corresponding amino acid alignments of arbitrary length. Sites in the alignment may be either: (1) H_A sites, that are the result of a simulation where changes in the selection dynamic occur between background and foreground branches; (2) H_0 sites, resulting from an evolutionary process where selection is constant. The number of sites of each type was controlled in the simulation, allowing the comparison of predictions on the nature of each site (H_0 or H_A) with its known type, to estimate the performances of the prediction method.

Performance estimates in all the benchmarks were done using two metrics : *precision* and *recall*. Precision is the proportion of true positive sites among all sites identified as positive. Recall, also known as sensitivity, is the proportion of H_A sites that are identified as positive. These metrics were summarized by computing the area under the precision–recall curve (PR AUC). Confidence intervals for the PR AUC were computed according to Boyd et al. (2013).

In this section, we compared the detection methods using our simulation model in several contexts: (1) synthetic trees with variable branch lengths and numbers of transitions; (2) empirical trees in the presence or absence of confounding factors in the simulation. In the following, all branch length values are given in expected numbers of codon substitutions.

Detection Performances on Synthetic Trees

To characterize the behavior of the methods with respect to the number of transitions, the time spent in a condition, and branch lengths, we relied on synthetic trees with carefully controlled features.

Detection Performances Increase with the Number of Transitions

We investigated whether the number of transitions from background to foreground conditions had an effect on detection performances. We generated a balanced tree of 128 tips in which all branch lengths equal 0.01, and generated a variable number of transitions on terminal branches (tree topology shown in supplementary fig. S1, Supplementary Material online). In this setting, both the number of foreground leaves and the total time in the foreground condition increase with the number of transitions. Results shown in figure 3*a* show that all methods take advantage from such increases.

The Amount of Time Spent in a Condition Has a Large Effect on Detection Performance for Phylogenetic Methods

We next evaluated the relative importance of the number of transitions and the amount of time spent in the foreground condition on the phylogeny. We used a different set of trees with the same general features (128 tips, branch lengths equal 0.01), varied the numbers of transitions, but kept the number of foreground leaves and total foreground length constant across trees. This was done by normalizing the branch lengths to achieve equal total times between foreground and background conditions, and across trees. As a result the number of tips in each foreground subtree is variable, depending on the depth of the transition event. For a given number of transitions, all transitions occur at the same depth in the tree (supplementary fig. S2, Supplementary Material online).

Figure 3*b* shows that the performances of Gemma and Multinomial increase with the number of transitions, even when the amount of time spent in the foreground condition is kept constant. They become the best-performing methods at 64 transitions. However, the phylogenetic methods codeml and Pelican seem to be less sensitive to this parameter in this experiment, suggesting that the determining factors for their performances in the previous experiment were the total foreground time and/or the number of foreground leaves, which are kept constant in this experiment.

Profile Methods Improve As Branch Lengths Increase

In order to assess the effect of branch lengths and of the distance between transition events and foreground leaves on method accuracy, although keeping the number of transitions constant, we evaluated each method on a balanced tree with four transition events where a scaling factor was applied to the branch lengths (supplementary fig. S3, Supplementary Material online). As a side-effect, this scaling factor also applies to the total foreground tree length.

Results in figure 3*c* highlight two opposite trends between profile methods and the d_N/d_S method codeml, in relation with the branch length scaling. Profile methods tend to be more accurate in detecting selection shifts when the branch lengths increase, whereas the performance of codeml decreases. We suspect that as branch lengths increase, the number of synonymous substitutions increases, which reduces d_N/d_S and makes it harder to detect H_A sites (see fig. 1*b*, right).

Among profile methods, the performance gap tends to decrease with longer branches.

Detection Performances on Empirical Phylogenies

To benchmark the methods in a more realistic context, we evaluated their performances on empirical phylogenies that differ in their size, depth, and number of transitions (table 1). The corresponding phylogenetic trees are shown as supplementary material (supplementary figs. S4–S9, Supplementary Material online).

Alignments were simulated as in the previous experiments, using the simulation model running along the empirical phylogenies. These alignments were used to measure the statistical calibration and the throughput of each method, and to evaluate each method as in the previous section.

Pelican Performs Well on Empirical Phylogenies

We assessed whether the methods were well calibrated, that is, how accurate was their reported false positive

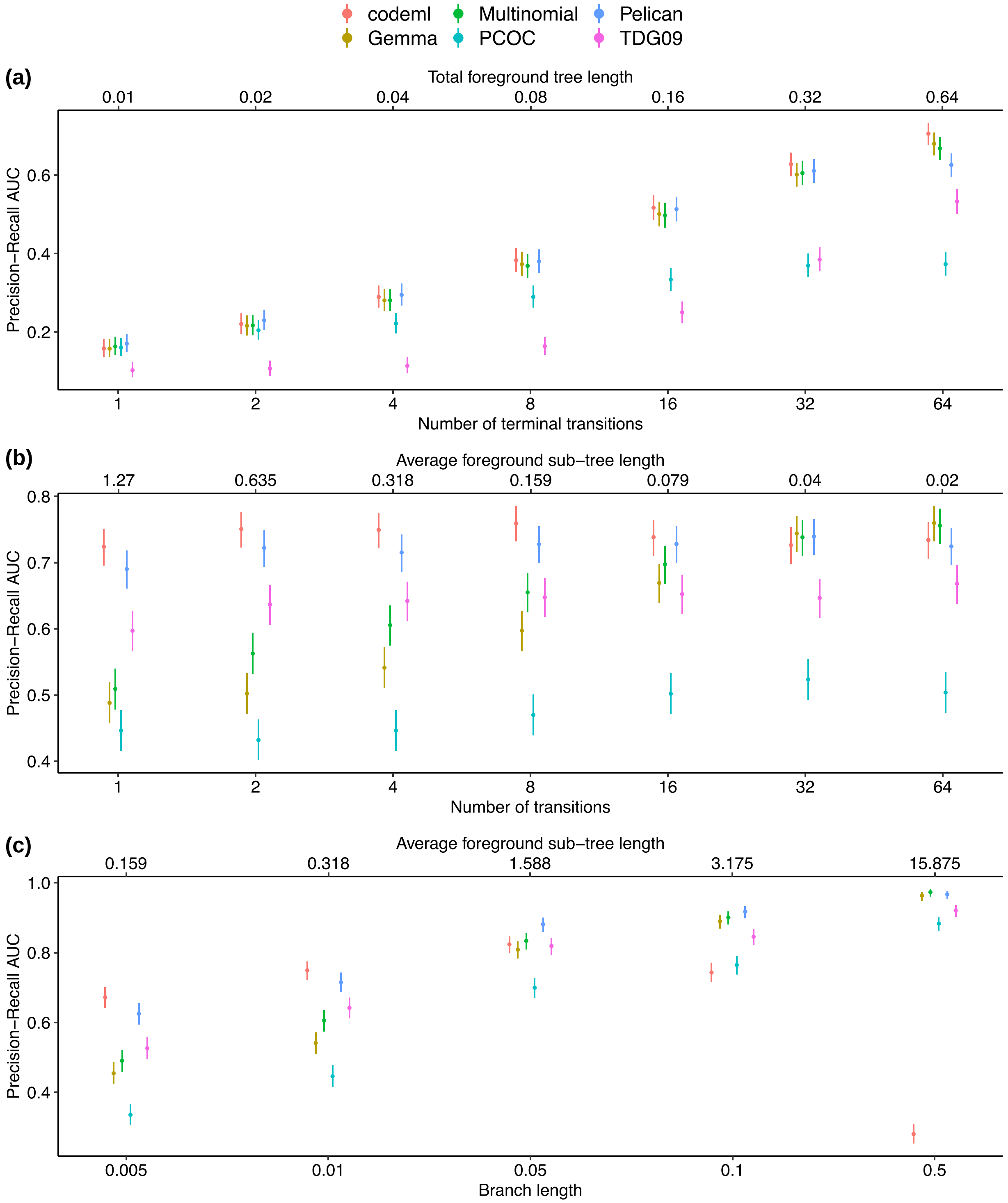

Fig. 3. Detection performances evaluated on synthetic trees. 95% confidence intervals accounting for the variability of the PR AUC estimates are shown. (*a*) Performance increases with the number of transitions on terminal branches. (*b*) The number of transitions is not the determining factor for the performance of the phylogenetic methods but has a strong effect on the performance of Gemma and Multinomial. (*c*) Performances of the profile methods are positively correlated to the branch lengths, whereas the performance of codeml decreases on longer branches.

rate under the null (H0) model. To this end, we simulated 9,000 sites under H0, and counted the number of false positives for each method, at the 0.05 *p*-value threshold. Under this setting, a well-calibrated method should produce on average a number of false positives equal to 5% of the total number of sites. Results shown in supplementary table S1, Supplementary Material online indicate that most methods are overly conservative, that is, their observed false positive rate is lower than their advertised (5% here) false positive rate. Multinomial is the

Table 1. Summary Statistics on Empirical Trees.

Data Set	Depth	Size	Transitions	Avg Branch Length		Avg Subtree Length
				Global	Foreground	Foreground
Rodents (Rey et al. 2019)	11	32	10	0.0192	0.0252	0.0353
Cyperaceae (Besnard et al. 2009)	25	79	5	0.0207	0.0239	0.196
Echolocation (Scornavacca et al. 2019)	18	116	3	0.0081	0.0061	0.0828
Amaranthaceae (Kapralov et al. 2012)	22	179	15	0.0045	0.0035	0.0356
HIV RTi (Murrell et al. 2012)	34	476	238	0.0063	0.0051	0.0051
Influenza H1 segment (Tamuri et al. 2009)	61	434	1	0.0603	0.0608	49.0709

Note.—Tree depth is defined here as the highest number of branches between a leaf and the root. Size is the number of leaves in the tree. Transitions are defined as changes from the background to the foreground condition.

only method that can yield a higher rate of false positives, particularly on the Influenza phylogeny. To further assess how conservative the methods were, we computed the observed false positive rate on nonconstant sites only, given that constant sites cannot be classified as positive. Supplementary table S2, Supplementary Material online indicates that even on this subset of sites, most methods still have low rates of false positives. This indicates that all methods except Multinomial are overly conservative.

We then assessed the performance of the methods to detect H_A sites by simulating 1,000 H_A sites and 9,000 H0 sites. Pelican, codeml and Diffsel consistently show the best performances on all data sets (fig. 4), with the exception of the Influenza H1 data set. It is worth noting that Diffsel is one of the best-performing methods, even though it estimates branch lengths and does not get them as input, like most other methods.

We note that, although Pelican is essentially a reimplementation of TDG09, it shows significantly better performances on every data set. codeml and Pelican have similar performances in general. However, on the Influenza H1 data set, which has the highest average foreground subtree length ($\bar{d} = 49.0709$), codeml incurs a large drop in its performances. These observations are consistent with the results obtained on synthetic trees (fig. 3*c*).

Even though the HIV data set has the lowest average foreground subtree length ($d = 0.0051$), Pelican performs better than codeml on this data set. Performances are strongly increased for all methods on this data set, compared with the other empirical phylogenies. Our explanation for the results on the HIV data set involves multiple effects: (1) the large number of transitions ($n = 238$) on terminal branches yields a strong signal for all methods, which benefits profile methods the most (see fig. 3*a*); (2) figure 3*c* seems to indicate that there is an optimal branch length for codeml: the signal for d_N/d_S falls off on longer branches, but branches can also be too short to allow reliable d_N and d_S estimations because of the insufficient number of substitutions occurring in such a short time span.

We showed that some characteristics of the phylogenies had a major effect on method performance, particularly the time spent in the foreground condition, as well as the number of transitions in the phylogeny. It is likely that variations in the detection performances are the results of interactions between the features of the phylogeny, possibly including more than the two we identified, as well as the sensitivity of the detection method to these features.

On a side note, we remark that Multinomial shows some surprisingly good performances despite its simplicity. As it does not take any information from the phylogeny, it is the simplest profile method, and also the fastest (table 2). However, experiments on synthetic trees show that in cases where phylogenetic structure creates a lot of phylogenetic inertia, the performance of Multinomial can be strongly reduced (supplementary fig. S10, Supplementary Material online).

Performances in the Detection of Changes in the Intensity of Selection

Profile methods are in principle particularly appropriate for detecting changes in the direction of selection, and in practice perform as well as codeml and better on long branches (see above). We evaluated how they perform in the presence of a change in the intensity of selection, by simulating a scenario of relaxation of selection. In this scenario, H_A sites are simulated such that all amino acids have equal fitness on foreground branches. This corresponds to a complete relaxation of selection.

Figure 5 indicates that profile methods, and Pelican in particular, can also detect relaxations of selection, but that their performance depends on the phylogeny. In particular, we find that in some cases the detection is unreliable (fig. 5, Influenza panel). We suspected that this lower performance was due to a lack of sensitivity, and tested this hypothesis by changing the computation of degrees of freedom in the LRT performed in Pelican (supplementary section S6, Supplementary Material online). Supplementary fig. S15, Supplementary Material online shows that much better performances can be obtained on the Influenza data set, but with some cost on the performance of the method on other data sets (notably Mammals echolocation). Future work on the LRT may result in an improved performance of Pelican across data sets, in settings of changes in both the direction and the intensity of selection.

Performances in the Presence of Confounding Factors

In order to assess the robustness of the detection to other evolutionary processes, we executed a benchmark on

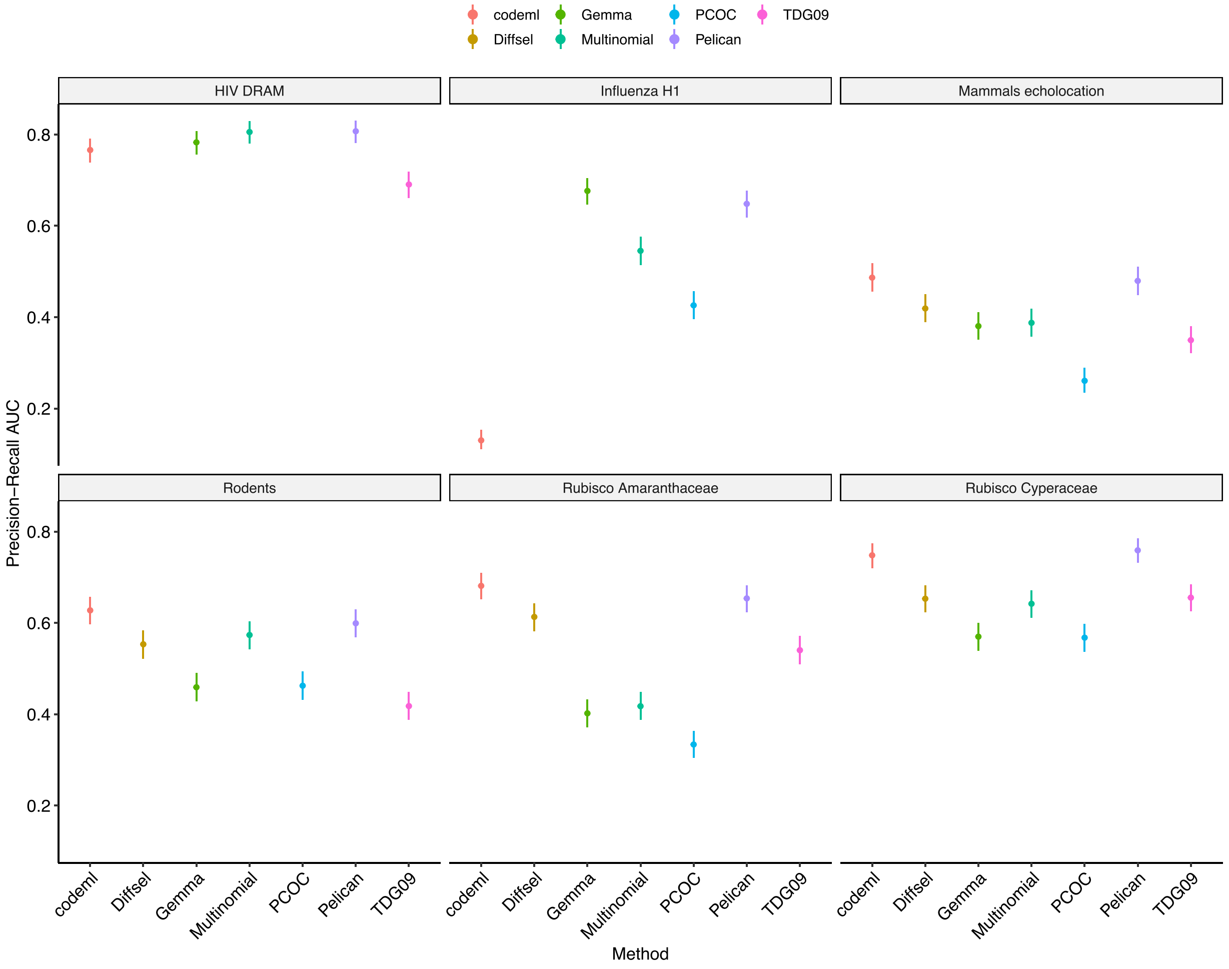

Fig. 4. Precision–recall area under the curve (AUC) estimates on simulated data sets using six empirical phylogenies, under changes in the direction of selection. Performances of TDG09 on the Influenza H1 data set were not successfully measured. Diffsel was not evaluated on the HIV and Influenza data set due to the large computation times involved. PCOC had an underflow error on the HIV data set.

Table 2. Execution Times for One Alignment Containing 100 H_0 and 100 H_A Sites Generated Using Our Collection of Empirical Phylogenies.

Method	Execution Time (s)					
	Cyperaceae	*Amaranthaceae*	Rodents	Echolocation	HIV	Influenza
Multinomial	0.01	0.02	0.01	0.02	0.04	0.03
Gemma	1.79	1.90	1.72	1.81	1.96	2.03
Pelican	10.93	19.42	2.58	6.72	87.72	266.84
TDG09	22.21	40.56	6.84	12.56	369.87	
codeml	60.60	172.50	27.37	100.76	443.66	614.33
PCOC	65.56	129.01	27.80	76.48	346.25	436.84
Diffsel	1,253.00	1,497.84	946.79	1,083.48	2,659.78	3,982.00

Note.—Result for TDG09 on the Influenza data set is not available due to the program not terminating within a reasonable amount of time.

simulations including confounding factors: (1) CpG hypermutability, which induces a higher mutation rate on methylated CpG dinucleotides; (2) gBGC, a nonadaptive process that increases the overall GC content in the genome and may be mistaken as a selective force (Ratnakumar et al. 2010); (3) PPS, as modeled by Tamuri (2021), which favors nonsynonymous substitutions over synonymous ones on the branches where it occurs. We used strong but realistic intensities for each of these processes, with two intensities for gBGC, and two intensities for PPS. In simulations of CpG hypermutability and gBGC, the processes were applied on foreground branches for both H0 and HA sites. In the simulation of PPS, the process was applied on all branches, but only on H0 sites, to assess the propensity of each method to generate false positives. Results are shown in figure 6 for the Echolocation phylogeny and are available as supplementary material for the other phylogenies.

We find that the presence of CpG hypermutability has no influence on the detection performance in most cases.

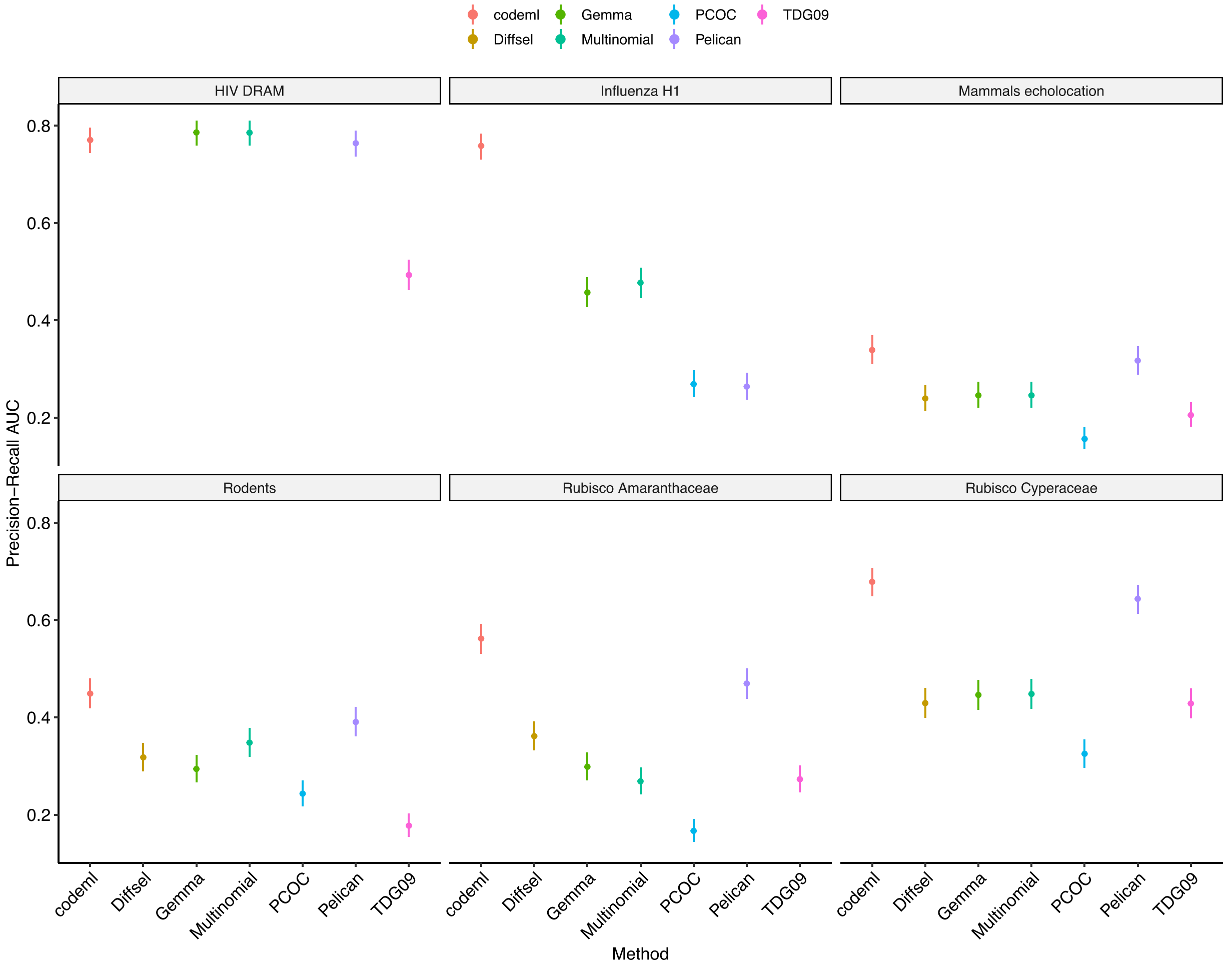

Fig. 5. Precision–recall area under the curve (AUC) estimates on simulated data sets using six empirical phylogenies, under relaxation of selection. Performances of TDG09 on the Influenza H1 data set were not successfully measured. Diffsel was not evaluated on the HIV and Influenza data set due to the large computation times involved. PCOC had an underflow error on the HIV data set.

In contrast, on simulations including gBGC, we notice a strong decrease of the performance for every method. Although gBGC happens at the nucleotide level, it generates selection-like signal at the codon (or amino acid) level, that is not the result of an adaptive process. This signal was strong enough to directly interfere with the signal for selection on genomic sites.

At a fixed effective population size N_e, an increase in PPS results in a decrease in the performance of all methods. Supplementary fig. S14, Supplementary Material online shows that this decrease is mostly due to an increase in the number of false positives. Under conditions of strong PPS and large N_e, the performance of codeml is strongly reduced, but the performance of profile methods can be improved. Overall, profile methods seem less prone to generating false positives in the presence of PPS, the effect of which is largely compensated by an increased value of N_e.

Throughput Varies Greatly Between Methods

We measured execution time for each method on six simulated data sets. Simulations were made using our simulation model on each empirical tree to generate an alignment of 100 H_0 and 100 H_A sites. Execution times were measured as the elapsed time at completion of a run for each method using a single CPU and are presented in table 2. The throughput of phylogenetic methods can vary by a large factor depending on the size of the phylogeny.

Multinomial and Gemma are the fastest methods by a large factor. None of these two methods require parameter estimations for a model of sequence evolution, allowing faster execution. At the other end, the two codon-level methods codeml and Diffsel are the slowest. Pelican is the fastest of the phylogenetic methods by a non-negligible factor on all data sets.

Discussion

In this paper, we used simulations to compare the performance of methods that detect changes in the direction and intensity of selection, given an annotation of a phylogeny. These simulations rely on mutation-selection models of codon sequence evolution running along phylogenies.

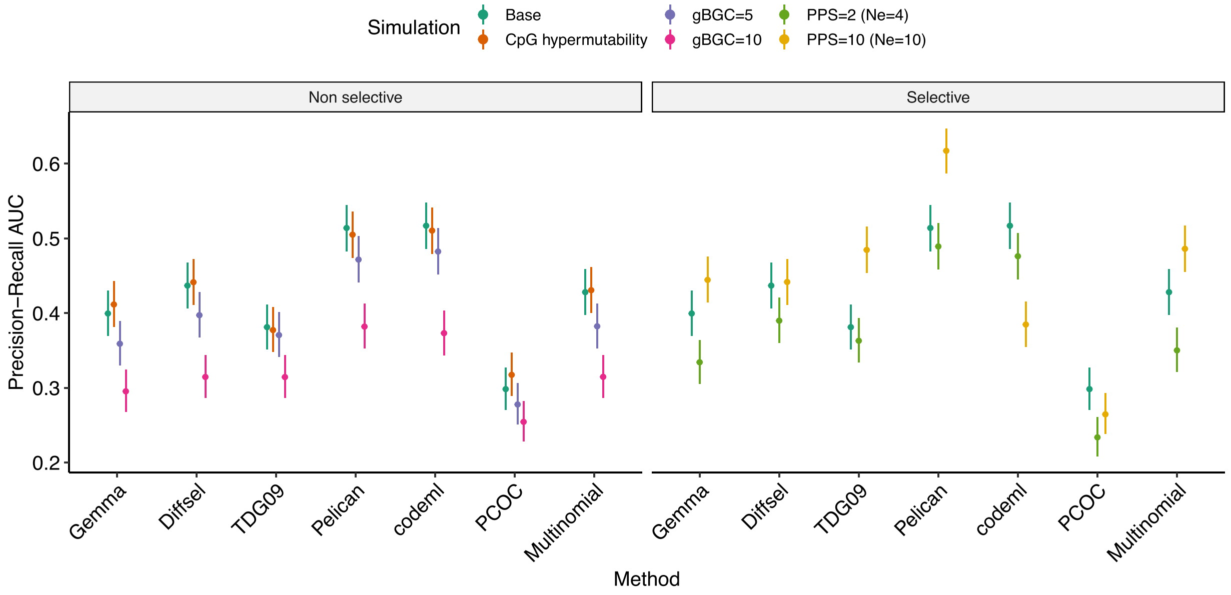

FIG. 6. Effects of gBGC, CpG hypermutability, and PPS on precision–recall AUC on the Echolocation data set.

Mutation-Selection Models for Simulating Coding Sequences

Our choice to rely on mutation-selection models stems from the fact that these models have been shown to be more realistic for coding sequences than d_N/d_S methods (Bloom 2014; Spielman and Wilke 2016). They distinguish between processes occurring at the mutation level, and processes occurring at the selection level among codons. This flexibility allowed us to implement in our simulations CpG hypermutability and gBGC. In addition, we have made the choice to use site-heterogeneous amino-acid fitness profiles to emulate the heterogeneity among positions in protein sequences. For improved realism, the profiles we used come from Rey et al. (2019), and are based on laboratory mutagenesis experiments (Bloom 2017). However, we assumed no fitness difference between synonymous codons, even though this can be implemented in the mutation-selection framework (Yang and Nielsen 2008; Pouyet et al. 2016) to reflect selective pressures at the RNA or DNA level. Such selective pressures could be mistaken for selection at the protein level (Rubinstein et al. 2011; Spielman and Wilke 2015), and therefore act as another confounding factor for the detection methods, but they were not investigated here. Despite this, and given the fact that we simulated along empirical phylogenies, we expect our results to be informative about the performance of the methods on empirical data sets.

Methods Working at the Amino-Acid Level Perform As Well As Codon-Based Methods

Some of the methods in the benchmark rely on models that are similar to our simulation model. In particular, Diffsel is also based on a mutation-selection model, and codeml works at the codon level. Expectedly, these two methods perform well on our simulations. In agreement with previous results (Spielman and Wilke 2015), codeml, which relies on d_N/d_S and does not use amino-acid fitness profiles, is very effective except on long branches and trees (figs. 3c and 4, the Influenza H1 phylogeny). All the other methods work at the amino-acid level. Among those, the models based on a phylogenetic model (Pelican, TDG09, PCOC) vary in their performance, with Pelican standing out as the best performer. The lower performance of PCOC is likely due to two of its characteristics. First, its reliance on a predefined set of amino-acid frequency vectors, which may prevent it from accurately fitting the sites under study. Second, its "One-Change" component, which requires an amino-acid change at each transition between background and foreground branches. This second limitation by design reduces the number of positive sites it can detect. TDG09 has a lower performance than its reimplementation Pelican. The two implementations agree in the majority of cases but disagree on some sites, likely due to optimization problems on boundary cases, which penalize TDG09. The fact that Pelican's performance is similar to the performance of codon-based models suggests that its reliance on a WAG exchangeability matrix, not used in the simulation model, is not harmful. Furthermore, it suggests that no information present only at the codon level is of much use to codeml or Diffsel, even when sequences are simulated with a model of CpG hypermutability (fig. 6). This may seem surprising, but probably relates to how we specified the detection problem we addressed. It is entirely centered around the amino-acid profiles, so the codon level does not provide much useful information. Finally, the nonphylogenetic methods perform quite well despite their simplicity. Multinomial, the simplest of our methods, performs better than Gemma, which has the ability to include the shape of the phylogeny as a covariate. This may be because Gemma was designed to handle binary characters, and we had to transform the amino acid data before feeding it into Gemma (see methods).

Beyond detection efficacy, the d_N/d_S and profile methods that we discuss in this manuscript vary in their execution speed. Methods that rely on models of sequence evolution typically have large computational footprints due to the use of the pruning or sum-product algorithm (Felsenstein 1981), and the need for frequent matrix exponentiations. The computational footprints of these operations become larger as the state space grows: methods that work at the codon level (61 states) are more demanding than methods that work at the amino-acid level (20 states) (fig. 2). Therefore, the profile methods that work at the amino-acid level benefit from a computational advantage compared with codon-level profile methods or d_N/d_S methods. Diffsel is the slowest method despite a thoroughly optimized code base, for several reasons. First, it works at the codon level. Second, it attempts to estimate more parameters than the other methods, and notably branch lengths. Third, it is the only Bayesian method here, and as such is the only one providing a credible interval for each parameter, at each position, where the other methods only provide point estimates. Pelican's speed is better than TDG09's, due to the reliance on high performance computing libraries (see methods). It also uses diagonalization for matrix exponentiations, or the contraction of sparse substitution matrices to matrices of lower sizes as in the original method (Tamuri et al. 2009). It has thus already been extensively optimized, but further improvements might be obtained by using substitution mapping and summary statistics as in Diffsel (Parto and Lartillot 2018).

Features of a Data Set that Affect Performances

Results obtained in this benchmark highlight that profile and d_N/d_S methods perform differently in detecting changes in directional selection, depending on the features of a data set. We identified a set of tree features that appear to have an effect on the performances: the number of transitions from background to foreground condition, the total time in the foreground condition, the number of foreground leaves, and the average length of foreground subtrees. The variations in the detection performances observed on empirical phylogenies (fig. 4) likely are the result of interactions between these features, and possibly others that have yet to be identified.

Both Pelican and codeml benefit from an increased number of leaves in the foreground condition (fig. 3*a*), but not from an increased number of transitions (fig. 3*b*). However, nonphylogenetic methods (Multinomial, Gemma) benefit from increasing any of these features, including the number of transitions. codeml tends to perform better than profile methods on phylogenies with shorter foreground subtrees on average. This conforms to our understanding of the two types of methods, and the kind of signal they rely on, as presented in section "Introduction." In the case of a change in the direction of selection, the resulting burst of the d_N/d_S ratio occurs over a short time period, and quickly decreases back to a purifying selection regime ($d_N/d_S < 1$, fig. 1*b*). This implies that on longer branches more time is spent in a purifying selection regime, reducing the signal for high d_N/d_S as the rate of nonsynonymous substitutions decreases. In contrast, profile methods rely on amino-acid frequencies to detect positive selection. In this case, the signal is the strongest when the amino-acid frequencies have reached an equilibrium and differ the most from the ancestral frequency distribution. Since reaching the foreground equilibrium distribution through substitutions takes time, detection performances tend to increase on longer branches (fig. 3*c*).

Profile methods that do not take into account the phylogeny have a reduced performance on short branches. In that case observations at the leaves of the phylogenetic tree are more strongly correlated and this may mislead methods that assume independent observations (like Multinomial) or rely on a less accurate model (like Gemma). On longer branches, observations at the leaves of the tree tend to become more independent, and nonphylogenetic methods exhibit performances similar to their more complex counterparts.

gBGC Is an Important Confounding Factor for both d_N/d_S and Profile Methods

In an effort to make our simulations more realistic, we introduced two nonadaptive confounding factors in our model: CpG hypermutability, which affects the mutation component, and GC-biased conversion (gBGC), which affects the selection component. We have found that introducing gBGC on foreground branches induces a significant drop in performances for all the methods, with higher values of gBGC resulting in larger decreases (fig. 6). gBGC mimics selection, independently of the underlying fitness profiles, and scrambles the signal used to detect changes in the selection regime. This corroborates previous studies on the role of gBGC in disrupting the detection of selection in genome sequences (Ratnakumar et al. 2010; Guéguen and Duret 2018; Rousselle et al. 2019; Ho and Hurst 2022). Mechanistic codon-level models such as Diffsel could be extended to account for this effect, and untangle it from directional selection.

On the other hand, strong CpG hypermutability was not found to induce changes in the performance in most cases. It is possible that codons that contain CG dinucleotides are not frequent enough in our simulations based on the mutsel framework to reduce the AUC metric.

PPS Is an Important Confounding Factor for d_N/d_S Methods, Less So for Profile Methods

Protein sites may be subject to a variety of selection regimes (fig. 1*a*). It may be difficult to distinguish sites undergoing changes in the direction of selection from sites evolving under a different selection regime, in particular PPS. In our simulations under the model of Tamuri

(2021), we found that codeml had difficulty distinguishing the two processes, in agreement with Parto and Lartillot (2018). PPS results in elevated (>1) d_N/d_S values throughout the phylogeny, which is not well modeled by codeml's branch-site model A, which assumes that positive selection only occurs on foreground branches. codeml has to choose between two alternatives, none of which fits the data very well: either consider that PPS sites never have $d_N/d_S > 1$, or consider that PPS sites have $d_N/d_S > 1$ on foreground branches only. The second alternative is closer to the truth, and therefore is chosen in a large number of cases, resulting in many false positives, and a low AUC. On the other hand, profile methods seem to suffer less from PPS, although it is harder to distinguish from episodic positive selection than purifying selection (supplementary fig. S14, Supplementary Material online). The effect of PPS can be compensated by increasing the effective population size N_e, which acts as a scaling factor for the intensity of selection: as a result, observed amino-acid frequencies are more representative of the actual fitness profile with higher values of N_e, and constitute a stronger signal for profile methods.

Interpreting Screens for Changes in Directional Selection

The methods we discussed in this paper can be used to detect sites in alignments whose selection regime has changed coincidentally to a punctual transition event. Such transitions are typically changes in the environment, for example, when a virus switches between hosts, and might also induce cases of convergent evolution (e.g., the multiple transitions of mammals to the marine environment Chikina et al. 2016). In this context, these models can be used to give insights into the relation between the genotype and a given binary phenotype (e.g., ancestral vs. convergent, marine vs. terrestrial, etc.), even though some methods can handle more than two categories of phenotypes (e.g., Pelican and Wertheim et al. 2015). The fact that all methods except Multinomial are conservative, that is, have low rates of false positives, indicates that the positives they output are likely to be worthy of further study.

The d_N/d_S and profile methods that we discuss in this manuscript all make similar assumptions. First, they can only handle a single phenotype or environmental condition at a time. This implicitly assumes that other phenotypes or conditions are unimportant for the evolution of the site under consideration. Such a strong assumption is likely to be incorrect in many cases: for instance a site may be important for several phenotypes, or its evolution may be more strongly associated to another phenotype or condition that has not been tested. Second, they assume that the evolution of the phenotype is known without uncertainty. d_N/d_S approaches that can handle uncertainty in the evolution of the phenotype when reconstructing the evolution of gene sequences have recently been proposed, but remain to be extended to the site level (Halabi et al. 2021). Third, they rely on the comparison of two scenarios, one of which assumes homogeneity of the process across the phylogeny. In the d_N/d_S method we consider, this means that the same d_N/d_S parameter applies to the site throughout the phylogeny. In the profile methods we consider, this means that the same profile applies to the site throughout the phylogeny. This is likely to be incorrect: the site may be evolving inhomogeneously because of non-adaptive processes (e.g., CpG hypermutability or gBGC), or because it is correlated to unaccounted-for phenotypes or conditions. The use of homogeneous null scenarios can result in model confounding whereby an incorrect model is chosen in the absence of the true generating model (Jones et al. 2019). This is what occurred in the gBGC simulations where the gBGC model generated data that were better fitted under our H_a model than under our homogeneous H_0 model. However, our simulations of PPS show that profile methods are robust to this particular confounding process.

Our results show that a site found as positive with a profile method could result from a change in the direction (fig. 4) or intensity (fig. 5) of selection, as well as from a change in gBGC or PPS (fig. 6). At this stage, distinguishing between these processes requires looking at the profiles estimated by the method at the site. These profiles have been shown to be inferred accurately by several mutsel models (Spielman and Wilke 2016). Since codon-based methods do not perform better than amino-acid-based methods in our hands, we suspect that the latter should also infer accurate profiles, although this will have to be verified in a future study. Given accurate profiles, one could distinguish between the different processes. Relaxation (respectively intensification) of selection should result in a flatter (resp. more heterogeneous) profile (fig. 1), which could be detected by computing its entropy and comparing it to the entropy of the other profiles at the site. gBGC should result in a shift toward GC-rich amino acids. PPS should result in a high amino-acid diversity (large number of amino acids with nonzero frequencies).

Looking Forward

The profile methods presented here have all been evaluated in the same setting, where the evolution of a site depends on two conditions that have been assigned to branches of a phylogeny. Not all phenotypes or conditions of interest can be known without uncertainty along a phylogeny, or can be accurately described by such a binary classification. Pelican can handle more than two conditions, but does not handle continuous annotations along a phylogeny, or uncertainty in the extant or ancestral states. Such extensions would be very useful. Similarly, the results show that accounting for gBGC in profile methods could be important. This could be done in codon models by following the approach that Guéguen and Duret (2018) used in d_N/d_S models.

Tamuri et al. (2014) and Spielman and Wilke (2016) showed that a penalized version of mutsel models performed better than the unpenalized version. We suspect that Pelican might also perform better with similar penalties. However, the use of penalized likelihoods would prevent us from relying on LRTs to compute *p*-values and detect positive sites. Instead, Tamuri (2021) relied on simulations to compute *p*-values, which is more resource intensive and would compromise Pelican's scalability. More work is needed to investigate the benefits of using penalization in Pelican, and, if any, come up with a fast method to compute *p*-values or scores. Such a method might also improve on the LRT that we have used here, as we saw that tinkering with its degrees of freedom improved the performance of the method in some cases (supplementary fig. S15, Supplementary Material online). It would not however change the ML estimate of a 0 fitness value for unseen amino acids, which is unrealistic (Rodrigue 2013). It will be important to develop a method that yields more realistic amino-acid-specific parameter values but remains fast enough for use at the genome scale.

Overall, the results show that profile methods constitute a solid alternative to d_N/d_S methods to screen for substitutions associated to changes in a phenotype or condition of interest. This opens new possibilities to better understand the link between a substitution, the structure of the protein where it occurs, and the phenotype or condition to which it is correlated. The amino-acid profiles inferred by a profile method at a site can be used to investigate the effect that having a high-fitness or a low-fitness amino acid has on a protein structure, in a particular condition. Profile methods could thus pair very well with the recent improvements in protein structure prediction (Jumper et al. 2021) to yield new insights into the molecular basis of adaptation.

Conclusion

In this paper, we evaluated on simulations a series of methods aiming to detect changes in selective pressures in coding sequences along a phylogeny. We found that some profile methods compare favorably to a commonly used d_N/d_S method, both in terms of power and in terms of speed, including in the presence of confounding factors. In particular, profile methods can readily distinguish changes in directional selection from PPS, something that the d_N/d_S method we tested cannot do. Among profile methods, we found that Pelican, a method operating at the amino-acid level, can be used to detect selective pressure changes efficiently. This makes genome-wide searches for sites correlating with a phenotype or condition of interest doable on a single computer within a few days.

Further extensions of Pelican are envisioned, for example, to handle continuous phenotypes. Integrating the effect of gBGC in the model would also be a major improvement, as we have found that it has a strong confounding effect on the detection of selection.

Methods

Detection of ω Variations using `codeml`

We used the `codeml` tool from the PAML package to detect variations of d_N/d_S as a proxy for variations of selective pressure, as was done in Thiltgen and Goldstein (2017). Branch lengths were re-estimated by `codeml`. `codeml` was configured to use the branch-site model A (Yang 2005; Zhang 2005). This model assumes there are three categories of sites in the alignment, whose proportions are estimated. Categories `0` and `1` have a homogeneous ω value throughout the phylogeny. Category `2` has one ω value estimated per branch condition: on background branches, the ω is between 0 and 1 (subcategory `2a`), characteristic of purifying selection, or at 1 (subcategory `2b`), characteristic of neutral evolution. On foreground branches, $\omega \geq 1$, characteristic of neutral or positive selection. A site is declared "positive" if it belongs to this category `2`. The probability for each site to be positive as inferred by the method was computed from the Bayes empirical Bayes probabilities, resulting from running `codeml` with parameter `fix_omega = 0` and summing up the probabilities to belong to categories `2a` and `2b` in the model.

Multinomial Method

The multinomial method models each site of an alignment as a collection of independent categorical variables, thus completely ignoring the phylogeny. It compares two models using a LRT, the first one assumes a single probability vector of length 20 (one frequency per each amino acid), the second a pair of vectors, one for each condition. Computing a *p*-value is however difficult in our setting, as at a given site, most of the amino acids are not observed and as a consequence their frequency estimated by maximum likelihood is zero, and thus lies at the boundary of the parameter space. In that case the usual convergence of the likelihood log-ratio to a χ^2 distribution known as Wilks theorem does not hold. Although there exists literature on the subject (see Mitchell et al. 2019 for a recent result), existing results are difficult to apply. We reused a heuristic we found in Tamuri et al. (2009), consisting in approximating the likelihood log-ratio distribution under the null by a χ^2 distribution with the number of degrees of freedom equal to the number of amino acids observed at the leaves of the tree minus one.

Pelican: Improvements on TDG09

Pelican is a reimplementation of the TDG09 method, originally published by Tamuri et al. (2009). TDG09 relies on a site-independent model of amino acid sequence evolution and the WAG exchangeability matrix. The model involves two kinds of parameters: stationary distributions of amino acids and branch scale.

Inference of selective pressure shifts is based on the postulate that stationary distributions of amino acids reflect the fitness profile in a condition (e.g., foreground or background). In a similar way to the multinomial method, the likelihoods of two models are compared using the LRT procedure, where one model assumes a single stationary distribution of amino acids shared between both conditions, and the other model assumes a specific stationary distribution per condition.

Parameters of the model, such as stationary distributions and branch scale, are optimized to maximum likelihood using the Nelder–Mead algorithm (Nelder and Mead 1965). We implemented an alternative approach using automatic differentiation, made available through the PyTorch library (Paszke et al. 2019), that converges to the same optima as the Nelder–Mead implementation. This alternative optimization algorithm is currently not used, but might be useful in future extensions of the method.

Pelican is implemented in the OCaml language (Leroy et al. 2021). The underlying mutation-selection model implementation takes advantage of LAPACK (Anderson et al. 1999) bindings for fast linear algebra computation, and optimizations for transition matrices exponentiation through diagonalization (Yang 2006). Pelican is available at https://gitlab.in2p3.fr/phoogle/pelican.

Simulations

In all our experiments, simulations were used to generate amino-acid or codon alignments with a constant number of sites $N = 10,000$. The simulator was configured to generate 90% of H_0 sites (no changes in selective pressure) and 10% of H_A sites (different selective pressure between background and foreground conditions). Simulations were done using a general time-reversible (GTR) mutation-selection model at the codon level. The model allows for two different regimes: one modeling selection in the background condition, and the other in the foreground condition. Selective pressure changes on H_A sites are simulated using either the foreground or background regime, depending on the condition of each branch in the phylogenetic tree. H_0 sites are generated using only the background regime, indicating no change in the selective pressure through the tree for these sites. Each regime is represented as a matrix of substitution rates between codons, which can be run along the phylogeny using Gillespie's algorithm (Gillespie 1976).

The substitution rates are the result of a mutation probability and a relative fixation probability, which depends on a selection coefficient associated with the transition to the mutated state. Mutation probabilities for the GTR model of nucleotide substitutions are based on exchangeabilities drawn from a *Gamma*(1, 1) distribution, and equilibrium frequencies from a *Dirichlet*(10, 10, 10, 10) distribution, and are shared across sites. The selection coefficient S (eq. 1) is defined as the difference in fitness between the ancestral state X and the mutated state Y in a condition c.

The relative fixation probability $u(S)$ for a mutation is computed from the selection coefficient S as per (Kimura 1983):

$$S^c_{X\to Y} = fitness(X, c) - fitness(Y, c), \tag{1}$$

$$u(S) = \frac{S^c_{X\to Y}}{1 - e^{-S^c_{X\to Y}}}. \tag{2}$$

Fitness values are determined from amino-acid frequency profiles, which are randomly picked at each site from a set of 263 preset profiles (Rey et al. 2019) for each condition. These frequency profiles are transformed into fitness profiles by multiplying them by a factor $Ne = 4$. As a result, values $S^c_{X\to Y}$ are between −4 and 4.

Codon substitution rates σ are the product of mutation rates μ and the relative probability of fixation:

$$\sigma^c_{X\to Y} = \mu_{X\to Y} \times u(S). \tag{3}$$

gBGC Simulation

gBGC acts as a fixed increase in fixation probability for mutations from either A or C nucleotides to G or C; conversely it is modeled as a probability decrease when mutating the other way around. We included GC-biased conversion in our simulation model as a bias term in the selection coefficient S:

$$S^c_{X\to Y} = B_{GC}(X, Y) + fitness(X, c) - fitness(Y, c). \tag{4}$$

Based on Glémin et al. (2015), we chose an intensity of $B_{GC} = 10$, that is applied on foreground branches, which is a strong effect for this process. Transition rates were not affected on background branches. This way, in H_A sites, the change in selective pressure between background and foreground branches that have to be detected is driven both by the shifted fitness profile, and the effect of gBGC. In H_0 sites, gBGC affects foreground branches.

CpG Simulation

CpG hypermutability is introduced in the simulation model as a scaling factor ρ for the mutation probability:

$$\sigma^c_{WXZ\to WYZ} = \mu_{X\to Y} \times \rho(W, X, Y, Z) \times u(S), \tag{5}$$

where W and Z are the states at the surrounding sites. This context is necessary because CpG dinucleotides can occur across two codons. As a consequence, the evolution of a whole sequence is not site-independent anymore, which led us to develop a dedicated Gillespie simulator. CpG hypermutability only occurs on methylated CpG dinucleotides and induces an increased probability of mutation from C to T in this context (or G to A on the reverse strand). We assume that any CpG dinucleotide in our simulation is methylated. If the conditions for hypermutability are not verified when comparing changes from X to Y, or the current branch is background, $\rho(W, X, Y, Z) = 1$ and has no effect. Otherwise, on foreground branches, we set $\rho(W, X, Y, Z) = 10$ based on Meunier et al. (2005), both on H_A and H_0 sites.

Simulation of PPS

PPS is introduced in the simulation model as a constant increasing the fitness of all other amino acids except the current one (Tamuri 2021). This is achieved by modifying equation 1 as

$$S^{c}_{X\rightarrow Y} = fitness(X, c) - fitness(Y, c) + PPS, \tag{6}$$

where $PPS \geq 0$ is a constant and describes the strength of positive selection. To simulate data, we relied on two-parameter settings. In the first setting, we simulate sequences under a mild selection strength, setting $Ne = 4$ and $PPS = 2$. This setting ensures that differences in amino acid fitnesses are between −4 and 4, as in the rest of the manuscript. In the second setting, we simulate under a strong selection regime, with $Ne = 10$ (i.e., differences in amino-acid fitnesses between −10 and 10), and $PPS = 10$. This second setting resembles parameter values observed on the sites showing the strongest positive selection in Tamuri (2021) and is also similar to their own simulation settings. H_A sites were simulated with different profiles for background and foreground branches, and H_0 sites were simulated with PPS running both on background and foreground branches.

Supplementary Material

Supplementary data are available at *Molecular Biology and Evolution online.*

Acknowledgments

We thank Nicolas Lartillot and Anamaria Necşulea for their comments on an early version of the manuscript and fruitful discussions. The research presented here was funded in part by the Convergenomix project (ANR-15-CE32-0005). L.D. was supported by a PhD fellowship from the Université Lyon 1 Claude Bernard.

Data Availability

Source code to reproduce the results in this paper is publicly available at https://gitlab.in2p3.fr/phoogle/spcd-benchmark. The implementation of Pelican is also made available at https://gitlab.in2p3.fr/phoogle/pelican. Plots were produced in R (R Core Team 2021) using the packages *ggplot2* (Wickham 2016) and *ggtree* (Yu 2020).

References

Abhiman S, Sonnhammer ELL. 2005. Large-scale prediction of function shift in protein families with a focus on enzymatic function. *Proteins*. **60**(4):758–768.

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J, Du Croz J, Greenbaum A, Hammarling S, McKenney A, *et al.* 1999. *LAPACK users' guide*. 3rd ed. Philadelphia (PA): Society for Industrial and Applied Mathematics.

Besnard G, Muasya AM, Russier F, Roalson EH, Salamin N, Christin P-A. 2009. Phylogenomics of C4 photosynthesis in sedges (Cyperaceae): multiple appearances and genetic convergence. *Mol Biol Evol*. **26**(8):1909–1919.

Bloom JD. 2014. An experimentally informed evolutionary model improves phylogenetic fit to divergent lactamase homologs. *Mol Biol Evol*. **31**(10):2753–2769.

Bloom JD. 2017. Identification of positive selection in genes is greatly improved by using experimentally informed site-specific models. *Biol Direct*. **12**(1):1.

Bolívar P, Guéguen L, Duret L, Ellegren H, Mugal CF. 2019. GC-biased gene conversion conceals the prediction of the nearly neutral theory in avian genomes. *Genome Biol*. **20**(1):5.

Boyd K, Eng KH, Page CD. 2013. Area under the precision–recall curve: point estimates and confidence intervals. In: Hutchison D, Kanade T, Kittler J, Kleinberg JM, Mattern F, Mitchell JC, Naor M, Nierstrasz O, Pandu Rangan C, Steffen B, Sudan M, Terzopoulos D, Tygar D, Vardi MY, Weikum G, Salinesi C, Norrie MC, and Pastor O, editors. *Advanced information systems engineering*, Lecture Notes in Computer Science, volume 7908. Berlin (HL): Springer. p. 451–466.

Bridgham JT, Carroll SM, Thornton JW. 2006. Evolution of hormone-receptor complexity by molecular exploitation. *Science*. **312**(5770):97–101.

Chikina M, Robinson JD, Clark NL. 2016. Hundreds of genes experienced convergent shifts in selective pressure in marine mammals. *Mol Biol Evol*. **33**(9):2182–2192.

Christin P-A, Salamin N, Savolainen V, Duvall MR, Besnard G. 2007. C4 Photosynthesis evolved in grasses via parallel adaptive genetic changes. *Curr Biol*. **17**(14):1241–1247.

Davydov II, Salamin N, Robinson-Rechavi M. 2019. Large-scale comparative analysis of codon models accounting for protein and nucleotide selection. *Mol Biol Evol*. **36**(6):1316–1332.

dos Reis M. 2015. How to calculate the non-synonymous to synonymous rate ratio of protein-coding genes under the Fisher–Wright mutation–selection framework. *Biol Lett*. **11**(4): 20141031.

Dutheil JY, Galtier N, Romiguier J, Douzery EJ, Ranwez V, Boussau B. 2012. Efficient selection of branch-specific models of sequence evolution. *Mol Biol Evol*. **29**(7):1861–1874.

Felsenstein J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J Mol Evol*. **17**(6):368–376.

Gillespie DT. 1976. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J Comput Phys*. **22**(4):403–434.

Glémin S, Arndt PF, Messer PW, Petrov D, Galtier N, Duret L. 2015. Quantification of GC-biased gene conversion in the human genome. *Genome Res*. **25**(8):1215–1228.

Goldman N, Yang Z. 1994. A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Mol Biol Evol*.

Gu X, Zou Y, Su Z, Huang W, Zhou Z, Arendsee Z, Zeng Y. 2013. An update of DIVERGE software for functional divergence analysis of protein family. *Mol Biol Evol*. **30**(7):1713–1719.

Guéguen L, Duret L. 2018. Unbiased estimate of synonymous and nonsynonymous substitution rates with nonstationary base composition. *Mol Biol Evol*. **35**(3):734–742.

Guindon S, Rodrigo AG, Dyer KA, Huelsenbeck JP. 2004. Modeling the site-specific variation of selection patterns along lineages. *Proc Natl Acad Sci USA*. **101**(35):12957–12962.

Halabi K, Karin EL, Guéguen L, Mayrose I. 2021. A codon model for associating phenotypic traits with altered selective patterns of sequence evolution. *Syst Biol*. **70**(3):608–622.

Halpern AL, Bruno WJ. 1998. Evolutionary distances for protein-coding sequences: modeling site-specific residue frequencies. *Mol Biol Evol*. **15**(7):910–917.

Ho AT, Hurst LD. 2022. Unusual mammalian usage of tga stop codons reveals that sequence conservation need not imply purifying selection. *PLoS Biol*. **20**(5):e3001588.

Jones CT, Susko E, Bielawski JP. 2019. Looking for Darwin in genomic sequences: validity and success depends on the relationship between model and data. In: Anisimova M, editor. *Evolutionary* *genomics: statistical and computational methods*, Methods in Molecular Biology. New York (NY): Springer. p. 399–426.

Jones CT, Youssef N, Susko E, Bielawski JP. 2017. Shifting balance on a static mutation–selection landscape: a novel scenario of positive selection. *Mol Biol Evol*. **34**(2):391–407.

Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A, *et al*. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature*. **596**(7873):583–589.

Kapralov MV, Smith JAC, Filatov DA. 2012. Rubisco evolution in C4 Eudicots: an analysis of amaranthaceae Sensu Lato. *PLoS ONE*. **7-**(12):e52974.

Kimura M. 1983. *The neutral theory of molecular evolution*. Cambridge: Cambridge University Press.

Korber B, Fischer WM, Gnanakaran S, Yoon H, Theiler J, Abfalterer W, Hengartner N, Giorgi EE, Bhattacharya T, Foley B, *et al*. 2020. Tracking changes in SARS-CoV-2 Spike: evidence that D614G increases infectivity of the COVID-19 virus. *Cell*. **182**(4): 812–827.e19.

Kosakovsky Pond SL, Murrell B, Fourment M, Frost SDW, Delport W, Scheffler K. 2011. A random effects branch-site model for detecting episodic diversifying selection. *Mol Biol Evol*. **28**(11):3033–3043.

Kosiol C, Anisimova M. 2019. Selection acting on genomes. In: Anisimova M, editor. *Evolutionary genomics: statistical and computational methods*. New York (NY): Springer. p. 373–397.

Kosiol C, Vinar T, da Fonseca RR, Hubisz MJ, Bustamante CD, Nielsen R, Siepel A. 2008. Patterns of positive selection in six Mammalian genomes. *PLoS Genet*. **4**(8):e1000144.

Leroy X, Doligez D, Frisch A, Garrigue J, Rémy D, Vouillon J. 2021. The ocaml system: documentation and user's manual. *INRIA*. **3**:42.

Li Y, Liu Z, Shi P, Zhang J. 2010. The hearing gene Prestin unites echolocating bats and whales. *Curr Biol*. **20**(2):R55–R56.

Marcovitz A, Turakhia Y, Chen HI, Gloudemans M, Braun BA, Wang H, Bejerano G. 2019. A functional enrichment test for molecular convergent evolution finds a clear protein-coding signal in echolocating bats and whales. *Proc Natl Acad Sci USA*. **116**(42): 21094–21103.

Merényi Z, Prasanna AN, Wang Z, Kovács K, Hegedüs B, Bálint B, Papp B, Townsend JP, Nagy LG. 2020. Unmatched level of molecular convergence among deeply divergent complex multicellular fungi. *Mol Biol Evol*. **37**(8):2228–2240.

Meunier J, Khelifi A, Navratil V, Duret L. 2005. Homology-dependent methylation in primate repetitive DNA. *Proc Natl Acad Sci USA*. **102**(15):5471–5476.

Mitchell JD, Allman ES, Rhodes JA. 2019. Hypothesis testing near singularities and boundaries. *Electron J Stat*. **13**(1):2150.

Moretti S, Laurenczy B, Gharib WH, Castella B, Kuzniar A, Schabauer H, Studer RA, Valle M, Salamin N, Stockinger H, *et al*. 2014. Selectome update: quality control and computational improvements to a database of positive selection. *Nucleic Acids Res*. **42-**(D1):D917–D921.

Murrell B, de Oliveira T, Seebregts C, Kosakovsky Pond SL, Scheffler K, On behalf of the Southern African Treatment and Resistance Network (SATuRN) Consortium. 2012. Modeling HIV-1 drug resistance as episodic directional selection. *PLoS Comput Biol*. **8**(5): e1002507.

Murrell B, Weaver S, Smith MD, Wertheim JO, Murrell S, Aylward A, Eren K, Pollner T, Martin DP, Smith DM, *et al*. 2015. Gene-wide identification of episodic selection. *Mol Biol Evol*. **32**(5): 1365–1371.

Nelder JA, Mead R. 1965. A simplex method for function minimization. *Comput J*. **7**(4):308–313.

Nielsen R, Bustamante C, Clark AG, Glanowski S, Sackton TB, Hubisz MJ, Fledel-Alon A, Tanenbaum DM, Civello D, White TJ, *et al*. 2005. A scan for positively selected genes in the genomes of humans and chimpanzees. *PLoS Biol*. **3**(6):e170.

Partha R, Kowalczyk A, Clark NL, Chikina M. 2019. Robust method for detecting convergent shifts in evolutionary rates. *Mol Biol Evol*. **36**(8):1817–1830.

Parto S, Lartillot N. 2018. Molecular adaptation in Rubisco: discriminating between convergent evolution and positive selection using mechanistic and classical codon models. *PLoS ONE*. **13**(2):e0192697.

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, *et al*. 2019. Pytorch: an imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché Buc F, Fox E, and Garnett R, editors. *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc. p. 8024–8035.

Penn O, Stern A, Rubinstein ND, Dutheil J, Bacharach E, Galtier N, Pupko T. 2008. Evolutionary modeling of rate shifts reveals specificity determinants in HIV-1 subtypes. *PLoS Comput Biol*. **4**(11): e1000214.

Pouyet F, Bailly-Bechet M, Mouchiroud D, Guéguen L. 2016. SENCA: a multilayered codon model to study the origins and dynamics of codon usage. *Genome Biol Evol*. **8**(8):2427–2441.

Pupko T, Galtier N. 2002. A covarion-based method for detecting molecular adaptation: application to the evolution of primate mitochondrial genomes. *Proc Biol Sci*. **269**(1498):1313–1316.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Ratnakumar A, Mousset S, Glémin S, Berglund J, Galtier N, Duret L, Webster MT. 2010. Detecting positive selection within genomes: the problem of biased gene conversion. *Philos Trans R Soc B Biol Sci*. **365**(1552):2571–2580.

Rey C, Guéguen L, Sémon M, Boussau B. 2018. Accurate detection of convergent amino-acid evolution with PCOC. *Mol Biol Evol*. **35**(9):2296–2306.

Rey C, Lanore V, Veber P, Guéguen L, Lartillot N, Sémon M, Boussau B. 2019. Detecting adaptive convergent amino acid evolution. *Philos Trans R Soc B Biol Sci*. **374**(1777):20180234.

Risso VA, Martinez-Rodriguez S, Candel AM, Krüger DM, Pantoja-Uceda D, Ortega-Muñoz M, Santoyo-Gonzalez F, Gaucher EA, Kamerlin SCL, Bruix M, *et al*. 2017. De novo active sites for resurrected Precambrian enzymes. *Nat Commun*. **8**: 16113.

Rodrigue N. 2013. On the statistical interpretation of site-specific variables in phylogeny-based substitution models. *Genetics*. **193**(2):557–564.

Rodrigue N, Lartillot N. 2017. Detecting adaptation in protein-coding genes using a Bayesian site-heterogeneous mutation-selection codon substitution model. *Mol Biol Evol*. **34**(1): 204–214.

Rodrigue N, Latrille T, Lartillot N. 2020. A Bayesian mutation–selection framework for detecting site-specific adaptive evolution in protein-coding genes. *Mol Biol Evol*. **38**(3):1199–1208.

Rodrigue N, Philippe H, Lartillot N. 2010. Mutation-selection models of coding sequence evolution with site-heterogeneous amino acid fitness profiles. *Proc Natl Acad Sci USA*. **107**(10): 4629–4634.

Rousselle M, Laverré A, Figuet E, Nabholz B, Galtier N. 2019. Influence of recombination and GC-biased gene conversion on the adaptive and nonadaptive substitution rate in mammals versus birds. *Mol Biol Evol*. **36**(3):458–471.

Rubinstein ND, Doron-Faigenboim A, Mayrose I, Pupko T. 2011. Evolutionary models accounting for layers of selection in protein-coding genes and their impact on the inference of positive selection. *Mol Biol Evol*. **28**(12):3297–3308.

Saunders CT, Green P. 2007. Insights from modeling protein evolution with context-dependent mutation and asymmetric amino acid selection. *Mol Biol Evol*. **24**(12):2632–2647.

Scornavacca C, Belkhir K, Lopez J, Dernat R, Delsuc F, Douzery EJP, Ranwez V. 2019. OrthoMaM v10: scaling-up orthologous coding sequence and exon alignments with more than one hundred mammalian genomes. *Mol Biol Evol*. **36**(4):861–862.

Spielman SJ, Wilke CO. 2015. The relationship between dN/dS and scaled selection coefficients. *Mol Biol Evol*. **32**(4):1097–1108.

Spielman SJ, Wilke CO. 2016. Extensively parameterized mutation–selection models reliably capture site-specific selective constraint. *Mol Biol Evol*. **33**(11):2990–3002.

Studer RA, Penel S, Duret L, Robinson-Rechavi M. 2008. Pervasive positive selection on duplicated and nonduplicated vertebrate protein coding genes. *Genome Res*. **18**(9):1393–1402.

Suzuki Y, Gojobori T, Kumar S. 2009. Methods for incorporating the hypermutability of CpG dinucleotides in detecting natural selection operating at the amino acid sequence level. *Mol Biol Evol*. **26**(10):2275–2284.

Tamuri AU. 2021. A mutation-selection model of protein evolution under persistent positive selection. preprint, Evolutionary Biology.

Tamuri AU, Goldstein RA. 2012. Estimating the distribution of selection coefficients from phylogenetic data using sitewise mutation-selection models. *Genetics*. **190**(3):1101–1115.

Tamuri AU, Goldman N, Reis MD. 2014. A penalized likelihood method for estimating the distribution of selection coefficients from phylogenetic data. *Genetics*. **197**:257–271.

Tamuri AU, Hay AJ, Goldstein RA. 2009. Identifying changes in selective constraints: host shifts in influenza. *PLoS Comput Biol*. **5**(11): e1000564.

Thiltgen G, Goldstein RA. 2017. Finding direction in the search for selection. *J Mol Evol*. **84**(1):39–50.

Wertheim JO, Murrell B, Smith MD, Kosakovsky Pond SL, Scheffler K. 2015. RELAX: detecting relaxed selection in a phylogenetic framework. *Mol Biol Evol*. **32**(3):820–832.

Wickham H. 2016. *ggplot2: elegant graphics for data analysis*. New York: Springer-Verlag.

Yang Z. 2005. Bayes empirical Bayes inference of amino acid sites under positive selection. *Mol Biol Evol*. **22**(4):1107–1118.

Yang Z. 2006. *Computational molecular evolution*. Oxford University Press. p. 78–79.

Yang Z. 2007. PAML 4: phylogenetic analysis by maximum likelihood. *Mol Biol Evol*. **24**(8):1586–1591.

Yang Z, Nielsen R. 2008. Mutation-selection models of codon substitution and their use to estimate selective strengths on codon usage. *Mol Biol Evol*. **25**(3):568–579.

Yu G. 2020. Using ggtree to visualize data on tree-like structures. *Curr Protoc Bioinform*. **69**(1):e96.

Zhang G, Li C, Li Q, Li B, Larkin DM, Lee C, Storz JF, Antunes A, Greenwold MJ, Meredith RW, *et al.* 2014. Comparative genomics reveals insights into avian genome evolution and adaptation. *Science*. **346**(6215):1311–1320.

Zhang J. 2005. Evaluation of an improved branch-site likelihood method for detecting positive selection at the molecular level. *Mol Biol Evol*. **22**(12):2472–2479.

Zhen Y, Aardema ML, Medina EM, Schumer M, Andolfatto P. 2012. Parallel molecular evolution in an herbivore community. *Science*. **337**(6102):1634–1637.

Zhou X, Stephens M. 2012. Genome-wide efficient mixed-model analysis for association studies. *Nat Genet*. **44**(7): 821–824.