Limited utility of residue masking for positive-selection inference

Stephanie J. Spielman[1*] and Eric T. Dawson[1] and Claus O. Wilke[1]

Address:

[1]Department of Integrative Biology, Center for Computational Biology and Bioinformatics, and Institute of Cellular and Molecular Biology. The University of Texas at Austin, Austin, TX 78731, USA.

[*]Corresponding author

Email: stephanie.spielman@utexas.edu

Manuscript type: research article

Keywords: multiple sequence alignment, alignment filters, positive-selection inference, sequence simulation

Abstract

Poor-quality multiple sequence alignments have the potential to reduce accuracy in positive-selection inference. It has therefore been suggested that alignments be filtered before further analyses. One widely-used filter, Guidance, generates position-based confidence scores for a given alignment using a bootstrap approach, thereby allowing users to remove positions of low confidence from the alignment. Recent studies on the utility of the Guidance filter for positive-selection inference have yielded inconsistent results; some have demonstrated that Guidance substantially improved accuracy, but others have found that Guidance neither improved nor worsened selection inference. Motivated by these discrepancies, we have conducted a extensive simulation-based study to elucidate Guidance's impact on positive-selection inference for realistic protein-coding sequences. In particular, we investigated whether novel Guidance-based scoring algorithms, which phylogenetically correct confidence scores, and a new gap-penalization score-normalization scheme could improve upon the original Guidance's performance. We found that no filter, including the original Guidance, substantially improved positive-selection inference relative to unfiltered alignments across multiple inference methods, including FUBAR and PAML. Instead, the analysis method used influenced positive-selection inference far more than did filtering alignments with a Guidance-based strategy.

Introduction

Constructing a multiple sequence alignment represents the first step of analysis in most studies of molecular evolution, namely phylogenetic reconstruction and evolutionary rate inference. Recently, several studies have shown that poor alignment quality can significantly hinder accuracy in such downstream analyses (Ogden and Rosenberg 2006; Talavera and Castresana 2007; Dwivedi and Gadagkar 2009; Markova-Raina and Petrov 2011; Jordan and Goldman 2012). In particular, using low-quality alignments to infer positive selection elevated false positive rates (Schneider et al. 2009; Fletcher and Yang 2010; Markova-Raina and Petrov 2011). As a consequence, some studies have advocated that users filter alignment before subsequent analyses (Jordan and Goldman 2012; Privman et al. 2012). Such filters, which include T-Coffee (Notredame et al. 2000), GBLOCKS (Castresana 2000), and Guidance (Penn et al. 2010; Privman et al. 2012) locate putatively poorly-aligned regions in alignments so that users can curate their alignments to maximize signal. The hope is that

culling unreliable regions from alignments will yield increased accuracy in positive-selection inference without excessively sacrificing power.

Of these filters, Guidance (Penn et al. 2010; Privman et al. 2012) is currently the most widely used in positive-selection inference. Guidance derives a confidence score for each alignment position by sampling variants in the guide tree used to construct progressive alignments. Users can mask positions with scores below a given threshold, thereby removing residues that cannot be confidently placed in the alignment. This method is fairly conservative, as particular positions of low confidence can be removed rather than entire columns.

Recently, Privman et al. (2012) found that applying the Guidance filter to alignments dramatically improved positive-selection inference. They described, for instance, that Guidance could reduce false positive rates from 90% to well below 10%. However, a comprehensive study by Jordan and Goldman (2012) on alignment methods and filtering found that Guidance affected positive-selection inference modestly, if at all. Jordan and Goldman (2012) noted that, while applying the Guidance filter did not substantially reduce power in positive-selection inference, neither did filtering offer any marked improvements under most conditions. Both studies reported that filtering with Guidance was most beneficial at high insertion/deletion (indel) rates or protein divergence levels. In particular, Privman et al. (2012) and Jordan and Goldman (2012) concluded, respectively, that Guidance improved inference most substantially at indel rates (defined as ratio of indels to substitutions) above 0.1 and at divergence levels of 1.8 mean-path-length (defined as mean root-to-tip branch length). However, it is important to recognize that protein sequences used in positive selection studies rarely, if ever, contain sequences separated by such high divergences; for instance, a typical mammalian gene tree's mean-path-length is only about 0.17 (Spielman and Wilke 2013), less than 10% the divergence level at which Jordan and Goldman (2012) detected improvements when using the Guidance filter.

In sum, previous studies on the benefits of filtering alignments with Guidance before inferring positive selection gave distinct recommendations; while Privman et al. (2012) advocated the filter's use, Jordan and Goldman (2012) primarily promoted using high-quality alignment software rather than relying on alignment filtering. To reconcile these different findings, we have conducted an extensive simulation-based study to fully elucidate how the Guidance filter influences positive-selection inference. In particular, we examined the poten-

tial benefits to modifying the Guidance scoring scheme in several ways. First, we assessed whether novel algorithms that correct Guidance scores for the sequences' phylogenetic relationships could improve upon the original Guidance algorithm. Second, we tested a new score-normalization scheme which scaled residue scores according to the number of gaps in its column; this strategy naturally assigned lower scores to residues in "gappy" columns, thereby capturing the inherent unreliability of such regions. To examine the influence of these novel algorithms, we reimplemented the Guidance software (available at **github**).

Overall, we found that neither the original Guidance filter nor our newly implemented filters conferred any substantial benefits to positive-selection inference. In cases in which filtering with a Guidance-based strategy did benefit inference, improvements recovered were of extremely small magnitude. Instead, the inference method used had a much greater effect on positive-selection detection. **May add something about the influence of masking cutoffs pending PAML results.**

Results

Guidance reimplementation and analysis pipeline

To systematically evaluate Guidance's influence on positive-selection inference, we reimplemented the Guidance software in Python and C++. Before conducting any analyses, we carefully verified that, given a set of perturbed alignments, our program produced the same confidence scores as did the original Guidance. This reimplementation was necessary for us to investigate the utility of several novel scoring schemes, described in detail in Methods. These new methods included two scoring algorithms which incorporated phylogenetic weights to assign confidence scores. Briefly, the first method incorporated a weight for each sequence in the alignment, as calculated by BranchManager (Stone and Sidow 2007), and the second method incorporated patristic distances (the sum of branch lengths between two taxa). We called these methods, respectively, BMweights and PDweights. Moreover, we defined a new "gap-penalization" score-normalization scheme, which assigned scores scaled by a column's gappiness. Specifically, this scheme naturally gave lower scores to highly-gapped columns, accounting for the fact that such regions were more likely to be poorly aligned. We referred to filters using the gap-penalization scheme as GuidanceP, BMweightsP, and PDweightsP.

We simulated 100 replicates of protein-coding sequences using Indelible (Fletcher and Yang 2009) along each of four different gene trees of sizes 11, 26, 60, and 158 taxa. To ensure that our simulations produced real sequence data to the extent possible, we simulated according to evolutionary parameters inferred from H1N1 influenza hemagluttinin (HA) (see Methods for details), a protein well known to contain positively selected regions (Bush et al. 1999; Kryazhimskiy et al. 2008; Meyer and Wilke 2012). We then processed the unaligned amino-acid sequences with our Guidance reimplementation using the aligner MAFFT L-INS-I (linsi) (Katoh et al. 2005). We chose to use only linsi because it strongly outperforms other progressive alignment softwares for protein alignments without sacrificing speed (Nuin et al. 2006; Thompson et al. 2011). Scores for all alignments were calculated using each of our three scoring algorithms (original Guidance, BMweights, and PDweights) with each of our two normalization schemes (original Guidance and gap-penalization). Unless otherwise specified, we masked positions with scores below 0.5, the same threshold as used by Jordan and Goldman (2012), with "?".

After back-translating these protein alignments to codon sequences using the original nucleotide data, we assessed positive selection with both the recently-described FUBAR (Murrell et al. 2013) as implemented in HyPhy (Kosakovsky Pond et al. 2005) and the widely-used M8 model in PAML (Yang 2007). Note that we use PAML throughout this paper to refer specifically to its M8 model. We inferred all phylogenies used for these inferences from the unfiltered amino-acid alignments in RAxML (Stamatakis 2006b) using the CAT model (Stamatakis 2006a) under the WAG substitution matrix (Whelan and Goldman 2001). All filtered alignments derived from the same unfiltered alignment were analyzed with the same phylogeny to remove any potential bias. We considered sites as positively selected if the given method returned a posterior probability ≥ 0.90. While we processed all alignments with FUBAR, we did not infer positive selection with PAML for the largest simulation set due to prohibitive runtimes. In total, we inferred positive selection for 5200 alignments with FUBAR and 3900 alignments with PAML.

Guidance-based filters have a minimal effect on positive-selection inference

We assessed how filtering alignments with a Guidance-based method influenced positive-selection detection by comparing the resulting true positive rates (TPRs) of positive-selection

inference between all filtered alignments and their corresponding unfiltered alignments. TPRs were calculated using the true evolutionary rates assigned during sequence simulation. We chose to primarily compare the TPRs as opposed to the false positive rates (FPRs), as the FPRs we recovered were exceedingly small, never exceeding an average of 0.017 across simulation sets and methods, similar to those recovered by Jordan and Goldman (2012).

We first compared performances between the two normalization schemes, original and gap-penalization, for each of the three scoring algorithms (Guidance, BMweights, and PDweights). For each simulation set and inference method, we fit a linear model, using the difference in TPR between normalization schemes as the response and algorithm as a fixed effect. Note that only filtered alignments were considered in this analysis, results from which are shown in Table 1. In all instances in which we detected a significant mean TPR difference between normalization schemes, the gap-penalization scheme outperformed the original Guidance scheme, albeit by very small magnitudes. Therefore, we considered only alignments filtered with the gap-penalization normalization in subsequent analyses.

We proceeded to compare the TPRs from gap-penalization algorithms (GuidanceP, BMweightsP, PDweightsP) to those of unfiltered alignments with a series of mixed-effects models. These models included TPR as the response variable, simulation count as a random effect (to account for the paired structure of our analysis), and algorithm as a fixed effect. Table 2 gives the results from these models for each simulation set and selection inference method.

Overall, we did not recover any significant difference in average TPR among gap-penalization filtering algorithms; the original Guidance filter and our two phylogenetically-weighted filters performed statistically indistinguishably for a given simulation set. When processed with FUBAR, alignment filtering did significantly increase the average TPR for the 60 and 158-sequence simulation sets. Even so, these improvements were of very small magnitudes; filtering increased the average unfiltered TPR by 0.03 for the 60-sequence set and 0.01 for the 158-sequence set. In other words, TPR increased on average by 10% for the 60-sequence simulation set and by 3% for the 158-sequence simulation set. Filtering neither improved nor worsened positive-selection inference for the 11 or 26-sequence simulation sets as analyzed with FUBAR.

Similarly, analyzing the data with PAML showed that alignment filtering did not significantly influence the mean TPR of the 11-sequence simulation set. Unlike when analyzing

with FUBAR, we did not recover significant change in mean TPR for the 60-sequence simulation set between filtered and unfiltered alignments. However, PAML results showed that filtering did improve TPRs in the 26-sequence simulation set, albeit by an average of only 4.68%. While statistically significant, an increase of this low a magnitude may not have any noticeable effect on positive-selection inference in studies of real sequences.

We used receiver operating characteristic (ROC) curves to qualitatively assess differences in positive-selection inference for unfiltered versus filtered alignments. Commonly used to evaluate the performance of binary classification methods, ROC illustrate the relationship between TPR and FPR. Classifier performances can be assessed by comparing area under the ROC curves, such that larger areas indicate higher power. Here, we used ROC curves to examine if alignment filtering helped FUBAR and PAML to classify sites more accurately as positively selected or not (Figure 1). Figure 1 displays ROC curves for positive-selection inference with FUBAR and PAML. Figures 1B and 1C specifically highlight the effect of alignment filtering with FUBAR and PAML, respectively, at low FPRs. For each respective method, areas under the unfiltered alignment curves were essentially the same as for the filtered alignment curves, implying that alignment filtering with a Guidance-based strategy does not dramatically increase power in positive-selection inference.

Influence of posterior probability on selection inference methods

Our analyses incidentally recovered several differences between how PAML and FUBAR behaved when assessing positive selection. In particular, whether FUBAR or PAML performed better depended on the number of sequences analyzed; FUBAR outperformed PAML for the two smaller simulation sets of 11 and 26 sequences, but PAML outperformed FUBAR for the 60-sequence simulation set (Table 2). We found that this trend resulted from the posterior probability threshold chosen to call sites as positively selected. Figure 2 shows how the use of different posterior probabilities affected the mean TPR and FPR for unfiltered alignments from the 26- and 60-sequence simulation sets.

With regards to TPR, FUBAR generally outperformed PAML at low posterior probabilities, whereas PAML outperformed FUBAR at high probabilities. As the number of sequences increased, the intersection between methods' TPRs shifted towards lower posterior probabilities. This shift was largely dictated by PAML's improved ability to call true positives with the inclusion of more sequences, given that FUBAR's relationship be-

tween posterior probability and TPR remained similar between the two simulation sets. Filtering alignments affected these broad trends that FUBAR and PAML portrayed only marginally, if at all (Figure 3). These results demonstrated that the analysis method used affected positive-selection inference substantially more than did filtering alignments with a Guidance-based method.

The relationship between FPR and posterior probability was largely consistent between simulation sets for both FUBAR and PAML. While PAML achieved mean FPRs of nearly zero essentially immediately for both the 26 and 60-sequence simulation sets, FUBAR approached a zero mean FPR more slowly, reaching zero around a posterior probability of 0.8 for each simulation set. As a typical study of positive selection would almost certainly use a posterior probability above at least 0.8, alignment filtering would likely not be able to substantially reduce FPR.

How TPR and FPR behaved for FUBAR and PAML evidenced the different approaches these methods employ to detect positive selection. PAML's M8 model employs a random-effects likelihood (REL) method (Nielsen and Yang 1998) to fit a specific probabilistic model of sequence evolution. FUBAR, on the other hand, approximates results one would achieve with an REL method, allowing for remarkably fast runtimes without excessively sacrificing performance. Having fewer sequences in an alignment hindered PAML's ability to precisely fit its given model, rendering its positive-selection inference worse than that of FUBAR. Increasing the number of sequences allowed PAML to achieve a more accurate model fit, and in turn positive-selection inferences, but had a relatively smaller effect on FUBAR's approximations.

Raising masking thresholds for gap-penalization algorithms can hinder positive-selection inference

When filtering alignments with Guidance-based methods, one must select a specific score cutoff below which to mask residues. We chose to filter all residues with scores less than 0.5, as previously done by Jordan and Goldman (2012). However, it was possible that selecting a different threshold would have yielded different results, so we analyzed how changing this threshold might impact our findings. For this analysis, we considered only the Guidance and GuidanceP scoring schemes since our phylogenetically-corrected algorithms did not perform significantly differently. Using the same position confidence scores previously generated, we

masked all such alignments at the additional cutoffs of 0.3, 0.7, and 0.9 and inferred positive selection with FUBAR and PAML.

To investigate the influence of different masking cutoffs, we fit mixed-effects linear models for each simulation set, with TPR as the response, masking cutoff as a fixed effect, and simulation count as the random effect, for Guidance and GuidanceP results each. Results showed that, for Guidance, all masking cutoffs yielded statistically similar TPRs (multiple comparisons gave all $P > 0.06$) for a given simulation set. For GuidanceP, on the other hand, masking cutoffs 0.3, 0.5, and 0.7 gave statistically similar TPRs (all $P > 0.45$), whereas using a masking cutoff of 0.9 always yielded significantly lower TPRs than did the other cutoffs. Table 3 highlights this result, specifically comparing performance between cutoffs 0.5 and 0.9 for GuidanceP. Importantly, for the 60-sequence simulation set (for which we noted the maximum benefit of alignment filtering when analyzed with FUBAR), masking at the stringent cutoff of 0.9 as opposed to the more lenient 0.5 reduced the TPR by roughly 26%. This extreme TPR decrease yielded an average TPR of 0.281 for alignments masked at 0.9, well below its corresponding average unfiltered TPR of 0.345 (Table 2). When combined with a strict masking cutoff, the gap-penalization normalization scheme, therefore, had the potential to remove excessive amounts of information from alignments and worsen positive-selection detection.

Discussion

We have conducted an extensive simulation-based study to evaluate how introducing phylogenetically-aware scoring algorithms into the Guidance alignment filter influenced the detection of positive selection in protein-coding sequences. Additionally, we tested a novel gap-penalization score-normalization method, which scaled residue confidence scores according to the number of gaps in that residue's column. Using coding-sequence data simulated according to evolutionary parameters of the H1N1 influenza hemagluttinin (HA) protein, we inferred positive selection for all alignments using two methods: the recently introduced and very fast FUBAR (Murrell et al. 2013) within the HyPhy package (Kosakovsky Pond et al. 2005) and the current standard for evolutionary rate inference, the PAML M8 model (Yang 2007). Overall, conducted over 9000 positive-selection inferences.

We found that, while the gap-penalization scheme marginally improved upon the original

normalization method, incorporating phylogenetic information did not significantly impact positive-selection inference relative to the original Guidance algorithm. Moreover, filtering alignments with any Guidance-based method, including the original, did not substantially improve positive-selection inference under any circumstance. That the phylogenetically-weighted algorithms did not improve upon the original Guidance algorithm indicated the minimal benefits that filtering in this manner produced at all. Were the original Guidance to offer robust improvements in positive-selection detection, one might expect that our more statistically controlled approach would boost the method's performance. However, as we have found that masking individual positions in an alignment only marginally affected positive-selection inference in the first place, the algorithmic changes we implemented would not be expected to have a dramatic effect.

We ensured realism in our simulations by applying evolutionary parameters inferred from a very large (1038 sequences) HA alignment. We chose to use HA as a reference for evolutionary parameters for two reasons: the pervasive positive selection (Bush et al. 1999; Kryazhimskiy et al. 2008; Meyer and Wilke 2012) marking its evolutionary history and its wealth of sequence data. In particular, having such a large data set allowed for robust evolutionary inference and the assurance that our simulation parameters mimicked realistic sequence data. However, rather than conduct our simulations along our HA phylogeny, we decided to evolve sequences along eukaryotic gene trees for tractability. Simulating along a phylogeny of over 1000 sequences would not only represent an excessive computational burden, but also a typical positive-selection study would likely not use this many sequences. Additionally, the most important aspect to consider when selecting a phylogeny along which to simulate sequences is the phylogeny's topology, and the gene trees we used were topologically typical of an evolutionary rate study.

While others (Schneider et al. 2009; Fletcher and Yang 2010; Markova-Raina and Petrov 2011; Privman et al. 2012) have noted a prevalence of false positives in positive-selection inference, we recovered very low false positive rates all below an average of 0.017 for a given simulation set. The FPRs we detected were very similar to those of Jordan and Goldman (2012), yet substantially lower than those detected by Privman et al. (2012), the two studies which have previously investigated the utility of the Guidance filter in positive-selection inference. The high FPRs Privman et al. (2012) recovered likely resulted from the overly high indel rates with which they simulated sequences. Our focus on realistic indel rates

and divergence levels supported conclusions made by Jordan and Goldman (2012), namely that the Guidance filter does not **(present tense seems appropriate here)** substantially increase accuracy in positive-selection inference.

Both Privman et al. (2012) and Jordan and Goldman (2012) found that filtering alignments with Guidance yielded fewer benefits when alignments were built with a high-quality aligner, namely, PRANK (Loytynoja and Goldman 2008), as opposed to with other alignment softwares like ClustalW (Thompson et al. 1994) or MUSCLE (Edgar 2004). While PRANK has been shown to substantially reduce alignment errors (Loytynoja and Goldman 2008; Privman et al. 2012; Jordan and Goldman 2012), it carries a rather long runtime that may not be practical for some users. To circumvent this time constraint, we conducted all alignments in the present study with linsi, an algorithm released with the MAFFT software that is substantially more accurate than MAFFT's default algorithm (Katoh et al. 2005; Nuin et al. 2006; Thompson et al. 2011) but retains MAFFT's rapid runtime. Importantly, neither previous study investigated how alignment filtering affected alignments made with linsi. We have demonstrated that alignment filtering did not largely increase accuracy in positive-selection inference, similar to how alignment filtering impacted PRANK alignments. Linsi, therefore, may represent a fast and accurate alternative to PRANK.

Our study incidentally recovered some differences in how FUBAR and PAML's M8 model behave when inferring positive selection. Specifically, we noted distinct behaviors of posterior probability thresholds for each method. Posterior probabilities for FUBAR behaved largely as expected; as the threshold became more stringent, fewer positive results were detected. PAML, on the other hand, was fairly robust to the choice of posterior probability threshold; its FPR was nearly zero even at the lowest posterior probabilities, and increasing the threshold's stringency did not strongly affect the TPR. Thus, while using a higher posterior probability in FUBAR may protect against recovering false positives, choosing a higher posterior probability did not similarly affect PAML. Lower posterior probabilities may not severely hinder accuracy in PAML's M8 model.

Moreover, PAML outperformed FUBAR when processing larger sequence sets. However, this increase in performance was coupled with a dramatic increase in runtime. While a single PAML took roughly an hour for each 11-sequence alignment, it took anywhere from 2 days to a week to infer evolutionary rates for a 60-sequence alignment. FUBAR, on the other hand, consistently required under 20 minutes minutes to analyze an alignment of any size

and, while not as accurate as PAML for larger data sets, did perform fairly well. Users, therefore, should consider the trade-off between accuracy and runtime when selecting an inference software.

In sum, we have found that, while alignment filtering offered some benefits to positive-selection inference, those improvements were marginal at best. With such a minimal effect, alignment filtering could easily decrease accuracy in a given positive selection study. Indeed, we noted that using a stringent masking cutoff of 0.9 for algorithms normalized with our gap-penalization strategy resulted in extreme decreases in TPR relative to an unfiltered alignment. Choosing a low filtering threshold was necessary to achieve any improvement in positive-selection inference.

Overall, we cannot unequivocally recommend the use of a Guidance-based alignment filter when inferring positive selection. Once an alignment has been constructed, it does not seem that much can be done to eliminate any misleading information. Instead, users should employ inference methods in which the error can be minimized as much as possible without necessitating post-hoc correction. Therefore, we recommend that users select high-quality alignment and inference methods to minimize any obscuring signal, instead of relying on filters. If one must filter an alignment, we recommend using a lenient cutoff (≤ 0.5) to avoid sacrificing power, which might worsen inferences.

Methods

Guidance Reimplementation

Our reimplemented Guidance is written in Python and C++. Following the algorithm set forth in Penn et al. (Penn et al. 2010), we first create a reference alignment using a user-specified progressive alignment software, with choices of Clustalw (Thompson et al. 1994), MUSCLE (Edgar 2004), or MAFFT (Katoh et al. 2002, 2005). We then generate N (where $N = 100$, by default) bootstrapped alignment replicates, each of which is used to create a bootstrapped tree in FastTree2 (Price et al. 2010). We then use these N trees as guide trees to create N new perturbed alignments, which we subsequently compare to the reference alignment to generate a confidence score for each residue. Users can specify options for aligner and phylogeny as desired.

Scoring Algorithms

Before calculating confidence scores, a phylogeny is built from the reference alignment. Our program includes functionality to build this phylogeny using either FastTree2 (Price et al. 2010) or RAxML (Stamatakis 2006b). Two types of phylogenetic weights can be calculated from this tree. The first uses the software package BranchManager (Stone and Sidow 2007) to calculate a weight for each taxon in the phylogeny representing that taxon's contribution to the phylogeny as a whole. We call this method "BMweights." The second method calculates patristic distances (sum of branch lengths) between each taxon in the phylogeny using the python package DendroPy (Sukumaran and Holder 2010). We call this method "PDweights."

We calculate positional confidence scores for each of the N bootstrap alignments as follows. A raw score, S_{ij}, for a given residue in row i, column j of the reference alignment is calculated as

$$S_{ij} = \sum_{k \in R_{\text{ng}}^{(j)}} I_{ik}^{(j)} s_{ik}^{(j)}, \tag{1}$$

where $R_{\text{ng}}^{(j)}$ represents all rows in column j which are not gaps. We calculate $s_{ik}^{(j)}$ according to the given scoring algorithm:

$$s_{ik}^{(j)} = \begin{cases} 1 & \text{if Guidance} \\ w_i w_k & \text{if BMweights} \\ d_p(i,k) & \text{if PDweights} \end{cases}, \tag{2}$$

where w_i is the phylogenetic weight of the taxon at row i, as calculated by BranchManager, and $d_p(i,k)$ is the patristic distance between the taxa at rows i and k. The indicator function

$$I_{ik}^{(j)} = \begin{cases} 1 & \text{if reference alignment residue pair } (i,k)^{(j)} \text{ is present in bootstrap alignment} \\ 0 & \text{if reference alignment residue pair } (i,k)^{(j)} \text{ is absent in bootstrap alignment} \end{cases} \tag{3}$$

serves to compare the bootstrap- and reference-alignment residue pairings.

We then sum positional scores S_{ij} determined from each bootstrap replicate n. We normalize these scores across bootstrap replicates to yield a final score $\widetilde{S}_{ij}$ for each residue in the reference alignment. We use two different normalization schemes: original Guidance (defined in Penn et al. (2010)) and a novel gap-penalization scheme. These normalization

schemes are given by

$$\widetilde{S}_{ij} = \sum_{n}^{N} S_{ij} \Bigg/ \begin{cases} \sum_{n} \sum_{k \in R_{\text{ng}}^{(j)}} s_{ik}^{(j)}(n) & \text{if original Guidance} \\ \sum_{n} \sum_{k \in R_{\text{all}}^{(j)}} s_{ik}^{(j)}(n) & \text{if gap-penalization} \end{cases}, \tag{4}$$

where $R_{\text{all}}^{(j)}$ represents all rows in column j, including gaps, $s_{ik}^{(j)}(n)$ represents the quantity $s_{ik}^{(j)}$ for bootstrap replicate n, and the sum runs over all N replicates. By considering all rows instead of just rows that are not gaps, the gap-penalization scheme will naturally assign lower scores to highly gapped columns. We refer to the algorithms normalized by the original Guidance scheme as Guidance, BMweights, and PDweights. When normalized with the gap-penalization scheme, we refer to them, respectively, as GuidanceP, BMweightsP, and PDweightsP. Note that scores calculated using the Guidance algorithm with the original normalization scheme are equivalent to those originally derived by Penn et al. (2010).

Sequence Simulation

Coding sequences were simulated using Indelible (Fletcher and Yang 2009). To ensure that our simulations reflected realistic protein sequences, we simulated according to evolutionary parameters of the H1N1 hemagluttinin (HA) influenza protein. To derive these parameters, we aligned 1038 HA protein sequences collected from the Influenza Research Database (http://www.fludb.org) with mafft, specifying the "–auto" flag, (Katoh et al. 2002, 2005) and then back-translated to a codon alignment using the original nucleotide sequence data. We generated a phylogeny from this codon alignment in RAxML (Stamatakis 2006b) using the GTRGAMMA model. Using the codon alignment and phylogeny, we inferred evolutionary parameters with the REL (random effects likelihood) method (Nielsen and Yang 1998) using the HyPhy software (Kosakovsky Pond et al. 2005), with five dN/dS rate categories as free parameters under the GY94 evolutionary model (Goldman and Yang 1994). We employed a Bayes Empirical Bayes approach (Yang et al. 2000) to obtain infer dN/dS values at each site, which we used to assess a complete distribution of site rates. The resulting distribution was log-normal with a mean $dN/dS = 0.37$ with 8.3% of sites under positive selection ($dN/dS > 1$). We binned these rates into 50 equally spaced categories for specification in Indelible, which required a discrete distribution of dN/dS values. Again according to parameters derived from the HA analysis, we fixed κ, the transition-to-transversion ratio,

at 5.3 for all simulations. We additionally set the average alignment length to 400 codons with state codon frequencies equal to those directly calculated from the HA alignment.

We simulated 100 alignments across four different real gene trees each, yielding a total of 400 simulated alignments. Phylogenies used included an 11-taxon tree of the mammalian olfactory receptor OR5AP2 (Spielman and Wilke 2013), a 26-taxon tree of mammalian rhodopsin sequences (Spielman and Wilke 2013), a 60-sequence tree of phosphoribulokinase (PRK) genes from photosynthetic eukaryotes (Yang et al. 2011), and a 158-taxon multilocus tree of flatfish sequences (Betancur-R et al. 2013). The latter two phylogenies were obtained from TreeBASE (http://treebase.org). For each simulation set, we directly calculated an indel (insertion-deletion) rate directly from these trees' original alignments, to use as simulation parameters, by dividing the total number of gaps present by the total number of positions in each alignment. Respectively, indel rates for the 11-, 26-, 60-, and 158-sequence simulation sets were 0.053, 0.019, 0.0041, and 0.0066. We used indel rates given by the trees' alignments rather than an indel rate calculated from the HA alignment because phylogeny branch lengths directly impact how this indel rate is applied in Indelible; using a single indel rate across all simulation sets could introduce an unrealistic number of gaps. Setting indel rates according to each phylogeny's original data set protected against excessively frequent or infrequent indels.

Alignment and Positive-Selection Inference

We constructed all alignments using linsi (Katoh et al. 2002, 2005) within the context of our Guidance reimplementation. Phylogenies used to calculate phylogenetic weights for the BMweights and PDweights algorithms were constructed in RAxML, specifying "-m PROTCATWAG" as the model of sequence evolution (Stamatakis 2006b,a). In addition to an unfiltered alignment, we generated six filtered alignments (one for each filtering algorithm and each normalization scheme), masking residues with scores ≤ 0.5 "?". To investigate potential biases introduced by the scoring threshold, we also masked residues below the scoring cutoffs of 0.3, 0.7, and 0.9 for alignments constructed with the Guidance and GuidanceP filters.

We inferred positive selection for every condition using both FUBAR (Murrell et al. 2013) with default parameters. We processed only the unfiltered alignments and alignments masked at a scoring cutoff of 0.5 with the PAML M8 model, specifying F3x4 codon

frequency and "cleandata = 0" in the control file (Yang 2007). Phylogenies specified for positive-selection inference were those constructed during the Guidance alignment procedure when deriving phylogenetic weights. All filtered alignments derived from a unfiltered alignment were processed with identical phylogenies to remove any confounding effects of differing phylogenies. Note that while we employed FUBAR to assess positive selection for all simulation sets, we did not use PAML to infer positive selection for the largest set (158 sequences).

We then compared resulting positive-selection inferences for each alignment to its respective true alignment's dN/dS values, given by Indelible during simulation, to assess performance accuracy. As residues may have been differently aligned relative to the true simulated alignment, we adopted a consensus method to compare the alignments we constructed to the true alignments. If at least 50% of the residues present in a true alignment column were present in an inferred alignment column, we considered the true alignment column's dN/dS as the true value for that inferred alignment column. We considered sites positively selected if the posterior probability of ($dN/dS > 1$) was ≥ 0.9 **is the ¿= a verb?**

Statistics were performed using Python and R. Linear modeling was conducted using the R package lme4 (Bates et al. 2012). We inferred effect magnitudes and significance and corrected for multiple testing using the R multcomp package's glht() function with default settings (Hothorn et al. 2008). All code used is available at **the github or something**.

References

Bates D, Maechler M, Bolker B. 2012. lme4: Linear mixed-effects models using S4 classes. R package version 0.999999-0.

Betancur-R R, Li C, Munroe T A, Ballesteros J A, Orti G. 2013. Addressing gene tree discordance and non-stationarity to resolve a multi-locus phylogeny of the flatfishes (Teleostei: Pleuronectiformes). Systematic Biology 62(5):763–785.

Bush R, Bender C, Subbarao K, Cox N, Fitch W. 1999. Predicting the evolution of human influenza a. Science 286:1921–1925.

Castresana J. 2000. Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. Molecular Biology and Evolution 17(4):540–552.

Dwivedi B, Gadagkar S R. 2009. Phylogenetic inference under varying proportions of indel–induced alignment gaps. BMC Evolutionary Biology 9(1):211.

Edgar R C. 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Research 32:1792–1797.

Fletcher W, Yang Z. 2009. INDELible: A Flexible Simulator of Biological Sequence Evolution. Molecular Biology and Evolution 26(8):1879–1888.

Fletcher W, Yang Z. 2010. The Effect of Insertions, Deletions, and Alignment Errors on the Branch–Site Test of Positive Selection. Molecular Biology and Evolution 27(10):2257–2267.

Goldman N, Yang Z. 1994. A codon-based model of nucleotide substitution for protein-coding DNA sequences. Molecular Biology and Evolution 11:725–736.

Hothorn T, Bretz F, Westfall P. 2008. Simultaneous inference in general parametric models. Biometrical Journal 50(3):346–363.

Jordan G, Goldman N. 2012. The effects of alignment error and alignment filtering on the sitewise detection of positive selection. Mol Biol Evol 29:1125–1139.

Katoh K, Kuma K I, Toh H, Miyata T. 2005. MAFFT version 5: improvement in accuracy of multiple sequence alignment. Nucleic Acids Res 33:511–518.

Katoh K, Misawa K, Kuma K I, Miyata T. 2002. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. Nucleic Acids Res 30:3059–3066.

Kosakovsky Pond S L, Frost S D W, Muse S V. 2005. HyPhy: hypothesis testing using phylogenies. Bioinformatics 12:676–679.

Kryazhimskiy S, Bazykin G, Plotkin J, Dushoff J. 2008. Directionality in the evolution of influenza a haemagglutinin. Proc Royal Soc B 275:2455–2464.

Loytynoja A, Goldman N. 2008. Phylogeny–aware gap placement prevents errors in sequence alignment and evolutionary analysis. Science 320:1632–1635.

Markova-Raina P, Petrov D. 2011. High sensitivity to aligner and high rate of false positives in the estimates of positive selection in the 12 Drosophila genomes. Genome Research 21(6):863–874.

Meyer A G, Wilke C O. 2012. Integrating sequence variation and protein structure to identify sites under selection. Mol Biol Evol 30:36–44.

Murrell B, Moola S, Mabona A, Weighill T, Scheward D, Kosakovsky Pond S L, Scheffler K. 2013. FUBAR: A Fast, Unconstrained Bayesian AppRoximation for inferring selection. Molecular Biology and Evolution 30:1196–1205.

Nielsen R, Yang Z. 1998. Likelihood models for detecting positive selected amino acid sites and applications to the HIV-1 envelope gene. Genetics 148:929–936.

Notredame C, Higgins D G, J H. 2000. T–Coffee: a novel method for fast and accurate multiple sequence alignment. J Mol Biol 302:205–217.

Nuin P A S, Wang Z, Tillier E R M. 2006. The accuracy of several multiple sequence alignment programs for proteins. BMC Bioinformatics 7:471.

Ogden T H, Rosenberg M S. 2006. Multiple Sequence Alignment Accuracy and Phylogenetic Inference. Systematic Biology 55(2):314–328.

Penn O, Privman E, Landan G, Graur D, Pupko T. 2010. An alignment confidence score capturing robustness to guide tree uncertainty. Mol Biol Evol 27:1759–1767.

Price M N, Dehal P S, Arkin A P. 2010. FastTree2: Approximately maximum–likelihood trees for large alignments. PLoS One 5:e9490.

Privman E, Penn O, Pupko T. 2012. Improving the performance of positive selection inference by filtering unreliable alignment regions. Mol Biol Evol 29:1–5.

Schneider A, Souvorov A, Sabath N, Landan G, Gonnet G H, Graur D. 2009. Estimates of positive darwinian selection are inflated by errors in sequencing, annotation, and alignment. Genome Biology and Evolution 1(0):114–118.

Spielman S J, Wilke C O. 2013. Membrane environment imposes unique selection pressures on transmembrane domains of G protein–coupled receptors. Journal of Molecular Evolution 76:172–182.

Stamatakis A. 2006a. Phylogenetic models of rate heterogeneity: a high performance computing perspective.

Stamatakis A. 2006b. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics 22:21:2688–2690.

Stone E, Sidow A. 2007. Constructing a meaningful evolutionary average at the phylogenetic center of mass. BMC Bioinformatics 8:222.

Sukumaran J, Holder M T. 2010. DendroPy: A python library for phylogenetic computing. Bioinformatics 26:1569–1571.

Talavera G, Castresana J. 2007. Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments. Systematic Biology 56(4):564–577.

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

Thompson J D, Linard B, Lecompte O, Poch O. 2011. A comprehensive benchmark study of multiple sequence alignment methods: Current challenges and future perspectives. PLoS ONE 6(3):e18093.

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

Yang Y, Maruyama S, Sekimoto H, Sakayama H, Nozaki H. 2011. An extended phylogenetic analysis reveals ancient origin of "non-green" phosphoribulokinase genes from two lineages of "green" secondary photosynthetic eukaryotes: Euglenophyta and Chlorarachniophyta. BMC Research Notes 4:330.

Yang Z. 2007. PAML 4: Phylogenetic analysis by maximum likelihood. Molecular Biology and Evolution 24:1586–1591.

Yang Z, Nielsen R, Goldman N, Pedersen A M K. 2000. Codon-substitution models for heterogeneous selection pressure at amino acid sites. Genetics 155:431–449.

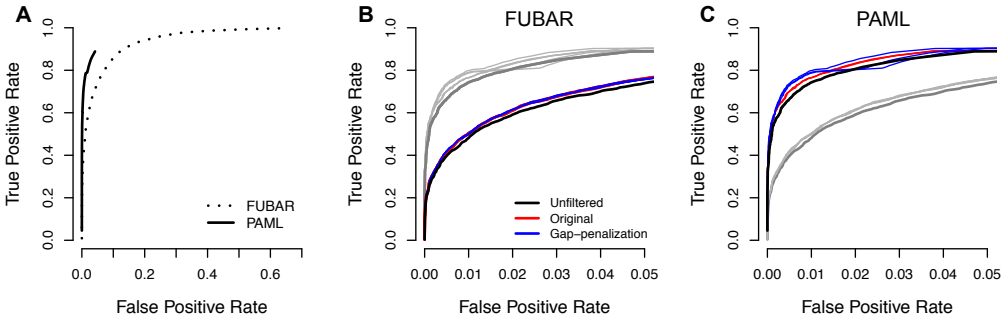

Figure 1: ROC curve as averaged across 60-sequence simulation set. A) Unfiltered alignments for FUBAR (dashed) and PAML (solid). Note that neither method achieved FPRs greater than shown. B) FUBAR in color with PAML results shown in grey. C) PAML in color with FUBAR results shown in grey.

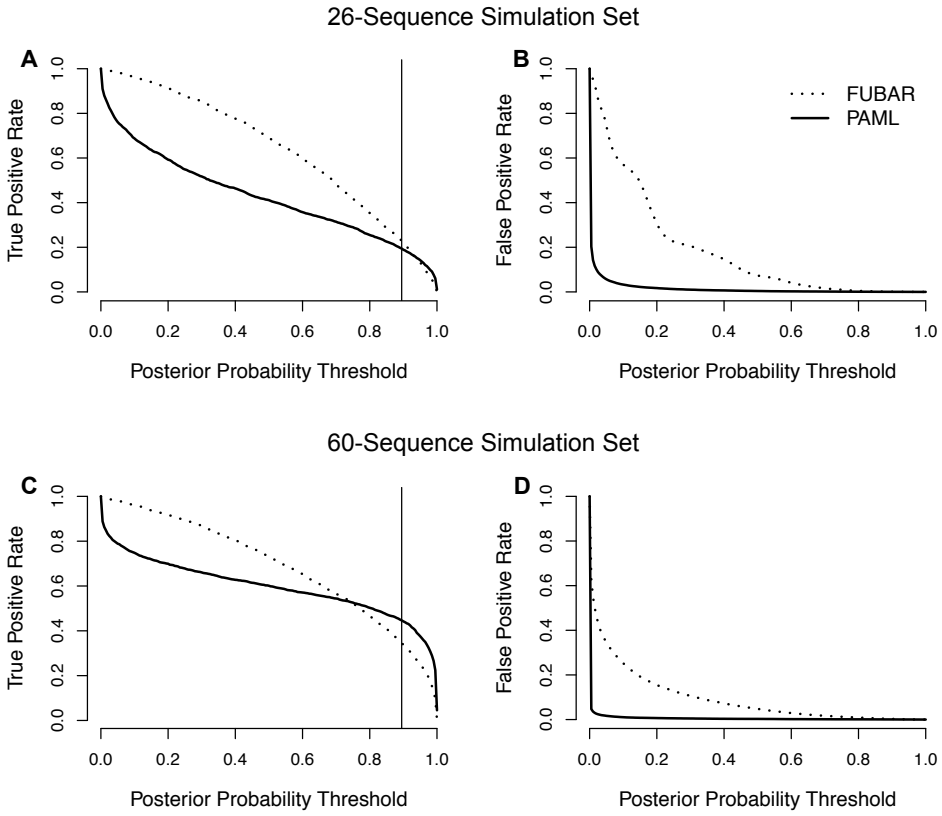

Figure 2: True and false positive rates recovered from FUBAR (dashed lines) and PAML (solid lines) analysis with unfiltered alignments, averaged across the 26-sequence and the 60-sequence simulation sets, against across posterior probability thresholds used to call sites as positively selected. Vertical lines indicate the posterior probability threshold of 0.9 used in the present study to call positively-selected sites. A) TPR, 26-sequence set. B) FPR, 26-sequence set. C) TPR, 60-sequence set. D) FPR, 60-sequence set. As the number of sequences increased, PAML's TPR improved at lower posterior probabilities, while its FPR remained remarkably low across all posterior probabilities for both simulation sets. While FUBAR's performance did improve with the inclusion of more sequences, its overall TPR behavior did not change as dramatically as did PAML's.

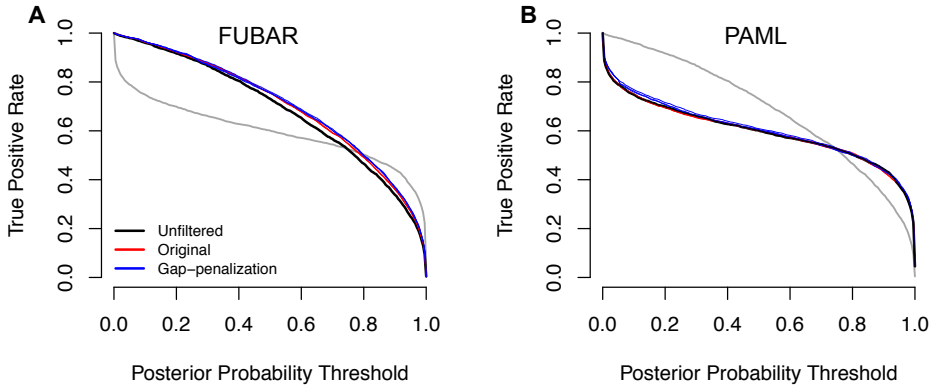

Figure 3: True positive rate against posterior probability threshold for calling positively selected sites, averaged across the 60-sequence simulation set. A) FUBAR in color with PAML results shown in grey. B) PAML in color with FUBAR results shown in grey. Filtered alignments behave similarly to unfiltered alignments across all posterior probabilities.

Table 1: Comparison between original and gap-penalization normalization schemes on TPR of positive-selection inference.

Simulation Set	Method	Guidance	BMweights	PDweights
11 taxa	FUBAR	0.001 (0.98%)	-1.76×10^{-5} (-0.002%)	0.001 (0.95%)
	PAML	-2.68×10^{-4} (-0.31%)	2.57×10^{-5} (0.03%)	-0.001 (-1.50%)
26 taxa	FUBAR	0.0032 (1.38%)	4.02×10^{-4} (1.75%)	0.0018 (0.77%)
	PAML	**0.006** (3.02%)*	**0.007** (3.32%)**	**0.006** (2.89%)*
60 taxa	FUBAR	**0.010** (2.67%)*	**0.012** (3.30%)**	0.005 (1.29%)
	PAML	-0.002 (-0.46%)	0.010 (2.23%)	0.008 (1.84%)
158 taxa	FUBAR	0.003 (0.79%)	0.002 (0.48%)	0.001 (0.28%)

NOTE.— Significance levels: $^{**}P < 0.01$; $^{*}P < 0.05$. Each column displays the magnitude of TPR difference between normalization schemes, represented as gap-penalization minus original, for each algorithm, respectively. Percentage changes from the original normalization scheme are shown in parentheses. All significance levels were corrected for multiple comparisons using the R multcomp package (Hothorn et al. 2008).

Table 2: Effect of alignment filtering on TPR of positive-selection inference.

			Filtered TPR		
Simulation Set	Method	Unfiltered TPR	GuidanceP	BMweightsP	PDweightsP
11 taxa	FUBAR	0.108	0.109 (1.04%)	0.110 (1.86%)	0.110 (1.37%)
	PAML	0.087	0.088 (0.49%)	0.088 (0.83%)	0.088 (0.79%)
26 taxa	FUBAR	0.229	0.232 (1.54%)	0.233 (1.83%)	0.233 (1.91%)
	PAML	0.194	**0.204** (4.87%)*	**0.203** (4.56%)*	**0.203** (4.58%)*
60 taxa	FUBAR	0.345	**0.379** (9.92%)**	**0.377** (9.31%)**	**0.375** (8.75%)**
	PAML	0.447	0.447 (0.19%)	0.440 (-1.43%)	0.445 (-0.30%)
158 taxa	FUBAR	0.374	**0.388** (3.89%)**	**0.387** (3.68%)**	**0.387** (3.47%)**

NOTE.— Significance levels: $^{**}P < 10 \times 10^{-5}$; $^{*}P < 10 \times 10^{-4}$. *Unfiltered TPR:* average TPR for unfiltered alignments; *Filtered TPR:* average TPR for alignments filtered with each respective algorithm, with percent change from unfiltered alignment shown in parentheses. All significance levels are relative to the given simulation set's unfiltered alignment. We detected no significant TPR differences among filters tested within sequence simulation sets. All significance levels were corrected for multiple comparisons using the R multcomp package (Hothorn et al. 2008).

Table 3: Effect of masking cutoff on TPR of positive-selection inference.

Simulation Set	0.5 TPR	0.9 TPR	Percent TPR Decrease
11 taxa	0.109	0.104	4.99%*
26 taxa	0.232	0.220	5.28%**
60 taxa	0.379	0.281	25.8%***
158 taxa	0.388	0.374	3.57%***

NOTE.— Significance levels: $^{***}P < 1 \times 10^{-6}$; $^{**}P < 1 \times 10^{-5}$; $^{*}P < 1 \times 10^{-3}$.
0.5 TPR: average TPR for alignments masked at a cutoff of 0.5 with GuidanceP; *0.9 TPR:* average TPR for alignments masked at a cutoff of 0.9 with GuidanceP; *Percent TPR Decrease:* average percent decrease in TPR recovered between alignments masked at cutoffs of 0.5 and 0.9. All significance levels were corrected for multiple comparisons using the R multcomp package (Hothorn et al. 2008).