Detecting signatures of selection from DNA sequences using Datamonkey

Art F.Y. Poon, Simon D.W. Frost and Sergei L. Kosakovsky Pond*

Antiviral Research Center, Department of Pathology, University of California San Diego, La Jolla, California, USA

*Corresponding Author

Sergei L Kosakovsky Pond, PhD

Assistant Project Scientist

Department of Pathology,

Antiviral Research Center

150 W. Washington St., #100

San Diego, CA 92103-2005

Phone: 619-543-8899

E-mail: spond@ucsd.edu

Keywords: Positive selection, adaptive evolution, *dN* and *dS* estimation, HyPhy, phylogenetic analysis, maximum likelihood inference, parallel algorithms, web service.

Abstract

Natural selection is a fundamental process affecting all evolving populations. In the simplest case, positive selection increases the frequency of alleles that confer a fitness advantage relative to the rest of the population, or increases its genetic diversity, and negative selection removes those alleles that are deleterious. Codon-based models of molecular evolution are able to infer signatures of selection from alignments of homologous sequences by estimating the relative rates of synonymous (*dS*) and non-synonymous substitutions (*dN*). Datamonkey (http://www.datamonkey.org) provides a user-friendly web interface to a wide collection of state-of-the-art statistical techniques for estimating *dS* and *dN* and identifying codons and lineages under selection, even in the presence of recombinant sequences.

1. Introduction

Natural selection plays a pivotal role in shaping the genetic variation of populations and driving the differentiation of biological taxa. The study of causes and mechanisms of molecular adaptation is one of the fundamental goals of evolutionary biology, with numerous important applications.

Most current techniques for the inference of selection from protein-coding sequences are based on the following observation: a *nonsynonymous* (or replacement) substitution in a protein-coding sequence changes the primary sequence of the encoded protein and is more likely to influence the fitness of an organism than a random *synonymous* substitution that leaves the amino acid sequence unchanged. If nonsynonymous mutations at a particular codon site in the sequence have a negligible effect on the function or expression of the protein (and hence on its fitness), then the rate of nonsynonymous substitutions (*dN*) should be comparable to the rate of synonymous substitutions (*dS*), and the site evolves *neutrally*. An excess of nonsynonymous substitutions (*dN*>*dS*) can be interpreted as *positive selection* – suggestive that replacement substitutions increase fitness. A paucity of replacement changes (*dN*<*dS*) indicates that negative selection is working to remove such substitutions from the gene pool.

The ratio ω=*dN*/*dS* (sometimes also denoted K_A/ K_S) has seen wide adoption as a measure of selective pressure (***1, 2***). Datamonkey (http://www.datamonkey.org) (***3***) is one of the many available software tools for estimating ω (Datamonkey actually reports *dN – dS, see* **Note 1**) using a variety of evolutionary models, with several unique advantages. Datamonkey has an intuitive and streamlined interface that pro-

vides easy access to complex, state-of-the-art evolutionary models. Complex models are quickly fitted using a remote computer cluster; an analysis that would otherwise take hours to run on a conventional desktop computer will finish in minutes on a cluster. The collection of available methods is constantly updated as novel techniques are published, obviating the need for a practicing scientist to keep apace of new methodological advances and to install and learn how to use a plethora of software packages. Finally, Datamonkey can analyze selection in the presence of recombination – something few other publicly-accessible programs can currently do.

Datamonkey uses the HyPhy package (***4***) as its computational engine. All of the selection analyses implemented in Datamonkey, as well as a number of other analyses, can also be carried out directly in HyPhy. For an in-depth discussion of the methods and a tutorial on how sequences can be analyzed for selection in HyPhy, we direct the interested reader to **ref.** (***5***).

Currently, Datamonkey may be used to address the following questions:

- Which codon sites in the alignment are subject to positive or negative selection? SLAC/FEL/REL methods (***6***) can estimate *dN* and *dS* at each codon site. More specialized hypotheses can also be tested. For instance, if the alignment contains sequences from multiple individuals (e.g. viruses) which sites are positively selected at the level of a population? (the IFEL method (***7***)).

- At what point in the evolutionary history of sequences did selection occur? The GA Branch method (***8***) can be used to assign values of *dN*/*dS* to every branch (lineage) in the phylogenetic tree.

- Does a sequence alignment contain recombinant sequences (GARD (***9***))? Many traditional selection techniques can be misled by recombination (***10***), but recombination can be corrected for by identifying non-recombinant fragments in the alignment and reconstructing a phylogenetic tree for each fragment (***9***).

- Is there evidence of positive selection operating within recombining fragments of the alignment? The PARRIS test (***11***) is used to determine whether a proportion of sites have *dN*>*dS* in the context of recombination.

2. Program Usage

To perform a selection analysis, Datamonkey requires an uploaded alignment of at least three homologous coding nucleotide sequences (*see* **Note 2** for browser compatibility). Codon-based methods for estimating *dN* and *dS* can be applied to any sequence alignment, but there are several considerations to keep in mind. Ideally, the alignment should represent a single gene or a part thereof (*e.g.* a subunit), sampled over multiple taxa (*e.g.* mammalian interferon genes) or a diverse population sample (*e.g.* Influenza A viruses infecting different individuals;*see* **Note 3**). The number of sequences in the alignment is important: too few sequences will contain too little information for meaningful inference, while too many may take too long to run. At the time of this writing, Datamonkey permits up to 150 sequences for SLAC analyses, 100 for FEL/IFEL analyses, 40 for REL and PARRIS and 25 for GA-Branch. These numbers are determined by current hardware availability and will be

increased in the future (*see* **section 2.1**). As a rule of thumb, at least 10 sequences are needed to detect selection at a single site (SLAC/FEL/IFEL/REL) with any degree of reliability, while as few as 4 may be sufficient for alignment-wide inference (PARRIS/GA-Branch). The median number of sequences in an alignment submitted to Datamonkey is 19. In addition, comparative methods may be ill suited to study certain kinds of selection (*see* **Note 4**).

It is a good practice to visually inspect your data to make sure that the sequences are aligned correctly. Of course, one can never be sure that an alignment is objectively "correct", but gross misalignments (*e.g.* sequences that are out of frame) are easy to spot with software that provides a graphical visualization of the alignment, such as HyPhy(***4***), Se-Al (http://tree.bio.ed.ac.uk/software/seal/), or BioEdit (http://www.mbio.ncsu.edu/BioEdit/bioedit.html). You should verify that the alignment is in frame, *i.e.* that it does not contain stop codons, including premature stop codons, indicative of a frame shift, *e.g.* due to misalignment, or a non-functional coding sequence, and the terminal stop codon. Your alignment should exclude any non-coding region of the nucleotide sequence, such as introns or promoter regions, for which existing models of codon substitution would not apply. When coding nucleotide sequences are aligned directly, frameshifting (i.e. not in multiples of 3) gaps may be inserted, since the alignment program often does not take the coding nature of the sequence into account. Therefore it is generally a good idea to align translated protein sequences and then map them back onto constituent nucleotides. Datamonkey will perform a number of checks when it receives coding sequences and report all problems it encounters (*see* **Section 2.1.**).

Since Datamonkey uses the HyPhy package as its processing engine, it will accept files containing alignments in FASTA, PHYLIP, MEGA, and NEXUS formats.

If the alignment contains identical sequences, Datamonkey will discard all but one of the duplicate sequences before proceeding. This is done to speed up the analyses, because identical sequences do not contribute any information to the likelihood inference procedure (except via base frequencies), but the computational complexity of phylogenetic analyses grows with the number of sequences.

Finally, Datamonkey may rename some of the sequences to conform to HyPhy naming conventions for technical reasons (all sequence names must be valid identifiers, *e.g.* they cannot contain spaces). This is done automatically and has no effect on the subsequent analyses.

2.1 **Common issues when preparing the data for *Datamonkey***.

2.1.1. Non-text files. Datamonkey expects sequence alignments to be uploaded as text files. Any other format (Word, RTF, PDF) will not be recognized and must be converted into plain text prior to submission.

2.1.2. Nonstandard characters in the alignment. For instance, BioEdit may use the tilde ('~') character to denote a gap. The dot ('.') character is sometimes used as 'match the first sequence' character and sometimes as the gap character. Datamonkey will accept IUPAC nucleotide characters (ACGT/U and ambiguity characters) and '?', 'X', 'N' or '-' for gap or missing data (Datamonkey is not case sensitive). All other characters in sequence data will be skipped and could result in frame shifts, which will be reported upon upload.

2.1.3. Uploading an amino-acid alignment. Datamonkey employs codon models which require the knowledge of silent substitutions, lost upon translation to amino-acids.

2.1.4. Termination codons. Datamonkey will reject any alignments that contains stop codons, even if the stop codon is at the end of the sequence (*i.e.* is a proper termination codon). Please strip all stop codons out of the alignment prior to uploading it (the HyPhy standard analysis `Data File Tools:CleanStopCodons.bf` can do this by replacing all stop codons with indels).

2.1.5. Alignments that are too gappy. If an alignment contains more than 50% of indels, it may not be properly processed (*e.g.* it could be read as a protein alignment, depending on the alignment format).

2.1.6. Alignments that are too large. If your alignment exceeds the size currently allowed by Datamonkey, consider running your analysis locally in HyPhy. A detailed discussion of how HyPhy can be used for that purpose can be found in **ref. (*5*)**

2.1.7. Incorrect genetic code. If the genetic code is misspecified (e.g. the mitochondrial code is applied to nuclear sequences), valid alignments may fail to upload and if they do, then the results may be compromised (because codons are mistranslated). Make sure the correct genetic code is selected on the data upload page.

3. Examples

To demonstrate a typical workflow with Datamonkey (http://www.datamonkey.org), we begin by analyzing 21 sequences of the H5N1 Influenza A virus hemagglutinin gene, available for download in FASTA format from http://www.datamonkey.org/data/Flu.fasta (download this alignment to a text file on your computer). Hemagglutinin is a viral protein expressed on the surface of influenza virions and responsible for binding to the sialic acid receptors of host cells. This protein is heavily targeted by the immune response of the host. Within-gene recombination in influenza is thought to be rare, hence we will proceed with the assumption that a single phylogeny is adequate.

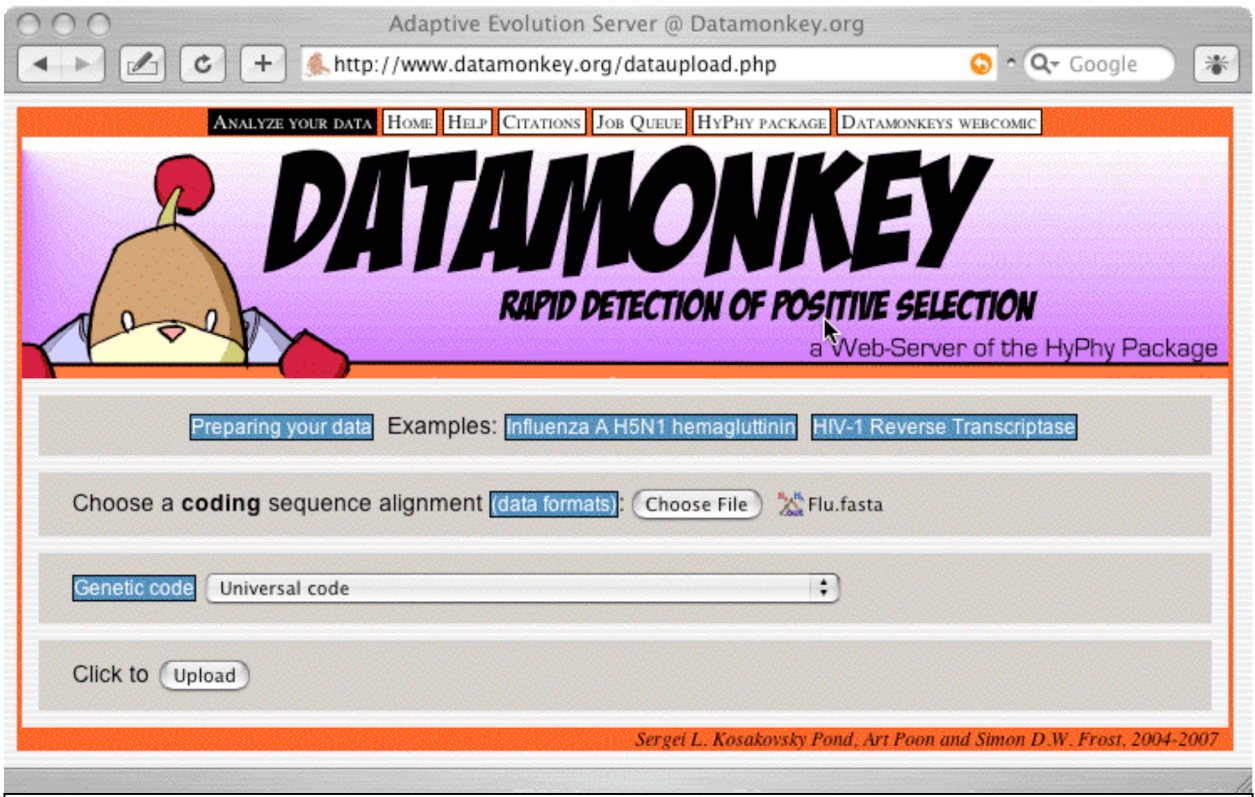


Figure 1 Datamonkey data upload page.

3.1. Upload the alignment.

3.1.1 Select the alignment file. The front page of http://www.datamonkey.org includes a link to the data upload page (**Fig. 1**), either in the tool bar at the top of the page, or via a large graphical button at the bottom of the page. Click on the "Browse" button, located next to the field labeled "Choose a sequence file:" to use your browser's interactive window to locate the file on your computer.

3.1.2. Choose the genetic code. Datamonkey can interpret codons using one of the twelve standard genetic codes, but it will employ the "Universal" genetic code by default, which is appropriate for Influenza A.

3.1.3. Examine the uploaded file. Upon a successful upload, Datamonkey reports some basic statistics on the alignment, including the number of sequences, columns (codon sites) and partitions (for alignments with recombinant sequences, there will be multiple partitions — one for each non-recombinant fragment), base frequencies and an amino-acid translation of the alignment — the PDF version is handy for an at-a-glance sequence consensus and minor variants report (**Fig. 2**). You can also verify that the alignment was uploaded correctly by selecting a sequence (using the drop-down menu labeled "BLAST your sequences?") to BLAST against the NCBI non-redundant nucleotide sequence database.

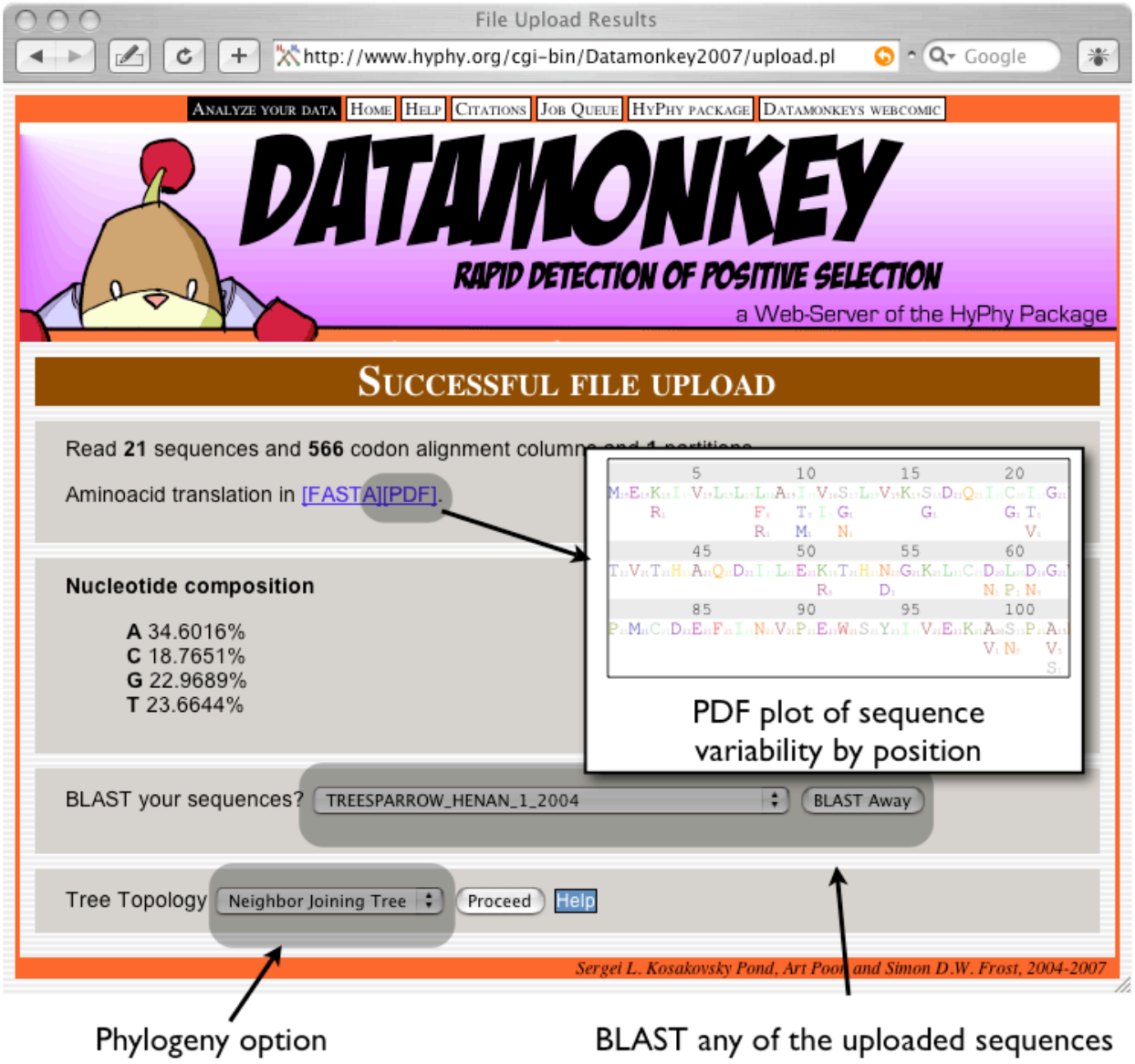


Figure 2 Data upload summary page.

3.1.4. Phylogeny. Datamonkey will automatically detect whether a tree topology (or multiple topologies for recombinant data) is present in the file, *e.g.* the TREES block in a NEXUS-formatted file. If each tree tip can be matched up with a sequence in the alignment, Datamonkey will make this tree available for analysis. If no tree(s) were found in the file (as is the case for `Flu.fasta`), then Datamonkey can estimate a neighbor joining (NJ, (***12***)) phylogeny from the alignment, using Tamura-Nei (***13***) nucleotide distances. There is empirical evidence that selection analyses are robust to some error in the phylogeny, hence a “quick and dirty” method like NJ should be suf-

ficient in most cases. Click on the `"Proceed"` button to move on to the analysis setup page.

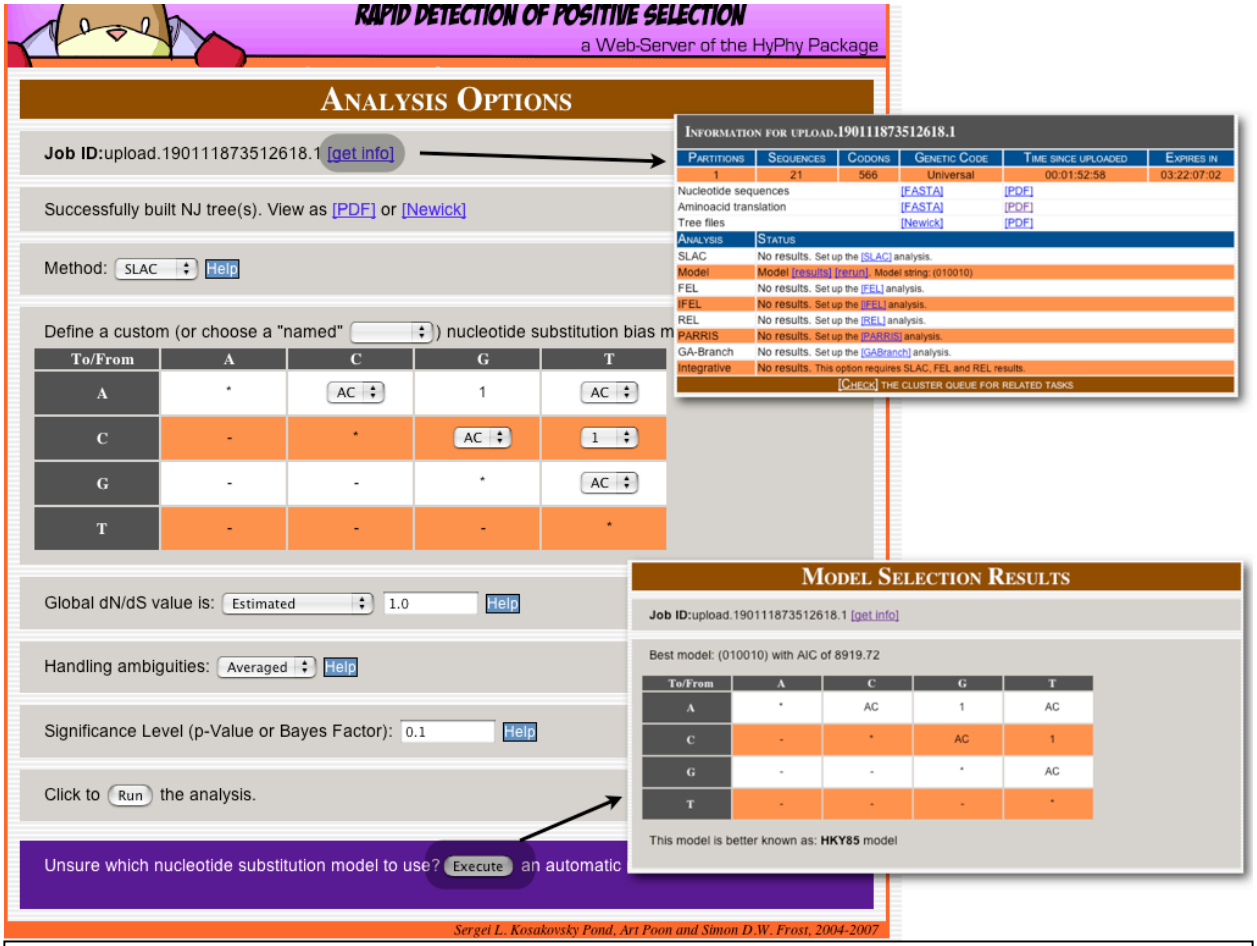


Figure 3 Analysis setup page

3.2. Analysis setup page. The analysis setup page (**Fig. 3**) is used to configure all selection analyses available via Datamonkey.

3.2.1. Job Status. The first thing to note is the **Job ID** bar. Each successfully uploaded file will be assigned a random identifier, which can be used to track all the analyses performed on the alignment. Clicking on the `[get info]` link brings up the status page for this alignment. Nearly every Datamonkey page will have a link to

the status page, and we will later discuss how the status page can be used in more detail.

3.2.1. Method. This drop-down menu lists all analyses that can be run on the uploaded alignment. Some of the options may not be present for large and/or recombinant alignments. We will perform SLAC, FEL and REL analyses (for detecting sites under selection) on `Flu.fasta`,

3.2.2. Nucleotide substitution bias model. Each of the methods implemented by Datamonkey makes use of a nucleotide substitution model to estimate the branch lengths and nucleotide substitution biases (such as transition/transversion biases) of the tree from your alignment. Datamonkey can make use of one of the 203 time-reversible nucleotide substitution models. The most general supported time-reversible model (denoted as REV) is comprised of eight free parameters (3 nucleotide frequencies + 5 substitution rates). Four of the most frequently used models (F81, HKY85, TrN93 and REV) are predefined as "named" options.

A parameter-rich model could conceivably overfit a small alignment, while a model that is too simple may lead to biased inference; for this reason, Datamonkey provides an automated tool (link at the bottom of the analysis setup page, see **Fig. 3**) that will select the best-fitting nucleotide model (*see* **Note 5**) from all 203 reversible models (***14***) . Run the model selection procedure on `Flu.fasta` and verify that the HKY85 (***15***) model is the best fitting model for influenza hemagglutinin. After the model selection analysis is finished, you can return to the analysis setup page from the model results page by clicking on the `[get info]` link in the Job ID bar, and then on the link offering to set-up the SLAC analysis.

3.2.3. Analysis Options. There are up to three analysis options (the first two only apply to SLAC, see **Note 6**). Each option has a link to the relevant help page, explaining what each settings means. `Significance level` determines how conservative each method should be, but this option only affects how the results are presented and can be adjusted after the analysis has finished. We begin by submitting a SLAC analysis with the HKY85 model and default analysis options.

3.3. The job queue page. After you submit the analysis by clicking on 'Run' button, Datamonkey will display a page with all the jobs currently queued for execution. The newly submitted page analyses are inserted at the end of the queue, and must wait for the jobs in front of it to finish. You may bookmark the queue page in your web browser and return to it later to check on the progress update. Once Datamonkey begins processing your analysis, it will display intermediate progress reports for the task and present a result page when it becomes available.

3.4. SLAC results page.

All Datamonkey analysis result pages consist of the following three parts (**Fig. 4**).

3.4.1 The job ID bar including the universal `[get info]` link to bring up the central job summary and page.

3.4.2. Data and analysis summary. For SLAC, Datamonkey reports descriptive statistics of the alignment (partitions, codons and estimated evolutionary tree lengths – note that very long trees can be indicative of misaligned sequences and are flagged as such), the inferred nucleotide substitution biases, log likelihood scores for the fitted models and the estimate of the alignment-wide *dN*/*dS*.

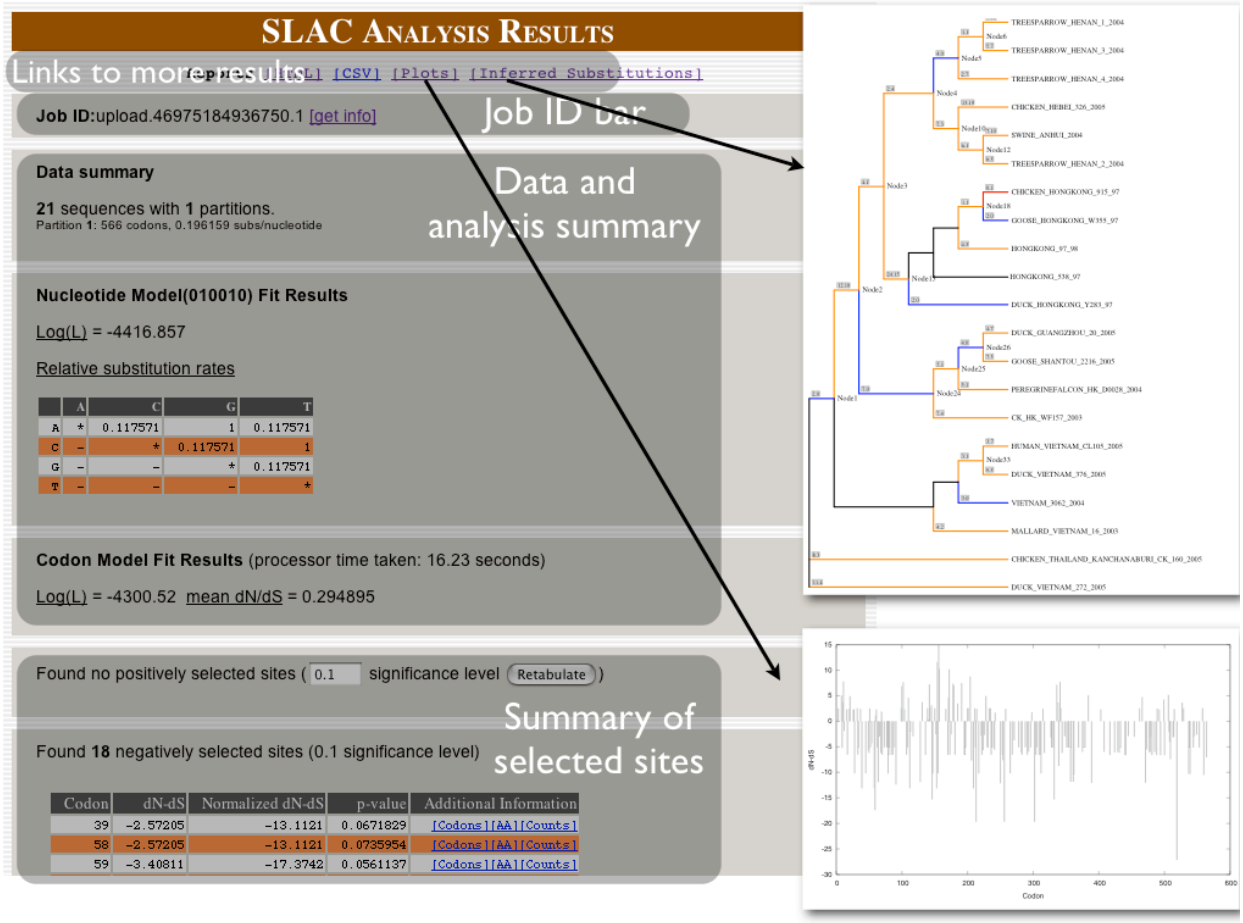


Figure 4 SLAC analysis results page

3.4.3. Links to more detailed results. A detailed or graphical output of various, analysis specific quantities, can be accessed via these links. For SLAC you can, for instance, plot *dN-dS* across sites, or map the number of inferred synonymous and non-synonymous substitutions to each branch of the tree (***Fig. 4***).

3.4.4. (SLAC/FEL/IFEL/REL) Summary of selected sites. Given a specific significance level, all those codons, which the method detected to be under positive or negative selection are reported. This section can be regenerated on the fly for a different significance level (`Retabulate`).

3.5. Interpreting SLAC results.

The summary output of selected sites lists every site where $dN/dS \neq 1$ with statistical significance (p-value) no greater than the value supplied by the user. The p-value bounds the rate of false positives, *e.g.* p=0.05 means that up to 5% of neutrally evolving sites may be incorrectly classified as selected (i.e. false positives). The p-value should be taken as a guideline, because the statistical properties of the test may vary from alignment to alignment. For instance, SLAC tends to be a very conservative test (***6***), hence the actual rate of false positives can be much lower than the significance level (hence the default of 0.1, *see* **Note 7**). For each codon site, Datamonkey will report the estimated the unadjusted *dN*–*dS* (*see* **Note 1**), *dN*–*dS* scaled by the total length of the tree (to facilitate direct comparison between different data sets), p-value for the test $dN \neq dS$ at that codon, and links to investigate the inferred mutations at that site.

You may notice that at p=0.1, SLAC reports no positively selected sites. Increase p to 0.2 and `Retabulate` the results to find that 4 codons (154,156,157,172) have p-values for positive selection in the 0.1-0.2 range (borderline selection). For codon 156, `Normalized` *dN*–*dS* = 14.93. If you now click on the `[Counts]` link in the `Additional Info` column, Datamonkey will display the list of inferred substitutions at that codon, showing at least 6 non-synonymous and 0 synonymous substitutions, mapped to 5 branches of the tree (**Fig. 5**). The p-value of 0.12 corresponds to the binomial probability that all six substitutions at that site will be non-synonymous by chance (*see* **Note 8**). SLAC estimates a number of quantities, which enable it to compute this probability (accessible via the detailed HTML report from the SLAC re-

sult page). For site 156, its inferred codon composition and relative branch lengths predict that a random substitution is synonymous with probability 0.30, assuming neutral evolution, such that the probability of observing 0/6 synonymous substitutions equals $(1–0.3)^6 \approx 0.12$, yielding the p-value.

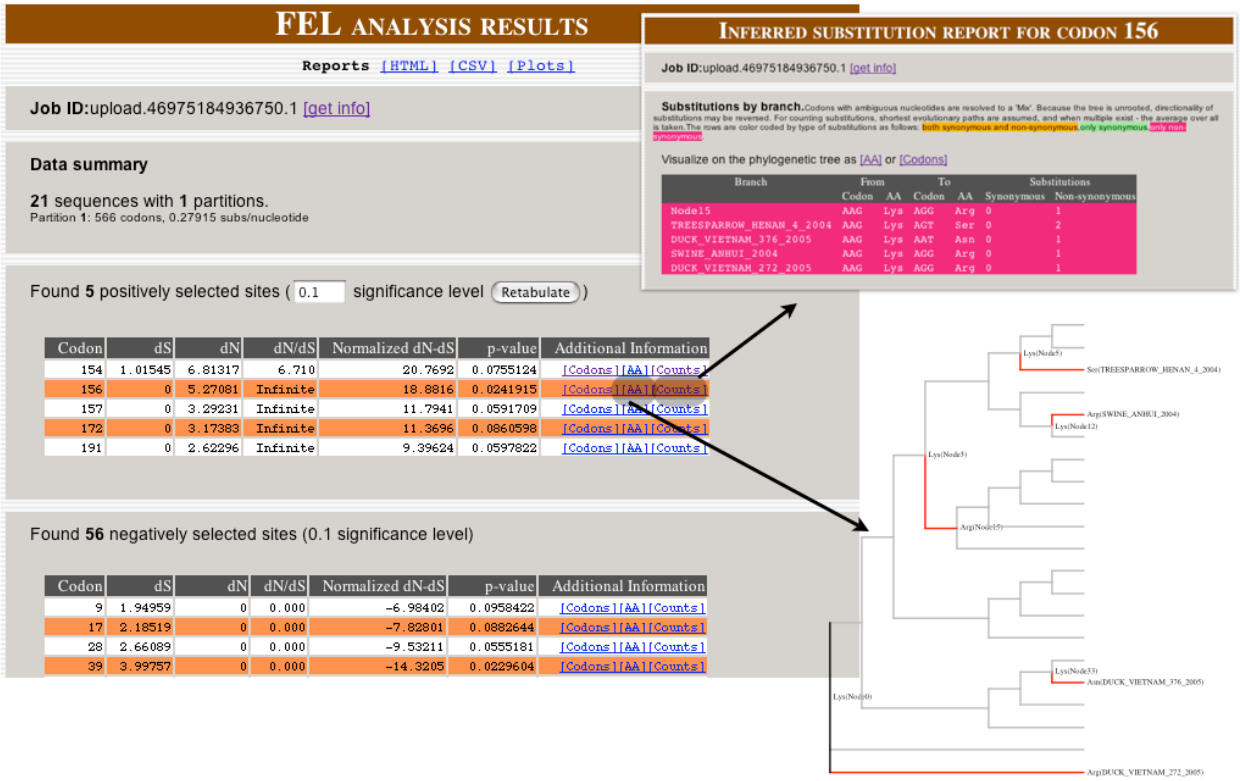


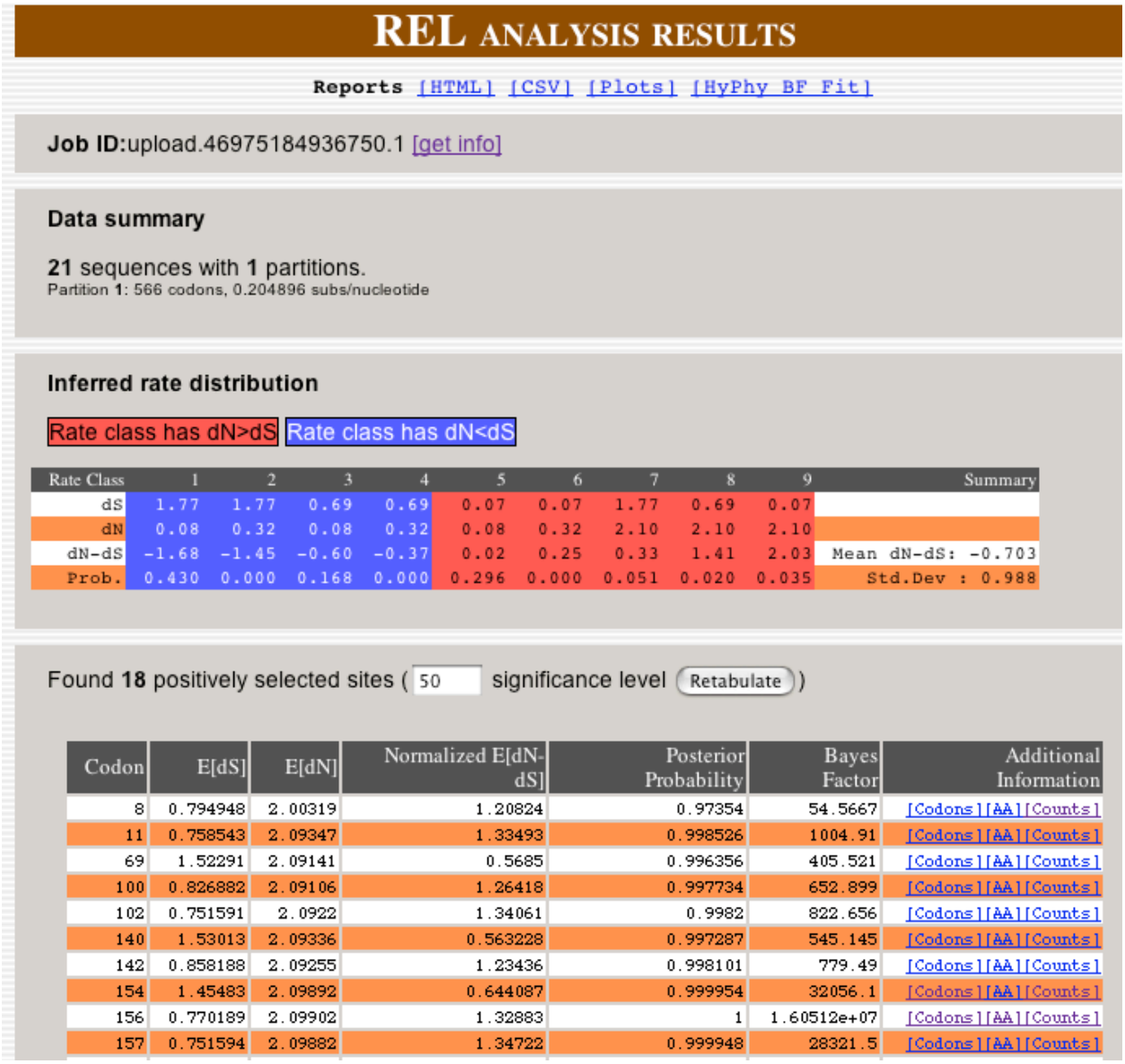
Figure 5 FEL results page for the influenza analysis and substitution maps for codon 156

3.6. FEL.

Next, return to the analysis setup page (**Fig. 2**), via the Job ID bar, and run the FEL analysis with default options (note that if you have run a substitution model selection,

Datamonkey will automatically select the appropriate model on the analysis setup page). The FEL result page (**Fig. 5**) is similar to the SLAC result page.

Like SLAC, FEL evaluates *dS* and *dN* at each site, but instead of basing its inference on the expected and inferred numbers of synonymous and non-synonymous substitutions, FEL directly estimates *dN* and *dS* based on a codon-substitution model, and derives the p-value for the test $dN \neq dS$ using a likelihood ratio test (***6***). FEL tends to be quite a bit more powerful than SLAC, *e.g.* note that all 4 borderline positively selected SLAC sites, have p-value in 0-0.1 for FEL), but it is an order of magnitude more computationally expensive.



REL ANALYSIS RESULTS

Reports [HTML] [CSV] [Plots] [HyPhy BF Fit]

Job ID:upload.46975184936750.1 [get info]

Data summary

21 sequences with **1** partitions.
Partition **1**: 566 codons, 0.204896 subs/nucleotide

Inferred rate distribution

Rate class has dN>dS | Rate class has dN<dS

Rate Class	1	2	3	4	5	6	7	8	9	Summary
dS	1.77	1.77	0.69	0.69	0.07	0.07	1.77	0.69	0.07	
dN	0.08	0.32	0.08	0.32	0.08	0.32	2.10	2.10	2.10	
dN-dS	-1.68	-1.45	-0.60	-0.37	0.02	0.25	0.33	1.41	2.03	Mean dN-dS: -0.703
Prob.	0.430	0.000	0.168	0.000	0.296	0.000	0.051	0.020	0.035	Std.Dev : 0.988

Found **18** positively selected sites (50 significance level Retabulate)

Codon	E[dS]	E[dN]	Normalized E[dN-dS]	Posterior Probability	Bayes Factor	Additional Information
8	0.794948	2.00319	1.20824	0.97354	54.5667	[Codons][AA][Counts]
11	0.758543	2.09347	1.33493	0.998526	1004.91	[Codons][AA][Counts]
69	1.52291	2.09141	0.5685	0.996356	405.521	[Codons][AA][Counts]
100	0.826882	2.09106	1.26418	0.997734	652.899	[Codons][AA][Counts]
102	0.751591	2.0922	1.34061	0.9982	822.656	[Codons][AA][Counts]
140	1.53013	2.09336	0.563228	0.997287	545.145	[Codons][AA][Counts]
142	0.858188	2.09255	1.23436	0.998101	779.49	[Codons][AA][Counts]
154	1.45483	2.09892	0.644087	0.999954	32056.1	[Codons][AA][Counts]
156	0.770189	2.09902	1.32883	1	1.60512e+07	[Codons][AA][Counts]
157	0.751594	2.09882	1.34722	0.999948	28321.5	[Codons][AA][Counts]

Figure 6 REL results page for the influenza analysis.

3.7. REL

Run the REL analysis with default options (**Fig. 6**). REL is similar to the popular likelihood methods implemented in the PAML package (***16***), with several important additions (e.g. synonymous rate variation, see (***6***)). Instead of directly estimating *dS* and *dN* at each site, REL estimates the parameters for discretized *distributions* of *dS* and

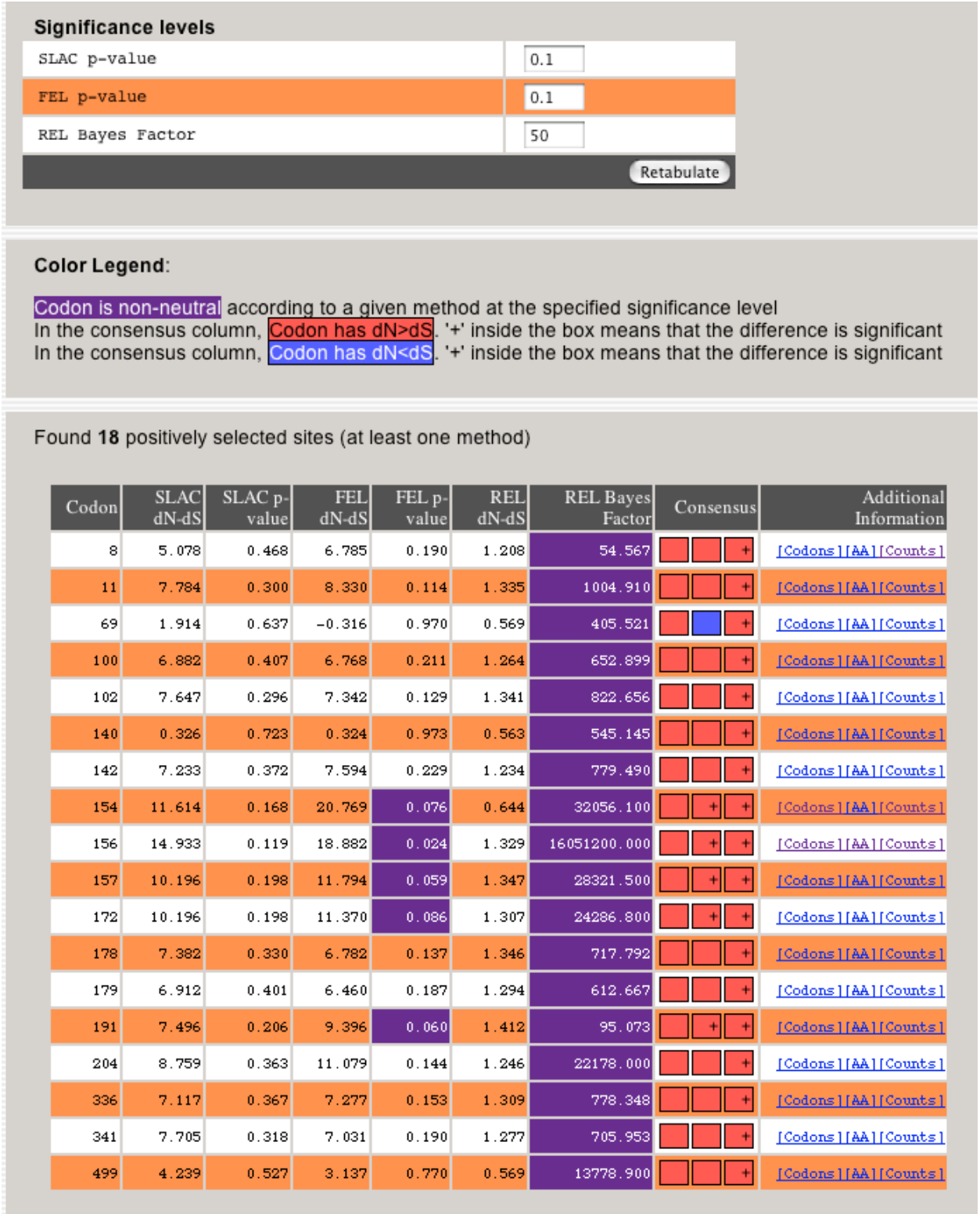
dN (with 3 rate categories for a total of 9 possible rate combinations) from the entire alignment, and then infers which of these each site is most likely to have. To decide if a codon has *dN*>*dS*, the REL method computes a Bayes factor (BF) defined as the ratio of posterior odds of having *dN*>*dS* to the prior odds. A large Bayes factor suggests that the data lend strong support to the hypothesis that a site is positively selected (as a rule of thumb, 1/BF is similar to the p-value). REL tends to be the most powerful of the three tests because it uses the entire alignment to make inferences about rates at each site, but also generally has the highest rate of false positives, because the distribution of rates to be fitted must be defined *a priori*, and it may not adequately model the unobserved distribution of rates.

For our example, 18 sites are called positively selected (*dN*>*dS*). However, REL does not take the magnitude of *dN*-*dS* into account, and nearly neutral or invariable sites (such as those from rate classes 5 or 7, **Fig. 6**) may count towards the positively selected category.

3.8. Integrative selection analysis.

Practical statistical inference is never 100% accurate, and there will be some false positives and negatives. As our example illustrates, different approaches to inferring the same effect can lead to different inferences. Fortunately, for larger datasets (e.g. > 100 sequences), all three methods tend to agree (***6***), with the SLAC being the most conservative (i.e. it may miss some selected sites, but will not identify many neutral sites as selected), followed by FEL (more detected sites), and then REL (most detected sites, but highest errors). For smaller datasets, like the example influenza alignment, it is a good idea to run all three methods, and compare their re-

sults side by side. When the methods corroborate each other's findings, we may be more confident in their inference.



Significance levels

SLAC p-value	0.1
FEL p-value	0.1
REL Bayes Factor	50

Retabulate

Color Legend:

Codon is non-neutral according to a given method at the specified significance level
In the consensus column, Codon has dN>dS. '+' inside the box means that the difference is significant
In the consensus column, Codon has dN<dS. '+' inside the box means that the difference is significant

Found **18** positively selected sites (at least one method)

Codon	SLAC dN-dS	SLAC p-value	FEL dN-dS	FEL p-value	REL dN-dS	REL Bayes Factor	Consensus	Additional Information
8	5.078	0.468	6.785	0.190	1.208	54.567	+	[Codons][AA][Counts]
11	7.784	0.300	8.330	0.114	1.335	1004.910	+	[Codons][AA][Counts]
69	1.914	0.637	-0.316	0.970	0.569	405.521	+	[Codons][AA][Counts]
100	6.882	0.407	6.768	0.211	1.264	652.899	+	[Codons][AA][Counts]
102	7.647	0.296	7.342	0.129	1.341	822.656	+	[Codons][AA][Counts]
140	0.326	0.723	0.324	0.973	0.563	545.145	+	[Codons][AA][Counts]
142	7.233	0.372	7.594	0.229	1.234	779.490	+	[Codons][AA][Counts]
154	11.614	0.168	20.769	0.076	0.644	32056.100	+	[Codons][AA][Counts]
156	14.933	0.119	18.882	0.024	1.329	16051200.000	+	[Codons][AA][Counts]
157	10.196	0.198	11.794	0.059	1.347	28321.500	+	[Codons][AA][Counts]
172	10.196	0.198	11.370	0.086	1.307	24286.800	+	[Codons][AA][Counts]
178	7.382	0.330	6.782	0.137	1.346	717.792	+	[Codons][AA][Counts]
179	6.912	0.401	6.460	0.187	1.294	612.667	+	[Codons][AA][Counts]
191	7.496	0.206	9.396	0.060	1.412	95.073	+	[Codons][AA][Counts]
204	8.759	0.363	11.079	0.144	1.246	22178.000	+	[Codons][AA][Counts]
336	7.117	0.367	7.277	0.153	1.309	778.348	+	[Codons][AA][Counts]
341	7.705	0.318	7.031	0.190	1.277	705.953	+	[Codons][AA][Counts]
499	4.239	0.527	3.137	0.770	0.569	13778.900	+	[Codons][AA][Counts]

Figure 7 Integrative selection page for the influenza analysis.

Once SLAC, FEL and REL have been run, the `Integrative Selection Analysis` option is enabled on the job status page. The integrative selection page (**Fig. 7**) tabulates all codons, which are detected by at least one of the methods, based on the specified significance levels. For the influenza analysis, codons 154, 156, 157, 172 and 191 are supported by FEL and REL, and all have large *dN-dS* values for SLAC with borderline (0.1-0.25) p-values. Hence, they are likely to be under diversifying selection. On the other extreme, codon 69 is only identified by REL, while FEL assigned *dN<dS* (negative selection, but not significant), and SLAC assigns a very large p-value for positive selection; this discordance does not inspire confidence!

It also helps to find confirmatory evidence for why detected sites may be under positive selection (e.g. based on the structure of the protein or *in vitro* experiments). For example, codons 154,156 and 157 reside in the previously characterized antigenic domain of hemagluttinin (***17***), while codon 172 is involved in the formation of a new glycosylation site - an immune evasion mechanism (***18***).

3.9. Lineage-specific selection.

If sequences in the alignment come from different selective environments, *dN*/*dS* may vary from branch to branch in the phylogenetic tree. Consider 9 HIV-1 envelope sequences isolated from two patients (***19***) where one (the source, 5 sequences), transmitted the virus to the other (the recipient, 4 sequences), available for download in NEXUS format from http://www.datamonkey.org/data/env.nex (download this alignment to a text file on your computer). The evolution of HIV envelope protein is strongly influenced by selective pressures exerted by the humoral immune response mounted by the host, and in our example is subject to three potentially different se-

lective environments: the source, the recipient, and the transmission period represented by the branch separating the sequences from each host. We run the GA Branch analysis (*see* **Note 9**) to demonstrate how Datamonkey can be used to segregate all branches into a smaller number of different selective regimes. First, upload the alignment, perform model selection (HKY85 should be selected) and start the GA Branch analysis from the model setup page (**Fig. 3**).

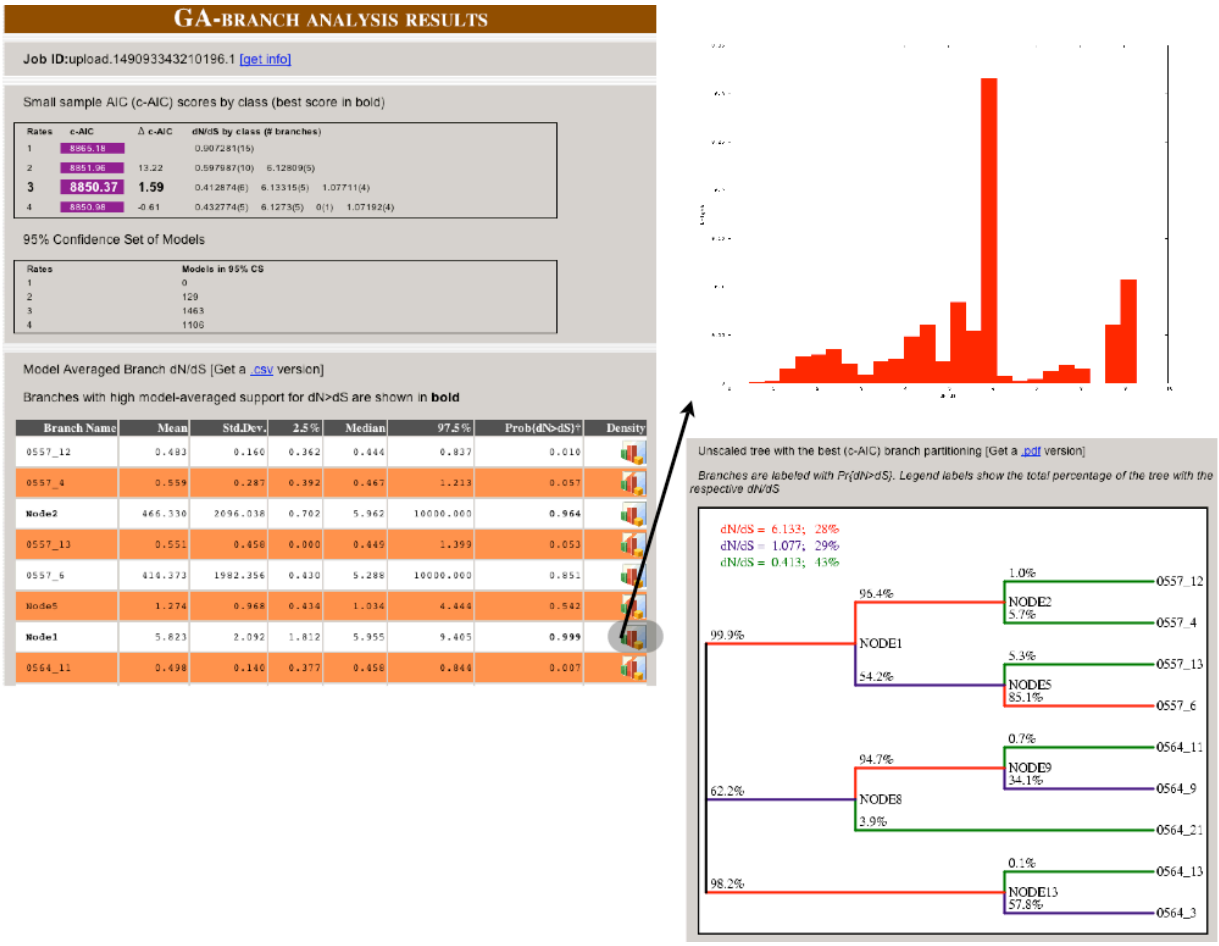


Figure 8 Lineage-specific selection in a two-patient HIV-1 envelope sample.

The resulting page (**Fig. 8**) shows that the data support 3 rate classes, with a large number of models, which can credibly describe the data (over 2500 in the 95% confidence set). The summary table for model-averaged *dN*/*dS* shows statistical de-

scription for estimated *dN*/*dS* at each branch. For instance, Node1, which is the internal branch (see tree plot in **Fig. 8**) representing the transmission event, has *dN*>*dS* with high degree (over 0.999 probability support) of confidence, as also evidenced by the histogram of *dN*/*dS* tabulated from all credible models (weighted appropriately, see (***8***)). The tree plot is a useful visual representation of the results: each tree branch is labeled with percent support for positive selection along that branch (4 branches have over 95% support), while the coloring reflects the selection pattern according to the best fitting model.

As expected, there is evidence of complex and variable selective pressures on the envelope of HIV-1 in this sample, with strong support of positive selection along some (but not all) branches representing the evolution in each patient, as well as along the connecting (transmission) branch (*see* **Note 4**).

3.10. Selection in the presence of recombination.

The presence of recombinants among aligned sequences frequently makes it impossible to represent the evolutionary history of the sample as a single phylogenetic tree. Most traditional selection detection techniques (***19-21***) assume a single phylogeny, and can be misled if recombinants are present, *e.g.* (***9, 10***). Datamonkey can avoid this shortcoming by first screening the alignment to locate all non-recombinant fragments (partitions) using the GARD method (***9***), and then allowing each partition to have its own phylogenetic tree. We demonstrate this approach on an alignment of 16 HIV-1 polymerase genes sampled from the Democratic Republic of Congo – an area where multiple divergence variants of HIV are in common circu-

lation (***22***). When a single host is infected with multiple HIV-1 viruses, this creates the potential for the generation and transmission of recombinant HIV variants, because of very high recombination rates in multiply infected cells (***23***).

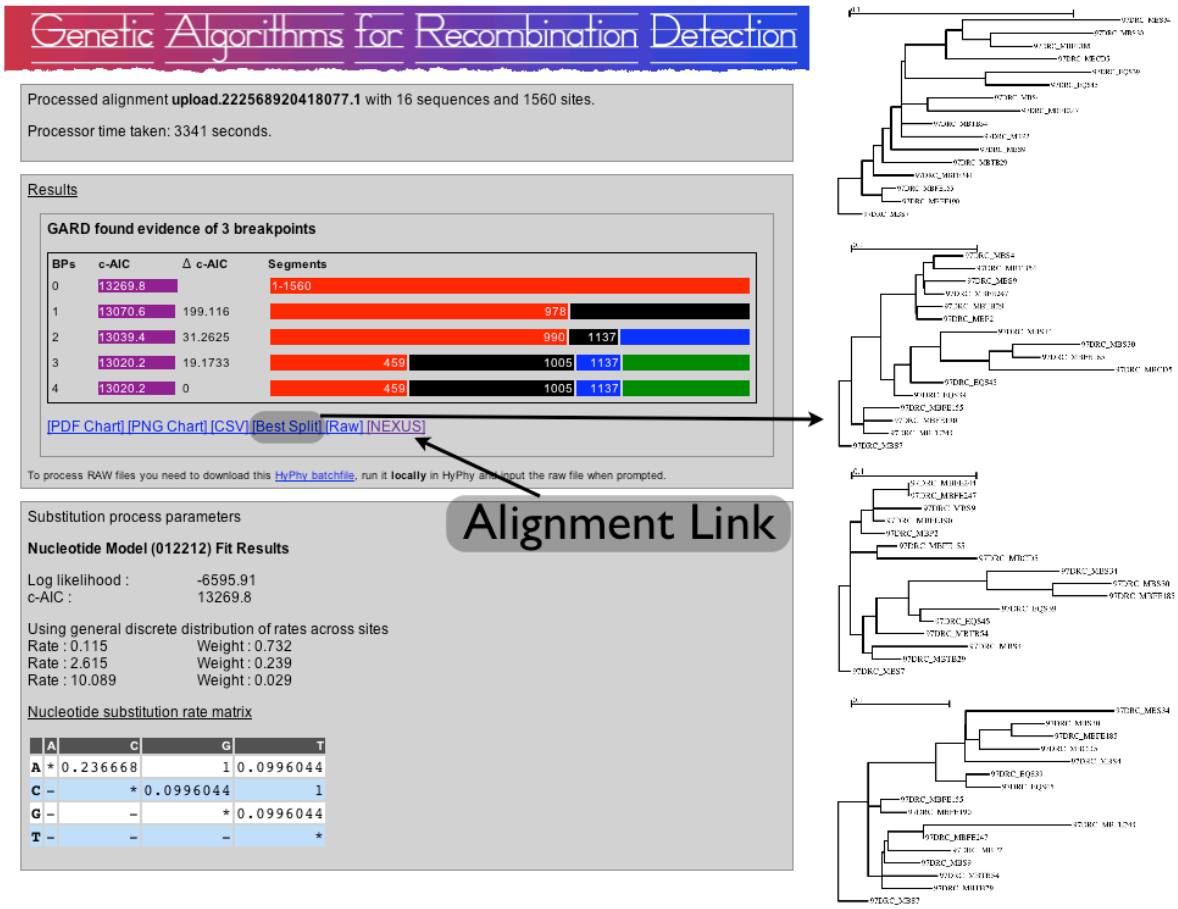


Figure 9 GARD recombination screen on HIV-1 pol alignment.

Download the file from http://www.datamonkey.org/data/pol.nex. When screened with GARD (http://www.datamonkey.org/GARD), this alignment shows evidence of 3 recombination breakpoints (**Fig. 9**). As one of outputs of a GARD analysis (which we encourage the interested users to run), there is NEXUS format file, which includes the information about all non-recombinant partitions (ASSUMPTIONS block), and the trees inferred for each partition.

This information will be recognized by Datamonkey and incorporated into the analyses (for instance the data upload page will show that 4 partitions were read, and what they were). The rest of the analyses can be carried out exactly as we did previously with a non-recombinant example alignment.

As an exercise in the effect of recombination, compare the multiple-segment analysis with the results on the same alignment assuming a single phylogeny (http://www.datamonkey.org/data/pol-1.nex). In many cases the difference is not dramatic, but noticeable nonetheless (**Fig. 10**). For example, both analyses agree of codon 476, the evidence for selection at codon 273 is stronger when multiple trees are allowed, codon 371 is no longer detected as positively selected when multiple trees are considered.

Single tree

Found **10** positively selected sites (at least one method)

Codon	SLAC dN-dS	SLAC p-value	FEL dN-dS	FEL p-value	REL dN-dS	REL Bayes Factor	Consensus	Additional Information
60	1.082	0.446	0.062	0.977	1.556	159.060		[Codons][AA][Counts]
139	5.106	0.176	1.374	0.582	1.946	527.770		[Codons][AA][Counts]
222	3.093	0.381	1.390	0.490	1.755	5755.920		[Codons][AA][Counts]
272	5.396	0.127	1.798	0.259	1.986	337.832		[Codons][AA][Counts]
273	4.600	0.166	1.832	0.075	1.034	32.112		[Codons][AA][Counts]
344	1.458	0.511	2.855	0.111	1.839	1119.460		[Codons][AA][Counts]
371	4.068	0.131	1.419	0.067	0.377	22.720		[Codons][AA][Counts]
433	5.151	0.203	2.430	0.330	1.987	1106.420		[Codons][AA][Counts]
456	1.649	0.530	3.006	0.106	1.999	2939.530		[Codons][AA][Counts]
476	8.066	0.090	6.585	0.081	2.222	40444.300		[Codons][AA][Counts]

Multiple trees

Found **14** positively selected sites (at least one method)

Codon	SLAC dN-dS	SLAC p-value	FEL dN-dS	FEL p-value	REL dN-dS	REL Bayes Factor	Consensus	Additional Information
35	-0.074	0.695	-0.310	0.928	1.419	380.281		[Codons][AA][Counts]
63	1.226	0.547	0.009	0.998	1.404	1072.030		[Codons][AA][Counts]
139	6.520	0.196	2.022	0.680	1.532	5742.190		[Codons][AA][Counts]
221	-0.654	0.737	-1.176	0.700	1.048	100.643		[Codons][AA][Counts]
222	3.432	0.309	1.955	0.495	1.490	17790.300		[Codons][AA][Counts]
234	1.633	0.597	0.318	0.933	1.496	616.356		[Codons][AA][Counts]
272	5.477	0.178	2.613	0.321	1.604	5779.450		[Codons][AA][Counts]
273	4.845	0.205	2.800	0.089	1.369	143.799		[Codons][AA][Counts]
310	0.005	0.679	-1.219	0.658	0.791	50.458		[Codons][AA][Counts]
344	2.404	0.442	3.002	0.140	1.609	6029.640		[Codons][AA][Counts]
350	3.861	0.259	1.281	0.072	-0.184	2.852		[Codons][AA][Counts]
433	4.006	0.175	1.467	0.322	1.611	4206.950		[Codons][AA][Counts]
456	-0.330	0.698	1.120	0.209	1.650	2519.710		[Codons][AA][Counts]
476	6.274	0.090	4.243	0.093	1.630	32027.300		[Codons][AA][Counts]

Figure 10 Effect of including multiple trees on the detection of sites under selection (HIV-1 pol alignment).

4. Notes.

1. The difference is used in place of a more common ratio *dN*/*dS*, because *dS* could be 0 for some sites, rendering the ratio infinite.

2. Datamonkey.org has been developed and tested using Mozilla based browsers (Firefox, Camino) and the Safari browser. While every attempt has been made to write standard compliant HTML and JavaScript code, certain compatibility issues may exist (e.g. when using Internet Explorer). Certain features require that JavaScript be enabled.

3. Because comparative methods estimate relative rates of synonymous and non-synonymous substitution, substantial sequence diversity is needed for reliable inference. For example when Suzuki and Nei (***24***) applied a REL-type method to a very low divergence (1 or 2 substitutions per sequence along a star phylogeny) sample of the Human T-lymphotropic virus (HTLV), they found that the method performed poorly. Yang and colleagues (***25, 26***) have suggested that the total length of the phylogenetic tree should be at least one expected substitution per codon site, but this is merely a guideline, not a requirement. However, sequences that are too divergent could lead to *saturation*, i.e. the inability to reliably infer branch lengths and substitution parameters.

4. For example, comparative methods should not be applied to the detection of selective sweeps (rapid replacement of one allele with a more fit one, resulting in a homogeneous population), unless sequences sampled prior to and following the

selective sweep are included in the sample. We refer an interested reader to (***5***) for further insight.

5. The model test procedure is based on repeated likelihood ratio tests between nested model, and AIC comparisons for non-nested models. Akaike's Information Criterion - a goodness-of-fit criterion that rewards the model for higher log-likelihood score (logL) but penalizes it for each additional parameter (p) as follows: AIC = -2logL+2p. The model with the lowest AIC explains the data best. See (***27***) for an excellent treatment on model selection in the phylogenetic context.

6. If JavaScript is enabled in your browser, selecting an analysis will automatically hide all the inapplicable options.

7. Readers interested in technical details should see (***28***) for discussion and a practical example of determining an appropriate significance level in the context of detecting sites under selection.

8. The binomial distribution provides a tractable approximation to the expected proportions of synonymous and non-synonymous substitutions at a site assuming no selection (see http://www.math.cornell.edu/~durrett/sg/sgnote0123.pdf for technical details). Our simulation studies (unpublished) suggest that the binomial approximation is adequate for a wide range of scenarios.

9. The GA branch analysis (***8***) considers a large number of potential models, where each of the branches is allocated to one of several selective regimes, evaluates the fitness of each model by its small sample AIC score and uses a genetic algo-

rithms to evolve the population of potential models towards the best fitting one. The unique benefit of this approach is that the information from all models can be combined (weighted by the credibility of each model) to avoid incorrect statistical inference due to model mis-specification. The main advantage of the GA approach, as opposed to the alternative branch-site family (***21***) of tests is that the GA procedure will automatically mine the data in search of selection patterns, whereas in branch-site tests, one has to select “interesting” branches a priori, potentially oversimplifying or misleading the inference procedure (***5***).

References

1. Nielsen, R., and Yang, Z. (1998) *Genetics* **148,** 929-36.
2. Yang, Z. H., Nielsen, R., Goldman, N., and Pedersen, A. M. K. (2000) *Genetics* **155,** 431-49.
3. Kosakovsky Pond, S. L., and Frost, S. D. W. (2005) *Bioinformatics* **21,** 2531-33.
4. Kosakovsky Pond, S. L., Frost, S. D. W., and Muse, S. V. (2005) *Bioinformatics* **21,** 676-9.
5. Kosakovsky Pond, S. L., Poon, A. F. Y., and Frost, S. D. W. (2007) *in* "Phylogenetic Handbook" (Lemey, P., and Pybus, O., Eds.), pp. (in press; preprint available at http://www.hyphy.org/pubs/hyphybook2007.pdf), Cambridge University Press.
6. Kosakovsky Pond, S. L., and Frost, S. D. W. (2005) *Mol Biol Evol* **22,** 1208-22.
7. Kosakovsky Pond, S. L., Frost, S. D. W., Grossman, Z., Gravenor, M. B., Richman, D. D., and Brown, A. J. L. (2006) *PLoS Computational Biology* **2,** e62.
8. Kosakovsky Pond, S. L., and Frost, S. D. W. (2005) *Mol Biol Evol* **22,** 478-85.
9. Kosakovsky Pond, S. L., Posada, D., Gravenor, M. B., Woelk, C. H., and Frost, S. D. W. (2006) *Mol Biol Evol* **23,** 1891-901.
10. Shriner, D., Nickle, D. C., Jensen, M. A., and Mullins, J. I. (2003) *Genet Res* **81,** 115-21.
11. Scheffler, K., Martin, D. P., and Seoighe, C. (2006) *Bioinformatics* **22,** 2493-9.
12. Saitou, N., and Nei, M. (1987) *Mol Biol Evol* **4,** 406-25.
13. Tamura, K., and Nei, M. (1993) *Mol Biol Evol* **10,** 512--26.
14. Kosakovsky Pond, S. L., and Frost, S. D. W. (2005) *Mol Biol Evol* **22,** 223-34.
15. Hasegawa, M., Kishino, H., and Yano, T. A. (1985) *J Mol Evol* **22,** 160-74.
16. Yang, Z. H. (1997) *Computer Applications In The Biosciences* **13,** 555-56.
17. Caton, A. J., Brownlee, G. G., Yewdell, J. W., and Gerhard, W. (1982) *Cell* **31,** 417-27.
18. Perdue, M. L., and Suarez, D. L. (2000) *Vet Microbiol* **74,** 77-86.
19. Nielsen, R., and Yang, Z. H. (1998) *Genetics* **148,** 929-36.
20. Suzuki, Y., and Gojobori, T. (1999) *Mol Biol Evol* **16,** 1315-28.
21. Yang, Z., and Nielsen, R. (2002) *Mol Biol Evol* **19,** 908-17.
22. Vergne, L., Peeters, M., Mpoudi-Ngole, E., Bourgeois, A., Liegeois, F., Toure-Kane, C., Mboup, S., Mulanga-Kabeya, C., Saman, E., Jourdan, J., Reynes, J., and Delaporte, E. (2000) *J Clin Microbiol* **38,** 3919-25.
23. Posada, D. (2002) *Mol. Biol. Evol.* **19,** 708-17.
24. Suzuki, Y., and Nei, M. (2004) *Molecular Biology And Evolution* **21,** 914-21.
25. Anisimova, M., Bielawski, J. P., and Yang, Z. H. (2002) *Molecular Biology And Evolution* **19,** 950-58.
26. Anisimova, M., Bielawski, J. P., and Yang, Z. H. (2001) *Molecular Biology And Evolution* **18,** 1585-92.
27. Posada, D., and Buckley, T. R. (2004) *Syst Biol* **53,** 793-808.
28. Sorhannus, U., and Kosakovsky Pond, S. L. (2006) *J Mol Evol* **63,** 231-9.