



METHOD ARTICLE

REVISED Measuring evolutionary rates of proteins in a structural context [version 2; referees: 4 approved]

Dariya K. Sydykova¹, Benjamin R. Jack ¹, Stephanie J. Spielman², Claus O. Wilke ¹

¹Department of Integrative Biology, The University of Texas at Austin, Austin, TX, 78712, USA

²Institute for Genomics and Evolutionary Medicine, Temple University, Philadelphia, PA, 19122, USA

v2 First published: 16 Oct 2017, 6:1845 (doi: 10.12688/f1000research.12874.1)
 Latest published: 09 Feb 2018, 6:1845 (doi: 10.12688/f1000research.12874.2)

Abstract

We describe how to measure site-specific rates of evolution in protein-coding genes and how to correlate these rates with structural features of the expressed protein, such as relative solvent accessibility, secondary structure, or weighted contact number. We present two alternative approaches to rate calculations: One based on relative amino-acid rates, and the other based on site-specific codon rates measured as *dN/dS*. We additionally provide a code repository containing scripts to facilitate the specific analysis protocols we recommend.

Open Peer Review

Referee Status:

	Invited Referees			
	1	2	3	4
REVISED				
version 2 published 09 Feb 2018		report		
		↑		
version 1 published 16 Oct 2017				
	report	report	report	report

- 1 **Ugo Bastolla** , Autonomous University of Madrid, Spain
- 2 **Yu Xia**, McGill University, Canada
Avital Sharir-Ivry, McGill University, Canada
- 3 **David D Pollock**, University of Colorado School of Medicine, USA
- 4 **Mario dos Reis** , Queen Mary University of London, UK

Discuss this article

Comments (0)

Corresponding author: Claus O. Wilke (wilke@austin.utexas.edu)

Author roles: **Sydykova DK:** Conceptualization, Methodology, Resources, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Jack BR:** Conceptualization, Methodology, Resources, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Spielman SJ:** Conceptualization, Methodology, Resources, Software, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Wilke CO:** Conceptualization, Funding Acquisition, Methodology, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

How to cite this article: Sydykova DK, Jack BR, Spielman SJ and Wilke CO. **Measuring evolutionary rates of proteins in a structural context [version 2; referees: 4 approved]** *F1000Research* 2018, **6**:1845 (doi: [10.12688/f1000research.12874.2](https://doi.org/10.12688/f1000research.12874.2))

Copyright: © 2018 Sydykova DK *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution Licence](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Grant information: This work was supported by National Science Foundation Cooperative (agreement no. DBI-0939454; BEACON Center), National Institutes of Health (grant R01 GM088344), and Army Research Office (grant W911NF-12-1-0390).

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

First published: 16 Oct 2017, **6**:1845 (doi: [10.12688/f1000research.12874.1](https://doi.org/10.12688/f1000research.12874.1))

REVISED Amendments from Version 1

We have made numerous changes and improvements throughout the manuscript. In particular, we have made changes to step 7 in Protocol 2, specifically: (i) we have modified the step 7 title from "Change dN/dS for conserved sites" to "Calculate site-wise dN/dS "; (ii) we have updated our recommendations for how to calculate dN/dS for fully conserved sites and for sites with only one non-gap position; (iii) we have explained why we use an amino acid alignment to find conserved sites as opposed to a codon alignment. In the process of the revision, we have found that the new version of HyPhy does not tolerate the characters "!" and "." in the tree file. We have therefore added a new note that describes this issue and that provides a script to circumvent this problem. We have also added a newly available reference to HyPhy's LEISR method, which is part of the protocol to measure relative amino-acid rates. Lastly, we have addressed reviewers' comments and suggestions. Our responses contain detailed explanations on further modifications we have made to version 1.

See referee reports

Introduction

Different sites within a protein-coding gene evolve at different rates^{1,2}. This evolutionary rate heterogeneity across protein sites results from a complex interplay of both functional and structural constraints³. For example, residues that are critical to a given protein's function, such as residues involved in enzymatic activity, protein-protein interactions, or protein-ligand interactions, tend to evolve more slowly than do other residues in the protein⁴⁻¹⁰. In addition, a given protein's structure plays a major role in shaping its evolutionary rates, due to the overarching evolutionary constraint exerted by the imperative for a protein to stably fold. Structurally-important protein residues (namely residues in the protein core) tend to be highly conserved and evolve very slowly, whereas residues with a relatively minor influence on structure (namely surface residues) tend to evolve more rapidly^{4,9,11-19}.

To study evolutionary conservation in a structural context, we require methods to (i) measure evolutionary rates at individual sites in a protein alignment, (ii) map those rates onto the protein structure, and (iii) quantify site-level structural properties. Here, we describe in detail how to perform these three steps, considering a few commonly used alternatives at both steps (i) and (iii). In addition, we provide extensive notes highlighting specific technical issues and/or describing alternative analysis approaches.

At step (i), we demonstrate how evolutionary rates can be measured using either amino-acid or codon data. For amino-acid data, we consider relative amino-acid rates, i.e., rates of evolutionary variation normalized by the mean of the rate in the entire protein^{10,20}. For codon data, we consider site-specific dN/dS values. These are site-specific rates of nonsynonymous variation normalized by (in this case, the whole-gene) rates of synonymous variation^{21,22}.

At step (iii), we discuss two related but somewhat distinct structural measures. First, we consider the solvent accessibility, which measures the extent to which a site is exposed to the solvent environment. Specifically, we consider the relative solvent

accessibility (RSA)²³, which is the solvent accessibility of a residue in a structure normalized by the maximum possible solvent accessibility of that residue in a Gly-X-Gly tripeptide. Second, we consider the packing density, which measures the proximity to and number of neighboring residues. Specifically, we consider the side-chain weighted contact number (WCN)¹⁹, which is calculated relative to the geometric center of the residue side-chain atoms and employs an inverse-square weighting term.

Materials

Below we list the software packages needed to perform the analysis. Please download the most recent version of each software, unless a specific version is specified in the text. The links provided contain instructions for installing and testing the software. All analyses we present assume that these software packages have been installed and are available in your path.

1. HyPhy (see Note 1)

HyPhy is a general-purpose software platform for inference in a phylogenetic framework²⁴. To install, either clone the HyPhy git repository to your desired directory, or download the latest release. The HyPhy repository can be found at <https://github.com/veg/hyphy.git>. Follow the instructions available from <http://hyphy.org/installation> to install HyPhy. Importantly, ensure that you are installing version 2.3.8 or above.

2. MAFFT

MAFFT is a program for generating multiple sequence alignments²⁵. Download MAFFT from <http://mafft.cbrc.jp/alignment/software/>.

3. RAxML

RAxML is a tool for phylogenetic inference using maximum likelihood²⁶. Clone the RAxML repository to a local directory. The RAxML git repository can be found at <https://github.com/stamatak/standard-RAxML>. Analyses presented here utilize the `raxmlHPC-SSE3` executable, which can be compiled with `Makefile.SSE3.gcc` or `Makefile.SSE3.mac`. Note that this executable does not allow threading. (See Note 2 for information on how to enable threading.)

4. mkDSSP

mkDSSP is a tool that calculates solvent accessibilities and parses secondary structure assignments from a PDB input file into a standardized format²⁷. This format follows that of the entries in the DSSP database²⁸. Download the mkDSSP software from <https://slackbuilds.org/repository/14.2/academic/mkDSSP/>.

5. Python

Download python from <https://www.python.org/downloads/>.

6. Biopython

Biopython is a python library for computational molecular biology²⁹. Download Biopython from <http://biopython.org/wiki/Download>.

Biopython has several dependencies that also need to be installed. You can find the information about installing the dependencies in the link provided.

7. argparse

argparse is a python module providing user-friendly command-line interfaces. We use argparse in most of our custom python scripts. Install argparse using the link <https://pypi.python.org/pypi/argparse>.

8. pandas

pandas is a python module for data manipulation and analysis. You can download pandas from <https://pandas.pydata.org/getpandas.html>.

9. R

Download R from <https://www.r-project.org/>. We recommend to use RStudio to execute and edit R scripts. RStudio can be installed from <https://www.rstudio.com/>. We will use R for data visualization. Our scripts require the packages `dplyr`, `readr`, `cowplot`, and their dependencies. You can install an R package by typing the command `install.packages("dplyr")` (for installing `dplyr`) in the R shell. By default, this command will also install any dependencies needed for the package to work.

10. Custom scripts (*see Note 3*)

All our custom python, R, and HyPhy scripts can be found at: <https://github.com/claustwilke/proteinER/tree/master/src>.

Protocols

In this section, we provide four separate protocols to (i) measure relative amino acid rates, (ii) measure site-specific codon evolutionary rates (expressed via the metric dN/dS), (iii) measure structural quantities such as RSA and WCN, and (iv) combine the measured quantities into a combined analysis. To provide an example, we demonstrate all four protocols on an empirical dataset consisting of mammalian orthologs of histamine receptor 1 (HRH1; ENST00000438284) and an accompanying HRH1 PDB structure, 3rzc³⁰. This dataset was originally analyzed by Spielman and Wilke³¹.

Throughout, we assume that we are working on UNIX-like command line interface. We recommend that a user is comfortable with command execution and syntax, which includes flags, arguments, and directories. No prior knowledge of any of the listed software is essential. For python and R scripts, we provide detailed description for each script's function. As such, it is not strictly necessary that a user knows python or R to execute our pipeline. However, if more detailed understanding of the custom scripts is

desired, a user should be familiar with python and R. For your convenience, we have provided a git repository at <https://github.com/claustwilke/proteinER/> that contains the input and output files used in each step.

Our overarching strategy throughout this work is to first infer a given measurement (e.g., dN/dS or RSA) for each site in the multiple sequence alignment or protein structure. To compare the different measurements, we then map them all to columns in the multiple sequence alignment.

Protocol 1: Measuring relative amino-acid rates

The input and output files used in this section can be found at: https://github.com/claustwilke/proteinER/tree/master/measuring_aa_rates.

1. Collect amino acid sequences

One of the most popular methods to collect sequences is BLAST^{32,33}. To search for orthologous sequences in the NCBI database, determine a query sequence, on which BLAST will base its homology search. The BLAST output will specify the number and percent of sites with matches, near matches, and no matches. BLAST refers to these as identities, positives, and gaps, respectively. We recommend the specific algorithm PSI-BLAST³³ if one is interested in collected amino-acid sequences (as opposed to nucleotide). For either data type, we recommend that users specify that BLAST query NCBI "RefSeq" (reference sequence)³⁴, which has been heavily curated to contain only nonredundant and reliably annotated sequences. Aside from BLAST, many other approaches, including collecting orthologs from databases such as ENSEMBL³⁵ or UniProt³⁶, are also suitable for this step. Regardless of the approach taken, we recommend that a final dataset contain at least 20 sequences, with more being preferable, to achieve reliable evolutionary rate estimates.

2. Align sequences with MAFFT (*see Note 4*)

Store all of the sequences you wish to align into one FASTA-formatted file. The FASTA format contains two pieces of information for each sequence: the sequence ID preceded by a ">" sign and followed by a new line, and then the sequence itself. We will use the FASTA file `HRH1_unaligned.fasta` that contains homologous sequences that are not aligned. We align them with the command:

```
mafft --auto --inputorder \
HRH1_unaligned.fasta > \
HRH1_aligned.fasta
```

Arguments above correspond to the following:

- `--auto`, Select the optimal alignment algorithm for the given data.
- `--inputorder`, Output sequences in the same order in which they were provided. Without this option, the order of the sequences in the alignment is arbitrary.

The output file `HRH1_aligned.fasta` will contain the aligned sequences.

3. Infer tree with RAxML (*see Notes 2, 5*)

Using the file with the alignment `HRH1_aligned.fasta`, run RAxML with the following command:

```
raxmlHPC-SSE3 -s HRH1_aligned.fasta \
              -n HRH1_tree \
              -m PROTCATLG \
              -p 12345
```

Arguments above correspond to the following:

- `-s`, The multiple sequence alignment file.
- `-n`, The extension for the outputted tree files. Here, the outputted files will contain `HRH1_tree` in their names.
- `-m`, The model of sequence evolution, in this case the LG amino-acid model³⁷ with RAxML's "CAT" model³⁸ of sequence heterogeneity.
- `-p`, The random number seed initializing this phylogenetic inference. To reproduce the exact phylogeny we have, specify this random seed.

The desired tree file is `RAxML_bestTree.HRH1_tree`.

4. Infer site-wise rates with HyPhy (*see Note 6*)

We calculate rates with the LEISR method in HyPhy³⁹. To run this method, the file `runLEISR.bf` must be edited to specify the directories and file names that will be used in the analysis. Edit these two lines of `runLEISR.bf`

```
"0": "/path/to/HRH1_aligned.fasta",
"1": "/path/to/RAxML_bestTree.HRH1_tree",
```

Here, "0" should specify the full path to the alignment file `HRH1_aligned.fasta`, and "1" should specify the full path to the tree file `RAxML_bestTree.HRH1_tree`.

Run HyPhy with the command

```
HYPHYMP runLEISR.bf
```

An output file `HRH1_aligned.fasta.LEISR.json` is written to the folder that contains the alignment.

5. Parse HyPhy output (*see Note 3*)

For further downstream processing, the HyPhy output file in JSON format needs to be converted to CSV format. The custom python script `parse_LEISR.py` will extract each site's position, rate, and other site-specific inference information from the JSON output file. Parse the JSON file with the command

```
python parse_LEISR.py \
-j HRH1_aligned.fasta.LEISR.json \
-r extracted_HRH1_rates.csv
```

Arguments above correspond to the following:

- `-j`, JSON file outputted by HyPhy.
- `-r`, The output CSV file. If not specified, the output file is `site_rates.csv`.

6. Calculate relative site-wise rates (*see Note 7*)

As discussed by Jack *et al.*¹⁰, we recommend calculating relative evolutionary rates by normalizing inferred site-specific rates by their average. In other words, to compute the relative amino-acid rates, calculate the mean rate of the entire sequence and divide each site's rate by this mean rate. Once normalized, a rate below 1 will indicate a site that evolves more slowly than average. For example, a rate of 0.5 implies that the corresponding site evolves half as quickly as does the average. Similarly, a rate above 1 will indicate a site that evolve more quickly than average. For example, a rate of 2 implies that the corresponding site evolves twice as quickly as does the average.

Protocol 2: Measuring site-specific *dN/dS*

The input and output files used in this section can be found at: https://github.com/clauswilke/proteinER/tree/master/measuring_dNdS.

1. Collect nucleotide sequences

Collect nucleotide sequence using step 1 in Protocol 1, using a nucleotide sequence as the query. Alternatively, the Ensembl database's Biomart tool³⁵ may represent a more reliable approach for collecting strictly protein-coding sequences. By contrast, the UniProt database³⁶ should not be used, as it contains only protein sequences and lacks clear cross-references to the corresponding nucleotide sequences.

2. Translate codon sequences (*see Note 3*)

In this section, both codon and amino-acid sequences are required to perform site-wise rate calculations. Store all of the desired nucleotide sequences into one FASTA file. Use our custom script to convert a codon FASTA file to an amino acid FASTA files. We use the FASTA file `HRH1_unaligned_codon.fasta` that contains homologous nucleotide sequences we wish to translate. Translate with the command:

```
python translate_aln_codon_to_aa.py \
-n HRH1_unaligned_codon.fasta \
-o HRH1_unaligned_aa.fasta
```

Arguments above correspond to the following:

- `-n`, The input file with codon sequences. Both aligned and unaligned sequences are accepted.
- `-o`, The output file with amino acid sequences. If not specified, the script outputs `aa_aln.fasta`. If the

input file contains aligned sequences, the output file will also contain aligned sequences.

3. Align amino acid sequences with MAFFT

Align amino acid sequences using step 2 in Protocol 1.

4. Back-translate the amino acid alignment into a codon alignment (*see Note 3*)

This step requires the original codon sequences and the amino acid alignment. Note that the amino acid alignment is retained, and the script simply inserts corresponding codons in place of amino acids at each column of the alignment. Use this command to back-translate the sequences:

```
python translate_aln_aa_to_codon.py \
  -a HRH1_aligned_aa.fasta \
  -n HRH1_unaligned_codon.fasta \
  -o HRH1_aligned_codon.fasta
```

Arguments above correspond to the following:

- -a, The inputted amino-acid alignment.
- -n, The file of codon sequences. The script accepts either aligned or unaligned sequences.
- -o, The output file to contain the codon alignment. This argument is optional, and, if it is missing, the script outputs a file `codon_aln.fasta`.

5. Infer tree with RAxML

The following step is the same as step 3 in Protocol 1. Use the amino-acid alignment file `HRH1_aligned_aa.fasta` to infer the tree.

6. Infer site-wise rates with HyPhy (*see Note 8*)

To calculate site-wise dN/dS , we use the Fixed Effects Likelihood (FEL) method in HyPhy²¹. To run FEL in HyPhy, the file `runFEL.bf` must be edited to specify the directories and file names that will be used in the analysis. Edit the following two lines of the `runFEL.bf` script:

```
"1": "/path/to/HRH1_aligned_codon.fasta",
"2": "/path/to/RAxML_bestTree.HRH1_tree",
```

Here, "1" should specify the full path to the alignment file `HRH1_aligned_codon.fasta`, and "2" should specify the full path to the tree file `RAxML_bestTree.HRH1_tree`.

Run HyPhy with the following command:

```
HYPHYMP runFEL.bf
```

An output file `HRH1_aligned_codon.fasta.FEL.json` is written to the folder that contains the alignment file.

7. Parse HyPhy output (*see Note 3*)

For further downstream processing, the HyPhy output file in JSON format needs to be converted to CSV format. The custom python script `parse_FEL.py` will extract the site's position, dN (referred to as 'beta' in HyPhy output), dS (referred to as 'alpha' in HyPhy output), and other site information outputted by HyPhy:

```
python parse_FEL.py \
  -j HRH1_aligned_codon.fasta.FEL.json \
  -r extracted_HRH1_dNdS.csv
```

Arguments above correspond to the following:

- -j, JSON file from the FEL analysis.
- -r, The output CSV file. If not specified, the output file is `site_rates.csv`.

8. Calculate site-specific dN/dS (*see Notes 3*)

FEL will calculate dS and dN values for all informative sites. However, the FEL method will assign $dS = 0$ and $dN = 0$ to sites without any synonymous or non-synonymous substitutions, respectively. When calculating dN/dS at these entirely conserved sites, we recommend to use the value $dN/dS = 0$. For sites with only one non-gap residue, FEL will similarly assign both dN and dS a value of 0. For those sites, we also recommend to use the value $dN/dS = 0$.

We provide a custom script that will calculate site-wise dN/dS and will assign $dN/dS = 0$ to such sites. For simplicity, this script checks for conserved sites in amino acid alignments. That the script considers amino acid rather than codon conservation does not influence rate assignments, as $dN/dS = 0$ in both cases of fully conserved amino acids and codons. For sites with substitutions, HyPhy's FEL method, used specifically as recommended here, assigns $dS = 1$. At those sites, our script calculates site-wise dN/dS by simply dividing site's dN by site's dS . The original format of `extracted_HRH1_dNdS.csv` will not be changed.

```
python calc_dNdS.py \
  -a HRH1_aligned_aa.fasta \
  -r extracted_HRH1_dNdS.csv \
  -o processed_HRH1_dNdS.csv
```

Arguments above correspond to the following:

- -a, The amino acid alignment file.
- -r, The CSV file with parsed FEL rates.
- -o, The output CSV file. If not specified, the script outputs `processed_dNdS.csv`.

Protocol 3: Measuring structural features

All structural features in this section are calculated from an example PDB file, `3rze.pdb`³⁰. This PDB file defines the crystal structure of a human histamine receptor 1 (HRH1), whose rates were computed in Protocols 1 and 2, fused to an unrelated lysozyme protein. The lysozyme is required for crystallization, but is not biologically relevant. We have pre-processed the PDB file to exclude residues from the lysozyme protein (residue numbers 1000 and above). The input and output files used in this section can be found at: https://github.com/clauswilke/proteinER/tree/master/measuring_structural_features.

1. Calculate relative solvent accessibility (RSA) from the PDB file (*see Note 3*)

We provide a custom script `calc_rsa.py` that will run the software `mkDSSP`^{27,28}, extract absolute solvent accessibilities from the `mkDSSP` output, and calculate relative solvent accessibilities²³. The first argument is the PDB file, and the second optional argument (`-o 3rze`) is the prefix used for the output files.

```
python calc_rsa.py 3rze.pdb -o 3rze
```

This command will generate two output files: `3rze.asa.txt` containing the raw `mkDSSP` output, and `3rze.rsa.csv` containing RSA values and secondary structure classifications.

2. Calculate weighted contact numbers (WCN) from the PDB file (*see Note 3*)

WCN measures amino acid packing density and may be calculated with respect to either the α -carbon or the geometric center of the side-chain^{19,40}. We provide a custom script that will calculate both types of WCN values, although we strongly recommend using the side-chain WCN values¹⁹. The command line arguments follow the same format as the `calc_rsa.py` script.

```
python calc_wcn.py 3rze.pdb -o 3rze
```

The above command will produce an output file `3rze.wcn.csv` that contains both side-chain and α -carbon WCN values for each position in the input PDB file.

Protocol 4: Combining rates with structural features

The input and output files used in this section can be found at: https://github.com/clauswilke/proteinER/tree/master/map_structural_features.

1. Generate sequence alignment map (*see Note 3*)

To map site specific evolutionary rates to residues in a PDB structure, we first align the sequence of amino acids extracted from the PDB structure to the multiple sequence alignment used for rate inference. We provide a script

that calls MAFFT to align a PDB sequence to a multiple sequence alignment and reformat the output.

```
python make_map.py \
    HRH1_aligned.fasta 3rze.pdb
```

Running the above command produces a CSV file `3rze.map.csv` with four columns. The first and second columns contain the numbered position of a given residue in the *alignment* used for rate inference and the numbered position of a given residue in the *PDB structure*, respectively. The numbered positions in the second column are obtained directly from the PDB input file and may therefore include PDB insertion codes (*see Note 10*). The third and fourth columns contain the single-letter amino acid present in the PDB structure and the PDB chain, respectively. If an amino acid is in the alignment but not in the PDB structure, the PDB position is assigned a value of NA. Likewise, if an amino acid is in the PDB structure but not the alignment, the alignment position is assigned NA.

2. Map rates to structural features (*see Note 3*)

After mapping the alignment used for rate inference to the sequence of the PDB structure, we merge rates with structural features for each residue. We provide a script that uses the map generated above to combine rates and structural features into a single CSV.

```
python map_features.py 3rze.map.csv \
    -r processed_HRH1_dNdS.csv \
    extracted_HRH1_rates.csv \
    -f 3rze.rsa.csv 3rze.wcn.csv
```

Arguments above correspond to the following:

- The input file containing a map between the alignment residue positions and the structure residue positions.
- `-r`, The rates files.
- `-f`, The structural feature files.
- `-o`, The CSV output file. If not specified, the script outputs a file `<pdb_id>.rates_features.csv`. Here, `<pdb_id>` is the name of the PDB ID used to make the map file.

The output from this command provides all the data needed to compute correlations between rates and structural features and corresponding visualizations, as, for example, shown in [Figure 1](#).

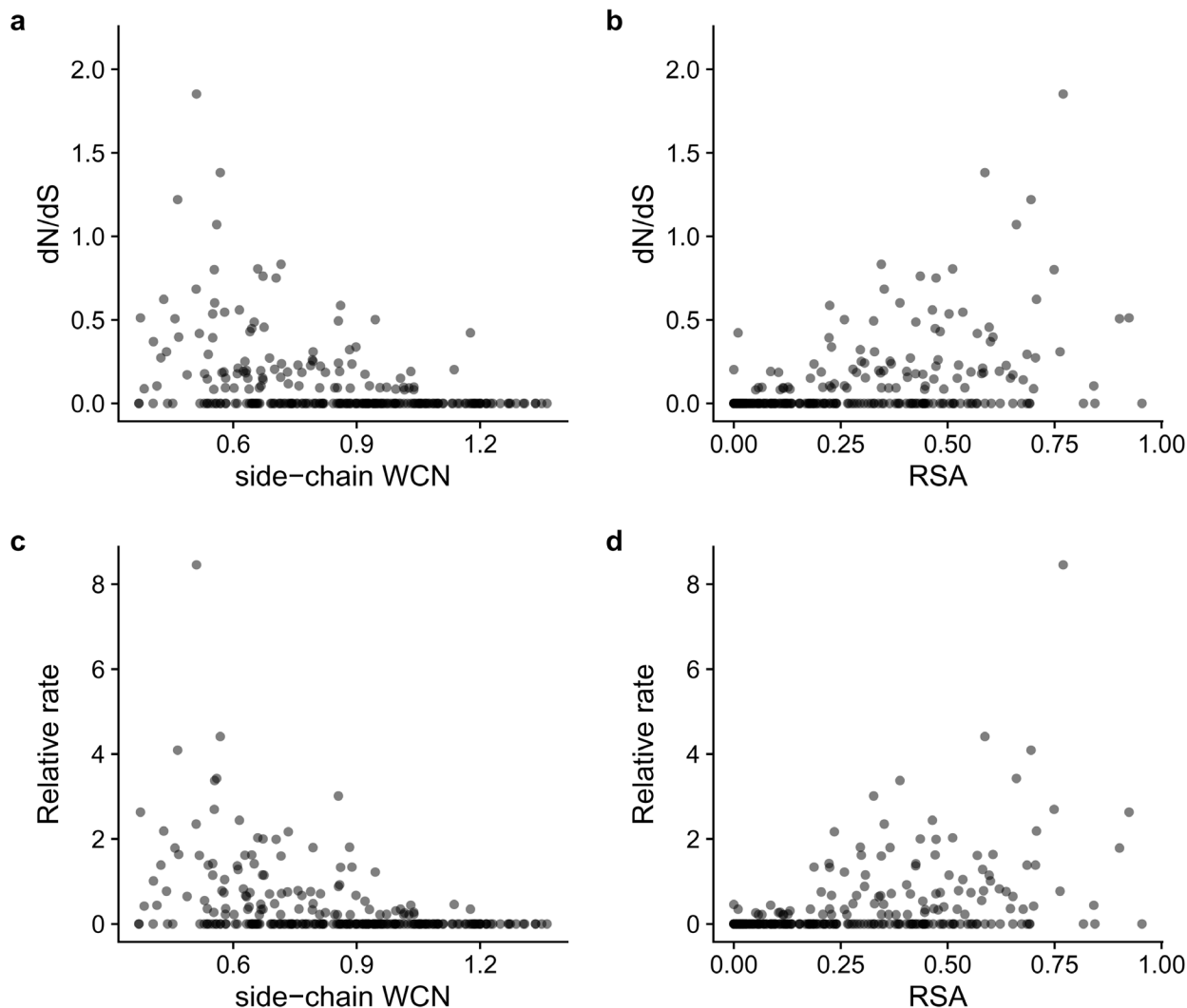


Figure 1. Amino-acid packing density and solvent accessibility correlate with site specific evolutionary rates. (a–d) Each point represents a residue in the structure of the HRH1 protein (PDB: 3rze). The Pearson correlation coefficients r between structural features (RSA or WCN) and rates (dN/dS or amino acid) are as follows for each panel: (a) -0.39 , (b) 0.43 , (c) -0.39 , (d) 0.42 .

Conclusions

We have provided four separate protocols that jointly enable the analysis of protein evolutionary rates in a structural context. The first two protocols measure site-specific evolutionary rates from multiple-sequence alignments, either at the amino-acid or the codon level. In practice, a given study will generally employ only one of these two protocols. The third protocol quantifies local characteristic of a protein structure, such as relative solvent accessibility or weighted contact number, and the fourth protocol maps the structural quantities and evolutionary rates to one another.

In the first two protocols we used two different methods to calculate evolutionary rates, HyPhy's FEL and LEISR approaches. As an alternative method to measuring amino acid evolutionary rates

with LEISR, we presented a brief pipeline in the notes that uses Rate4Site software. Other methods for calculating evolutionary rates have been provided by Rodrigue *et al.* and Tamuri *et al.*^{41–43}, whose methods infer codon-level site-wise evolutionary rates in a population genetics framework. Other relevant works by Jones *et al.* and by Halpern and Bruno cover theoretical approaches to infer site-wise rates^{44,45}.

In sum, we hope that the protocols presented here will be useful for further research into disentangling structural and functional constraints on protein evolution.

Notes

1. The minimum required HyPhy version for FEL dN/dS inference is 2.3.3. The minimum required version for

relative amino-acid rate inference with LEISR is 2.3.8. In addition, for users who feel more comfortable working in a python scripting environment than with HyPhy directly from the command line, we note that users can accomplish all HyPhy analyses described here (including parsing HyPhy output) through the package “phyphy”⁴⁶, available from <https://github.com/sjspielman/phyphy>.

- To thread RAXML, compile the `raxmlHPC-PTHREADS-SSE3` executable with `Makefile.SSE3.PTHREADS.gcc` or `Makefile.SSE3.PTHREADS.mac`. The options to call RAXML stay the same. Add the option `-T` to thread, and run RAXML with

```
raxmlHPC-PTHREADS-SSE3 -T 48 \
    -s HRH1_aligned.fasta \
    -n HRH1_tree \
    -m PROTCATLG \
    -p 12345
```

- All of our custom python scripts provide documentation when called with the options `-h` or `--help`. For example, to view the documentation for the script `calc_rsa.py`, run the command

```
python calc_rsa.py -h
```

The script’s use and required input files will be described in the documentation. Additionally, where applicable, the documentation also provides a description of the information stored in the output files.

- HyPhy will not properly read data if either a pipe character (“|”) or a period (“.”) is present in the input alignment/phylogeny sequence IDs. We recommend to change these characters, consistently in both the alignment and phylogeny, to “_”, which HyPhy does accept. We provide a custom python script to execute this step in the alignment, which will in turn propagate to a tree reconstructed from this alignment:

```
python format_aln_id.py \
    -a HRH1_aligned.fasta \
    -o HRH1_aligned_reformatted.fasta
```

Arguments above correspond to the following:

- `-a`, The input file containing sequences in the FASTA format. Both aligned and unaligned sequences are accepted.
- `-o`, The output file with reformatted sequence IDs. If not specified, the output file is `reformatted_aln.fasta`.

- RAXML can also infer trees from nucleotide sequence data in addition to amino-acid data. Importantly, if the analyzed sequences don’t show much divergence at the amino-acid level, then trees inferred from nucleotide sequences may yield better rate predictions. To infer a tree from nucleotide data with RAXML, issue the following command (specifically, `-m PROTCATLG` has been changed to

a GTR nucleotide model with CAT heterogeneity, `-m GTRCAT`):

```
raxmlHPC-SSE3 -s HRH1_aligned.fasta \
    -n HRH1_tree -m GTRCAT \
    -p 12345
```

Furthermore, if the dataset of interest contains fewer than 50 taxa, a discrete Gamma distribution should be used rather than the CAT model for modeling rate heterogeneity³⁸. To specify this model, simply replace the phrase CAT with GAMMA: For amino-acids, use the model specification `-m PROTGAMMALG`, and for nucleotides use the model specification `-m GTRGAMMA`.

- As an alternative method to infer site-wise amino acid rates one can use Rate4Site (or its accompanying web-server, ConSurf⁴⁷). Rate4Site is a tool for inferring site-wise evolutionary rates in amino acid sequences²⁰. Download Rate4Site from <https://www.tau.ac.il/~itaymay/cp/rate4site.html>. Analyses presented here use Rate4Site downloaded as `rate4site.3.2.source.zip` and compiled with the `Makefile_slow` file.

LEISR, in fact, is based on the Rate4Site algorithm, and these approaches therefore produce virtually identical rates³⁹. However, LEISR provides increased functionality relative to Rate4Site, namely by assigning rates to all alignment positions and by allowing for datasets of arbitrary size. In addition, Rate4Site normalizes rates to standard z-scores, whereas LEISR performs no such normalization. Finally, Rate4Site is available both as a random-effects and fixed-effects implementation, and LEISR adopts the fixed-effects approach. The fixed-effects implementation may be preferable, because random-effects models shrink and/or smooth rate estimates, which can produce undesirable artifacts in the inferred rates.

The options to run Rate4Site may be different for different Rate4Site installation files. We recommend using the `rate4site -h` command to find the proper options for your version, as opposed to using the software’s website.

Run the following command to infer site-wise rates:

```
rate4site -Mw -s HRH1_aligned.fasta \
    -t RAXML_bestTree.HRH1_tree \
    -o HRH1_norm_rates.txt \
    -y HRH1_orig_rates.txt
```

Arguments above correspond to the following:

- `-Mw`, Specify the WAG model of amino-acid evolution (see **Note 9**).
- `-s`, The multiple sequence alignment file.
- `-t`, The input phylogeny.
- `-o`, The output file of *normalized* amino-acid rates.
- `-y`, The output file of *raw* amino-acid rates.

Rate4Site normalizes rates by converting them into standard z-scores. The z-scores are written to `HRH1_norm_rates.txt`. Rate4Site also outputs the raw (unnormalized) scores in `HRH1_orig_rates.txt`. We advise you to use raw scores and to normalize them by the average score in the sequence, as discussed in protocol 1 step 6. Note that Rate4Site also outputs a new tree file `TheTree.txt` and an empty rates file `r4s.res`. These files are not needed for further analysis.

For further downstream processing, the Rate4Site output file needs to be converted to a CSV file. The following command will extract the site's position, amino acid, and Rate4Site score (see **Note 3**).

```
python parse_r4s.py \
    HRH1_orig_rates.txt \
    -o extracted_HRH1_orig_rates.csv
```

Arguments above correspond to the following:

- The Rate4Site output file.
- `-o`, The output CSV file name.

By default, the Rate4Site software will compute rates only for the sites in the first sequence of the alignment file. In other words, Rate4Site will ignore any alignment columns where the site in the first sequence is a gap. To circumvent losing information outputted from Rate4Site, we suggest finding the sequence in the alignment with the fewest gaps and using it as the reference sequence for the output. The reference sequence for Rate4Site can be specified with the option `-a sequence_ID`, where `sequence_ID` is the name of the sequence in a FASTA file provided for rate inference.

7. If you are interested in calculating relative rates in R, our script `make_plots.R` contains code to normalize the rates to the mean of 1. This script reads in the last output file from protocol 4 to plot **Figure 1**. Prior to plotting, the script will normalize rates relative to the gene-wide average.
8. The file `runFEL.bf` implements fixed-effect likelihood (FEL) inference without synonymous rate variation, which is sometimes referred to as a one-rate FEL model. This

parameterization infers one dN value per site and one dS value for the entire sequence²¹. The one-rate FEL model has been found to infer more accurate dN/dS values than models which infer a separate dS at each site²².

9. For reasons that are beyond the scope of this paper, the specific matrix choice has little effect on the final rates, as long as rates are normalized relative to their means as we do here. The underlying reason for this insensitivity to matrix choice is that the available matrices were all derived by pooling data from many sites in many proteins (see e.g. 37), and this pooling yields matrices that are close to uninformative^{48,49}.
10. The residue numbers in PDB files are not strictly sequential or numeric. If multiple residues share the same numeric value, they will be distinguished by a single letter insertion code (e.g. 53A or 53B)⁵⁰. These insertion codes appear when there are several homologous proteins with crystal structures. Generally, each new structure retains the numbering of the earliest crystallized structure to preserve the alignment among structures of homologous proteins. If the new structure contains deletions relative to the original structure, the PDB file will skip residue numbers. If the new structure contains insertions, the PDB file will have residue numbers with insertion codes.

Data and software availability

All information required to reproduce the analysis is provided at <https://github.com/cluswilke/proteinER>. Version 2.0 of this code is archived at <https://doi.org/10.5281/zenodo.1160661>.

Competing interests

No competing interests were disclosed.

Grant information

This work was supported by National Science Foundation Cooperative (agreement no. DBI-0939454; BEACON Center), National Institutes of Health (grant R01 GM088344), and Army Research Office (grant W911NF-12-1-0390).

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

1. Kimura M, Ohta T: **On some principles governing molecular evolution.** *Proc Natl Acad Sci U S A.* 1974; **71**(7): 2848–2852. [PubMed Abstract](#) | [Free Full Text](#)
2. Perutz MF, Kendrew JC, Watson HC: **Structure and function of haemoglobin: II. Some relations between polypeptide chain configuration and amino acid sequence.** *J Mol Biol.* 1965; **13**(3): 669–678. [PubMed Abstract](#) | [Free Full Text](#)
3. Echave J, Spielman SJ, Wilke CO: **Causes of evolutionary rate variation among protein sites.** *Nat Rev Genet.* 2016; **17**(2): 109–121. [PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
4. Dean AM, Neuhauser C, Grenier E, et al.: **The pattern of amino acid replacements in alpha/beta-barrels.** *Mol Biol Evol.* 2002; **19**(11): 1846–1864. [PubMed Abstract](#) | [Publisher Full Text](#)
5. Kimura M, Ohta T: **Mutation and evolution at the molecular level.** *Genetics.* 1973; **73**(Suppl 73): 19–35. [PubMed Abstract](#)

6. Huang YW, Chang CM, Lee CW, *et al.*: **The conservation profile of a protein bears the imprint of the molecule that is evolutionarily coupled to the protein.** *Proteins*. 2015; **83**(8): 1407–1413.
[PubMed Abstract](#) | [Publisher Full Text](#)
7. Mintseris J, Weng Z: **Structure, function, and evolution of transient and obligate protein-protein interactions.** *Proc Natl Acad Sci U S A*. 2005; **102**(31): 10930–10935.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
8. Kim PM, Lu LJ, Xia Y, *et al.*: **Structural determinants of protein evolution are context-sensitive at the residue level.** *Mol Biol Evol*. 2009; **26**(10): 2387–2395.
[PubMed Abstract](#) | [Publisher Full Text](#)
9. Franzosa EA, Xia Y: **Structural determinants of protein evolution are context-sensitive at the residue level.** *Mol Biol Evol*. 2009; **26**(10): 2387–2395.
[PubMed Abstract](#) | [Publisher Full Text](#)
10. Jack BR, Meyer AG, Echave J, *et al.*: **Functional sites induce long-range evolutionary constraints in enzymes.** *PLoS Biol*. 2016; **14**(5): e1002452.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
11. Mirny LA, Shakhnovich EI: **Universally conserved positions in protein folds: reading evolutionary signals about stability, folding kinetics and function.** *J Mol Biol*. 1999; **291**(1): 177–196.
[PubMed Abstract](#) | [Publisher Full Text](#)
12. Zhou T, Drummond DA, Wilke CO: **Contact density affects protein evolutionary rate from bacteria to animals.** *J Mol Evol*. 2008; **66**(4): 395–404.
[PubMed Abstract](#) | [Publisher Full Text](#)
13. Ramsey DC, Scherrer MP, Zhou T, *et al.*: **The relationship between relative solvent accessibility and evolutionary rate in protein evolution.** *Genetics*. 2011; **188**(2): 479–488.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
14. Scherrer MP, Meyer AG, Wilke CO: **Modeling coding-sequence evolution within the context of residue solvent accessibility.** *BMC Evol Biol*. 2012; **12**: 179.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
15. Shahmoradi A, Sydykova DK, Spielman SJ, *et al.*: **Predicting evolutionary site variability from structure in viral proteins: buriedness, packing, flexibility, and design.** *J Mol Evol*. 2014; **79**(3–4): 130–142.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
16. Yeh SW, Liu JW, Yu SH, *et al.*: **Site-specific structural constraints on protein sequence evolutionary divergence: local packing density versus solvent exposure.** *Mol Biol Evol*. 2014; **31**(1): 135–139.
[PubMed Abstract](#) | [Publisher Full Text](#)
17. Yeh SW, Huang TT, Liu JW, *et al.*: **Local packing density is the main structural determinant of the rate of protein sequence evolution at site level.** *Biol Med Res Int*. 2014; **2014**: 572409.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
18. Huang TT, del Valle Marcos ML, Hwang JK, *et al.*: **A mechanistic stress model of protein evolution accounts for site-specific evolutionary rates and their relationship with packing density and flexibility.** *BMC Evol Biol*. 2014; **14**: 78.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
19. Marcos ML, Echave J: **Too packed to change: side-chain packing and site-specific substitution rates in protein evolution.** *PeerJ*. 2015; **3**: e911.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
20. Pupko T, Bell RE, Mayrose I, *et al.*: **Rate4Site: an algorithmic tool for the identification of functional regions in proteins by surface mapping of evolutionary determinants within their homologues.** *Bioinformatics*. 2002; **18** Suppl 1: S71–S77.
[PubMed Abstract](#) | [Publisher Full Text](#)
21. Kosakovsky Pond SL, Frost SD: **Not so different after all: a comparison of methods for detecting amino acid sites under selection.** *Mol Biol Evol*. 2005; **22**(5): 1208–1222.
[PubMed Abstract](#) | [Publisher Full Text](#)
22. Spielman SJ, Wan S, Wilke CO: **A comparison of one-rate and two-rate inference frameworks for site-specific dN/dS estimation.** *Genetics*. 2016; **204**(2): 499–511.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
23. Tien MZ, Meyer AG, Sydykova DK, *et al.*: **Maximum allowed solvent accessibilities of residues in proteins.** *PLoS One*. 2013; **8**(11): e80635.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
24. Pond SL, Frost SD, Muse SV: **HyPhy: hypothesis testing using phylogenies.** *Bioinformatics*. 2005; **21**(5): 676–679.
[PubMed Abstract](#) | [Publisher Full Text](#)
25. Katoh K, Standley DM: **MAFFT multiple sequence alignment software version 7: improvements in performance and usability.** *Mol Biol Evol*. 2013; **30**(4): 772–780.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
26. Stamatakis A: **RAXML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies.** *Bioinformatics*. 2014; **30**(9): 1312–1313.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
27. Kabsch W, Sander C: **Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features.** *Biopolymers*. 1983; **22**(12): 2577–2637.
[PubMed Abstract](#) | [Publisher Full Text](#)
28. Joosten RP, te Beek TA, Krieger E, *et al.*: **A series of PDB related databases for everyday needs.** *Nucleic Acids Res*. 2011; **39**(Database issue): D411–D419.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
29. Cock PJ, Antao T, Chang JT, *et al.*: **Biopython: freely available Python tools for computational molecular biology and bioinformatics.** *Bioinformatics*. 2009; **25**(11): 1422–1423.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
30. Shimamura T, Shiroishi M, Weyand S, *et al.*: **Structure of the human histamine H1 receptor complex with doxepin.** *Nature*. 2011; **475**(7354): 65–70.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
31. Spielman SJ, Wilke CO: **Membrane environment imposes unique selection pressures on transmembrane domains of G protein-coupled receptors.** *J Mol Evol*. 2013; **76**(3): 172–182.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
32. Altschul SF, Gish W, Miller W, *et al.*: **Basic local alignment search tool.** *J Mol Biol*. 1990; **215**(3): 403–410.
[PubMed Abstract](#) | [Publisher Full Text](#)
33. Altschul SF, Madden TL, Schaffer AA, *et al.*: **Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.** *Nucleic Acids Res*. 1997; **25**(17): 3389–3402.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
34. O’Leary NA, Wright MW, Brister JR, *et al.*: **Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation.** *Nucleic Acids Res*. 2016; **44**(D1): D733–D745.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
35. Zerbino DR, Achuthan P, Akanni W, *et al.*: **Ensembl 2018.** *Nucleic Acids Res*. 2018; **46**(D1): D754–D761.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
36. The UniProt Consortium: **UniProt: the universal protein knowledgebase.** *Nucleic Acids Res*. 2017; **45**(D1): D158–D169.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
37. Le SQ, Gascuel O: **An improved general amino acid replacement matrix.** *Mol Biol Evol*. 2008; **25**(7): 1307–1320.
[PubMed Abstract](#) | [Publisher Full Text](#)
38. Stamatakis A: **Phylogenetic models of rate heterogeneity: a high performance computing perspective.** In *Proc of IPDPS2006*. 2006.
[Publisher Full Text](#)
39. Spielman SJ, Kosakovsky Pond SL: **Relative evolutionary rate inference in HyPhy with LEISR.** *PeerJ*. 2018; **6**: e4339.
[Publisher Full Text](#)
40. Yeh SW, Liu JW, Yu SH, *et al.*: **Site-specific structural constraints on protein sequence evolutionary divergence: local packing density versus solvent exposure.** *Mol Biol Evol*. 2014; **31**(1): 135–139.
[PubMed Abstract](#) | [Publisher Full Text](#)
41. Rodrigue N, Philippe H, Lartillot N, *et al.*: **Mutation-selection models of coding sequence evolution with site-heterogeneous amino acid fitness profiles.** *Proc Natl Acad Sci U S A*. 2010; **107**(10): 4629–4634.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
42. Rodrigue N, Lartillot N: **Site-heterogeneous mutation-selection models within the PhyloBayes-MPI package.** *Bioinformatics*. 2014; **30**(7): 1020–1021.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
43. Tamuri AU, dos Reis M, Goldstein RA: **Estimating the distribution of selection coefficients from phylogenetic data using sitewise mutation-selection models.** *Genetics*. 2012; **190**(3): 1101–1115.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
44. Jones CT, Youssef N, Susko E, *et al.*: **Shifting balance on a static mutation-selection landscape: a novel scenario of positive selection.** *Mol Biol Evol*. 2017; **34**(2): 391–407.
[PubMed Abstract](#) | [Publisher Full Text](#)
45. Halpern AL, Bruno WJ: **Evolutionary distances for protein-coding sequences: modeling site-specific residue frequencies.** *Mol Biol Evol*. 1998; **15**(7): 910–917.
[PubMed Abstract](#) | [Publisher Full Text](#)
46. Spielman SJ: **phyphy: Python package for facilitating the execution and parsing of HyPhy standard analyses.** *J Open Source Softw*. 2018; **3**(21): 514.
[Publisher Full Text](#)
47. Ashkenazy H, Abadi S, Martz E, *et al.*: **ConSurf 2016: an improved methodology to estimate and visualize evolutionary conservation in macromolecules.** *Nucleic Acids Res*. 2016; **44**(W1): W344–W350.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
48. Goldstein RA, Pollock DD: **The tangled bank of amino acids.** *Protein Sci*. 2016; **25**(7): 1354–1362.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
49. Echave J, Wilke CO: **Biophysical Models of Protein Evolution: Understanding the Patterns of Evolutionary Sequence Divergence.** *Ann Rev Biophys*. 2017; **46**: 85–103.
[PubMed Abstract](#) | [Publisher Full Text](#)
50. **Protein Data Bank Contents Guide: Atomic Coordinate Entry Format Description.** wwPDB, 2012; Version 3.30.
[Reference Source](#)

Open Peer Review

Current Referee Status:



Version 2

Referee Report 12 February 2018

doi:[10.5256/f1000research.15080.r30718](https://doi.org/10.5256/f1000research.15080.r30718)



Yu Xia , Avital Sharir-Ivry

Department of Bioengineering, Faculty of Engineering, McGill University, Montreal, QC, Canada

All comments have been adequately addressed.

Competing Interests: No competing interests were disclosed.

We have read this submission. We believe that we have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Referee Report 17 November 2017

doi:[10.5256/f1000research.13954.r27040](https://doi.org/10.5256/f1000research.13954.r27040)



Mario dos Reis 

School of Biological and Chemical Sciences, Queen Mary University of London, London, UK

This paper by Sydykova *et al.* is written in tutorial style, with the aim of demonstrating how to use a series of software tools to calculate site-specific rates of evolution in proteins and protein-coding sequences. In brief, the user starts with a molecular sequence alignment as input data, and then proceeds to estimate a phylogenetic tree, site-specific evolutionary rates, and other statistics related to protein structure and evolution.

The things I look for in a good tutorial are clarity and brevity. This manuscript achieves both. Brevity is important as, in my opinion, a tutorial user should not be bogged down with lengthy discussions of theory, caveats, comparisons to other methods, etc. However, what I do expect from a good tutorial is a set of key sentences pointing out to the user where to find the appropriate theory papers, simulation studies, and work by others that may have developed similar methods.

I have the following recommendations:

- Perhaps at the beginning of the protocol section, the authors could clearly state to the user what level of knowledge is expected from them. Does the user need to know how to use the command line? R? Must they have some basic experience of using phylogenetic software? This is important,

specially in this case, where the user base may come from a very mixed background (physical sciences, computer science, chemistry or biology).

- The authors may consider adding a section at the end of the paper with general recommendation for this type of analysis, and where they point the user towards additional literature methods. By putting this section at the end, the user is not bogged down with this information in the middle of the tutorial. For example, Yu Xia recommends mentioning the ConSurf server. This appears a reasonable request. I think the authors should also consider mentioning here other related methods, as this will be useful information to the user. For example, the works by Rodrigue *et al.*¹⁻² and Tamuri *et al.*³ could be mentioned in reference to population genetic ways to measure site-specific rates and dN/dS. It may also be desirable to cite some of the more modern theoretical works on this, for example Halpern and Bruno⁴, Jones *et al.*⁵, etc. These are just suggestions. Similarly, Ugo Bastolla suggests adding some mention of what the results would like under other software, for example, in comparison with Rate4site. Although an extensive analysis of this topic may seen out of the scope of the tutorial, I think a set of references from the literature would be in order.
- Finally, I think the tutorial would benefit with the inclusion of a few additional figures. For example, the authors may want to consider adding a histogram of dN/dS vs. protein site, and perhaps also add screenshots of key program outputs. A well-illustrated tutorial is more appealing to students. Screenshots of program output are essential as it allows the user to check whether they are indeed obtaining the correct results. Perhaps also include tables summarising result values for key parameters.

References

1. Rodrigue N, Philippe H, Lartillot N: Mutation-selection models of coding sequence evolution with site-heterogeneous amino acid fitness profiles. *Proc Natl Acad Sci U S A*. 2010; **107** (10): 4629-34 [PubMed Abstract](#) | [Publisher Full Text](#)
2. Rodrigue N, Lartillot N: Site-heterogeneous mutation-selection models within the PhyloBayes-MPI package. *Bioinformatics*. 2014; **30** (7): 1020-1 [PubMed Abstract](#) | [Publisher Full Text](#)
3. Tamuri AU, dos Reis M, Goldstein RA: Estimating the distribution of selection coefficients from phylogenetic data using sitewise mutation-selection models. *Genetics*. 2012; **190** (3): 1101-15 [PubMed Abstract](#) | [Publisher Full Text](#)
4. Halpern A, Bruno W: Evolutionary distances for protein-coding sequences: modeling site- specific residue frequencies. *Molecular Biology and Evolution*. 1998; **15** (7): 910-917 [Publisher Full Text](#)
5. Jones CT, Youssef N, Susko E, Bielawski JP: Shifting Balance on a Static Mutation-Selection Landscape: A Novel Scenario of Positive Selection. *Mol Biol Evol*. 2017; **34** (2): 391-407 [PubMed Abstract](#) | [Publisher Full Text](#)

Is the rationale for developing the new method (or application) clearly explained?

Partly

Is the description of the method technically sound?

Yes

Are sufficient details provided to allow replication of the method development and its use by others?

Yes

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?

Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 26 Jan 2018

Claus Wilke, The University of Texas at Austin, USA

Dear Dr. Mario dos Reis,

We thank you for your helpful comments and suggestions. We have added additional notes, where appropriate, to indicate other theory papers, methods, and simulations a user might find useful. In the note about Rate4Site, we have added a brief paragraph that discusses the differences between Rate4Site and HyPhy's LEISR method. We have also added a paragraph in the Conclusions describing other relevant works. In that paragraph, we have cited the papers you suggested.

In the protocols section, we have added some information about the level of skill needed to run our analysis. We have replaced "Throughout, we assume that we are working on a UNIX-like command line interface" with the following: "Throughout, we assume that we are working on UNIX-like command line interface. We recommend that a user is comfortable with command execution and syntax, which includes flags, arguments, and directories. No prior knowledge of any of the listed software is essential. For python and R scripts, we provide detailed description for each script's function. As such, it is not strictly necessary that a user knows python or R to execute our pipeline. However, if more detailed understanding of the custom scripts is desired, a user should be familiar with python and R."

We also wanted to address your comment to include screenshots with program outputs. We have provided the git repository for this purpose. The git repository contains all input and output files the pipeline has generated. We feel that this is sufficient for the user to check their execution of our pipeline. A user can compare the output files we have provided to the output files he or she generated. We also have provided detailed descriptions of the output files for each step. We believe those should be enough to assess if the software has run and whether it has run correctly.

Competing Interests: No competing interests.

Referee Report 07 November 2017

doi:10.5256/f1000research.13954.r27041



David D Pollock

Department of Biochemistry & Molecular Genetics, University of Colorado School of Medicine, Aurora, Colorado, 80045, USA

The other two reviewers did a good job, such that I am going to be more philosophical about my review by focusing on two questions: Is this the best format for this kind of information, and am I (or the other two reviewers) the right person to review it? Other than a brief introduction and conclusion, the manuscript is mostly a detailed recipe of a specific pathway to assess correlations between evolutionary rates and some crystal structural features (when such information is available). This might be useful to a range of users such as undergraduates and beginning graduate students for whom it would serve as a gateway to thinking about these kinds of questions. The described pipeline is not a substantive modification to existing methods or an innovative application to new scientific questions, but rather is a tool designed to facilitate performance of experiments in a particular way. I would hope that more advanced students would quickly want to modify the experiments and tools that make up this pipeline, asking questions about the assumptions that the included tools make, and asking questions about alternative methods of analysis. Should one use only a maximum likelihood “best tree”, or should one consider a bootstrap or Bayesian posterior distribution? How does model choice interact with inference of site-specific rates? What assumptions were made in the alignment, and do those assumptions and errors in the alignment interact with the phylogenetic and rates inferences? In many ways, rather than a fixed and published document, the author’s pipeline might be better implemented in the context of an interactive and perhaps continuously modified lesson plan to teach students about the underlying issues, with more direction and encouragement to substitute in different methods and models. This leads to the question of whether I, or other professors, are the right people to review this. I trust Wilke and his laboratory members to have tested and made this pipeline function, and I don’t have time in the context of reviewing a paper to install and implement and test their scripts. But what they and I and other professors or advanced students are going to be bad at anyway is figuring out how hard it is and how it can go wrong in the hands of beginning students. So my suggestion is that it is beginning students who should be reviewing this pipeline, trying to implement it and trying to break it (or breaking it without trying). That would seem to me to be a more meaningful review.

As an aside, publishing answers to simplistic bubble questions seems inane to me, especially without the choice to opt out, and that is the context in which they were filled.

Is the rationale for developing the new method (or application) clearly explained?

Partly

Is the description of the method technically sound?

Partly

Are sufficient details provided to allow replication of the method development and its use by others?

Yes

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?

Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?

Partly

Competing Interests: No competing interests were disclosed.

Referee Expertise: Molecular evolution, evolutionary genomics, protein structure function and evolution, mathematical and computational biology

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 26 Jan 2018

Claus Wilke, The University of Texas at Austin, USA

Dear Dr. David Pollock,

We thank you for your comments and ideas. For specific feedback, you referred us to the other reviewers, and we have addressed their comments and modified our paper where appropriate.

Competing Interests: No competing interests.

Referee Report 06 November 2017

doi:[10.5256/f1000research.13954.r27042](https://doi.org/10.5256/f1000research.13954.r27042)



Yu Xia , Avital Sharir-Ivry

Department of Bioengineering, Faculty of Engineering, McGill University, Montreal, QC, Canada

This paper provides a complete and detailed recipe for calculating evolutionary rates and structural features of residues in proteins, and for correlating site-specific evolutionary rates with structural features. Such analysis is important for better understanding of protein structure-evolution relationships at the residue level. This paper describes all the computational steps in detail including the required computer programs and their installation, provides scripts to run the different programs and to process the results, and introduces a repository with all the scripts. This paper very clearly explains the steps needed to fully perform such analyses. The computational pipeline provided is expected to become a useful resource to the scientific community. This paper could potentially benefit from addressing a few minor issues described below.

1. While the paper aims to describe a complete analysis, the initial step of how to properly collect protein sequences for subsequent evolutionary analyses is not described. For the sake of completeness and to help interested readers to perform such analyses from scratch, it will be beneficial to at least mention this step (and preferably provide a more detailed description).
2. A full and automated procedure for analyzing relative amino acid rates already exists with the ConSurf server. At the same time, the second protocol described in this paper can be especially useful for site-specific codon rate calculations, which is not included in ConSurf. It will be beneficial to provide more

context regarding other available procedures and how the procedure described in this paper complements these other available procedures.

3. It will be beneficial to include the scripts for normalizing the inferred relative rates as well as for finding the sequence in the MSA with smallest number of gaps. These steps are described in the paper but it appears that the corresponding scripts are not included.

4. The file name of the output for Step 1 of Protocol 4 (pdb_id.map.csv) is not described in the text.

5. It is unclear how the order of the lines in the output pdb_id.rates_features.csv is decided. Because this is the final output of the entire analysis, it will be useful to organize this output as clearly as possible.

Is the rationale for developing the new method (or application) clearly explained?

Yes

Is the description of the method technically sound?

Yes

Are sufficient details provided to allow replication of the method development and its use by others?

Partly

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?

Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

We have read this submission. We believe that we have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 26 Jan 2018

Claus Wilke, The University of Texas at Austin, USA

Dear Drs. Yu Xia and Avital Sharir-Ivry,

We thank you for your helpful comments and suggestions. We have added an additional step about collecting homologous sequences to both protocols 1 and 2. We also added a paragraph in the Conclusions about other methods to infer site-wise evolutionary rates. We also now discuss the differences between the different methods. See also our response to Dr. Mario dos Reis's comments.

Further, we now provide the name of the output file for step 1 of protocol 4, and we have added a note to step 6 of protocol 1 to state that our script to plot the figure contains code to normalize

inferred rates. When we worked on version 1 of our publication, HyPhy's LEISR method was not yet published. Both Rate4Site and LEISR infer similar rates. We encourage users to use LEISR over Rate4Site to circumvent little issues Rate4Site has, such as not outputting site-wise rates for all site in the alignment.

We have refrained from organizing the order of rows in the output file `pdb_id.rates_features.csv`. There are two columns by which we could order: the site position of the protein structure or the site position of the alignment. Because the sequence of the PDB structure is typically different from the ones in the alignment, the two will not align perfectly to each other. This introduces gaps in either of the two or both sequences. It is, therefore, unclear what the order of lines with gaps should be.

Competing Interests: No competing interests.

Referee Report 02 November 2017

doi:10.5256/f1000research.13954.r27048



Ugo Bastolla 

Severo Ochoa Molecular Biology Centre (CSIC-UAM), Autonomous University of Madrid, Madrid, Spain

The structural properties of a protein site in the native state of the protein are known to strongly influence its evolutionary rate, according to research of which the corresponding author has been a prominent exponent. This finding gives useful insights on how natural selection targets the structural properties of proteins.

This paper presents computational protocols for measuring site-specific substitution rates in protein families, either at the amino-acid or at the codon level, and mapping them to the site-specific structural descriptors that have been found to influence the rates most strongly, such as relative solvent accessibility and weighted contact number. All the necessary software is clearly illustrated and all the relevant input and output files are made available to researchers willing to repeat the exercise, to whom this report will be for sure useful.

However, I think that the paper falls a bit short in motivating why it is interesting to perform this computation for a protein family of interest, besides recovering the known correlations between evolutionary rates and structural descriptors. Secondly, different methods and software exist to compute substitution rates. The authors show that, for the protein family that they choose, rates at the amino-acid and at the nucleotide level obtained with the same software yield almost identical results. How do results compare using different software (Rate4site versus HyPhy)? How do the authors interpret the sites with estimated $dN/dS > 1$, which is considered an indication of positive selection, in the context of the protein structure?

Is the rationale for developing the new method (or application) clearly explained?

Partly

Is the description of the method technically sound?

Yes

Are sufficient details provided to allow replication of the method development and its use by others?

Yes

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?

Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Referee Expertise: Computational biology of proteins: Folding stability, evolution, protein dynamics with elastic network models. Theoretical ecology.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 26 Jan 2018

Claus Wilke, The University of Texas at Austin, USA

Dear Dr. Ugo Bastolla,

We thank you for your helpful comments and suggestions. We now cite the new LEISR paper that demonstrates that Rate4Site and HyPhy's method LEISR produce similar rates. We reference this paper in the note about Rate4Site.

Competing Interests: No competing interests.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research

