The Site/Group Extended Data format and tools

Julien Y. Dutheil[1,*], Diyar Hamidi[1], and Basile Pajot[1]

[1]Research Group "Molecular Systems Evolution", Department of Theoretical Biology, Max Planck Institute for Evolutionary Biology, August-Thienemann-Str. 2, 24306 Plön, Germany

[*]Corresponding author: August-Thienemann-Str. 2, 24306 Plön.
dutheil@evolbio.mpg.de

November 14, 2023

Abstract

Comparative sequence analysis permits unravelling the molecular processes underlying gene evolution. Many statistical methods generate candidate positions within genes, such as fast or slowly-evolving sites, coevolving groups or residues, sites undergoing positive selection or changes in evolutionary rates. Understanding the functional causes of these evolutionary patterns requires combining the results of these analyses and mapping them onto molecular structures, a complex task involving distinct coordinate referential systems. To ease this task, we introduce the site/group extended data (SGED) format, a simple text format to store (groups of) site annotations. We developed a toolset, the SgedTools, which permits SGED files manipulation, creating them from various software outputs and translating coordinates between individual sequences, alignments, and three-dimensional structures. The package also includes a Monte-Carlo procedure to generate random site samples, possibly conditioning on site-specific features. This eases the statistical testing of evolutionary hypotheses, accounting for the structural properties of the encoded molecules.

1 Introduction

Evolutionary comparative sequence analysis can unravel information about the evolutionary processes that shape the observed genetic diversity. When applied to gene sequence alignments,

dedicated statistical methods detect positions that evolved under a particular evolutionary scenario, such as negative/positive selection or coevolution [Yang, 2006, Pollock et al., 1999]. Further insights into the functional role of these positions in the molecule and organism can be obtained by mapping them onto the three-dimensional structure of the encoded molecule and assessing their structural properties.

Mapping evolutionary predictions onto three-dimensional structures requires translating positions between three distinct reference systems: alignment positions, individual sequences, and three-dimensional structures. While software that allows the joint visualisation of sequence alignments, phylogenies and protein structures is available [Meng et al., 2006, Waterhouse et al., 2009], it requires manual interaction to visualize the results of evolutionary analyses, restricting their usage to case studies and preventing their use in genomic pipelines. Furthermore, each analysis software outputs results in its distinct format, complicating the development of generic analysis tools.

We designed the Site/Group Extended Data (SGED) format to facilitate the cross-analysis of sequence sites and their annotations. We introduce the SgedTools package, which contains utilities to manipulate and analyse SGED files. Lastly, we demonstrate their application on a classic example of positively selected sites in Primates lysozyme sequences.

2 The Site/Group Extended Data (SGED) format

We propose generalising the text tabular format to account for site coordinates. The Site/Group Extended Data (SGED) format is based on the widely used comma-separated values (CSV) and tab-separated values (TSV) formats, where columns represent variables and rows, data points – here in the form of (groups of) sites in a sequence or alignment. The SGED file contains one or several columns to store coordinates (*e.g.* a site's position in the alignment), with a dedicated syntax: the coordinates are specified within square brackets, and coordinates are separated by semi-columns (see Table 1). Other columns represent any measure or statistic for the corresponding groups. The SgedTools offers a collection of programs that specifically deal with the coordinates of the groups. They also compute statistics that will be added as columns in the SGED files.

3 Generating and manipulating SGED files

As SGED files are CSV/TSV files, they can be easily generated and edited, either manually or with dedicated software, such as spreadsheets, R, or the Python package `pandas`. The format is supported natively by programs using the Bio++ libraries, outputting various alignment statistics [Guéguen et al., 2013]. The SgedTools contains several conversion utilities that generate SGED files from the output of programs for sequence and structure analysis (Supplementary Table 1). SGED files can be further manipulated by dissociating sites within groups or combining sites into groups according to the content of a column. Finally, the columns of two SGED files can be merged based on the groups coordinates.

4 Indexing and coordinate translation

A prerequisite for analysing candidate positions in a sequence or sequence alignment is the conversion of coordinates to a common reference (Supplementary Table 1). The most basic conversion is between sequences within an alignment and is easily achieved by indexing each sequence position according to their alignment column (Figure 1A). Another sequence-only conversion task is when sequences or alignments are concatenated, for instance, to reconstruct a joint phylogeny, jointly estimate model parameters on multiple genes, or perform an inter-gene coevolution analysis. Positions in the super-alignment subsequently need to be converted back to the original sequence coordinates for further analysis (Figure 1B).

Coordinates are required to cross results between different analyses, particularly evolutionary analyses (alignment-based) and functional analyses (single-sequence-based). A class of widely used functional analyses involve the three-dimensional structure of the encoded molecule, RNA

Table 1: Example of SGED file showing group statistics and their associated P values. The group coordinates are specified in the 'Group' column.

Group	Statistic	P value
[147; 157]	0.816295	0.04376
[334; 363]	0.533308	0.05941
[178; 316]	0.289917	0.99998
[167; 170; 186]	0.581136	0.04328
[154; 172; 162]	0.534361	0.27306
[142; 144; 158; 335]	0.648215	0.09130
[145; 347; 200; 242]	0.610141	0.29092
[198; 248; 329; 217; 312]	0.563759	1
[139; 232; 236; 150; 202; 205]	0.733876	0.00215

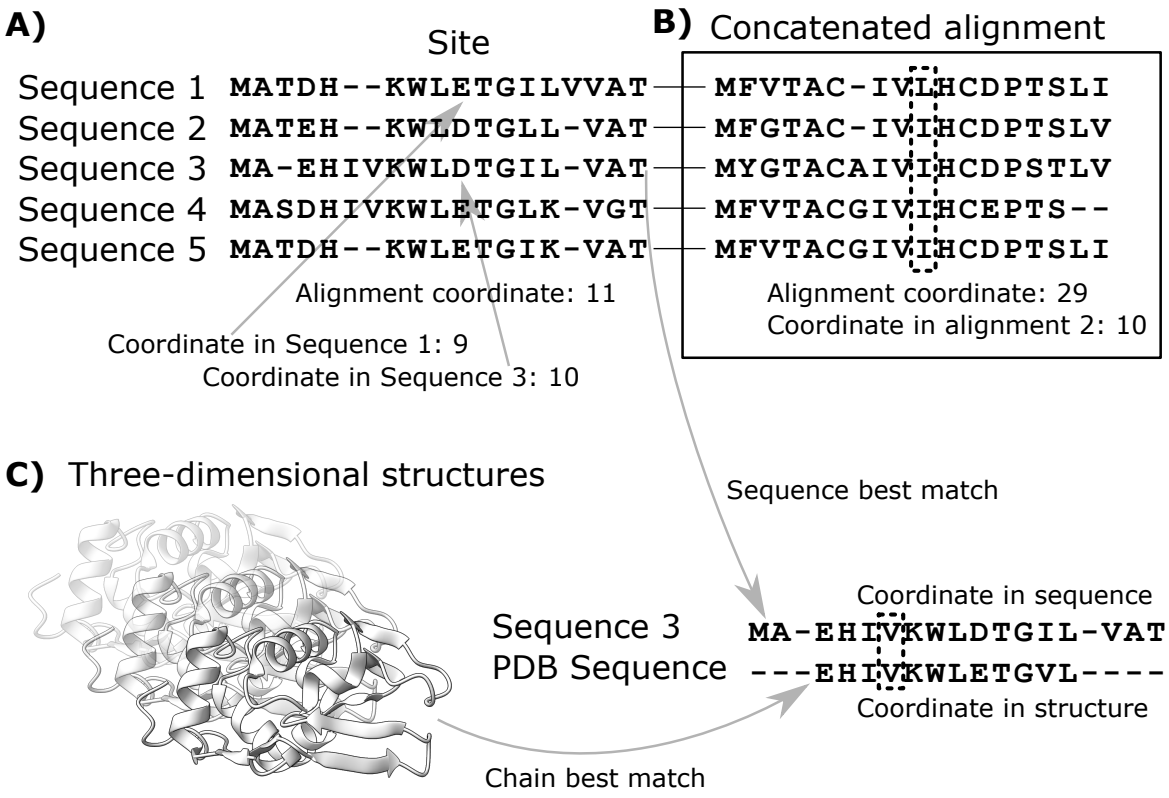

Figure 1: **Distinct coordinate systems.** A) Sites (= alignment columns) correspond to distinct positions within each aligned sequence. B) When alignments are concatenated, one needs to keep track of the original alignment coordinates in the concatenated alignment. C) To map alignment positions onto a three-dimensional structure, the sequence of each chain must be aligned with each sequence of the alignment to find the best match.

or protein. Three-dimensional structures can be obtained experimentally or predicted computationally. In both cases, some data may be missing so that the structure of some part of the sequence could not be obtained. Furthermore, the individual sequence used to predict the structure is rarely the same as the one used for the evolutionary analysis, possibly from a different species. Mapping candidate sites from evolutionary analyses onto a protein or RNA structure is a challenging task that requires sequence alignment between reference sequences (Figure 1C).

The `create-structure-index` program from the SgedTools permits the automation of such a task. Using a set of PDB structures, it aligns each sequence in a sequence alignment with the sequence of every chain in every PDB entry provided as input. Using the best matching pair of sequences, it then creates an alignment-structure index that maps all alignment positions onto the selected three-dimensional structure with minimal data loss. The sequence alignment is done using methods from the BioPython package [Cock et al., 2009]. Structure-mapped positions can then be used to extract structural properties.

5 Adding structural properties

Information about the functional relevance of predicted sites can be obtained by knowledge of their three-dimensional position. Relevant structural characteristics include location in secondary structure motifs, solvent exposure, number of residue contacts and inter-residue distances. Some information is directly accessible from the three-dimensional structure file; others can be predicted with dedicated software. The `structure-infos` program uses the `BioPython.PDB` package [Hamelryck and Manderick, 2003] to automatically retrieve structural properties from PDB and mmCIF files, such as secondary structure motives (Supplementary Table 1). It can also compute three-dimensional distances between sets of residues. `structure-infos`, and can further retrieve information about residues's RSA and depth using the `BioPython.PDB` parsers for the DSSP [Kabsch and Sander, 1983] and MSMS [Sanner et al., 1996] programs.

`structure-infos` further includes an algorithm computing the number of residue clusters in a group of sites. It first generates the matrix of pairwise distances between all pairs of residues in a group. A hierarchical clustering tree is then computed from the distance matrix, using the nearest linkage algorithm, as implemented in the `cluster.hierarchy.single` function in the SciPy package [Virtanen et al., 2020]. A distance threshold is then used to obtain clusters of residues. To assess the significance of structural statistics, we need to compare their observed values to their expectation under a null model. Such expectations can be derived using randomization procedures.

6 Advanced hypothesis testing using randomization

The `randomize-groups` program (Supplementary Table 1) generates random groups from two input SGED files: a first file with test groups, whose characteristics will be reproduced in the randomly generated groups and a second file providing the list of sites to sample from, with their properties. Each site can only be sampled once in each test group, but a site can be sampled multiple times between test groups if several are provided.

`randomize-groups` can perform a conditional sampling by selecting sites with similar properties to those in the tested group. This is achieved by specifying a conditional variable, provided as a dedicated column in the list of sites to sample. Continuous variables are discretized, and a bias correction for skewed distributions is implemented, as described in Chaurasia and Dutheil [2022]. In section 8, we demonstrate how the SgedTools can be used to statistically analyse

the structural properties of sites detected to evolve under positive selection, using conditional sampling to disentangle the effect of RSA and residue dispersal.

7 Program installation and usage

The SgedTools package is a collection of independent scripts written in Python (version 3.1 minimum). It makes use of several Python packages:

`pandas` for CSV/TSV file reading, manipulating and writing [The pandas development team, 2020],

`numpy` **and** `scipy` for numerical calculations and statistics [Harris et al., 2020, Virtanen et al., 2020],

`biopython` for sequence and three-dimensional structures manipulation [Cock et al., 2009].

Once the packages are available in the Python environment, each script can be copied and run 'as is' without any further installation needed. The programs are run from the command line, using options which are specified using standard short (e.g. `-a`) or long arguments (e.g. `--alignment`). The SgedTools package is distributed with detailed example analyses that can serve as templates for developing dedicated pipelines.

8 Application example: structural analysis of positively selected sites.

To illustrate the use of the SgedTools, we evaluate the results of the positive selection analysis of Yang and Nielsen [2002]. This data set serves as an example for the widely used package PAML [Yang, 2007]. The PAML output file can be converted to the SGED format using the `paml2sged` program, keeping only the seven sites with a posterior probability calculated by the empirical Bayesian method and at least equal to 0.7:

```
python3 sged-paml2sged.py \
    --paml mlc \
    --output lysozymeLarge-possel.sged \
    --method bayesian \
    --threshold 0.7
```

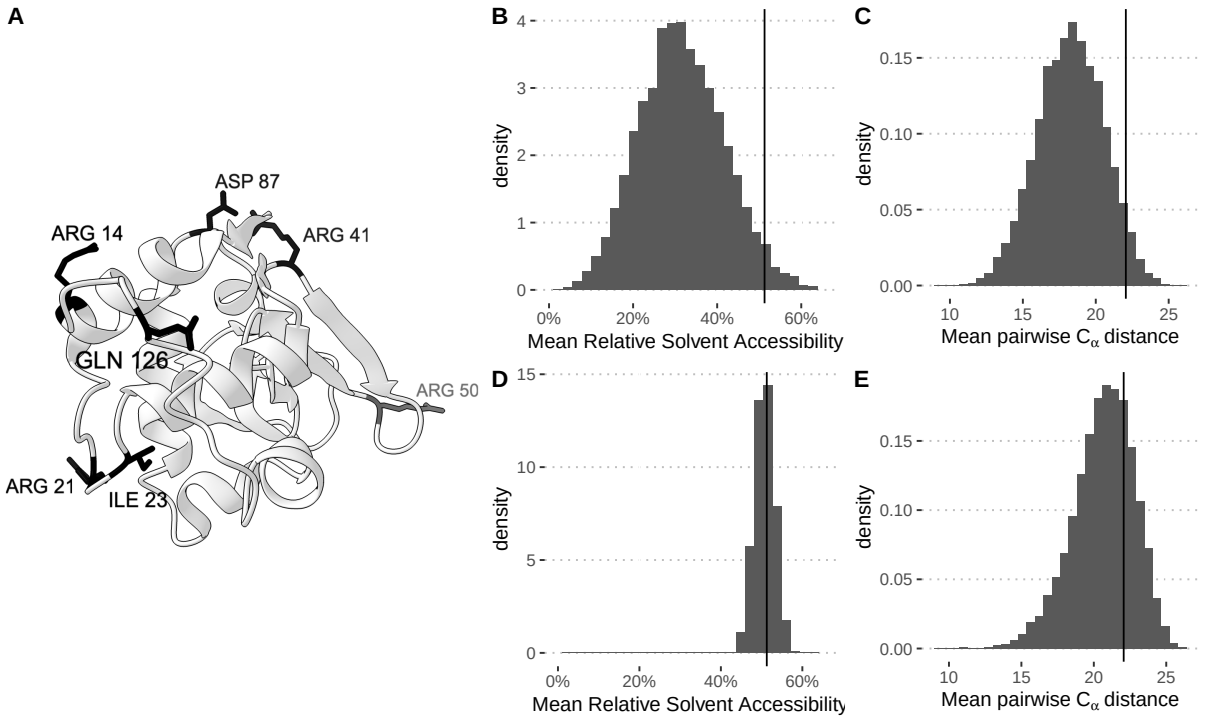

Figure 2: **Analysis of positively selected sites in the lysozyme.** A) Three-dimensional structure of the human lysozyme (PDB structure 134l). Residues corresponding to sites evolving under a positive selection scenario with a posterior probability higher or equal to 70% are shown in full (labelled residues). B-E: Histograms of distributions over 10,000 random groups. Vertical lines show the corresponding observed values. B, D: average relative solvent accessibility (RSA). C, E: average pairwise C_α distance. B, C: sampling over all residues in the structure. D, E: sampling conditioned on the RSA value of each residue.

The resulting file `lysozymeLarge-possel.sged` has the following content:

```
Group   amino_acid      probability
[14]    R       0.859
[21]    R       0.858
[23]    I       0.853
[41]    R       0.71
[50]    R       0.704
[87]    D       0.869
[126]   Q       0.71
```

Using the Colobus sequence as a reference, we search the protein data bank (PDB) [Berman et al., 2000] for three-dimensional structures of lysozymes. After downloading the ten best matching PDB files, we use the `create-structure-index` program to align all chains from all structures and find the best alignment, which is used to create a *structure index*:

```
python3 sged-create-structure-index.py \
          --pdb "*.pdb" \
          --pdb-format PDB \
          --alignment colobus_aa.fas \
          --alignment-format fasta \
          --gap-open -2 \
          --output lysozymeLarge_PdbIndex.txt \
          --exclude-incomplete
```

We use a gap-opening penalty of -2 to maximize the overlap of the structure with the selected sequence, as they are not from the same species. Incomplete structures are excluded from the comparison. Chain A from the `134L` PDB entry was selected as the closest match. We then use the generated index to obtain the coordinates of the positively selected sites in the protein structure:

```
python3 sged-translate-coords.py \
          --sged lysozymeLarge-possel.sged \
          --output lysozymeLarge-possel_PDB.sged \
          --index lysozymeLarge_PdbIndex.txt \
          --name PDB
```

resulting in the SGED file:

```
Group   PDB     amino_acid      probability
[14]    [A:ARG14]       R       0.859
[21]    [A:ARG21]       R       0.858
[23]    [A:ILE23]       I       0.853
[41]    [A:ARG41]       R       0.71
[50]    [A:ARG50]       R       0.704
[87]    [A:ASP87]       D       0.869
[126]   [A:GLN126]      Q       0.71
```

The translated coordinates can be used to visualize the candidate sites with software like PyMol [Schrödinger, LLC, 2015] or ChimeraX [Meng et al., 2023] (Figure 2A). The positively selected sites are located at the protein's surface and seem to be spread. We can statistically assess this by measuring the mean pairwise distance between the α carbons (C_α) of the residues and their

mean relative solvent accessibility (RSA). We first need to create an SGED file where all sites are listed as a single group:

```
python3 sged-group.py \
         --sged lysozymeLarge-possel_PDB.sged \
         --group PDB \
         --output lysozymeLarge-possel-group.sged
```

resulting in the following SGED file:

```
Group
[A:ARG14;A:ARG21;A:ILE23;A:ARG41;A:ARG50;A:ASP87;A:GLN126]
```

We then compute the structural properties of this group using the `structure-info` program, using the best-matching PDB entry:

```
python3 sged-structure-infos.py \
         --sged lysozymeLarge-possel-group.sged \
         --pdb 134l.pdb \
         --pdb-format PDB \
         --measure AlphaDist \
         --measure DSSPsum \
         --output lysozymeLarge-possel-group_PDB_infos.sged
```

The program computes two statistics for the group, the C_α distance (argument `--measure AlphaDist`) and several summary statistics generated by the DSSP program (argument `--measure DSSPsum`). The resulting mean C_α distance is 22.06 Å, and the mean relative solvent accessibility is 51.27%. We then compare these statistics to their null expectation, obtained by sampling groups of sites in the protein structure using the program `randomize-groups`. We need to provide the list of sites to sample from using the `structure-list` program:

```
python3 sged-structure-list.py \
         --pdb 134l.pdb \
         --pdb-format PDB \
         --output 134l_residues.sged
```

We generate 10,000 random groups:

```
python3 sged-randomize-groups.py \
```

```
        --sged-groups lysozymeLarge-possel-group.sged \
        --sged-sites 134l_residues.sged \
        --number-replicates 10000 \
        --output lysozymeLarge-possel-group_random.sged
```

Finally, we compute the structural properties of the random groups, as it was done for the group of positively selected sites:

```
python3 ../../src/sged-structure-infos.py \
        --sged lysozymeLarge-possel-group_random.sged \
        --pdb 134l.pdb \
        --pdb-format PDB \
        --measure AlphaDist \
        --measure DSSPsum \
        --output lysozymeLarge-possel-group_random_PDB_infos.sged
```

The two observed statistics appear larger than their random expectation (Figure 2 B and C). We compute an upper bound for the P value as

$$P\ value = \frac{|S_{sim} \geq S_{obs}| + 1}{10,000 + 1,} \tag{1}$$

where $|S_{sim} \geq S_{obs}|$ represents the number of simulated groups with a statistic at least equal to the observed value (one-tail test). This gives 0.0304 for the solvent exposure and 0.0444 for the C_α distance, both significant at the 5% level.

These results indicate that the surface exposure of the candidate sites is likely linked to their function. We further ask whether their dispersal is also possibly a signature of their function or whether it is a by-product of their location at the surface of the protein. We perform a *conditional sampling* by sampling exclusively sites with a solvent exposure similar to the candidate sites. For this, we first compute the exposure of every residue of the structure:

```
python3 sged-structure-infos.py \
        --sged 134l_residues.sged \
        --pdb 134l.pdb \
        --pdb-format PDB \
        --measure DSSP \
```

```
        --output 134l_residues_infos.sged
```

and then condition on the RSA of each site, which is stored in the "Rsa" column of the `134l_residues_infos.sged` file:

```
python3 sged-randomize-groups.py \
        --sged-groups lysozymeLarge-possel-group.sged \
        --sged-sites 134l_residues_infos.sged \
        --measure Rsa \
        --similarity-threshold 0.2 \
        --number-replicates 10000 \
        --output lysozymeLarge-possel-group_random-rsa.sged
```

We finally compute the structural characteristics of the random groups:

```
python3 sged-structure-infos.py \
        --sged lysozymeLarge-possel-group_random-rsa.sged \
        --pdb 134l.pdb \
        --pdb-format PDB \
        --measure AlphaDist \
        --measure DSSPsum \
        --output lysozymeLarge-possel-group_random-rsa_PDB.sged
```

The distribution to the average RSA is now centered on the observed value, showing that the exposure effect is accounted for (Figure 2D). However, the C_α distance is no longer significant (Figure 2E), P value = 0.2674, indicating that the apparent residues' dispersal results from a spurious correlation with the RSA.

9 Availability

The SgedTools are distributed under the GNU General Public Licence version 3 (GPL3) and can be downloaded from GitHub.com at `https://jydu.github.io/sgedtools/`.

10 Conclusion

We introduced a set of generic tools that permit integrating results from various evolutionary analyses with functional annotations, including three-dimensional structures. This interoperabil-

ity is made possible by a generic file format for storing position-specific sequence annotations. The format supports annotations for single sites and groups of sites while being simple and flexible. Besides basic data manipulation, the SgedTools implement more complex algorithms, for mapping three-dimensional structures to sequence alignments and a conditional sampling of sites for the statistical testing of hypotheses. The tools can be combined as modules to create pipelines for testing functional and structural hypotheses from evolutionary predictions.

11 Acknowledgments

JYD acknowledges funding from the Max Planck Society.

12 Author's contributions

JYD: Conceptualization, Methodology, Software, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Supervision

DH: Software, Data Curation, Writing - Review & Editing, Visualization

BP: Software, Data Curation, Writing - Review & Editing

References

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, Jan. 2000. ISSN 0305-1048. doi: 10.1093/nar/28.1.235.

S. Chaurasia and J. Y. Dutheil. The Structural Determinants of Intra-Protein Compensatory Substitutions. *Mol Biol Evol*, 39(4):msac063, Apr. 2022. ISSN 1537-1719. doi: 10.1093/molbev/msac063.

P. J. A. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. L. de Hoon. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11): 1422–1423, June 2009. ISSN 1367-4811. doi: 10.1093/bioinformatics/btp163.

L. Guéguen, S. Gaillard, B. Boussau, M. Gouy, M. Groussin, N. C. Rochette, T. Bigot, D. Fournier, F. Pouyet, V. Cahais, A. Bernard, C. Scornavacca, B. Nabholz, A. Haudry,

L. Dachary, N. Galtier, K. Belkhir, and J. Y. Dutheil. Bio++: Efficient Extensible Libraries and Tools for Computational Molecular Evolution. *Mol. Biol. Evol.*, June 2013. ISSN 1537-1719. doi: 10.1093/molbev/mst097.

T. Hamelryck and B. Manderick. PDB file parser and structure class implemented in Python. *Bioinformatics*, 19(17):2308–2310, Nov. 2003. ISSN 1367-4803. doi: 10.1093/bioinformatics/btg299.

C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020. doi: 10.1038/s41586-020-2649-2.

W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, Dec. 1983. ISSN 0006-3525. doi: 10.1002/bip.360221211.

E. C. Meng, E. F. Pettersen, G. S. Couch, C. C. Huang, and T. E. Ferrin. Tools for integrated sequence-structure analysis with UCSF Chimera. *BMC Bioinformatics*, 7(1):339, July 2006. ISSN 1471-2105. doi: 10.1186/1471-2105-7-339.

E. C. Meng, T. D. Goddard, E. F. Pettersen, G. S. Couch, Z. J. Pearson, J. H. Morris, and T. E. Ferrin. UCSF ChimeraX: Tools for Structure Building and Analysis. *Protein Sci*, page e4792, Sept. 2023. ISSN 1469-896X. doi: 10.1002/pro.4792.

D. D. Pollock, W. R. Taylor, and N. Goldman. Coevolving protein residues: maximum likelihood identification and relationship to structure. *J. Mol. Biol.*, 287(1):187–198, Mar. 1999. ISSN 0022-2836. doi: 10.1006/jmbi.1998.2601.

M. F. Sanner, A. J. Olson, and J. C. Spehner. Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers*, 38(3):305–320, Mar. 1996. ISSN 0006-3525. doi: 10.1002/(SICI)1097-0282(199603)38:3%3C305::AID-BIP4%3E3.0.CO;2-Y.

Schrödinger, LLC. The PyMOL Molecular Graphics System, Version 1.8, Nov. 2015.

The pandas development team. pandas-dev/pandas: Pandas, Feb. 2020. URL `https://doi.org/10.5281/zenodo.3509134`.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

A. M. Waterhouse, J. B. Procter, D. M. A. Martin, M. Clamp, and G. J. Barton. Jalview Version 2—a multiple sequence alignment editor and analysis workbench. *Bioinformatics*, 25(9):1189–1191, May 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp033. URL `https://doi.org/10.1093/bioinformatics/btp033`.

Z. Yang. *Computational Molecular Evolution*. Oxford University Press, Oxford, Dec. 2006. ISBN 978-0-19-856702-8.

Z. Yang. PAML 4: phylogenetic analysis by maximum likelihood. *Mol. Biol. Evol.*, 24(8): 1586–1591, Aug. 2007. ISSN 0737-4038. doi: 10.1093/molbev/msm088.

Z. Yang and R. Nielsen. Codon-substitution models for detecting molecular adaptation at individual sites along specific lineages. *Mol. Biol. Evol.*, 19(6):908–917, June 2002. ISSN 0737-4038.