scoup: Simulate Codon Sequences with Darwinian Selection Incorporated as an Ornstein-Uhlenbeck Process

Hassan Sadiq[1,2,*], and Darren P. Martin[2]

[1] Department of Statistics and Actuarial Science, Stellenbosch University, South Africa
[2] Institute of Infectious Diseases and Molecular Medicine, Division of Computational Biology, Department of Integrative Biomedical Sciences, University of Cape Town, South Africa

[*] Corresponding author: hassan.t.sadiq@gmail.com
ORCID: HS, 0000-0003-0192-7134; DPM, 0000-0002-8785-0870

Abstract

Motivation:
Genetic analyses of natural selection within and between populations have increasingly developed along separate paths. The two important genres of evolutionary biology (i.e. phylogenetics and population genetics) borne from the split can only benefit from research that seeks to bridge the gap. Simulation algorithms that combine fundamental concepts from both genres are important to achieve such unifying objective.

Results:
We introduce `scoup`, a codon sequence simulator that is implemented in R and hosted on the Bioconductor platform. There is hardly any other simulator dedicated to genetic sequence generation for natural selection analyses on the platform. Concepts from the Halpern-Bruno mutation-selection model and the Ornstein-Uhlenbeck (OU) evolutionary algorithm were creatively fused such that the end-product is a novelty with respect to computational genetic simulation. Users are able to seamlessly adjust the model parameters to mimic complex evolutionary procedures that may have been otherwise infeasible. For example, it is possible to explicitly interrogate the concepts of static and changing fitness landscapes with regards to Darwinian natural selection in the context of codon sequences from multiple populations.

Availability:
`scoup` is published for free public use under the GPL-2 license. It is available for download from the Bioconductor platform (`doi.org/10.18129/B9.bioc.scoup`), along with detailed help and example files.

Keywords: Codons, Fitness Landscape, Molecular Evolution, Ornstein-Uhlembeck, Mutation, MutSel, Selection Coefficient, Simulate, vN/vS

1 Introduction

Statistical inference of the extent to which Darwinian natural selection has impacted genetic data commands a healthy portion of the phylogenetic literature [1]. Validation of these largely codon-based models relies heavily on simulated data. Given the ever increasing diversity of natural selection inference models that exist [2, 3], there is a need for more sophisticated simulators to match the ever expanding model complexities.

Bioconductor [4] is a leading platform where peer-reviewed bioinformatic software useful for biological data analyses are hosted. A search of the entries on the platform, in Version 3.19 on 29 October 2024, with keywords including, `codon`, `mutation`, `selection`, `simulate` and `simulation` returned a total of 72 unique packages out of the 2300 available. None of the retrieved entries was dedicated to codon data simulation for natural selection analyses. Thus, `scoup` is designed on the basis of the mutation-selection (MutSel) framework [5] as an overdue contribution to the void.

Software and/or packages for simulating genetic sequences are also rare in the scientific literature [6]. Existing simulators tend to be more suitable for quantitative character evolution or population genetic analyses. These include, `ape` [7], `ouch` [8], `geiger` [9], `SLiM` [10] and `SimBit` [11]. Other extensively used DNA sequence simulators including, `Seq-Gen` [12], `INDELible` [13], `PhyloSim` [14] and `phangorn` [15] are parameterised in accordance with likelihood-based (ω) models. More recent sequence simulators, such as, `phastSim` [16] and `AliSim-HPC` [17] prioritise output capacity. Only a few genetic simulators have been built upon the more elaborate MutSel evolutionary concept. These include, Pyvolve [18] and SGWE [19]. To the best of our knowledge, these existing MutSel friendly simulators are only able to generate data from static fitness landscapes. With our proposed simulator, it is possible to generate codon sequences from fitness landscapes that are either static or changing [also known as *seascapes*, 20].

2 Algorithm

`scoup` is further unique for at least three reasons. First, it incorporates Darwinian natural selection into the MutSel model in terms of variability of selection coefficients [21]. Second, it directly utilises the concept of fitness landscapes. Third, fitness landscape updates can be executed in either a deterministic or a stochastic format. The stochastic updates are implemented in terms of the more biologically amenable, Ornstein-Uhlenbeck (OU) process [22, 23]. An indirect benefit of the OU update is that it accommodates controlled correlation in evolutionary paths within a site. These three characteristics are described next.

2.1 Codon substitution model

Our choice of the MutSel model was because it permits straightforward incorporation of the fitness landscape concept in its original form, as discussed in the seminal work of Sewall Wright [24]. The instantaneous codon substitution matrix ($\mathbb{Q} = \{q_{ij}\}$) was defined as

$$q_{ij} = \begin{cases} k \times m_{ij} \times f_{ij} & \text{if } i \text{ \& } j \text{ differ by 1 nuc. pos.,} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

for $i \neq j$, $q_{ii} = -\sum_{i \neq j} q_{ij}$ and i, j are the sense codons. The scaling constant k is arbitrary. In inference models, k is often set such that $-\sum_i \pi_i q_{ii} = 1$, to ensure that the estimates of branch lengths reflect time of divergence from the most recent common ancestor. In `scoup`, branch length is accepted as an input. Therefore, a value $k = 1$ is used to ensure that the degree of the imposed selection effect remains undiminished. The codon mutation rate m_{ij} is equated to a nucleotide analogue m_{gh}, where nucleotide g and h respectively occupy the positions that differ between codons i and j. Nucleotide mutation is modelled following Hasegawa et al. [25]. The transition to transversion rate ratio and the average mutation rate parameters were embedded in the simulator as 4 and 0.25, respectively. Allowing the parameter pair of the mutation model to be user-specified was inconsequential, particularly given that the mutation rate can be influenced by the population size input [26]. The notation f_{ij} is the fixation probability and it is computed as follows.

$$f_{ij} = \frac{1 - e^{-2s_{ij}}}{1 - e^{-2N_{\mathrm{e}} s_{ij}}}, \tag{2}$$

where N_{e} is the effective population size and

$$s_{ij} = s_j - s_i, \tag{3}$$

where, s_i and s_j are the selection coefficients associated with the resident and the mutant sense codons respectively [27, 28]. These coefficients were also used as weights to generate residue frequencies with the following softmax identity.

$$\pi_i = \frac{e^{s_i - c}}{\sum_{j=1}^{61} e^{s_j - c}}, \tag{4}$$

where $c = \max_j s_j$. These frequencies serve as weights with which the root residue is randomly sampled at every site. The codon selection coefficients s_i and s_j are obtained as functions of the corresponding amino acid selection coefficients, $s_{\mathtt{aa}}$, where $\mathtt{aa} = 1, 2, \ldots, 20$ [21]. It is possible to generate $s_{\mathtt{aa}}$ in one of two ways. As a random sample from (a.) a Normal$(0, \sigma_{\mathtt{n}}^2)$ or (b.) a Gamma$(1,\ \sigma_{\mathtt{n}}^{-1})$ distribution, where $\sigma_{\mathtt{n}}^2$ is set by the user. Support for our choice of distributions is established in the literature [29–31]. The role of $s_{\mathtt{aa}}$ as a parameter for the generating function of the synonymous coefficients meant that negative values may cause computational overflows. As a result, the Normal distribution is truncated such that only non-negative values may be obtained. No such restriction is imposed on the synonymous coefficients. Given that s_{ij} is defined in terms of differences, observe that the restriction imposed on $s_{\mathtt{aa}}$ does not eliminate negative selection.

Synonymous selection [18] may be varied in `scoup`. The inputs of the simulator include, $\sigma_{\mathtt{n}}^2/\sigma_{\mathtt{s}}^2$ (interchangeably referred to as $\mathrm{vN/vS}$) and $\sigma_{\mathtt{n}}^2$. The variance of the synonymous codon coefficients ($\sigma_{\mathtt{s}}^2$) which corresponds to the synonymous selection metric is then obtained as a function of these two inputs. Given the selection coefficient sampled for amino acid aa ($s_{\mathtt{aa}}$), the coefficients for the associated synonymous codons are sampled from a Uniform distribution with mean $s_{\mathtt{aa}}$ and variance $\sigma_{\mathtt{s}}^2$. Variance among the amino acid coefficients ($\sigma_{\mathtt{n}}^2$) loosely represents the variance among the non-synonymous selection coefficients. That is, $\sigma_{\mathtt{n}}^2$ captures non-synonymous selection. Therefore, as an uncommon feature of `scoup`, the magnitude of Darwinian natural selection can be controlled with $\mathrm{vN/vS}$.

2.2 Fitness Landscape

Two fitness landscape types are accessible in `scoup`: the static landscapes and the seascapes. Static landscapes are time-independent and non-adaptive. They are such that positive selection occurs to compensate for perturbations in the system [20, 32]. In seascapes, positive selection is adaptive and may vary over time [20]. With respect to molecular evolution, fitness landscapes could be defined in terms of substitution rate matrices [33]. Therefore, we incorporate fitness landscape in terms of the parameters needed to sample the amino acid selection coefficients ($s_{\mathtt{aa}}$) that are necessary for the corresponding MutSel model. We allow for deterministic and stochastic landscape control.

A deterministic static fitness landscape may be executed in `scoup` with a single set of parameters ($\xi = \{\sigma_{\mathtt{n}}^2, \mathrm{vN/vS}\}$) needed to sample a 20-element numerical vector of amino acid selection coefficients. The coefficients are sampled once and separately for individual amino acids. The consequent codon substitution matrices are then fixed for a site. Perturbations are introduced to the system as frequency-dependent [34]. That is, the active substitution matrix is determined by the current wild-type residue. The process is repeated independently at each site. With careful parameter specifications, it is possible to simulate directional evolution using this framework.

A deterministic seascape may be executed as a function of multiple sets of parameters ($\xi_1, \xi_2, \ldots,\ \xi_k$) needed for the different vectors of selection coefficients that match the changing fitness landscapes which distinguish such a setting. For a phylogeny with 2^d extant taxa, parameter sets ($\xi_1, \xi_2, \ldots,\ \xi_d$) need to be inputted. A separate substitution matrix is then generated at each node of the phylogeny. The process restarts independently at each site. Such a seascape implementation results in an episodic evolutionary phenomenon [35]. We emphasise that the evolutionary procedure for these "deterministic" implementations remains a random process and that the contents of the substitution matrices are functions of random variables, as described previously. Also note that the evolutionary process does not completely restart at each node. The codon generated from the substitution events at a node is used to initiate the process at a subsequent node.

Non-deterministic executions of both the static and the changing landscapes are also possible with `scoup`. Stochastic landscape settings are incorporated as OU processes.

2.3 Ornstein-Uhlenbeck process

Concepts of Brownian motion are conventionally incorporated in modelling natural selection as an *Ornstein-Uhlenbeck* (OU) process [22, 23]. Traditional Brownian models allow random movements within a system,

while OU models include a reversion parameter (θ) that pulls the system towards an asymptotic mean (μ). In the context of molecular evolution, the random movements represent genetic drifts and the pull towards the mean could be thought of as the force of selection ensuring evolution toward beneficial traits [36].

Consider a codon site that evolves according to an OU process. The codon evolutionary model for the site may be defined as a function of a vector of 20 independently sampled amino acid selection coefficients ($s_{\mathtt{aa}}$). The values of the vector contents at time t_k (that is, $s_{\mathtt{aa}}^{(t_k)}$) can be expressed as observations of a Gaussian distribution as follows.

$$s_{\mathtt{aa}}^{(t_k)} \sim \mathsf{N}\left(\mu + \left(s_{\mathtt{aa}}^{(t_{k-1})} - \mu\right) e^{-\theta \Delta t};\ \frac{\sigma^2}{2\theta}\left(1 - e^{-2\theta\Delta t}\right)\right), \tag{5}$$

where, k is a positive integer. Observe that when time interval $\Delta t = t_k - t_{k-1}$ approaches infinity, the mean and the variance of the distribution tend toward μ and $\sigma^2/2\theta = \Sigma^2$ respectively. Further observe that setting $\Sigma^2 = 0$ eliminates intermittent variation of the OU process. Therefore, the stochastic landscape setting may be controlled in `scoup` by varying the value of the asymptotic variance parameter (Σ^2). A static stochastic landscape corresponds to the case of $\Sigma^2 = 0$. Perturbations are introduced into the system with the reversion parameter (θ) which causes the coefficients to be uniformly inflated thereby only influencing the synonymous effects. When $\Sigma^2 > 0$, a stochastic seascape is activated. Only a single set (ξ) of selection coefficient parameters is required to execute the stochastic framework, regardless of the landscape setting. The selection coefficients are updated during every substitution event. Multi-partitioning is not directly possible, but may be achieved with minimal difficulty using R's concatenation function.

Equation (5) can be stated simply as $s_{\mathtt{aa}}^{(t_k)} = f\left(s_{\mathtt{aa}}^{(t_{k-1})}\right)$. More significantly however, is the avenue created by this implementation of the OU process in a MutSel setting. It permits interrogation of the consequence of the interplay between genetic drift, effective population size and mutation on the inferred magnitude of the selection pressure acting on genetic sequences. Such interrogations are not new to population genomic studies [37]. They are however uncommon in the context of substitution events with respect to codon sequences. Therefore, `scoup` avails a novel avenue to thoroughly interrogate some of these facets of molecular evolution.

3 Implementation

Quality of the outputs from `scoup` were assessed. We simulated (see sample code in Figure 1) 20 independent sequence alignments made up of 1000 codon sites and 8 extant taxa for each of the parameter combinations presented in Figure 2. The phylogeny used was balanced and the length of its branches were 0.10 each. The stochastic OU framework was implemented and other function inputs were left at their default values (Figure 1). Estimates of $\mathrm{dN/dS}$ were obtained following Spielman and Wilke [21] and were averaged over all $s_{\mathtt{aa}}$ updates at each site and across the alignment. Inferences of ω were obtained with `CODEML` in `PAML` [2].

Estimates of ω and $\mathrm{dN/dS}$ summarised in Figure 2 strongly agree, except for the case of $(\sigma_{\mathtt{n}}^2, \sigma_{\mathtt{s}}^2) = (0.10, 0.02)$. The suppressed ω estimates, that is most pronounced for $\sigma_{\mathtt{n}}^2, \sigma_{\mathtt{s}}^2 > 0$, is likely a consequence of the well-documented conservative property of homogeneous ω inference techniques (see for example, Nielsen and Yang [38]). Regardless, a correlation coefficient of approximately 0.9971 was obtained when the ω and $\mathrm{dN/dS}$

```
adaptEnrty <- ouInput() # Line01
modelEntry <- hbInput() # Line02
sqEntry <- seqDetails() # Line03
seqData <- alignsim(adaptEnrty, sqEntry, modelEntry)
```

Figure 1: An example R code for simulating a codon sequence alignment with `scoup`. Default values were left unchanged. `Line01`: OU adaptation parameters where, $\mu = 0$, $\Sigma^2 = 0.01$ and $\theta = 0.01$. `Line02`: evolution model input where, $\boldsymbol{s} \sim \mathsf{Gamma}(\mathbf{1}, \boldsymbol{\sigma_n^{-1}})$, $\sigma_{\mathtt{n}}^2 = 10^{-5}$, $N_{\mathsf{e}} = 1000$ and $\mathrm{vN/vS} = 1$. `Line03`: sequence information where, site count is 250, extant taxa count is 64 and branch length is 0.1.

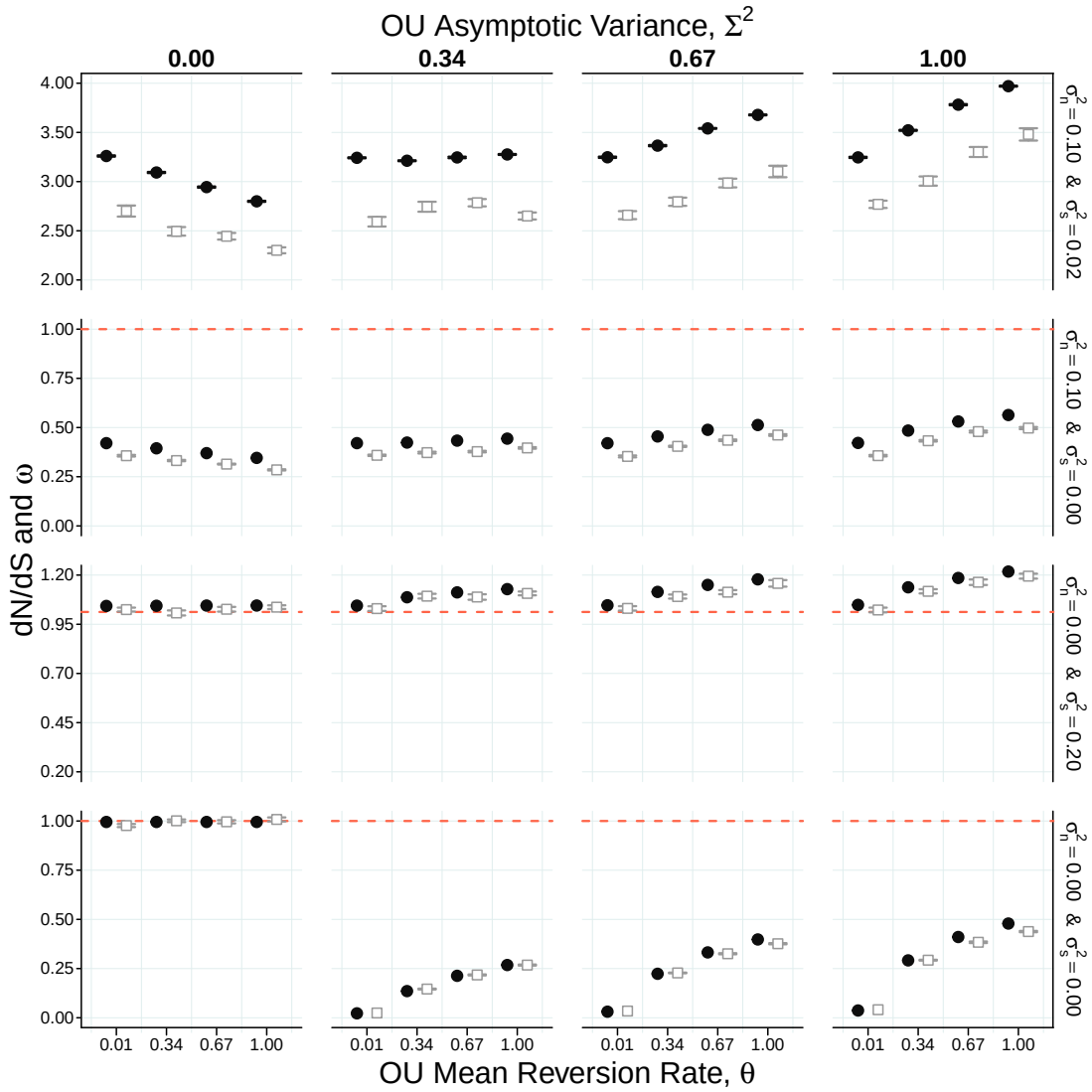

Figure 2: Demonstration of the accuracy of outputs from `scoup` in terms of the likelihood ω and the analytical $\mathrm{dN/dS}$ measures of natural selection. The variance of the amino acid selection coefficients is denoted σ^2_{n} and the variance of synonymous codon selection coefficients is denoted σ^2_{s}. The estimates of the selection measures were obtained homogeneously from each alignment generated for every combination of the stochastic landscape (Σ^2 and θ) and the Darwinian selection (σ^2_{n} and σ^2_{s}) parameters. The filled circles represent the average $\mathrm{dN/dS}$ estimates while the empty squares represent the average ω estimates, across 20 independent codon sequence alignments. The widths of the arrows correspond to twice the standard errors. The dashed lines highlight point of neutral selection effect.

averages were compared. The standard errors ranged between $[0.0000,\ 0.0077)$ and $(0.0004,\ 0.0627)$ for the $\mathrm{dN/dS}$ and ω estimates respectively. These measures confirm that the outputs from `scoup` are accurate.

The results in Figure 2 are supported in the literature. We begin with the static landscape ($\Sigma^2 = 0$). The inverse relationship between the magnitude of positive selection effect and the reversion rate (θ) is consistent with the knowledge that positive selection in static landscapes is compensatory [20, 32]. The presence of significant evidence of positive selection only when synonymous selection was present was also reported by Spielman and Wilke [21]. We add to their results by showing that the trend is similar even for adaptive seascapes.

More interesting features may be deduced from the seascape settings ($\Sigma^2 > 0$) in Figure 2. The direct relationship between mean reversion rate (θ) and the nominal magnitude of the selection effect may be interpreted in two ways. First, it fits the definition that positive selection is adaptive in changing landscapes [20]. In other words, rising θ tends to be associated with an increasing balancing selection [39] effect. The two-way effect of balancing selection is also evident from the summaries. That is, exaggerated diversifying selection effect (when $\sigma_{\mathrm{s}}^2 \neq 0$) and dampened purifying effect (when $\sigma_{\mathrm{s}}^2 = 0$). Second, the relationship reflects the movements in the evolutionary system caused by the interplay between landscape drift and mutational pull. Movements in the system increase as θ and Σ^2 increase [40]. These insights highlight some of the value and utility of `scoup`.

4 Conclusions

We present `scoup`, a R package that allows for simulation of codon sequences in a way that is capable of recapitulating the evolutionary processes of biological systems more realistically than most existing simulators. Our framework creatively incorporates the Ornstein-Uhlenbeck process into the mutation-selection evolutionary model. This attribute was demonstrated to be potentially important to unlock exciting research avenues that will improve existing knowledge about the complex interactions of different, potentially interacting, molecular evolutionary processes. In another unique contribution to the literature, the magnitude of the Darwinian selection affect on the simulated sequences was controlled with the ratio of the variances of selection coefficients ($\mathrm{vN/vS}$). Given the summaries in Figure 2, we state two hypotheses with respect to natural selection inference from multi-population genetic sequences. First, with ω, it is difficult to fully distinguish between compensatory and adaptive diversifying selection occurring on static and changing landscapes, respectively. Second, $\mathrm{dN/dS}$ and $\mathrm{vN/vS}$ are related such that the latter could complement ω as a selection pressure discriminant metric. To establish these hypotheses, at least numerically, `scoup` should be an invaluable resource.

It is acknowledged that `scoup` is not without limitations. Specifically: (a.) only symmetric trees are allowed. Empirical non-balanced trees are expected to be accommodated in future versions of the package; (b.) implementation of the OU process is stepwise, because the jump size (Δt) is taken as the node length whereas a completely continuous implementation of the process might improve the simulation output; and (c.) an effective caching technique may reduce the high computational cost of `scoup` with, for example, the package taking an average of 5.07 (standard deviation $= 0.31$) seconds over 50 repetitions of the sample code in Figure 1 on a *2.9GHz Quad-Core Intel Core i7 Ventura 13.6.7 MacBook Pro 2017* computer. Despite these limitations, we anticipate that `scoup` will become an important reference tool when codon sequences need to be simulated to validate and benchmark selection inference models. We are particularly enthusiastic because of the ability of the simulator to permit exploration of natural selection as a static and varying stochastic landscape phenomenon.

Simulations have played important roles in assessing and understanding some of the uncertainties in molecular evolutionary inference [41] and in the surging molecular ecology discipline of landscape genomics [42, 43]. There are however complex questions that continuously develop from the unrelenting pace with which molecular data are being generated [44]. There is also the void in our limited knowledge about the effectiveness of ω with respect to adaptive evolution [45], where tools that permit exploration of the concept could prove very useful. Simulators, like `scoup`, that can handle these complexities are therefore necessary to advance existing knowledge in different fields of modern biology.

5 Competing interests

We declare that we have no competing interests.

6 Funding

We have no funding to declare.

7 Data Accessibility Statement

Simulated sequences obtained from `scoup`, a script that contains the code used to execute the simulations with the package, a sample script that contains the `PAML` code that produced the ω estimates presented herein as well as the estimates obtained, are available as supplementary files on Github (`https://github.com/thsadiq/scoup/tree/main/vignettes/exData`). Code, detailed documentation and tutorial files that are designed to facilitate utility of the package are available on the Bioconductor platform at `doi.org/10.18129/B9.bioc.scoup`.

8 Author Contributions

HS: concept, code, data, analyses and original draft. DPM: concept, supervision and final draft.

9 Acknowledgements

We thank Ben Murrell for suggesting modelling varying selection coefficients with an OU process. Computations were performed using the HPC1 (url: `http://www.sun.ac.za/hpc`) facility at Stellenbosch University, South Africa.

References

[1] Jacques, F., Bolivar, P., Pietras, K., et al. Roadmap to the Study of Gene and Protein Phylogeny and Evolution – A Practical Guide. *PLoS One*, 18(2):e0279597, 2023.

[2] Yang, Z. PAML 4: Phylogenetic Analysis by Maximum Likelihood. *MBE*, 24(8):1586–1591, 2007.

[3] Kosakovsky Pond, S. L., Poon, A. F. Y., Velazquez, R., et al. HyPhy 2.5 – A Customizable Platform for Evolutionary Hypothesis Testing Using Phylogenies. *MBE*, 37(1):295–299, 2020.

[4] Gentleman, R. C., Carey, V. J., Bates, D. M., et al. Bioconductor: Open Software Development for Computational Biology and Bioinformatics. *Genome Biol.*, 5(10):R80, 2004.

[5] Halpern, A. L. and Bruno, W. J. Evolutionary Distances for Protein-Coding Sequences: Modelling Site-Specific Residue Frequencies. *MBE*, 15(7):910–917, 1998.

[6] Gearty, W., O'Meara, B., Berv, J., et al. CRAN Task View: Phylogenetics, 2024. Version 2024-01-30.

[7] Paradis, E. and Schliep, K. ape 5.0: An Environment for Modern Phylogenetics and Evolutionary Analyses in R. *Bioinformatics*, 35(3):526–528, 2019.

[8] Cressler, C. E., Butler, M. A., and King, A. A. Detecting Adaptive Evolution in Phylogenetic Comparative Analysis Using the Ornstein-Uhlenbeck Model. *Syst. Biol.*, 64(6):953–968, 2015.

[9] Pennell, M. W., Eastman, J. M., Slater, G. J., et al. geiger v2.0: An Expanded Suite of Methods for Fitting Macroevolutionary Models to Phylogenetic Trees. *Bioinformatics*, 30(15):2216—2218, 2014.

[10] Haller, B. C. and Messer, P. W. SLiM 4: Multispecies Eco-Evolutionary Modeling. *Am. Nat.*, 201(5): 619–762, 2023.

[11] Matthey-Doret, R. SimBit: A High Performance, Flexible and Easy-to-Use Population Genetic Simulator. *Mol. Ecol. Resour.*, 21(5):1745–1754, 2021.

[12] Rambaut, A. and Grass, N. C. Seq-Gen: An Application for the Monte Carlo Simulation of DNA Sequence Evolution Along Phylogenetic Trees. *Bioinformatics*, 13(3):235–238, 1997.

[13] Fletcher, W. and Yang, Z. INDELible: A Flexible Simulator of Biological Sequence Evolution. *MBE*, 26 (8):1879–1888, 2009.

[14] Sipos, B., Massingham, T., Jordan, G. E., et al. PhyloSim – Monte Carlo Simulation of Sequence Evolution in the R Statistical Computing Environment. *BMC Bioinformatics*, 12(104), 2011.

[15] Schliep, K. P. phangorn: Phylogenetic Analysis in R. *Bioinformatics*, 27(4):592—593, 2011.

[16] Maio, N. D., Boulton, W., Weilguny, L., et al. phastSim: Efficient Simulation of Sequence Evolution for Pandemic-Scale Datasets. *PLoS Comput. Biol.*, 18(4):e1010056, 2022.

[17] Ly-Trong, N., Barca, G. M. J., and Minh, B. Q. AliSim-HPC: Parallel Sequence Simulator for Phylogenetics. *Bioinformatics*, 39(9):btad540, 2023.

[18] Spielman, S. J. and Wilke, C. O. Pyvolve: A Flexible Python Module for Simulating Sequences along Phylogenies. *PLoS One*, 10(9):e0139047, 2015.

[19] Arenas, M. and Posada, D. Simulation of Genome-wide Evolution under Heterogeneous Substitution Models and Complex Multispecies Coalescent Histories. *MBE*, 31(5):1295–1301, 2014.

[20] Mustonen, V. and Lässig, M. From Fitness Landscapes to Seascapes: Non-Equilibrium Dynamics of Selection and Adaptation. *Trends Genet.*, 25(3):111–119, 2009.

[21] Spielman, S. J. and Wilke, C. O. The Relationship between dN/dS and Scaled Selection Coefficients. *MBE*, 32(4):1097–1108, 2015.

[22] Bartoszek, K., Glémin, S., Kaj, I., et al. Using the Ornstein–Uhlenbeck Process to Model the Evolution of Interacting Populations. *J. Theor. Biol.*, 429:35–45, 2017.

[23] Uhlenbeck, G. E. and Ornstein, L. S. On the Theory of the Brownian Motion. *Phys. Rev*, 36:823–841, 1930.

[24] Wright, S. Evolution in Mendelian Populations. *Genetics*, 16:97–159, 1931.

[25] Hasegawa, M., Kishino, H., and Yano, T. Dating of the Human-Ape Splitting by a Molecular Clock of Mitochondrial DNA. *J. Mol. Evol.*, 22:160–174, 1985.

[26] Kryazhimskiy, S. and Plotkin, J. B. The Population Genetics of $\mathrm{dN/dS}$. *PLoS Genet.*, 4(12):e1000304, 2008.

[27] Bershtein, S., Serohijos, A. W. R., and Shakhnovich, E. I. Bridging the Physical Scales in Evolutionary Biology: From Protein Sequence Space to Fitness of Organisms and Populations. *Curr. Opin. Struct. Biol.*, 42:31–40, 2017.

[28] Sella, G. and Hirsh, A. E. The Application of Statistical Physics to Evolutionary Biology. *PNAS*, 102(27): 9541–9546, 2005.

[29] Thurman, T. J. and Barrett, R. D. H. The Genetic Consequences of Selection in Natural Populations. *Mol. Ecol.*, 25(7):1429–1448, 2016.

[30] Nielsen, R. and Yang, Z. Estimating the Distribution of Selection Coefficients from Phylogenetic Data with Applications to Mitochondrial and Viral DNA. *MBE*, 20(8):1231–1239, 2003.

[31] Kimura, M. Model of Effectively Neutral Mutations in which Selective Constraint is Incorporate. *PNAS*, 76(7):3440–3444, 1979.

[32] Rowe, W., Platt, M., Wedge, D. C., et al. Analysis of a Complete DNA-Protein Affinity Landscape. *J. R. Soc. Interface*, 7(44):397–408, 2010.

[33] McCandlish, D. M. Visualizing Fitness Landscapes. *Evolution*, 65(6):1544–1558, 2011.

[34] Ayala, F. J. and Campbell, C. A. Frequency-Dependent Selection. *Annu. Rev. Ecol. Syst.*, 5:115–138, 1974.

[35] Gupta, M. K. and Vadde, R. Next-Generation Development and Application of Codon Model in Evolution. *Front. Genet.*, 14, 2023.

[36] Butler, M. A. and King, A. A. Phylogenetic Comparative Analysis: A Modeling Approach for Adaptive Evolution. *Am. Nat.*, 164(6):683–695, 2004.

[37] Säll, T. and Bengtsson, B. O. *Understanding Population Genetics*. Wiley-Blackwell, 2017.

[38] Nielsen, R. and Yang, Z. Likelihood Models for Detecting Positively Selected Amino Acid Sites and Applications to the HIV-1 Envelope Gene. *Genetics*, 148(3):929–936, 1998.

[39] Fijarczyk, A. and Babik, W. Detecting Balancing Selection in Genomes: Limits and Prospects. *Mol. Ecol.*, 24(14):3529–3545, 2015.

[40] Gillespie, D. T. The Mathematics of Brownian Motion and Johnson Noise. *Am. J. Phys.*, 64(3):225–240, 1996.

[41] Arenas, M. Simulation of Molecular Data under Diverse Evolutionary Scenarios. *PLoS Comput. Biol.*, 8 (5):e1002495, 2012.

[42] Riginos, C. and Jahnke, M. Comparative Landscape Genomics has Arrived with a Splash. *Mol. Ecol.*, 32 (24):6725–6728, 2023.

[43] Andrew, R. L., Bernatchez, L., Bonin, A., et al. A Road Map for Molecular Ecology. *Mol. Ecol.*, 22(10): 2605–2626, 2013.

[44] Nassar, L. R., Barber, G. P., Benet-Pagès, A., et al. The UCSC Genome Browser Database: 2023 Update. *Nucleic Acids Res.*, 51(D1):D1188–D1195, 2023.

[45] Parto, S. and Lartillot, N. Molecular Adaptation in Rubisco: Discriminating Between Convergent Evolution and Positive Selection Using Mechanistic and Classical Codon Models. *PLoS One*, 13(4):e0196267, 2018.