The Journal of Open Source Software

Ngesh: a Python library for synthetic phylogenetic data

Tiago Tresoldi[1, 2]

1 Department of Linguistics and Philology, Uppsala University **2** Department of Linguistic and Cultural Evolution, Max Planck Institute for Evolutionary Anthropology

DOI: 10.21105/joss.03173

Software

- Review
- Repository
- Archive

Editor: Mark A. Jensen

Reviewers:

- @DavidNickle
- @rvosa

Submitted: 24 March 2021
Published: 07 October 2021

Summary

This work presents `ngesh`, a Python library for simulating phylogenetic trees and data, designed for usage in development, debugging, and benchmarking of analysis pipelines and methods for phylogenetic inference, particularly in historical linguistics and stemmatics. The package generates reproducible stochastic simulations of evolution according to various criteria, including character mutation rates and probability of horizontal transfer, and its results can include the simulation of inadequate data compilation and sampling. Different output formats are supported, both for visualization (such as plain text and with integrated graphical viewers) and for software interoperability (such as Newick and NEXUS).

Background

Computational phylogenetics is being increasingly accepted in fields beyond biology, such as historical linguistics (Bouckaert et al., 2012) and stemmatics (Robinson, 2016). Stochastic simulations, long advocated for natural sciences in general (Bailey, 1964) and genetics in specific (Foote et al., 1999; Harmon, 2018), are not used enough in these fields. Nonetheless, they are very desirable, allowing to evaluate evolutionary analogies, models, and performance through vast amounts of simulated histories, without limits imposed by data availability and collection time, with quantifiable precision of results. Simulations can also be used to perform fuzzy testing of software and to support studies on which evolutionary models, processes, and evolutionary parameters better match the observed phenomena.

The `ngesh` library is a tool that allows to perform such simulations, designed for easy integration into phylogenetic pipelines. It can generate reproducible trees and correlated data following both user-established parameters, such as ratios of birth and death, and constraints, such as branch lengths and minimum number of taxa. The library can label taxa progressive enumeration or with random names that are easy to pronounce (e.g., "Sume" and "Fekobir") or which imitate the binominal nomenclature (e.g., "Sburas wioris" and "Zurbata pusso"). Character evolution related to the tree topology can likewise be simulated, including *ex novo* mutations and horizontal gene transfers. Results can be manipulated in diverse manners, for example by pruning extinct leaves or simulating uneven sampling. The simulated trees are standard ETE3 objects (Huerta-Cepas et al., 2016) and may be exported into different formats such as Newick trees, ASCII-art representation, and tabular lists.

Statement of need

The library addresses the need of more tools to investigate and teach phylogenetics in historical linguistics and stemmatics. As a building block for evaluating pipelines of analysis, it is an

alternative to the basic technique of randomizing taxa placement in existing cladograms, and to simpler tools such as the one by Noutahi (2017) or the `populate()` method of ETE3's `Tree` class (Huerta-Cepas et al., 2016). While there are many other alternatives available for simulating trees, including `TreeSim` (Stadler, 2011), `geiger` (Pennell et al., 2014), `ape` (Paradis & Schliep, 2018), and `DendroPy` (Sukumaran & Holder, 2021), `ngesh` compares favorably in historical linguistics and stemmatics. For the former, it provides default parameters that produce trees closer to those found in the field, particularly in terms of the simulation of horizontal transfers (i.e., loanword), all while using formats that better fit the existing linguistic pipelines, such as CLDF (Forkel et al., 2018), and laying ground for the usage of different character values (such as sound changes) besides the default cognate-sets for modelling lexical replacement. For the latter, where Bayesian phylogenetics have been gaining traction at a slower pace, the library constitutes the first general-purpose tool available and should help make these methods for popular.

Installation, Usage, & Examples

Users can install the library with the standard `pip` tool for managing Python packages. Trees can be generated from the command-line, defaulting to small phylogenies in Newick format:

```
$ ngesh
(Ukis:1.11985,(Koge:0.880823,(Rozkob:0.789548,(Meu:0.706601,
(((Felbuh:0.189693,Kefa:0.189693)1:0.117347,((Epib:0.153782,
Vugog:0.153782)1:0.0884745,Puluk:0.242256)1:0.0647836)1:0.0469885,
Efam:0.354028)1:0.352573)1:0.0829465)1:0.0912757)1:0.23903);
```

The tool supports both configuration files and command-line flags that take precedence over the former. Here we specify a model to generate Nexus data for a reproducible Yule tree, with a birth rate of 0.75, at least 5 leaves, "human" labels, and 20 presence/absence features:

```
$ cat my_tree.conf
[Config]
labels=human
birth=0.75
death=0.0
output=nexus
min_leaves=5
num_chars=20
$ ngesh -c my_tree.conf --seed 12345
begin data;
  dimensions ntax=6 nchar=33;
  format datatype=standard missing=? gap=-;
  matrix
Buza      111110110111011011010101000100110
Lenlar    111111010110111101100010010011001
Mukom     111110111011011011101001000100110
Pagil     111110110111011011100100100100110
Suglu     111110110111011011100011001001010
Wite      111110110111011011100101000100110
  ;
end;
```

Despite the benefit of a stand-alone tool, the package is designed to be run as a library. The two primary functions are `gen_tree()`, which returns a random tree, and `add_characters()`,

which adds character evolution data to a tree. Users can generate random trees without character information or simulate character evolution within existing trees, including non-simulated ones.

```
>>> import ngesh
>>> tree = ngesh.gen_tree(1.0, 0.5, max_time=0.3, labels="bio",
                          seed="135")
>>> print(tree)

   /-Lubedsas larpes
--|
  |   /-Rasso wimapudda
   \-|
      \-Sbaes rapis
>>> print(tree.write())
(Lubedsas larpes:0.201311,(Rasso wimapudda:0.0894405,Sbaes rapis:0.0894405)
1:0.11187);
>>> tree = ngesh.add_characters(tree, 15, 2.0, 0.5)
```

Besides the `write()` method above, which outputs Newick trees, results can be exported in either NEXUS format with `tree2nexus()` or in a textual tabular format with `tree2wordlist()`. Phylogenetic reconstruction can then be carried either by manually building an XML model for BEAST2 (Bouckaert et al., 2019) (normally with the aid of the graphical interface BEAUTi) or by using tools such as BEASTling (Maurits et al., 2017), producing a tree distribution. This distribution can be summarized to a maximum clade credibility ("MCC") tree with phylogenetic packages, allowing both visual and quantitative comparisons. A demonstration of such steps is provided with the user documentation ("Integrating with other software").

Code and Documentation Availability

The `ngesh` source code is available on GitHub at https://github.com/tresoldi/ngesh.

User documentation is available at https://ngesh.readthedocs.io/.

Acknowledgements

The author has received funding from the Riksbankens Jubileumsfond (ID: MXM19-1087:1, "Cultural Evolution of Texts") and from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (No. ERC #715618, "Computer-Assisted Language Comparison").

References

Bailey, N. T. J. (1964). *The elements of stochastic processes with applications to the natural sciences*. John Wiley & Sons. https://doi.org/10.2307/2333730

Bouckaert, R., Lemey, P., Dunn, M., Greenhill, S. J., Alekseyenko, A. V., Drummond, A. J., Gray, R. D., Suchard, M. A., & Atkinson, Q. D. (2012). Mapping the origins and expansion of the indo-european language family. *Science*, *337*(6097), 957–960. https://doi.org/10.1126/science.1219669

Bouckaert, R., Vaughan, T. G., Barido-Sottani, J., Duchêne, S., Fourment, M., Gavryushkina, A., Heled, J., Jones, G., Kühnert, D., De Maio, N., Matschiner, M., Mendes, F. K., Müller, N. F., Ogilvie, H. A., Plessis, L. du, Popinga, A., Rambaut, A., Rasmussen, D., Siveroni, I., … Drummond, A. J. (2019). BEAST 2.5: An advanced software platform for bayesian evolutionary analysis. *PLOS Computational Biology*, *15*(4), 1–28. https://doi.org/10.1371/journal.pcbi.1006650

Foote, M., Hunter, J. P., Janis, C. M., & Sepkoski, J. J. (1999). Evolutionary and preservational constraints on origins of biologic groups: Divergence times of eutherian mammals. *Science*, *283*(5406), 1310–1314. https://doi.org/10.1126/science.283.5406.1310

Forkel, R., List, J.-M., Greenhill, S. J., Rzymski, C., Bank, S., Cysouw, M., Hammarström, H., Haspelmath, M., Kaiping, G. A., & Gray, R. D. (2018). Cross-linguistic data formats, advancing data sharing and re-use in comparative linguistics. *Scientific Data*, *5*(1), 1–10. https://doi.org/10.1038/sdata.2018.205

Harmon, L. J. (2018). *Phylogenetic comparative methods: Learning from trees*. CreateSpace Independent Publishing. https://doi.org/10.32942/osf.io/e3xnr

Huerta-Cepas, J., Serra, F., & Bork, P. (2016). ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data. *Molecular Biology and Evolution*, *33*(6), 1635–1638. https://doi.org/10.1093/molbev/msw046

Maurits, L., Forkel, R., Kaiping, G. A., & Atkinson, Q. D. (2017). BEASTling: A software tool for linguistic phylogenetics using BEAST 2. *PloS One*, *12*(8). https://doi.org/10.1371/journal.pone.0180908

Noutahi, M.-R. (2017). *How to simulate a phylogenetic tree?* https://mrnoutahi.com/2017/12/05/How-to-simulate-a-tree/

Paradis, E., & Schliep, K. (2018). Ape 5.0: An environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics*, *35*, 526–528. https://doi.org/10.1093/bioinformatics/bty633

Pennell, M., Eastman, J., Slater, G., Brown, J., Uyeda, J., FitzJohn, R., Alfaro, M., & Harmon, L. (2014). Geiger v2.0: An expanded suite of methods for fitting macroevolutionary models to phylogenetic trees. *Bioinformatics*, *30*, 2216–2218. https://doi.org/10.1093/bioinformatics/btu181

Robinson, P. (2016). The digital revolution in scholarly editing. *Ars Edendi Lecture Series*, *4*, 181–207. https://doi.org/10.16993/baj.h

Stadler, T. (2011). Simulating trees with a fixed number of extant species. *Systematic Biology*, *60*(5), 676–684. https://doi.org/10.1093/sysbio/syr029

Sukumaran, J., & Holder, M. T. (2021). *The DendroPy phylogenetic computing library documentation. Retrieved apr 08, 2021*. http://dendropy.org/