The Journal of Open Source Software

treex: a Python package for manipulating rooted trees

Romain Azais[1], **Guillaume Cerutti**[1], **Didier Gemmerle;**[2], **and Florian Ingels**[1]

1 Laboratoire Reproduction et Developpement des Plantes, Univ Lyon, ENS de Lyon, UCB Lyon 1, CNRS, INRA, Inria, F-69342, Lyon, France **2** Universite de Lorraine, CNRS, IECL, F-54000 Nancy, France

DOI: 10.21105/joss.01351

Software

- Review
- Repository
- Archive

Submitted: 18 March 2019
Published: 24 June 2019

Summary

Trees form an expanded family of combinatorial objects that offers a wide range of application fields, from plant modeling to XML files analysis through study of lineage trees. One can also mention classification trees that arise in statistical learning, as well as game trees. In all these examples, the underlying structure is a rooted tree which can be ordered (if the order of siblings is significant) or not. For the sake of modeling, the vertices of a tree might be augmented with labels describing their properties: for example, in a cell lineage tree, each vertex represents a cell and the labels might describe its volume, the tissue type, etc. Despite the diversity of applications, one may want a unique data structure to encode rooted trees and perform operations on them.

`treex` is a Python package for manipulating rooted trees, ordered or not, with or without labels on their vertices. Basically, `treex` allows (i) random generation of trees, (ii) edit operations (e.g., add or remove vertices or labels), (iii) visualization of structures and their properties (in command line or a `Matplotlib` figure and exportation to `TeX`), (iv) conversion to different formats, and (v) application of various algorithms. Concerning the latter two, coding processes (Pitman, 2006) have been implemented, as well as DAG compression (Godin & Ferraro, 2010). In addition, comparison between trees can be performed via an edit distance algorithm (Azais, Durand, & Godin, 2019). Self-nested approximations of trees (Godin & Ferraro, 2010, Azais (2017), Azais et al. (2019)) have been implemented through different algorithms. To the best of our knowledge, `treex` is the only Python library that permits encoding of rooted trees in various formats together with such a diversity of treatments. Let us mention the related Java library `TED` (Pawlik & Augsten, 2016) for efficient edit distance algorithms.

`treex` offers converters to the standard encoding of nested brackets (see for instance (Aho, Hopcroft, & Ullman, 1974)) and L-strings as manipulated by `L-Py`, a simulation framework for modeling plant architectures (Boudon, Pradal, Cokelaer, Prusinkiewicz, & Godin, 2012). Numerical experiments and/or figures of recent publications (Azais, Genadot, & Henry, 2019, Azais (2017), Azais et al. (2019)) have been made using the current or previous versions of `treex`. Furthermore, ongoing academic projects on the development and implementation of supervised classification methods for tree data, the study of lineage trees, as well as investigations on plant modeling, make intensive use of structures and algorithms implemented in `treex`.

`treex` is open source and distributed under the LGPL License. The source code is hosted on GitLab. Releases are automatically built and tested for 64-bit Linux and Mac OS X machines using Jenkins CI, and packaged on Anaconda Cloud (Azais, Cerutti, Gemmerle, & Ingels, 2019a). The documentation of all classes, methods and functions is built upon release and made available online (Azais, Cerutti, Gemmerle, & Ingels, 2019b).

Acknowledgements

The authors are grateful to Christophe Godin for valuable discussions about tree structures and algorithms and Jonathan Legrand for significant help on continuous integration. The work of R.A. and F.I. has been partially supported by the European Union's H2020 project ROMI.

References

Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1974). *The design and analysis of computer algorithms* (1st ed.). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Azais, R. (2017). Nearest embedded and embedding self-nested trees. *Preprint*.

Azais, R., Cerutti, G., Gemmerle, D., & Ingels, F. (2019a). Treex: Archive. Retrieved February 27, 2019, from https://anaconda.org/MOSAIC/treex

Azais, R., Cerutti, G., Gemmerle, D., & Ingels, F. (2019b). Treex: Documentation. Retrieved February 27, 2019, from https://azais.gitlabpages.inria.fr/treex

Azais, R., Durand, J.-B., & Godin, C. (2019). Approximation of trees by self-nested trees. In *2019 proceedings of the twenty-first workshop on algorithm engineering and experiments (alenex)* (pp. 39–53). doi:10.1137/1.9781611975499.4

Azais, R., Genadot, A., & Henry, B. (2019). Inference for conditioned galton-watson trees from their harris path. *ALEA*. doi:10.30757/ALEA.v16-21

Boudon, F., Pradal, C., Cokelaer, T., Prusinkiewicz, P., & Godin, C. (2012). L-py: An l-system simulation framework for modeling plant architecture development based on a dynamic language. *Frontiers in Plant Science*, *3*, 76. doi:10.3389/fpls.2012.00076

Godin, C., & Ferraro, P. (2010). Quantifying the degree of self-nestedness of trees: Application to the structural analysis of plants. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, *7*(4), 688–703. doi:10.1109/TCBB.2009.29

Pawlik, M., & Augsten, N. (2016). Tree edit distance. Retrieved February 27, 2019, from http://tree-edit-distance.dbresearch.uni-salzburg.at/

Pitman, J. (2006). *Combinatorial stochastic processes*. Lecture notes in mathematics (Vol. 1875). Berlin: Springer-Verlag. doi:10.1007/b11601500