**TREEasy**

Manual version 1.0

May, 2019

**1. Introduction**

Recently, methods to species trees and phylogenetic networks have become commonly used tools (Bastide et al., 2018; Liu et al., 2015; Morrison, 2014; Szöllősi et al., 2014). However, the number of programs, and the handling time of learning to manipulate each program is an impediment to research. There is no tool to integrate sequence alignment, gene tree reconstruction, species tree inference, and phylogenetic network inference. Here, we present a multi-thread open-source tool, named TREEasy, to infer species tree and phylogenetic network from multilocus molecular sequences directly (https://github.com/MaoYafei/TREEasy).

TREEasy is under GNU General Public License v3.0.

The program has to run on the current directory.

**2. Linux System**

This program will run on Linux System. Windows and MacOS versions will be available soon. However, Ubuntu is a good alternative tool to run Linux system on Windows and MacOS.

Ubuntu download: <https://www.ubuntu.com/download/desktop>

Ubuntu install: <https://tutorials.ubuntu.com/tutorial/tutorial-install-ubuntu-desktop?_ga=2.239894848.218635587.1557441498-2036767044.1557441498#0>

Moreover, for Windows users, Cygwin (<http://www.cygwin.com/>) / VirtualBox (<https://www.virtualbox.org/>) / other programs are also useful. For MacOS users, Fedora(<https://getfedora.org/>) / Linuxmint (<https://linuxmint.com/>

) /VirtualBox (<https://www.virtualbox.org/>) / other programs are useful too.

**3. Installation**

TREEasy is written in Python integrating sequence alignment, gene tree reconstruction, species tree inference, and phylogenetic network inference. BioPython must be installed and a few executable dependencies are needed: MAFFT (Nakamura et al., 2018), Translatorx (Abascal et al., 2010), AMAS (Borowiec, 2016), IQ-TREE (Nguyen et al., 2014), ASTRAL (Mirarab et al., 2014), MP-EST (Liu et al., 2010), STELLS2 (Pei and Wu, 2017), PhyloNet (Wen et al., 2018) and SNaQ (Solís-Lemus et al., 2017).

**Note: The below installation examples are based on Linux system.**

*Python packages: BioPython;* <http://biopython.org/>

wget http://biopython.org/DIST/biopython-1.73.tar.gz

python setup.py install --user

*Python packages: ete3;* <http://etetoolkit.org/download/>

wget https://files.pythonhosted.org/packages/21/17/3c49b7fafe10ed63bb7904ebf9764b98db726aa5fd482fb006818854bc04/ete3-3.1.1.tar.gz

python setup.py install --user

*MAFFT:* <https://mafft.cbrc.jp/alignment/software/source.html>

wget https://mafft.cbrc.jp/alignment/software/mafft-7.427-with-extensions-src.tgz

gunzip -cd mafft-x.x-src.tgz | tar xfv -

cd mafft-x.x/core/

make clean

make

make install

cp mafft-x.x /somewhere/bin/mafft

*IQ-TREE:* <http://www.iqtree.org/#download>

wget https://github.com/Cibiv/IQ-TREE/releases/download/v1.6.10/iqtree-1.6.10-Linux.tar.gz

tar zxvf iqtree-1.6.10-Linux.tar.gz

cd iqtree-1.6.10-Linux/bin/

cp iqtree /somewhere/bin/

*MP-EST:* <http://faculty.franklin.uga.edu/lliu/mp-est>

wget http://faculty.franklin.uga.edu/lliu/sites/faculty.franklin.uga.edu.lliu/files/mpest\_2.0.zip

unzip mpest\_2.0

cd mpest\_2.0/linux/

cp mpest /somewhere/bin/

*STELLS2:* <https://github.com/yufengwudcs/STELLS2>

git clone <https://github.com/yufengwudcs/STELLS2.git>

cd STELLS2

cp stells-v2-1-0-linux64 /somewhere/bin/stells-v2

*Java:* <https://java.com/en/download/help/>

*Julia:* <https://julialang.org/downloads/oldreleases.html>

Warning: SNaQ only can run on v0.6.4 julia

*SNaQ:* <https://github.com/crsl4/PhyloNetworks.jl>

julia

using Pkg

Pkg.add("PhyloNetworks")

*Translatorx, AMAS, PhyloNet, ASTRAL*

Note: we pre-install all these programs in the current directory. Users do not need to install them again.

**4. Examples**

The below is an example for command line running, while, the GUI example will be shown in Section7 GUI later.

*Simulated date example:*

cd examples

tar zxvf Simulated\_data.tar.gz

cd ..

python TREEasy.py -d examples/6taxon\_300genes\_4cores -s examples/6taxon\_spe\_name.txt -g examples/6taxon\_gene\_name.txt -b 0 -r A -n 3 -k 10 -t 4 -c noncds

*Acropora example:*

cd examples

tar zxvf Acropora\_data.tar.gz

cd ..

python TREEasy.py -d examples/Acropora -s examples/Acro\_spe\_name.txt -g examples/Acro\_gene\_name.txt -b 50 -r Aten -n 3 -k 3 -t 8 -c CDS

**5. Inputs**

*1-1. For CDS data type (-c cds): 1-1-1. Nuclear\_sequence\_files in a directory, and each FASTA file contains nuclear sequences of all taxa (No missing data allowed). Moreover, each FASTA file must be named as \*\_nc.fasta.*

eg: CO1\_nc.fasta

>Taxon1

ATCG

>Taxon2

ATCC

...

*1-1-2. Protein\_sequence\_files in the same directory, and each FASTA file contains protein amino acid sequences of all taxa (No missing data allowed). Moreover, each FASTA file must be named as \*\_aa.fasta.*

eg: CO1\_aa.fasta

> Taxon1

PAPA

> Taxon2

PAPA

...

Note: Users can see the empirical data with follow commands:

cd examples

tar zxvf Acropora\_data.tar.gz

less Acropora/led2829\_nc.fasta

less Acropora/led2829\_aa.fasta

*1-2. For Non-CDS data type (-c noncds): Nuclear\_sequence\_files in a directory, and each FASTA file contains molecular sequences of all taxa (No missing data allowed). Moreover, each of FASTA file must be named as \*.fasta.*

eg: CO1.fasta

>Taxon1

ATCG

>Taxon2

ATCC

...

Note: Users can see the empirical data with follow commands:

cd examples

tar zxvf Simulated\_data.tar.gz

less 6taxon\_300genes\_4cores /1\_seqgen1.nex.fasta

*1-3. Species-individual(s) information (-s). There are at least three columns separated by tab. The first column is a species name. The second column is the number of individuals for a species. The third, fourth and continued columns are individual names.*

eg:

Adif 1 Adif

Aech 1 Aech

Agem 2 Agem-1 Agem-2

...

Note: Users can see the empirical data with follow commands:

cd examples

tar zxvf Acropora\_data.tar.gz

less Acro\_spe\_name.txt

*1-4. Gene - species name information (-g). There are two columns separated by tab. The first column is a species name. The second column is a gene name.*

eg:

Adif Adif\_sc0000028.g769.t1

Asub Asub\_sc0000129.g5077.t1

Adif Adif\_sc0000104.g157.t1

...

Note: Users can see the empirical data with follow commands:

cd examples

tar zxvf Acropora\_data.tar.gz

less Acro\_gene\_name.txt

*1-5. Other required inputs (-r/-n/-t/-k/-b) could see Section6 Usage below.*

**6. Usage**

-d: A directory path. This directory contains the required input files.

-s: A text file. This file contains species-individual(s) information.

-g: A text file. This file contains gene-species name information.

-b: A value from 0 to 100. This parameter is a cutoff value for bootstrap values in gene trees. If bootstrap values of all nodes in a gene tree are greater than the cutoff value, the gene tree will be used to infer species tree and networks.

-r: (a) species name(s). The TREEasy regards this taxon as root. If you have multi-outgroups, use “,” to separate species names.

eg:

Aten

Asub, Agem

-n: An integer. This parameter is the maximum number of hybridizations that can be inferred. This value has a big impact on running time. We suggest that this value should be less than 5.

-k: An integer. This value is a cross-validation value for PhyloNet. This value has to be a multiple of the number of gene trees which are used to infer the phylogenetic network.

-t: An integer. This parameter is the number of processors to be used. We do not suggest using all processors in your device to run this program.

-c: Data type. Data type. This parameter describes the input data type. There are two possible data types for the TREEasy. The first type is protein-coding sequences (CDS). You have to prepare two FASTA files for one gene. The first is the protein amino acid sequences in FASTA file named as XXX\_aa.fasta (eg. CO1\_aa.fasta), the other is corresponding nucleotide sequences in FASTA file named as XXX\_nc.fasta (eg. CO1\_nc.fasta). The second type is non-cds. Only one FASTA file is required as input named as XXX.fasta (eg. CO1.fasta).

**7. GUI**

You could start GUI by simply double click the file named *TREEasy* or you can always start it with the following command line:

python TREEasy\_GUI.py

When the GUI is starting, the tool will check all dependencies by itself. Until all dependencies are installed properly, you will see a window like below:

Figure example:A screenshot of a cell phone

Description automatically generated

You have to put all parameters in the GUI window,

①Gene Locus Directory

②Gene Name File

③Species Name File

④Root Taxon(s) (separated by comma)

⑤Cross Value

⑥Maximum Network Number

⑦Bootstrap Value

⑧Gene/Locus Type (select CDS or nonCDS)

⑨Number of Threads

We provided two example configs in the GUI. You can select one from the top menu “Example”, and all parameters will be filled in automatically (see figure below). Make sure you have already decompressed Acropora\_test.tar.gz and Simulated\_data\_test.tar.gz before running the example configs.

A screenshot of a cell phone

Description automatically generated

After you fill out all parameters, click start, and a new window will show up (see figure below). Click start again, the execution will start. You can check the progress in the new window, or close the window to stop execution.

Figure here

**A screenshot of a cell phone

Description automatically generated8. Help**

If you have any problems on dependencies, please run below command firstly:

python TREEasy\_Help.py

If you have any problems, error-reports, other questions and cooperation opportunities about TREEasy, feel free to contact me (yafei.mao@oist.jp).

**9. Citing this program**

XXXX

**10. Copyright**

©by Yafei Mao, May 2019

**References**

Abascal, Federico, Rafael Zardoya, and Maximilian J. Telford. “TranslatorX: multiple alignment of nucleotide sequences guided by amino acid translations.” Nucleic acids research 38.suppl\_2 (2010): W7-W13.

Barraclough, T.G., and Nee, S. (2001). Phylogenetics and speciation. Trends in ecology & evolution 16, 391-399.

Bastide, P., Solis-Lemus, C., Kriebel, R., William Sparks, K., and Ané, C. (2018). Phylogenetic comparative methods on phylogenetic networks with reticulations. Systematic biology 67, 800-820.

Borowiec, M.L. (2016). AMAS: a fast tool for alignment manipulation and computing of summary statistics. PeerJ 4, e1660.

Katoh, Kazutaka, and Daron M. Standley. “MAFFT multiple sequence alignment software version 7: improvements in performance and usability.” Molecular biology and evolution 30.4 (2013): 772-780.

Liu, L., Xi, Z., Wu, S., Davis, C.C., and Edwards, S.V. (2015). Estimating phylogenetic trees from genome‐scale data. Annals of the New York Academy of Sciences 1360, 36-53.

Mao, Y., Economo, E.P. and Satoh, N., 2018. The Roles of Introgression and Climate Change in the Rise to Dominance of Acropora Corals. Current Biology, 28(21), pp.3373-3382.

Mirarab, S., Reaz, R., Bayzid, M.S., Zimmermann, T., Swenson, M.S., and Warnow, T. (2014). ASTRAL: genome-scale coalescent-based species tree estimation. Bioinformatics 30, i541-i548.

Morrison, D.A. (2014). Is the tree of life the best metaphor, model, or heuristic for phylogenetics? Systematic biology 63, 628-638.

Nguyen, L.-T., Schmidt, H.A., von Haeseler, A., and Minh, B.Q. (2014). IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. Molecular biology and evolution 32, 268-274.

Pei, J., and Wu, Y. (2017). STELLS2: fast and accurate coalescent-based maximum likelihood inference of species trees from gene tree topologies. Bioinformatics 33, 1789-1797.

Shen, X.-X., Opulente, D.A., Kominek, J., Zhou, X., Steenwyk, J.L., Buh, K.V., Haase, M.A.B., Wisecaver, J.H., Wang, M., and Doering, D.T. (2018). Tempo and Mode of Genome Evolution in the Budding Yeast Subphylum. Cell 175, 1533-1545. e1520.

Solís-Lemus, C., Bastide, P. and Ané, C., 2017. PhyloNetworks: a package for phylogenetic networks. Molecular biology and evolution, 34(12), pp.3292-3298.

Soltis, D., and Soltis, P. (2018). The Great Tree of Life. (Academic Press).

Szöllősi, G.J., Tannier, E., Daubin, V., and Boussau, B. (2014). The inference of gene trees with species trees. Systematic biology 64, e42-e62.

Wen, D., Yu, Y., Zhu, J., and Nakhleh, L. (2018). Inferring phylogenetic networks using PhyloNet. Systematic biology 67, 735-740.