User Level Documentation

Application of a test for reversibility of Markov chains in molecular evolution

Authors: Luzia Berchtold (11813328), Zhasmina Stoyanova (11806556)

Pipeline for the application of the tests Bowker, Stuart, Internal Symmetry and Quasi-symmetry on simulated or biological data. The software is executable on the command line. It comprises two bash scripts, which serve as a link between a C++ program and an R Script for the analysis of the provided data. The program uses IQ-TREE to simulate an alignment (AliSim) provided a tree topology or it can call IQ-TREE to compute a maximum-likelihood tree provided an alignment. In addition to this it provides the option for the computation of three saturation tests.

Background

We assume DNA evolution can be modelled using a Markovian model of sequence evolution. A common assumption when performing phylogenetic analysis is that the model is stationary and reversible. Provided a multiple sequence alignment (MSA) or a phylogenetic tree this pipeline can apply 4 different test:

- Bowker Test for Symmetry: stationarity + reversibility if Null hypothesis is kept
- Stuart Test for Symmetry: stationarity if Null hypothesis is kept
- Test for Quasi-Symmetry: reversibility if Null hypothesis is kept
- Test for Internal-Symmetry

Each test is applied pairwise on each unique sequence pair of the MSA. There is not yet a decision rule for the model on a bigger tree, but we aim to help with the pairwise analysis and visualisations. To avoid including non-informative pairs in the decision process, saturation tests can be applied on the MSA. These are applied pairwise again:

- Cassius' Test for Saturation 1: saturation if Null hypothesis is rejected (uses relative nucleotide frequency of the whole MSA)
- Cassius' Test for Saturation 2: saturation if Null hypothesis is rejected (uses relative nucleotide frequency of the sequence pair)
- Chi-square Test for Saturation: saturation if Null hypothesis is rejected

Structure

RevTest │ `analysis_simulation.sh` - Bash script for the analysis of simulated data │ `analysis_biological.sh` - Bash script for the analysis of real data └──bin │-------- │ `analysis.R` - the R script │-------- │ `all_tests.cpp` - the C++ script │-------- │ `all_tests.out` - the compiled C++ script │-------- └──lib - all libraries and header files needed for the C++ script │---------------- │ stats.h - the implementations of the four tests (Bowker, Stuart, Internal Symmetry, Quasi-symmetry) │---------------- │ sat_tests.h - the implementations of the three saturation tests │---------------- │ file_handling.h - code for reading in the files │---------------- │ Sequence.h - declarations of used class structures └──test_input - example input for testing

Installation & Dependencies

The software requires **R** and **IQ-Tree** to run.

C++ Script

The program uses an already compiled C++ code, that is able to run on Linux machines. If there is a problem with executing the compiled code and the following error occurs:

```
bash: ./bin/all_tests.out: Permission denied
```

you can try running the following command from the directory of the C++ script (here the `bin`). It will grant permission upon entering the password of the user.

```
sudo chmod 744 all_tests.out
```

R Script - built under R 4.2.2

R must be installed and be able to execute scripts by entering `Rscript`. It uses the following packages, that are installed automatically if not already present:

- ggpubr
- ggtree
- ggvenn
- phangorn
- tidytree
- tidyverse
- treeio

IQ-Tree

The software uses the AliSim extension, as well as ModelFinder, contained in the IQ-TREE software. The user has to have IQ-TREE with AliSim installed and be able to run IQ-TREE by simply entering `iqtree2`.

There is otherwise no installation needed, simply run the desired bash script in the command line, providing the needed input files.

Usage

To start, open a terminal and navigate to the path of the program.

`analysis_simulation.sh`

The analysis of simulated data requires a tree file and the parameters for the simulation.

```
bash analysis_simulation.sh -t treefile.tree -m JC -n 1000 -k 100 -s true
```

- -t - the tree file in standard Newick format
- -m - specifies a substitution model to use for the simulation (default: JC, all possible options can be seen in substitution models for alisim)
- -n - specifies the length of the root sequence (default: 500)
- -k - specifies how many simulations to be ran (default: 1)
- -s (optional) - can be true or false, if true it will also compute the saturation tests (default: false)

`analysis_biological.sh`

The analysis of biological data requires the multiple sequence alignment file and optionally a tree file.

```
bash analysis_biological.sh -a alignment.phy -t treename.nwk -s true
```

- -a - specifies the sequence alignment file in PHYLIP or NEX format
- -t (optional) - the tree file in standard Newick format
- -s (optional) - can be true or false, if true it will also compute the saturation tests (default: false)

If no tree file is available, there is an option to call IQ-TREE to compute the ML tree and to use it in the downstream analysis.

```
bash analysis_biological.sh -a alignment.phy -I -m GTR -s true
```

- -a - specifies the sequence alignment file in PHYLIP or NEX format
- -I - specifies whether to call IQ-TREE to compute the ML tree
- -m (optional) - specifies the model for IQ-TREE to use (default: none, if none IQ-TREE will call ModelFinder to infer which model to use, all substitution models)
- -s (optional) - can be true or false, if true it will also compute the saturation tests (default: false)

If there is no tree file provided and IQ-TREE is not called, the program will compute only the raw test statistics.

Output

The output is saved in a new folder created by the program, called `results_<treename>` (simulation study) or `results_<alignment>` (biological study), depending on the input file. The outputs are described with `<name>` as placeholder, depending on the input file. New runs override the contents of previous ones, if the folder is not moved.

Both scripts produce up to 4 `.csv` files.

- `results_raw_<name>.csv` - contains the raw values of the test statistics for each pair and for every test (saturation tests if chosen)
- `results_raw_<name>.csv` - contains the p-values against the null hypotheses with significance 0.05
- `results_rev_test.csv` - contains the results of the decision for each pair and the 4 tests (whether the null hypothesis is retained/rejected)
- `results_sat_test.csv` - contains the results of the decision for each pair and each saturation test (if chosen)

Additionally one PDF file for each of the tests computed:

- plot_Bowker_test.pdf - coloured tree plot, heatmap and distribution of test statistics for Bowker test
- plot_Stuart_test.pdf - coloured tree plot, heatmap and distribution of test statistics for Stuart test
- plot_IS_test.pdf - coloured tree plot, heatmap and distribution of test statistics for Test for Internal Symmetry
- plot_QS_test.pdf - coloured tree plot, heatmap and distribution of test statistics for Test for Quasi-Symmetry
- plot_Sat_Cassius1_test.pdf - coloured tree plot, heatmap and distribution of test statistics for Cassius' Test for Saturation 1
- plot_Sat_Cassius2_test.pdf - coloured tree plot, heatmap and distribution of test statistics for Cassius' Test for Saturation 2
- plot_Sat_Chi_test.pdf - coloured tree plot, heatmap and distribution of test statistics for Chi-square Test for Saturation
- venn_diag.pdf - Venn diagram for each pair (simulation study)/ all pairs combined (biological study), comparing the number of rejections in Bowker, Stuart and Quasi-Symmetry Test

Additionally for the biological study if there more than 50 species, a compressed tree separated in smaller subtrees is computed and saved in the PDF files.

Lastly the result tree with added labels on the branches for all test rejections in NEXUS format:

- annotated_<name>.tree

All command line messages in the process will be saved in simulation.log or biological.log.

Example Input

To test if everything is working correctly, there is a test input provided with an example tree file and example alignment. To run the software open a terminal and navigate to the directory, then execute the following commands:

Simulation Study

For the simulation study:

```
bash analysis_simulation.sh -t test_input/example-treefile.nwk -m JC -n 1000 -k 50
-s true
```

If everything worked there should be a new folder called results_example-treefile, containing all of the outputs.

Biological Study

For the biological study:

```
bash analysis_biological.sh -a test_input/example-alignment.phy -t
test_input/example-treefile-bio.nwk -s true
```

If everything worked there should be a new folder called `results_example-alignment`, containing all of the outputs.

References

[Gubela, 2022] Gubela, N. (2022). A test for reversibility of markov chains in molecular evolution. **[Kalyaanamoorthy et al., 2017]** Kalyaanamoorthy, S., Minh, B., Wong, T., von Haeseler, A., and Jermiin, L. (2017). Modelfinder: Fast model selection for accurate phylogenetic estimates. Nature Methods, 14. **[Ly-Trong et al., 2021]** Ly-Trong, N., Naser-Khdour, S., Lanfear, R., and Minh, B. Q. (2021). Alisim: A fast and versatile phylogenetic sequence simulator for the gen- omic era. bioRxiv. **[Wang et al., 2020]** Wang, L.-G., Lam, T. T.-Y., Xu, S., Dai, Z., Zhou, L., Feng, T., Guo, P., Dunn, C. W., Jones, B. R., Bradley, T., Zhu, H., Guan, Y., Jiang, Y., and Yu, G. (2020). treeio: an r package for phylogenetic tree input and output with richly annotated and associated data. Molecular Biology and Evolution, 37:599–603. **[Wickham, 2016]** Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. **[Yan, 2021]** Yan, L. (2021). ggvenn: Draw Venn Diagram by ‘ggplot2’. R package version 0.1.9.