quartetsampling Documentation

Release 1.3.0

James B. Pease
Stephen A. Smith
Cody E. Hinchliff
Joseph W. Brown
Joseph F. Walker

Aug 05, 2019

CONTENTS:

CHAPTER

ONE

GETTING STARTED

1.1 What is Quartet Sampling?

Quartet Sampling (QS) is a method for quantifying branch support values for a phylogenetic tree. The software requires an input tree topology (branch lengths not required or used) and a molecular alignment in phylip format. Optionally, a RAxML-formatted partition file may be used for either a concatenated alignment in partitioned mode or to establish gene barriers for using individual gene trees. The Quartet Sampling method takes a phylogenetic topology and considers each internal (i.e., non-terminal) branch. Each internal branch has four branches connected to it that lead to one or more taxa. For each replicate, Quartet Sampling randomly selects one taxon from each of the four groups and evaluates the likelihoods of the three possible topological arrangements of the four taxa. These replicates then form the basis of branch scores for the QS method. Full details about the scientific background and equations behind QS can be found in the Quartet Sampling paper in the American Journal of Botany at <http://dx.doi.org/10.1002/ajb2.1016>.

1.2 How do I cite this program?

James B Pease, Joseph W Brown, Joseph F Walker, Cody E Hinchliff, Stephen A Smith. 2018. Quartet Sampling distinguishes lack of support from conflicting support in the green plant tree of life. American Journal of Botany. 105(3): 385–403. doi:10.1002/ajb2.1016

Please also include the URL <https://www.github.com/fephyfofum/quartetsampling> in your methods section where the program is referenced.

Please also be sure to cite RAxML-ng, RAxML, PAUP, or IQ-TREE depending on which engine you use (the default is RAxML-ng).

1.3 Installation

No installation is required, quartetsampling scripts should work as long as Python and RAxML are installed. The repository can be cloned or downloaded as a .zip file from GitHub.

1.3.1 Requirements

- Python 3.x (2.7 may also work, but **3.x strongly recommended** for optimal parallelization) https://www.python.org/downloads/
- RAxML-ng 0.9.0 (beta versions not guaranteed) https://github.com/amkozlov/raxml-ng

Alternative likelihood engines available * RAxML 8.1+ (8.0.x and 7.x will not work) https://sco.h-its.org/exelixis/web/software/raxml/index.html * PAUP http://people.sc.fsu.edu/~dswofford/paup_test/ * IQ-TREE 1.6.x (earlier versions may not work) http://www.iqtree.org

*Optional, but recommended:**

- Numpy http://www.numpy.org (required only for `calc_qs_stats.py`)
- Scipy https://www.scipy.org (required only if `--calc-qdstats` is activated)
- Figtree http://tree.bio.ed.ac.uk/software/figtree/ (view FigTree output)

1.4 Preparing your data

1.4.1 Phylogeny

A phylogenetic tree in Newick (parenthetical) format should be used. Branch lengths are optional and will be ignored by the program. Removal of support scores in square brackets is recommended.

Note: Internal branch labels will be replaced in output trees.

1.4.2 Sequence Alignment

An alignment in Relaxed Phylip format (such as used for RAxML) is required. The alignment can be DNA nucleotides or amino acids (use `--amino-acid` in that case). If you have an alignment in FASTA format, we also include a script called `utils/fasta2phy.py` that will convert FASTA alignments to Relaxed Phylip.

Important: Labels for the alignment sequences must match the labels on tree terminal branches exactly. All tips in the phylogeny must have a sequence represented in the alignment. Sequences appearing in the alignment, but not in the tree are allowed, and will be ignored.

1.4.3 Partitions File (Optional)

A partitions file in the style of RAxML may also be used either for a partitioned analysis of likelihood or separate gene tree evaluation. DNA partitions should use the `DNA` prefix, and proteins should use the `WAG` prefix instead of `DNA` in the example below. Note that partition names are arbitrary in this case and that number ranges are inclusive.

Example file:: DNA, p1=1-30 DNA, p2=31-60 DNA, p3=61-90 . . .

Important: If your alignment is sparse and you use partitioned most, such that you will frequently have partitions with no sequence data for randomly selected quartets of taxa, you may wish to invoke the `--ignore-error` option in `quartet_sampling.py` to ignore the RAxML errors that will result from these empty partitions.

1.5 Basic usage

:: python3 quartet_sampling.py –tree TREE.nwk –align ALIGNMENT.phy –reps 100 –threads 4 –lnlike 2

1.5.1 Required Paramters

`--tree`: File containing a single Newick-formatted phylogeny.

`--align` Phylip-formatted alignment containing one sequence for each tip in the phylogeny provided.

`--reps` Number of replicates per branch to perform.

`--threads` Number of threads for Python to use in parallel (does not pass through to RAxML, this is for Python multiprocessing of per-branch replicates in parallel)

1.5.2 Recommended Parameters

`--lnlike`: log-likelihood threshold cutoff, determines the minimum difference by which the best likelihood tree must exceed the second-best likelihood tree when comparing the three possible topologies for a given quartet replicate.

Note: If `--lnlike` is omitted, this will invoke an alternative mode where a tree is simply inferred from the sequence data by RAxML (or PAUP*) and likelihoods are not evaluated. This will result in Quartet Informativeness (QI) scores of 'NA' for all branches.

CHAPTER

TWO

PROGRAM PARAMETER DESCRIPTIONS

2.1 quartet_sampling

2.1.1 Description

quartet_samping.py: Quartet Sampling method for phylogenetic branch support evaluation <http://www.github.com/FePhyFoFum/quartetsampling>

2.1.2 Parameters

`-h/--help`

Description: show this help message and exit

Type: boolean flag

`--align/--alignment` (required)

Description: Alignment file in “relaxed phylip” format, as used by RAxML.

Type: file path; **Default:** None

`--reps/--number-of-reps` (required)

Description: The number of replicate quartet topology searches to be performed at each node.

Type: integer; **Default:** 100

`--threads/--number-of-threads` (required)

Description: The number of parallel threads to be used by Python for quartet topology searches.

Type: integer; **Default:** 1

`--tree` (required)

Description: The input tree in Newick (parenthetical) format.

Type: file path; **Default:** None

`--calc-qdstats`

Description: EXPERIMENTAL: Calculates Chi-square test for QD tree frequencies. Use only if Scipy is available. Will increase running time.

Type: boolean flag

`--clade`

Description: Conduct analysis on specific clade identified by CSV taxon list

Type: string; **Default:** None

`--data-type`

Description: (nuc)leotide, (amino) acid, or (cat)egorical data

Type: None; **Default:** [‘nuc’]

Choices: (‘nuc’, ‘amino’, ‘cat’)

`--engine`

Description: Name of the program to use to infer trees or evaluate tree model likelihoods.

Type: None; **Default:** (‘raxml-ng’,)

Choices: (‘raxml-ng’, ‘raxml’, ‘paup’, ‘iqtree’)

`--engine-exec`

Description: Full file path of the tree inference or likelihood evaluation engine.

Type: None; **Default:** None

`--engine-model`

Description: Advanced: specify a custom model name for the tree engine

Type: None; **Default:** None

`--genetrees`

Description: Use partitions file (RAxML format) to divide the alignment into separate gene tree regions. Gene alignments will be sampled random for the quartet topology searches.

Type: file path; **Default:** None

`--ignore-errors`

Description: Ignore RAxML and PAUP erroneous runs

Type: boolean flag

`--lnlike/--lnlike-thresh`

Description: The lnlike threshhold that is the minimum value by which the log-likelihood value of the best-likelihood tree must be higher than the second-best-likelihood tree for the replicate to register as the best-likelihood topology rather than 'uncertain'. If set to zero, this turns off likelihood evaluation mode and invokes tree inference mode where a tree is simply inferred from the alignment without considering likelihood (QI values are N/A in this case).

Type: float; **Default:** 2.0

`--low-mem`

Description: Do not store large alignment in memory for whole-alignment (non-genetree) mode

Type: boolean flag

`--max-random-sample-proportion`

Description: The proportion of possible replicates explored unsuccessfully by the random generation procedure before it gives up. Because this generates random replicates, it takes progressively longer as it proceeds. To avoid long runtimes, the recommended range is < 0.5 (which is the default).

Type: float; **Default:** None

`--min-overlap`

Description: The minimum sites required to be sampled for all taxa in a given quartet.

Type: integer; **Default:** None

`--partitions`

Description: Partitions file in RAxML format. If omitted then the entire alignment will be treated as one partition for all quartet replicate topology searches.

Type: file path; **Default:** None

`--result-prefix`

Description: A prefix to put on the result files.

Type: string; **Default:** None

`--results-dir`

Description: A directory to which output files will be saved. If not supplied, the current working directory will be used. (default is current folder).

Type: file path; **Default:** None

`--retain-temp`

Description: Do not remove temporary files

Type: boolean flag

`--start-node-number`

Description: An integer denoting the node to which to start from. Nodes will be read from topologically identical (and isomorphic!) input trees in deterministic order, so this argument may be used to restart at an intermediate position (in case the previous run was canceled before completion, for example).

Type: integer; **Default:** None

`--stop-node-number`

Description: An integer denoting the node at which to stop. Will include nodes with indices <= the stop node number. This argument may be used to limit the length of a given run in case only a certain part of the tree is of interest. Nodes will be read from topologically identical (and isomorphic!) input trees in deterministic order.

Type: integer; **Default:** None

`--temp-dir`

Description: A directory to which temporary files will be saved. If not supplied, 'QuartetSampling' will be created in the current working directory. When specifying a custom temporary output the characters 'QuartetSampling' must appear in the directory name to prevent accidental file deletion. (default='./QuartetSampling'

Type: file path; **Default:** None

`--verbose`

Description: Provide more verbose output if specified.

Type: boolean flag

`--verbout`

Description: Provide output of the frequencies of each topology and QC.

Type: boolean flag

2.2 merge_output

2.2.1 Description

Combines RESULT.node.scores.csv files from separate runs for the same phylogeny into a single set of csv and tree outputs.

http://www.github.com/FePhyFoFum/quartetsampling

2.2.2 Parameters

`-h/--help`

Description: show this help message and exit

Type: boolean flag

`--nodedata` (required)

Description: file containing paths of one or moreRESULT.node.score.csv files

Type: None; **Default:** None

`--out` (required)

Description: new output files prefix

Type: None; **Default:** None

`--tree` (required)

Description: tree file in Newick format

Type: file path; **Default:** None

`--clade`

Description: ==SUPPRESS==

Type: None; **Default:** None

`--startk`

Description: ==SUPPRESS==

Type: integer; **Default:** 0

`--stopk`

Description: ==SUPPRESS==

Type: integer; **Default:** None

`--verbose`

Description: None

Type: boolean flag

2.3 query_tree

2.3.1 Description

Tree query script to find specific nodes numbers in large trees when using the post-run annotated trees.

http://www.github.com/FePhyFoFum/quartetsampling

2.3.2 Parameters

`-h/--help`

Description: show this help message and exit

Type: boolean flag

`--clade`

Description: ==SUPPRESS==

Type: None; **Default:** None

`--data`

Description: CSV output from quartet_sampling (RESULT.node.score.csv)

Type: file path; **Default:** None

`--startk`

Description: ==SUPPRESS==

Type: integer; **Default:** 0

`--stopk`

Description: ==SUPPRESS==

Type: integer; **Default:** None

`--tree`

Description: input tree in newick format

Type: file path; **Default:** None

`--verbose`

Description: verbose screen output

Type: boolean flag

2.4 calc_qstats

2.4.1 Description

Calculate basic statistics on the RESULTS.node.score.csv output file from quartet_sampling

2.4.2 Parameters

`-h/--help`

Description: show this help message and exit

Type: boolean flag

`--data` (required)

Description: RESULT.node.score.csv file output fromquartet_sampling.py

Type: file path; **Default:** None

`--clade`

Description: specify a clade using a comma-separatedlist of 2+ descendant taxa

Type: None; **Default:** None

`--out`

Description: output file path for statistics

Type: file path; **Default:** None

`--startk`

Description: starting branch numerical index

Type: integer; **Default:** 0

`--stopk`

Description: stopping branch numerical index

Type: integer; **Default:** None

`--verbose`

Description: verbose screen output

Type: boolean flag

CHAPTER

THREE

OUTPUTS

3.1 RESULT.node.scores.csv

A comma-separated values (CSV) file with * **node_label** The label of the internal branch (QS###) or terminal branch (original label) * **freq0** The number of concordant replicates over the non-uncertain total * **qc** The Quartet Concordance score (internal branches only; measures frequency of concordant over discordant) * **qd** The Quartet Differential score (internal branches only; measures skew in the two discordant tree counts) * **qi** The Quartet Informativeness score (internal branches only; measures number of replicates that fail likelihood cutoff) * **qf** The Quartet Fidelity score (terminal branches only; measures the number of replicates for which this branch produced concordant quartets) * **diff** Example likelihood differentinal from last replicate (used for diagnostic purposes) * **num_replicates** Number of replicates actually sampled per branch (may not equal the number specified by `-N` for internal branches when fewer replicates are possible, and will not equal that number for terminal branches). * **notes** Additional notes (enables user to add custom notes to their output after running; e.g., labeling key branches)

3.2 RESULT.labeled.tre

A Newick tree with each internal branch labeled with their QS## identifier.

3.3 RESULT.labeled.tre.freq/qc/qd/qu

A Newick tree with the each internal branch labeled with frequency of concordant replicates or QC/QD/QI scores.

3.4 RESULT.labeled.tre.figtree

A FigTree format phylogeny <http://tree.bio.ed.ac.uk/software/figtree/> that contains all QS scores and a "score" field with QC/QD/QI for internal branches.

CHAPTER

FOUR

FREQUENTLY ASKED QUESTIONS

4.1 What is an appropriate `--lnlike` value to use?

The parameter `--lnlike` or `--lnlike-thresh` specifies a log-likelihood differential cutoff. When the three possible topologies for a given quartet are evaluated using molecular data for that quartet by RAxML, three separate likelihood values are generated, which are then log-transformed. When the cutoff value is specified this means that you want to ensure that the tree with the best likelihood exceeds the next-most likely tree's likelihood by at least the value of the cutoff. This prevents the method from selecting from two or three trees with nearly indistinguishable likelihoods. These quartet replicates that pass the likelihood cutoff are tabulated separated and calculated as the QI score.

On a practical level, increasing the `--lnlike` parameter will decrease QI and increase the number of replicates counted as part of QC and QD. Lowering the `--lnlike` parameter will increase the QI score, but increase the number of potentially arbitrary counts in QC and QD.

More statistically, the log-likelihood threshold is equivalent to specifying a minimum likelihood ratio (subtracting log-likelihoods is equivalent to dividing raw likelihoods). Let the null hypothesis be zero difference between the tree likelihoods (i.e., any difference is random noise). For each of the two likelihood estimates compared, if error is distributed in a standard normal distribution N(0,1), then 95% of the distribution lies within ~2 standard deviations. This means the difference in the two log-likelihoods is chi-square distributed with the critical value for 1 degree of freedom at 3.8414 for a 95% confidence cutoff. Therefore, log-likelihood values that differ by 3.84 divided by 2 (for a two-tailed test) equals 3.84/2 = 1.92, which is approximately equal to 2. Therefore, a `--lnlike` value of 2 conservatively evaluates that the two likelihoods differ significantly at the alpha =0.05 level.

4.2 My partitioned data is not working properly, and is a sparse partitioned set.

RAxML and RAxML-ng both require that none of data partitions being evaluated are empty. This creates a unique challenge for quartet samplingon large-sparse matrices with partitioned data, since with sparse data and partitions there might be many partitions of a given quartet that contain insufficient data. We are working on a solution, but for now recommend using gene-tree mode or unpartitioned on sparse data.

CHAPTER

FIVE

RELEASES

5.1 v1.3 - Change to RAxML-ng, modernization of flags

- **The default is now RAxML-ng**, though classic RAxML 8.1+ is still available. Performance of QS in RAxML-ng on test datasets gives similar results, but is much more efficient.
- **Discontinuation of single-letter flags.** Single-letter flags are no longer functional, but have been retained in the comments of the main *quartet_sampling.py* script to allow forward updating. This change brings QS into line with evolving commnuity best practices that find better clarity and reproducibility when word-based flags are used exclusively.
- **–engine syntax change:** As we expand support for new tree inference and likelihood evaluation engines, this flag is a more flexible alternative to the previous named ones. See also notes on *–engine-exec* and *–engine-model*.
- Support for IQ-TREE is also available FOR TESTING PURPOSES ONLY! We have not fully evaluated QS under the IQ-TREE engine, so use this with caution and be sure to document this in your methods.

5.2 v1.2.1 - Efficiency update

- RAxML now evaluates all three tree configurations in the same program call. This reduces RAxML calls and increases efficiency.
- Clarification in the manual that RAxML 8.1+ is required for the program to execute correctly.

5.3 v1.2 - Major Update

- In order to make all four QS components have a “perfect” score of 1, the Quartet Uncertainty (QU) and Quartet Differential (QD) scores have been inverted. QU is now known Quartet as Informativeness (QI) is the inverse measure (where QI = 1 - QU). QD is also now inverted in scale, so that 1.0 means no differential in the discordant trees, and 0 means all one discdorant option.

- This version is consistent with the release of the updated bioRxiv manuscript.

5.4 v1.1.1

- Changed from -A/–amino-acids to -d/–data-type with ‘nuc’, ‘amino’, and ‘cat’ for binary/categorical data
- Amino acid mode should now work in PAUP as well.
- Fixed some issues with temp-dir and results-dir that were causing paths to revert to the root
- **the -e/–temp-dir parameter folder name now MUST contain ‘QuartetSampling’ to prevent file deletion**

5.5 v1.1

Contains small fixes to enable efficient documentation and minor changes to argument documentation.

5.6 v1.0

First public release, concurrent with bioRxiv publication.