
quartetsampling Documentation

Release 1.1

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

Jun 13, 2017

CONTENTS:

1	Getting Started	1
1.1	What is Quartet Sampling?	1
1.2	How do I cite this program?	1
1.3	Installation	1
1.4	Preparing your data	2
1.5	Basic usage	3
2	Program Parameter Descriptions	5
2.1	quartet_sampling	5
2.2	merge_output	9
2.3	query_tree	11
2.4	calc_qstats	12
3	Outputs	15
3.1	RESULT.node.scores.csv	15
3.2	RESULT.labeled.tre	15
3.3	RESULT.labeled.tre.freq/qc/qd/qu	15
3.4	RESULT.labeled.tre.figtree	16
4	Frequently Asked Questions	17
5	Releases	19
6	Indices and tables	21

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. A RAxML-formatted partition file may (optionally) be also used. 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 at <http://biorxiv.org/content/early/2017/06/10/148536.1>.

1.2 How do I cite this program?

Quartet Sampling distinguishes lack of support from conflicting support in the plant tree of life
James B Pease, Joseph W Brown, Joseph F Walker, Cody E Hinchliff, Stephen A Smith bioRxiv
148536; doi: <https://doi.org/10.1101/148536>

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

1.3 Installation

No installation is required, quartetsampling scripts should work as long as Python and RAxML are installed.

1.3.1 Requirements

- Python 3.x (2.7 should also work, but 3.x recommended) <https://www.python.org/downloads/>
- RAxML 8.x (7.x should also work, but 8.x recommended) <https://sco.h-its.org/exelixis/web/software/raxml/index.html>

Optional, but recommended:

- PAUP http://people.sc.fsu.edu/~dswofford/paup_test/ (required only if PAUP is used instead of RAxML)
- Numpy <http://www.numpy.org> (required only for `calc_qs_stats.py`)
- Scipy <https://www.scipy.org> (required only if `--calc-qdstats.py` is activated)
- Figtree <http://tree.bio.ed.ac.uk/software/figtree/> (required to 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 `-A/--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

```
:: python quartet_sampling.py -t TREE.nwk -a ALIGNMENT.phy -N 100 -T 4 -L 2
```

1.5.1 Required Paramters

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

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

`-N/--number-of-reps` Number of replicates per branch to perform.

`-T/--number-of-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

`-L/--lnlike-thresh`: 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 `-L/--lnlike-thresh` 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 Uncertainty (QU) scores of 'NA' for all branches.

PROGRAM PARAMETER DESCRIPTIONS

2.1 quartet_sampling

2.1.1 Description

quartet_sampling.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

-a/--alignment (required)

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

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

-t/--tree (required)

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

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

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

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

Type: integer; **Default:** 100

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

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

Type: integer; **Default:** 1

`--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

`-e/--temp-dir`

Description: A directory to which temporary files will be saved. If not supplied, “temp” will be created in the current working directory.

Type: file path; **Default:** ./temp

`-g/--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

`--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

`-o/--results-dir`

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

Type: file path; **Default:** .

`-p/--stop-node-number`

Description: An integer denoting the node at which to stop. Will include nodes with indices \leq 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

`--paup-executable`

Description: The name or path of the PAUP executable to be used for calculated quartets. (default='paup')

Type: None; **Default:** None

`-q/--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

`-r/--result-prefix`

Description: A prefix to put on the result files.

Type: string; **Default:** None

`--retain-temp`

Description: Do not remove temporary files

Type: boolean flag

`-s/--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

`-v/--verbose`

Description: Provide more verbose output if specified.

Type: boolean flag

`-A/--amino-acid`

Description: use amino acids instead of nucleotides

Type: boolean flag

`-C/--clade`

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

Type: string; **Default:** None

`-L/--lnlike-thresh`

Description: The lnlike threshold 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 (QU values are N/A in this case).

Type: float; **Default:** 2.0

-O/--min-overlap

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

Type: integer; **Default:** None

-P/--paup

Description: Use PAUP instead of RAxML.

Type: boolean flag

-V/--verbout

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

Type: boolean flag

-X/--raxml-executable

Description: The name (or absolute path) of the raxml executable to be used for calculating likelihoods on quartet topologies.(default='raxml')

Type: None; **Default:** None

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

-d/--nodedata (required)

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

Type: None; **Default:** None

-o/--out (required)

Description: new output files prefix

Type: None; **Default:** None

-t/--tree (required)

Description: tree file in Newick format

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

-c/--clade

Description: ==SUPPRESS==

Type: None; **Default:** None

-p/--stopk

Description: ==SUPPRESS==

Type: integer; **Default:** None

-s/--startk

Description: ==SUPPRESS==

Type: integer; **Default:** 0

-v/--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

-c/--clade

Description: ==SUPPRESS==

Type: None; **Default:** None

-d/--data

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

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

-p/--stopk

Description: ==SUPPRESS==

Type: integer; **Default:** None

-s/--startk

Description: ==SUPPRESS==

Type: integer; **Default:** 0

-t/--tree

Description: input tree in newick format

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

-v/--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

-d/--data (required)

Description: RESULT.node.score.csv file output from quartet_sampling.py

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

-c/--clade

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

Type: None; **Default:** None

-o/--out

Description: output file path for statistics

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

-p/--stopk

Description: stopping branch numerical index

Type: integer; **Default:** None

-s/--startk

Description: starting branch numerical index

Type: integer; **Default:** 0

-v/--verbose

Description: verbose screen output

Type: boolean flag

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) * **qu** The Quartet Uncertainty 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 differential 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/QU 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/QU for internal branches.

FREQUENTLY ASKED QUESTIONS

Coming soon

RELEASES

v1.0 == First public release, concurrent with bioRxiv publication.

v1.1 == Contains small fixes to enable efficient documentation and minor changes to argument documentation.

INDICES AND TABLES

- genindex
- modindex
- search