MAPLE documentation

Nicola De Maio, 1st June 2025

MAPLE performs maximum likelihood phylogenetic estimation from large datasets with short divergence. See De Maio et al. 2023 Nature Genetics for a description of the basic approach, and see De Maio et al 2024a bioRxiv for a description of new features. Specifically for a description of the SPRTA method to assess and represent tree uncertainty, see De Maio et al. 2024b bioRxiv. An alternative to MAPLE is CMAPLE which implements some of the methods of MAPLE in C++ and can be run within IQ-TREE, see Ly-Trong et al. 2024 MBE.

1 Installation

You can download the latest version of MAPLE python script from https://github.com/NicolaDM/MAPLE ("MAPLEv*.py").
No installation of the .py script is needed, however you will need python3 to be installed on your machine (see https://www.python.org/downloads/) to run the script.
Even better, running the script using pypy3 (see https://pypy.org/ , assumed to have been installed below) leads to around 10x runtime improvements on larger datasets, and so is highly recommended.

2 Data formatting

Like most other phylogenetic inference tools, MAPLE needs a multiple sequence alignment (MSA) in input. To obtain an MSA from unaligned sequences, you can use one of many aligners, such as MAFTT, Clustal, PRANK, etc. For large collections of similar genomes, MAFFT is a good option, for exampleusing these settings.
We recommend using MAPLE only with closely related genomes. When analysing non-closely related genomes (e.g. branch lengths >0.01) the software will be both slower and less accurate.

While most of these aligners will output a fasta format MSA, given our focus on large datasets of similar genomes, to improve memory and time demand we have devised our own alignment format, which we call "MAPLE format" (however other names for this format are also used in the literature).

This format represents sequences in terms of differences with respect to a reference.
We typically ignore inserted material with respect to the reference.

However, these insertion can in principle still be included by changing the definition of the reference sequence to include inserted material.

Our MSA format can help readability of the sequence information and can save a lot of space.
Here is an example:

>reference
ACGTACGTACGTACGT
>sample1
n 2 4
t 11
k 12
>sample2
a 11

The first 2 lines of the file are the reference name and the reference genome sequence (however MAPLE can also read the fasta file of the reference from a separate file when given the "--reference" option). In this example the reference genome is just 16 nucleotides long.
The third line is the name of the first genome in the MSA (here character ">" indicates that the line contains a name, just like in a fasta file).
Each line until the next genome name represents a difference with respect to the reference. Each of these lines contains 2 or 3 values separated by a tab "\t" character.
- The first value is the nucleotide character (or IUPAC ambiguity code) in the considered genome that differs from the reference nucleotide at the same position.
- The second value is the position (the first genome position is "1") of the genome at which this difference occurs. Entries need to be sorted from smaller to larger and cannot overlap.
- The third value, if present (it should only be used for "n" characters), represents how many consecutive positions contain this character in the considered genome (if the third value is not present, it is assumed to be equal to 1). For example in this case "*n 2 4*" means that character N is present in the considered genome from position 2 to position 5 included.

In this example, the genome sequence of sample1 is:
ANNNNCGTACTKACGT

Some example maple format MSAs are available from
https://github.com/NicolaDM/MAPLE/tree/main/example_files .

2.1 Translating a fasta file to maple format

We include a python script to translate a fasta MSA into a maple MSA.
This is the file "createMapleFile.py" from the MAPLE repository
https://github.com/NicolaDM/MAPLE.

This script can be run for example with command:

python3 createMapleFile.py --path /pathToFolder/ --fasta alignment.fa --output alignment.maple

Which, given a fasta MSA “alignment.fa” in the folder “/pathToFolder/” will create the same MSA in maple format in the same folder under the name “alignment.maple”.

Here we are not specifying a reference genome. In this case the script will define the consensus of the input MSA as the reference. In case a specific reference needs to be used, it can be specified with option “--reference”.

2.2 Masking columns of a MSA

Recurrent sequence errors can be prevalent and deleterious in genomic epidemiological data (see [1,2,3,4,5]).
MAPLE offers a model to infer and partially account for recurrent sequence errors (see “Models” section). The best approach is however to remove/mask columns of the MSA that are affected by such recurrent errors.

To help in this task, we provide two python scripts we used to mask and otherwise process a maple alignment.

3 Basic usage

The main MAPLE script ("MAPLEv...py") takes in input a maple format MSA file in MAPLE (containing a reference genome sequence, but if not the reference file can be specified with the “--reference” option):

pypy3 MAPLEv0.7.4.py --input alignment.maple --output outputFilePrefix

The “--output” option is used by MAPLE to name the output files: in this case the final output tree will be named "outputFilePrefix_tree.tree", and the file containing the estimated model parameters will be "outputFilePrefix_subs.txt". Option “--overwrite” can be used to overwrite existing files with those names.

With these options, MAPLE will
1 Create an initial tree by adding one sample at the time.
2 Infer model parameters
3 Improve the initial tree using subtree prune and regraft (“SPR”) moves.

Inference of model parameters is not only performed between steps 1 and 3, but often also during steps 1 and 3 themselves.

3.1 Online tree inference (adding sequences to an existing tree) and starting trees

Given a tree previously estimated, and given an alignment containing the sequences of the samples in the tree, plus possibly some additional sequences, one can use MAPLE to add these additional samples to the given tree, and/or to improve the topology of the input tree. This can be done using option “--inputTree”:

pypy3 MAPLEv0.7.4.py --inputTree inputTreeFile.tree --input alignment.txt --output outputFile

If the samples in the alignment are the same as in those in the tree, MAPLE will skip the step of initial tree creation.
If instead the input alignment contains sequence names not present in the input tree (the sequences in the input tree MUST however be always present in the alignment), MAPLE will add one by one these additional samples to the initial tree.

When given an initial tree, by default MAPLE will perform a small-scale fast SPR search near the nodes added to the tree when placing new samples - a much faster option that assumes the initial tree is of good quality.
One can instead use option “--largeUpdate” to perform in this scenario the standard large-scale intensive SPR search the MAPLE would perform without an initial tree.

3.2 parallelization

The most time-demanding part of MAPLE is the SPR search. For this reason, MAPLE now allows parallelization of the SPR search using option “--numCores”. For example:

pypy3 MAPLEv0.7.4.py --input alignment.txt --output outputFile --numCores 10

will parallelize the SPR search over 10 cores. This parallelization requires linearly more memory and initialization runtime with each additional core, so it is not recommended to try to parallelize over an excessive number of cores (e.g. >20) since this can deteriorate the method's performance. No matter the number of cores used, the initial tree inference via stepwise addition can still only be run sequentially on 1 core.

3.2 Inferring mutation events

MAPLE can be used to infer mutation events on the output tree (that is, estimate a probabilistic mutation-annotated tree) using option “--estimateMAT”.
Inferred mutations are annotated with their posterior probability, so that the same mutation

can be annotated on multiple branches in case of non-negligible uncertainty in its inference.

4 Models

Three basic nucleotide substitution models are implemented in MAPLE: JC69, GTR, and UNREST. The model can be chosen with option “--model”; the default model is GTR, but we recommend UNREST for sufficiently large datasets (option “--model UNREST”) .

4.1 Rate variation

MAPLE also offers a model of rate variation, see De Maio et al 2024a. This model assigns a free rate parameter to each genome position; since this model is parameter-rich, its use is recommended in particular for larger datasets. Use of this rate variation model is triggered in MAPLE by the option “--rateVariation”. For large datasets we recommend the use of this model jointly with UNREST (option “--model UNREST --rateVariation”).

4.2 Recurrent sequence errors model

MAPLE also implements a model of heterogeneous recurrent sequence errors, which can be used to estimate recurrent sequence errors, see De Maio et al 2024a.
This model assigns a free error rate parameter to each genome position; its use is recommended only for larger datasets and in conjunction with the rate variation model and the UNREST substitution model, so to prevent misinterpreting mutation rate variation for sequence error variation.
To use this error model in MAPLE, in conjunction with mutation rate variation and UNREST, you can specify option “--model UNREST --rateVariation --estimateSiteSpecificErrorRate”.
When using this recurrent sequence error model in MAPLE, it is possible to estimate individual sequence errors in the input alignment with option --estimateErrors . An output file will then contain estimated sequence errors, each with its posterior probability of being an error.

Note however that only part of the highly recurrent errors might be identified by MAPLE if the error rates are too high or strongly correlated with other recurrent sequence errors. In these cases we recommend first to identify highly recurrent errors with MAPLE with this model, then mask these columns from the alignment (see Section 2.2 above), then re-run MAPLE with the error model again, and repeat.
It is also important to start inference from a high-quality alignment, and in particular we recommend using Viridian genomes for SARS-CoV-2 as they prevent calling many wrong reversions to the reference, see Hunt et al. 2024 bioRxiv.

In summary, to run the most advanced model in MAPLE, you can use options:

pypy3 MAPLEv0.7.4.py --input alignment.txt --output outputFile --model UNREST --rateVariation --estimateSiteSpecificErrorRate --estimateErrors

4.3 Accounting for genome abundance

MAPLE can now account for lineage abundance during phylogenetic inference.
For example, when placing a genome onto a phylogeny, it will favor placement onto an abundant lineage rather than of a rare one. We have two approaches in MAPLE to achieve this, collectively called HnZ (Horse not Zebra, manuscript in preparation).

The first approach (MAPLE option "--HnZ 1") multiplies a multifurcating tree likelihood by the number of bifurcating tree topologies consistent with the tree.

The second approach (option "--HnZ 2") multiplies the phylogenetic tree likelihood by a sampling tree prior, representing the probability that genomes were sampled given their abundance (inferred from the tree itself).

In our simulations, these HnZ methods substantially improve the accuracy of tree inference from simulated SARS-CoV-2 data.

An example command line is:

pypy3 MAPLEv0.7.4.py --input alignment.txt --output outputFile --model UNREST --rateVariation --HnZ 1

5 Phylogenetic uncertainty - SPRTA

MAPLE implements a new pandemic-scale approach called SPRTA (see De Maio et al. 2024b bioRxiv) to assess phylogenetic uncertainty.
For each branch in the tree (including terminal branches) SPRTA assesses if there are alternative origins in the tree for the subtree under that branch.
For the case of terminal branches, this is equivalent to assessing the placement probability of the genome under the branch.
SPRTA not only assigns a support probability to each branch in the tree, but it can also show alternative placements (with corresponding probabilities) of subtrees.
Note that SPRTA is very different from other phylogenetic branch support scores (such as Felsenstein's bootstrap) and its scores should not be interpreted in the same way.

SPRTA can be used in MAPLE by specifying option "--SPRTA".
Option "--supportFor0Branches" will make SPRTA also evaluate the support of 0-length branches (for example the placement of all genomes, even those within

multifurcations). Expect slightly longer runtime when using this option.
Option “--networkOutput” makes MAPLE record and output not only the scores of the branches in the tree, but also alternative subtree placements.

An example command line for running SPRTA in MAPLE is:

pypy3 MAPLEv0.7.4.py --input alignment.txt --output outputFile --model UNREST --rateVariation --SPRTA --networkOutput --supportFor0Branches

6 Output processing

Output trees from MAPLE can be very large and therefore hard to visualize/explore. For this reason we recommend the use of Taxonium (see Sanderson 2022 eLife), and in particular its Desktop app to visualise these trees.
To convert a newick tree (as in output from MAPLE) into an input tree for Taxonium, you can use the newick_to_taxonium script.

When using SPRTA in MAPLE, and in particular when using the “--networkOutput” option, MAPLE will create an output .tsv file containing information regarding phylogenetic uncertainty.
The "support" column of this file will contain, at a row corresponding to branch A, the support score of A.
Column "supportTo" will contain the list of branches that could plausibly be alternatively placed at A.
To graphically visualise these alternative placements, first we can create a json file to be used in Taxonium combining the MAPLE output tree with the SPRTA output metadata:

newick_to_taxonium -i MAPLE_tree.tree -m MAPLE_metaData.tsv -o Taxonium_inputFile.jsonl -c support,supportGroup,supportTo

The resulting .jsonl file can then be opened in Taxonium, and alternative plausible placements of a given branch/node/subtree can then be jointly visualised by searching for the branch name in the "supportTo" search field in Taxonium.

The MAPLE output .tsv file can also contain information regarding inferred mutation events on the tree (when using option “--estimateMAT”) which will appear in column “mutationsInf” of the .tsv output file; and of possible alternative rooting of the tree: the support probability of rooting at a branch will appear in column “rootSupport” of the .tsv file.
These additional features can also be included in the jsonl file for Taxonium by adding the corresponding column names to the “-c” option of the newick_to_taxonium command line. This allows for example the graphical search of recurrent mutations in MAPLE trees.

7 Other features/options of MAPLE

7.1 Lineage definition and assignment

MAPLE implements an optional strategy to assign lineages to samples and branches in a given tree. This is done using reference genomes - one reference representing one lineage. These references are placed onto the tree, and then each sample/branch is assigned the lineage corresponding to the closest related ancestral reference (branches of length 0 are ignored, so that references can be ancestral to other samples in the tree).
This is achieved with MAPLE options:

--lineageRefs (default="") path and name to an alignment file (in MAPLE format) containing reference genomes, each representing one lineage. When using this option, option --inputTree should also be used. Then MAPLE will find the best placement for each lineage reference (note that these lineage references do not exist in the input tree). Each sample is assigned a lineage same as its closest reference ancestor.

--lineageRefsThresh (default=0.2) The threshold (in term of #mutation) to check whether a reference lineage genome could be considered as the parent of a subtree. Default: 0.2 mutation.

--lineageRefsSupportThresh (default=0.95) A lineage will be assigned to a subtree only if the SPRTA support for that lineage placement exceeds this threshold.

--allowMultiLineagesPerNode (default=False) When a node is selected as the best placements for multiple lineages, whether we allow assigning all of these lineages (instead of only the closest lineage) to the subtree.

MAPLE can also be used for assigning lineages to samples not in the tree. This can be done by first assigning lineages to branches in the tree (as explained above) and then mapping new sample onto the tree, with the following option:

--findSamplePlacements (default=False) Find placements (in an input tree) for new samples (without changing the input tree - find placements only).

7.2 Robinson-Foulds distance calculation

MAPLE can be used to perform fast Robinson-Foulds distance calculation (using the algorithm from Day 1985) instead of performing tree inference. This implementation of the RF distance allows multifurcating trees: branches of length 0 are considered absent from the tree.
An example command line is:

pypy3 MAPLEv0.7.4.py --inputTree inputTree.tree --inputRFtrees otherInputTrees.tree

With this command, MAPLE compares the newick tree in the file inputTree.tree to all the newick trees in the otherInputTrees.tree file. Having multiple trees in this second file is faster than running the script many times with 2 trees at the time, but you will need to specify option “--multipleInputRFTrees” to prevent MAPLE from only reading the first tree in the otherInputTrees.tree file.

7.3 Other running options

--reference (default="") Optional input reference file name. By default it assumes instead that the reference is part of the MAPLE format input.

--fast (default=False) Set parameters to run faster, less accurate tree inference. It will overrule user choices for options --thresholdLogLK , --thresholdLogLKtopology , --allowedFails , --allowedFailsTopology .

--defaultBLen (default=0.000033) Default length of branches, for example when the input tree has no branch length information.

--normalizeInputBLen (default=1.0) For the case the input tree has branch lengths expressed not in a likelihood fashion (that is, expected number of substitutions per site), then multiply the input branch lengths by this factor. Particularly useful when using parsimony-based input trees.

--onlyNambiguities (default=False) Treat all ambiguities as N (total missing information).

7.4 Other output options

--overwrite (default=False) Overwrite previous results if already present.

--saveInitialTreeEvery (default=50000) Every these many samples placed (default 50,000) save the current tree to file. This way if there is any problem, the building of the initial tree can be restarted from the last saved tree.

--nonBinaryTree (default=False) Write output tree with multifurcations - by default the tree is written as binary so to avoid problems reading the tree in other software.

--writeTreesToFileEveryTheseSteps (default=0) By default, don't write intermediate trees to file. If however a positive integer is specified with this option, intermediate trees will be written to file every this many topological changes.", type=int, default=0)

--writeLKsToFileEveryTheseSteps (default=0) By default, don't write likelihoods of intermediate trees to file. If however a positive integer is specified with this option, likelihoods of intermediate trees will be written to file every this many topological changes.

--noSubroundTrees (default=False) Do not write to file subround trees (intermediate trees generated during rounds of SPR search).

7.5 Options to switch off MAPLE features

--doNotImproveTopology (default=False) Do not perform SPR moves, despite searching for them; this is useful if one wants to analyse a tree and calculate branch supports without changing the input tree.

--doNotPlaceNewSamples (default=False) Given an input tree, skip the placement of samples on the tree (in case the input alignment contains more samples than the tree), so keep only the samples originally already on the tree.

--doNotReroot (default=False) Skip rooting optimization.

--noLocalRef (default=False) Do not use local references (this will usually take longer to run).

--noFastTopologyInitialSearch (default=False) Don't run a first fast short-range topology search (before the extensive one in case the latter is performed).

--doNotOptimiseBLengths (default=False) Do not optimise the branch lengths of the tree (useful if the input tree is already optimal or doesn't need changing).

7.6 Other model options

--inputRates (default="") Name of input containing pre-estimated substitution rates and possibly other model parameters; this is optional, and is only used for online inference. The same format as the MAPLE output is expected, and information about all parameters of the selected substitution model is expected.

--estimateErrorRate (default=False) Estimate a single error rate for the whole genome. Input value is used as starting value.

--estimateSiteSpecificErrorRate (default=False) Estimate a separate error rate for each genome genome. Input value is used as starting value.

--errorRateInitial (default=0.0) Initial value used for estimating the error rate. The default is the inverse of the reference genome length (one error expected per genome).

--errorRateFixed (default=0.0) Fix the error rate to a given input value.

--errorRateSiteSpecificFile (default=None) File containing site-specific error rates to be used by MAPLE.

--estimateErrors (default=False) Estimate erroneous positions in the input sequences. This option is only allowed if the option --estimateSiteSpecificErrorRate or --errorRateSiteSpecificFile is also used.

7.7 Options to set MAPLE thresholds

--minNumNon4 (default=1) Minimum number of mutations threshold to define a subreference in the MAT - smaller values will cause denser references in the MAT.

--maxNumDescendantsForMATClade (default=50) Number of positive-branch-length descendants allowed before triggering mutation list (i.e. local reference) creation in the tree.

--thresholdProb (default=0.00000001) relative probability threshold used to ignore possible states with very low probabilities.

--thresholdLogLK (default=18.0) logLK difference threshold (in number of mutations) to consider a logLk close to optimal.

--thresholdLogLKtopology (default=14.0) Maximum logLK difference threshold (in number of mutations) to consider a logLk close to optimal when looking for topology improvements.

--allowedFails (default=5) Number of times one can go down the tree without increasing placement likelihood before the tree traversal is stopped (only applies to non-0 branch lengths).

--allowedFailsTopology (default=4) Maximum number of times one can crawl along the tree without increasing placement likelihood before the tree traversal is stopped during topology search (only applies to non-0 branch lengths).

--numTopologyImprovements (default=1) Number of times we traverse the tree looking for deep (and slow) topological improvements. Default is 1, select 0 to skip deep topological search. values >1 are not recommended.

--thresholdTopologyPlacement (default=-0.1) Don't try to re-place nodes that have current appending logLK cost above this threshold.

--updateSubstMatrixEveryThisSamples (default=25) How many new samples to place before each update of the substitution rate matrix.

--nonStrictStopRules (default=False) If specified, then during the initial placement stage, a slower, non-strict rule for stopping the placement search is applied: the search is

stopped if enough many consencutive LK worsening are observed, AND if LK is below the considered threshold.

--strictTopologyStopRules (default=False) If specified, then during the topological improvement stage, a faster, strict rule for stopping the SPR search is applied: the search is stopped if enough many consencutive LK worsening are observed, OR if LK is below the considered threshold.

--thresholdDiffForUpdate (default=0.00001) Consider the probability of a new partial changed if the difference between old and new is above this threshold.

--thresholdFoldChangeUpdate (default=1.01) Consider the probability of a new partial changed, if the fold difference between old and new if above this threshold.

--thresholdLogLKconsecutivePlacement (default=1.0) logLK difference threshold to consider something as a significant decrease in log-LK when considering consecutive likelihood decreases.

--thresholdLogLKTopologySubRoundImprovement (default=3.0) logLK difference threshold to consider something as a significant decrease in log-LK when considering sub-rounds of SPR moves.

--minBLenSensitivity (default=0.001) Fraction of a mutation to be considered as a precision for branch length estimation (default 0.001, which means branch lengths estimated up to a 1000th of a mutation precision).

--thresholdLogLKoptimization (default=1.0) logLK difference threshold (in number of mutations) to consider a logLk close to optimal when deciding for which possible placements to perform branch length optimization.

--thresholdLogLKoptimizationTopology (default=1.0) LogLK difference threshold (in number of mutations) to consider a logLk close to optimal when deciding for which possible placements to perform branch length optimization during the SPR stage.

--maxReplacements (default=10) Maximum number of replacements attempts per node per SPR round (prevents loops).

--useFixedThresholdLogLKoptimizationTopology (default=False) Use this option if you want to specify the value in --thresholdLogLKoptimizationTopology instead of estimating it from the data.

--minNumSamplesForRateVar (default=510000) When creating the initial tree, start using the rate variation model only after these many samples have been added (better not to decrease this too much to avoid overfitting).

--minNumSamplesForErrorModel (default=510000) When creating the initial tree, start using the error model only after these many samples have been added (better not to decrease this too much to avoid overfitting).

--minErrorProb (default=0.01) Minimum error probability to be reported when using option --estimateErrors .

--minBranchSupport (default=0.01) Minimum branch support to be considered when using option --networkOutput .

--minMutProb (default=0.01) Minimum mutation probability to be written to output when using option --estimateMAT.