TreeTime User Manual

Lin Himmelmann
www.linhi.de

Dirk Metzler
www.zi.biologie.uni-muenchen.de/evol/StatGen.html

March 2009

TreeTime is controlled via an input–file in Nexus–file format (view Maddison 1997). The Nexus–file contains all data, operations and parameters to set up the analysis. If you run TreeTime as a stand alone application or in the console without parameter, TreeTime will search a Nexus–file in the current directory with the name input.nex. Optional you can assign a path as a parameter, which should point to an appropriate Nexus–file.

All commands and data are encapsulated by enclosing blocks. The molecular data must be given in the Data–block, which is described in Maddison 1997. All commands and common data specifications are fully supported by TreeTime. The TreeTime specific instuctions and parameter settings are given in the TreeTime-Block. The block begins with the command „`Begin TreeTime;`" and ends with the command „`End;`". All TreeTime commands are inside this block. A command in TreeTime begins with a command–keyword, which is followed by parameters. The parameters are assignd values with the equals sign. Every command is completed by a semicolon. Comments can be pasted in squared brackets, and will not be considered by TreeTime. In the following command reference optional commands and parameters are in squared brackets ([optional]). These commands are not obligatory, while the other commands must be given to run TreeTime. The following para-

graph contains a brief enumeration of commands and parameters which can be used in the TreeTime–block.

Command reference

`settings`

    `outputPath`=string [default=./]
    Declaration of the outpt directory.
    `seed`=positiv integer [default=currentTime, optional]
    Seed of the random generator.
    `rateChanges`={0,1} [default=0, optional]
    With/without changes of the rate change model: 0/1.
    `memChanges`={0,1} [default=1, optional]
    With/without changes of the molecular evolution model: 0/1.

`mcmcmc`

    `nChains`=positiv integer
    Number of chains.
    `nSamples`=positiv integer
    Number of sampled states.
    `burnInPeriod`=non-negativ integer
    Number of burn–in steps.
    `sampleFrequency`=positiv integer
    Number of steps between the sampled states.
    `heatingCoefficient`=positiv integer [default=0.3, optional]
    Heating-Parameter, when multiple chains are being used.

`parameter` [optional]

    `cNodeTime`=[0.0-4.0] [default=1.0, optional]
    Parameter to calibrate the acceptance probability of a change to modify the location of an inner node. The smaller the value, the more changes will be accepted.
    `cRootTime`=positiv integer [default=1.0, optional]
    Parameter to calibrate the acceptance probability of a change to modify the location of the root node. The smaller the value, the more changes will be accepted.
    `cTimeSynchronisation`=positiv integer [default=1.0, optional]

Parameter to calibrate the acceptance probability of a change to modify the synchronisation between real time and molecular time scale. The smaller the value, the more changes will be accepted.

weighting [optional]

topology=positiv number [default=0.2, optional]
Weighting-parameter to set the proportion of steps to be used for changes of the topology. This parameter is in relation to the other weighting-parameters.
nodes=positiv number [default=0.2, optional]
Weighting-parameter to set the proportion of steps to be used to change the timepoints at the inner nodes. This parameter is in relation to the other weighting-parameters.
evolution=positiv number [default=0.8, optional]
Weighting-parameter to set the proportion of steps to be used to change the parameters of the molecular evolution models. This parameter is in relation to the other weighting-parameters.
ratechange=positiv number [default=0.3, optional]
Weighting-parameter to set the proportion of steps to be used to change the parameters of the rate change models. This parameter is in relation to the other weighting-parameters.
timeconversion=positiv number [default=0.02, optional]
Weighting-parameter to set the proportion of steps to be used to change the time synchronisation between real time and molecular time. This parameter is in relation to the other weighting-parameters.

initialTree [optional]

tree=Newick–string [optional]
Declaration of an initial starting tree in newick format, which must be ultrametric. If no starting tree is given, TreeTime will estimate a tree via UPGMA from the molecular data.

taxa

outgroup=string [optional]
Declaration of an outgroup taxa.
active={string,...,string}
Declaration of taxa, which will be included in the tree reconstruction

method. In the current version it is mandatory to list all taxa, which are contained in the Data-block.

`topologicalConstraint` [optional]

> `list1`={string,...,string}
>
> List of comma-separated taxa, which must lie in the left subtree as a result of a certain split.
>
> `list2`={string,...,string}
>
> List of comma-separated taxa, which must lie in the right subtree as a result of a certain split.

`gammaConstraint` [optional]

> `list1`={string,...,string}
>
> `list2`={string,...,string}
>
> Two Lists of comma-separated taxa, which are separated in the tree according to a split.
>
> `min`=positiv number
>
> Minimum value for the time of the splitting event.
>
> `mean`=positiv number
>
> Mean time of the splitting event.
>
> `alpha`=[0.5,1.5]
>
> Parameter α to controle the shape of the gamma-distribution to get an appropriate prior distribution for the time of the split.

`gaussianConstraint` [optional]

> `list1`={string,...,string}
>
> `list2`={string,...,string}
>
> Two Lists of comma-separated taxa, which are separated in the tree according to a split.
>
> `mean`=positiv number
>
> Mean of the normal distribution.
>
> `sd`=positiv number
>
> Deviation of the normal distribution.

`dnaData`

> `nSites`=positiv integer
>
> Number of sites in the alignment.
>
> `nLoci`=positiv integer
>
> Number of loci (number of different molecular data partitions).

`beginningFragments`=(0,positiv integer,...,positiv integer)
Numbers of the beginnings of the loci in the alignment.
`models`=(MEM,...,MEM)
Comma-separated list of the molecular evolution models for each loci.
The definition of all implemented molecular evolution models (MEM)
is given below.

`rateChange`

`globalModel`=RCM
Declaration of a global rate change model (RCM). The definition of all
implemented rate change models is given below.
`localModel`=(RCM,...,RCM)
Comma-separated list of local rate change models for each loci. The
definition of all implemented rate change models (RCM) is given below.

Molecular evolution model (MEM): A molecular evolution model is given
in a certain string representation. Each model has a unique prefix, followed
by model parameters, which are encapsulated in round brackets. The prefix
is used to identify the correct model. The parameters in the string represen-
tation represent the model parameters to initialise and set up the the model.
Currently TreeTime has implemented the following molecular evolution mo-
dels:

1. Jukes–Cantor model
   The prefix to identify this model is "JC". The Jukes–Cantor model
   without invariant sites and rate heterogenity has no parameters. To use
   this model, you don't need to specify any parameters. The notation is
   "JC()".

2. General Timereversible model.
   The prefix to identify this model is "GTR". To use this model you
   have to specify the following parameters: $\text{GTR}(\pi_A, \pi_C, \pi_G, \pi_T, t_{A\to C},$
   $t_{A\to G}, t_{A\to T}, t_{C\to G}, t_{C\to T}, t_{G\to T})$, where $\pi_A, \pi_C, \pi_G, \pi_T$ is the initial sta-
   tionary distribution of the nucleotides, and $t_{A\to C}, t_{A\to G}, t_{A\to T}, t_{C\to G}, t_{C\to T},$
   $t_{G\to T}$ are parameters to control the initial transition rates between the
   nucleotides. The transition rates are automatically rescaled, such that
   the evolutionar rates are calibrated to 1 PEM.

All molecular evolution models can be extended by invariant sites. To do this, you have to add the string „+I" after the unique model prefix. You also have to append one additional parameter between the brackets in the string representation of the model, which will be used as the initial proportion of invariant sites. Additionally, all molecular evolution models can be extended by discretised gamma–distributed rate categories. For this you have to add the string „+Gx" after the unique model prefix. The variable x must be an integral value equal or greater than 2, which specifies the number of rate categories in the approximation of the gamma–distribution. If you use gamma–rate heterogenity among sites, you have to append an additional parameter between the brackets in the string representation of the model, which will be used as the initial parameter to control the shape of the gamma–distribution. If you want to extend a molecular evolution model with invariant sites and gamma–rate heterogenity, you have to append the parameter for invariant sites first and afterwards append the parameter to control the shape of the gamma–distribution.

Rate change model (RCM): A rate change model is given in a certain string representation. Each model has a unique prefix, followed by model parameters, which are encapsulated in round brackets. The prefix is used to identify the correct model. The parameters in the string representation represent the model parameters to initialise and set up the the model. Currently TreeTime has implemented the following rate change models:

1. Molecular clock model
   The prefix to identify this model is "MC". The molecular clock model has no parameters, so the notation is "MC()".

2. Dirichlet model
   The prefix to identify this model is "DM". To use this model you have to specify the following parameters: DM(priorExpectedVariance, initialVariance, nrOfSpecies, changeCalibrationVariance [optional], changeCalibrationAll [optional], changeCalibrationTwo [optional]). The interpretation of these parameters is:
   `priorExpectedVariance`=positiv number
   Expected variance for the heterogenity of rates specified by the prior distribution.
   `initialVariance`=positiv number
   Initial variance for the heterogenity of rates.

6

`nrOfSpecies`=positiv integer
Number of taxa in the tree.
`changeCalibrationVariance`=positiv number [default=1.0]
Parameter to calibrate the acceptance probability of a change which
modifies the variance of rates. The smaller the value, the more changes
will be accepted.
`changeCalibrationAll`=positiv number [default=1.0]
Parameter to calibrate the acceptance probability of a change which
modifies all rates simultaneously. The smaller the value, the more chan-
ges will be accepted.
`changeCalibrationTwo`=positiv number [default=1.0]
Parameter to calibrate the acceptance probability of a change which
modifies two rates. The smaller the value, the more changes will be
accepted.

3. Uncorrelated lognormal model
   The prefix to identify this model is "ULN". To use this model you have
   to specify the following parameters: ULN(priorExpectedVariance, initi-
   alVariance, nrOfSpecies, changeCalibrationVariance [optional], change-
   CalibrationAll [optional], changeCalibrationSingle [optional]). The in-
   terpretation of these parameters is:
   `priorExpectedVariance`=positiv number
   Expected variance for the heterogenity of rates specified by the prior
   distribution.
   `initialVariance`=positiv number
   Initial variance for the heterogenity of rates.
   `nrOfSpecies`=positiv integer
   Number of taxa in the tree.
   `changeCalibrationVariance`=positiv number [default=1.0]
   Parameter to calibrate the acceptance probability of a change which
   modifies the variance of rates. The smaller the value, the more changes
   will be accepted.
   `changeCalibrationAll`=positiv number [default=1.0]
   Parameter to calibrate the acceptance probability of a change which
   modifies all rates simultaneously. The smaller the value, the more chan-
   ges will be accepted.
   `changeCalibrationSingle`=positiv number [default=1.0]
   Parameter to calibrate the acceptance probability of a change which

modifies one rate. The smaller the value, the more changes will be accepted.

4. Uncorrelated exponential model
   The prefix to identify this model is "UEX". To use this model you have to specify the following parameters: UEX(nrOfSpecies, changeCalibrationAll [optional], changeCalibrationSingle [optional]). The interpretation of these parameters is:
   `nrOfSpecies`=positiv integer
   Number of taxa in the tree.
   `changeCalibrationAll`=positiv number [default=1.0]
   Parameter to calibrate the acceptance probability of a change which modifies all rates simultaneously. The smaller the value, the more changes will be accepted.
   `changeCalibrationSingle`=positiv number [default=1.0]
   Parameter to calibrate the acceptance probability of a change which modifies one rate. The smaller the value, the more changes will be accepted.

5. Compound poisson process
   The prefix to identify this model is "CPP". To use this model you have to specify the following parameters: CPP(priorExpectedRateChangeIntensity, initialRateChangeIntensity, initialDistributionParameterAlpha, nrOfSpecies, changeCalibrationIntensity [optional], changeCalibrationDistribution [optional]). The interpretation of these parameters is:
   `priorExpectedRateChangeIntensity`=positiv number
   Expected intensity of the poisson process, which controls the dispersion of points where the rate is being changed.
   `initialRateChangeIntensity`=positiv number
   Initial intensity of the poisson process.
   `initialDistributionParameterAlpha`=positiv number
   Initial distribution for the the magnitude of the rate changes.
   `nrOfSpecies`=positiv integer
   positiv integer: Number of taxa in the tree.
   `changeCalibrationIntensity`=positiv number [default=1.0]
   Parameter to calibrate the acceptance probability of a change which modifies the intesity of the poisson process. The smaller the value, the

more changes will be accepted.

`changeCalibrationDistribution`=positiv number [default=1.0]
Parameter to calibrate the acceptance probability of a change which modifies the magnitude of rate changes. The smaller the value, the more changes will be accepted.

Output

TreeTime generates for every tree reconstruction analysis seven output files. In these files are stored relevant information of the sampled states and used parameters. the files are:

1. tt_treesUM.tre
2. tt_treesUR.tre
3. tt_mems.txt
4. tt_rcms.txt
5. tt_changes.txt
6. tt_swaps.txt
7. tt_stateProbs.txt

In the following is a description of the content and format of the output files:

tt_treesUM.tre
>   The sampled phylogenies were stored in newick–format in this file. The sampled phylogenies are in real time units and thus are ultrametric.

tt_treesUR.tre
>   Here, the sampled phylogenies were stored in newick–format on a molecular timescale. At first, a sampled phylogeny is stored in units of the global molecular timescale. After that, the phylogeny is stored for every loci according to the corresponding local molecular timescale.

tt_mems.txt
>   The molecular evolution models were stored in this file. For every sampeld state, the molecular evolution model to every loci is printed, with the same ordering of the models as the ordering of the loci. For the Jukes–Cantor model the output is ”JC”, followed by the parameters:

Number of gamma–distributed rate categories, shape parameter of the gamma-distribution for the rate heterogenity, are there invariant sites, proportion of invariant sites, stationary distribution of the nucleotides A, C, G and T as well as the time scaling factor. For the generell timereversible model the output is "GTR", followed by the parameters: Number of gamma–distributed rate categories, shape parameter of the gamma-distribution for the rate heterogenity, are there invariant sites, proportion of invariant sites, stationary distribution of the nucleotides A, C, G and T, the transition rates $A \rightarrow C$, $A \rightarrow G$, $A \rightarrow T$, $C \rightarrow G$, $C \rightarrow T$ and $G \rightarrow T$ as well as the time scaling factor.

tt_rcms.txt

The rate change models were stored in this file. For every sampeld state, the global rate change model is printed. After that, the rate change model is printed for every loci, with the same ordering of the models as the loci. For the molecular clock model the output is "MC". For the compound poisson process the output is "CPP", followed by the expected intensity, the current intensity lambda, the distribution parameters alpha and beta controlling the rate modulation and the number of rate change points. For the uncorrelated lognormal model the output is "ULN", followed by the expected and current variance. For the uncorrelated exponential model the output is "UEX". For the Dirichlet model the output is "DM", followed by the expected and current variance.

tt_changes.txt

In this file are tracked all proposed changes in the MCMC–method and how often a proposed change was accepted or rejected. In the first column are the unique identifiers of the corresponding proposed changes. In the second column are the counts of the accepted steps and in the third column are the counts of the rejected steps. These counts are repeatedly emitted for all chains, whearas the output begins with the cold to the most heated chain.

tt_swaps.txt

If several chains were run in parallel, the relative frequencies of the acceptance of a change between the state of chain i and chain j are recorded for all i and j. The proportions are given in a matrix, which is sorted from the cold to the most heated chain.

tt_stateProbs.txt

Here are printed the probabilities and probability densities on a logarithmic scale. Recorded are the posterior probabilities (up to a constant factor), the likelihood of the tree, the probability density of the temporal calibrations and the probability density of the global rate change model (up to a constant factor). Thereafter are printed the probability densities of the local rate change models.