HExT – Manual for computation of Homoplasy Excess Tests

(Kevin Schneider, Stephan Koblmüller, Kristina Sefc – 17.07.2015)
(kevin.schneider@edu.uni-graz.at, stephan.koblmueller@uni-graz.at, kristina.sefc@uni-graz.at)

Index

1. R GUI installation & packages

The program is executed as source code within the R general user interface (R GUI), via the command line, or an integrated development environment (IDE) of your choice (e.g. R Studio). You can download the current and older versions of R from the CRAN website **cran.r-project.org** for Linux, Mac OS X, and Windows. For instructions on how to install R on your computer as well as for further information regarding R statistical packages, we also refer to this website. The HExT program requires several R packages which have to be installed before running the program. The required packages are:

- **ape** (analysis of phylogenetics and evolution; Popescu et al. 2012, Paradis et al. 2004)
- **phangorn** (phylogenetic analysis in R; Schliep 2011)
- **tools** (basic R package included in standard installation)
- **foreach**
- **doMC** on UNIX based operating systems (Linux, Mac OS X) or **doSNOW** on Windows (You do not have to further specify the operating system)

To install the packages, open the R GUI and select the required R packages via **Packages → Install package(s)… → Select a CRAN mirror (e.g. the country you live in) → Select the package to install**.
Alternatively, in an R IDE of your choice (e.g. R Studio) or in the command line, use the command

- **install.packages(c('ape', 'phangorn', 'foreach', 'doMC'))** on Unix based operating systems

or

- **install.packages(c('ape', 'phangorn', 'foreach', 'doSNOW'))** on Windows.

If installation has been successful, a message is printed to your screen.

2. Input files

2.1. Data file

The data file contains sample names and genotype data.
Currently, HExT accepts three types of data: binary data (such as AFLP or RFLP), SNP nucleotide data, and genotype data.
Sample names may include letters, digits, and the following special characters **. - + ! ?** .

2.1.1. Binary data

Nexus and fasta file formats are supported. Rows represent samples and columns represent loci (one column per locus). Loci within each row are not separated by any character. **1** means presence of an AFLP/RFLP band (or any other binary trait), **0** means absence of a band, and **?** refers to missing data.

Example for nexus format, 5 individuals genotyped at 14 marker loci:

```
#NEXUS

Begin data;
        Dimensions ntax=5 nchar=14;
        Format datatype=standard;
        Matrix
sample1     11100000111101
sample2     1??00000001001
sample3     110111100101?0
sample4     11011101110111
sample5     11011001110110
;
End;
```

The same example as above in fasta format:

```
>sample1
11100000111101
>sample2
1??00000001001
>sample3
110111100101?0
>sample4
```

```
11011101110111
>sample5
11011001110110
```

2.1.2. SNP data: nucleotides

Here, SNP loci are scored as diploid nucleotide data in the form AA, AT, AG, TC, TT, etc. Thus, one pair of nucleobases represents the two bases per locus in the diploid genome of one individual. This format allows multiallelic loci.

The SNP data input file may be in fasta or nexus format without separators between loci. For example (fasta):

```
>sample1
AAGGCTTTAAGGTT
>sample2
GGGGATTTAAGCTT
>sample3
GGGCATTTAAGGAT
>sample4
GGCCATTCTTGCAT
>sample5
GGCCATTCTTCCGG
…
```

, alternatively, in nexus format:

```
#NEXUS

Begin data;
        Dimensions ntax=5 nchar=14;
        Format datatype=standard;
        Matrix
sample1         AAGGCTTTAAGGTT
sample2         GGGGATTTAAGCTT
sample3         GGGCATTTAAGGAT
sample4         GGCCATTCTTGCAT
sample5         GGCCATTCTTCCGG
;
End;
```

2.1.3. Genotype data

The genotype format can be used for any codominant biallelic markers. Genotypes are coded as heterozygous ('0', '2') and homozygous ('1'); in other words, **0** stands for 'homozygous for allele A', **2** stands for 'homozygous for allele B', and **1** stands for 'heterozygous with alleles A and B'. Missing data are coded by **-1**. This data format is, for example, generated by SNP genotyping arrays.

In this format, loci are given in rows and samples in columns. Within each row, genotypes of different samples are separated by empty spaces.

For example:

```
       sample1 sample2 sample3 sample4 sample5
Locus1 1 1 2 0 0
Locus2 -1 -1 1 2 0
Locus3 -1 1 0 1 1
Locus4 2 2 2 2 2
Locus5 1 1 1 2 2
…
```

2.2. Specification of outgroup taxa

The use of an outgroup is optional. A list of samples to be used as outgroup can be typed in the command line, or be provided in a separate text file.
In both cases, sample names have to be separated by empty spaces. Make sure that sample names in the outgroup specification file are spelled exactly as in the data file. End with a line break (otherwise, HExT will write a warning, but computations will be ok).

Example for outgroup specification:

```
sample11 sample12 sample13 sample14
```

… saved as text file or typed/pasted into the command line.

2.3. Taxon-jackknifing

Taxon-jackknife sets are samples (or groups of samples) that are excluded from the dataset one at a time. For each of these taxon-jackknives, a neighbour-joining tree and bootstrap support values are calculated.
The number of taxon-jackknives is not limited by the program, but keep in mind that the bootstraps will be calculated for each jackknife dataset and computation time may become long.

Taxon-jackknife sets can be specified within the data file by the addition of jackknife set identifiers to the sample names (see 2.3.1). In this case, each sample can only belong to one single jackknife set.
Sometimes it may be desirable to have more flexibility in the specification of taxon-jackknife sets, e.g. to include particular samples in more than one taxon-jackknife set. Therefore, it is also possible to specify taxon-jackknife sets in a separate file or as command line input (see 2.3.2).

Note: Keep in mind that the taxon-jackknives produce the bootstrap values among which outliers shall be detected. Jackknife sets with overlapping composition will not produce independent data, which interferes with outlier detection. For example, including a hybrid taxon in five different jackknife sets may result in five taxon-jackknife trees with increased bootstrap support for the parental clades. In this case, these increased bootstrap values may not be recognized as outliers.

2.3.1. Specification in data file

Here, the endings of the sample names in the data file identify the taxon-jackknife set to which a sample belongs. The set identifiers must be separated by an underscore character _ from the rest of the sample name. No more than one underscore character is allowed within a sample name. For example (data in fasta format):

```
>sample1_set1
11100000111101
>sample2_set1
1??00000001001
>sample3_set1
110111100101?0
>sample4_set2
11011101110111
>sample5_set2
11011101110111
…
```

Consequently, HExT will exclude sample1 - sample3 for one taxon-jackknife run, and sample4 – sample5 for another taxon-jackknife run.

HExT asks whether the taxon-jackknife sets are specified in the data file, or input separately. If you choose to use the data file specification, it is still possible to specify additional jackknife sets as described in 2.3.2.

2.3.2. Specification in command line or in a jackknife file

As alternative to specifying taxon-jackknife sets in the data file, the composition of taxon-jackknife sets can be typed/pasted into the command line or read from a separate file. The format is the same in both cases. Sample names within a jackknife set are separated by empty spaces, and jackknife sets are separated by commas. **Important: commas must not be flanked by spaces!**

The following example specifies the same two jackknife sets as in 2.3.1:
sample1 sample2 sample3,sample4 sample5

It is possible to include samples in more than one jackknife set:
sample1 sample2 sample3,sample4 sample5,sample1 sample2 sample3 sample4 sample5

Note: Keep in mind that the taxon-jackknives produce the bootstrap values among which outliers shall be detected. Jackknife sets with similar compositions will not produce independent data, which interferes with outlier detection. For example, including a hybrid taxon in five different jackknife sets may result in five taxon-jackknife trees with increased bootstrap support for the parental clades. In this case, these increased bootstrap values may not be recognized as outliers anymore.

3. Running the program

Chosen options within the explanatory text are in ***bold italics***.

To run the program in your R GUI, select **File → Source R code → Load HExT_script.r**. Alternatively, if you run the program in an R IDE or at the command line, use the command **source('HExT_script.r')** if the program is saved in your current R working directory, or replace the part within quotes with the respective complete file path. Then just follow the instructions printed on your screen (see 3.1 to 3.10).

It is not necessary to load the required R packages (see section 1) manually, as this will be done by HExT upon execution of the program (but they have to be installed on your computer before you start to use HExT).

After starting the program, the following user input will be prompted:

3.1. Analysing previous results

This prompt will appear only if HExT has been run before on your computer. Each HExT run outputs the file HExT.RData with all run data, which can be accessed and analysed as described in section 5 (Exploring analysis results).
Typing ***n*** (no) takes you to the start of a new analysis.

3.2. Bootstrap number

Next, the program will ask for the number of bootstrap trees to be calculated. Note that bootstraps are calculated for the full dataset and for each taxon-jackknife dataset. Very large numbers of bootstraps will result in long computation times.

3.3. Rooting

Next, you can specify whether or not to root the trees with an outgroup. If no outgroup is specified, you still have the option to apply midpoint rooting or no rooting. The rooting option determines how trees are visualized but does not affect how the HET is computed (with the exception that it is not possible to include outgroup taxa in taxon-jackknife sets).

3.4. Outgroup

In the case of rooted trees, outgroup taxon names can be typed in the command line, or, by typing ***file***, can be retrieved from a text file (see 2.2). In both cases, the outgroup sample names have to be separated by empty spaces and have to be identical to the sample names used in the input data file. **Note:** If you root by outgroup and specify taxon-jackknife sets in the command line or in a file (see 2.3.2.), make sure that the outgroup taxa are not included in any taxon-jackknife set.
After you have specified your outgroup, you will be asked whether you expect your outgroup to be monophyletic in all bootstrap trees (consistent outgroup monophyly). The monophyly criterion of outgroups to root the tree is required by the package ape (see ape documentation for details). Choosing ***y*** (yes) for a non-monophyletic outgroup will lead to abortion of the program and an error message. However, by choosing the alternative option ***n*** (no), the root is set using the first sample listed in your outgroup. Nevertheless, all individuals of the outgroup are used in distance calculations and tree construction. Thus, both options will lead to the same analysis results and only tree visualization will be altered (as the root is set differently).

3.5. Taxon-jackknifing

Next, you are asked whether HExT should use taxon-jackknife sets as specified in the data file, or whether jackknife sets are specified separately (see 2.3). When choosing the data file option, it is still possible to specify additional jackknife sets. HExT will prompt you for input of additional jackknife sets.

3.6. Missing data

Now you will be asked whether or not to include positions with missing data in the analysis.
n (no): complete deletion of positions with missing data (positions deleted in all samples).
y (yes): pairwise deletion of positions with missing data; increases computation time (positions deleted only in comparisons of samples for which data are actually missing).

3.7. Data type

Next, you are asked to specify the type of your data (binary, SNP, or genotype) as described in the data file section (2.1).

3.8. Number of threads

Next, you are asked how many threads you want to use for bootstrapping. Specifying more threads than are available on your computer will lead to an error message. Running threads in parallel reduces computation times substantially.

3.9. Boxplot orientation

Next, the program prompts you to choose the orientation of the bootstrap boxplots in the graphical output (***h*** for horizontal and ***v*** for vertical).

3.10. Boxplot whiskers/outlier range

The default outlier rule in R is Q3 + 1.5*IQR and Q1 – 1.5*IQR (Tukey 1977), with Q3 = quartile 3, Q1 = quartile 1, and IQR = interquartile range (i.e. Q3 – Q1). Thus, values exceeding these range boundaries are identified as outliers. To offer the possibility of a more conservative outlier definition (or the opposite), the value of 1.5 in the above outlier rule can be changed.

3.11. Data file

Finally, you have to select the input data file (see 2.1.).

4. Output

HExT creates up to 18 output files, which are saved in the same directory as the data file. Additionally, the file **parameters.txt** lists the run parameters (number of bootstraps, rooting options, taxon-jackknife options, etc.).

In the following, the output files are described in the order in which they appear while the program is running.

4.1. *full_tree.nwk*

Stores a neighbour-joining tree in newick format including all taxa of the data file (the 'full tree'). The file just includes the topology without bootstrap support values and is output before the actual bootstrapping starts (to allow a quick check of the phylogeny). The node labels are consecutive numbers starting with 1 at the most inclusive node and with highest numbers given to the outer nodes.

4.2. *trees_bootlabels.nwk*

Includes all newick trees (i.e. the tree including all taxa and the taxon-jackknife trees). Nodes are labelled with bootstrap support values.

4.3. *trees_numlabels.nwk*

Same trees as in 4.2., with nodes labelled by consecutive numbers.

These numbers are used to refer to the nodes in the other output files. Starting from the most basal node (node number 1), node label numbers in rooted trees increase as one moves from the base of the tree to the tips and from the top (if the root is placed left) to the bottom. In unrooted trees, node label numbers increase with increasing (absolute) distance from the most basal (inclusive) node, which is again given the number 1. This file does not include bootstrap support values.

4.4. *nodes.txt*

Includes bootstrap support values and descending tips (samples) for each node in each tree. Tree number 0 is the tree calculated with the full dataset. The taxon-jackknife trees are numbered consecutively according to the order of taxon-jackknife sets.

4.5. *jackknife_specification.txt*

This file allows you to look up the composition of the taxon-jackknife sets. It provides a list of taxon-jackknife trees and the samples included in the respective taxon-jackknife sets.

4.6. *raw_bootstrap_table.txt*

Here, bootstrap support values of all nodes of the full tree present in the full and jackknife trees are stored in tabular format. For each of the nodes in the full tree, bootstrap values obtained in the full tree as well as in each taxon-jackknife tree are given (tree 0 = full tree; tree 1, tree 2, tree 3, etc. = taxon-jackknife trees).

In the taxon-jackknife trees, cells for nodes which had joined the excluded taxa in the full tree are filled with NA.
This table also includes BS values for nodes which were sister to the excluded taxon in the full tree. We do not recommend these BS values in the HET, because excluding a taxon from a phylogenetic tree will in most cases increase the bootstrap support for its sister clade irrespective of hybridization-induced homoplasy (“support carryover”).
In the file *bootstrap_table.txt* (4.9.), the bootstrap values arising from the exclusion of a sister clade (see 4.6) are replaced by NA.

4.7. *boxplots.pdf*

For each of the nodes in the full tree, this graph shows standard R boxplots of the bootstrap values in the taxon-jackknife trees.
Bootstrap support values for clades that are sister to the excluded taxa are not considered, because they are expected to increase independent of hybridization-induced homoplasy.

4.8. *boxplots.eps*

The same as in 4.7. in eps format.

4.9. *bootstrap_table.txt*

A tabular compilation of bootstrap values. For each of the nodes in the full tree, bootstrap values obtained in the full tree as well as in each taxon-jackknife tree are given (tree 0 = full tree; tree 1, tree 2, tree 3, etc. = taxon-jackknife trees).

Bootstrap values arising from the exclusion of a sister clade (see 4.6) are replaced by NA in this table and can be looked up in the file *raw_bootstrap_table.txt* (4.6).

This file allows the user the examination and further analyses of the bootstrap values produced by taxon-jackknifing; for example, the use of alternative outlier detection methods or analyses of particular nodes/trees.

4.10. *outliers.txt*

Provides information on the bootstrap outliers identified by the boxplot method. For each detected outlier, the file reports the node at which the outlier occurred, the taxon-jackknife tree in which it occurred, the bootstrap value for this node in the jackknife tree and in the full tree and the difference between the two, and the composition of the taxon-jackknife set whose exclusion gave rise to this outlier.
The outliers are sorted according to the difference in bootstrap support between the full and the taxon-jackknife tree, with the largest increase in bootstrap support at the top and the largest decrease at the bottom. You may open the file in Excel in order to sort by different criteria.

4.11. *upper_outlier_boxplots.pdf*

Boxplots of bootstrap values for nodes with upper outliers (larger than median of bootstrap distribution).
Note: Lower outliers (e.g. the disintegration of certain clades upon exclusion of certain taxa) may also be informative with respect to hybridization.

4.12. *upper_outlier_boxplots.eps*

The same as in 4.11. in eps format.

4.13. *alt_topologies.txt*

Lists all nodes that are found in the course of the bootstrap analyses of the full and taxon-jackknife datasets (including, for instance, nodes occurring only in a jackknife tree but not in the full tree, and nodes which are only present in trees constructed in the process of bootstrapping).
Includes information on the tree in which a node occurred, bootstrap support for this node in this tree, and the samples descending from this node.

The nodes are sorted by their bootstrap support and by the trees in which they occurred. Opening the file in Excel allows to sort the information according to other criteria, e.g. by clade (samples joined by node). This can be useful to look up bootstrap support for clades which do not occur in the full tree. For this purpose, HExT also offers an option to retrieve bootstrap information on particular nodes identified by the user (see section 5).

4.14. *HExT.RData*

Stores all analysis results in R format and can be accessed in subsequent HExT sessions (the program will ask you if you want to explore previous analysis results). Two copies of this file are stored, one in your current R working directory and the other in the input file directory. The path to the working directory can be looked up with the command **getwd()**. Data stored in HExT.RData can be used for further analyses in R. The data in this file includes the following objects:

- Bootstrap support data (HExT_list[[1]])
- Clades ordered by node number (HExT_list[[2]])
- Number of bootstraps per tree (HExT_list[[3]])
- Number of jackknives (HExT_list[[4]])
- Taxon-jackknife sets (HExT_list[[5]])
- Directory of your data file (HExT_list[[6]])
- Complete set of clades of all bootstrap trees (HExT_list[[7]])

The indexing shown in parentheses can be used to access the desired objects.
The HExT.RData file in your current R working directory is overwritten after each run. To access the file again with HExT, you have to copy it from the input file directory back into the current R working directory and use the original filename (HExT.RData).

+#
The following output files are created in the course of the analysis of results stored in HExT.RData (see sections 4.13 and 5).

4.15. *custom_#.pdf*

Boxplots of bootstrap values for a custom set of nodes/trees.

4.16. *custom_#.eps*

The same as in 4.15. in eps format.

4.17. *custom_#.txt*

Bootstrap values retrieved for a custom set of nodes/trees.

4.18. *outliers_custom_#.txt*

Same as *outliers.txt* (4.10.) for a custom set of nodes/trees.

5. Exploring analysis results

This option allows you to retrieve bootstrap information (boxplots and underlying bootstrap values) for particular nodes, including nodes that do not occur in the full tree.

HExT stores all analysis results in the file HExT.RData, which it can access in subsequent sessions (see 4.14) in order to extract bootstrap support values for any nodes that occurred in the course of the bootstrap analyses of the full and the taxon-jackknife datasets.

If the R working directory contains a file HExT.RData, HExT will ask you whether you want to analyse these results. Typing **n** (no) takes you to the start of a new analysis. If you type ***y*** (yes), you are asked to specify whether you want the boxplots in horizontal or vertical orientation.

Next, you are asked to specify the nodes for which you would like to extract and display bootstrap values. A node is specified by the samples which it joins into a clade, or by its node number in the full tree (only possible if this node occurs in the full tree). Separate the sample names within a node specification by empty spaces, and node specifications by commas (no spaces around commas). If you are searching for a node that occurs in the full tree, type ***#nodenumber*** (e.g. #25). Look up node numbers in the tree plotted in *trees_numlabels.nwk*.

You can either search for a node in all taxon-jackknife trees or in one particular tree (full tree or any taxon-jackknife tree). At the end of the sample list, type ***all*** for searching in all jackknife trees, or the tree number to search for in one particular tree.

Example: to search for the node (sample1 sample2 sample3) in the full tree (tree 0) as well as in all trees; and search for node #10 in the full tree and in all trees, type

```
sample1 sample2 sample3 0,sample1 sample2 sample3 all,#10 0,#10 all
```

This provides you with a graph marking the bootstrap support for these nodes in the full tree and the distributions of bootstrap values for these nodes across all taxon-jackknife trees.

HExT offers to save the boxplots and underlying data, which creates output files 4.15 to 4.18. HExT asks whether you would like to change the title and the tick labels (nodes) of the boxplots. In boxplots with horizontal orientation, the order of tick labels is from bottom to top.

Note: The HExT.RData file in your current R working directory is overwritten after each run. To access the file again with HExT, you have to copy it from the input file directory back into the current R working directory and use the original filename (HExT.RData).

6. Program details

6.1. Pairwise genetic distance

6.1.1. Nei-Li distance

If the input data is **binary**, **Nei-Li distances** (Nei & Li 1979) are calculated, which put more weight on the presence of a trait in both sequences of a compared sequence pair and are particularly suitable for AFLP and RFLP data, i.e. dominant binary genetic markers. The formula for Nei-Li distance calculations is the following:

$$NL_{xy} = 1 - \frac{2 * n_{11}}{2 * n_{11} + n_{10} + n_{01}}$$

, where NL_{xy} is the Nei-Li distance, n_{11} is the number of alleles shared by *x* and *y*, n_{10} is the number of alleles present in *x* and absent in *y*, and n_{01} the number of alleles present in *y* and absent in *x* (Lombard et al. 1999). If you choose to exclude missing data from the calculation of the distance matrix, complete deletion of missing data will be applied, meaning that positions with missing data will be deleted from all samples. Alternatively, pairwise deletion of missing data will be applied, and positions are deleted only in comparisons with those samples in which the data are missing.

6.1.2. Allele sharing distance

If your data matrix is of type **SNP** or **genotype**, **allele-sharing distances** (e.g. McCue et al. 2012, Purcell et al. 2007) are calculated. The formula used is the following:

$$D = 1 - \frac{IBS2 + 0.5 * IBS}{N}$$

, where *D* is the allele-sharing distance, *IBS2* and *IBS1* are the number of loci that share either 2 or 1 alleles identical by state (IBS), respectively, and *N* is the number of loci in your data (McCue et al. 2012).

6.2. Phlyogenetic tree construction

This program employs the **neighbour-joining** tree estimation as implemented in *ape* (Popescu et al. 2012, Paradis et al. 2004) and originally proposed by Saitou and Nei (1987). One of the main arguments in favour of the distance-based neighbour-joining method is the short computation time compared to more complex probability-based methods that would increase analysis time by several orders of magnitude.

6.3. Multithreaded bootstrapping

Bootstrapping can be done using more than one thread of your processor at a time. Parallel computing is achieved using the R packages *foreach*, as well as *doMC* on UNIX based operating systems (Linux, Mac OS X) and *doSNOW* on Windows.

7. References

Lombard, V., Baril, C. P., Dubreuil, P., Blouet, F., & Zhang, D. (1999). Potential use of AFLP markers for the distinction of rapeseed cultivars. In *INTERNATIONAL RAPESEED CONGRESS* (Vol. 20).

Felsenstein, J. (1985). Confidence limits on phylogenies: an approach using the bootstrap. *Evolution*, 783-791.

McCue, M. E., Bannasch, D. L., Petersen, J. L., Gurr, J., Bailey, E., Binns, M. M., ... & Mickelson, J. R. (2012). A high density SNP array for the domestic horse and extant Perissodactyla: utility for association mapping, genetic diversity, and phylogeny studies. *PLoS genetics*, 8(1), e1002451.

Nei, M., & Li, W. H. (1979). Mathematical model for studying genetic variation in terms of restriction endonucleases. *Proceedings of the National Academy of Sciences*, 76(10), 5269-5273.

Paradis, E., Claude, J., & Strimmer, K. (2004). APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20(2), 289-290.

Popescu, A. A., Huber, K. T., & Paradis, E. (2012). ape 3.0: New tools for distance-based phylogenetics and evolutionary analysis in R. *Bioinformatics*, 28(11), 1536-1537.

Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A., Bender, D., ... & Sham, P. C. (2007). PLINK: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics*, 81(3), 559-575.

R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Saitou, N., & Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4), 406-425.

Schliep, K. P. (2011). phangorn: Phylogenetic analysis in R. *Bioinformatics*, 27(4), 592-593.

Tukey, J.W. (1977). Exploratory Data Analysis. Addison-Wesley, Reading, MA.

8. Appendix

Example analysis

Example data, questions asked by the program, and answers to these questions are written in Arial font. Explanations to example data, questions, and answers appear in Calibri font.

Suppose you have a dataset of diploid SNPs in nucleotide form such as the following:

```
>ind1_species1
GGCC??TTCCCCGGAAGGGGTCGG …
>ind2_species1
GGCCCCTTCCCCGGAAGGGGCCGG …
>ind1_species2
GGAA??TTCCCCGGAAGGGG??GG …
>ind2_species2
GGCC??TTCCCCGGAAGGGGCCGG …
…
```

In the following example, we use the input files provided along with the HExT script.

Start by loading HExT_script.r

If HExT.RData files of previous sessions are located in R’s current working directory, you are asked if search for specific nodes in this file (i.e. custom search) is desired. We do not want this now, so proceed by typing ***n*** (no):

```
ANALYSING PREVIOUS RESULTS -
Analysis results of a previous session were found in your current
R working directory. Do you want to search for specific nodes in
these results?
(y/n): n
```

Now, you can specify the number of bootstraps per tree (i.e. full and jackknifed datasets):

```
BOOTSTRAP NUMBER -
Please type in the number of bootstrap replications.
: 1000
```

Next, you are asked whether or not outgroup rooting should be applied:

```
ROOTING (OUTGROUP) -
Do you want to root the tree using an outgroup?
(y/n): y
```

The set of outgroup taxa can now be specified. This can be done by typing ***file*** to load a text file with the desired outgroup taxon names separated by empty spaces.

```
OUTGROUP SPECIFICATION -
Please specify the outgroup taxa by which the tree is to
be rooted. (separate taxon names by empty spaces)
[type 'file' for input via text file]
: file
```

… and select the outgroup specification file ‘4_example_outgroup.txt’

… or, alternatively, you can copy the outgroup taxon names from the file (4_example_outgroup.txt) directly in the command line.

Due to bootstrapping, outgroups can lose their monophyly in the bootstrapped tree. For this reason, you are asked to specify whether or not consistent outgroup monophyly during bootstrapping is to be expected. If you choose ***n*** (no), which is recommended, the first individual/sample of the specified outgroup taxa is taken for tree rooting. Keep in mind that this has only consequences regarding graphical representation in the tree output files.

```
CONSISTENT OUTGROUP MONOPHYLY -
Please note that inconsistent outgroups could cause errors.
Do you expect your outgroup to be monophyletic across bootstrap trees?
(if not: the first sample of your outgroup is used to root the tree)
(y/n): n
```

If jackknife sets are specified in your data file (as is the case in the example dataset), type ***y*** (yes):

```
DATA FILE TAXON-JACKKNIFING -
Please specify whether jackknife sets are specified in the data file.
(if yes: sample names must have the form 'samplename_set')
(y/n): y
```

If you want to use more jackknife sets than are specified in your data file, you have the option to define additional jackknife sets by typing ***y*** (yes).

```
ADDITIONAL TAXON-JACKKNIFE SETS -
Do you want to specify additional jackknife sets?
(y/n): y
```

If the additional taxon-jackknife sets shall be read from a file, type ***file*** upon the next prompt:

```
TAXON-JACKKNIFE SPECIFICATION -
Please specify the taxon-jackknife sets
(separate sample names by empty spaces and sets by commas).
(e.g. 01_species1 02_species1 03_species1,01_species2 02_species2)
[type 'file' for input via text file]
: file
```

and select the taxon-jackknife specification file ‘3_example_jackknifesets.txt’

or (instead of typing ***file***) copy the content of that file into the command line.

The next step is to choose either the option of including missing data with ***y*** (yes) or to exclude missing data with ***n*** (no). Inclusion of missing data increases computation times, sometimes to a substantial degree.

```
MISSING DATA -
Should missing data be included? (inclusion of missing
data may lead to longer computation times)
(y/n): y
```

Now, the type of your data has to be specified (see section 2.1 for details regarding supported data types). The example data are coded as SNP nucleotides, therefore choose ***n***

```
BINARY, SNP, OR GENOTYPE -
Is your data of type binary (b), SNP nucleotide (n),
or SNP genotype (g)?
(b/n/g): n
```

The number of threads to be occupied by the program during bootstrapping can now be chosen.

```
THREADS -
How many threads do you want to use in parallel for bootstrapping?
: 4
```

The preferred orientation of all boxplots in the output can be specified in the next step.

```
BOXPLOT ORIENTATION -
Do you want the boxplot orientation to be horizontal (h) or vertical (v)?
(h/v): h
```

Next, you can choose the range of whiskers/outliers in the boxplots. This number is defined as the multiple of the interquartile range (i.e. the width of the box/the range from the first to the third quartile). When specifying ***1.5,*** a value with a distance ≤1.5 times the interquartile range from either the lowest or highest quartile will still be in the range of the boxplot whiskers. Values that are more extreme than that are counted as outliers.

```
BOXPLOT WHISKERS/OUTLIER RANGE -
Up to which multiple of the interquartile range should the boxplot
whiskers extend? (with values more extreme than that being outliers)
[1.5 is the standard value of R]
: 1.5
```

Finally, the input data file has to be chosen.

```
DATA FILE -
Please select a data file in nexus, fasta, or text format.
```

Select the file '2_example_data.fas'

After completion of a HExT run, you have the possibility to display single bootstrap support values or boxplots (bootstrap distributions) of custom sets of nodes.

```
CUSTOM -
Do you want to display bootstrap support of custom sets of nodes?
(y/n): y
```

Now, you can specify the nodes of interest in two ways.
1.) Write the node number (in the full tree, file 'trees_numlabels.nwk') after a ***#*** symbol*,* and ***all*** (to display boxplots consisting of bootstrap support values for this node in all jackknife trees) or, e.g., ***5*** (to display the bootstrap support value of this node in jackknife tree 5; look up the various jackknife trees in file 'jackknife_specification.txt'). Nodes must be separated by a comma. For example: ***#17 all,#17 5*** (to display the distribution of bootstrap values for node 17 and, additionally, the bootstrap value for this node in jackknife tree 5).
2.) Write the taxon labels of the desired node, separated by empty spaces, and ***all*** or a tree number as in 1.). The order of taxon labels does not matter. Again, nodes must be separated by a comma. It is also possible to combine both ways of specifying desired nodes.

```
NODE SPECIFICATION (CUSTOM) -
: #21 all,#21 5,#45 all #45 5
```

or (e.g. to look at the node defining monophyly of *V. x champinii*):

```
NODE SPECIFICATION (CUSTOM) -
: 4560195077I_champinii 4560195108K_champinii 4560195047H_champinii
4560195066H_champinii 4560195110I_champinii all
```

Tick labels have the form '#Node_Tree' (e.g. '#17_all' and '#17_5' for node 17 in all trees and jackknife tree 5 only, respectively) …

```
Tick labels: #Node_Tree
```

You are now asked whether or not your boxplots and the underlying outlier information should be saved (in pdf and text format, respectively).

```
SAVE -
Do you want to save the boxplots and underlying data of the
specified nodes to a file?
(y/n): y
```

You also have the option to change plot title and tick labels.

```
CHANGE TITLE & LABELS -
Do you want to change the plot title and tick labels?
(y/n): y
```

When changing plot title and tick labels, be sure to separate the plot title from the tick labels using a comma. The individual tick labels have to be separated by empty spaces (e.g. ***new title,newticklabel1 newticklabel2***).

```
TITLE & LABEL SPECIFICATION -
Specify the plot title first, followed by the tick labels.
Plot title and tick labels should be separated by a comma,
while the tick labels should be separated by empty spaces.
(e.g. main title,label1 label2 label3)
: outlier example1,node_21 node_21_tree_5 node_45 node_45_tree_5
```

Finally, you have the option to proceed or finish the analysis.

```
PROCEED (CUSTOM) -
Do you want to proceed accessing bootstrap support values
of custom sets of nodes?
(y/n): n
```

In case you want to search for nodes in bootstrap trees that were not present in the full or jackknife trees, you can also use the custom analysis tool in the same way as above. If the specified node occurred in none of the trees, the following message appears on your screen …

```
The specified node has not been found in any of the trees or bootstrap trees.
```

Function description

function	*description*
access_alttop.r	accesses BS support information from alternative topologies
access_custom	enables customized BS support queries; displays and (if desired) stores query results
all_descendants	returns the descending taxa for each clade of an input tree
alttop_search	searches for topologies of bootstrap trees not present in the full tree, sorts the obtained data frame, and stores it in a file
boot_excl	determines sister nodes and those that are lost in the course of jackknifing
descendants	determines the descending tips of a specified node
excl_spec	writes jackknife sets to a file
fas_parse	exctracts data and sample names from an input fasta file

file_input	calls input parsing functions on the basis of input file extensions
geno_parse	extracts the SNP genotype data and sample names from an input text file
get_trees	performs jackknifing and calls functions for tree construction and for obtaining tree bipartition information
inp_manip	splits specified jackknife sets or outgroups into taxon names
input_mod	parses input from the "interact" function
interact	accepts input and parameter values from the user
make_bin	converts an integer genotype matrix to four binary matrices
nex_parse	extracts data and sample names from an input nexus file
njf	calls tree construction and rooting functions
node_comp	compares tree topologies and collects BS support data
outlier_sink	stores boxplot outlier information in a text file and additional boxplots
parse_trees	organizes comparison of full and jackknifed trees, obtains BS support for trees and organizes storage of tree topologies
plot_manip	manipulates boxplots from the "access_custom" function
rowcol_names	assigns row and column names to the input matrix
seq_dist	calculation of Nei-Li or allele sharing distance matrices
sisternodes	determines sister nodes of jackknifed clades
store_plot	creates BS boxplots and stores these and the underlying BS information in pdf and txt files, respectively
tax_comp	collects taxa for jackknifing in a vector
write_nodes	writes descending taxa and BS support values of all nodes of all trees to a txt file
write_par	writes parameter information of HExT runs to a txt file