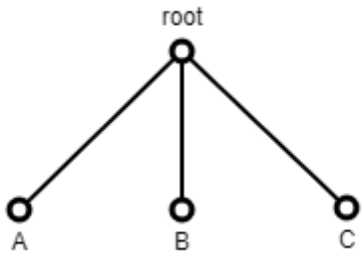
TreeCmp 2.0: comparsecison of trees in polynomial time – manual

1. Introduction

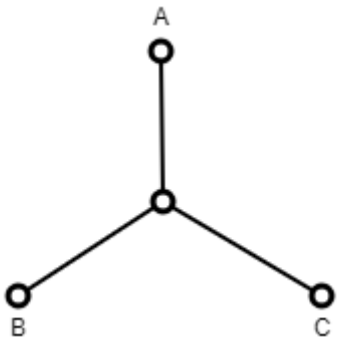
A phylogenetic tree represents historical evolutionary relationship between different species or organisms. There are various methods for reconstructing phylogenetic trees. Applying those techniques usually results in different trees for the same input data. An important problem is to determine how distant two trees reconstructed in such a way are from each other. Comparing phylogenetic trees is also useful in mining phylogenetic information databases. The TreeCmp application was designed to compute distances between arbitrary (not necessary binary) phylogenetic trees. All distances are implemented using polynomial time algorithms and all of them are metrics generating metrizable topological space.

2. Input data format

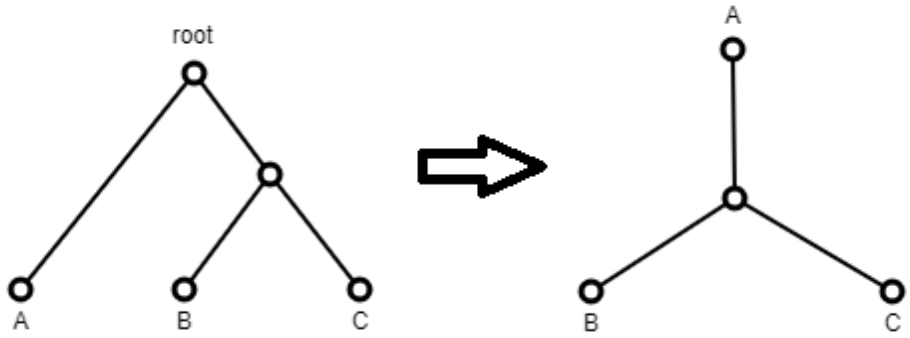
The TreeCmp software was designed to support BEAST (http://beast.bio.ed.ac.uk/) and MrBayes (http://mrbayes.csit.fsu.edu/) date files, where phylogenetic trees are stored in the NEWICK format. Note that plain text files containing only trees in this format are supported as well. The NEWICK format clearly specifies the vertex that is a candidate for the root. After choosing a metric for rooted trees, this vertex will always be treated as the root. For example, if a unrooted binary tree in the NEWICK format: (A, B, C) is entered in the metric dedicated for a rooted tree, it will be interpreted as a rooted, non-binary tree consisting of a 3-degree root and 3 descendant vertices A, B and C.



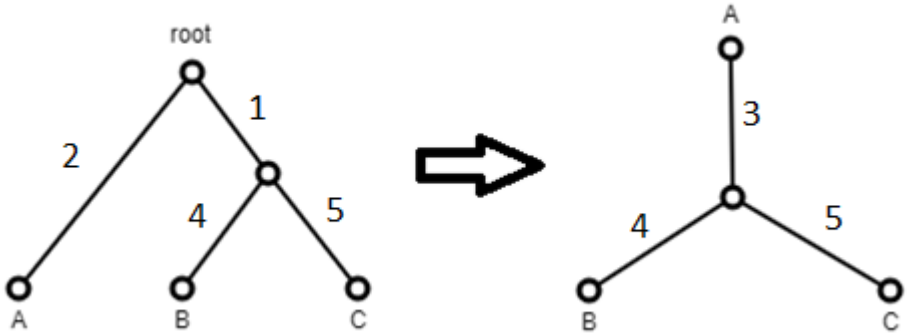
In the case when a rooted binary tree is entered to the metric dedicated for a unrooted tree, the root will be treated as an internal vertex or will be automatically shrinked if it's degree equals 2. For example, if a rooted binary tree in the NEWICK format (A, B, C) is entered in the metric for a unrooted tree, then root will be treated as an internal vertex and tree will be interpreted as a unrooted, binary tree (A, B, C).



However, after entering the rooted tree (A, (B, C)) to the metric dedicated for unrooted trees, the root will be shrinked as in the figure below.



In the case with weighted tree (tree with weights on the edge) the sum of edge weights incident to removed root vertex has been assigned to the new created edge as in the figure below.



In summary, any interference in the given rooted tree will take place only if calculated metric is dedicated to unrooted trees and the root degree is 2. In any other case it will be treated as an internal vertex.

3. Running TreeCmp

The TreeCmp application is distributed as a zip archive. In order to unpack the file any software supporting zip compression, for example free software 7-zip (http://www.7-zip.org/), can be used. In order to run the TreeCmp application Java VM in version at least 1.6 is required.

3.1. Directory structure

Description

bin		contains main jar file: **TreeCmp.jar** and lib folder with necessary open source libraries: pal-1.5.1 (http://www.cebl.auckland.ac.nz/pal-project/) and commons-cli-1.2 (http://commons.apache.org/cli/)
config		contains xml configuration file
data		contains text files with pre-computed data (average value and other statistics) for all the 12 metrics under the two models of generation of random binary trees: the Yule model and the uniform model.
examples		contains subdirectories with examples
	align	contains an example of creating alignments
	beast	contains an example input file created using BEAST
	mr_bayes	contains an example input file created using MrBayes
	plain	contains an example input file with plain trees
	plain2	contains an example input file with plain trees
	prune	contains an example of comparing trees having different sets of taxa
	ref_tree	contains an example of comparing reference trees to a set of trees
	scaled	contains an example with reporting scaled values of chosen metrics
src		contains source code of this application

3.2. Command line syntax

Usage:

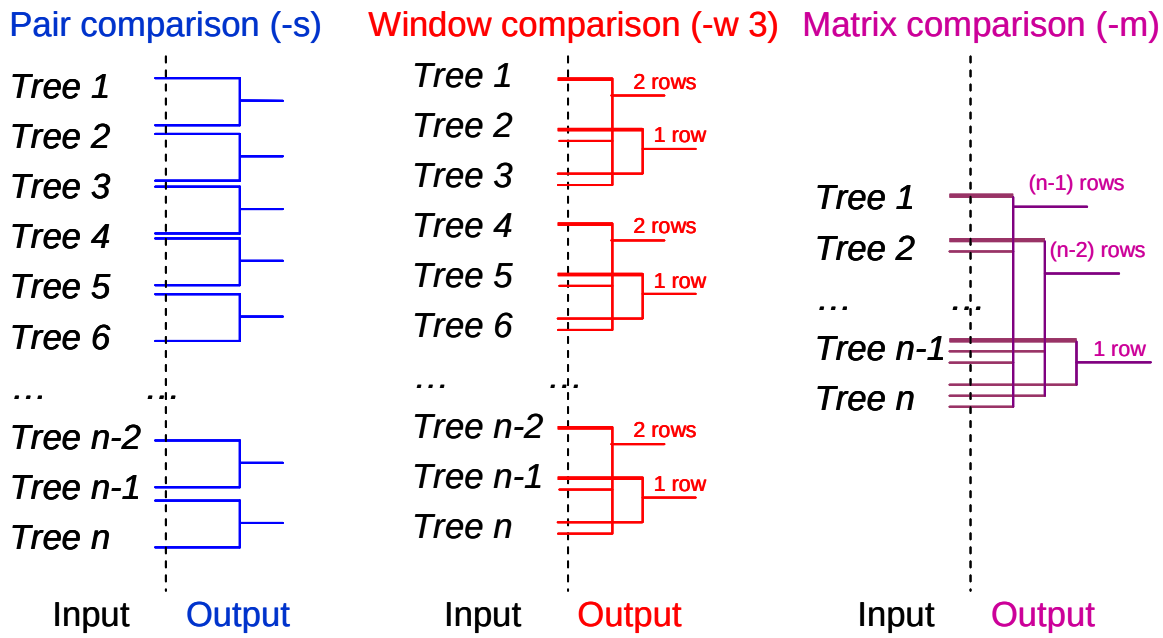
```
java -jar TreeCmp.jar -w <size>|-s|-m|-r <refTreeFile> -d <metrics> -i
<inputfile> -o <outputfile> [-N] [-P] [-I] [-A|-O]
```

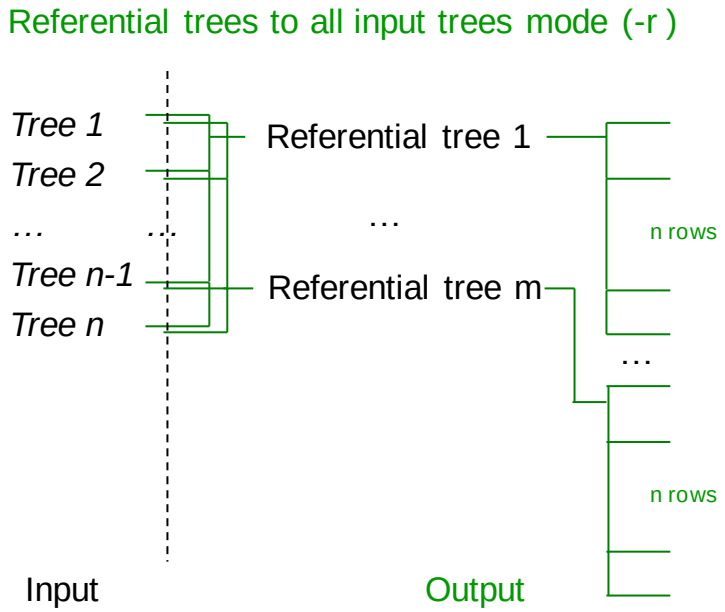
See section 4 for details regarding output file format for a particular combination of the options.

Mandatory switches:

- The comparison mode options (only one option should be specified):
  - `-s` – overlapping pair comparison mode; every two neighboring trees in the input file are compared,
  - `-w <size>` – window comparison mode; every two trees within a window with a specified size are compared – the average distance and the standard deviation go to the output file,
  - `-m` – matrix comparison mode; every two trees in the input file are compared.
  - `-r <refTreeFile>` – reference trees to all trees mode. Each tree in the input file is compared to all reference trees.

  Details of the computation flow in each of these case are explained in the pictures below.





- The metric option (`-d`). At least one and at most 12 metrics can be specified (numbers in square brackets correspond to the reference list. Metrics should be separated by space character.

  Metrics for rooted trees:
  - `tt` – the Triples metric (Crichlow et al. 1996),
  - `rc` – the Robinson-Foulds metric based on clusters (Robinson and Foulds 1981),
  - `mp` – the Matching Pair metric (Bogdanowicz and Giaro 2014),
  - `ns` – the Nodal Splitted metric with L^2 norm (Cardona et al. 2010),
  - `mc` – the Matching Cluster metric (Bogdanowicz et al. 2012),
  - `mt` – the Rooted maximum agreement subtree distance (Farach and Thorup 1994),
  - `co` – the Cophenetic Metric with L^2 norm (Cardona, Mir, Rosselló, Rotger and Sánchez 2013).

Metrics for unrooted trees:

- `qt` – the Quartet distance (Estabrook 1985),
- `pd` – the Path difference distance (Steel and Penny 1993),
- `rf` – the Robinson-Foulds distance (Robinson and Foulds 1981),
- `ms` – the Matching Split distance (Bogdanowicz and Giaro 2012),
- `um` – the Unrooted maximum agreement subtree distance (Farach and Thorup 1994).

Example: `-d ms rf`

- IO options (both options should be specified):
  - `-i <inputfile>` – input data file with trees in the NEWICK format,
  - `-o <outputfile>` – output data file with the results of computations.

Optional switches:

- General options:
  - `-N` – report normalized distances δ_m for a particular metric *m* (Bogdanowicz et al. 2012; based on an average value from pre-computed data). This functionality is available for trees with number of leaves between 4 and 1000. Note that normalized tree similarity for a particular metric *m* (NTS_m) can be expressed by normalized distance as follows: $NTS_m = 1 - \delta_m$ (Bogdanowicz et al. 2012).
  - `-P` – prune compared trees if needed. This option is design to allow comparing trees having different (partially overlapping) sets of taxa. After using this option three additional columns appear in the output file (see section 4 for details).
  - `-I` – -include summary section in the output file.

- Matching metric specific options (only one option should be specified).
  - `-A` – Generate alignment files – this option should be used together with selection the MS or MC metrics. As a result additional files containing aligned splits or clusters are generated:
    - [output_file_name].out.aln_MS.txt,
    - [output_file_name].out.aln_MC.txt,

    where [output_file_name] is the file name specified after -o option.
  - `-O` – use special implementations of MS/MC metrics optimized for similar trees.

Note that if a rooted tree (with bifurcation in the root) is compared using metrics for unrooted trees the tree will be automatically transform into unrooted one, i.e., the bifurcation will be replaced with an arbitrary trifurcation.

4. Output data format

Output files created by the application regardless of chosen mode have similar structure. Output files are tab separated text files (TSV), which means that they can be easily read by various data analysis software (e.g. MS Excel, R, OpenOffice.org). For direct saving in CSV or Microsoft Excel format see subsection 6.4 An output file consists of two sections. The first section contains formatted in rows values of distances in selected metrics. The second (optional) section contains summary data computed based on all rows that appears in the first section.

4.1. Basic output file structure

Base output file format for options -s, -m, and -w

No	Tree1	Tree2	MetricName_1	MetricName_2	…	MetricName_n
Comparison number	Tree1 number	Tree2 number	Distance value	Distance value	…	Distance value

Base output file format for option -r,

No	RefTree	Tree	MetricName_1	MetricName_2	…	MetricName_n
Comparison number	Reference tree number	Tree number	Distance value	Distance value	…	Distance value

`Tree, tree1, tree2 numbers` in the output file correspond to the number of the tree in the input file.

The following table contains a mapping between available metrics and column names in the output file that are related to them.

Metric name in the output file	Full metric name	TreeCmp command line parameter
Triples	the Triples metric	tt
R-F_Cluster (0.5)	the Robinson-Foulds metric based on clusters	rc
MatchingPair	the Matching Pair metric	mp
NodalSplitted	the Nodal Splitted metric with L^2 norm	ns
MatchingCluster	the Matching Cluster metric	mc
MAST	the Rooted maximum agreement subtree distance	mt
CopheneticL2Metric	the Cophenetic Metric with L^2 norm	co
Quartet	the Quartet distance	qt
PathDiffernce	the Path difference distance	pd
R-F(0.5)	the Robinson-Foulds distance	rf
MatchingSplit	the Matching Split distance	ms
UMAST	the Unrooted maximum agreement subtree distance	um

4.2. Additional columns (-P and -N options)

After using switch -P the following three columns appear additionally in the output file.

Tree1_taxa	Tree2_taxa (or RefTree_taxa)	Common_taxa
Number of taxa in the first tree	Number of taxa in the second (or reference) tree	Number of taxa in common

After using switch -N the following two columns per each chosen metric appear additionally in the output file. These columns contain the value of the distance in a particular metric divided by its empirical average value. If the number of common leaves in compared trees is out of supported range (which is form 4 to 1000), then "N/A" value is inserted.

MetricName_toYuleAvg	MetricName_toUnifAvg
(Distance value)/(Empirical average value in the Yule model)	(Distance value)/(Empirical average value in the uniform model)

For details regarding generating phylogenetic trees under the Yule and uniform models see (McKenzie and Steel 2000; Semple and Steel 2003).

4.3. Summary section format (-I option)

Name	Avg	Std	Min	Max	Count
Metric name 1	Average value	Standard deviation value	Minimal value	Maximal value	Number of analyzed values
Metric name 2	…	…	…	…	…
…	…	…	…	…	…
Metric name n	…	…	…	…	…

5. Useful Java VM parameters

In the case of an analysis of large trees the following exceptions might occur:

1. Exception in thread "main" java.lang.OutOfMemoryError: Java heap space

   To solve the problem increase Java heap space memory limit using JVM option –Xmx
   Example:

   ```
   java –Xmx700m –jar TreeCmp.jar <further options>
   ```

2. Exception in thread "main" java.lang.StackOverflowError at
   pal.io.FormattedInput.skipWhiteSpace(FormattedInput.java:111)
   at pal.io.FormattedInput.readNextChar(FormattedInput.java:131)
   at pal.tree.ReadTree.readNH(ReadTree.java:81)
   …..
   at pal.tree.ReadTree.readNH(ReadTree.java:89)

   To solve the problem increase Java thread stack size limit using JVM option –Xss
   Example:

   ```
   java –Xss1m –jar TreeCmp.jar <further options>
   ```

These options can be used in conjunction.

6. Examples

6.1. Running application to compare trees using MS

Input file: `\examples\beast\testBSP.newick`
Invocation:

```
java -jar TreeCmp.jar -w 2 -d ms -i testBSP.newick -o testBSP.newick_w_2.out -I
```

Console output:

```
TreeCmp version 1.0-b291

Active options:
Type of the analysis: window comparison mode (-w) with window size: 2
Metrics:
  1. MatchingSplit (ms)
Input file: testBSP.newick
Output file: testBSP.newick_w_2.out
Additional options:
I - Include summary section in the output file.
-----
2011-08-27 16:03:17: Start of scanning input file: testBSP.newick
2011-08-27 16:03:17: End of scanning input file: testBSP.newick
2011-08-27 16:03:17: 11 valid trees found in file: testBSP.newick
2011-08-27 16:03:17: Start of calculation...please wait...
2011-08-27 16:03:17: 0.00% completed...
2011-08-27 16:03:17: 20.00% completed...
2011-08-27 16:03:17: 40.00% completed...
2011-08-27 16:03:17: 60.00% completed...
2011-08-27 16:03:17: 80.00% completed...
2011-08-27 16:03:17: 100.00% completed.
2011-08-27 16:03:17: End of calculation.
2011-08-27 16:03:17: Total calculation time: 62 ms.
```

Output file testBSP.newick_w_2.out:

```
No     Tree1  Tree2  MatchingSplit
1      1      2      58.0000
2      3      4      24.0000
3      5      6      10.0000
4      7      8      13.0000
5      9      10     14.0000
---------
Summary:
Name   Avg    Std    Min    Max    Count
MatchingSplit      23.8   17.73583942191629  10.0   58.0   5
```

6.2. Computing normalized distances

Reporting distances divided by pre-computed empirical average values for random trees (generated according to Yule and uniform models, -N option) can help in an interpretation of the similarity level of analyzed trees in chosen metric. This functionality is available for trees with number of leaves between 4 and 1000 by using `-N` option. In the following example, the distance in the MS metric of each tree from a given set to the reference tree is computed. Analyzed trees have 15 leaves.

Input files: `\examples\sclaed\ref_tree.trees`
`\examples\sclaed\test_set.trees`

Invocation:

```
java -jar TreeCmp.jar -r ref_tree.trees -d ms -i test_set.trees -o
test_set.trees.r.out -N
```

Output file test_set.trees.r.out:

No	RefTree	Tree	MatchingSplit	MatchingSplit_toYuleAvg	MatchingSplit_toUnifAvg
1	1	1	43.0000	1.0742	0.9663
2	1	2	43.0000	1.0742	0.9663
3	1	3	41.0000	1.0242	0.9214
4	1	4	40.0000	0.9992	0.8989
5	1	5	43.0000	1.0742	0.9663
6	1	6	41.0000	1.0242	0.9214
7	1	7	43.0000	1.0742	0.9663
8	1	8	41.0000	1.0242	0.9214
9	1	9	39.0000	0.9742	0.8764
10	1	10	40.0000	0.9992	0.8989
11	1	11	0.0000	0.0000	0.0000
12	1	12	6.0000	0.1499	0.1348

Basic interpretation:

- Tree number 11 has the same topology as the reference tree.
- Tree number 12 is very similar to the reference tree in comparison to similarly of random on 15 leaves (the normalized distance is about 0.15 and 0.13 depending on the random model).
- Trees with numbers 1 to 10 are approximately as similar to the reference tree as random trees to each other (the normalized distance is close to 1).

In ordered to perform more advance similarity analysis, e.g. involving different model of generation of random trees, user my need to use TreeCmp twice:

- to compute distances between custom set of random trees generated by other software, e.g. Evolver application form PAML package (http://abacus.gene.ucl.ac.uk/software/paml.html) to obtain the empirical average distance in a particular metric or its distribution,
- to compute the distance between analyzed trees.

6.3. Generating new data for computing normalized distances

If the number of compared trees leaves is greater than 1000, it is possible to manually generate a set of random trees and calculate statistics for them. To generate a set of trees we can use PRTGen program – phylogenetic random trees generator. Let's assume that we want to generate 2000 rooted trees on 1001 leaves using uniform model and save them to the file: trees.newick. Let's use command:

```
PRTGen -n 1001 -e 2000 -r -f trees.newick
```

Then, using TreeCmp, we calculate the value of the selected metric (for instance MC) between each subsequent pair of trees. We will get 1000 values:

```
java -jar TreeCmp.jar -w 2 -d MC -i trees.newick -o results.out
```

Based on these results, we can calculate desired values: (average, standard deviation, minimum, maximum, and subsequent quantiles: 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.97), e.g. in RStudio:

```
filename<-"<path_to_file>\results.out"
m<-read.table(filename,header = TRUE,sep = "\t")
v<-m[,4]
q_seq<-c(0.02,0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.95,0.97)
q<-quantile(v,q_seq,type=1,name=FALSE)
row<-c(1001,mean(v),sd(v),min(v),max(v),q)
outfile<-"<path_to_file>\row.out"
write(row,file=outfile,append=TRUE,ncolumns=length(row),sep="\t")
```

Line such obtained in row.out file should be pasted into the appropriate file in the data folder. In that case it will be: unif_MC.txt. Now Treecmp is ready for computing normalized MC distances for rooted trees on 1001 leaves based on uniform model.

6.4. Finding the most similar trees in the input file

The most convenient comparison mode for such purpose is a matrix mode (-m). In the following example, the Matching Split distance is used.

Input file: `\examples\plain2\plain2.trees`

```
(a,(b,c),(d,e));
(a,b,(c,(d,e)));
(((a,b),c),d,e);
(a,(b,(c,d)),e);
```

Invocation:

```
java -jar TreeCmp.jar -m -d ms -i plain2.trees -o plain2.trees.m.out
```

Output file plain2.trees.m.out:

No	Tree1	Tree2	MatchingSplit	
1	1	2	2.0000	
2	1	3	2.0000	
3	1	4	3.0000	
4	2	3	0.0000	← The most similar trees
5	2	4	3.0000	
6	3	4	3.0000	

Trees number 2, i.e.: (a,b,(c,(d,e))) and 3, i.e.:(((a,b),c),d,e) in the input file are the most similar. In fact, they have the same topology (trees are assumed to be unrooted as metric for unrooted trees is used) because their distance is 0.

6.5. Exporting data to other applications: MS Excel, R

To save a file in MS Excel format, just use the .xlsx extension in output data file name (option: -o <outputfile>.xlsx). Similarly, to save a file in CSV format, use the .csv extension in output data file name (option: -o <outputfile>.csv).

In order to pass data to R (http://www.r-project.org/) it is convenient to have the TreeCmp output file in a simple tabular form (therefore, it is recommended to avoid -I option, because it results in generation the summary section, which disturb the tabular order). Such files can be easily read by R environment by using for example the read.table function as follows:

```
treeCmpData<-read.table("C:\\Program
Files\\TreeCmp\\examples\\plain\\plain.trees.m.out", header = TRUE, sep = "\t")
```

In the example, the file to read “plain.trees.m.out” is placed in “C:\Program Files\TreeCmp\examples\plain” folder.

7. License

References

1. Bogdanowicz D, Giaro K: **Matching Split Distance for Unrooted Binary Phylogenetic Trees**. *IEEE/ACM Trans Comput Biol Bioinform* 2012, **9**: 150-160.
2. Bogdanowicz D, Giaro K: **Comparing Phylogenetic Trees by Matching Nodes Using the Transfer Distance between Partitions**. Submitted, 2014.
3. Bogdanowicz D, Giaro K., Wróbel B. **TreeCmp: comparison of trees in polynomial time.** *Evol. Bioinform.* 2012, in press.
4. Cardona G, Llabrés M, Rosselló F, Valiente G: **Nodal distances for rooted phylogenetic trees**, *J Math Biol* 2010 **61**:253-276.
5. Critchlow DE, Pearl DK, Qian C**: The Triples Distance for Rooted Bifurcating Phylogenetic Trees,** *Syst Biol* 1996, **45**: 323-334.
6. Estabrook GF, McMorris FR, Meacham CA: **Comparison of Undirected Phylogenetic Trees Based on Subtrees of Four Evolutionary Units**. *Syst Biol* 1985, **34**:193-200.
7. McKenzie A, Steel M: **Distributions of cherries for two models of trees**. *Math Biosci* 2000, **164**:81-92.
8. Robinson DF, Foulds LR: **Comparison of phylogenetic trees**. *Math Biosci* 1981, **53**:131-147.
9. Steel MA, Penny D: **Distributions of Tree Comparison Metrics – Some New Results**. *Syst Biol* 1993, **42**:126-141.
10. Semple C, Steel M: **Phylogenetics**, Oxford University Press 2003.