TreeCmp 1.0 – a tool for comparing phylogenetic trees using the Matching Split distance and other metrics - 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) **unrooted** phylogenetic trees.

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.

3. Output data format

All 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, OpenOffice.org). An output file consists of two sections. The first section contains formatted in rows values of distances in selected metrics (or aggregate values in the case of widow mode). The second section contains summary data computed based on all rows that appears in the first section.

4. 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 1.5 is required.

4.1. Directory structure

Directory			Description
bin			contains main jar file: **TreeCmp.jar** and lib folder with necessary open source libraries: pal-1.5.1.jar (http://www.cebl.auckland.ac.nz/pal-project/) and commons-cli-1.2.jar (http://commons.apache.org/cli/)
config			contains xml configuration file
doc			contains this manual
	examples		contains subdirectories with examples
		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
src			contains source code of this application

4.2. Command line syntax

Usage:

```
java -jar TreeCmp.jar -w <size>|-s|-m –d <metrics> -i <inputfile> -o
<outputfile>
```

Note that options order is important.

- 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.

- The metric option (`-d`). At least one and at most 4 metrics can be specified. Metrics should be separated by space character:
  - `ms` – the Matching Split distance [1],
  - `rf` – the Robinson-Foulds distance [2],
  - `pd` – the path difference distance [3],
  - `qt` – the quartet distance [4].

  Example: `-d ms rf`

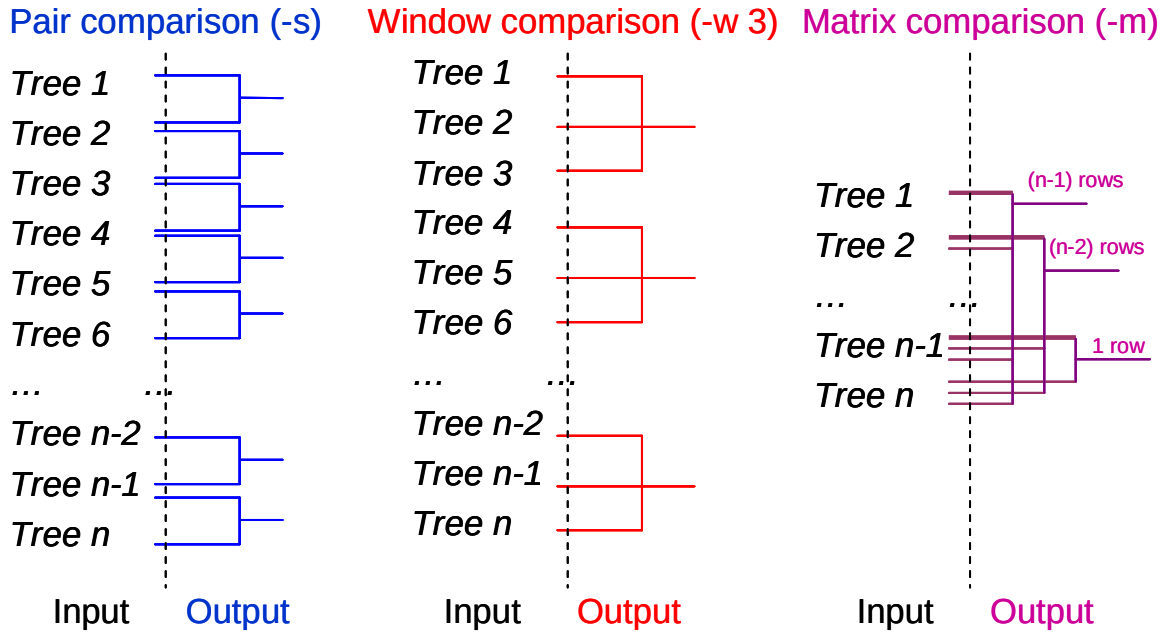
- IO options (both options should be specified):
  - `-i <inputfile>` – input data file with trees in the Newick format,
  - `-o <outputfile>` – output data file with the results of computations.

4.3. Types of analysis

There are three different types of available reports:

- overlapping pair comparison,
- window comparison,
- matrix comparison.

Details of the computation of these reports are explained in the picture below.



4.4. 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.

5. Example

Input file: `\doc\examples\beast\testBSP.newick`
Invocation: `java -jar TreeCmp.jar -w 3 -i testBSP.newick -o testBSP.newick_w_3.out`
Console output:

```
TreeCmp version 1.0-b179

Active options:
Type of the analysis: window comparison mode (-w) with window size: 3
Metrics:
  1. MatchingSplit (ms)
Input file: testBSP.newick
Output file: testBSP.newick_w_3.out
-----
2011-04-22 02:27:17: Start of scanning input file: testBSP.newick
2011-04-22 02:27:17: End of scanning input file: testBSP.newick
2011-04-22 02:27:17: 11 valid trees found in file: testBSP.newick
2011-04-22 02:27:17: Start of calculation...please wait...
2011-04-22 02:27:17: 0.00% completed...
2011-04-22 02:27:17: 10.00% completed...
2011-04-22 02:27:17: 20.00% completed...
2011-04-22 02:27:17: 30.00% completed...
2011-04-22 02:27:17: 40.00% completed...
2011-04-22 02:27:17: 50.00% completed...
2011-04-22 02:27:17: 60.00% completed...
2011-04-22 02:27:17: 70.00% completed...
2011-04-22 02:27:17: 80.00% completed...
2011-04-22 02:27:17: 90.00% completed...
2011-04-22 02:27:17: 100.00% completed.
2011-04-22 02:27:17: End of calculation.
2011-04-22 02:27:17: Total calculation time: 63 ms.
```

Output file testBSP.newick_w_3.out:

```
state  MatchingSplit (avg)      MatchingSplit (stddev)
1      60.0000      5.8878
2      10.6667      2.4944
3      14.0000      1.4142
4      9.0000 0.0000
---------
Summary:
Name   Avg    Std    Min    Max    Count
MatchingSplit      23.416666666666668 21.1979755322688   9.0    60.0   4
```

6. License

Copyright (C) 2011, Damian Bogdanowicz

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses/.

References

1. Bogdanowicz D, Giaro K: **Matching Split Distance for Unrooted Binary Phylogenetic Trees**. *IEEE/ACM Trans Comput Biol Bioinform*, 17 Feb. 2011. IEEE computer Society Digital Library. IEEE Computer Society, <http://doi.ieeecomputersociety.org/10.1109/TCBB.2011.38>
2. Robinson DF, Foulds LR: **Comparison of phylogenetic trees**. *Math Biosci* 1981, **53**:131-147.
3. Steel MA, Penny D: **Distributions of Tree Comparison Metrics -- Some New Results**. *Syst Biol* 1993, **42**:126-141.
4. Estabrook GF, McMorris FR, Meacham CA: **Comparison of Undirected Phylogenetic Trees Based on Subtrees of Four Evolutionary Units**. *Syst Biol* 1985, **34**:193-200.