HashRF: HOW TO Document
Version 1.0

Seung-Jin Sul and Tiffani L. Williams
Department of Computer Science
Texas A&M University
{sulsj, tlw}@cse.tamu.edu

04-20-2009

1 Introduction

This document is a brief description on how to run the algorithms for computing topological distances using a sample tree file consisting of 150 taxa and 10 trees. All example usages are based on Linux operating system, but it should be straightforward to run the software and experiments on Mac OS X and Windows using the cygwin environment. In the following description, the '>' symbol is used to represent the command-line prompt. We assume that the required files are located in the same directory in which the executable program was executed. The example file consists of 150 taxa and 10 trees. The user can use the tree files in the Newick format.

2 How to compile HashCS

If you want to compile the HashRF, download the hashrf-*version*.tgz file and compile using the below commands: To unzip and untar the file, type the following at the command prompt ('>')

```
> tar -zxvf hashrf-6.0.0.tgz
```

To configure and compile:

```
> cd hashrf-6.0.0
> ./configure
> make all
```

3 How to run HashRF

Trees are input in a tree file in standard nested-parenthesis notation, or Newick format.

To get a complete list of options for the program type

```
> ./hashrf -h
```

To compute RF distance among trees in a sample tree file:

```
> ./hashrf ./examples/150-taxa-10-trees.tre 10
```

or

```
> ./hashrf ./examples/150-taxa-10-trees.tre 0
```

If you set the "number of tree" argument to '0', HashRF counts the number of trees in the input file before computing RF distance.

The result matrix should be:

```
Robinson-Foulds distance (matrix format):
0 34 34 39 33 42 42 38 41 42
34 0 32 34 28 30 31 28 30 38
34 32 0 35 31 29 32 36 33 37
39 34 35 0 24 29 30 26 29 31
33 28 31 24 0 29 26 26 29 29
42 30 29 29 29 0 26 24 21 34
42 31 32 30 26 26 0 21 24 27
38 28 36 26 26 24 21 0 25 27
41 30 33 29 29 21 24 25 0 28
42 38 37 31 29 34 27 27 28 0
```

By default, the software prints the result RF distance matrix on screen. You can save the result in a specific file using "-o" flag.

```
> ./hashrf ./examples/150-taxa-10-trees.tre 10 -o result.rf
```

Using "-c" option, the probability of double collision can be specified. The default value is $1,000$ which means the probability is $O(\frac{1}{1,000})$. You can find the details in [1] [2].

```
> ./hashrf ./examples/150-taxa-10-trees.tre 10 -c 2000 -o result.rf
```

Three types of formats are supported in HashRF for displaying (saving) the result RF distance values.

- -p *list*, print RF distance in list format.
- -p *rate*, print RF distance rate in list format.
- -p *matrix*, print resulting distance in matrix format.

For example, the below command saves RF distance values in the file "result.rf" with list format.

```
> ./hashrf ./examples/150-taxa-10-trees.tre 10 -c 2000 -p list -o result.rf
```

To show the saved RF values in list format:

```
> cat result.rf
<0,0> 0
<0,1> 34
<0,2> 34
<0,3> 39
<0,4> 33
<0,5> 42
<0,6> 42
<0,7> 38
<0,8> 41
<0,9> 42
<1,0> 34
<1,1> 0
```

```
<1,2> 32
<1,3> 34
<1,4> 28
<1,5> 30
<1,6> 31
<1,7> 28
<1,8> 30
<1,9> 38
...
```

To compute weighted RF distance, use ”-w” option:

```
> ./hashrf ./examples/150-taxa-10-trees.tre 10 -w
```

The result matrix from the above command:

```
Robinson-Foulds distance (matrix format):
 0  0.802548 0.74228 0.806134 0.855208 0.953118 0.915377 0.927519 0.896039 0.868616
0.802548  0  0.866605 0.958495 0.998747 1.01795 0.929366 0.915059 0.926608 1.03013
0.74228 0.866605  0  0.862665 0.926458 0.926813 0.816611 0.953529 0.843838 0.887207
0.806134 0.958495 0.862665  0  0.708002 0.790275 0.823506 0.801888 0.715232 0.710572
0.855208 0.998747 0.926458 0.708002  0  0.787495 0.755494 0.828644 0.86654 0.678509
0.953118 1.01795 0.926813 0.790275 0.787495  0  0.778163 0.854551 0.773001 0.806544
0.915377 0.929366 0.816611 0.823506 0.755494 0.778163  0  0.770639 0.763618 0.783948
0.927519 0.915059 0.953529 0.801888 0.828644 0.854551 0.770639  0  0.728569 0.777803
0.896039 0.926608 0.843838 0.715232 0.86654 0.773001 0.763618 0.728569  0  0.788841
0.868616 1.03013 0.887207 0.710572 0.678509 0.806544 0.783948 0.777803 0.788841  0
```

References

[1] S.-J. Sul and T. L. Williams. A randomized algorithm for comparing sets of phylogenetic trees. In *Proc. Fifth Asia Pacific Bioinformatics Conference (APBC'07)*, pages 121–130, 2007.

[2] S.-J. Sul and T. L. Williams. An experimental analysis of robinson-foulds distance matrix algorithms. In *European Symposium of Algorithms (ESA'08)*, pages 793–804, 2008.