NEPAL: Phylogenetic Network Reconstruction Using the Maximum Parsimony and Maximum Likelihood Criteria

Developers The NEPAL (stands for NEtwork Parsimony And Likelihood) is a suite of algorithmic tools that is based on models and algorithms developed in a series of papers, mainly by Guohua Jin, Luay Nakhleh, Sagi Snir, and Tamir Tuller; see [5, 1, 2, 3, 4, 7]. Software was designed and implemented by Guohua Jin, Hyun Jung Park, and Luay Nakhleh.
Contact: Luay Nakhleh (`nakhleh@cs.rice.edu`)

Acknowledgments This work is supported in part by a National Science Foundation grant (CCF-0622037), a Department of Energy grant (DE-FG02-06ER25734), and grant R01LM009494 from the National Library of Medicine to Luay Nakhleh. The contents are solely the responsibility of the authors and do not necessarily represent the official views of the DOE, NSF, National Library of Medicine or the National Institutes of Health.

Description `NEPAL` is sequence-based tool for infering phylogenetic networks based on the maximum parsimony and maxmum likelihood criteria. It is used to identify horizontal gene (or partial gene) transfers between species. `NEPAL` reads in a species tree in Newick format or a network from `NEPAL` or `RIATA-HGT` output [6], and sequence data. It returns the maximum parsimony or maximum likelihood score of the input or generated trees or networks.

Usage `nepal [-c` *compmode*`] [-e` *numedges*`] [-i` *tree*`] [-a` *algorithm*`]`
`[-b` *blksize*`] [-d] [-h` *heuristic*`] [-l` *likelihood*`]`
`[-m` *matrix*`] [-n` *netsNepal*`] [-o` *output*`] [-p` *parallel*`]`
`[-r` *netsRiata*`] -s` *seqs* `[-t` *matrixType*`] [-M` *method*`]`

`-a` *algorithm* for maximum parsimony, 0: FPT (default), 1: improved FPT, 2: approximated FPT

`-b` *blksize*: block size (default: whole sequence length as a single block)

`-B` : compute bootstrap confidence value
`-c` *compmode*: 0: network generation; 1: MP or ML computation
`-d` deleting identical sites
`-e` *numedges*: the maximum number of edges added into the input tree
`-h` *heuristic*: 0: exhaust (default); 1: branch & bound; 2: hamming-distance
`-i` *tree*: an input file of species tree represented in Newick format
`-l` *likelihood* of network and tree: 0: (all, average) (default);
1: (all, ancestral); 2: (best, average); 3: (best, ancestral)
`-m` *matrix*: an input file for user-defined scoring matrix
`-n` set of networks: an input file of phylogenetic networks in nepal format
`-o` *output*: an output file with networks and scoring information
`-r` set of networks: an input file of phylogenetic networks in Riata format
`-s` *seqs*: an input file including information about sequences
`-t` *matrixType*: scoring matrix type, eg. pam120 or blosum65.
`-v` : verbose control: 0: print less information, 1: print more information.
`-M` *method*: 0: maximum parsimony (default), 1: maximum likelihood.

As an example, if we have the following species tree in `example.st`:

```
(T7,((T5,T6),(T1,T2),(T3,T4)));
```

Given the following sequence data set in `example.seq` where the first line of the file specifies the number of taxa, original sequence length after alignment, sequence type (0 for DNA sequence, 1 for AA sequence), number of exclusive regions followed by a list of exclusive regions specified by *startsite* − *endsite*. In this example, there are 7 taxa and the aligned DNA sequences are 23 site long in which the first two sites should be excluded in analysis.

```
7 23 0 1 1-2
T1 --TTCTGATGACAGCCCGAT
T2 CTTTCCGATGACAGCCTGAC
T3 CTTTCTGATGGCAGCCTGAC
T4 CTTTCCGATAGCAGCCTGAC
T5 --TTCCAATGACAGTCTGAC
T6 --TTCCGATGACAGCCTGAC
T7 --TTCCAATAGTAGTCTGAC
```

If we want to check with maximum parimony criteria all possible networks by adding a potential HGT into the original species tree, we can run `nepal` with the following command:

```
nepal -e 1 -B -h 1 -v 1 -i example.st -s example.seq -o example.out
```

Option -e gives an upper limit (1 in this example) of the number of HGT events in the run. The actual number of HGT events that are identified can be less based on the significance of the events. `NEPAL` generates all possible network and computes the maximum parsimony scores for all legal networks. It prints out the HGT events that generate networks with maximum parsimony scores, together with the their best maximum parsimony scores. The process of identifying the most significant HGT events and generating networks is controlled by a statistical model. `NEPAL` computes and uses p-value to determine the amount of reticulation required. For each inferred reticulation event, `NEPAL` also computes (with option `-B`) its bootstrapping support by a process of sequence sampling with replacement and recalculation of MP scores. At the end of entire search, the best maximum parsimony score is printed out. The following output shows an example of run using a branch and bound heuristic algorithm with bootstrapping.

```
(((T5,T6)_I2,(T1,T2)_I3,(T3,T4)_I4)_I1,T7)_I0
c l2(T7), l5(T5), l6(T6), l8(T1), l9(T2), l11(T3), l12(T4)
c e2(r1, v3), e3(v3, v4), e1(r1, l2), e4(v4, l5), e5(v4, l6), e6(v3, v7),
e7(v7, l8), e8(v7, l9), e9(v3, v10), e10(v10 , l11), e11(v10, l12)
     HGT1:  T7(1) --> T5(1)
     HGT1:  T5(1) --> T7(1)
     HGT1:  T4(1) --> T7(1)
Best Scores with 0 and 1 HGT: [ 16 14 ]
start bootstraping ....................................................
.........................................
BS support for network 1:  55
BS support for network 2:  51
BS support for network 3:  55
BS support for union(intersection) of best networks:  92(14)
p-value for adding 1 HGT: 0.347826
Best Maximum Parsimony Score:  14
```

NEPAL can also be used for computing maximum likelihood values of networks, as well as identify HGT under the maximum likelihood criteria.

References

[1] G. Jin, L. Nakhleh, S. Snir, and T. Tuller. Efficient parsimony-based methods for phylogenetic network reconstruction. *Bioinformatics*, 23:e123–e128, 2006. Proceedings of the European Conference on Computational Biology (ECCB 06).

[2] G. Jin, L. Nakhleh, S. Snir, and T. Tuller. Maximum likelihood of phylogenetic networks. *Bioinformatics*, 22(21):2604–2611, 2006.

[3] G. Jin, L. Nakhleh, S. Snir, and T. Tuller. Inferring phylogenetic networks by the maximum parsimony criterion: a case study. *Molecular Biology and Evolution*, 24(1):324–337, 2007.

[4] G. Jin, L. Nakhleh, S. Snir, and T. Tuller. A new linear-time heuristic algorithm for computing the parsimony score of phylogenetic networks: Theoretical bounds and empirical performance. In I. Mandoiu and A. Zelikovsky, editors, *Proceedings of the International Symposium on Bioinformatics Research and Applications*, pages 61–72, 2007. Lecture Notes in Bioinformatics, #4463.

[5] L. Nakhleh, G. Jin, F. Zhao, and J. Mellor-Crummey. Reconstructing phylogenetic networks using maximum parsimony. In *Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference (CSB2005)*, pages 93–102, 2005.

[6] L. Nakhleh, D. Ruths, and L.S. Wang. RIATA-HGT: A fast and accurate heuristic for reconstrucing horizontal gene transfer. In L. Wang, editor, *Proceedings of the Eleventh International Computing and Combinatorics Conference (COCOON 05)*, pages 84–93, 2005. LNCS #3595.

[7] H.J. Park, G. Jin, and L. Nakhleh. On the significance of phylogenetic networks inferred by maximum parsimony. Under review, 2009.