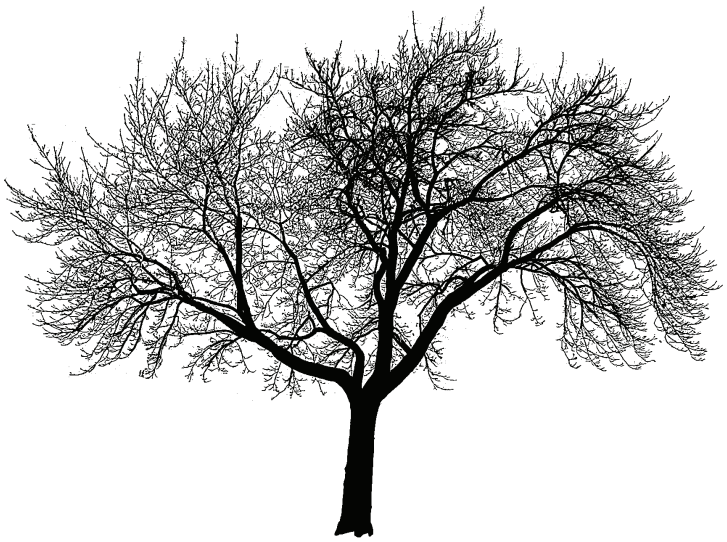
User Guide of ESPRIT-Tree

Yunpeng Cai and Yijun Sun*

Interdisciplinary Center for Biotechnology Research
University of Florida, Gainesville, FL 32610-3622



*Please address all correspondence to: Dr. Yijun Sun, Interdisciplinary Center for Biotechnology Research, University of Florida, P.O. Box 103622, Gainesville, FL 32610-3622, USA. E-mail: sunyijun@biotech.ufl.edu.

1 Introduction

ESPRIT-Tree is a computational algorithm that allows researchers to perform taxonomy independent analysis of millions of 16S rRNA tag sequences efficiently on their desktop computers. It consists of three modules: (1) removes low quality reads using various criteria, (2) computes pairwise distances of reads and groups them into OTUs at different dissimilarity levels, and (3) performs statistical inference to estimate species richness. The second module is computationally the most expensive part and is described in detail in [1]. A benchmark study was performed that showed that ESPRIT-Tree has a quasilinear time and space complexity comparable to greedy heuristic clustering algorithms, and achieves a similar accuracy to the standard hierarchical clustering algorithm. The software is freely available at `http://plaza.ufl.edu/sunyijun/ES-Tree.htm`. If you have any questions and comments, please feel free to contact Dr. Yijun Sun at `sunyijun@biotech.ufl.edu`.

Throughout the manual, parameters in angle brackets <> are mandatory and those in square brackets [ ] are optional.

2 Installation and Quick Start

The software supports four different platforms, including Windows, Mac, 32-bit and 64-bit Linux. Download the codes into your designated directory and add it to the system execution path. In order to run ESPRIT-Tree, users should have `perl` installed in their computers. For Linux, `perl` is usually pre-installed, and for Windows `perl` can be downloaded at `http://www.perl.org`.

To run ESPRIT-Tree with the default parameters, Windows users type the following command:

```
>> esprit-tree <input> [output-prefix]
```

and for Linux users the command is:

```
>> esprit-tree.sh <input> [output-prefix]
```

where <`input`> is a FASTA file containing input sequences, and `[output-prefix]` specifies the path of the directory where you want to put result files. If `[output-prefix]` is not given, result files are saved in the same directory as input files.

ESPRIT-Tree generates three output files. `[output-prefix].OTU` reports the number of OTUs obtained at each distance level. `[output-prefix].org.Clusters` provides the

detailed information of how input sequences are clustered at different distance levels. In order to reduce the size of the file, each sequence is represented by a number, instead of the original sequence ID. It should be noted that the number 0 refers to the first sequence in the input file. `[output-prefix].Cluster_List` contains the information of OTUs that can be used to perform statistical inference.

To perform statistical inference, simply type:

```
>> do_stat <output-prefix>.Cluster_List
```

The output files report the ACE and Chao 1 estimates, and the results of a rarefaction analysis.

The following command generates a set of FASTA files. Each file contains the sequences in each cluster at a distance level:

```
>> parsecluster <input> <output-prefix>.org.Cluster [level_down] [level_up]
```

where `<input>` is a FASTA file containing input sequences, and parameters `level_down` and `level_up` specifies the smallest and largest distance levels, respectively. The clustering information allows users to compute ecological metrics, to derive a consensus sequence of each cluster, and to align the sequences in rarely occurred OTUs against a database, which may lead to the identification of novel pathogenic and uncultured microbes.

3 Advanced Settings

The default settings work well in most cases. We provide users with options allowing them to use user-specified parameters to optimize the performance. Users need to execute each command manually following the guide provided below, or organize all commands in an executable script.

3.1 Preprocessing

preproc

Remove low-quality reads and merge duplicated reads.

```
preproc [-e] [-p primer_file] [-m mis_allowed] [-w] [-v var_allowed] [-l minlen]
[-u maxlen] <input.fas> [output] [freq_output]
```

`-e`: if present, only identical sequences are merged.
The default setting is to merge sequences with a zero genetic distance by ignoring end gaps.

`primer_file`: FASTA file containing PCR primers.

`mis_allowed`: maximum allowed mismatches with primer sequences, default 1.

`-w`: remove sequences containing ambiguous characters.
The default setting is to remove ambiguous characters but keep sequences.

`var_allowed`: maximum allowed deviations from the average length, default 1.0 (i.e., one standard deviation).

`minlen`: remove sequences with lengths < minlen.

`maxlen`: remove sequences with lengths > maxlen.

`input`: FASTA file containing original sequences.

`output`: FASTA file containing trimmed sequences, default name <`input`>`_Clean.fas`.

`freq_output`: sequence frequency output, default name <`input`>`_Clean.frq`.

3.2 Performing Clustering

pbpcluster

Perform online-learning based hierarchical clustering

```
pbpcluster [-l lower] [-s intv] [-u upper] [-k klen] [-g gap_open] [-e gap_ext]
[-m meml] [-r errorrate] [-o output] [-f freqfile] <seqfile> [kfile]
```

`output`: path and file name of output files.

`seqfile`: FASTA file generated by `preproc`.

`freqfile`: frequency file generated by `preproc`.
If not given, it is assumed that each sequence appears only once.

`kfile`: *k*mer configuration file.

`lower`: lowest distance level for clustering, default 0.01.

`intv`: interval of distance levels, default 0.01.

`upper`: top distance level for clustering, default 0.10.

`klen`: *k*mer length, default 6.

`gap_open`: gap open penalty of the NW algorithm, default -10 .

`gap_ext`: gap extension penalty of the NW algorithm, default -0.5 .

`meml`: maximum allowed memory in gigabyte, default 1.0.

`errorrate`: base-wise error rate for sequencing error correction.
default 0 (no correction).

ESPRIT-Tree uses $k$mer statistics to reduce the number of sequence comparisons. The correlation information between $k$mer and genetic distances is saved in configuration files (`*.krate`). The $k$mer lengths of 5 and 6 are used. Given an input dataset, ESPRIT-Tree automatically selects a proper $k$mer configuration file based on the average length of input sequences. Specifically, `l50*.krate` is used for input sequences of length $50 \sim 100$, `l100*.krate` for length $100 \sim 200$, `l200*.krate` for length $200 \sim 400$, `l400*.krate` for length $400 \sim 800$, and `ful*.krate` for full-length 16S rRNA sequences. Users can use a different configure file by changing parameter `[kfile]`. We provide an auxiliary program `findkrate` in the software package that allows users to generate customized configuration files (see Section 3.4 for details).

ESPRIT-Tree uses spare memory to cache pairwise distances. If the memory is limited, only a small cache can be created. Parameter `[-m meml]` specifies the amount of memory that is available to the program. Due to many unpredictable factors, such as system page swapping, memory fragments and inaccurate estimation of memory usage, the amount of memory requested by the program may be larger than that is actually available, which results in an "Out of Memory" error. In this case, you can reduce the value of `meml` to suppress the error message.

3.3 Mapping Clustering Result back to Original Data

`pbpcluster` uses a trimmed sequence dataset. The command `invmap.pl` maps the clustering result generated by `pbpcluster` back to the original FASTA file:

```
>> perl invmap.pl sequence.Cluster sequence_Clean.map sequence.org.Cluster
```

where `sequence_Clean.map` is generated by `preproc`. The result is saved in `sequence.org.Cluster`, where 0 refers to the first sequence in the original FASTA file `sequence.fas`.

3.4 Generating $k$mer Configuration Files

findkrate

`findkrate` generates customized $k$mer configuration files from a given dataset and parameters:

`findkrate [-c seqnum] [-k klen] [-g gap_open] [-e gap_extend] [-s step] <input> <output>`

`seqnum`: number of sequences, default 1000.

`klen`: $k$mer length, default 6.

`gap_open`: gap open penalty of the NW algorithm, default -10 .

`gap_extend`: gap extension penalty of the NW algorithm, default -0.5 .

`step`: step size of the distance levels. [0.01-0.1], default 0.03.

4 Examples

We have used ESPRIT-Tree to process several large-scale 16S rRNA datasets. Table 1 summarizes the datasets and the CPU times. All experiments were performed on a desktop computer with Intel Xeon E5430 2.66GHZ CPU and 16G memory.

Table 1: CPU times of ESPRIT-Tree performed on three large datasets.

Data	Region	# Seqs	# Non-redundant[a]	Ave Len	Time	Memory
Seawater data [2] [b]	V6	689K	28K	62	9m	<1G
Human gut data [3] [c]	V2	1.1M	470K	233	11h8m	<16G
Mice gut data [4] [d]	V2	1.6M	321K	232	8h12m	<16G

[a] sequences with a zero genetic distance are considered redundant.

[b] dataset included in the software package

[c] http://gordonlab.wustl.edu/NatureTwins_2008/V2.fasta.gz

[d] http://gordonlab.wustl.edu/TurnbaughSE_10_09/HMiceV216S.fna.gz

References

[1] Cai Y, Sun Y (2010) ESPRIT-Tree: taxonomy independent analysis of millions of 16S rRNA pyrosequences in quasilinear computational time. Technical Report.

[2] Huber JA, Welch DBM, Morrison HG, Huse SM, Neal PR, Butterfield DA, Sogin ML. (2007) Microbial population structures in the deep marine biosphere. *Science* **318**(5): 97-100.

[3] Turnbaugh PJ, Hamady M, Yatsunenko T, Cantarel BL, Duncan A, Ley RE, Sogin ML, Jones WJ, Roe BA, Affourtit JP, et al. (2009) A core gut microbiome in obese and lean twins. *Nature* **457**: 480-485.

[4] Turnbaugh PJ, Ridaura VK, Faith JJ, Rey FE, Knight R, Gordon JI. (2009) The effect of diet on the human gut microbiome: a metagenomic analysis in humanized gnotobiotic mice. *Sci Transl Med* **6**:ra14.