MetaPhyler Usage Manual

Bo Liu
boliu@umiacs.umd.edu

March 13, 2012

Contents

1 What is MetaPhyler **1**

2 Installation **1**

3 Quick Start **2**
3.1 Taxonomic profiling for metagenomic sequences 2
3.2 Build your own classifier, and perform classification 2
3.3 Which Metaphyler mode: blastn or blastx? 3
3.4 Run Metaphyler pipeline step by step 3

4 Program Usage **4**
4.1 simuReads 4
4.2 metaphylerTrain 4
4.3 metaphylerClassify 4
4.4 combine 5
4.5 taxprof 5

5 Citation **5**

1 What is MetaPhyler

MetaPhyler is a taxonomic classifier for metagenomic (or anonymous) DNA or protein sequences. Given a set of reference sequences and their taxonomic labels, MetaPhyler can train itself using these information, and classify anonymous query sequences into the taxonomic labels. We have pre-trained MetaPhyler on a set of phylogenetic marker genes, and these pre-trained models can be used to classify metagenomic sequences and estimate the taxonomic profile.

2 Installation

Requirements: g++; Perl; NCBI BLAST.
Installation has been tested on RHEL Server r5.5, with gcc v4.5.2 and perl v5.8.8.

Uncompress the software package:
`tar xzvf package-name`

Then enter into the directory and install Metaphyler:
`./installMetaphyler.pl`

3 Quick Start

Here we show how to perform common tasks using test datasets in folder 'test':
(1) test.ref.dna: reference DNA sequences in FASTA format.
(2) test.ref.protein: reference protein sequences in FASTA format. Same as in (1).
(3) test.ref.taxonomy: taxonomy labels for reference sequences.
(4) test.query.dna: DNA sequences to be classified.

3.1 Taxonomic profiling for metagenomic sequences

Here, we have a set of metagenomic DNA sequences, which could be reads or predicted ORFs, and you want to know the taxonomic composition. Metaphyler tries to identify the phylogenetic marker genes present in the sample, classify them and compute the taxonomy profile. The classifiers for these phylogenetic marker genes have already been trained. Suppose your reads are in file test.query.dna in folder test; outputs are stored in files with prefix 'test.blastn' or 'test.blastx'; BLAST was run with 1 thread. For Illumina reads with average length 100bp, we recommend:

```
./runMetaphyler.pl ./test/test.query.dna blastn test.blastn 1
```

For longer sequences (e.g., 454 or Sanger), we recommend:

```
./runMetaphyler.pl ./test/test.query.dna blastx test.blastx 1
```

Best performance can be achieved by running both blastn and blastx models, and combine the classifications:

```
./combine test.blastn.classification test.blastx.classification
```

3.2 Build your own classifier, and perform classification

Suppose test.query.dna contains DNA sequences with length range from 200bp to 400bp. Then we build classifier by simulating reads from test.ref.dna, and align them to test.ref.protein using blastx. The following command specifies that we simulate 200bp, 300bp and 400bp DNA fragments; outputs are stored in files with prefix 'test'; run blastx with single thread.

```
./buildMetaphyler.pl norm ./test/test.ref.dna ./test/test.ref.protein
200,300,400 ./test/test.ref.taxonomy blastx test 1
```

Run classification:

```
./runClassifier.pl ./test/test.query.dna ./test/test.ref.protein
./test/test.ref.taxonomy test.blastx.classifier blastx test 1
```

Suppose you only have the DNA sequences test.query.dna, then the training and classification can only be performed with blastn:

```
./buildMetaphyler.pl norm ./test/test.ref.dna ./test/test.ref.dna
200,300,400 ./test/test.ref.taxonomy blastn test 1

./runClassifier.pl ./test/test.query.dna ./test/test.ref.dna
./test/test.ref.taxonomy test.blastn.classifier blastn test 1
```

Usually blastn works better for short reads (e.g., 100bp Illumina reads); blastx works better for sequences longer than 100bp. But you are welcome to try both approaches on you dataset, and combine the classification results.

3.3 Which Metaphyler mode: blastn or blastx?

Our suggestion is that try both, and then merge the classification results: for each query read, pick the classification that has higher confidence score. In addition, we have some empirical guidlines:
(1) for short reads (e.g., 100bp Illumina reads), use blastn model; otherwise, use blastx.
(2) for finding remote homologies, and classify query sequences that are not very close to your reference genes, use blastx.
(3) if your reference sequences are not protein genes (e.g., 16S rRNA), which means they do not have amino acid sequences, then use blastn.
(4) if your sequences have many deletions or insertions (e.g., 10 insertions or deletions in a 300bp read), then use blastn.

3.4 Run Metaphyler pipeline step by step

If you want to have full control of the program, e.g., BLAST usually takes long time to run, and you want to parallelize it in your computer clusters, in the following we show how to run the whole pipeline step by step.

(1) Simulate reads (300bp, 30bp step size):
```
./simuReads 300 30 ./test/test.ref.dna > test.300.30.dna
```

(2) Compile blast database:
```
formatdb -p T -i ./test/test.ref.protein
```

(3) Run blastx:
```
blastall -p blastx -m8 -b1000 -e1e-3 -i test.300.30.dna
-d ./test/test.ref.protein > test.300.30.blastx
```
You can parallelize blast in various ways. Also you can tune evalue parameter depending on your sequence length.

(4) Compute classifier:
```
./metaphylerTrain norm ./test/test.ref.taxonomy ./test/test.ref.protein
test.300.30.blastx 300 blastx > test.300.blastx.classifier
```
You can repeat above steps (1, 3 and 4) for other lengths, e.g., 200 and 400, and put all classifiers into one file, say test.blastx.classifier.

(5) Align query sequences to references:
```
blastall -p blastx -m8 -b1 -e1e-2 -i ./test/test.query.dna
-d ./test/test.ref.protein > test.query.blastx
```
Note that you only need 1 blast hit for each query sequence, so use option -b1. You can adjust the evalue parameter depending your sequence length, and parallelize blast in various ways.

(6) Classify query reads:
```
./metaphylerClassify test.blastx.classifier ./test/test.ref.taxonomy
test.query.blastx > test.classification
```
Note that classifiers based on phylogenetic marker genes for taxonomic profiling have been computed: markers.blastn.classifier, markers.blastx.classifier and markers.taxonomy in folder 'markers'. In this case, you only need to run step 5 and 6 for your own metagenomic sequences.

4 Program Usage

4.1 simuReads

Step size 30bp is recommended. The smaller the step size, the more accurate. But at the same time, it takes longer time to train the classifier in the next few steps.

```
Usage:
        ./simuReads <length> <step size> <FASTA file>

Options:
        <length>     length of reads to be simulated.
        <step size>  distance between two simulated reads.
```

4.2 metaphylerTrain

If the the taxonomy is well defined, and there are sampling biases between the clusters in the taxonomy, then normalization is recommended. For example, in the NCBI taxonomy, some species (Escherichia coli) have much more genomes sequenced that others do. Option "norm" is used to normalize against these biases.

```
Usage:
   ./metaphylerTrain <norm|unnorm> <taxonomy> <ref seq> <BLAST file> <length> <BLAST>

Options:
   <norm|unnorm>   Perform normalization (norm) or not (unnorm).
   <taxonomy>      Taxonomy labels of reference sequences in the BLAST file.
   <ref seq>       Reference sequence FASTA file.
   <BLAST file>    BLAST alignment between simulated reads and reference sequences.
                   All simulated reads come from reference sequences,
                   and are named as follows:
                   If n reads come from A, then their IDs are A_0, A_1,..., A_n-1.
   <length>        Length of simulated reads.
   <BLAST>         BLASTN, BLASTP, BLASTX or TBLASTX.
```

4.3 metaphylerClassify

To classify the reads in the blast file, you may have trained several classifiers through simulation for different sequence lengths. To use all these classifiers, you can simply concatenate them into one file, or you can list them one by one in the program arguments.

```
Usage:
   ./metaphylerClassify <classifiers> <taxonomy file> <BLAST file>

Options:
 <scores file>   Output from program blast2TaxScores.
                 If there are multiple files, separate them with comma
                 (e.g., fileA,fileB). You can also merge them into one file.
 <taxonomy file> Taxonomy labels of reference sequences
                 in the BLAST file.
 <BLAST file>    BLAST alignment between query reads and reference sequences.
```

4.4 combine

For a set of query reads, you may have several classifications. For example, one from blastn classifiers and one from blastx classifiers. To combine them, simply list the files as the argument of this program.

```
Usage:
        ./combine <classification 1> <classification 2> ...

Options:
        <classification> Result file from program metaphylerClassify.
```

4.5 taxprof

Metaphyler has already trained classifiers for phylogenetic marker genes. After classification of metagenomic sequences, we want to know the taxonomy profile. This program summarizes the classification information, and reports the taxonomy profiles at different taxonomic levels.

```
Usage:
        ./taxprof <conf. cutoff> <classification> <prefix> <taxonomy names>

Options:
        <conf. cutoff>   Cutoff for confidence score.
                         Recommendation: 0.9.
        <classification> Result file from program metaphylerClassify.
        <prefix>         Output files prefix.
        <taxonomy names> File: 1st column, taxonomy ID; 2nd, name.
                         If omitted, output will just use taxonomy IDs.
Output files:
        prefix.<genus|family|order|class|phylum>.taxprof.
                         Taxonomy profiles at each level.
```

5 Citation

Please cite the following paper as the reference to this program.

Liu B., Gibbons T, Ghodsi M, Treangen T and Pop M(2010). Accurate and fast estimation of taxonomic profiles from metagenomic shotgun sequences. BMC Genomics 2011, 12(Suppl 2):S4.