MetaGen User Manual

Xin Xing

Contents

0.1 Overview

MetaGen a statistically based algorithm to simultaneously identify microbial species and estimate their abundances in multiple metagenomic samples without using any reference genome. Since MetaGen solely uses the cross-sample abundance patterns for binning, we recommend that the number of samples is larger than 10 samples or 2% of total number of species as a practical guide.

0.2 Installation

0.2.1 Dependencies

- Assembly software:
  - Ray Assembler($\geq v2.3.1$)
  - or MegaHIT Assembler($\geq v1.1$)
- Dependent software for extracting the reads count mapping matrix:
  - Bowtie2($\geq v2.2.4$)
  - Samtools($\geq v1.3$)
- R and R packages for implementing the statistical binning algorithm:
  - R ($\geq v3.2.1$)
  - Rcpp ($\geq v0.12.5$)
  - MASS ($\geq v7.3-45$)
  - mixtools ($\geq v1.0.4$)
  - doParallel ($\geq v1.0.10$)
  - foreach ($\geq v4.1.3$)
  - seqinr ($\geq v3.2-0$)
  - getopt ($\geq v1.20.0$)

Once you run R in command line, you can run the following R code to install all R packages:

```
packages <- c("Rcpp", "MASS", "mixtools", "doParallel", "foreach",
"seqinr","getopt")
for(i in 1:length(packages)){
        if(packages[i] %in% installed.packages()[, "Package"]){
    next
    }else{
        install.packages(packages[i], dependencies=TRUE)
    }
}
```

0.2.2 Installation

The installation is for Linux computer or computer cluster. MetaGen is tested on a Linux computer with Ubuntu Server 16.04 and a computer cluster with Red Hat Enterprise release 6.7. All the dependencies can be installed following the instruction in the linked address. Once all dependencies are properly installed, you need to download MetaGen, for example MetaGen is downloaded in the following directory:

```
/home/username/metagen-v1.0.1/
```

The scripts for extracting read counts mapping matrix(RCMM) are located in the directory:

```
/home/username/metagen-v1.0.1/scripts
```

The R function for implementing the main statistical binning algorithm is located in the directory:

```
/home/username/metagen-v1.0.1/R
```

Other source codes written by c++ to accelerate the computation are located in the directory:

```
/home/username/metagen-v1.0.1/src
```

Run the following code to store MetaGen path to bash variable:

```
metagen=/username/metagen-v1.0.1/
```

0.3 Usage

Add the MetaGen's installation directory, the path of data set and the working directory to bash variables:

```
metagen=/home/username/metagen_v1.0.1
metagen_data=/home/username/test_data
metagen_work_dir=/home/username/example
```

0.3.1 Pooled assembly for multiple DNA samples

The first step to begin the analysis is to assemble the reads of multiple samples.

```
cd $metagen_data
mpiexec -n 10 Ray -k 31 -detect-sequence-files ./ -o \
$metagen_work_dir/ray
```

The second command calls Ray assembler with "-n" option to use 10 CPU cores(this number is based on your computer resources), "-k" option specifies the length of k-mers(longer length requires more memory), "-detect-sequence-files" option can automatically detect both paired-end reads and single-end reads, "-o" option specifies the output directory.

If you are prefer to use MegaHIT, see https://github.com/voutcn/megahit for detailed instruction. Then copy the assembled contigs to the folder "contigs".

```
mkdir $metagen_work_dir/contigs
cp $metagen_work_dir/ray/Contigs.fasta $metagen_work_dir/contigs
```

After assembly, you need to build a bowtie2 index for the assembled contigs.

```
cd $metagen_work_dir/contigs
bowtie2-build Contigs.fasta ./contigs-ref
```

0.3.2 Extracting the read counts mapping matrix

Based on the bowtie2 index, you can extract the read counts mapping matrix(RCMM) through aligning the original reads back to assembled contigs. The usage of the scripts to extract RCMM are the following:

```
bash $metagen/scripts/bowtie2-align.sh \
[options] <ref> <outdir> <reads-dir> <reads-name>

[options]:
-h To show help documentation
-s single-end reads
-p paired-end reads
-a fasta files
-q fastq files(It is recommended to convert fastq to fasta and use -a option)

[input arguments]:
<ref>: The reference name of the bowtie2 index.
<outdir>: The output directory for the alignment result.
<reads-dir>: The directory of original reads.
<sample-name>: The sample name, for paired-ends reads
              sample-name_1.fastq sample-name_2.fastq,
              for single end reads sample-name.fastq.
```

Extract the read counts mapping matrix and combine the read counts for each sample into a matrix:

```
bash MetaGen/scripts/combine-counts.sh \
[options] <ref> <outdir> <reads-dir> <reads-name>

[options]:
-h To show help documentation
-s single-end reads
-p paired-end reads

[input argument]:
<work-dir>: Specify the working directory.

[output]:
$metagen_work_dir/output/count-map.tsv:
  The extracted read counts mapping matrix.
```

0.3.3 Statistical binning algorithm

Install all dependent R packages and run the following R script for statistical binning algorithm:

```
Rscript MetaGen/R/metagen.R [options]
[options]:
--metagen_path -m Specify the MetaGen's installation
```

```
  directory.
--work_dir -w Specify the working directory of current
  data set.

[optional options:]
--help -h Show help documentation.
--num_threads -n Specify the number of CPU cores used
  for parallel computing. When there is a large number of
  contigs, it is recommended to set multiple CPU cores to
  accelerate the computation. The default number is 1.
--bic_min -i Specify the minimum number of clusters. The
  default is 2.
--bic_max -a Specify the maximum number of clusters. The
  default is 0, which will let the algorithm sets the maximum
  number of cluster automatically.
--bic_step -s Specify the increment of the number of
  clusters from bic_min to bic_max. The default is 1.
--thred -t Specify the threshold for setting the initial
  value. It is recommended to set this number smaller(0.01
  -0.1) when the number of samples is less than $10$ and larger
  (0.1-0.2) when the number of samples is larger than $10$. The
  default value is set to 0.1.
--initial_per -p Specify how many percent contigs are used
  to set the initial value of the algorithm. The default
  value is 1, which means that all the contigs is used to
  find initial value of the algorithm. The number can be
  set to a smaller one, when there are a very large number
  of contigs.
--ctg_len_trim -l Specify the minimum contig length,
  contigs shorter than this value will not be included.
  Default is 500.
--plot-bic -p If the value is "T", output the plot of BIC scores.
  The default is "F".
-o The value is "1" for the simple metagenomic community.
   The value is "2" for the complex metagenomic community.

[output]:
$metagen_work_dir/output/segs.txt:
  The binning results for each contigs in a table with
  two columns. The first column lists the names of contigs.
  The second column lists the cluster ID for each contig.
$metagen_work_dir/output/scaled_relative_abundance:
  The scaled relative abundance matrix with column sum equals to 1.
  The first row specifies the sample names.
$metagen_work_dir/output/relative_abundance.txt
```

```
The relative abundance matrix with first row specifies
the sample names.
```

0.4 Example

In this section, a small example is presented to illustrate how to use MetaGen to analyze metagenomic data set. A test data is available to download through the repository: Test-Data. It is recommended to run the following examples on a computer with large memory. If you are running the example on a local computer, please make sure that the RAM of computer is larger than 8Gb.

Set MetaGen's installation directory, the working directory and the path of test data set to bash variables:

```
metagen=/home/username/metagen_v1.0.1
metagen_data=/home/username/Test-Data
metagen_work_dir=/home/username/example
```

Run Ray assembler for pooled assembly:

```
cd $metagen_data
mpiexec -n 10 Ray -k 31 -detect-sequence-files ./ -o \
$metagen_work_dir/ray
```

Build the bowtie2 index for the assembled contigs:

```
mkdir $metagen_work_dir/contigs
cp $metagen_work_dir/ray/Contigs.fasta $metagen_work_dir/contigs
cd $metagen_work_dir/contigs
bowtie2-build Contigs.fasta ./contigs-ref
```

Align the reads of each sample to the bowtie2 index. Here we use the "xargs" to run alignment in parallel. You can also set the "-P" option of "xargs" to 1, if you do not prefer the parallel computation. For paired-end reads, you need to change the "-s" option of "$metagen/bowtie2-align.sh" to "-p".

```
cd ../
chmod +x $metagen/script/bowtie2-align.sh
ls $metagen_data/*.fasta | \
gawk '{gsub(/.*[/]|.fasta/, "", $0)} 1' | \
xargs -P 6 -n 1 $metagen/script/bowtie2-align.sh -s -a \
$metagen_work_dir/contigs/contigs-ref \
```

```
$metagen_work_dir/map $metagen_data
```

Extract the read counts mapping matrix from the alignment results:

```
bash $metagen/script/combine-counts.sh -s $metagen_work_dir
```

Extract the number of reads for each sample using "-s" for single-end reads and "-p" for paired-end reads.

```
bash $metagen/script/sum-reads.sh -s $metagen_data $metagen_work_dir
```

Run the main statistical algorithm of MetaGen for binning and simultaneously estimating the relative abundance:

```
Rscript $metagen/R/metagen.R -m $metagen -w $metagen_work_dir
```

Check the binning result and the estimated scaled relative abundance matrix and relative abundance matrix in:

```
[output]:
$metagen_work_dir/output/segs.txt:
  The binning results for each contigs in a table with
  two columns. The first column lists the names of contigs.
  The second column lists the cluster ID for each contig.
$metagen_work_dir/output/scaled_relative_abundance:
  The scaled relative abundance matrix with column sum equals to 1.
  The first row specifies the sample names.
$metagen_work_dir/output/relative_abundance.txt
  The relative abundance matrix with first row specifies
  the sample names.
```