Tutorial

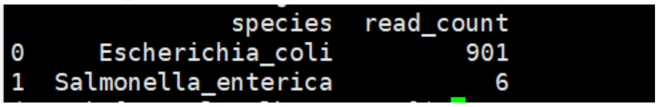
Step 1.

In the folder ‘Example_Dir/input/read/’ there is a demo mNGS dataset, which comprises the paired-end fastq files ‘test.1.fq’ and ‘test.2.fq’. To explore which bacterial pathogens have reads in this dataset, you can run the ‘species’ module:

```
$ python MIST.py species --threads 8 --pair_1 Example_Dir/input/read/test.1.fq --
pair_2 Example_Dir/input/read/test.2.fq --database Pre-built-pangenome/ --output
Example_Dir/output/
```

where ‘Pre-built-pangenome’ is the folder for the bowtie2-indexed database that stores the pan-genomes of common bacterial pathogens.

In the output folder ‘Example_Dir/output/_MIST_species/’ the result file ‘species_count.txt’ reveals that in the mNGS dataset there are 872 E. coli reads and a few reads from other species. You may assume that E. coli is the causative pathogen and choose E. coli for further strain-level typing accordingly. The result file ‘_MIST.Escherichia_coli.fq’ stores the E. coli reads retrieved from the mNGS dataset.



```
              species  read_count
0    Escherichia_coli         901
1  Salmonella_enterica          6
```

Step 2.

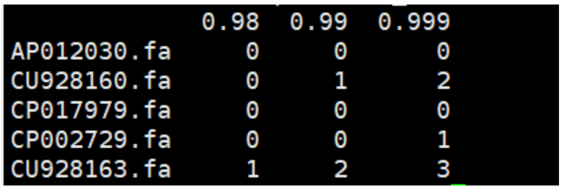
For strain-level typing of E. coli, you need to prepare a customized E. coli database; or you can download the pre-built database available at https://sourceforge.net/projects/mist-1-1/. Note that the size of the pre-built database is quite huge; be patient if you would like to download it.

Assume you choose to build the database by yourself, prepare your reference genomes in fasta format (e.g. downloaded from the NCBI Genbank) and place them in a folder. In the folder ‘Example_Dir/input/ref_dir/’ there are five demo E. coli genomes available. Run the ‘cluster’ module:

```
$ python MIST.py cluster --threads 8 --refdir Example_Dir/input/ref_dir/ --cutoff
0.98,0.99,0.999 --output Example_Dir/output/
```

Where the value ‘0.98,0.99,0.999’ for the -s option means that the input reference genomes will be clustered according to the ANI values of 0.98, 0.99, and 0.999, respectively.

When you open the output file ‘Example_Dir/output/_MIST_ref_cluster.csv’ in Excel, you can see that, at the 98% ANI level, the genome AP012030.fa, CU928160.fa, CP017979.fa and CP002729.fa belong to the same cluster; but at the 99% level, CU928160.fa splits and forms its independent cluster.



```
             0.98  0.99  0.999
AP012030.fa     0     0      0
CU928160.fa     0     1      2
CP017979.fa     0     0      0
CP002729.fa     0     0      1
CU928163.fa     1     2      3
```

Step 3.

As MIST calls Bowtie2 for mapping reads, now you need to build the bowtie2-index files for the 5 E. coli reference genomes by using the "index" module.

```
$ python MIST.py index --refdir Example_Dir/input/ref_dir/ --output
Example_Dir/output/
```

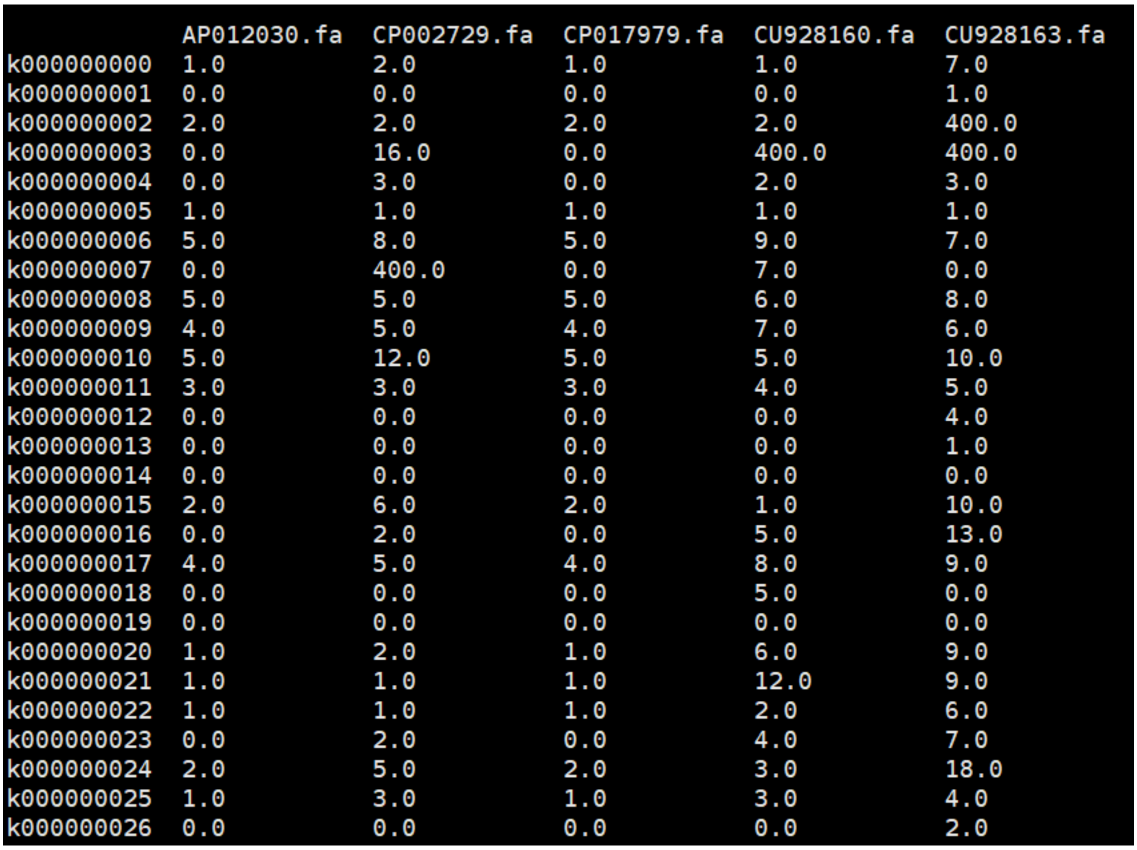
In the output folder "Example_Dir/output/_MIST_index/" there are 5 subfolders, which contains the index files for each reference genome.

Step 4.

Next you can align the E. coli mNGS reads against the E. coli reference genomes by using the "map" module:

```
$  python MIST.py map --threads 8 --indexpath Example_Dir/output/_MIST_index/ --
pair_1 Example_Dir/input/read/test.1.fq --pair_2 Example_Dir/input/read/test.2.fq --
read_length 200 --output Example_Dir/output/
```

Where the input folder 'Example_Dir/output/_MIST_index/' that stores the bowtie2-index files is obtained from Step 3; the input fastq files 'test.1.fq' and 'test.1.fq' are your paired-end mNGS dataset. Note that the length "-l" should be 200 if your inputs are 2*100bp paired-end sequencing files. Alternatively, you can input the single-end '_MIST.Escherichia_coli.fq' (obtained from Step 1) by using the -U option. In the output folder 'Example_Dir/output/' the result file '_MIST_map_Mismatch_matrix.csv' is a matrix, which stores the mismatch values of each read against each reference genome.



```
            AP012030.fa  CP002729.fa  CP017979.fa  CU928160.fa  CU928163.fa
k000000000  1.0          2.0          1.0          1.0          7.0
k000000001  0.0          0.0          0.0          0.0          1.0
k000000002  2.0          2.0          2.0          2.0          400.0
k000000003  0.0          16.0         0.0          400.0        400.0
k000000004  0.0          3.0          0.0          2.0          3.0
k000000005  1.0          1.0          1.0          1.0          1.0
k000000006  5.0          8.0          5.0          9.0          7.0
k000000007  0.0          400.0        0.0          7.0          0.0
k000000008  5.0          5.0          5.0          6.0          8.0
k000000009  4.0          5.0          4.0          7.0          6.0
k000000010  5.0          12.0         5.0          5.0          10.0
k000000011  3.0          3.0          3.0          4.0          5.0
k000000012  0.0          0.0          0.0          0.0          4.0
k000000013  0.0          0.0          0.0          0.0          1.0
k000000014  0.0          0.0          0.0          0.0          0.0
k000000015  2.0          6.0          2.0          1.0          10.0
k000000016  0.0          2.0          0.0          5.0          13.0
k000000017  4.0          5.0          4.0          8.0          9.0
k000000018  0.0          0.0          0.0          5.0          0.0
k000000019  0.0          0.0          0.0          0.0          0.0
k000000020  1.0          2.0          1.0          6.0          9.0
k000000021  1.0          1.0          1.0          12.0         9.0
k000000022  1.0          1.0          1.0          2.0          6.0
k000000023  0.0          2.0          0.0          4.0          7.0
k000000024  2.0          5.0          2.0          3.0          18.0
k000000025  1.0          3.0          1.0          3.0          4.0
k000000026  0.0          0.0          0.0          0.0          2.0
```

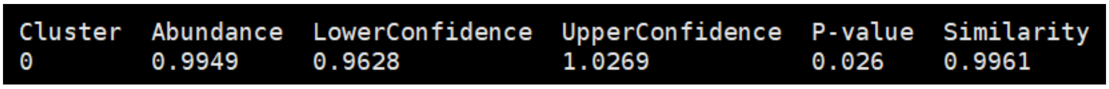
Step 5.

Finally, you need to use the “subspecies” module to estimate the strain-level abundance:

```
$ python MIST.py subspecies --cluster_output
Example_Dir/output/_MIST_ref_cluster.csv --mismatch_matrix_output
Example_Dir/output/_MIST_map_Mismatch_matrix.csv --read_length 200 --output
Example_Dir/output/
```

Where the input file “_MIST_map_Mismatch_matrix.csv” is obtained from Step 4; the input cluster file “_MIST_ref_cluster.csv” is obtained from Step 2.

The output files “_MIST_0.98_measure.csv”, “_MIST_0.99_measure.csv”, and “_MIST_0.999_measure.csv” predict the abundance at the 98%, 99%, and 99.9% ANI levels, respectively, which correspond to the clustering levels set in Step 2. In the “_MIST_0.999_measure.csv”, for example, we can see that cluster 1 account for 99.49% of E. coli mNGS reads in the sample. From the clustering file “_MIST_ref_cluster.csv”, we know that, at the 0.999 level, cluster 1 corresponds to the reference genomes AP012030.fa and CP002729.fa.



```
Cluster  Abundance  LowerConfidence  UpperConfidence  P-value  Similarity
0        0.9949     0.9628           1.0269           0.026    0.9961
```