Meteor Quick Start

This document describes the usage of the Meteor pipeline and additional data preparation tools (MeteorImportFastq.rb and MeteorReferenceBuilder.rb).

Getting and installing Meteor pipeline

Download the meteor source package from https://forgemia.inra.fr/metagenopolis/meteor
Follow the installation instructions described in the README file.

Data preparation tools are located in the folder data_preparation_tools/. This folder will be referenced as $TOOLS in the rest of this document.

Dependencies

- **ruby** interpreter (>=1.9) + module **inifile** (>=3.0.0)
  We recommend to install ruby using your Linux distribution package manager.
  Otherwise, be sure that the ruby interpreter path is in the environment variable PATH.
  To install module inifile , execute the following command as root [or common user] :

```
gem install [--user-install] inifile
```

- bowtie1 if you have colorspace (SOLiD) csfasta and qual files, bowtie2 otherwise.
  https://sourceforge.net/projects/bowtie-bio/files/bowtie2/
  https://sourceforge.net/projects/bowtie-bio/files/bowtie/

Be sure that the folder where bowtie2 and/or bowtie1 executables are located is in the PATH.

Setup a Meteor project

1- Create a folder *projects* (for the Meteor project)
2- Create a folder *reference* (for the gene catalog)
3- Create a folder *workflow* (for the meteor workflow)

Folders *projects*, *reference*, and *workflow* are referenced as $PROJECTS, $REFERENCE, and $WORKFLOW respectively in the rest of the document.

4- In your $PROJECTS space, create your project folder and sub-folders (sample, mapping and profiles).
   NB: projectname shall not contain any whitespace character.

```
mkdir -p $PROJECTS/projectname/{sample,mapping,profiles}
```

The sample folder will contain sequenced data and descriptive metadata (ini file) for each sequencing library, in separated sub-folders (one sub-folder per sample). The mapping folder will contain mapping/counting informations generated by Meteor for each sample. The profiles folder will contain the final abundance table of a collection of selected samples.

5- Copy fastq files in sample folder
6- Deploy sample data with MeteorImportFastq.rb script

```
ruby $TOOLS/MeteorImportFastq.rb -i $PROJECTS/projectname/sample -p projectname -t
sequencing_techno -m "sample_name_pattern"
```

Option -t : sequencing technology (Proton, Illumina, SOLiD).

Option -m : pattern (as a ruby regular expression) to extract sample names from the fastq file names.
Example: Given the list of fastq files below, the pattern to extract the sample name (in red) is : "SAMPLE_\d+"

Illumina_lib1-SAMPLE_01_trimmed_and_filtered.fastq
Illumina_lib1-SAMPLE_02_trimmed_and_filtered.fastq
Illumina_lib2-SAMPLE_01_trimmed_and_filtered.fastq
Illumina_lib2-SAMPLE_02_trimmed_and_filtered.fastq

For this example, the command would be :

```
ruby $TOOLS/MeteorImportFastq.rb -i $PROJECTS/projectname/sample -p projectname -t
Illumina -m "SAMPLE_\d+"
```

Use the option -c to indicate file compression (gzip) in the ini descriptive files.

If the fastq files are already organized in sub folders (one per sample), add option -d to the previous command.

Build an indexed gene reference catalog for Meteor

To build an indexed gene reference catalog from a gene fasta file, use the script MeteorReferenceBuilder.rb:

```
ruby $TOOLS/MeteorReferenceBuilder.rb -i my_fasta_file -p $REFERENCE -n my_catalog_name
```

By default, this command builds bowtie1 and bowtie2 indexes. If you have no color space (SOLiD) data, use the option -1 to avoid building bowtie1 index. Similarly, use option -2 to avoid building bowtie2 index.

Setup a Meteor workflow

Meteor uses a dedicated workflow (as a INI file) storing informations for mapping parameters against main reference (the ecosystem gene reference catalog) and contaminant (or excluded) references. The following example is a workflow named 10M_catalog_rmHosts95_illumina.ini for mapping/counting against a fictive 10M genes catalog :

```
[worksession]
meteor.reference.dir = /path/to/reference
meteor.db.type = binary
meteor.mapping.program = bowtie2
meteor.mapping.file.format = sam
meteor.is.cpu.percentage = 0
meteor.cpu.count = 8
meteor.excluded.reference.count = 1

[main_reference]
meteor.reference.name = 10M_catalog
meteor.matches = 10000
meteor.mismatches = 5
meteor.is.perc.mismatches = 1
meteor.bestalignment = 1
meteor.mapping.prefix.name = mapping_vs_10M_catalog
meteor.counting.prefix.name = vs_10M_catalog_id95

[excluded_reference_1]
meteor.reference.name = Homo_sapiens_GRCh38-p13
meteor.mismatches = 10
meteor.is.perc.mismatches = 1
meteor.bestalignment = 1
meteor.mapping.prefix.name = mapping_vs_human_GRCh38
```

In your own workflow, you have to modify the entry meteor.reference.dir with the path of your reference data ($REFERENCE)

In the reference sections:

- meteor.reference.name : reference name as mentioned to the script MeteorReferenceBuilder.rb
- meteor.matches : max number of permitted alignments by bowtie
- meteor.mismatches : max number of permitted mismatches during counting
- meteor.is.perc.mismatches : 1 (meteor.mismatches is in percentage) or 0 (absolute value)
- meteor.bestalignment : 1 : keep best alignments (or equal alignments) in term of identity, 0 else

NB: The value of meteor.excluded.reference.count has to be equal to the number of excluded_reference_N sections.

Typically, all workflow files are centralized in the $WORKFLOW folder.

Run Meteor

First, be sure that the folder where meteor.rb, meteor-counter and meteor-profiler are located is in the environment variable PATH and that these programs have execution permission.
Then use the ruby script **meteor.rb** to process the mapping and the counting on each sample, as follows:

```
meteor.rb -w $WORKFLOW/10M_catalog_rmHosts95_illumina.ini -i
$PROJECTS/projectname/sample/SAMPLE_01 -p $PROJECTS/projectname -o mapping
```

The result is stored in the mapping/SAMPLE_01 folder, including severals subfolders:

- sub-folders mapping_vs* : results of the mapping against ecosystem gene catalog (the 10M gene catalog in this example) and against excluded contaminant reference (human in this example).
- sub-folder SAMPLE_01_vs_10M_catalog_id95_gene_profile : counting results (total, unique, shared, smart_shared counts…) for the sample SAMPLE_01, gathered in a tabulated file census.dat.

Launch as many meteor.rb process as the number of samples in your project.

meteor.rb uses lock files to avoid simultaneous meteor jobs on same data. At the end of a successful meteor process on a given sample, lock files are removed. But, if a meteor job fails (whatever the reason), some lock files may persist and block any later meteor job on the same samples.

The option -L allows to ignore lock file that could still exist after a previous meteor job that failed.

Moreover, by default Meteor does not redo already done mapping and counting tasks (from a previous meteor session). So use the option -f to tell meteor to ignore (overwrite) all previous mapping results on these data.

When the mapping and counting of all the samples are done, process the census.dat files with **meteor-profiler** to generate a single abundance table considering one particular counting method (e.g. smart_shared). Suppose all your sample names start with SAMPLE_, then the command will look like this:

```
meteor-profiler -p $PROJECTS/projectname -w $WORKFLOW/10M_catalog_rmHosts95_illumina.ini -f
$PROJECTS/projectname/profiles -t smart_shared_reads
-o projectname_vs_10M_catalog_rmHost95_illumina_smart_shared_reads
$PROJECTS/projectname/mapping/SAMPLE_*/*_gene_profile/census.dat
```

The census.dat file list (last parameter) can also be stored in a unique text file ending with .txt
See the output of `meteor-profiler -h` for more details on options.

The resulting abundance table will be stored in
profiles/projectname_vs_10M_catalog_rmHost95_illumina_smart_shared_reads.tsv

In addition, you may be interested in various counting statistics stored in csv files ending with *_counting_report.csv*.