Running the small example dataset

A small example dataset using *C. elegans* is available for download. Please see below for the exact contents of the download, as well as instructions for downloading it from the command line.

Additionally, the expected outputs of the pipeline are included. Due to differences among versions of the external programs used in the pipeline (samtools, etc.), it is unlikely that all intermediate files will match exactly. However, we do include a script to compare the ORFs predicted as translated using the pipeline to those which are expected. If these differ significantly, it suggests something is not working correctly in the pipeline.

If the results differ significantly, please run the pipeline using the "DEBUG" logging level (see the usage instructions). This causes the scripts to output detailed runtime information which can be helpful for tracking down problems. If the problem is still not clear, please report the problem at the github bug tracker.

In total, creating the reference index files should take about 5 minutes and running the main pipeline should take an additional 15 to 20 minutes on a commodity laptop.

- Example dataset files
- Creating the reference index files
- Running the Rp-Bp pipeline (also with replicates)
- Common problems

Example dataset files

The example dataset is distributed as a .tar.gz file and includes the following:

- `WBcel235.79.chrI.yaml`. The configuration file for creating the reference index files. It includes all possible options for creating the indices as well as detailed descriptions.

- `WBcel235.chrI.fa`. The reference sequence of Chromosome I for *C. elegans*.

- `WBcel235.79.chrI.gtf`. The Ensembl, version 79 annotations for Chromosome I for *C. elegans*.

- `X03680_1.fasta`. The sequences of the ribosomal subunits for *C. elegans*. The reference accession is X03680.1.

- `c-elegans-test.yaml`. The configuration file for running the prediction pipeline. This example configuration file includes all possible options for the pipeline with detailed explanations of the options. The **exception** is the `min_metagene_profile_count` option, which has a value of 10 rather than its default of 1000. This is set artificially low because of the small number of reads in the sample dataset.

- `riboseq-adapters.fa`. An example adapter file for use with `flexbar`. It includes typical TruSeq and ArtSeq adapters, as well as a few adapters from the literature. It also includes a custom adapter used to create the sample dataset.
- `c-elegans.test-chrI.rep-1.fastq.gz`. A small test sequencing dataset. It has been constructed to include some reads which uniquely map to the annotated transcripts, some reads which map to ribosomal sequences, some reads which do not uniquely map to the genome and some reads which are filtered due to quality issues.
- `c-elegans.test-chrI.rep-2.fastq.gz`. Another small test sequencing dataset.
- `expected-orf-predictions`. The expected predictions and sequence files for each replicate (c-elegans-rep-1 and c-elegans-rep-2 files) and the merged replicates (c-elegans-test files). Please see the usage instructions for the meaning of each of the files.

Downloading from the command line

The following commands can be used to download and extract the example .tar.gz file:

```
wget http://cloud.dieterichlab.org/index.php/s/7XHsCqZqU9AbQqB/download -O c-elegans-chrI-ex
tar -xvf c-elegans-chrI-example.tar.gz
```

Back to top

Creating the reference index files

Before running the example the paths in the `WBcel235.79.chrI.yaml` configuration file must be updated to point to the correct locations. The following configuration values should be updated to point to the appropriate files in the example. (Mostly, `/home/bmalone/python-projects/rp-bp/data/` should be replaced to the location of the examples.)

- `gtf`
- `fasta`
- `ribosomal_fasta`
- `genome_base_path`
- `ribosomal_index`
- `star_index`

The following command will create the necessary reference files using 2 CPUS and 4GB of RAM for STAR. Please see the usage instructions for the expected output files.

The `--use-slurm` and related options can also be used if SLURM is available. Please see the usage instructions for more information.

N.B. The `--overwrite` flag is given below to ensure all of the files are (re-)created. In typical use cases, if some of the files already exist (e.g., the STAR index), then this flag can be omitted.

This command should only take about 5 minutes on recent commodity hardware (such as a laptop).

N.B. This command may print some warning messages such as:

```
WARNING  misc.utils 2016-11-02 17:25:05,023 : [utils.call_if_not_exists]:
This function is deprecated. Please use the version in misc.shell_utils
instead.
```

These are not problematic and will be updated in future releases.

```
prepare-rpbp-genome WBcel235.79.chrI.yaml --num-cpus 2 --mem 4G --overwrite --logging-level
```

Back to top

Running the Rp-Bp pipeline

Before running the example the paths in the `c-elegans-test.yaml` configuration file must be updated to point to the correct locations. The following configuration values should be updated to point to the appropriate files in the example. (Mostly, `/home/bmalone/python-projects/rp-bp/data/` should be replaced to the location of the examples.)

Reference files and locations should be exactly the same as used in the `WBcel235.79.chrI.yaml` file.

- `gtf`
- `fasta`
- `genome_base_path`
- `ribosomal_index`
- `star_index`

The sample and output file paths must also be updated.

- `riboseq_samples`
- `riboseq_data`
- `adapter_file`

The following command will run the Rp-Bp (and Rp-chi) translation prediction pipelines using 2 CPUS. Please see the usage instructions for the expected output files.

The `--use-slurm` and related options can also be used if SLURM is available. Please see the usage instructions for more information.

N.B. The `--overwrite` flag is given below to ensure all of the files are (re-)created. In typical use cases, if some of the files already exist (e.g., the quality-filtered reads), then this flag can be omitted.

N.B. While performing the MCMC sampling, many messages indicating the "Elapsed Time" will be printed. This is a known issue with pystan. Additionally, many "Informational Message: The current Metropolis proposal is about to be rejected because of the following issue" may also appear. These are also expected and (typically) do not indicate an actual problem.

Using replicates

The Rp-Bp pipeline handles replicates by adding the (smoothed) ORF profiles. The Bayes factors and predictions are then calculated based on the combined profiles. The `--merge-replicates` flag indicates that the replicates should be merged. By default, if the `--merge-replicates` flag is given, then predictions will not be made for the individual datasets. The `--run-replicates` flag can be given to override this and make predictions for both the merged replicates as well as the individual datasets.

The replicates are specified by `riboseq_biological_replicates` in the configuration file. This value should be a dictionary, where the key of the dictionary is a string description of the condition and the value is a list that gives all of the sample replicates which belong to that condition. The names of the sample replicates must match the dataset names specified in `riboseq_samples`.

N.B. These calls may also produce deprecation warnings like:

```
WARNING  misc.utils 2016-11-02 17:31:47,545 : [utils.check_programs_exist]: This function is
```

These are again not problematic and will be corrected in future releases.

```
# do not merge replicates
run-all-rpbp-instances c-elegans-test.yaml --overwrite --num-cpus 2 --logging-level INFO --k

# merging the replicates, do not calculate Bayes factors and make predictions for individual
run-all-rpbp-instances c-elegans-test.yaml --overwrite --num-cpus 2 --logging-level INFO --m

# merging the replicates and calculating Bayes factors and making predictions for individual
run-all-rpbp-instances c-elegans-test.yaml --overwrite --num-cpus 2 --logging-level INFO --m
```

Back to top

Common problems

Some common problems result due to versions of external programs. The can be controlled using command line options to `run-all-rpbp-instances`.

- `--flexbar-format-option`. Older versions of flexbar used `format` as the command line option to specify the format of the fastq quality scores, while newer versions use `qtrim-format`. Depending on the installed version of flexbar, this option may need to be changed. Default: `qtrim-format`

- `--star-executable`. In principle, `STARlong` (as opposed to `STAR`) could be used for alignment. Given the nature of riboseq reads (that is, short due to the experimental protocols of degrading everything not protected by a ribosome), this is unlikely to be a good choice, though. Default: `STAR`
- `--star-read-files-command`. The input for `STAR` will always be a gzipped fastq file. `STAR` needs the system command which means "read a gzipped text file". As discovered in Issue #35, the name of this command is different on OSX and ubuntu. The program now attempts to guess the name of this command based on the system operating system, but it can be explicitly specified as a command line option. Default: `gzcat` if `sys.platform.startswith("darwin")`; `zcat` otherwise. Please see python.sys documentation for more details about attempting to guess the operating system.

Back to top