Using custom alignment files

While the `run-all-rpbp-instances` (and `run-rpbp-pipeline`) scripts are designed to handle all steps from translating the raw fastq files from the sequencer into a high-confidence list of translated ORFs, it is also possible to start the pipeline from any step. For example, the trimming, filtering and aligning steps could be handled using a different preprocessing strategy. The configuration file must be created as usual, but the `riboseq_samples` field only needs to contain the names of the samples. (The path can be left blank.)

Then, the files produced by the external processing pipelines must be placed at the appropriate location according to the names generated by Rp-Bp. In particular, depending on which steps of preprocessing have been performed, the files should be placed in the following locations:

- Trimmed and quality filtered reads
  - **trimmed and filtered reads**. `<riboseq_data>/without-adapters/<sample_name>[.<note>].f`
- Reads not aligning to ribosomal sequences
  - **retained reads**. `<riboseq_data>/without-rrna/<sample_name>[.<note>].fastq.gz`
- Aligned reads
  - **sorted reads aligned to the genome**. `<riboseq_data>/without-rrna-mapping/<sample_name>`
  - **aligned reads which map uniquely to the genome**. A sorted bam file containing all alignments of reads to the genome with multimapping reads filtered out. `<riboseq_data>/without-rrna-mapping/<sample_name>[.<note>]-`

The files can also be symlinks with the appropriate name. Please see the usage instructions for more details about the expected content of each file. Only the last file must be in the expected location. For example, if trimming, filtering and aligning has been performed, only the alignment files must be present. The pipeline will issue warning messages that the earlier files are missing, but it will begin as normal once it finds the, e.g., alignment files.

Example

This example shows how to run Rp-Bp starting with the alignment files for the "original example". It uses the `c-elegans.alignments-only.yaml` config file. The config file can also be downloaded with the following command.

```
wget http://cloud.dieterichlab.org/index.php/s/fdrhJDKJfqhaGIT/download -O c-elegans.alignme
```

Because this example uses the alignment files from the original example, it is required to run the original example first.

Before running this example the paths in the configuration file must be updated to point to the correct locations. (Mostly, `/home/bmalone/python-projects/rp-bp/data/` should be replaced to the location of the examples.)

First, we create an empty folder for the example.

```
$ cd <base_dir>
$ mkdir c-elegans.alignments-only
$ cd c-elegans.alignments-only
```

N.B. The path of the `c-elegans.alignments-only` file **must** exactly match the `riboseq_data` path in the config file.

We then create the necessary folder structure.

```
$ mkdir without-rrna-mapping
$ cd without-rrna-mapping
```

N.B. The subfolder must exactly be named `without-rrna-mapping`.

Finally, we symlink the bam and index files from running the original example.

```
$ ln -s <original-example>/without-rrna-mapping/c-elegans-rep-1.test.bam c-elegans-rep-1.ali
$ ln -s <original-example>/without-rrna-mapping/c-elegans-rep-2.test.bam c-elegans-rep-2.ali
$ ln -s <original-example>/without-rrna-mapping/c-elegans-rep-1.test.bam.bai c-elegans-rep-1
$ ln -s <original-example>/without-rrna-mapping/c-elegans-rep-2.test.bam.bai c-elegans-rep-2
```

N.B The filenames must **exactly** match the patterns described above, for example, `<sample_name>[.<note>].bam` for genome alignment files. In this particular example, the `note` in the `c-elegans.alignments-only.yaml` config file is `alignments-only`. Similarly, the `riboseq_samples` in the config file have names `c-elegans-rep-1` and `c-elegans-rep-2`. Thus, we concatenate the respective values to form the filenames.

After copying the yaml config file into the directory, the folder structure is as follows.

```
$ find c-elegans.alignments-only/c-elegans.alignments-only/

c-elegans.alignments-only/without-rrna-mapping
c-elegans.alignments-only/without-rrna-mapping/c-elegans-rep-1.alignments-only.bam
c-elegans.alignments-only/without-rrna-mapping/c-elegans-rep-2.alignments-only.bam
c-elegans.alignments-only/without-rrna-mapping/c-elegans-rep-1.alignments-only.bam.bai
c-elegans.alignments-only/without-rrna-mapping/c-elegans-rep-2.alignments-only.bam.bai
c-elegans.alignments-only/c-elegans.alignments-only.yaml
```

We can now run the Rp-Bp pipeline.

```
run-all-rpbp-instances c-elegans.alignments-only.yaml --overwrite --num-cpus 2 --logging-lev
```

The log file indicates that various files were missing. For example, it includes the following messages:

```
WARNING  misc.shell_utils 2017-06-15 19:02:35,224 : Some input files ['/this/does/not/matter
flexbar --qtrim-format sanger --max-uncalled 1   -n 2 --zip-output GZ -r /this/does/not/matt
```

However, as desired, the pipeline finds the alignment file and proceeds from there.

```
INFO     root     2017-06-15 19:02:35,250 : remove-multimapping-reads /home/bmalone/python-p
INFO     root     2017-06-15 19:02:35,250 : calling
```

After running the above command, the predictions should appear as usual in the `orf-predictions` folder. Please see the usage instructions for the exact set of expected output files.