genome-profile-construction

April 18, 2016

0.1 Creating filtered genome profiles

This document describes the steps necessary to create the filtered genome profiles for use in the Rp-Bp and Rp-chi pipelines. This process must be run on each sample independently. The various reference sequence indices must already be created.

The entire profile creation process can be run automatically using the `create-filtered-genome-profile` script. It reads most of the required paths from a (YAML) configuration file. Additionally, it automatically creates some of the output paths.

The script also accepts a `--overwrite` flag. If this is given, then steps for which the output file already exists will be skipped.

It also accepts a `--do-not-call` flag. If this flag is given, then the commands below will be printed but not executed.

Logging options can also be given to this script.

Configuration file keys

The following keys are read from the configuration file. Keys with [`brackets`] are optional.

- `riboseq_data`. The base output location for all created files.
- `ribosomal_index`. The base output path for the Bowtie2 index of the ribosomal sequence
- `gtf`. The path to the reference annotations
- `star_index`. The base output path for the STAR index of the genome sequence
- `periodic_models`. A list of paths to pickled StanModel objects which somehow represent a periodic metagene profile
- `nonperiodic_models`. A list of paths to pickled StanModel objects which model non-periodic behavior

Input files

The required input files are only those suggested by the configuration file keys.

- `ribosomal_index`.
- `gtf`.
- `star_index`.
- `periodic_models`.
- `nonperiodic_models`.

Output files

```
In [ ]: create-filtered-genome-profile /path/to/my/raw-data.fastq.gz /path/to/my/input-config.yaml my-s
```

0.1.1 Creating the base genome profile

First, the base genome profile is created from the raw reads following the procedure outlined in the paper with `create-base-genome-profile`. This script wraps calls to * `flexbar`, for removing adapter sequences and low-quality reads * `bowtie2`, for removing reads which align to ribosomal sequences * `STAR`, for aligning reads to the genome, taking into account splicing * `remove-multimapping-reads` (a light wrapper around `samtools`), for removing reads with multiple genomic alignments

This script uses a YAML configuration file to control the behavior of these programs. The configuration file also includes the base output location for the intermediate files.

The script also accepts a `--overwrite` flag. If this is given, then steps for which the output file already exists will be skipped.

It also accepts a `--do-not-call` flag. If this flag is given, then the commands below will be printed but not executed.

Logging options can also be given to this script.

Configuration file keys

The following keys are read from the configuration file. Keys with [`brackets`] are optional. * `riboseq_data`. The base output location for all created files. * `ribosomal_index`. The base output path for the Bowtie2 index of the ribosomal sequence * `gtf`. The path to the reference annotations * `star_index`. The base output path for the STAR index of the genome sequence

- [`adapter_file`]. A fasta file containing a set of adapter sequences used by flexbar
- [`max_uncalled`]. The maximum number of Ns to permit in a read without filtering
- [`pre_trim_left`]. The number of bases to remove from the 5' end of all reads
- [`adapter_sequence`]. A single sequence used to remove adapters within flexbar
- [`quality_format`]. The quality format of the reads in the raw fastq file
- [`flexbar_compression`]. The type of compression used for the raw fastq file. TODO: guess this based on the raw_data extension

STAR options. These options are passed through to STAR unchanged. * [`align_intron_min`]. default: 20 * [`align_intron_max`]. default: 100000 * [`out_filter_intron_motifs`]. default: RemoveNoncanonicalUnannotated * [`out_filter_mismatch_n_max`]. default: 1 * [`out_filter_mismatch_n_over_l_max`]. default: 0.04 * [`out_filter_type`]. default: BySJout * [`out_sam_attributes`]. default: AS NH HI nM MD

Output files

This script primarily creates the following files. (STAR also creates some temporary and log files.) * `riboseq_data`/without-adapters/`sample-name`.fastq.gz * `riboseq_data`/with-rrna/`sample-name`.fastq.gz * `riboseq_data`/without-rrna/`sample-name`.fastq.gz * `riboseq_data`/without-rrna-mapping/sam/`sample-name`Aligned.out.sam * `riboseq_data`/without-rrna-mapping/sam/`sample-name`Aligned.toTranscriptome.out.bam * `riboseq_data`/without-rrna-mapping/bam/`sample-name`.bam * `riboseq_data`/without-rrna-mapping/bam/`sample-name`.transcriptome.bam * `riboseq_data`/without-rrna-mapping/bam/`sample-name`-unique.bam

Indices are also created for the bam files. `sample-name`.bam and `sample-name`.transcriptome.bam are sorted versions of the respective sam files. `sample-name`-unique.bam is sorted and does not contain any multimappers.

Difference from paper

The fifth step of creating the base genome profile in the paper is "Everything except the 5' end of the remaining reads is removed." This profile is not explicitly constructed in the pipeline. The `sample-name`-unique.bam already contains the necessary information.

```
In [ ]: create-base-genome-profile /path/to/my/raw-data.fastq.gz /path/to/my/input-config.yaml my-sample
```

0.1.2 Constructing metagene profiles

We next extract the metagene profiles for each read length using the `extract-metagene-profiles` script. This script is largely a wrapper around pysam; it uses pandas to efficiently find reads with 5' ends around translation initiation sites.

Command line options

This script does not use the configuration file. It accepts the following command line options. Options with [`brackets`] are optional.

- `bam`. The `sample-name`-unique.bam file, which contains the base genome profile.

- `orfs`. The BED file created while preprocessing the reference annotations.
- `out`. A csv.gz file which contains the metagene profiles around the translation initiation and termination sites.
- [`--lengths`]. If specified, then metagene profiles will be created for reads of each length. Otherwise, profiles will be created for each read length present in the bam file.
- [`--seqids-to-keep`]. If this list is given, then only transcripts appearing on these identifiers will be used to construct the metagene profiles. The identifiers must match exactly (e.g., "2" and "chr2" do not match)
- [`--start-upstream`]. The number of bases upstream of the translation initiation site to begin constructing the metagene profile. default: 50
- [`--start-downstream`]. The number of bases downstream of the translation initiation site to end the metagene profile. default: 20
- [`--end-upstream`]. The number of bases upstream of the translation termination site to begin constructing the metagene profile. default: 50
- [`--end-downstream`]. The number of bases downstream of the translation termination site to end the metagene profile. default: 20

Input file
This script uses the `riboseq_data`/without-rrna-mapping/bam/`sample-name`-unique.bam file as input.
Output file
This script creates a csv.gz file with the following columns: * count * position * type * length

```
In [ ]: extract-metagene-profiles /path/to/my/input/`sample-name`-unique.bam /path/to/my/input-orfs.bed
```

0.1.3 Estimating periodicity Bayes factors

The penultimate step in creating the filtered genome profile is identifying the offsets for each read length which exhibit periodicity. We identify periodicity by estimating the Bayes factor of the metagene profile starting at each possible offset . This involves Markov chain Monte Carlo sampling with Stan, which is managed through the pystan interface. The `estimate-metagene-profile-bayes-factors` script controls this sampling.

Command line options
This script does not use the configuration file. It accepts the following command line options. Options with [`brackets`] are optional.

- `metagene_profiles`. The csv.gz file containing the metagene profiles produced in the previous step.
- `out`. The output csv.gz file, which contains Bayes factor estimates for all offsets in the specific range. The estimates are made for all read lengths present in the input file.
- `periodic_models`. A list of paths to pickled StanModel objects which somehow represent a periodic metagene profile.
- `nonperiodic_models`. A list of paths to pickled StanModel objects which model non-periodic behavior.
- [`--periodic-offset-start`]. The position, relative to the translation initiation site, to begin calculating periodicity Bayes factors (inclusive)
- [`--periodic-offset-end`]. The position, relative to the translation initiation site, to stop calculating periodicity Bayes factors (inclusive)

- [`--metagene-profile-length`]. The length of the profile to use in the models. `metagene_profile_length` + `periodic_offset_end` must be consistent with the length of the extracted metagene profile
- [`--seed`]. The random seed for the MCMC sampling. default: 8675309
- [`--chains`]. The number of chains to use in the MCMC sampling. default: 2
- [`--iterations`]. The number of iterations to use for each chain in the MCMC sampling. The first half of the iterations are discarded as burn-in samples. All of the remaining samples are used to estimate the posterior distributions. That is, we do not use thinning. default: 500

Input file
The input is a csv(.gz) file which contains at least the following fields.

- count
- position
- type
- length

Presumably, it is created using the `extract-metagene-profiles` script.
Output file
The output is a csv.gz file which contains the following information (fields) for each offset and read length.

- p_periodic_mean. The mean of the marginal distribution of the best periodic model.
- p_periodic_var. The variance of the marginal distribution of the best periodic model.
- p_nonperiodic_mean. The mean of the marginal distribution of the best nonperiodic model.
- p_nonperiodic_var. The variance of the marginal distribution of the best nonperiodic model.
- profile_sum. The number of reads which are used in this profile.
- bayes_factor_mean. The mean of the posterior distribution of the Bayes factor
- bayes_factor_var. The variance of the posterior distribution of the Bayes factor

```
In [ ]: estimate-metagene-profile-bayes-factors
```

0.1.4 Selecting periodic read lengths and P-site offsets

The second half of the final step in creating the filtered genome profile is selecting the read lengths which exhibit periodicity and determining their P-site offsets. This is handled by the `get-best-periodicity-and-offsets` script.

This reads in each of the estimated Bayes factors file and decides whether to use the given read length based on the estimates.

TODO: The last three steps can remain separate, but they should not produce so many files. It would be sufficient to create one file for each step.

```
In [ ]:
```