reference-genome-indexing

April 18, 2016

0.1 Creating the reference genome index files

This document describes the steps necessary to prepare a reference genome and matching annotation for use in the Rp-Bp and Rp-chi pipelines. The process must only be run once for each reference genome and set of annotations.

It shows some sample calls. For all programs, the `--help` option can be given to see the complete list of parameters.

Input * Reference genome sequence * GFF3/GTF annotations matching the reference sequence * Ribosomal sequence

The entire index creation process can be run automatically using the `prepare-genome` scripts. 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. * `gtf`. The path to the reference annotations * `fasta`. The path to the reference genome sequence * `ribosomal_fasta`. The path to the ribosomal sequence

- `base_path`. The path to the output directory for the transcript fasta and ORFs
- `name`. A descriptive name to use for the created files
- `ribosomal_index`. The base output path for the Bowtie2 index of the ribosomal sequence
- `star_index`. The base output path for the STAR index of the genome sequence
- [`orf_note`]. An additional description used in the filename of the created ORFs
- [`start_codons`]. A list of strings that will be treated as start codons when searching for ORFs. default: [`ATG`]
- [`stop_codons`]. A list of strings that will be treated as stop codons when searching for ORFS. default: [`TAA`, `TGA`, `TAG`]

Input files

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

- `gtf`. The GTF/GFF3 file containing the reference annotations. This file must be compatible with gffread for extracting coding sequences. Typically, this means at least the `exon` features must be present, and the transcript identifiers must match for exons from the same transcript. Furthermore, the ORFs are labeled based on their positions relative to annotated coding sequences. This labeling is based on the `CDS` information output by `gffread`. Thus, for it to work correctly, the `CDS` features must also be present in the annotation file.

- `fasta`. The input fasta file should contain all chromosome sequences. The identifiers must match those in the GTF file. Typically, the "dna.toplevel.fa" file from Ensembl contains the appropriate sequences and identifiers.

- `ribosomal_fasta`. The ribosomal DNA sequence is typically repeated many times throughout the genome. Consequently, it can be difficult to include in the genome assembly and is often omitted. Therefore, a separate fasta file is required for these sequences (which are later used to filter reads).

Output files
`base_path`/transcript-index/`name`.transcripts.fa
`base_path`/transcript-index/`name`.genomic-orfs.`orf_note`.bed.gz

```
In [ ]: prepare-genome /path/to/my/input-config.yaml --num-procs p --mem m
```

0.1.1 Extracting spliced transcript sequences

First, spliced transcript sequences are extracted using `extract-transcript-fasta`. This script is a light wrapper around `gffread` from the Cufflinks package.

```
In [ ]: extract-transcript-fasta /path/to/my/input-annotations.gtf /path/to/my/input/reference-sequence
```

0.1.2 Extracting ORFs from transcripts

The open reading frames within the spliced transcript sequences are identified using `extract-orfs`. This script uses pybedtools.

```
In [ ]: extract-orfs /path/to/my/input/transcript-sequences.fa /path/to/my/output/orfs.bed.gz --num-pro
```

0.1.3 Building the ribosomal sequence index

The ribosomal sequence Bowtie2 index must be created with `bowtie2-build-s`.

```
In [ ]: bowtie2-build-s /path/to/my/input/ribosomal-fasta.fa /path/to/my/output/ribosomal-index
```

0.1.4 Creating the STAR index

The `STAR` genome index is created for mapping reads to the genome. The `create-star-reference` script creates the reference. It is a light wrapper around `STAR -runMode genomeGenerate`.

```
In [ ]: create-star-reference /path/to/my/input-annotations.gtf /path/to/my/input/reference-sequence.fa
```