Variant Calling and Annotation Pipeline

Contents

Description

This is a pipeline for SNP, indels, and structural variant calling and annotation implemented in snakemake. It takes in input paired fastq files and sample sheets with details of the reference data.

The intended application is to detect variants in haploid, unicellular parasite strains (e.g. find variants in *P. berghei* strain 820 relative to reference genome(s)).

Scroll the `Snakefile` and the files in `workflow` directory for details of the pipeline, briefly:

- Optionally trim reads with cutadapt
- Align reads with `bwa`, sort and index the output bam file
- Detect snp and indels with freebayes
- Detect structural variants with delly
- Annotate variants with VEP
- Perform some basic quality control on raw reads (`fastQC`) and alignments (with `samtools stats` and mosdepth)

Variant calling is performed on all samples simultaneously.

Setup

If not already done, install conda, bioconda, and mamba.

Create a dedicated environment and install the dependencies listed in `requirements.txt`. The environment name here is `genetic-variant-detection-in-cell-pops`, but you can choose a different name:

```
conda create --yes -n genetic-variant-detection-in-cell-pops
conda activate genetic-variant-detection-in-cell-pops
mamba install -n genetic-variant-detection-in-cell-pops --yes --file requirements.txt
```

Sample sheets

See the test sample sheets in directory `test/sample_sheets` for examples.

Populate the tab-separated file `sample_sheet.tsv` as appropriate. This tabular file links libraries to their raw fastq files and genomes. Columns are:

Column	Description
library_id	Identifier for the library

Column	Description
genome	Identifier of the reference genome to align this library_id (see `genomes.tsv`)
cutadapt_adapter	Adapter sequence to trim from the fastq reads. Use NA to skip trimming. The sequence `AGATCGGAAGAGC` is suitable for most Illumina library preps
fastq_r1	Full path to fastq read 1
fastq_r2	Full path to fastq read 2

If you have more than one pair of fastq file per library, add one row per fastq pair and use the same library_id for each of them.

Populate the tab-separated file `genomes.tsv`. This file indicates where to download the reference data. Columns are:

Column	Description
genome	Genome identifier matching the ID in `sample_sheet.tsv`
fasta	URL or path to local file for the genome fasta reference
gff	URL or path to local file for GFF annotation. Use NA if this genome doesn't have a GFF file. This pipeline has been developed with gff files from VEuPathDB in mind

Note that it's fine to have the same library processed against different genomes.

NB: Identifiers of libraries and genomes can be any string you like as long as they don't contain blank spaces and special characters.

Run

With option `--dry-run` you can check what would be executed. Remove `--dry-run` to actually execute the pipeline. The output will be in the directory specified by option `-d`.

This command using the test data should complete wihtout errors with and without `--dry-run`:

```
snakemake --dry-run -p -j 5 --use-conda -d output/ \
   -C ss=$PWD/test/sample_sheets/sample_sheet.tsv \
      genomes=$PWD/test/sample_sheets/genomes.tsv
```

See `snakemake -h` for explanation of the various options and for additional options (e.g. for cluster execution).

Output

(This section may be out of date)

All output will be in the directory set by `-d/--directory` option and it should be mostly self-explanatory. The most relevant output of the variant calling is probably:

- `{genome}/freebayes/variants.vcf.gz`: VCF output of freebayes normalised and with additional FORMAT tags about alternate allele frequency.
- `{genome}/freebayes/vep.vcf.gz`: Same as above but with additional annotation from VEP provided a gff file for this genome was provided.

- `{genome}/freebayes/vep.tsv.gz`: This is the vcf file `vep.vcf.gz` reformatted to tab-separated table and with samples vertically concatenated.
- `{genome}/delly/variants.vcf.gz|vep.vcf.gz|vep.tsv.gz`: Same files as above but from delly.

Miscellanea

This is how we made the test fastq files:

```
cd test
samtools faidx PlasmoDB-52_PbergheiANKA.fasta PbANKA_01_v3:90936-111656 > tmp.fasta
wgsim -S 1 -r 0.01 -N 2000 -e 0.001 tmp.fasta \
    PbANKA_820_S1_R1.fastq PbANKA_820_S1_R2.fastq
wgsim -S 2 -r 0.01 -N 2000 -e 0.001 tmp.fasta \
    PbANKA_820_S2_R1.fastq PbANKA_820_S2_R2.fastq
gzip -f *.fastq
rm tmp.fasta
```

To compile this markdown document to pdf:

```
pandoc -V colorlinks=true -V geometry:margin=1in README.md -o README.pdf
```