SNP

Zofia Kochańska, Małgorzata Kukiełka, Jakub Białecki, Julia Smolik

About the program

This program was developed as part of a project on the subject Architecture of large bioinformatics projects at the Faculty of Mathematics, Informatics and Mechanics, University of Warsaw. It is a Snakemake pipeline for SNP detection using tools from GATK. The only argument that the user will need to do is provide a config file specifying parameters (e.g., whether the data comes from single-end (SE) or pair-end (PE) sequencing as well as whether the input files are from DNA or RNA sequencing) and locations of raw FASTQ (.GZ) files.

Getting Started

The user must have Anaconda installed. To run the program, the user does not need to install the necessary packages themselves. It is enough to import the environment included by the authors. In this environment there are libraries necessary to run the program. To install the environment in the console you must type:

```
conda env create -f environment.yml
```

After the installation is complete, the program is ready to use. After the user enters all the parameters for their desired analyses into the config file, to run the program, the user should be in the directory where the Snakefile is located and type in the console

```
snakemake -c [number] --use-conda
```

where [number] is an integer indicating how many threads to use.

The program will load the raw DNA/RNA-seq files, perform pre-processing on them to eventually identify the SNPs and SNVs found in them. The program also performs annotation of the variants found and produces plots summarizing the results obtained.

Pipeline overview

Input:

- Config file

  In the config file, the user defines all the rules for the program. This is the file that the user can edit. In it, the path to the sequencing files must be set, as well as the reference genome file and the file that defines the control group. The config file contains various parameters that will determine the pre-processing of the data, its analysis, and the result of the variant detection. Most importantly, the user must specify in the config file whether the data they are using comes from single-end or pair-end reads as well as whether the files come from DNA or RNA sequencing. The user can choose to trim the reads using the trimmomatic tool, to which they can also set selected parameters (e.g., SLIDINGWINDOW or MINLEN). They can also choose a tool to map the reads (HISAT2, Bowtie2, Bwa). Finally, the user can choose to summarize their analysis using the generated plot.

- Snakefile

  This is the main code of the program. The pipeline steps are defined here.

- Reference genome fasta file
- Sequencing fasta files (single-end or pair-end)

Functions and tools used

Pre-processing the data:

- Read trimming
  - Trimmomatic is a program that performs several helpful trimming operations on illumina paired-end and single-end data. The command line is used to provide the trimming stages with their associated parameters.
    The user can specify the following parameters:

    SLIDINGWINDOW:<windowSize>:<requiredQuality> Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold; default: 4:15
    windowSize: specifies the number of bases to average across
    requiredQuality: specifies the average quality required

    LEADING:<quality> Cut bases off the start of a read, if below a threshold quality; default: 3
    quality: Specifies the minimum quality required to keep a base

TRAILING:<quality> Cut bases off the end of a read, if below a threshold quality; default: 3
quality: Specifies the minimum quality required to keep a base.

MINLEN:<length> Drop the read if it is below a specified length; default: 36
length: Specifies the minimum length of reads to be kept

The trimming works with FASTQ, either uncompressed or gzipp'ed FASTQ. Use of gzip format is determined based on the .gz extension.

For single-ended data, one input and one output file are specified, plus the processing steps. For paired-end data, two input files are specified, and 4 output files, 2 for the 'paired' output where both reads survived the processing, and 2 for corresponding 'unpaired' output where a read survived, but the partner read did not.

- Read mapping
  - BWA (Burrows-Wheeler Alignment Tool) – a software package for mapping low-divergent sequences against a large reference genome, such as the human genome. It consists of three algorithms: BWA-backtrack, BWA-SW and BWA-MEM. The first algorithm is designed for Illumina sequence reads up to 100bp, while the rest two for longer sequences ranged from 70bp to 1Mbp. BWA-MEM and BWA-SW share similar features such as long-read support and split alignment, but BWA-MEM, which is the latest, is generally recommended for high-quality queries as it is faster and more accurate. BWA-MEM also has better performance than BWA-backtrack for 70-100bp Illumina reads.

    The user can specify the number of threads used, default: 6

  - Bowtie 2 – an ultrafast and memory-efficient tool for aligning sequencing reads to long reference sequences. It is particularly good at aligning reads of about 50 up to 100s of characters to relatively long (e.g., mammalian) genomes. Bowtie 2 supports gapped, local, and paired-end alignment modes. Multiple processors can be used simultaneously to achieve greater alignment speed.

    The user can specify the number of threads used, default: 6

  - HISAT2 – a fast and sensitive alignment program for mapping next-generation sequencing reads (both DNA and RNA) to a population of human genomes as well as to a single reference genome.

    The user can specify the number of threads used, default: 6

- Marking of duplicates

  MarkDuplicates – used to identify duplicate reads. This tool locates and tags duplicate reads in a BAM or SAM file, where duplicate reads are defined as originating from a single fragment of DNA or RNA. Duplicates can arise during sample preparation e.g., library construction using PCR. MarkDuplicates also produces a metrics file indicating the numbers of duplicates for both single- and paired-end reads.

- Modification of reads

  SplitNCigarReads - splits reads that contain Ns in their cigar string (e.g., spanning splicing events in RNAseq data). Identifies all N cigar elements and creates k+1 new reads (where k is the number of N cigar elements). The first read includes the bases that are to the left of the first N element, while the part of the read that is to the right of the N (including the Ns) is hard clipped and so on for the rest of the new reads. Used for post-processing RNA reads aligned against the full reference. This function is only used if the data comes from RNA-seq.

  The user can specify the number of threads used, default: 6

Searching for variants:

- Variant calling:

  HaplotypeCaller – call germline SNPs and indels via local re-assembly of haplotypes. The HaplotypeCaller can call SNPs and indels simultaneously via local de-novo assembly of haplotypes in an active region. In other words, whenever the program encounters a region showing signs of variation, it discards the existing mapping information and completely reassembles the reads in that region. This allows the HaplotypeCaller to be more accurate when calling regions that are traditionally difficult to call, for example when they contain different types of variants close to each other.

  The user can specify the number of threads used, default: 6

- Select variants:

  SelectVariants – select a subset of variants from a VCF file. This tool makes it possible to select a subset of variants based on various criteria to facilitate certain analyses. Examples of such analyses include comparing and contrasting cases vs. controls, extracting variant or non-variant loci that meet certain requirements, or troubleshooting some unexpected results, to name a few.

Filtering and evaluation:

- Filtering variants:

  VariantFiltration – filter variant calls based on INFO and/or FORMAT annotations. This tool is designed for hard-filtering variant calls based on certain criteria. Records are hard filtered by changing the value in the FILTER field to something other than PASS. Filtered records will be preserved in the output unless their removal is requested in the command line.

  The user can specify the following parameters. SNPs that do not match any of these conditions will be considered good and marked PASS in the output VCF file:

  QualByDepth (QD) This is the variant confidence (from the QUAL field) divided by the unfiltered depth of non-reference samples; default: <5.0 for snv and indels

  Quality (QUAL) This is the quality of the variant; default: <35.0 for snv and indels

  StrandOddsRatio (SOR) The StrandOddsRatio annotation is one of several methods that aims to evaluate whether there is strand bias in the data. Higher values indicate more strand bias; default: >2.5 for snvs

  FisherStrand (FS) Phred-scaled p-value using Fisher's Exact Test to detect strand bias (the variation being seen on only the forward or only the reverse strand) in the reads. More bias is indicative of false positive calls; default: >50.0 for snvs, >75.0 for indels

  RMSMappingQuality (MQ) This is the Root Mean Square of the mapping quality of the reads across all samples; default: <55 for snvs

  MappingQualityRankSumTest (MQRankSum) This is the u-based z-approximation from the Mann-Whitney Rank Sum Test for mapping qualities (reads with ref bases vs. those with the alternate allele). Note that the mapping quality rank sum test cannot be calculated for sites without a mixture of reads showing both the reference and alternate alleles, i.e., this will only be applied to heterozygous calls; default: <-12.5 for snvs

  ReadPosRankSumTest (ReadPosRankSum) This is the u-based z-approximation from the Mann-Whitney Rank Sum Test for the distance from the end of the read for reads with the alternate allele. If the alternate allele is only seen near the ends of reads, this is indicative of error. Note that the read position rank sum test cannot be calculated for sites without a mixture of reads showing both the reference and alternate alleles, i.e., this will only be applied to heterozygous calls; default: <-8.0 for snvs, <-10.0 for indels

  DP This annotation is used to provide counts of read depth at two different levels, with some important differences. At the sample level (FORMAT), the DP value is the

count of reads that passed the caller's internal quality control metrics (such as MAPQ >17, for example) ; default: <10.0 for snvs and indels

- Annotation of variants

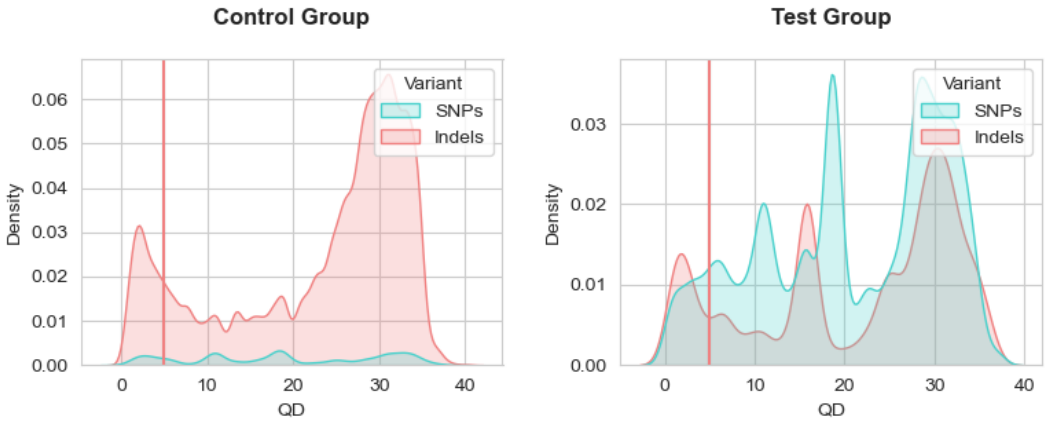
  VEP – determine the effect of the variants (SNPs, insertions, deletions, CNVs or structural variants) on genes, transcripts, and protein sequence, as well as regulatory regions.

  Simply input the coordinates of the variants and the nucleotide changes to find out the genes and Transcripts affected by the variants, location of the variants (e.g., upstream of a transcript, in coding sequence, in non-coding RNA, in regulatory regions), consequence of the variants on the protein sequence (e.g. stop gained, missense, stop lost, frameshift), see variant consequences, known variants that match the variants from the input, and associated minor allele frequencies from the 1000 Genomes Project.

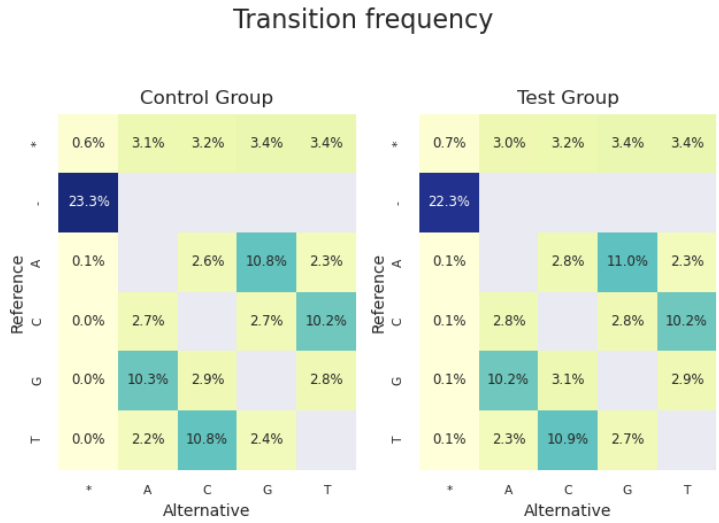
- Visualizing the variants found - Custom scripts creating summary charts

  - Visualize – creates a report with density plots for features used to filter variants found in the test and control samples, along with the values of filters used in the analysis (marked by red vertical lines).

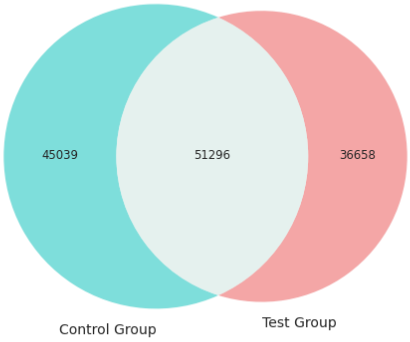
    Example: QD (QualByDepth) < 5.0 for snv and indels



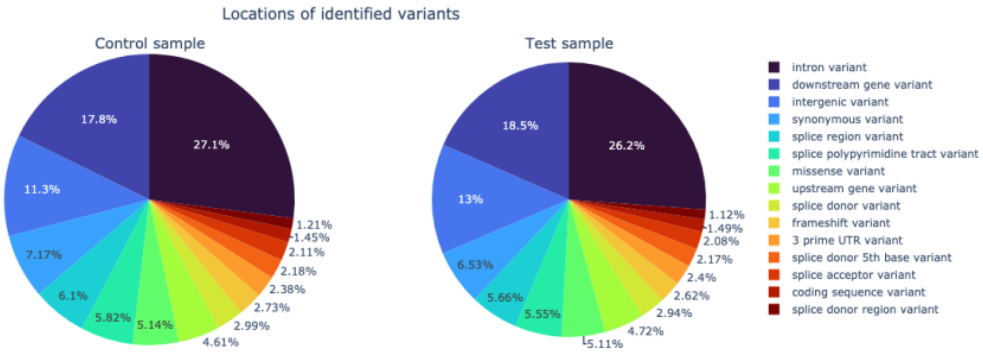
  - Final plots – generates heatmaps with variant frequencies in the analyzed samples. The asterisk represents sequences longer than a single nucleotide, while a hyphen marks a single-nucleotide base in the reference genome in insertions.



Additionally, the script outputs a Venn diagram of variants (snvs and indels) found in the test and control samples (based on CHROM and POS from the final VCF files)



The script also outputs pie charts of locations of variants identified in the test and control sample based on consequence terms retrieved from VCF files (column: INFO, field: Consequence, additional information about the terms here).



Evaluate the data – quality control, statistics:

- MultiQC – a tool to create a single report visualising output from multiple tools across many samples, enabling global trends and biases to be quickly identified. MultiQC can plot data from many common bioinformatics tools and is built to allow easy extension and customization.

Output

- Variant calls (vcf file)
- MultiQC report (includes summaries of the input data after data pre-processing)
- Plots visualizing the found variants (and the filters used on them) and comparing the results for the test and control group

Intermediate output files such as bam files are also kept. In addition to the above tools, there are other tools used to combine the steps. If you are interested in the details, please see the snakemake rules for each step.