# A Snakemake pipeline for detection of SNPs using tools from GATK

Zofia Kochańska, Małgorzata Kukiełka, Jakub Białecki, Julia Smolik

# Project goal

Our goal for this project was to create a pipeline for detecting SNPs using the Snakemake workflow management system. Our intention was to include a full range of operations in the pipeline from initial pre-processing to final visualising of the variants found.

# Analysis flowchart

**Downloading the data**

Data type: pair-end RNA-seq
Database: SRA, accession number: PRJNA508203
Reference genome: ENA, GCA_000298735.1

**Analysis with Snakemake scripts**

**Comparison of the pipeline performance with the authors' results**

https://doi.org/10.1038/s41598-020-70527-8

# Data origin

Bakhtiarizadeh, M. R., & Alamouti, A. A. (2020). *RNA-Seq based genetic variant discovery provides new insights into controlling fat deposition in the tail of sheep.*

Zel

Lori-Bakhtiari

# Getting started

Prerequisites:

- Anaconda
- Project folder from github

Before running the program, it is important to install the necessary conda environment with the command:

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

# Getting started

After creating the environment the program is ready to run. However, the user should first specify all of the parameters (and path to the input files) by editing the ***config.yaml*** file.

Afterwards, the program can be started with the command:
***snakemake -c [number] --use-conda*** , where [number] is the number of threads used

# Config file

Arguments to specify in the config file:

- Path to input files, reference genome file
- Choose Single-End or Paired-End reads
- Choose type of data (DNA or RNA sequencing)
- Whether to trim the data or not
- Select mapping program (eg. Bowtie2)
- Whether to annotate the data using vep or not
- Whether to visualise the output or not

# Tools used

- Pre-processing:
  - Read trimming: Trimmomatic
  - Read mapping: BWA, Bowtie2, HISAT2
  - Marking of duplicates: MarkDuplicates
  - Modification of reads: SplitNCigarReads (only if data is from RNA-seq)

# Tools used

- Searching for variants:
  - Variant calling: HaplotypeCaller
  - Selecting variants: SelectVariants
- Filtering and evaluation
  - Filtering variants: VariantFiltration
  - Annotation of variants: VEP
  - Visualising found variants: Custom scripts
- Data evaluation: MultiQC

# Trimmomatic

Parameters to specify:

- SLIDINGWINDOW
- LEADING
- TRAILING
- MINLEN

# VariantFiltration

Parameters:

- QualByDepth (QD)
- Quality (QUAL)
- StrandOddsRatio (SOR)
- FisherStrand (FS)
- RMSMappingQuality (MQ)
- MappingQualityRankSumTest (MQRankSum)
- ReadPosRankSumTest (ReadPosRankSum)
- DP value

# Output files

MultiQC

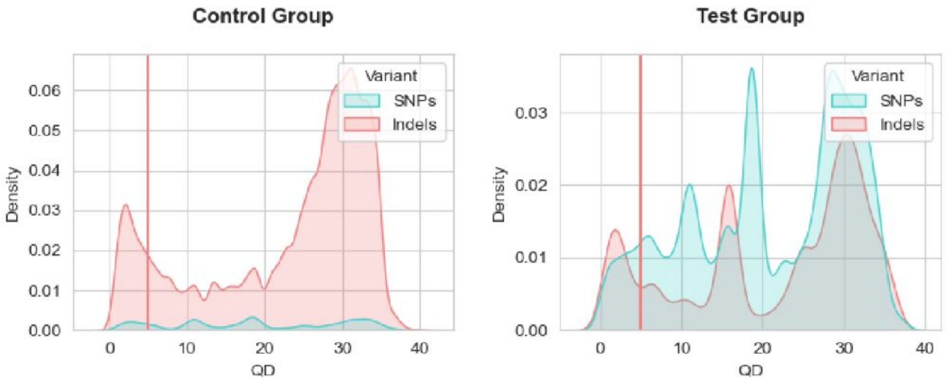
- Variant calls (vcf files)
- MultiQC report (includes summaries of the input data after data pre-processing)
- Plots visualizing the found variants (and the filters used on them) and comparing the results for the test and control group

# Visualisation

Visualise: create a density plot for features used to filter variants.

Example: QD

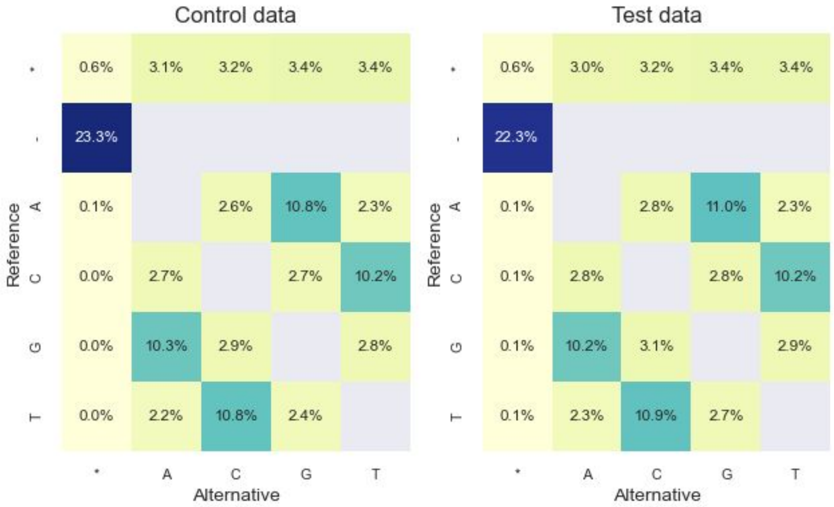
Example: QD (QualByDepth) < 5.0 for snv and indels



# Visualisation

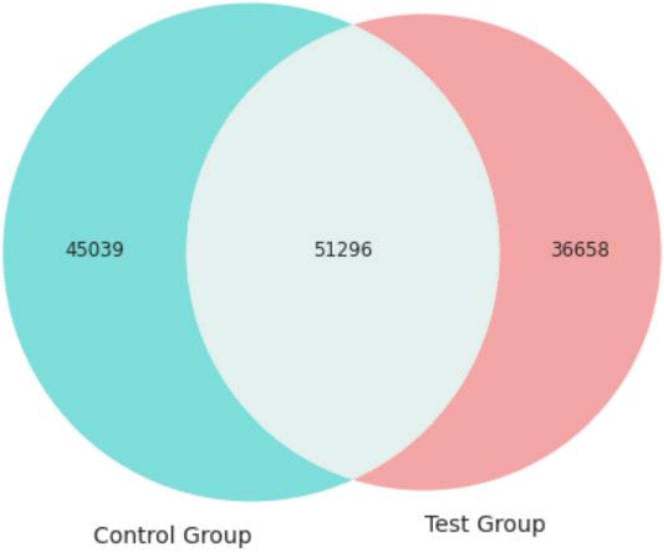
Final plots – generates heatmaps with variant frequencies in the analyzed samples.

Variant frequency heatmaps

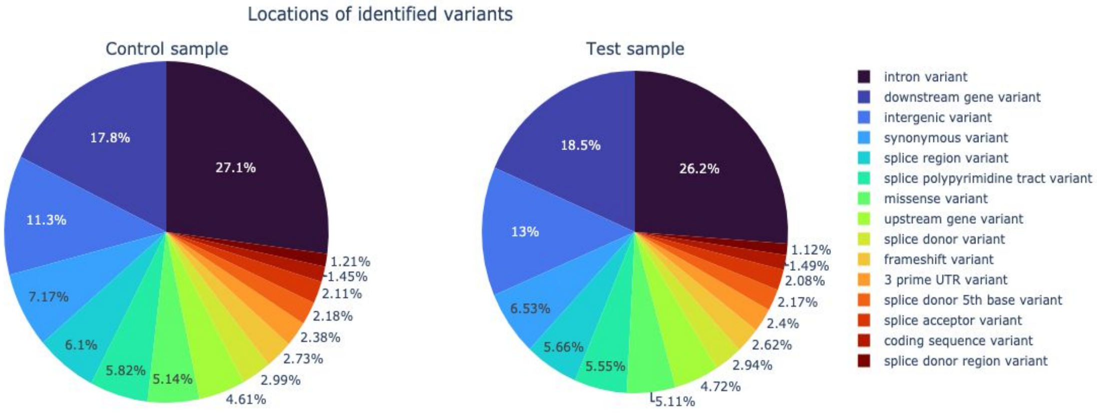


# Visualisation

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)



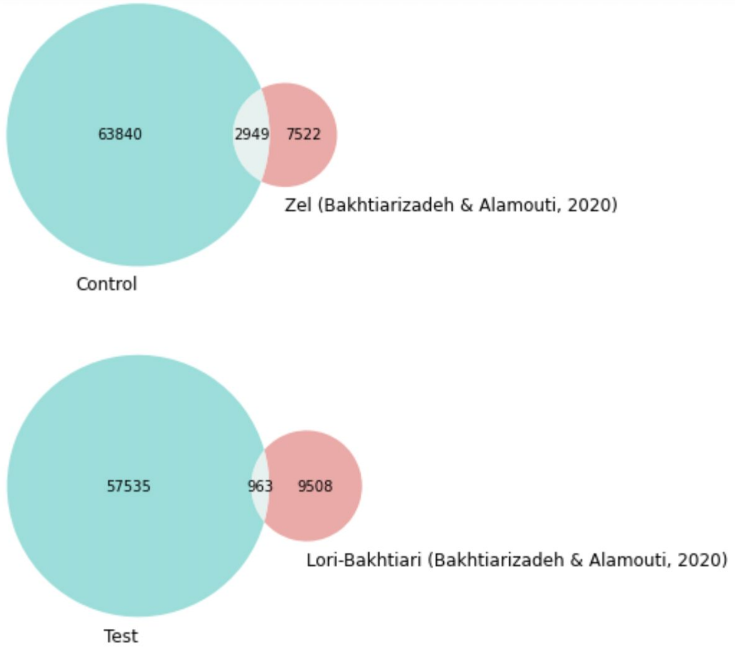
# Visualisation



The last plot that is being created by the workflow is summarizing placements in the genome of the found variations

# Results comparison

Comparison of variants found by our pipeline and published by the authors of the aforementioned article



# Documentation

Full documentation can be found at:

***https://students.mimuw.edu.pl/~js406162/documentation/index.html***

# Thank you for your attention!