GENE EXPRESSION ANALYSIS OF ARABIDOPSIS THALIANA USING BIOINFORMATIC TOOLS

Nucleotide sequence analysis

Ana Hojan

SOFTWARE INSTALLATION

Conda and other programs

Our analysis will be performed with the help of conda, which is an open source package management system and environment management system. Our backages will be downloaded with the channel that specialises in bioinformatics software - bioconda.

The following programs/packages should be installed in the same (or switch inbetween use) conda invironment:

- Sra-tools

```
conda install -c bioconda sra-tools
```

- Trimmomatic

```
conda install -c bioconda trimmomatic
```

- Multiqc

```
conda install -c bioconda multiqc
```

- STAR

```
conda install -c bioconda star
```

- Tablet

```
conda install -c bioconda tablet
```

- Subread

```
conda install -c bioconda subread
```

- rseqc

```
conda install -c bioconda reseq
```

- Entrez-direct

```
conda install -c bioconda entrez-direct
```

- Bedparse

```
conda install -c bioconda bedparse
```

- GATK

```
wget
https://github.com/broadinstitute/gatk/releases/download/4.2.6.1/gatk-4.
2.6.1.zip
```

Extract the archive and run the following command to create a new conda environment “gatk” with all dependencies to run gatk

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

- Snakemake

```
conda install -n base -c conda-forge mamba
mamba create -c conda-forge -c bioconda -n snakemake snakemake
```

FORMATION ABOUT RESEARCH AND SEQUENCING DATASET

Our dataset is part of a study: Combining chemical and genetic approaches to increase drought resistance in plants. The aim of this study was to find an effective way to increase drought resistance. Addition of fluorine atoms in the benzyl ring of the abscisic acid (ABA) receptor agonist AM1 optimizes its binding to ABA receptors by increasing the number of hydrogen bonds between the compound and the surrounding amino acid residues in the receptor ligand-binding pocket. The new chemicals, known as AMFs, have long-lasting effects in promoting stomatal closure and inducing the expression of stress-responsive genes. Application of AMFs or transgenic overexpression of the receptor PYL2 in Arabidopsis and soybean plants confers increased drought resistance. Best results are demonstrated when used simultaneously.

Sample: ABA_6h_rep2

SAMN07357157 • SRS2358483 • All experiments • All runs

Organism: Arabidopsis thaliana

Library:

Instrument: Illumina HiSeq 2500
Strategy: RNA-Seq
Source: TRANSCRIPTOMIC
Selection: cDNA
Layout: PAIRED
Construction protocol: Total RNAs were extracted using the TRIzol (Invitrogen) method, and RNase-free Dnase (Qiagen) was used to remove contaminating DNA before RNA-seq. Total RNAs were sent to Genomics Core Facilities of the Shanghai Center for Plant Stress Biology, SIBS, CAS (Shanghai, China) for library preparation and sequencing using standard Illumina protocols.

Experiment attributes:

GEO Accession: GSM2704829

Runs: 1 run, 20.2M spots, 5.1G bases, 1.8Gb

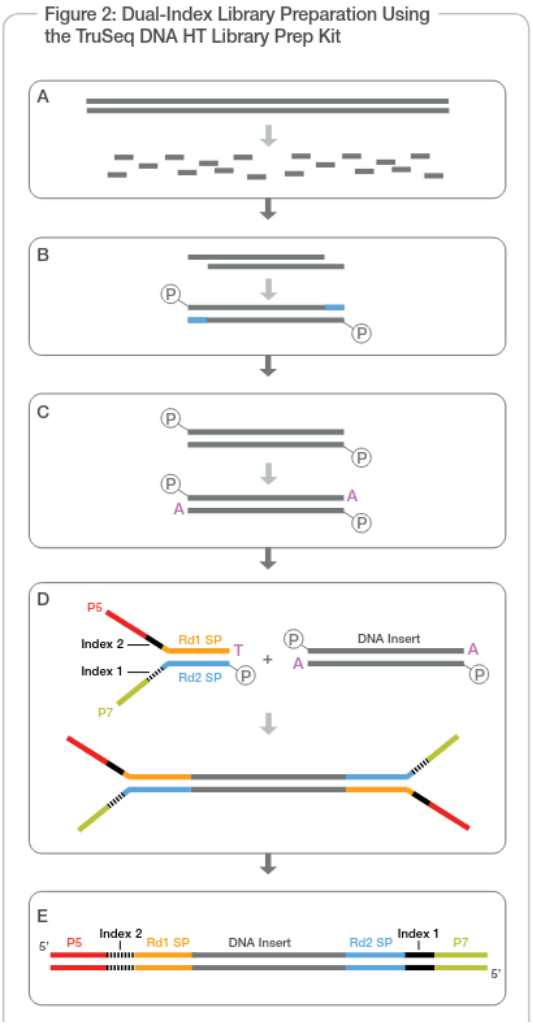
Run	# of Spots	# of Bases	Size	Published
SRR5831025	20,179,692	5.1G	1.8Gb	2017-08-14

For sample preparation the kits TRIzol (Invitrogen), and RNase-free Dnase (Qiagen) were used to remove contaminating DNA before RNA-seq. The information on kits for sequencing library preparation was not provided.
Stranded mRNA library preparation is considered advantageous due to retainment of strand specificity of origin for each transcript. Without strand information it is difficult

and sometimes impossible to accurately quantify gene expression levels for genes with overlapping genomic loci that are transcribed from opposite strands.

[Link to the BioProject record]

DNA library preparation using TruSeq kit:



The TruSeq DNA Library Preparation Kits are used to prepare DNA libraries with insert sizes from 300–500 bp for single, paired-end, and indexed sequencing. The protocol supports shearing by either sonication or nebulization with an input requirement of 1 µg of DNA. Library construction begins with fragmented gDNA (Figure 2A). Blunt-end DNA fragments are generated using a combination of fill-in reactions and exonuclease activity (Figure 2B). An ‘A’-base is then added to the blunt ends of each strand, preparing them for ligation to the sequencing adapters (Figures 2C). Each adapter contains a ‘T’-base overhang on the 3’-end, providing a complementary overhang for ligating the adapter to the A-tailed fragmented DNA. These adapters contain the full complement of sequencing primer hybridization sites for single, paired-end, and indexed reads. This eliminates the need for additional PCR steps to add the index tag and index primer sites (Figure 2D). Following the denaturation and amplification steps (Figure 2E), libraries can be pooled for sequencing. Master-mixed reagents and an optimized protocol contribute to a simple library construction workflow, requiring minimal hands-on time and few cleanup steps for processing large sample numbers. The workflow allows for high-throughput and automation-friendly solutions, as well as simultaneous manual processing of hundredsof samples. In addition, enhanced troubleshooting features are incorporated into each step of the workflow, with quality control sequences supported by Illumina RTA software.

DOWNLOADING DATA

With the tool fastq-dump we download a fastq file from a SRA database with the accession id: SRR5831025

```
(nsa) ana@HojansPC:~/nsa_project$ fastq-dump -X 500000 --origfmt
--split-3 --clip SRR5831025
```

Read 500000 spots for SRR5831025
Written 500000 spots for SRR5831025

Tags:

- -X 500000 Maximum spot id
- --origfmt Defline contains only original sequence name
- --split-3 3-way splitting for mate-pairs. For each spot, if there are two biological reads satisfying filter conditions, the first is placed in the `*_1.fastq` file, and the second is placed in the `*_2.fastq` file. If there is only one biological read satisfying the filter conditions,it is placed in the `*.fastq` file. All other reads in the spot are ignored.
- --clip Remove adapter sequences from reads

Output:
Two or three fastq files are formed:

@1
GGACGATGGGAAGGAGAAGAGCTAAGCCACGGTTGTCGCTACTGGCGTTTGCTGCAGC
GAGCTGAGCGATCTGTTGGATGGCTAAAGCTGGCTCAGAGTAGCTGAGGGAAGAGAAG
ACGGCTCCGG
+1
BBBBBFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF<FFFFFFFFFFFFFFFFFFFFBF<BFFF
@2
GTCATAAGTGTGCTGCTTCGACAACTCCAGAACAAACTCATCTTCTTTAGGTTTTTCCAG
AGCACGAAACCGGACCAGCTGTCTATTCTGGACATATTCAAGAAACGAAGGCACAGCTG
GGTAGAG
+2
BBBBBFFFFFFFFFFFFFFFFFFFFFFFFFBF/FFFFFFFFFFFFFFFFFFF<FFFFFFFFFFFF
FFFBB/F<BFFFFFFFFFFFFFFFFFF/F<<FF/<FFFFFFBBFFFFFFFF</BFFF7
@3
NTCGGAAGAGCACACGTCTGAACTCCAGTCACAGTCAACAATCTCGTATGCCGTCTTCT
GCTTGAAAAAAAAAAAAAAAAACAAAACCGCCCGCTTCTAGGAAAACAAAAACACAAACA
CGCAGAC

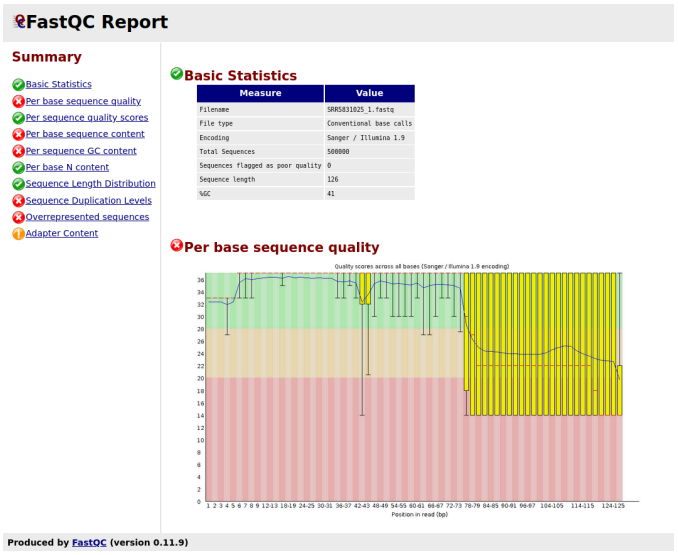
QUALITY OF THE SEQUENCES

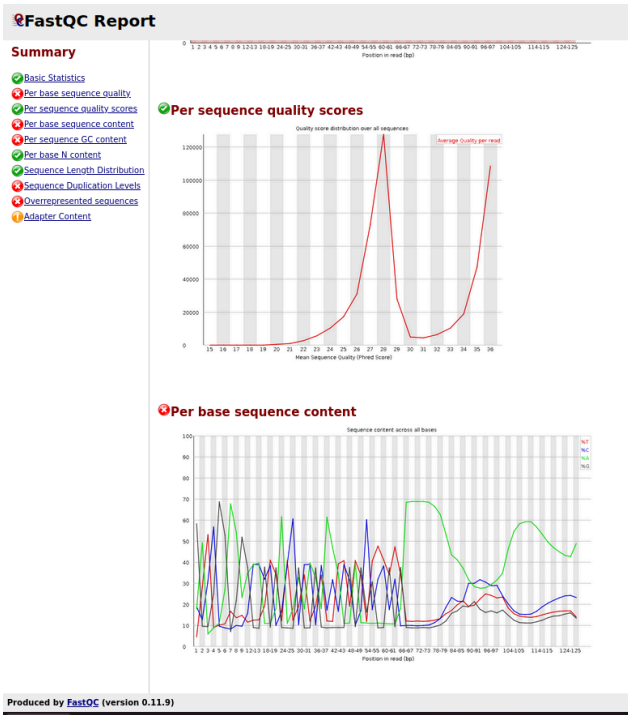
We check the quality of our data with the fastqc tool:

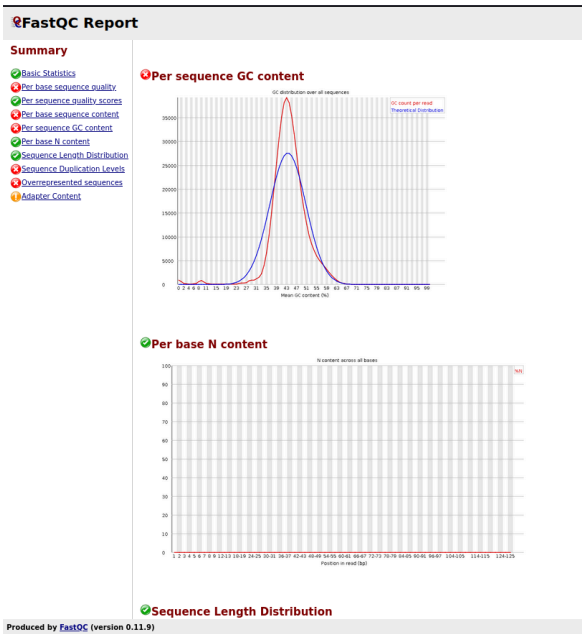
```
(nsa) ana@HojansPC:~/nsa_project$ fastqc SRR5831025_1.fastq
SRR5831025_2.fastq
```

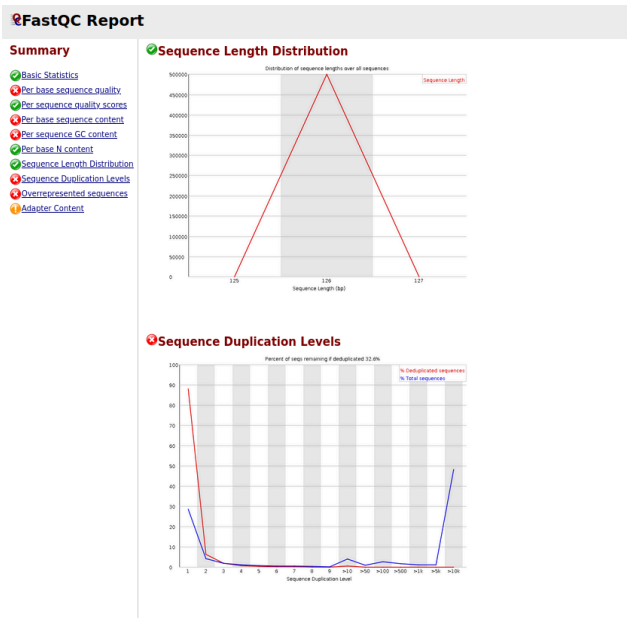
The tool creates a html report and a zip file for each of the files:

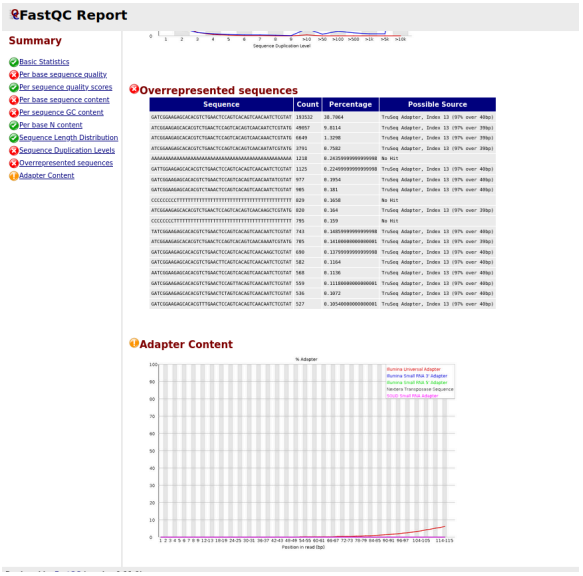
Report for SRR5831025_1.fastq



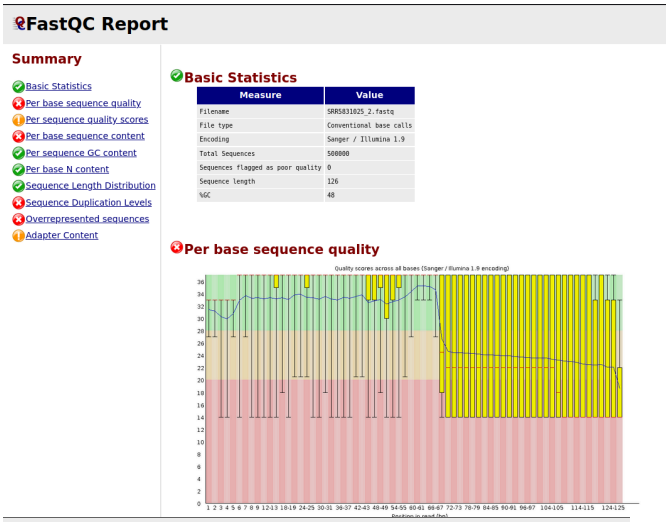


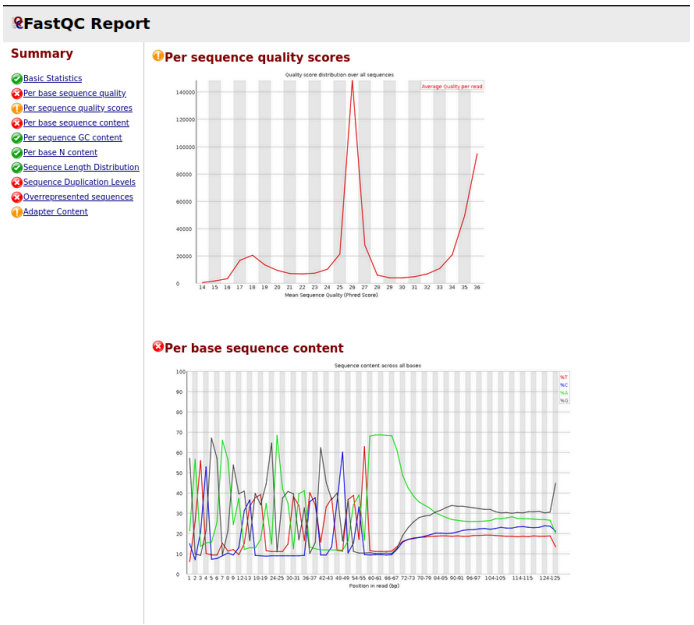


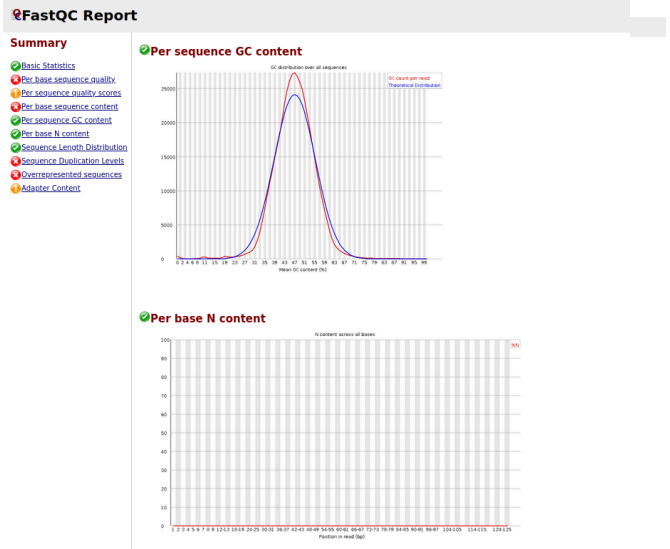


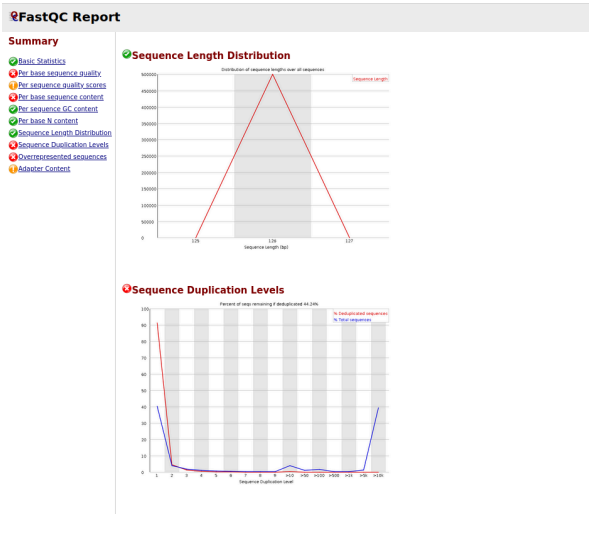


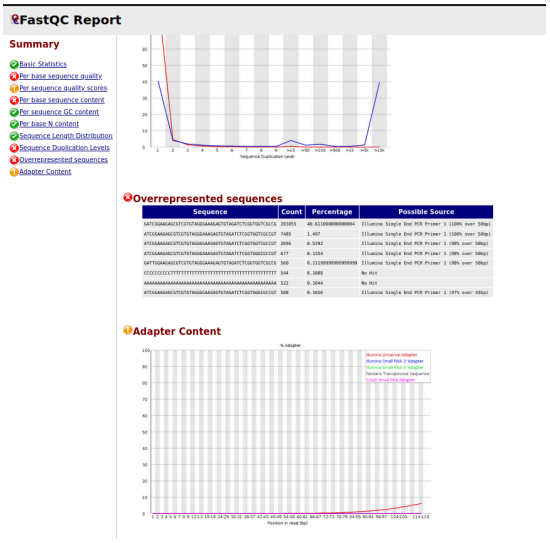
Report for SRR5831025_2.fastq











One of the more important tests they both fail is **Per base sequence quality**, which can be explained due to signal decay or phasing. No other worrisome signs are present, so the sequencing data from the facility is of good quality. With this plot we are looking primarily for overclustering and instrumental breakdowns.

At **per sequence quality score** test the plot gives you the average quality score on the x-axis and the number of sequences with that average on the y-axis. Majority of our reads have a high average quality score with no large bumps at the lower quality values.

Per base sequence plot content always gives a FAIL for RNA-seq data. This is because the first 10-12 bases result from the 'random' hexamer priming that occurs during RNA-seq library preparation.

Per sequence GC content plot gives the GC distribution over all sequences. Generally it is a good idea to note whether the GC content of the central peak corresponds to the expected % GC for the organism. Also, the distribution should be normal unless over-represented sequences (sharp peaks on a normal distribution) or contamination with another organism (broad peak). Our results of the first fastqc show a sharp peak on the normal distribution, which is why this test failed.

Sequence duplication levels plot can help identify a low complexity library, which could result from too many cycles of PCR amplification or too little starting material. For RNA-seq we don't normally do anything to address this in the analysis, but if this were a pilot experiment, we might adjust the number of PCR cycles, amount of input, or amount of sequencing for future libraries.

The **"Overrepresented sequences"** table is another important module as it displays the sequences (at least 20 bp) that occur in more than 0.1% of the total number of sequences. This table aids in identifying contamination, such as vector or adapter sequences. If the %GC content was off in the above module, this table can help identify the source. Our reads still include illumina adaptors, which is why this test failed.

Overall the results are favourable for continuation of the analysis. Another filtering will be performed with trimmomatic.

FILTERING BASED ON QUALITY PARAMETERS, TRIMMING…

Trimming illumina adapters with trimmomatic

With trimmomatic we have cut the illumina adapters:

```
(nsa) ana@HojansPC:~/nsa_project/trimmed_data trimmomatic PE
../raw_data/SRR5831025_1.fastq ../raw_data/SRR5831025_2.fastq
SRR5831025_1_paired_trimmed.fastq
SRR5831025_1_unpaired_trimmed.fastq
SRR5831025_2_paired_trimmed.fastq
SRR5831025_2_unpaired_trimmed.fastq
ILLUMINACLIP:/home/ana/Downloads/yes/envs/nsa/share/trimmomatic-0.
39-2/adapters/TruSeq2-PE.fa:2:30:10:2:True LEADING:3 TRAILING:3
SLIDINGWINDOW:4:15 MINLEN:36
```

Tags:

- Input files 1-2
- Output files 1-4
- ILLUMINACLIP: path to our fasta of the adapter used in the experiment. Cut adapter and other illumina-specific sequences from the read.
- SLIDINGWINDOW:<windowSize>:<requiredQuality> Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold.
    - 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
    - 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
    - quality: Specifies the minimum quality required to keep a base.
- CROP:<length> Cut the read to a specified length
    - length: The number of bases to keep, from the start of the read
- MINLEN:<length> Drop the read if it is below a specified length
    - length: Specifies the minimum length of reads to be kept

Output:

TrimmomaticPE: Started with arguments:

../raw_data/SRR5831025_1.fastq ../raw_data/SRR5831025_2.fastq SRR5831025_1_paired_trimmed.fastq SRR5831025_1_unpaired_trimmed.fastq SRR5831025_2_paired_trimmed.fastq SRR5831025_2_unpaired_trimmed.fastq ILLUMINACLIP:/home/ana/Downloads/yes/envs/nsa/share/trimmomatic-0.39-2/adap

ters/TruSeq2-PE.fa:2:30:10:2:True LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36
Multiple cores found: Using 4 threads
Using PrefixPair: 'AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCT' and 'CAAGCAGAAGACGGCATACGAGATCGGTCTCGGCATTCCTGCTGAACCGCTCTTCCGATCT'
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATT'
Using Long Clipping Sequence: 'AGATCGGAAGAGCGGTTCAGCAGGAATGCCGAGACCGATCTCGTATGCCGTCTTCTGCTTG'
Using Long Clipping Sequence: 'TTTTTTTTTTAATGATACGGCGACCACCGAGATCTACAC'
Using Long Clipping Sequence: 'TTTTTTTTTTCAAGCAGAAGACGGCATACGA'
Using Long Clipping Sequence: 'CAAGCAGAAGACGGCATACGAGATCGGTCTCGGCATTCCTGCTGAACCGCTCTTCCGATCT'
Using Long Clipping Sequence: 'AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCT'
ILLUMINACLIP: Using 1 prefix pairs, 6 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Quality encoding detected as phred33
Input Read Pairs: 500000 Both Surviving: 201409 (40.28%) Forward Only Surviving: 10568 (2.11%) Reverse Only Surviving: 1834 (0.37%) Dropped: 286189 (57.24%)
TrimmomaticPE: Completed successfully

Trimmomatic removed more than 60% of the reads, due to the MINLEN 36 tag, which removes all reads that are shorter than 36 base pairs. With other tags bases are cut, if the quality of the sliding window/read is poor.

Quality of the sequences after trimming with Multiqc

```
fastqc SRR5831025_1_paired_trimmed.fastq
SRR5831025_1_unpaired_trimmed.fastq SRR5831025_2_paired_trimmed.fastq
SRR5831025_2_unpaired_trimmed.fastq
(nsa) ana@HojansPC:~/Documents/trimmed_data$ multiqc .
```

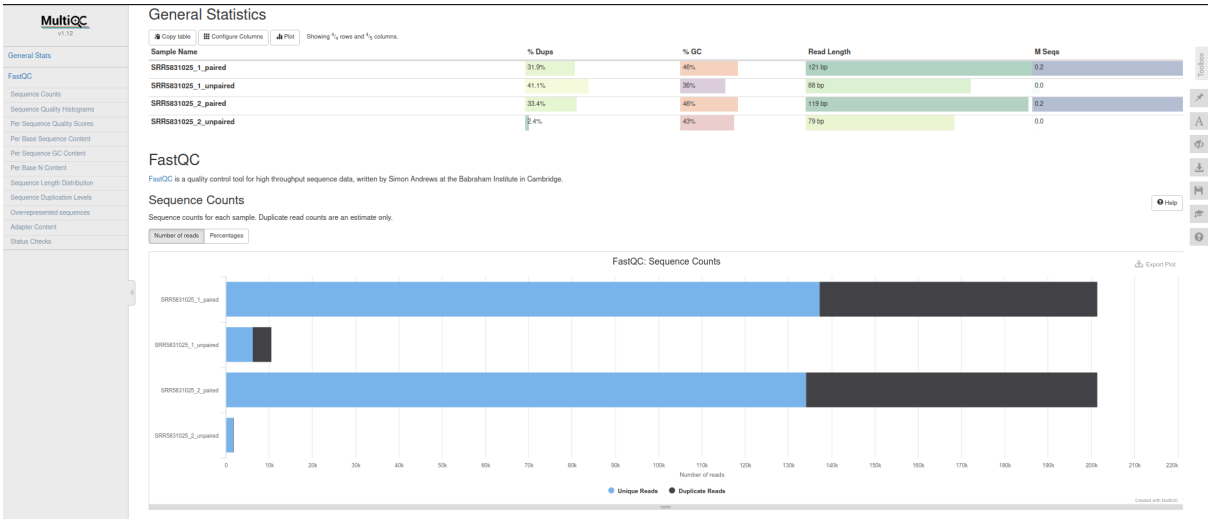
Tags:
The . means this directory. In the directory there should be the 4 fastqc files.

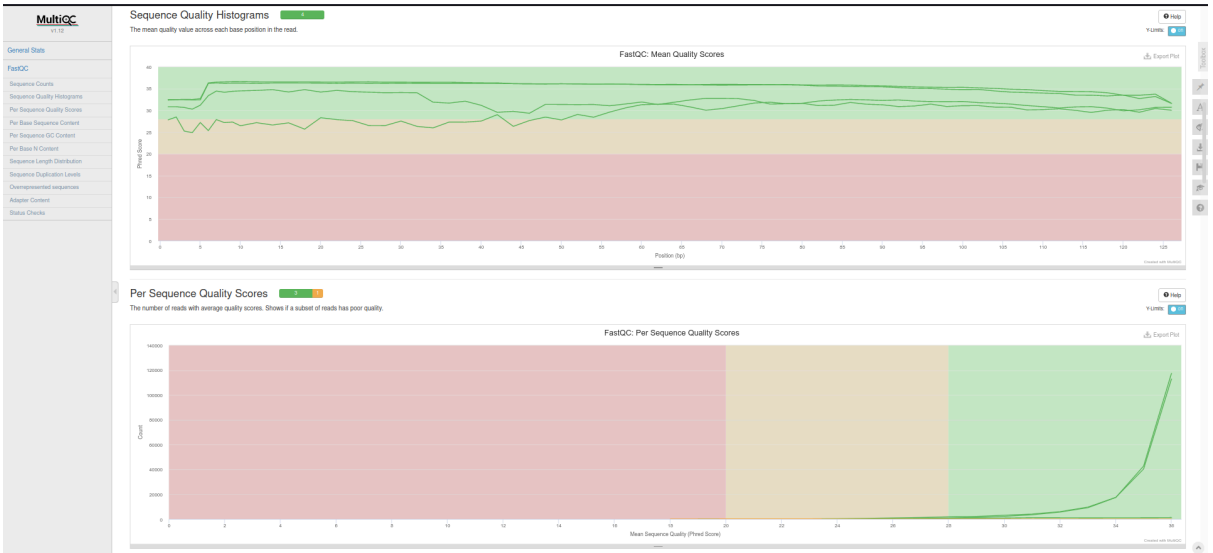
Output:
  /// MultiQC 🔍 | v1.12

```
|         multiqc | Search path : /home/ana/Documents/trimmed_data
|         searching |
```

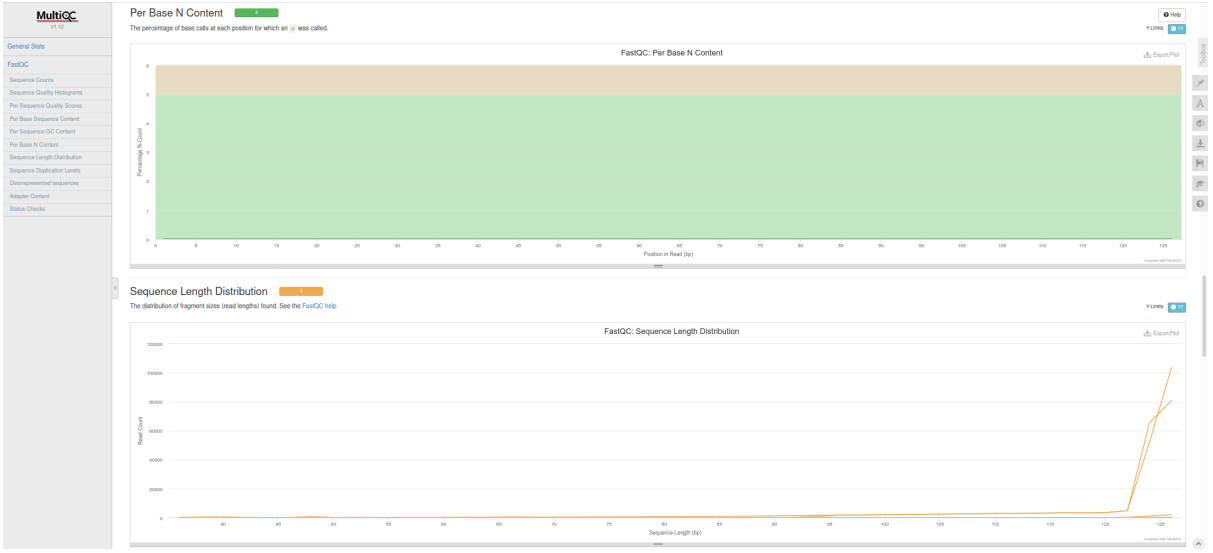
```
100% 13/13
|         fastqc | Found 4 reports
|         multiqc | Compressing plot data
|         multiqc | Previous MultiQC output found! Adjusting filenames..
|         multiqc | Use -f or --force to overwrite existing reports instead
|         multiqc | Report      : multiqc_report_1.html
|         multiqc | Data        : multiqc_data_1
|         multiqc | MultiQC complete
```

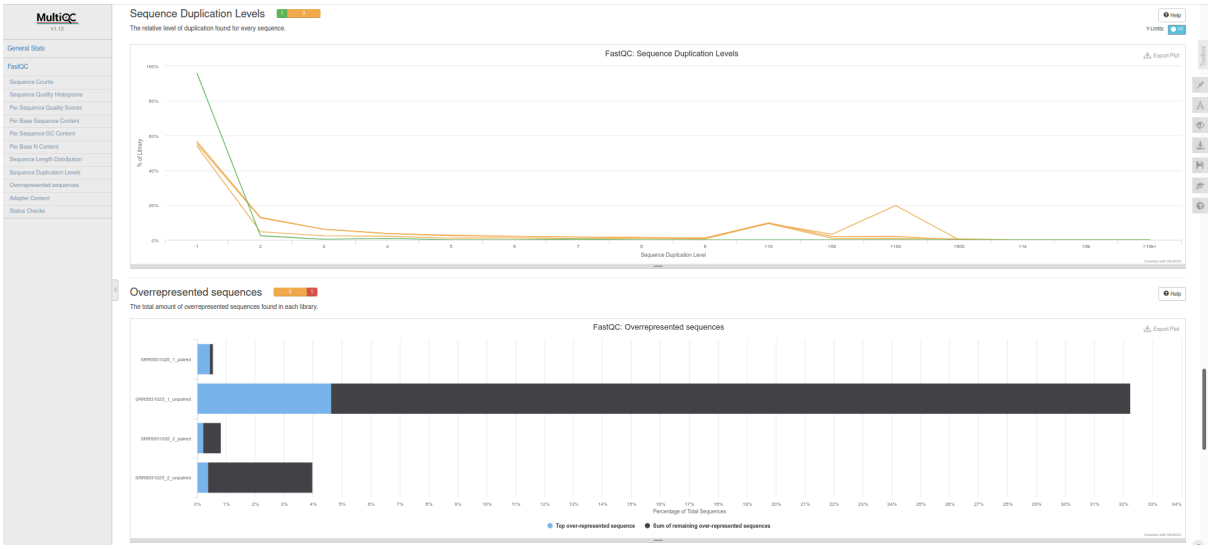
The report:






MultiQC
General Stats
FastQC
Sequence Counts
Sequence Quality Histograms
Per Sequence Quality Scores
Per Base Sequence Content
Per Sequence GC Content
Per Base N Content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Adapter Content
Status Checks
Per Base Sequence Content
The proportion of each base position for which each of the four normal DNA bases has been called.
Click a sample row to see a line plot for that dataset.
Rollover for sample name
Per Sequence GC Content
The average GC content of reads. Normal random library typically have a roughly normal distribution of GC content.
Percentages
Counts
FastQC: Per Sequence GC Content
Percentage
% GC


MultiQC
General Stats
FastQC
Sequence Counts
Sequence Quality Histograms
Per Sequence Quality Scores
Per Base Sequence Content
Per Sequence GC Content
Per Base N Content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Adapter Content
Status Checks
Per Base N Content
The percentage of base calls at each position for which an N was called.
FastQC: Per Base N Content
Percentage N-Count
Position in Read (bp)
Sequence Length Distribution
The distribution of fragment sizes (read lengths) found. See the FastQC help
FastQC: Sequence Length Distribution
Read Count
Sequence Length (bp)


MultiQC
General Stats
FastQC
Sequence Counts
Sequence Quality Histograms
Per Sequence Quality Scores
Per Base Sequence Content
Per Sequence GC Content
Per Base N Content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Adapter Content
Status Checks
Sequence Duplication Levels
The relative level of duplication found for every sequence.
FastQC: Sequence Duplication Levels
% of Library
Sequence Duplication Level
Overrepresented sequences
The total amount of overrepresented sequences found in each library.
FastQC: Overrepresented sequences
Percentage of Total Sequences
Top over-represented sequence
Sum of remaining over-represented sequences

With multiqc we are checking if the quality has improved compared to the fastqc results. Our data now passed Per base sequence quality, Per sequence quality score, and Adapter content. All tests still failed per base sequence content, which is expected as it fails for all RNA-seq data. The unpaired data also failed per sequence GC content, however the following analysis will be performed only on the paired data. Other tests showed improvement.

ALIGNMENT USING STAR

STAR

STAR is a splice-aware mapper (meaning that it aligns RNA-Seq data on genomes of eukaryotes considering introns).

Download GTF or GFF file, which will be used together with a reference genome for creating an index prior to alignment. STAR supports GFF and GTF files. GFF can be converted to GTF with Cufflinks.

GFF - general feature format consists of one line pre feature, each containing 9 columns of data, plus optional track definition lines.

GTF - general transfer format is identical to GFF2

They include:
Fields **must** be tab-separated. Also, all but the final field in each feature line must contain a value; "empty" columns should be denoted with a '.'

1. **seqname** - name of the chromosome or scaffold; chromosome names can be given with or without the 'chr' prefix. **Important note**: the seqname must be one used within Ensembl, i.e. a standard chromosome name or an Ensembl identifier such as a scaffold ID, without any additional content such as species or assembly. See the example GFF output below.
2. **source** - name of the program that generated this feature, or the data source (database or project name)
3. **feature** - feature type name, e.g. Gene, Variation, Similarity
4. **start** - Start position* of the feature, with sequence numbering starting at 1.
5. **end** - End position* of the feature, with sequence numbering starting at 1.
6. **score** - A floating point value.
7. **strand** - defined as + (forward) or - (reverse).
8. **frame** - One of '0', '1' or '2'. '0' indicates that the first base of the feature is the first base of a codon, '1' that the second base is the first base of a codon, and so on..
9. **attribute** - A semicolon-separated list of tag-value pairs, providing additional information about each feature.

For the comparison GFF3 file must include:

Fields must be tab-separated. Also, all but the final field in each feature line must contain a value; "empty" columns should be denoted with a '.'

1. **seqid** - name of the chromosome or scaffold; chromosome names can be given with or without the 'chr' prefix. Important note: the seq ID must be one used within Ensembl, i.e. a standard chromosome name or an Ensembl identifier such as a scaffold ID, without any additional content such as species or assembly. See the example GFF output below.
2. **source** - name of the program that generated this feature, or the data source (database or project name)
3. **type** - type of feature. Must be a term or accession from the SOFA sequence ontology
4. **start** - Start position of the feature, with sequence numbering starting at 1.

5. **end** - End position of the feature, with sequence numbering starting at 1.
6. **score** - A floating point value.
7. **strand** - defined as + (forward) or - (reverse).
8. **phase** - One of '0', '1' or '2'. '0' indicates that the first base of the feature is the first base of a codon, '1' that the second base is the first base of a codon, and so on..
9. **attributes** - A semicolon-separated list of tag-value pairs, providing additional information about each feature.

Download the reference genome and a gtf/gff3 file with wget

```
(nsa) ana@HojansPC:~/nsa_project/genome$ wget
http://ftp.ebi.ac.uk/ensemblgenomes/pub/release-53/plants/gff3/ara
bidopsis_thaliana
```

--2022-05-27 13:07:42--
http://ftp.ebi.ac.uk/ensemblgenomes/pub/release-53/plants/gff3/arabidopsis_thaliana
Resolving ftp.ebi.ac.uk (ftp.ebi.ac.uk)... 193.62.193.138
Connecting to ftp.ebi.ac.uk (ftp.ebi.ac.uk)|193.62.193.138|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location:
http://ftp.ebi.ac.uk/ensemblgenomes/pub/release-53/plants/gff3/arabidopsis_thaliana
/ [following]
--2022-05-27 13:07:42--
http://ftp.ebi.ac.uk/ensemblgenomes/pub/release-53/plants/gff3/arabidopsis_thaliana
/
Reusing existing connection to ftp.ebi.ac.uk:80.
HTTP request sent, awaiting response... 200 OK
Length: 1744 (1,7K) [text/html]
Saving to: ‘arabidopsis_thaliana’

arabidopsis_thaliana
100%[==========
==========>]
1,70K --.-KB/s in 0s

2022-05-27 13:07:42 (74,3 MB/s) - ‘arabidopsis_thaliana’ saved [1744/1744]

```
(nsa) ana@HojansPC:~/nsa_project/genome$ wget
http://ftp.ebi.ac.uk/ensemblgenomes/pub/release-53/plants/gff3/ara
bidopsis_thaliana/Arabidopsis_thaliana.TAIR10.53.gff3.gz
```

--2022-05-27 13:08:26--
http://ftp.ebi.ac.uk/ensemblgenomes/pub/release-53/plants/gff3/arabidopsis_thaliana
/Arabidopsis_thaliana.TAIR10.53.gff3.gz
Resolving ftp.ebi.ac.uk (ftp.ebi.ac.uk)... 193.62.193.138
Connecting to ftp.ebi.ac.uk (ftp.ebi.ac.uk)|193.62.193.138|:80... connected.

HTTP request sent, awaiting response... 200 OK
Length: 9515836 (9,1M) [application/octet-stream]
Saving to: ‘Arabidopsis_thaliana.TAIR10.53.gff3.gz’

Arabidopsis_thaliana.TAIR10.53.gff3.gz
100%[==========>]
9,07M 3,10MB/s in 2,9s

2022-05-27 13:08:29 (3,10 MB/s) - ‘Arabidopsis_thaliana.TAIR10.53.gff3.gz’ saved [9515836/9515836]

Uncompress the files using gunzip:

```
(nsa) ana@HojansPC:~/nsa_project/genome$ gunzip
Arabidopsis_thaliana.TAIR10.dna.toplevel.fa.gz
Arabidopsis_thaliana.TAIR10.53.gff3.gz
```

```
(nsa) ana@HojansPC:~/nsa_project/genome$ ls
```

arabidopsis_thaliana Arabidopsis_thaliana.TAIR10.53.gff3
Arabidopsis_thaliana.TAIR10.dna.toplevel.fa

Create an index:

```
(nsa) ana@HojansPC:~/nsa_project/genome$ STAR --runThreadN 2
--runMode genomeGenerate --genomeDir ../index --genomeFastaFiles
Arabidopsis_thaliana.TAIR10.dna.toplevel.fa --sjdbGTFfile
Arabidopsis_thaliana.TAIR10.53.gff3 --sjdbOverhang 125
```

Tags:

- –runThreadN 2 number of threads
- -runMode genomeGenerate
- --genomeDir directory of output files
- --genomeFastaFiles Our reference genome in fasta format
- --sjdbGTFfile our genome in gtf/gff3 format
- --sjdbOverhang 125 (read length -1)

Output:
STAR --runThreadN 2 --runMode genomeGenerate --genomeDir ../index
--genomeFastaFiles Arabidopsis_thaliana.TAIR10.dna.toplevel.fa --sjdbGTFfile
Arabidopsis_thaliana.TAIR10.53.gff3 --sjdbOverhang 125

STAR version: 2.7.10a compiled: 2022-01-14T18:50:00-05:00
:/home/dobin/data/STAR/STARcode/STAR.master/source
May 27 15:28:35 ..... started STAR run
May 27 15:28:35 ... starting to generate Genome files
May 27 15:28:37 ..... processing annotations GTF
!!!!! WARNING: --genomeSAindexNbases 14 is too large for the genome size=119667750, which may cause seg-fault at the mapping step. Re-run genome generation with recommended --genomeSAindexNbases 12
May 27 15:28:40 ... starting to sort Suffix Array. This may take a long time...
May 27 15:28:40 ... sorting Suffix Array chunks and saving them to disk...
May 27 15:30:51 ... loading chunks from disk, packing SA...
May 27 15:30:54 ... finished generating suffix array
May 27 15:30:54 ... generating Suffix Array index
May 27 15:31:50 ... completed Suffix Array index
May 27 15:31:50 ... writing Genome to disk ...
May 27 15:31:51 ... writing Suffix Array to disk ...
May 27 15:31:51 ... writing SAindex to disk
May 27 15:31:52 ..... finished successfully

First mapping:

Create a new directory and run star.

```
(nsa) ana@HojansPC:~/nsa_project$ mkdir star_al
(nsa) ana@HojansPC:~/nsa_project$ STAR --runThreadN 20 --genomeDir index
--readFilesIn trimmed_data/SRR5831025_1_paired_trimmed.fastq
trimmed_data/SRR5831025_2_paired_trimmed.fastq --outFileNamePrefix
SRR5831025_
```

Tags:

- –runThreadN 2 number of threads
- --genomeDir directory of our index
- --readFilesIn our paired trimmed data in fastq format
- --outNamePrefix prefix

Output:

STAR --runThreadN 20 --genomeDir index --readFilesIn
trimmed_data/SRR5831025_1_paired_trimmed.fastq
trimmed_data/SRR5831025_2_paired_trimmed.fastq --outFileNamePrefix SRR5831025_
STAR version: 2.7.10a compiled: 2022-01-14T18:50:00-05:00
:/home/dobin/data/STAR/STARcode/STAR.master/source
May 27 15:41:25 ..... started STAR run
May 27 15:41:25 ..... loading genome
May 27 15:41:26 ..... started mapping
May 27 15:41:41 ..... finished mapping
May 27 15:41:42 ..... finished successfully

Log file:

```
                        Started job on |    May 27 15:41:25
                        Started mapping on |    May 27 15:41:26
                        Finished on |    May 27 15:41:42
        Mapping speed, Million of reads per hour |    45.32

                Number of input reads |    201409
                Average input read length |    240
                        UNIQUE READS:
                Uniquely mapped reads number |    190340
                Uniquely mapped reads % |    94.50%
                Average mapped length |    239.28
                Number of splices: Total |    115122
        Number of splices: Annotated (sjdb) |    0
                Number of splices: GT/AG |    113827
                Number of splices: GC/AG |    973
                Number of splices: AT/AC |    23
        Number of splices: Non-canonical |    299
                Mismatch rate per base, % |    0.12%
                Deletion rate per base |    0.00%
                Deletion average length |    1.36
                Insertion rate per base |    0.00%
                Insertion average length |    1.13
                        MULTI-MAPPING READS:
        Number of reads mapped to multiple loci |    5031
        % of reads mapped to multiple loci |    2.50%
        Number of reads mapped to too many loci |    82
        % of reads mapped to too many loci |    0.04%
                        UNMAPPED READS:
  Number of reads unmapped: too many mismatches |    0
        % of reads unmapped: too many mismatches |    0.00%
        Number of reads unmapped: too short |    5951
                % of reads unmapped: too short |    2.95%
                Number of reads unmapped: other |    5
                % of reads unmapped: other |    0.00%
                        CHIMERIC READS:
                Number of chimeric reads |    0
                        % of chimeric reads |    0.00%
```

Second mapping:

```
(nsa) ana@HojansPC:~/nsa_project/2_mapping$ STAR --runThreadN 20
--genomeDir ../index --readFilesIn
../trimmed_data/SRR5831025_1_paired_trimmed.fastq
```

```
../trimmed_data/SRR5831025_2_paired_trimmed.fastq --outFileNamePrefix
SRR5831025_ --sjdbFileChrStartEnd ../star_al/SRR5831025_SJ.out.tab
```

Tags:

- –runThreadN 2 number of threads
- --genomeDir directory of our index
- --readFilesIn our paired trimmed data in fastq format
- --outNamePrefix prefix
- --sjdbFileChrStartEnd path to the files with genomic coordinates (chr start end strand) for the splice junction introns. Multiple files can be supplied wand will be concatenated.

Output:

```
  STAR --runThreadN 20 --genomeDir ../index --readFilesIn
../trimmed_data/SRR5831025_1_paired_trimmed.fastq
../trimmed_data/SRR5831025_2_paired_trimmed.fastq --outFileNamePrefix SRR5831025_
--sjdbFileChrStartEnd ../star_al/SRR5831025_SJ.out.tab
   STAR version: 2.7.10a   compiled: 2022-01-14T18:50:00-05:00
:/home/dobin/data/STAR/STARcode/STAR.master/source
May 27 15:47:07 ..... started STAR run
May 27 15:47:07 ..... loading genome
May 27 15:47:08 ..... inserting junctions into the genome indices
May 27 15:47:34 ..... started mapping
May 27 15:47:49 ..... finished mapping
May 27 15:47:49 ..... finished successfully
```

Log file:

```
                        Started job on |    May 27 15:47:07
                        Started mapping on |    May 27 15:47:34
                        Finished on |    May 27 15:47:49
        Mapping speed, Million of reads per hour |    48.34

                Number of input reads |    201409
                Average input read length |    240
                        UNIQUE READS:
                Uniquely mapped reads number |    190413
                Uniquely mapped reads % |    94.54%
                Average mapped length |    239.55
                Number of splices: Total |    126731
        Number of splices: Annotated (sjdb) |    125506
                Number of splices: GT/AG |    125270
                Number of splices: GC/AG |    1138
                Number of splices: AT/AC |    32
        Number of splices: Non-canonical |    291
                Mismatch rate per base, % |    0.11%
                Deletion rate per base |    0.00%
                Deletion average length |    1.36
                Insertion rate per base |    0.00%
                Insertion average length |    1.13
                        MULTI-MAPPING READS:
```

```
        Number of reads mapped to multiple loci |    5012
        % of reads mapped to multiple loci |    2.49%
        Number of reads mapped to too many loci |    69
        % of reads mapped to too many loci |    0.03%
                        UNMAPPED READS:
  Number of reads unmapped: too many mismatches |    0
        % of reads unmapped: too many mismatches |    0.00%
        Number of reads unmapped: too short |    5910
                % of reads unmapped: too short |    2.93%
                Number of reads unmapped: other |    5
                % of reads unmapped: other |    0.00%
                        CHIMERIC READS:
                Number of chimeric reads |    0
                        % of chimeric reads |    0.00%
```

Converting sam file to a bam file with samtools:

```
(nsa) ana@HojansPC:~/nsa_project/2_mapping$ samtools view -b
SRR5831025_Aligned.out.sam -o SRR5831025_Aligned.out.bam
```

Tags:

- -b format of the output file
- -o name of the output file

Sorting bam file with samtools

```
(nsa) ana@HojansPC:~/nsa_project/2_mapping$ samtools sort
SRR5831025_Aligned.out.bam  -o SRR5831025_Aligned_Sorted.out.bam
```

Generate index of the reference genome with samtools

Faidx is used to create an index of the reference sequence in the FASTA format or extract subsequence from indexed reference sequence. If no region is specified, **faidx** will index the file and create *<ref.fasta>.fai* on the disk. If regions are specified, the subsequences will be retrieved and printed to stdout in the FASTA format.

```
(nsa) ana@HojansPC:~/nsa_project/genome$ samtools faidx
Arabidopsis_thaliana.TAIR10.dna.toplevel.fa
```

Generate index from the bam file

```
(nsa) ana@HojansPC:~/nsa_project/2_mapping$ samtools index
SRR5831025_Aligned_Sorted.out.bam
```

Visualisation with tablet

With the tool tablet analyse the results with the settings

- Primary assembly: 2_mapping/SRR5831025_Aligned_Sorted.out.bam

- Assembly reference: genome/Arabidopsis_thaliana.TAIR10.dna.toplevel.fa
- Import features: genome/Arabidopsis_thaliana.TAIR10.53.gff3

Assembly summary
Description Value
Total number of contigs 7
Average contig length 17,095,392
Total number of reads 404,251
Average reads per contig 57,750
N50 23,459,830
N90 18,585,056
Assembly file SRR5831025_Aligned_Sorted.out.bam
Assembly file size 20 MB
Reference file Arabidopsis_thaliana.TAIR10.dna.toplevel.fa
Reference file size 116 MB

Ch: 2

In the middle we see a single substitution, which is more likely due to a sequencing error.

Star alignment accounts for introns thus long reads of N are produced.

3 reads show guanine and two thymine.

We have come across a SNP- in the reference genome we have a thymine, howeverer in the reads we have cytosine. We have another 2 SNP where adenine is mutated to thymine

NUMBER OF READS PER GENE USING FEATURECOUNTS

Bioinformatic tool featureCounts will be used for counting the aligned reads per gene (htseq-count (https://htseq.readthedocs.io/en/release_0.11.1/count.html) is also frequently used).

FeatureCounts is a part of subread package.
http://subread.sourceforge.net/

Creating binary file of gff/gtf genome

Befoure featurecounts we will need to convert our genome-gff3 format to a bed file, with tool bedops.

```
(new_nsa) ana@HojansPC:~/nsa_project/genome$ gff2bed <
Arabidopsis_thaliana.TAIR10.53.gff3 > Arabidopsis_thaliana.TAIR10.53.bed
```

or:

```
(new_nsa) ana@HojansPC:~/nsa_project/genome$  awk '{ if ($0 ~
```

```
"transcript_id") print $0; else print $0" transcript_id \"\";"; }'
Arabidopsis_thaliana.TAIR10.53.gff3 | gff2bed - > output.bed
```

Strand-specific testing with RSeQC:

Before running the featureCounts it should be checked if your dataset is strand-specific or not. You can do this with infer_experiment.py (part of RSeQC package, http://rseqc.sourceforge.net/#).

Because instaling RSeQC did not work on our conda environment, thus we installed it in a new conda enviroment.:

```
(new_nsa) ana@HojansPC:~$ pip3 install RSeQC
```

```
(new_nsa) ana@HojansPC:~/nsa_project/2_mapping$ infer_experiment.py -r
../genome/output.bed -i SRR5831025_Aligned_Sorted.out.bam
```

Output:
Reading reference gene model ../genome/output.bed ... Done
Loading SAM/BAM file ... Total 200000 usable reads were sampled

Results of infer_experiment.py
This is PairEnd Data
Fraction of reads failed to determine: 0.0398
Fraction of reads explained by "1++,1--,2+-,2-+": 0.0238
Fraction of reads explained by "1+-,1-+,2++,2--": 0.9365

The output will be either: This is PairEnd Data or This is SingleEnd Data
If the ratio of "1++,1--,2+-,2-+" and "1+-,1-+,2++,2--" is relatively the same then we have a non strand specific dataset. Results like ours or other way around conclude a strand specific dataset:

For pair-end RNA-seq, there are two different ways to strand reads (such as Illumina ScriptSeq protocol):

1. 1++,1–,2+-,2-+
- read1 mapped to '+' strand indicates parental gene on '+' strand
- read1 mapped to '-' strand indicates parental gene on '-' strand
- read2 mapped to '+' strand indicates parental gene on '-' strand
- read2 mapped to '-' strand indicates parental gene on '+' strand

2. 1+-,1-+,2++,2–
- read1 mapped to '+' strand indicates parental gene on '-' strand
- read1 mapped to '-' strand indicates parental gene on '+' strand
- read2 mapped to '+' strand indicates parental gene on '+' strand
- read2 mapped to '-' strand indicates parental gene on '-' strand

For single-end RNA-seq, there are also two different ways to strand reads:

1. ++,–
- read mapped to ‘+’ strand indicates parental gene on ‘+’ strand
- read mapped to ‘-‘ strand indicates parental gene on ‘-‘ strand

2. +-,-+
- read mapped to ‘+’ strand indicates parental gene on ‘-‘ strand
- read mapped to ‘-‘ strand indicates parental gene on ‘+’ strand

Number of reads per gene with featureCounts

Using featureCounts to count number of reads per gene:
We found problems with gff3 file, thus we downloaded the gtf file.

```
(nsa) ana@HojansPC:~/nsa_project/genome$ featureCounts -p -s 2 -O -a
Arabidopsis_thaliana.TAIR10.53.gtf -o featureCounts_S1.tsv
../2_mapping/SRR5831025_Aligned_Sorted.out.bam
```

Tags:
- -p Specify that input data contain paired-end reads
- -s 2 Indicate if strand-specific read counting should be performed. A single integer value (applied to all input files) or a string of comma-separated values (applied to each corresponding input file) should be provided. Possible values include:
    - 0 (unstranded),
    - 1 (stranded) and
    - 2 (reversely stranded).
- -O If specified, reads (or fragments) will be allowed to be assigned to more than one matched meta-feature (or feature if -f is specified). Reads/fragments overlapping with more than one meta-feature/feature will be counted more than once
- -a Provide name of an annotation file
- -o Give the name of the output file. The output file contains the number of reads assigned to each meta-feature

Output:

```
|| Load annotation file Arabidopsis_thaliana.TAIR10.53.gtf ...                 ||
||         Features : 313952                                                   ||
||         Meta-features : 32833                                               ||
||         Chromosomes/contigs : 7                                             ||
||                                                                             ||
|| Process BAM file SRR5831025_Aligned_Sorted.out.bam...                       ||
||         Strand specific : reversely stranded                                ||
||         Paired-end reads are included.                                      ||
||         Total alignments : 202232                                           ||
||         Successfully assigned alignments : 185213 (91.6%)                   ||
||         Running time : 0.01 minutes                                         ||
||                                                                             ||
|| Write the final count table.                                                ||
|| Write the read assignment summary.                                          ||
||                                                                             ||
||Summary of counting results can be found in file "featureCounts_S2.tsv.su    ||
|| mmary"                                                                      ||
||                                                                             ||
```

VARIANT DISCOVERY WITH GATK

The aim of this exercise is to identify SNPs in the RNA-Seq dataset with the Genome Analysis Toolkit (GATK).

Download GATK

```
(nsa) ana@HojansPC:~/gatk$ wget
https://github.com/broadinstitute/gatk/releases/download/4.2.6.1/gatk-4.
2.6.1.zip
```

Create a conda environment with gatk:

```
(nsa) ana@HojansPC:~/gatk/gatk-4.2.6.1$ conda env create -f
gatkcondaenv.yml
```

GATK AddOrReplaceReadGroups

Many tools (Picard and GATK for example) require or assume the presence of at least one RG tag, defining a "read-group" to which each read can be assigned (as specified in the RG tag in the SAM record). This tool enables the user to assign all the reads in the #INPUT to a single new read-group. Read groups are a set of reads that are generated from a single run of a sequencing instrument.

```
(gatk) ana@HojansPC:~/nsa_project/2_mapping$
./../../gatk/gatk-4.2.6.1/gatk AddOrReplaceReadGroups -I
SRR5831025_Aligned_Sorted.out.bam -O gatk_SRR5831025_RG.bam -RGID 1
-RGLB 1 -RGPL ILLUMINA -RGPU unit1 -RGSM Arabidopsis_control_1
```

Tags:

- -I input file
- -O output file
- -RGID 1 (Read-Group id)
- -RGLB 1 (Read-Group library)
- -RGPL ILLUMINA (Read-Group platform)
- -RGPU unit1 (Read-Group platform unit (eg. run barcode))
- -RGSM (Read-Group sample name)

Output:
Using GATK jar /home/ana/gatk/gatk-4.2.6.1/gatk-package-4.2.6.1-local.jar

```
(nsa) ana@HojansPC:~/nsa_project/2_mapping$ samtools view -h
gatk_SRR5831025_RG.bam | less
```

GATK MarkDuplicates:

```
(gatk) ana@HojansPC:~/nsa_project/2_mapping$
./../../gatk/gatk-4.2.6.1/gatk MarkDuplicates -I gatk_SRR5831025_RG.bam
-O gatk_SRR5831025_RG_MD.bam -M
../../gatk/gatk-4.2.6.1/mark_duplicates.txt
```

Tags:

- -I input file
- -O output file
- -M Additionally produce estimated library complexity metrics

Output:
Using GATK jar /home/ana/gatk/gatk-4.2.6.1/gatk-package-4.2.6.1-local.jar

Create Sequence Dictionary

```
(gatk) ana@HojansPC:~/nsa_project/2_mapping$
```

```
./../../gatk/gatk-4.2.6.1/gatk CreateSequenceDictionary -R
../genome/Arabidopsis_thaliana.TAIR10.dna.toplevel.fa
```

Tags:

- -R reference

SplitNCigarReads

This step splits reads with N in the cigar into multiple supplementary alignments and hard clips mismatching overhangs. By default this step also reassigns mapping qualities for good alignments to match DNA conventions.

```
(gatk) ana@HojansPC:~/nsa_project/2_mapping$
./../../gatk/gatk-4.2.6.1/gatk SplitNCigarReads -R
../genome/Arabidopsis_thaliana.TAIR10.dna.toplevel.fa -I
gatk_SRR5831025_RG_MD.bam -O SRR5831025_RG_MD_splited.bam
```

Tags:

- -R reference genome
- -I input file: our bam file produced in the markduplicates
- -O output file

Output:
Using GATK jar /home/ana/gatk/gatk-4.2.6.1/gatk-package-4.2.6.1-local.jar
Running:
java -Dsamjdk.use_async_io_read_samtools=false
-Dsamjdk.use_async_io_write_samtools=true -Dsamjdk.use_async_io_write_tribble=false
-Dsamjdk.compression_level=2 -jar
/home/ana/gatk/gatk-4.2.6.1/gatk-package-4.2.6.1-local.jar SplitNCigarReads -R
../genome/Arabidopsis_thaliana.TAIR10.dna.toplevel.fa -I gatk_SRR5831025_RG_MD.bam
-O SRR5831025_RG_MD_splited.bam
15:43:25.730 INFO NativeLibraryLoader - Loading libgkl_compression.so from
jar:file:/home/ana/gatk/gatk-4.2.6.1/gatk-package-4.2.6.1-local.jar!/com/intel/gkl/native/libgkl_
compression.so
15:43:25.892 INFO SplitNCigarReads - --
15:43:25.892 INFO SplitNCigarReads - The Genome Analysis Toolkit (GATK) v4.2.6.1
15:43:25.892 INFO SplitNCigarReads - For support and documentation go to
https://software.broadinstitute.org/gatk/
15:43:25.892 INFO SplitNCigarReads - Executing as ana@HojansPC on Linux
v5.13.0-44-generic amd64
15:43:25.892 INFO SplitNCigarReads - Java runtime: OpenJDK 64-Bit Server VM
v17.0.3+7-Ubuntu-0ubuntu0.20.04.1
15:43:25.893 INFO SplitNCigarReads - Start Date/Time: 30 May 2022 at 15:43:25 CEST
15:43:25.893 INFO SplitNCigarReads - --
15:43:25.893 INFO SplitNCigarReads - --
15:43:25.893 INFO SplitNCigarReads - HTSJDK Version: 2.24.1
15:43:25.894 INFO SplitNCigarReads - Picard Version: 2.27.1
15:43:25.894 INFO SplitNCigarReads - Built for Spark Version: 2.4.5
15:43:25.894 INFO SplitNCigarReads - HTSJDK Defaults.COMPRESSION_LEVEL : 2
15:43:25.894 INFO SplitNCigarReads - HTSJDK
Defaults.USE_ASYNC_IO_READ_FOR_SAMTOOLS : false

```
15:43:25.894 INFO  SplitNCigarReads - HTSJDK
Defaults.USE_ASYNC_IO_WRITE_FOR_SAMTOOLS : true
15:43:25.894 INFO  SplitNCigarReads - HTSJDK
Defaults.USE_ASYNC_IO_WRITE_FOR_TRIBBLE : false
15:43:25.894 INFO  SplitNCigarReads - Deflater: IntelDeflater
15:43:25.894 INFO  SplitNCigarReads - Inflater: IntelInflater
15:43:25.894 INFO  SplitNCigarReads - GCS max retries/reopens: 20
15:43:25.895 INFO  SplitNCigarReads - Requester pays: disabled
15:43:25.895 INFO  SplitNCigarReads - Initializing engine
15:43:26.033 INFO  SplitNCigarReads - Done initializing engine
15:43:26.058 INFO  ProgressMeter - Starting traversal
15:43:26.059 INFO  ProgressMeter -            Current Locus  Elapsed Minutes       Reads
Processed      Reads/Minute
15:43:36.430 INFO  ProgressMeter -            1:23486710              0.2               86000
        497589.2
15:43:46.437 INFO  ProgressMeter -            2:13139098              0.3              151000
        444597.1
15:43:56.569 INFO  ProgressMeter -            3:21199028              0.5              248000
        487724.9
15:44:06.578 INFO  ProgressMeter -            5:17843435              0.7              356000
        527173.1
15:44:14.297 WARN  IntelInflater - Zero Bytes Written : 0
15:44:14.298 INFO  SplitNCigarReads - 0 read(s) filtered by: AllowAllReadsReadFilter

15:44:14.317 INFO  OverhangFixingManager - Overhang Fixing Manager saved 136 reads
in the first pass
15:44:14.318 INFO  SplitNCigarReads - Starting traversal pass 2
15:44:16.667 INFO  ProgressMeter -            1:8398160               0.8              443000
        525213.4
15:44:27.188 INFO  ProgressMeter -            1:26625752              1.0              506000
        496662.7
15:44:37.206 INFO  ProgressMeter -            2:17893563              1.2              581000
        489971.5
15:44:47.258 INFO  ProgressMeter -            4:9557314               1.4              682000
        503947.1
15:44:57.280 INFO  ProgressMeter -            5:19098056              1.5              763000
        501863.6
15:45:06.067 WARN  IntelInflater - Zero Bytes Written : 0
15:45:06.067 INFO  SplitNCigarReads - 0 read(s) filtered by: AllowAllReadsReadFilter

15:45:06.067 INFO  ProgressMeter -             Pt:137317              1.7              808502
        485062.4
15:45:06.067 INFO  ProgressMeter - Traversal complete. Processed 808502 total reads in
1.7 minutes.
15:45:07.140 INFO  SplitNCigarReads - Shutting down engine
[30 May 2022 at 15:45:07 CEST]
org.broadinstitute.hellbender.tools.walkers.rnaseq.SplitNCigarReads done. Elapsed time:
1.69 minutes.
Runtime.totalMemory()=408944640
```

Base Quality Recalibration

For this step a vcf file with known mutations is required. It is available on the ENSEMBL database.

```
(base) ana@HojansPC:~/nsa_project/genome$ wget
http://ftp.ebi.ac.uk/ensemblgenomes/pub/release-53/plants/variation/vcf/
arabidopsis_thaliana/arabidopsis_thaliana.vcf.gz
```

UNZIPPING:

```
(base) ana@HojansPC:~/nsa_project/genome$ gunzip
arabidopsis_thaliana.vcf.gz
(base) ana@HojansPC:~/nsa_project/genome$ cat arabidopsis_thaliana.vcf |
sed 's/The 1001 Genomes Project_2016/The_1001_Genomes_Project_2016/g' >
arabidopsis_thaliana_no_spaces.vcf
(base) ana@HojansPC:~/nsa_project/genome$ awk '{if(NR!=1687355){print
$0}}' arabidopsis_thaliana_no_spaces.vcf >
arabidopsis_thaliana_no_spaces_v2.vcf
```

Create vcf index file:

```
(gatk) ana@HojansPC:~/nsa_project/genome$ ./../../gatk/gatk-4.2.6.1/gatk
IndexFeatureFile -I arabidopsis_thaliana_no:spaces_v2.vcf
```

BaseRecalibrator

```
(gatk) ana@HojansPC:~/nsa_project/genome$ ./../../gatk/gatk-4.2.6.1/gatk
BaseRecalibrator -I ../2_mapping/gatk_SRR5831025_RG_MD.bam -R
./Arabidopsis_thaliana.TAIR10.dna.toplevel.fa --known-sites
arabidopsis_thaliana_no_spaces_v2.vcf -O
../2_mapping/gatk_recall_data.table
```

Tags:

- -I input file
- -R reference our genome in fasta format
- --known-sites our vcf file
- -O output file

Apply recalibration

```
(gatk) ana@HojansPC:~/nsa_project/genome$ ./../../gatk/gatk-4.2.6.1/gatk
ApplyBQSR -R ./Arabidopsis_thaliana.TAIR10.dna.toplevel.fa -I
../2_mapping/SRR5831025_RG_MD_splited.bam --bqsr-recal-file
../2_mapping/gatk_recall_data.table -O
../2_mapping/gatk_SRR5831025_sorted_RG_MD_splitted_recalibrated.bam
```

Tags:

- -R reference our genome in fasta format
- -I input file
- --bqsr-recal-file Input recalibration table for BQSR

- -O output file

Variant Calling

```
(gatk) ana@HojansPC:~/nsa_project/2_mapping$
./../../gatk/gatk-4.2.6.1/gatk HaplotypeCaller -R
../genome/Arabidopsis_thaliana.TAIR10.dna.toplevel.fa -I
SRR5831025_RG_MD_splited.bam -O SRR5831025_RG_MD_splited.vcf -bamout
bamout.bam
```

Tags:

- -R reference our genome in fasta format
- -I input file
- -O output file
- -bamout create another output file in bam format

Output:
java -Dsamjdk.use_async_io_read_samtools=false
-Dsamjdk.use_async_io_write_samtools=true -Dsamjdk.use_async_io_write_tribble=false
-Dsamjdk.compression_level=2 -jar
/home/ana/gatk/gatk-4.2.6.1/gatk-package-4.2.6.1-local.jar HaplotypeCaller -R
../genome/Arabidopsis_thaliana.TAIR10.dna.toplevel.fa -I
SRR5831025_RG_MD_splited.bam -O SRR5831025_RG_MD_splited.vcf -bamout
bamout.bam

Variant calling of out recalibrated bam file:

```
(gatk) ana@HojansPC:~/nsa_project/genome$ ./../../gatk/gatk-4.2.6.1/gatk
HaplotypeCaller -R ./Arabidopsis_thaliana.TAIR10.dna.toplevel.fa -I
../2_mapping/gatk_SRR5831025_sorted_RG_MD_splitted_recalibrated.bam -O
output.vcf.gz -bamout vamout.bam
```

VCF file

The Variant Call Format (VCF) specifies the format of a text file used for storing gene sequence variations. The header begins the file and provides metadata describing the body of the file. Header lines are denoted as starting with #. Special keywords in the header are denoted with ##. Recommended keywords include fileformat, fileDate and reference. The body of VCF follows the header, and is tab separated into 8 mandatory columns and an unlimited number of optional columns.

Name	Brief description (see the specification for details).	
1	CHROM	The name of the sequence (typically a chromosome) on which the variation is being called. This sequence is usually known as 'the reference sequence', i.e. the sequence against which the given sample varies.
2	POS	The 1-based position of the variation on the given sequence.
3	ID	The identifier of the variation, e.g. a dbSNP rs identifier, or if unknown a ".". Multiple identifiers should be separated by semi-colons without white-space.
4	REF	The reference base (or bases in the case of an indel) at the given position on the given reference sequence.
5	ALT	The list of alternative alleles at this position.
6	QUAL	A quality score associated with the inference of the given alleles.
7	FILTER	A flag indicating which of a given set of filters the variation has failed or PASS if all the filters were passed successfully.
8	INFO	An extensible list of key-value pairs (fields) describing the variation. See below for some common fields. Multiple fields are separated by semicolons with optional values in the format: `<key>=<data>[,data]`.
9	FORMAT	An (optional) extensible list of fields for describing the samples. See below for some common fields.
+	SAMPLEs	For each (optional) sample described in the file, values are given for the fields listed in FORMAT

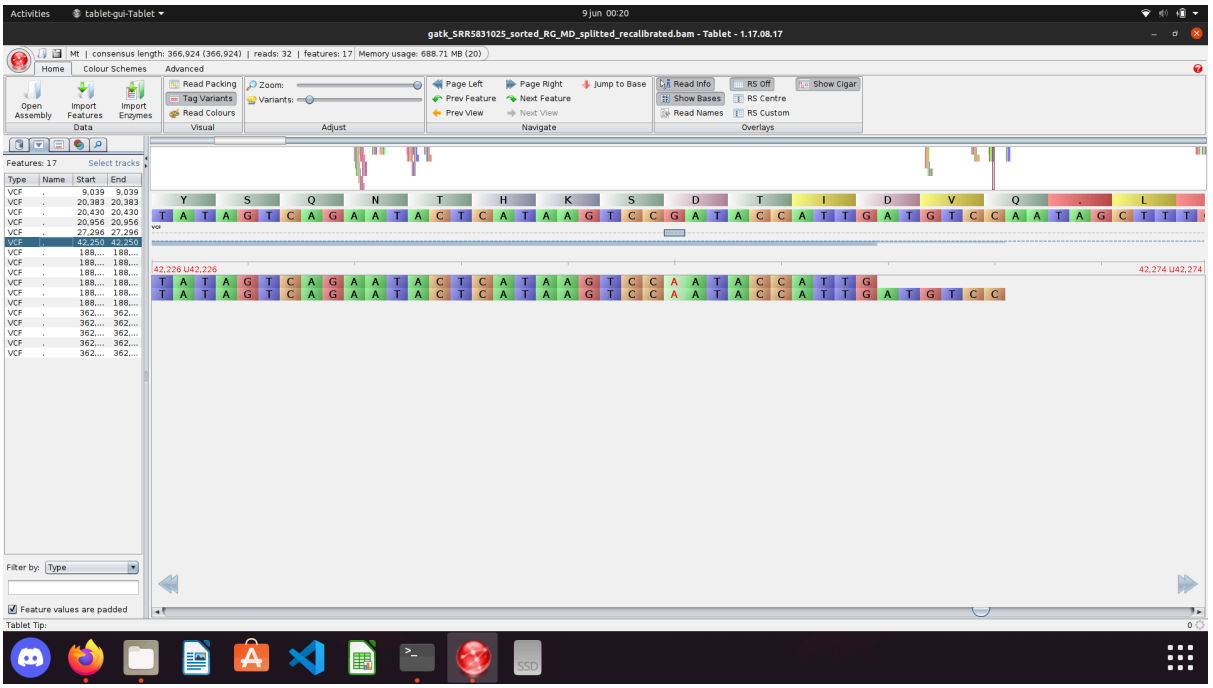
Examine SRR5831025_sorted_RG_MD_splitted_recalibrated.bam with Tablet and based on vcf file identify at least one SNP. Provide a screenshot.

Visualisation with tablet:

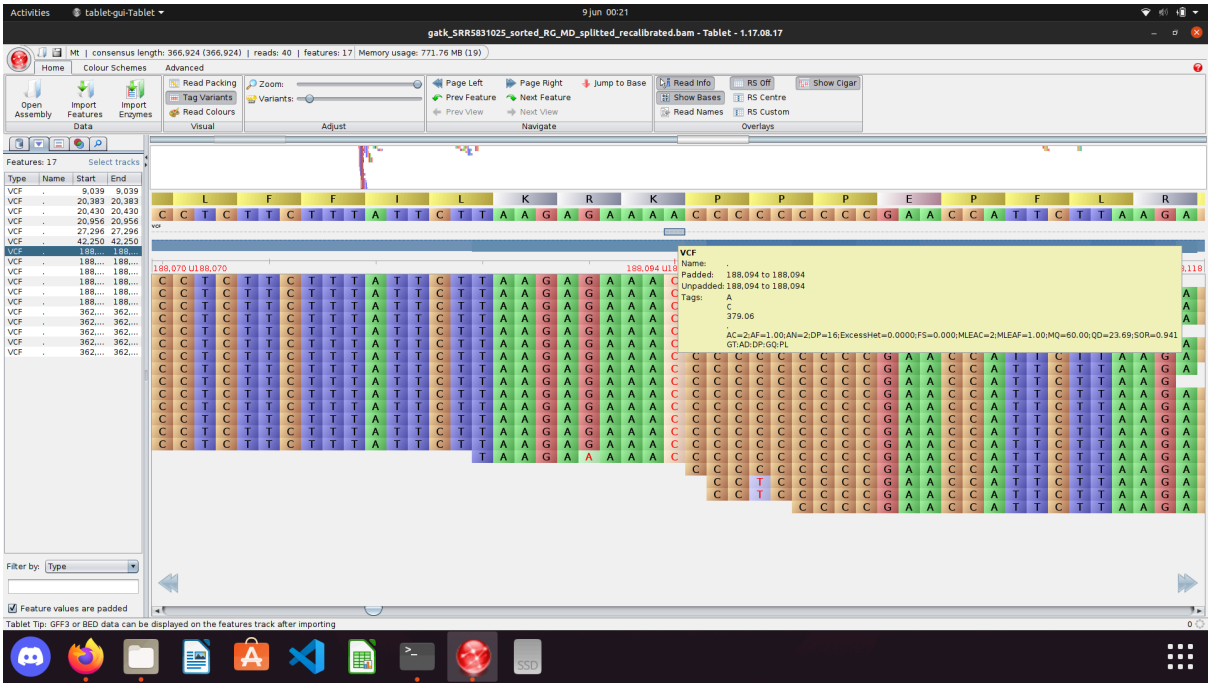
Tablet input files:

- Primary assembly: gatk_SRR5831025_sorted_RG_MD_splitted_recalibrated.bam
- Reference genome: Arabidopsis_thaliana.TAIR10.dna.toplevel.fa
- Import features: arabidopsis_thaliana_no_spaces_v2.vcf

SNP:

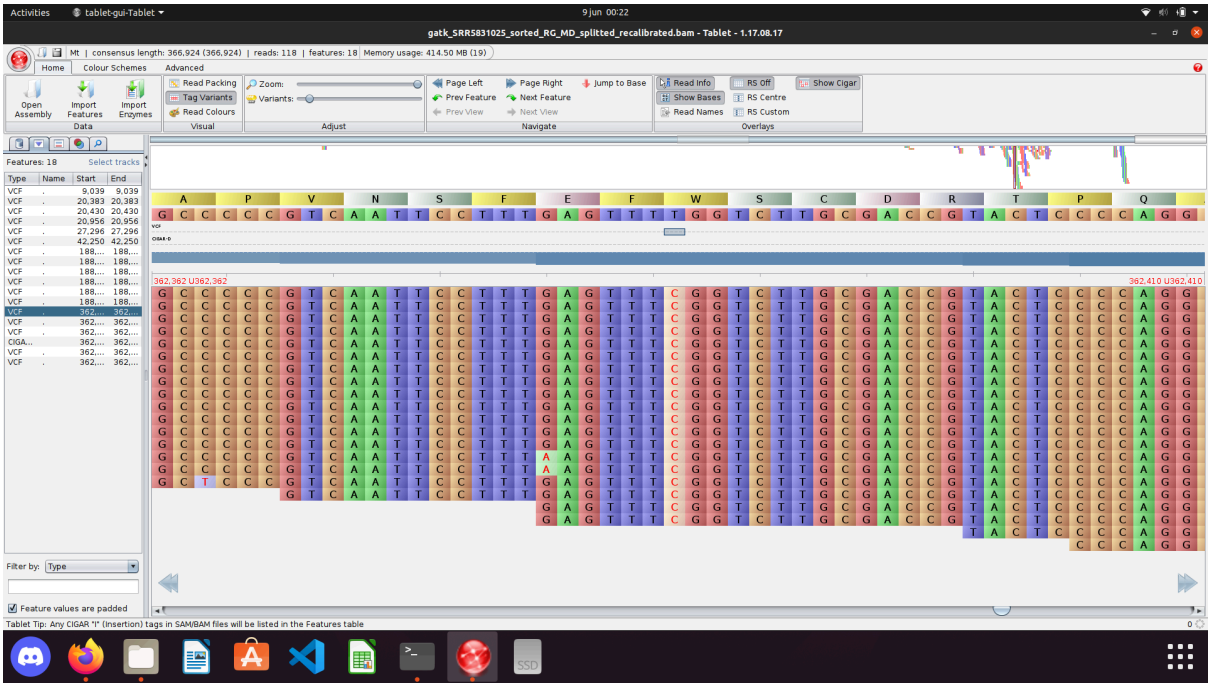


Known variation from guanine to adenine on Mt.



Known SNP from adenine to cytosine.

Know SNP Guanine to Adenine in two spots.



Known SNP from thymine to cytosine.

Known deletion from TC to T, and known deletion from AC to A.

CREATING A PIPELINE WITH SNAKEMAKE:

Dry run of our snakefile:

```
SAMPLES = ["SRR5831025", "SRR5831026"]
strand = [1, 2]

rule all:
       input:
       expand("fastqc/{sample}_{strand}_fastqc.html", sample=SAMPLES, strand=strand)s

rule fastqc:
       input:
       "{sample}.fastq"
       output:
       html = "fastqc/{sample}_{strand}_fastqc.html",
       zip = "fastqc/{sample}_{strand}_fastqc.zip"
       conda:
       "envs/quality_check.yaml"
       shell:
       '''
       fastqc {input} &&
       mv {wildcards.sample}_{wildcards.strand}_fastqc.zip {output.zip} &&
       mv {wildcards.sample}_{wildcards.strand}_fastqc.html {output.html}
       '''
```

```
(snakemake) ana@HojansPC:~/nsa_project$ snakemake -np
```

Output:
Building DAG of jobs...
Job stats:

job	count	min threads	max threads
all	1	1	1
fastqc	2	1	1
total	3	1	1

[Mon Jun 6 14:30:55 2022]
rule fastqc:
    input: SRR5831026.fastq
    output: fastqc/SRR5831026_1_fastqc.html, fastqc/SRR5831026_1_fastqc.zip
    jobid: 3
    reason: Missing output files: fastqc/SRR5831026_1_fastqc.html
    wildcards: sample=SRR5831026, strand=1
    resources: tmpdir=/tmp

    fastqc SRR5831026.fastq &&
    mv SRR5831026_1_fastqc.zip fastqc/SRR5831026_1_fastqc.zip &&

```
        mv SRR5831026_1_fastqc.html fastqc/SRR5831026_1_fastqc.html


[Mon Jun  6 14:30:55 2022]
rule fastqc:
        input: SRR5831026.fastq
        output: fastqc/SRR5831026_2_fastqc.html, fastqc/SRR5831026_2_fastqc.zip
        jobid: 4
        reason: Missing output files: fastqc/SRR5831026_2_fastqc.html
        wildcards: sample=SRR5831026, strand=2
        resources: tmpdir=/tmp


        fastqc SRR5831026.fastq &&
        mv SRR5831026_2_fastqc.zip fastqc/SRR5831026_2_fastqc.zip &&
        mv SRR5831026_2_fastqc.html fastqc/SRR5831026_2_fastqc.html


[Mon Jun  6 14:30:55 2022]
localrule all:
        input: fastqc/SRR5831025_1_fastqc.html, fastqc/SRR5831025_2_fastqc.html,
fastqc/SRR5831026_1_fastqc.html, fastqc/SRR5831026_2_fastqc.html
        jobid: 0
        reason: Input files updated by another job: fastqc/SRR5831026_1_fastqc.html,
fastqc/SRR5831026_2_fastqc.html
        resources: tmpdir=/tmp

Job stats:
job       count   min threads     max threads
------  -------  -------------  -------------
all       1                 1                 1
fastqc            2                 1                 1
total             3                 1                 1


This was a dry-run (flag -n). The order of jobs does not reflect the order of execution.
```

Full rerun of the whole process:

```
(snakemake) ana@HojansPC:~/nsa_project$ snakemake --cores 20 -F
```

Tags:

- --cores number of cores assigned for a job
- -F force run.

```
Building DAG of jobs...
Using shell: /usr/bin/bash
Provided cores: 20
Rules claiming more threads will be scaled down.
Conda environments: ignored
Job stats:
```

```
job                count   min threads     max threads
-----------  -------  -------------  -------------
all                1               1               1
fastqc             2               1               1
multiqc            1               1               1
star               1               1               1
star_index         1              20              20
trimmomatic        1               1               1
total              7               1              20

Select jobs to execute...

[Wed Jun  8 14:23:32 2022]
Job 5: Testing STAR index
Reason: Forced execution

[Wed Jun  8 14:26:21 2022]
Finished job 5.
1 of 7 steps (14%) done
Select jobs to execute...

[Wed Jun  8 14:26:21 2022]
rule trimmomatic:
        input: SRR5831025.fastq, SRR5831025.fastq
        output: trimmed/SRR5831025_1.fastq, trimmed/SRR5831025_1_unpaired.fastq,
trimmed/SRR5831025_2.fastq, trimmed/SRR5831025_2_unpaired.fastq
        jobid: 4
        reason: Forced execution
        wildcards: sample=SRR5831025
        resources: tmpdir=/tmp


[Wed Jun  8 14:26:21 2022]
rule fastqc:
        input: SRR5831025_2.fastq
        output: fastqc/SRR5831025_2_fastqc.html, fastqc/SRR5831025_2_fastqc.zip
        jobid: 2
        reason: Forced execution
        wildcards: sample=SRR5831025, strand=2
        resources: tmpdir=/tmp


[Wed Jun  8 14:26:21 2022]
rule fastqc:
        input: SRR5831025_1.fastq
        output: fastqc/SRR5831025_1_fastqc.html, fastqc/SRR5831025_1_fastqc.zip
        jobid: 1
        reason: Forced execution
        wildcards: sample=SRR5831025, strand=1
        resources: tmpdir=/tmp

TrimmomaticPE: Started with arguments:
 SRR5831025.fastq SRR5831025.fastq trimmed/SRR5831025_1.fastq
trimmed/SRR5831025_1_unpaired.fastq trimmed/SRR5831025_2.fastq
trimmed/SRR5831025_2_unpaired.fastq
```

```
ILLUMINACLIP:adapters/TruSeq2-PE.fa:2:30:10:2:True LEADING:3 TRAILING:3
SLIDINGWINDOW:4:15 MINLEN:36
Multiple cores found: Using 4 threads
java.io.FileNotFoundException: /home/ana/nsa_project/adapters/TruSeq2-PE.fa (No such
file or directory)
    at java.io.FileInputStream.open0(Native Method)
    at java.io.FileInputStream.open(FileInputStream.java:195)
    at java.io.FileInputStream.<init>(FileInputStream.java:138)
    at org.usadellab.trimmomatic.fasta.FastaParser.parse(FastaParser.java:54)
    at
org.usadellab.trimmomatic.trim.IlluminaClippingTrimmer.loadSequences(IlluminaClippingTri
mmer.java:110)
    at
org.usadellab.trimmomatic.trim.IlluminaClippingTrimmer.makeIlluminaClippingTrimmer(Illumi
naClippingTrimmer.java:71)
    at org.usadellab.trimmomatic.trim.TrimmerFactory.makeTrimmer(TrimmerFactory.java:32)
    at org.usadellab.trimmomatic.Trimmomatic.createTrimmers(Trimmomatic.java:59)
    at org.usadellab.trimmomatic.TrimmomaticPE.run(TrimmomaticPE.java:552)
    at org.usadellab.trimmomatic.Trimmomatic.main(Trimmomatic.java:80)
Quality encoding detected as phred33
Started analysis of SRR5831025_1.fastq
Started analysis of SRR5831025_2.fastq
Approx 5% complete for SRR5831025_1.fastq
Approx 5% complete for SRR5831025_2.fastq
Approx 10% complete for SRR5831025_1.fastq
Approx 10% complete for SRR5831025_2.fastq
Approx 15% complete for SRR5831025_1.fastq
Approx 15% complete for SRR5831025_2.fastq
Approx 20% complete for SRR5831025_1.fastq
Approx 20% complete for SRR5831025_2.fastq
Approx 25% complete for SRR5831025_1.fastq
Approx 25% complete for SRR5831025_2.fastq
Approx 30% complete for SRR5831025_1.fastq
Approx 30% complete for SRR5831025_2.fastq
Approx 35% complete for SRR5831025_1.fastq
Input Read Pairs: 500000 Both Surviving: 495675 (99.14%) Forward Only Surviving: 0
(0.00%) Reverse Only Surviving: 0 (0.00%) Dropped: 4325 (0.86%)
TrimmomaticPE: Completed successfully
[Wed Jun  8 14:26:22 2022]
Finished job 4.
2 of 7 steps (29%) done
Select jobs to execute...

[Wed Jun  8 14:26:22 2022]
rule star:
        input: Arabidopsis_thaliana, trimmed/SRR5831025_1.fastq,
trimmed/SRR5831025_2.fastq
        output: star/SRR5831025
        jobid: 6
        reason: Input files updated by another job: trimmed/SRR5831025_1.fastq,
trimmed/SRR5831025_2.fastq, Arabidopsis_thaliana
        wildcards: sample=SRR5831025
        resources: tmpdir=/tmp
```

```
    STAR --runThreadN 20 --genomeDir Arabidopsis_thaliana --readFilesIn
trimmed/SRR5831025_1.fastq trimmed/SRR5831025_2.fastq --outFileNamePrefix
star/SRR5831025/STAR_SRR5831025_
    STAR version: 2.7.10a   compiled: 2022-01-14T18:50:00-05:00
:/home/dobin/data/STAR/STARcode/STAR.master/source
Jun 08 14:26:22 ..... started STAR run
Jun 08 14:26:22 ..... loading genome
Approx 35% complete for SRR5831025_2.fastq
Approx 40% complete for SRR5831025_1.fastq
Approx 40% complete for SRR5831025_2.fastq
Approx 45% complete for SRR5831025_1.fastq
Approx 50% complete for SRR5831025_1.fastq
Approx 45% complete for SRR5831025_2.fastq
Approx 55% complete for SRR5831025_1.fastq
Approx 50% complete for SRR5831025_2.fastq
Approx 60% complete for SRR5831025_1.fastq
Approx 55% complete for SRR5831025_2.fastq
Approx 65% complete for SRR5831025_1.fastq
Approx 60% complete for SRR5831025_2.fastq
Approx 70% complete for SRR5831025_1.fastq
Approx 65% complete for SRR5831025_2.fastq
Approx 75% complete for SRR5831025_1.fastq
Jun 08 14:26:24 ..... started mapping
Approx 70% complete for SRR5831025_2.fastq
Approx 80% complete for SRR5831025_1.fastq
Approx 75% complete for SRR5831025_2.fastq
Approx 85% complete for SRR5831025_1.fastq
Approx 80% complete for SRR5831025_2.fastq
Approx 90% complete for SRR5831025_1.fastq
Approx 85% complete for SRR5831025_2.fastq
Approx 95% complete for SRR5831025_1.fastq
Approx 90% complete for SRR5831025_2.fastq
Approx 100% complete for SRR5831025_1.fastq
Analysis complete for SRR5831025_1.fastq
Approx 95% complete for SRR5831025_2.fastq
Approx 100% complete for SRR5831025_2.fastq
Analysis complete for SRR5831025_2.fastq
[Wed Jun  8 14:26:29 2022]
Finished job 2.
3 of 7 steps (43%) done
[Wed Jun  8 14:26:29 2022]
Finished job 1.
4 of 7 steps (57%) done
Select jobs to execute...

[Wed Jun  8 14:26:29 2022]
rule multiqc:
        input: fastqc/SRR5831025_1_fastqc.zip, fastqc/SRR5831025_2_fastqc.zip
        output: multiqc
        jobid: 3
        reason: Input files updated by another job: fastqc/SRR5831025_2_fastqc.zip,
fastqc/SRR5831025_1_fastqc.zip
        resources: tmpdir=/tmp
```

```
 /// MultiQC 🔍 | v1.12

|           multiqc | Search path : /home/ana/nsa_project/fastqc
|           searching |
————————————————————————————————————————————————————————————————————————————————
100% 6/6
|           fastqc | Found 2 reports
|           multiqc | Compressing plot data
|           multiqc | Report      : multiqc/multiqc_report.html
|           multiqc | Data        : multiqc/multiqc_data
|           multiqc | MultiQC complete
[Wed Jun  8 14:26:35 2022]
Finished job 3.
5 of 7 steps (71%) done
Jun 08 14:36:32 ..... finished mapping
Jun 08 14:36:32 ..... finished successfully
[Wed Jun  8 14:36:32 2022]
Finished job 6.
6 of 7 steps (86%) done
Select jobs to execute...

[Wed Jun  8 14:36:32 2022]
localrule all:
        input: fastqc/SRR5831025_1_fastqc.html, fastqc/SRR5831025_1_fastqc.zip,
fastqc/SRR5831025_2_fastqc.html, fastqc/SRR5831025_2_fastqc.zip, multiqc,
trimmed/SRR5831025_1.fastq, trimmed/SRR5831025_1_unpaired.fastq,
trimmed/SRR5831025_2.fastq, trimmed/SRR5831025_2_unpaired.fastq,
Arabidopsis_thaliana, star/SRR5831025
        jobid: 0
        reason: Input files updated by another job: fastqc/SRR5831025_2_fastqc.zip,
trimmed/SRR5831025_1_unpaired.fastq, trimmed/SRR5831025_2.fastq,
fastqc/SRR5831025_2_fastqc.html, Arabidopsis_thaliana,
fastqc/SRR5831025_1_fastqc.html, multiqc, trimmed/SRR5831025_2_unpaired.fastq,
trimmed/SRR5831025_1.fastq, star/SRR5831025, fastqc/SRR5831025_1_fastqc.zip
        resources: tmpdir=/tmp

[Wed Jun  8 14:36:32 2022]
Finished job 0.
7 of 7 steps (100%) done
Complete log: .snakemake/log/2022-06-08T142332.673597.snakemake.log
```

Final snakemake file:

```
SAMPLES = ["SRR5831025"]
strand = [1, 2]

rule all:
      input:
      expand("fastqc/{sample}_{strand}_fastqc.{arg}", sample=SAMPLES,
strand=strand, arg=["html","zip"]),
      "multiqc",
      expand("trimmed/{sample}_{strand}{ispair}.fastq", sample=SAMPLES,
```

```
strand=strand, ispair=["","_unpaired"]),
      "Arabidopsis_thaliana",
      expand("star/{sample}", sample=SAMPLES)
rule fastqc:
      input:
      "{sample}_{strand}.fastq"
      output:
      html = "fastqc/{sample}_{strand}_fastqc.html",
      zip = "fastqc/{sample}_{strand}_fastqc.zip"
      conda:
      "quality_check.yaml"
      shell:
      '''
      fastqc {input} &&
      mv {wildcards.sample}_{wildcards.strand}_fastqc.zip {output.zip}
&&
      mv {wildcards.sample}_{wildcards.strand}_fastqc.html {output.html}
      '''
rule multiqc:
      input:
      expand("fastqc/{sample}_{strand}_fastqc.zip", sample=SAMPLES,
strand=strand)
      output:
      directory("multiqc")
      conda:
      "quality_check.yaml"
      shell:
      "multiqc fastqc --outdir multiqc"
rule trimmomatic:
      input:
      F = "{sample}.fastq",
      R = "{sample}.fastq"
      output:
      Fp = "trimmed/{sample}_1.fastq",
      Fu = "trimmed/{sample}_1_unpaired.fastq",
      Rp = "trimmed/{sample}_2.fastq",
      Ru = "trimmed/{sample}_2_unpaired.fastq"
      conda:
      "quality_check.yaml"
      shell:
      '''
      trimmomatic PE {input.F} {input.R} {output.Fp} {output.Fu}
{output.Rp} {output.Ru} \
      ILLUMINACLIP:adapters/TruSeq2-PE.fa:2:30:10:2:True \
      LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36
      '''
```

```
rule star_index:
      input:
      fasta="genome/Arabidopsis_thaliana.TAIR10.dna.toplevel.fa",
      gtf = "genome/Arabidopsis_thaliana.TAIR10.53.gtf"
      output:
      directory("Arabidopsis_thaliana")
      message:
      "Testing STAR index"
      threads: 20
      params:
      extra="",
      log:
      "logs/star_index_Arabidopsis_thaliana.log",
      wrapper:
      "v1.5.0/bio/star/index"
rule star:
      input:
      #index = "{genome}",
      index = "Arabidopsis_thaliana",
      F = "trimmed/{sample}_1.fastq",
      R = "trimmed/{sample}_2.fastq"
      output:
      directory("star/{sample}"),
#     log:
#     "star/{sample}/Log.out"
      conda:
      "quality_check.yaml"
      shell:
      '''
      STAR --runThreadN 20 --genomeDir {input.index} \
      --readFilesIn {input.F} {input.R} \
      --outFileNamePrefix {output}/STAR_{wildcards.sample}_
      '''
```