Bioinformatics and Systems Biology: Expression, Regulation and Networks Exam project

Emiel Vanhulle r0308983, Jasper Vanhecke r0309321, Lena Smets r0387223

2e master Biomedical Sciences 2016-2017

All the commands performed, and the obtained files were stored under the account of Emiel Vanhulle. The exact location of the original ipynb file is /mnt/storage/r0308983/jupyternotebooks/Examenfolder

ChIP-Seq and RNA-Seq Analyses Identify Components of the Wnt and Fgf Signaling Pathways as Prep1 Target Genes in Mouse Embryonic Stem Cells

Source : https://www.ncbi.nlm.nih.gov/pubmed/25875616 (https://www.ncbi.nlm.nih.gov/pubmed/25875616)

At multiple stages of the mouse development, the Prep1- or Pknox1 homeodomain transcription factor is crucial. It belongs to the TALE (Three Amino acids Loop Extension) family of proteins. Prep1 and Pbx proteins dimerize through the amino-terminal domain of Pbx followed by DNA-binding activity to both proteins. Through the Pbx homeodomain, the dimers can also bind the anterior Hox proteins leading to functional trimers characterised with high DNA selectivity and activity. It is observed that Prep1 -/- cannot even reach the gastrulation stage because the epiblast dies through p53-dependent apoptosis around E6.25. On the other hand, it is observed that hypomorphic Prep1 i/i embryos, which only express 2% of the control Prep1 mRNA levels, die mostly at E17.5. On top, double prep1 -/i heterozygotes show a phenotype that is intermediate and the mice die at stage E12.5 showing major cranial and brain defects. We can conclude that Prep1 is an essential protein in at least three different stages of the development of mice: pre-gastrulation, brain-cranium formation and haematopoiesis. The embryonic phenotype can differ and be highly affected according to the different expression levels (even low) of Prep1.

In adult mice, Prep1 has a tumor suppression function. This was discovered after observing a variety of tumours occurring in the few surviving hypomorphic Prep1 i/I and the apparently normal Prep1+/i heterozygous mice. Likewise, Prep1 haploinsufficiency accelerates lymphogenesis that is Myc-driven. Double-stranded DNA breaks and chromosomal aberrations accumulate because of Prep1 downregulation, so the underlying mechanism seems to be DNA damage. We can conclude that Prep1 has multiple functions in different cells during the embryogenesis of the mouse and in its adult life. Prep1 has at least an essential role in some stem cells since the function and the self-renewal of hematopoietic stem cells is deficient in embryos having Prep1 i/i. On top, the Prep1 -/- blastocysts produce fewer and deficient ES cells. With previous ChiP-seq analysis the authors of the article identified the Prep1 target genes in the whole E11.5 embryo trunk (mixed population of embryonic progenitors and differentiated cells). Prep1 is observed to bind to a specific decameric DNA sequence at over 3000 genes.

We will first use the RNA sequencing to see which genes are down - and upregulated in Prep1 -/- ES cells. Afterwards, ChiP sequencing will be used to determine binding targets for Prep1.

A general overview of the Pknox1 homeodomain can be found on the next link:
http://www.humanmine.org/humanmine/report.do?id=1142390
(http://www.humanmine.org/humanmine/report.do?id=1142390)

After the commands were run and some gene ontology methods didn't give any results, we further thought about our dataset. We looked at the impact factor of the journal PLoS ONE, which is 3.54. We compared our article to other articles and saw that ours only had one datasat available for the knock-out condition (n=1). It would be much better and more statistically relevant if more datasets for the knock-out condition would have been available so that more data could have been compared.

Downloading data and convert SRA files

We downloaded the data from the site and made a new folder to store the data in. The commands were not performed in jupyter but in the terminal.

mkdir ~/data/ncbi vdb-config -s /repository/user/main/public/root=~/data/ncbi

mkdir ~/data/examwork cd ~/data/examwork

fastq-dump --split-files SRR1652476 fastq-dump --split-files SRR1652477 fastq-dump --split-files SRR1654369 fastq-dump --split-files SRR1654370

This converted the sra files to fastq files. This generated two fastq files per SRA files for the RNA sequence files since these were paired-end reads. The commands are shown above.

In [155]:

```
cd /mnt/storage/r0308983/jupyternotebooks/examwork
pwd
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork
```

The SRR1652477 failed, it was paired while normally it should be single layout. We redid the fastq-dump command for this file.

In [156]:

```
fastq-dump --split-files SRR1652477
```

```
Read 30912550 spots for SRR1652477
Written 30912550 spots for SRR1652477
```

The folder 'examwork' was moved into jupyternotebooks.

RNA-Sequencing

We create a new folder to store all the generated files and data in.

In [4]:

```
mkdir -p /mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq
cd /mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq
```

In [5]:

```
pwd
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq
```

We create a symbolic link to all the raw fastq files. (a symbolic link instead of a full copy)

In [6]:

```
ln -sf /mnt/storage/r0308983/jupyternotebooks/examwork/SRR1654370_1.fastq /mnt/storage/r030
ln -sf /mnt/storage/r0308983/jupyternotebooks/examwork/SRR1654370_2.fastq /mnt/storage/r030
ln -sf /mnt/storage/r0308983/jupyternotebooks/examwork/SRR1654369_1.fastq /mnt/storage/r030
ln -sf /mnt/storage/r0308983/jupyternotebooks/examwork/SRR1654369_2.fastq /mnt/storage/r030
```

Quality control using FASTQC

After obtaining the sequencing data, the first step is to check our data quality. This step helps to identify potential technical biases that might affect our analysis. In quality control we have seen that three most important parameters are: per base sequence quality, sequence duplication levels and overrepresented sequences. The code used was: fastqc fastqfile -o . outputdir

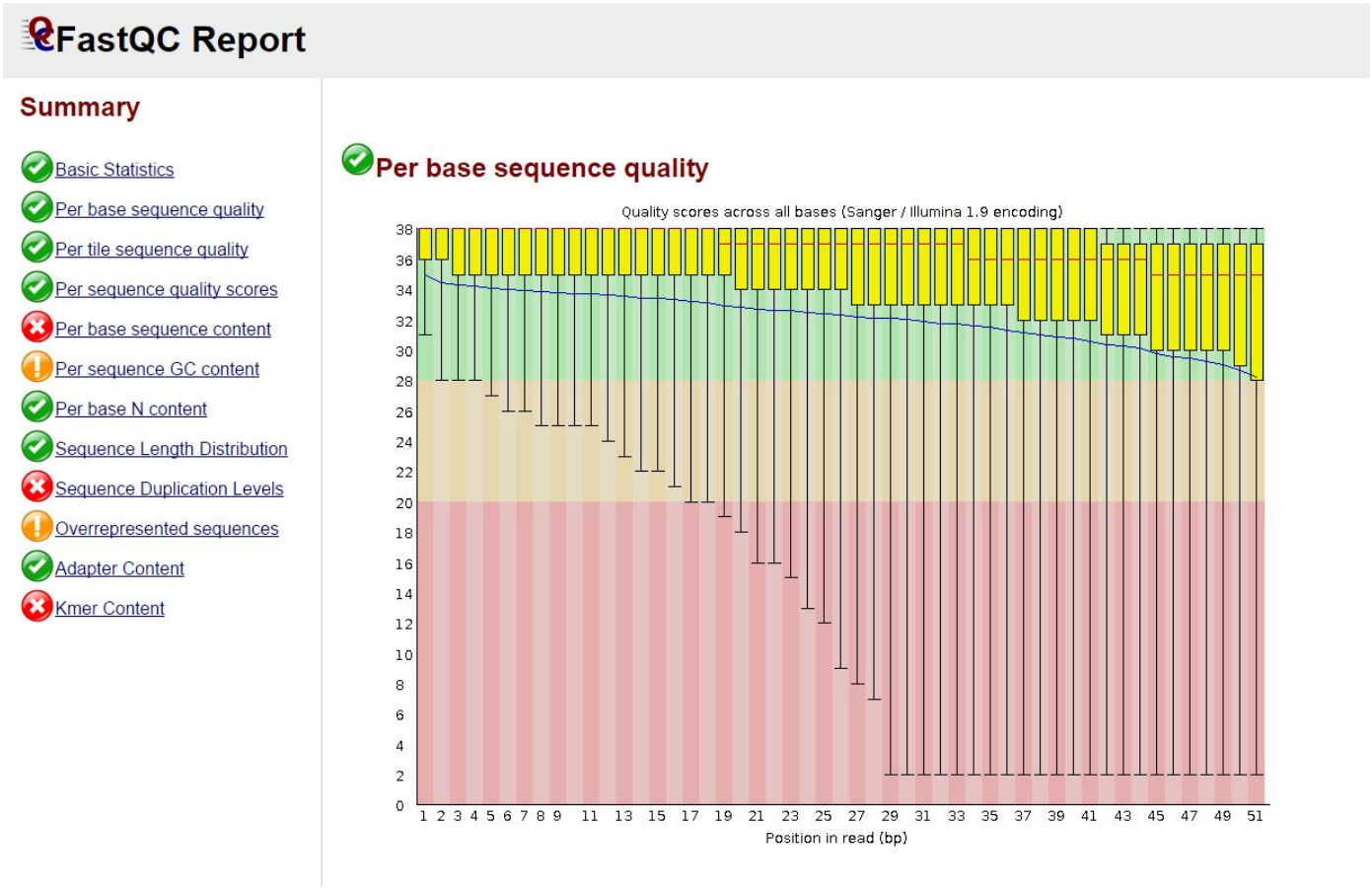
In [7]:

```
/usr/local/bin/FastQC/fastqc -o . SRR1654370_1.fastq
/usr/local/bin/FastQC/fastqc -o . SRR1654370_2.fastq
/usr/local/bin/FastQC/fastqc -o . SRR1654369_1.fastq
/usr/local/bin/FastQC/fastqc -o . SRR1654369_2.fastq
```

```
Started analysis of SRR1654370_1.fastq
Approx 5% complete for SRR1654370_1.fastq
Approx 10% complete for SRR1654370_1.fastq
Approx 15% complete for SRR1654370_1.fastq
Approx 20% complete for SRR1654370_1.fastq
Approx 25% complete for SRR1654370_1.fastq
Approx 30% complete for SRR1654370_1.fastq
Approx 35% complete for SRR1654370_1.fastq
Approx 40% complete for SRR1654370_1.fastq
Approx 45% complete for SRR1654370_1.fastq
Approx 50% complete for SRR1654370_1.fastq
Approx 55% complete for SRR1654370_1.fastq
Approx 60% complete for SRR1654370_1.fastq
Approx 65% complete for SRR1654370_1.fastq
Approx 70% complete for SRR1654370_1.fastq
Approx 75% complete for SRR1654370_1.fastq
Approx 80% complete for SRR1654370_1.fastq
Approx 85% complete for SRR1654370_1.fastq
Approx 90% complete for SRR1654370_1.fastq
Approx 95% complete for SRR1654370_1.fastq
Analysis complete for SRR1654370_1.fastq
Started analysis of SRR1654370_2.fastq
Approx 5% complete for SRR1654370_2.fastq
Approx 10% complete for SRR1654370_2.fastq
Approx 15% complete for SRR1654370_2.fastq
Approx 20% complete for SRR1654370_2.fastq
Approx 25% complete for SRR1654370_2.fastq
Approx 30% complete for SRR1654370_2.fastq
Approx 35% complete for SRR1654370_2.fastq
Approx 40% complete for SRR1654370_2.fastq
Approx 45% complete for SRR1654370_2.fastq
Approx 50% complete for SRR1654370_2.fastq
Approx 55% complete for SRR1654370_2.fastq
Approx 60% complete for SRR1654370_2.fastq
Approx 65% complete for SRR1654370_2.fastq
Approx 70% complete for SRR1654370_2.fastq
Approx 75% complete for SRR1654370_2.fastq
Approx 80% complete for SRR1654370_2.fastq
Approx 85% complete for SRR1654370_2.fastq
Approx 90% complete for SRR1654370_2.fastq
Approx 95% complete for SRR1654370_2.fastq
Analysis complete for SRR1654370_2.fastq
Started analysis of SRR1654369_1.fastq
Approx 5% complete for SRR1654369_1.fastq
Approx 10% complete for SRR1654369_1.fastq
Approx 15% complete for SRR1654369_1.fastq
Approx 20% complete for SRR1654369_1.fastq
Approx 25% complete for SRR1654369_1.fastq
Approx 30% complete for SRR1654369_1.fastq
Approx 35% complete for SRR1654369_1.fastq
Approx 40% complete for SRR1654369_1.fastq
Approx 45% complete for SRR1654369_1.fastq
Approx 50% complete for SRR1654369_1.fastq
Approx 55% complete for SRR1654369_1.fastq
Approx 60% complete for SRR1654369_1.fastq
Approx 65% complete for SRR1654369_1.fastq
Approx 70% complete for SRR1654369_1.fastq
Approx 75% complete for SRR1654369_1.fastq
Approx 80% complete for SRR1654369_1.fastq
Approx 85% complete for SRR1654369_1.fastq
```

```
Approx 90% complete for SRR1654369_1.fastq
Approx 95% complete for SRR1654369_1.fastq
Analysis complete for SRR1654369_1.fastq
Started analysis of SRR1654369_2.fastq
Approx 5% complete for SRR1654369_2.fastq
Approx 10% complete for SRR1654369_2.fastq
Approx 15% complete for SRR1654369_2.fastq
Approx 20% complete for SRR1654369_2.fastq
Approx 25% complete for SRR1654369_2.fastq
Approx 30% complete for SRR1654369_2.fastq
Approx 35% complete for SRR1654369_2.fastq
Approx 40% complete for SRR1654369_2.fastq
Approx 45% complete for SRR1654369_2.fastq
Approx 50% complete for SRR1654369_2.fastq
Approx 55% complete for SRR1654369_2.fastq
Approx 60% complete for SRR1654369_2.fastq
Approx 65% complete for SRR1654369_2.fastq
Approx 70% complete for SRR1654369_2.fastq
Approx 75% complete for SRR1654369_2.fastq
Approx 80% complete for SRR1654369_2.fastq
Approx 85% complete for SRR1654369_2.fastq
Approx 90% complete for SRR1654369_2.fastq
Approx 95% complete for SRR1654369_2.fastq
Analysis complete for SRR1654369_2.fastq
```

Quality control SRR1654369_1 (Mouse.ES.wild_type)



In general, the quality of the reads are good. The per base sequence quality is decent, but there is an error for the per base sequence content and an error for the sequence duplication levels. This warning is generated when these non-unique sequences takes up >20% of the total number of sequences. After

removing the duplicates, 49,29% of sequences remain. A warning is also given for the overrepresented sequences, this warning is given when a sequence represents more than 0.1% of the total amount of sequences. When blasting these overrepresented sequences, the outcome seems to point towards a contamination of other organisms (e.g orangutan, macaque,etc..).

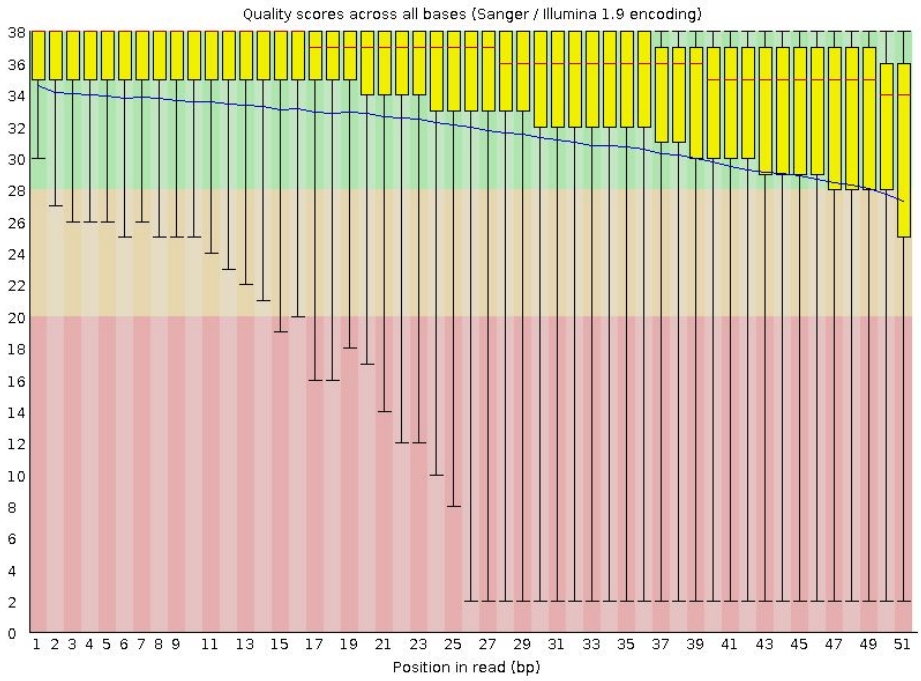
Quality control SRR1654369_2 (Mouse.ES.wild_type)

FastQC Report

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- 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
- Kmer Content

Per base sequence quality



In general the per base sequence is adequate. The FastQC report gives a warning (as above) concerning the duplication levels and the overrepresented sequences. After deduplication, 51,76% of the sequences remain. The sequence duplications can be the result of oversequencing of certain reads. For example, transcripts that were very low expressed. The overrepresented sequences can be the result of contamination or by having an important biological function. When blasting the overrepresented sequences, we also found some contamination in these samples, originating from other organisms, for example a rat.

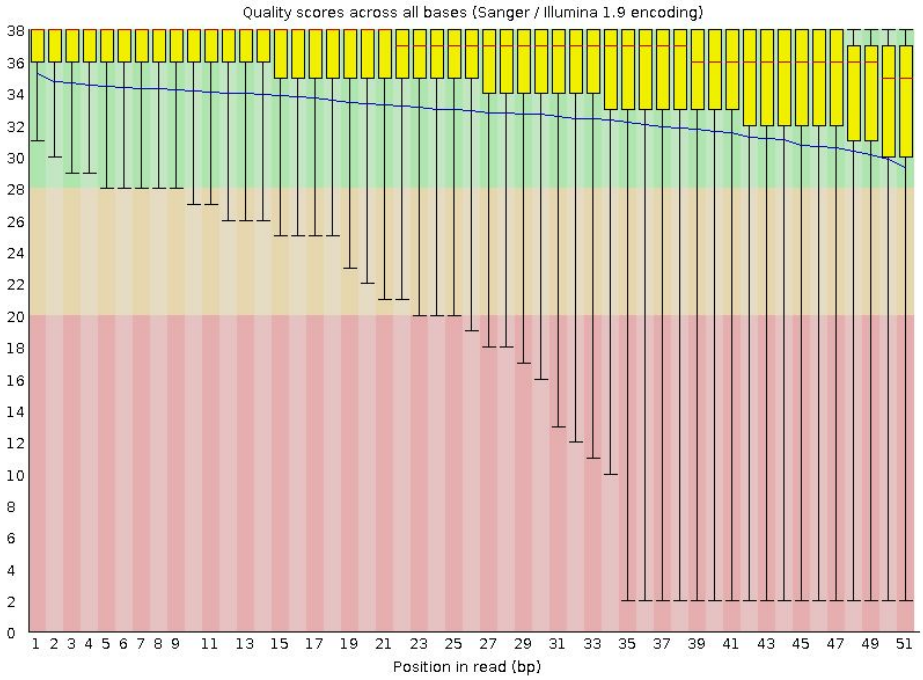
Quality control SRR1654370_1 (Mouse.ES.Prep1_KO)

FastQC Report

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- 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
- Kmer Content

Per base sequence quality



The per base sequence quality is good, but there seems to be an error concerning the duplication levels and a warning concerning the overrepresented sequences. In this case the percentage of sequences that remaine after removing duplicates is 49,54%. The sequence duplication levels are most likely the result of oversequencing to be able to detect lowly expressed transcripts. Several overrepresented sequences were detected. After blasting those sequences, it seems like there was contamination by Primates (more secific: macaques or orangutans), as seen in the previous QC quality report.

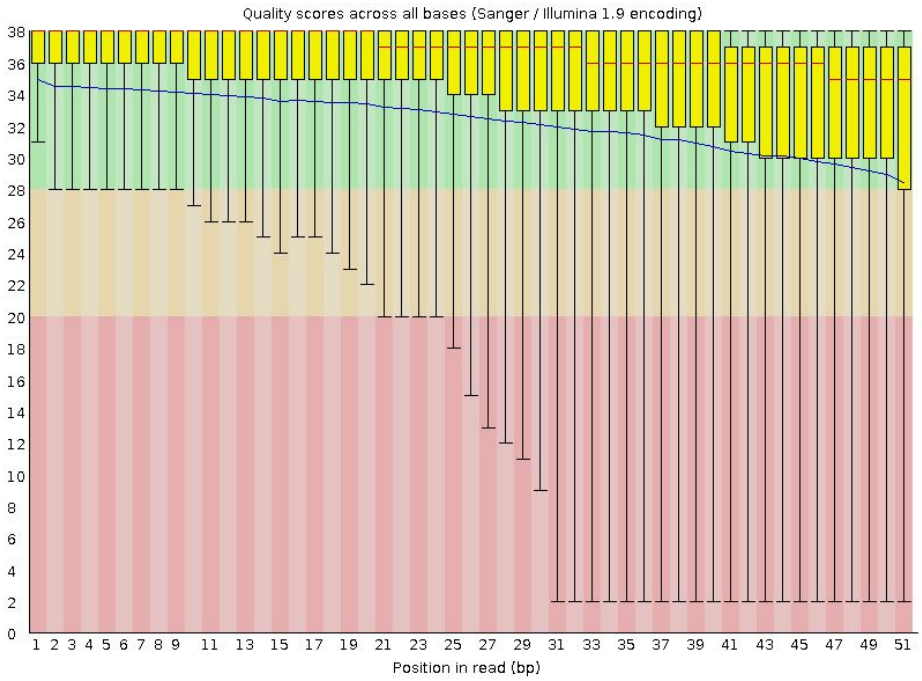
Quality control SRR1654370_2 (Mouse.ES.Prep1_KO)

FastQC Report

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- 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
- Kmer Content

Per base sequence quality



In the QC-quality report for this knock out, the base sequence quality seems to be good. The report gives a warning for the duplicated sequences. In this case the percentage of sequences that remain after removing duplicates is 52.68%. There is again warning about overrepresented sequences. In this case, an overrepresented sequence with a hit against a rat was found.

In [8]:

```
ls -l
```

```
total 3344
lrwxrwxrwx 1 r0308983 domain users     66 jan  3 15:06 SRR1654369_1.fastq ->
 /mnt/storage/r0308983/jupyternotebooks/examwork/SRR1654369_1.fastq
-rw-r--r-- 1 r0308983 domain users 338251 jan  3 15:24 SRR1654369_1_fastqc.h
tml
-rw-r--r-- 1 r0308983 domain users 464261 jan  3 15:24 SRR1654369_1_fastqc.z
ip
lrwxrwxrwx 1 r0308983 domain users     66 jan  3 15:06 SRR1654369_2.fastq ->
 /mnt/storage/r0308983/jupyternotebooks/examwork/SRR1654369_2.fastq
-rw-r--r-- 1 r0308983 domain users 374573 jan  3 15:27 SRR1654369_2_fastqc.h
tml
-rw-r--r-- 1 r0308983 domain users 517098 jan  3 15:27 SRR1654369_2_fastqc.z
ip
lrwxrwxrwx 1 r0308983 domain users     66 jan  3 15:06 SRR1654370_1.fastq ->
 /mnt/storage/r0308983/jupyternotebooks/examwork/SRR1654370_1.fastq
-rw-r--r-- 1 r0308983 domain users 347236 jan  3 15:20 SRR1654370_1_fastqc.h
tml
-rw-r--r-- 1 r0308983 domain users 479150 jan  3 15:20 SRR1654370_1_fastqc.z
ip
lrwxrwxrwx 1 r0308983 domain users     66 jan  3 15:06 SRR1654370_2.fastq ->
 /mnt/storage/r0308983/jupyternotebooks/examwork/SRR1654370_2.fastq
-rw-r--r-- 1 r0308983 domain users 367028 jan  3 15:22 SRR1654370_2_fastqc.h
tml
-rw-r--r-- 1 r0308983 domain users 505929 jan  3 15:22 SRR1654370_2_fastqc.z
ip
```

Mapping to the genome

We will use STAR (Spliced Transcripts Alignment to a Reference) to map to the genome. STAR delivers the best results according to several papers. We also used STAR in class, but there we already had a human reference genome to work with. Here, we are working with mus musculus samples, so we need to generate our own reference genome. First: make a database with the mouse genome and gene annotation -> make a reference genoom to which we will map our reads. We got the necessary files from https://www.ncbi.nlm.nih.gov/genome/?term=mus%20musculus%209 (https://www.ncbi.nlm.nih.gov/genome/?term=mus%20musculus%209)

Get latest STAR source from releases

In [9]:

```
wget https://github.com/alexdobin/STAR/archive/2.5.2b.tar.gz
tar -xzf 2.5.2b.tar.gz
cd STAR-2.5.2b
```

```
--2017-01-03 16:27:10--  https://github.com/alexdobin/STAR/archive/2.5.2b.ta
r.gz (https://github.com/alexdobin/STAR/archive/2.5.2b.tar.gz)
Resolving github.com (github.com)... 192.30.253.112, 192.30.253.113
Connecting to github.com (github.com)|192.30.253.112|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/alexdobin/STAR/tar.gz/2.5.2b (https://
codeload.github.com/alexdobin/STAR/tar.gz/2.5.2b) [following]
--2017-01-03 16:27:10--  https://codeload.github.com/alexdobin/STAR/tar.gz/
2.5.2b (https://codeload.github.com/alexdobin/STAR/tar.gz/2.5.2b)
Resolving codeload.github.com (codeload.github.com)... 192.30.253.121, 192.3
0.253.120
Connecting to codeload.github.com (codeload.github.com)|192.30.253.121|:44
3... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-gzip]
Saving to: ‘2.5.2b.tar.gz’

    [             <=>                     ] 9.735.014   3,11MB/s   in 3,0s


2017-01-03 16:27:14 (3,11 MB/s) - ‘2.5.2b.tar.gz’ saved [9735014]
```

We then unpacked the fasta reference genome file and the gene annotation file.

In [18]:

```
pwd
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq
```

unzip or unpack the reference genome (in fasta format)

In [22]:

```
gunzip GCF_000001635.25_GRCm38.p5_genomic.fna.gz
```

unzip or unpack the annotation file (gff format)

In [16]:

```
gunzip GCF_000001635.25_GRCm38.p5_genomic.gff.gz
```

convert gff file to gtf file to work with in generating genome indices:

In [23]:

```
gffread GCF_000001635.25_GRCm38.p5_genomic.gff -T -o GCF_000001635.25_GRCm38.p5_genomic.gtf
```

```
The program 'gffread' is currently not installed. To run 'gffread' please as
k your administrator to install the package 'cufflinks'
```

Generate genome indexes

Basic STAR workflow consists of two steps:

1) Generating genome indexes files.

In this step, the reference genome sequences (FASTA files) and annotations (GTF file), are supplied, from which STAR generates genome indexes that are utilized in the second mapping step.

2) Mapping reads to the genome.

In this step, genome files generated in the first step, as well as the RNAseq reads (sequences, FASTA / FASTQ files) are supplied by us. STAR maps the reads to the genome and gives us several output files, such as alignments (SAM/BAM), mapping summary statistics, splice junctions, unmapped reads, signal (wiggle) tracks etc.

We used the mus musculus genome and corresponding annotation file to generate a reference genome.

This is the command we used to generate the genome indexes (we ran this in an external terminal):

runThreadN is the number of processors that will be used.

genomeDir is the directory in which all the output will be stored.

genomeFastaFiles shows the path to the mus musculus genome.

sjdbGTFfile shows the path to the mus musculus annotation file.

sjdgOverhang tells STAR how many bases to concatenate from donor and acceptor sides of the junctions. We used the standard value of 100.

```
STAR --runThreadN 4 --runMode genomeGenerate --genomeDir
/mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq --genomeFastaFiles
/mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq/GCF_000001635.25_GRCm38.p5_genomic.
fna --sjdbGTFfile
/mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq/GCF_000001635.25_GRCm38.p5_genomic.
gff --sjdbOverhang 100
```

Mapping reads to the genome

The different fasta files containing the reads can be mapped to the reference genome wchich was generated in the previous step. For each samples two fastq files were mapped at the same time since these files are paired-end reads.

In [1]:

```
pwd
```

```
/mnt/storage/r0308983/jupyternotebooks/Examenfolder
```

In [11]:

```
cd /mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq
```

In [13]:

```
pwd
ls
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq
2.5.2b.tar.gz                           sjdbList.fromGTF.out.tab
chrLength.txt                           sjdbList.out.tab
chrNameLength.txt                       SRR1654369_1.fastq
chrName.txt                             SRR1654369_1_fastqc.html
chrStart.txt                            SRR1654369_1_fastqc.zip
exonGeTrInfo.tab                        SRR1654369_2.fastq
exonInfo.tab                            SRR1654369_2_fastqc.html
GCF_000001635.25_GRCm38.p5_genomic.fna  SRR1654369_2_fastqc.zip
GCF_000001635.25_GRCm38.p5_genomic.gff  SRR1654370_1.fastq
geneInfo.tab                            SRR1654370_1_fastqc.html
Genome                                  SRR1654370_1_fastqc.zip
genomeParameters.txt                    SRR1654370_2.fastq
Log.out                                 SRR1654370_2_fastqc.html
SA                                      SRR1654370_2_fastqc.zip
SAindex                                 STAR-2.5.2b
sjdbInfo.txt                            transcriptInfo.tab
```

genomeDir is the path to the mus musculus reference genome generated earlier.

runThreadN is the number of processors used

readFilesIn is the path to the fastq files that we will map.

outFileNamePrefix is the name that will be given to the output file.

In [15]:

```
STAR --genomeDir /mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq \
     --runThreadN 4 \
     --readFilesIn SRR1654369_1.fastq SRR1654369_2.fastq \
     --outFileNamePrefix SRR1654369.
```

```
Jan 03 21:36:31 ..... started STAR run
Jan 03 21:36:31 ..... loading genome
Jan 03 21:44:18 ..... started mapping
Jan 03 21:56:10 ..... finished successfully
```

The output file is now in `SRR1654369.Aligned.out.sam`. Apart from that we have a number of additional log files generated.

In [16]:

```
STAR --genomeDir /mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq \
     --runThreadN 4 \
     --readFilesIn SRR1654370_1.fastq SRR1654370_2.fastq \
     --outFileNamePrefix SRR1654370.
```

```
Jan 03 22:00:14 ..... started STAR run
Jan 03 22:00:14 ..... loading genome
Jan 03 22:05:34 ..... started mapping
Jan 03 22:14:44 ..... finished successfully
```

The output file is now in `SRR1654370.Aligned.out.sam`. Apart from that we have a number of additional log files generated.

In [17]:

```
ls -l SRR1654369.*
```

```
-rw-r--r-- 1 r0308983 domain users 14634176392 jan  3 21:56 SRR1654369.Align
ed.out.sam
-rw-r--r-- 1 r0308983 domain users        1866 jan  3 21:56 SRR1654369.Log.f
inal.out
-rw-r--r-- 1 r0308983 domain users       22637 jan  3 21:56 SRR1654369.Log.o
ut
-rw-r--r-- 1 r0308983 domain users        1544 jan  3 21:56 SRR1654369.Log.p
rogress.out
-rw-r--r-- 1 r0308983 domain users     8248660 jan  3 21:56 SRR1654369.SJ.ou
t.tab
```

In [18]:

```
ls -l SRR1654370.*
```

```
-rw-r--r-- 1 r0308983 domain users 13023377608 jan  3 22:14 SRR1654370.Align
ed.out.sam
-rw-r--r-- 1 r0308983 domain users        1865 jan  3 22:14 SRR1654370.Log.f
inal.out
-rw-r--r-- 1 r0308983 domain users       22637 jan  3 22:14 SRR1654370.Log.o
ut
-rw-r--r-- 1 r0308983 domain users        1190 jan  3 22:14 SRR1654370.Log.p
rogress.out
-rw-r--r-- 1 r0308983 domain users     8045243 jan  3 22:14 SRR1654370.SJ.ou
t.tab
```

SAM file structure

A SAM file is a text file containing the alignment of all reads. A SAM files starts with a header.

In [19]:

```
head -40 SRR1654369.Aligned.out.sam | grep '^@'
```

```
@HD     VN:1.4
@SQ     SN:NC_000067.6  LN:195471971
@SQ     SN:NT_166280.1  LN:169725
@SQ     SN:NT_166281.1  LN:241735
@SQ     SN:NT_166282.1  LN:153618
@SQ     SN:NT_166283.1  LN:39340
@SQ     SN:NT_162750.1  LN:206961
@SQ     SN:NC_000068.7  LN:182113224
@SQ     SN:NC_000069.6  LN:160039680
@SQ     SN:NC_000070.6  LN:156508116
@SQ     SN:NT_166291.1  LN:66673
@SQ     SN:NT_187052.1  LN:14945
@SQ     SN:NT_166434.1  LN:227966
@SQ     SN:NT_187053.1  LN:207968
@SQ     SN:NT_187054.1  LN:191905
@SQ     SN:NT_187055.1  LN:1976
@SQ     SN:NC_000071.6  LN:151834684
@SQ     SN:NT_187056.1  LN:199368
@SQ     SN:NT_187057.1  LN:205776
@SQ     SN:NT_187058.1  LN:184189
@SQ     SN:NT_166438.1  LN:195993
@SQ     SN:NT_187059.1  LN:953012
@SQ     SN:NC_000072.6  LN:149736546
@SQ     SN:NC_000073.6  LN:145441459
@SQ     SN:NT_166307.1  LN:175968
@SQ     SN:NC_000074.6  LN:129401213
@SQ     SN:NC_000075.6  LN:124595110
@SQ     SN:NC_000076.6  LN:130694993
@SQ     SN:NC_000077.6  LN:122082543
@SQ     SN:NC_000078.6  LN:120129022
@SQ     SN:NC_000079.6  LN:120421639
@SQ     SN:NC_000080.6  LN:124902244
@SQ     SN:NC_000081.6  LN:104043685
@SQ     SN:NC_000082.6  LN:98207768
@SQ     SN:NC_000083.6  LN:94987271
@SQ     SN:NC_000084.6  LN:90702639
@SQ     SN:NC_000085.6  LN:61431566
@SQ     SN:NC_000086.7  LN:171031299
@SQ     SN:NT_165789.2  LN:336933
@SQ     SN:NC_000087.7  LN:91744698
```

In [26]:

```
head -70 SRR1654370.Aligned.out.sam | grep '^@'
```

```
@HD	VN:1.4
@SQ	SN:NC_000067.6	LN:195471971
@SQ	SN:NT_166280.1	LN:169725
@SQ	SN:NT_166281.1	LN:241735
@SQ	SN:NT_166282.1	LN:153618
@SQ	SN:NT_166283.1	LN:39340
@SQ	SN:NT_162750.1	LN:206961
@SQ	SN:NC_000068.7	LN:182113224
@SQ	SN:NC_000069.6	LN:160039680
@SQ	SN:NC_000070.6	LN:156508116
@SQ	SN:NT_166291.1	LN:66673
@SQ	SN:NT_187052.1	LN:14945
@SQ	SN:NT_166434.1	LN:227966
@SQ	SN:NT_187053.1	LN:207968
@SQ	SN:NT_187054.1	LN:191905
@SQ	SN:NT_187055.1	LN:1976
@SQ	SN:NC_000071.6	LN:151834684
@SQ	SN:NT_187056.1	LN:199368
@SQ	SN:NT_187057.1	LN:205776
@SQ	SN:NT_187058.1	LN:184189
@SQ	SN:NT_166438.1	LN:195993
@SQ	SN:NT_187059.1	LN:953012
@SQ	SN:NC_000072.6	LN:149736546
@SQ	SN:NC_000073.6	LN:145441459
@SQ	SN:NT_166307.1	LN:175968
@SQ	SN:NC_000074.6	LN:129401213
@SQ	SN:NC_000075.6	LN:124595110
@SQ	SN:NC_000076.6	LN:130694993
@SQ	SN:NC_000077.6	LN:122082543
@SQ	SN:NC_000078.6	LN:120129022
@SQ	SN:NC_000079.6	LN:120421639
@SQ	SN:NC_000080.6	LN:124902244
@SQ	SN:NC_000081.6	LN:104043685
@SQ	SN:NC_000082.6	LN:98207768
@SQ	SN:NC_000083.6	LN:94987271
@SQ	SN:NC_000084.6	LN:90702639
@SQ	SN:NC_000085.6	LN:61431566
@SQ	SN:NC_000086.7	LN:171031299
@SQ	SN:NT_165789.2	LN:336933
@SQ	SN:NC_000087.7	LN:91744698
@SQ	SN:NT_187060.1	LN:182347
@SQ	SN:NT_187061.1	LN:259875
@SQ	SN:NT_187062.1	LN:155838
@SQ	SN:NT_187063.1	LN:158099
@SQ	SN:NT_166338.1	LN:40056
@SQ	SN:NT_166451.1	LN:42057
@SQ	SN:NT_166462.1	LN:31602
@SQ	SN:NT_166465.1	LN:25871
@SQ	SN:NT_166466.1	LN:23158
@SQ	SN:NT_166467.1	LN:38659
@SQ	SN:NT_166469.1	LN:35240
@SQ	SN:NT_166474.1	LN:24668
@SQ	SN:NT_166476.1	LN:23629
@SQ	SN:NT_166477.1	LN:55711
@SQ	SN:NT_166478.1	LN:24323
@SQ	SN:NT_166443.1	LN:22974
@SQ	SN:NT_166444.1	LN:31704
@SQ	SN:NT_166480.1	LN:21240
@SQ	SN:NT_166456.1	LN:28664
@SQ	SN:NT_166471.1	LN:24685
```

```
@SQ     SN:NT_166473.1  LN:28772
@SQ     SN:NT_166454.1  LN:26764
@SQ     SN:NT_166463.1  LN:72385
@SQ     SN:NT_166450.1  LN:47073
@SQ     SN:NT_166452.1  LN:20208
@SQ     SN:NT_187064.1  LN:114452
@SQ     SN:NW_016097314.1       LN:686942
@SQ     SN:NW_006763128.1       LN:456798
@SQ     SN:NW_016097312.1       LN:200008
@SQ     SN:NW_016097313.1       LN:266016
```

Followed by the alignment files:

In [32]:

```
tail SRR1654369.Aligned.out.sam
```

```
SRR1654369.31547757     355     NC_000081.6     36598202        3       5S46
M       =       36598989        818     GNNNNCNNNNNNNNGTGTTGACATGACTCGTGGAAC
CTGTGAGGAAGCTGG @&%%%=%###$$$$A;;?:8@>@@>>@@@??>>>>A@>@>?>?7>>4=?>?     NH:
i:2     HI:i:1  AS:i:66 nM:i:0
SRR1654369.31547757     403     NC_000081.6     36598989        3       31M2
0S      =       36598202        -818    TGGTCTTAGTTGAGCNATTTGGCTAGGAGGANNNNN
NNNNNNNNNNNNNNN ###################################################     NH:
i:2     HI:i:1  AS:i:66 nM:i:0
SRR1654369.31547757     163     NC_000083.6     26269753        3       20S3
1M      =       26269886        184     NNNNNNNNNNNNNNNNNNNNTCCTCCTAGCCAAATN
GCTCAACTAAGACCA ###################################################     NH:
i:2     HI:i:2  AS:i:67 nM:i:0
SRR1654369.31547757     83      NC_000083.6     26269886        3       51M
        =       26269753        -184    CCAGCTTCCTCACAGGTTCCACGAGTCATGTCAACA
CNNNNNNNNGNNNNC ?>?=4>>7?>?>@>@A>>>>??@@@>>@@>@8:?;;A$$$$###%=%%%&@     NH:
i:2     HI:i:2  AS:i:67 nM:i:0
SRR1654369.31547758     163     NC_000073.6     126979554       255     20S3
1M      =       126979653       150     NNNNNNNNNNNNNNNNNNNNCCGGATGGGCCTCTGN
GTCACAGCTCCCATC ###################################################     NH:
i:1     HI:i:1  AS:i:67 nM:i:0
SRR1654369.31547758     83      NC_000073.6     126979653       255     51M
        =       126979554       -150    TGCAGCAGACCGCGCCAAGAAGAAATGATCCTCTCC
CNNNNNNNNCNNNNC @>@@A@>@?>AA@@@@?@A@@A@AA@A@??@8A?A?A$$$$$$$%A&&''=     NH:
i:1     HI:i:1  AS:i:67 nM:i:0
SRR1654369.31547762     99      NC_000075.6     64186207        255     51M
        =       64186296        120     TNNNNGNNNNNNNNCTCATTTCAGTAGCAGCAAAGC
ATGCTTTCCATGCAT 9&%%%<%###$$$$A7:795A?@@@A@@A@@@@@:A@@9@>>>A@@>:8@=     NH:
i:1     HI:i:1  AS:i:67 nM:i:0
SRR1654369.31547762     147     NC_000075.6     64186296        255     31M2
0S      =       64186207        -120    ACTGAGAGGTCACCANCAAGAGTGAAGTACCNNNNN
NNNNNNNNNNNNNNN ###################################################     NH:
i:1     HI:i:1  AS:i:67 nM:i:0
SRR1654369.31547763     163     NC_000075.6     120955420       255     20S3
1M      =       120955482       113     NNNNNNNNNNNNNNNNNNNNACCTCACTTGCAATAN
TTACAAAAACAAGAT ###################################################     NH:
i:1     HI:i:1  AS:i:67 nM:i:0
SRR1654369.31547763     83      NC_000075.6     120955482       255     51M
        =       120955420       -113    CTTGTACGCACCGTCCTTCGTGCTGGTGACAGGGAA
GNNNNNNNNGNNNNT ######??8,5@A>>A9A@A@>?@>@>>>>>A<>1A6$$$$$$$%;%%%&7     NH:
i:1     HI:i:1  AS:i:67 nM:i:0
```

In [34]:

```
tail SRR1654370.Aligned.out.sam
```

```
SRR1654370.28007893     355     NC_000078.6     110691144       1       51M
        =       110691241       128     ANNNNGGNANNNNNGCCACTTGGGGAATTCCTGTCA
GGTTGTTTTAACTTG ?&%%%>=%@%&%%$;AAA;?A@@@A>>AA@A@@@A@@A>A@@A@@A@@@?A     NH:
i:3     HI:i:2  AS:i:67 nM:i:0
SRR1654370.28007893     403     NC_000078.6     110691241       1       31M2
0S      =       110691144       -128    AAAATTTTAAGATNTNNTTTAAGCCATCTTANNNNN
NNNNNNNNNNNNNNN ###################################################     NH:
i:3     HI:i:2  AS:i:67 nM:i:0
SRR1654370.28007893     419     NC_000077.6     77296167        1       20S3
1M      =       77296244        128     NNNNNNNNNNNNNNNNNNNNTAAGATGGCTTAAANN
ANATCTTAAAATTTT ###################################################     NH:
i:3     HI:i:3  AS:i:67 nM:i:0
SRR1654370.28007893     339     NC_000077.6     77296244        1       51M
        =       77296167        -128    CAAGTTAAAACAACCTGACAGGAATTCCCCAAGTGG
CNNNNNTNCCNNNNT A?@@@A@@A@@A>A@@A@@@A@AA>>A@@@A?;AAA;$%%&%@%=>%%%&?     NH:
i:3     HI:i:3  AS:i:67 nM:i:0
SRR1654370.28007894     99      NC_000068.7     4991144 255     51M     =
        4991225 112     CNNNNGTNANNNNNCAGTTCTGGGGCAGGCATCTGTGACTCAAGGAAAAGC
        7&%%%7=&7%%%%#434@33@@>1774?===>@A:@>9@<A>>>4>=A>@>     NH:i:1  HI:
i:1     AS:i:67 nM:i:0
SRR1654370.28007894     147     NC_000068.7     4991225 255     31M20S  =
        4991144 -112    AGTGTCAAATATGNTNNCAGGCCAGCACATGNNNNNNNNNNNNNNNNNNNN
        ###################################################     NH:i:1  HI:
i:1     AS:i:67 nM:i:0
SRR1654370.28007896     163     NC_000077.6     74667107        255     20S3
1M      =       74667158        102     NNNNNNNNNNNNNNNNNNNNTGGCCCTGTACTGCNN
CNCCAGCAGCCAGCT ###################################################     NH:
i:1     HI:i:1  AS:i:67 nM:i:0
SRR1654370.28007896     83      NC_000077.6     74667158        255     51M
        =       74667107        -102    TGTACCGTGCCCGCTACCTCATGCTGCTTGTGTTTG
GNNNNNANCANNNNG :@1=7@@@@@@@:::<87=2@@@@:@<5<<5@*77>1#%%%%*&57%%%&7     NH:
i:1     HI:i:1  AS:i:67 nM:i:0
SRR1654370.28007899     99      NC_000073.6     30224168        255     51M
        =       30224257        120     ANNNNAANANNNNNACAGGCATTCAGAGGGGCTGTG
TCAGGGGCGGGGCCT >&&%%=;%7%%%%$AAAA<?A@A@AA7@A@@AA@>@?@A>>@@@?@>>3A=     NH:
i:1     HI:i:1  AS:i:67 nM:i:0
SRR1654370.28007899     147     NC_000073.6     30224257        255     31M2
0S      =       30224168        -120    CTCCTCTGGGAAANCNNCTACGGTGACAGCTNNNNN
NNNNNNNNNNNNNNN ###################################################     NH:
i:1     HI:i:1  AS:i:67 nM:i:0
```

SAMfile converting to BAMfile (compressed smaller file), followed by indexing to BAIfile for quick access

The next step was to convert these sam files generated, to bam files. This was done using samtools.

In [41]:

```
pwd
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq
```

In [42]:

```
ls
```

```
2.5.2b.tar.gz                              sjdbList.fromGTF.out.tab
chrLength.txt                              sjdbList.out.tab
chrNameLength.txt                          SRR1654369_1.fastq
chrName.txt                                SRR1654369_2.fastq
chrStart.txt                               SRR1654369.Aligned.out.sam
exonGeTrInfo.tab                           SRR1654369.Log.final.out
exonInfo.tab                               SRR1654369.Log.out
GCF_000001635.25_GRCm38.p5_genomic.fna     SRR1654369.Log.progress.out
GCF_000001635.25_GRCm38.p5_genomic.gff     SRR1654369.SJ.out.tab
geneInfo.tab                               SRR1654370_1.fastq
Genome                                     SRR1654370_2.fastq
genomeParameters.txt                       SRR1654370.Aligned.out.sam
Log.out                                    SRR1654370.Log.final.out
Mouse.ES.Prep1_KO (RNA-Seq)_SRR1654370     SRR1654370.Log.out
Mouse.ES.wild_type (RNA-Seq)_SRR1654369    SRR1654370.Log.progress.out
quality control                            SRR1654370.SJ.out.tab
SA                                         STAR-2.5.2b
SAindex                                    to
sjdbInfo.txt                               transcriptInfo.tab
```

In [43]:

```
samtools sort -o SRR1654369.bam SRR1654369.Aligned.out.sam
```

```
[bam_sort_core] merging from 19 files...
```

In [44]:

```
ls -l SRR1654369*[bs]am
```

```
-rw-r--r-- 1 r0308983 domain users 14634176392 jan  3 21:56 SRR1654369.Align
ed.out.sam
-rw-r--r-- 1 r0308983 domain users  2859519581 jan  4 14:42 SRR1654369.bam
```

In [45]:

```
samtools sort -o SRR1654370.bam SRR1654370.Aligned.out.sam
```

```
[bam_sort_core] merging from 17 files...
```

As we can see, the output bam file is much smaller. We will now generate an index, again using `samtools`:

Next we also generated an index file of the bam files. The output is a bai file.

In [46]:

```
samtools index SRR1654369.bam
samtools index SRR1654370.bam
```

In [47]:

```
ls -l SRR1654369*[bs]a[mi]
ls -l SRR1654370*[bs]a[mi]
```

```
-rw-r--r-- 1 r0308983 domain users 14634176392 jan  3 21:56 SRR1654369.Align
ed.out.sam
-rw-r--r-- 1 r0308983 domain users  2859519581 jan  4 14:42 SRR1654369.bam
-rw-r--r-- 1 r0308983 domain users     2916240 jan  4 15:00 SRR1654369.bam.b
ai
-rw-r--r-- 1 r0308983 domain users 13023377608 jan  3 22:14 SRR1654370.Align
ed.out.sam
-rw-r--r-- 1 r0308983 domain users  2489262785 jan  4 14:58 SRR1654370.bam
-rw-r--r-- 1 r0308983 domain users     2927328 jan  4 15:00 SRR1654370.bam.b
ai
```

Inspect the BAM files

In [48]:

```
ls -l *.ba[mi]
```

```
-rw-r--r-- 1 r0308983 domain users 2859519581 jan  4 14:42 SRR1654369.bam
-rw-r--r-- 1 r0308983 domain users    2916240 jan  4 15:00 SRR1654369.bam.ba
i
-rw-r--r-- 1 r0308983 domain users 2489262785 jan  4 14:58 SRR1654370.bam
-rw-r--r-- 1 r0308983 domain users    2927328 jan  4 15:00 SRR1654370.bam.ba
i
```

In [49]:

```
samtools view SRR1654369.bam | head -3
```

```
SRR1654369.3797050      419     NC_000067.6     3040881 1       51M     =
        3041013 183     AGTTGACCATCTAAACAGTCCCATATCACCTAAAGAAATAGAAGCAGTTAT
        GGGGEGGGGGGGGGGGFGFGGGGGFGGGFGGFFEG?EAEDG?GGBGG=F?F     NH:i:4  HI:
i:3     AS:i:96 nM:i:2
SRR1654369.3797050      339     NC_000067.6     3041013 1       51M     =
        3040881 -183    TCCAATTCTGCACAAACTATTTCACAAATAGAAGTAGAAGGTACTCTACCC
        FGGGDGEGFEGFGGGEEEDFGFFFGGGGGGGGGGGGGGGGGGGGFEFFF     NH:i:4  HI:
i:3     AS:i:96 nM:i:2
SRR1654369.17785009     163     NC_000067.6     3042981 255     50M1S   =
        3043079 149     GCAGGCTTGTACAAACACTCTGGGAAACAGTCTGGCGGGTCCTCCGGGAAA
        AAAAAAAA,?'23?#####################################     NH:i:1  HI:
i:1     AS:i:89 nM:i:5
samtools view: writing to standard output failed: Broken pipe
samtools view: error closing standard output: -1
```

In [50]:

```
samtools view SRR1654370.bam | head -3
```

```
SRR1654370.6814704      99      NC_000067.6     3062511 255     51M     =
        3062604 144     CTGAGGAAGTCATACGGCGCACATGTCTTACACGCTATTCCCAGTTTATAC
        ##################################################      NH:i:1  HI:
i:1     AS:i:96 nM:i:2
SRR1654370.6814704      147     NC_000067.6     3062604 255     51M     =
        3062511 -144    GCCTCTGTTCCAGGAATTAATTTGCTTTGTGCAATTCAGCTCTTAGCTGAG
        A5>=:5CC;CC;6C;?==BB::5=5=;AA6?;5;-9;;=9-5=:;=85959     NH:i:1  HI:
i:1     AS:i:96 nM:i:2
SRR1654370.17730201     99      NC_000067.6     3063443 255     51M     =
        3063457 65      AGCTACCATGTGGTTGCTGGGATTTGAACTCAGGACCTTCGGAAGAGCAGT
        FFFFFFFEFFDFFDEDBBAAB?EEEBFEFFFFDEEDDEEDEBD?E@EDBE?     NH:i:1  HI:
i:1     AS:i:100        nM:i:0
samtools view: writing to standard output failed: Broken pipe
samtools view: error closing standard output: -1
```

Next, we will look how many reads map to each accession number corresponding to each chromosome of mm10 using idxstats.

From left to right: reference sequence name, sequence length, # mapped reads and # unmapped reads.

Al the NC files seen below represent a chromosome. The exact chromosome can be found by looking up the NC number on https://www.ncbi.nlm.nih.gov/genome/?term=mus+musculus+9 (https://www.ncbi.nlm.nih.gov/genome/?term=mus+musculus+9). There are also a lot of NT numbers, these files can be looked up in the nucleotide database of NCBI. For example: NT_166280.1 is an unlocalized genomic contig. This means that it is a sequence found in an assembly that is associated with a specific chromosome but cannot be ordered or oriented on that chromosome. Apart from the NC and NT numbers, there are also some NW numbers, which are patches. Patches are accessioned scaffold sequences that represent assembly updates.

The largest number of mapped reads corresponds to the chromosomes while only a small number of reads are mapped to the NT and NW numbers.

In [51]:

```
samtools idxstats SRR1654369.bam | head -20
```

```
NC_000067.6     195471971       3580680 0
NT_166280.1     169725  94      0
NT_166281.1     241735  150     0
NT_166282.1     153618  120     0
NT_166283.1     39340   188     0
NT_162750.1     206961  136     0
NC_000068.7     182113224       4326938 0
NC_000069.6     160039680       2551350 0
NC_000070.6     156508116       3764656 0
NT_166291.1     66673   2152    0
NT_187052.1     14945   0       0
NT_166434.1     227966  406     0
NT_187053.1     207968  596     0
NT_187054.1     191905  962     0
NT_187055.1     1976    112     0
NC_000071.6     151834684       3710772 0
NT_187056.1     199368  96      0
NT_187057.1     205776  52      0
NT_187058.1     184189  30      0
NT_166438.1     195993  78      0
```

In [52]:

```
samtools idxstats SRR1654370.bam | head -20
```

```
NC_000067.6     195471971       3618714 0
NT_166280.1     169725  140     0
NT_166281.1     241735  274     0
NT_166282.1     153618  216     0
NT_166283.1     39340   218     0
NT_162750.1     206961  254     0
NC_000068.7     182113224       3725774 0
NC_000069.6     160039680       2193986 0
NC_000070.6     156508116       3291536 0
NT_166291.1     66673   1920    0
NT_187052.1     14945   0       0
NT_166434.1     227966  436     0
NT_187053.1     207968  512     0
NT_187054.1     191905  774     0
NT_187055.1     1976    146     0
NC_000071.6     151834684       3226770 0
NT_187056.1     199368  80      0
NT_187057.1     205776  52      0
NT_187058.1     184189  24      0
NT_166438.1     195993  96      0
```

Next, we run samtools flagstats. This tells us what the distribution of mapping flags is. e.g. P means the read is paired in sequencing.

In [53]:

```
samtools flagstat SRR1654369.bam
```

```
72424586 + 0 in total (QC-passed reads + QC-failed reads)
16685732 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
72424586 + 0 mapped (100.00% : N/A)
55738854 + 0 paired in sequencing
27869427 + 0 read1
27869427 + 0 read2
55738854 + 0 properly paired (100.00% : N/A)
55738854 + 0 with itself and mate mapped
0 + 0 singletons (0.00% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)
```

In [54]:

```
samtools flagstat SRR1654370.bam
```

```
64511762 + 0 in total (QC-passed reads + QC-failed reads)
14879618 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
64511762 + 0 mapped (100.00% : N/A)
49632144 + 0 paired in sequencing
24816072 + 0 read1
24816072 + 0 read2
49632144 + 0 properly paired (100.00% : N/A)
49632144 + 0 with itself and mate mapped
0 + 0 singletons (0.00% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)
```

We also run samtools stats. The output is below.

In [55]:

```
samtools stats SRR1654369.bam
```

Reads to gene counts

Next, we will count all the mapped reads per gene.

Looking at our annotation gff file, which we downloaded from https://www.ncbi.nlm.nih.gov/genome/?term=mus%20musculus%209 (https://www.ncbi.nlm.nih.gov/genome/?term=mus%20musculus%209) , we are able to determine on what we needed to filter.

Cufflinks was not yet installed on the server, so we couldn't convert our gff file to a gtf file with gffread. But it didn't matter because it also works with the gff file.

In [56]:

```
head -10 GCF_000001635.25_GRCm38.p5_genomic.gff
```

```
##gff-version 3
#!gff-spec-version 1.21
#!processor NCBI annotwriter
#!genome-build GRCm38.p5
#!genome-build-accession NCBI_Assembly:GCF_000001635.25
##sequence-region NC_000067.6 1 195471971
##species https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?id=10090
(https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?id=10090)
NC_000067.6     RefSeq  region  1       195471971       .       +       .
        ID=id0;Dbxref=taxon:10090;Name=1;chromosome=1;gbkey=Src;genome=chrom
osome;mol_type=genomic DNA;strain=C57BL/6J
NC_000067.6     BestRefSeq%2CGnomon     gene    3199733 3672278 .       -
        .       ID=gene0;Dbxref=GeneID:497097,MGI:MGI:3528744;Name=Xkr4;desc
ription=X-linked Kx blood group related 4;gbkey=Gene;gene=Xkr4;gene_biotype=
protein_coding;gene_synonym=AY534250,Gm210,mKIAA1889,XRG4
NC_000067.6     Gnomon  mRNA    3199733 3672278 .       -       .       ID=r
na0;Parent=gene0;Dbxref=GeneID:497097,Genbank:XM_006495550.3,MGI:MGI:352874
4;Name=XM_006495550.3;gbkey=mRNA;gene=Xkr4;model_evidence=Supporting evidenc
e includes similarity to: 3 mRNAs%2C 32 ESTs%2C 4 Proteins%2C 110 long SRA r
eads%2C and 99%25 coverage of the annotated genomic feature by RNAseq alignm
ents%2C including 7 samples with support for all annotated introns;product=X
-linked Kx blood group related 4%2C transcript variant X1;transcript_id=XM_0
06495550.3
```

In [61]:

```
pwd
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq
```

The command we used below compiles the number of mapped reads per gene of all the samples together in one file. Q is the quality controle. We used the same filter as in class. g shows us the mapped reads per gene. a is the path to the annotation file. o determines the output file followed by all the bam files that are the input of this command.

In [62]:

```
featureCounts -Q 10 -g gene -a GCF_000001635.25_GRCm38.p5_genomic.gff -o test1_GCF_0000016
# -Q 10: de kwaliteit van de reads
# -g gene: filteren op gene name
#-a: file die hierna komt gebruiken om te annoteren
# file die erna komt wordt de output
```

```
        ==========     _____ _    _ ____  _____  ______          _____
        =====         / ____| |  | |  _ \|  __ \|  ____|   /\   |  __ \
          =====      | (___ | |  | | |_) | |__) | |__     /  \  | |  | |
            ====      \___ \| |  | |  _ <|  _  /|  __|   / /\ \ | |  | |
              ====    ____) | |__| | |_) | | \ \| |____ / ____ \| |__| |
        ==========   |_____/ \____/|____/|_|  \_\______/_/    \_\_____/
        v1.5.0-p1

//========================== featureCounts setting ===========================
==\\
||
  ||
||            Input files : 2 BAM files
  ||
||                          P SRR1654369.bam
  ||
||                          P SRR1654370.bam
  ||
||
  ||
||            Output file : test1_GCF_000001635.25_GRCm38.p5_genomic.counts
  ||
||            Annotations : GCF_000001635.25_GRCm38.p5_genomic.gff (GTF)
  ||
||
  ||
||                Threads : 1
  ||
||                  Level : meta-feature level
  ||
||             Paired-end : no
  ||
||        Strand specific : no
  ||
||      Multimapping reads : not counted
  ||
|| Multi-overlapping reads : not counted
  ||
||       Read orientations : fr
  ||
||
  ||
\\====================== http://subread.sourceforge.net/ (http://subread.sour
ceforge.net/) ======================//

//================================== Running =================================
==\\
||
  ||
|| Load annotation file GCF_000001635.25_GRCm38.p5_genomic.gff ...
  ||
||    Features : 1225991
  ||
||    Meta-features : 41575
  ||
||    Chromosomes/contigs : 185
  ||
||
  ||
```

```
  ..
|| Process BAM file SRR1654369.bam...
  ||
||    Paired-end reads are included.
  ||
||    Assign reads to features...
  ||
||    Total reads : 72424586
  ||
||    Successfully assigned reads : 37653664 (52.0%)
  ||
||    Running time : 1.37 minutes
  ||
||
  ||
|| Process BAM file SRR1654370.bam...
  ||
||    Paired-end reads are included.
  ||
||    Assign reads to features...
  ||
||    Total reads : 64511762
  ||
||    Successfully assigned reads : 32388672 (50.2%)
  ||
||    Running time : 1.21 minutes
  ||
||
  ||
||                         Read assignment finished.
  ||
||
  ||
\\====================== http://subread.sourceforge.net/ (http://subread.sour
ceforge.net/) ======================//
```

In [63]:

```
head -20 test1_GCF_000001635.25_GRCm38.p5_genomic.counts.summary | column -t
```

```
Status                     SRR1654369.bam  SRR1654370.bam
Assigned                   37653664        32388672
Unassigned_Ambiguity       1302241         1131854
Unassigned_MultiMapping    0               0
Unassigned_NoFeatures      7491569         7642104
Unassigned_Unmapped        0               0
Unassigned_MappingQuality  25977112        23349132
Unassigned_FragmentLength  0               0
Unassigned_Chimera         0               0
Unassigned_Secondary       0               0
Unassigned_Nonjunction     0               0
Unassigned_Duplicate       0               0
```

In [64]:

```
head test1_GCF_000001635.25_GRCm38.p5_genomic.counts
```

```
# Program:featureCounts v1.5.0-p1; Command:"featureCounts" "-Q" "10" "-g" "g
ene" "-a" "GCF_000001635.25_GRCm38.p5_genomic.gff" "-o" "test1_GCF_00000163
5.25_GRCm38.p5_genomic.counts" "SRR1654369.bam" "SRR1654370.bam"
Geneid  Chr     Start   End     Strand  Length  SRR1654369.bam  SRR1654370.b
am
Xkr4    NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6     3199
733;3213439;3322816;3421702;3670552     3207317;3216968;3323760;3421901;3672
278     -;-;-;-;-       13987   155     98
Gm39585 NC_000067.6;NC_000067.6 3361552;3368342 3366505;3377812 +;+     1442
5       8       12
LOC108167595    NC_000067.6     3671524 3672264 -       741     0       0
Gm38717 NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_00006
7.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;
NC_000067.6;NC_000067.6;NC_000067.6     4115673;4119668;4120015;4142612;4147
812;4148612;4163855;4170205;4197534;4206660;4226611;4228443;4231053;4233184;
4243133 4115948;4119712;4120073;4142766;4147963;4148744;4163941;4170404;4197
641;4206837;4226823;4228619;4231144;4233728;4243365     -;-;-;-;-;-;-;-;-;-;
-;-;-;-;-       2653    0       0
Rp1     NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_00006
7.6     4290846;4343507;4351910;4352202;4360200;4409170 4293012;4350091;4352
081;4352837;4360314;4409266     -;-;-;-;-;-     9772    0       0
Sox17   NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6 4490928;4493100;4493
772;4496291     4492668;4493490;4495942;4497386 -;-;-;- 5399    29      11
Gm7357  NC_000067.6;NC_000067.6 4522817;4524432 4523591;4526931 +;+     3275
        4       6
LOC108167609    NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6 4583127;4583
406;4598113;4608720     4583243;4583478;4598958;4608781 -;-;-;- 1098    0
        0
```

Note that there are a number of columns on the gene structure, and a number with the actual counts. We'll separate these to have a file containing the gene name, the chromosome on which the gene is located, the start and end of the gene, which strand it is located on and the length of the gene.

In [65]:

```
cut -f-6 test1_GCF_000001635.25_GRCm38.p5_genomic.counts > genedata.test1.tsv
```

In [66]:

```
head genedata.test1.tsv
```

```
# Program:featureCounts v1.5.0-p1; Command:"featureCounts" "-Q" "10" "-g" "g
ene" "-a" "GCF_000001635.25_GRCm38.p5_genomic.gff" "-o" "test1_GCF_00000163
5.25_GRCm38.p5_genomic.counts" "SRR1654369.bam" "SRR1654370.bam"
Geneid  Chr     Start   End     Strand  Length
Xkr4    NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6     3199
733;3213439;3322816;3421702;3670552     3207317;3216968;3323760;3421901;3672
278     -;-;-;-;-       13987
Gm39585 NC_000067.6;NC_000067.6 3361552;3368342 3366505;3377812 +;+     1442
5
LOC108167595    NC_000067.6     3671524 3672264 -       741
Gm38717 NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_00006
7.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;
NC_000067.6;NC_000067.6;NC_000067.6     4115673;4119668;4120015;4142612;4147
812;4148612;4163855;4170205;4197534;4206660;4226611;4228443;4231053;4233184;
4243133 4115948;4119712;4120073;4142766;4147963;4148744;4163941;4170404;4197
641;4206837;4226823;4228619;4231144;4233728;4243365     -;-;-;-;-;-;-;-;-;-;
-;-;-;-;-       2653
Rp1     NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6;NC_00006
7.6     4290846;4343507;4351910;4352202;4360200;4409170 4293012;4350091;4352
081;4352837;4360314;4409266     -;-;-;-;-;-     9772
Sox17   NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6 4490928;4493100;4493
772;4496291     4492668;4493490;4495942;4497386 -;-;-;- 5399
Gm7357  NC_000067.6;NC_000067.6 4522817;4524432 4523591;4526931 +;+     3275
LOC108167609    NC_000067.6;NC_000067.6;NC_000067.6;NC_000067.6 4583127;4583
406;4598113;4608720     4583243;4583478;4598958;4608781 -;-;-;- 1098
```

The next command will generate a file containing the gene names with the corresponding number of mapped reads.

In [67]:

```
cut -f1,7- test1_GCF_000001635.25_GRCm38.p5_genomic.counts | grep -v '^#' > gene.counts.tes
```

In [68]:

```
head gene.counts.test1.tsv
```

```
Geneid  SRR1654369.bam  SRR1654370.bam
Xkr4    155     98
Gm39585 8       12
LOC108167595    0       0
Gm38717 0       0
Rp1     0       0
Sox17   29      11
Gm7357  4       6
LOC108167609    0       0
Gm39586 0       0
```

In [69]:

```
tail gene.counts.test1.tsv
```

```
TrnD    0       0
TrnK    0       0
TrnG    1       0
TrnR    0       0
TrnH    83      131
TrnS2   6       16
TrnL2   44      47
TrnE    1341    1865
TrnT    135     272
TrnP    298     506
```

This file shows the gene names with their corresponding number of mapped reads per sample.

Little testquestion to ourselves: how many counts has the gene Pknox1 and Lypla1?

In [74]:

```
grep Pknox1 gene.counts.test1.tsv
```

```
Pknox1  1376    485
```

The first value is from the WT condition and the second from the KO condition

	A	B	C
1	Geneid	SRR1654369.bam (WT)	SRR1654370.bam (KO)
2	Pknox1	1376	485

In [72]:

```
grep Lypla1 gene.counts.test1.tsv
```

```
Lypla1  1664    2194
```

RNAseq analysis workflow using DESeq2

For this part of the examtask, we are going to work in an R notebook (see notebook examenopdracht R.ipynb) because this will be done in R in a seperate notebook. We use the output generated to look at the differential expression using DESeq2.

This generated a file with 303 differentially expressed genes.

GORILLA: Gene ontology

The data analysis was perfomed by using excel. The fold-changes of the Prep1 KO compared to the mouse wild-type were calculated and double checked with the output file from DESeq2 where the 2fold-log changes were already incorporated. The genes and their corresponding log2 transformations were sorted from high to low. The values greater than 0.5849 were genes being up regulated and all values less than -0.5849 (or FC =0.666) were the down regulated genes. The gene names were uploaded in Gorilla and the output of the

downregulated genes is visualised (figure 1). We selected the downregulation genes since the output revealed that the most frequent effect of the absence of Prep1 was a down-regulation of gene expression. These down-regulated genes had important functions in mouse development.

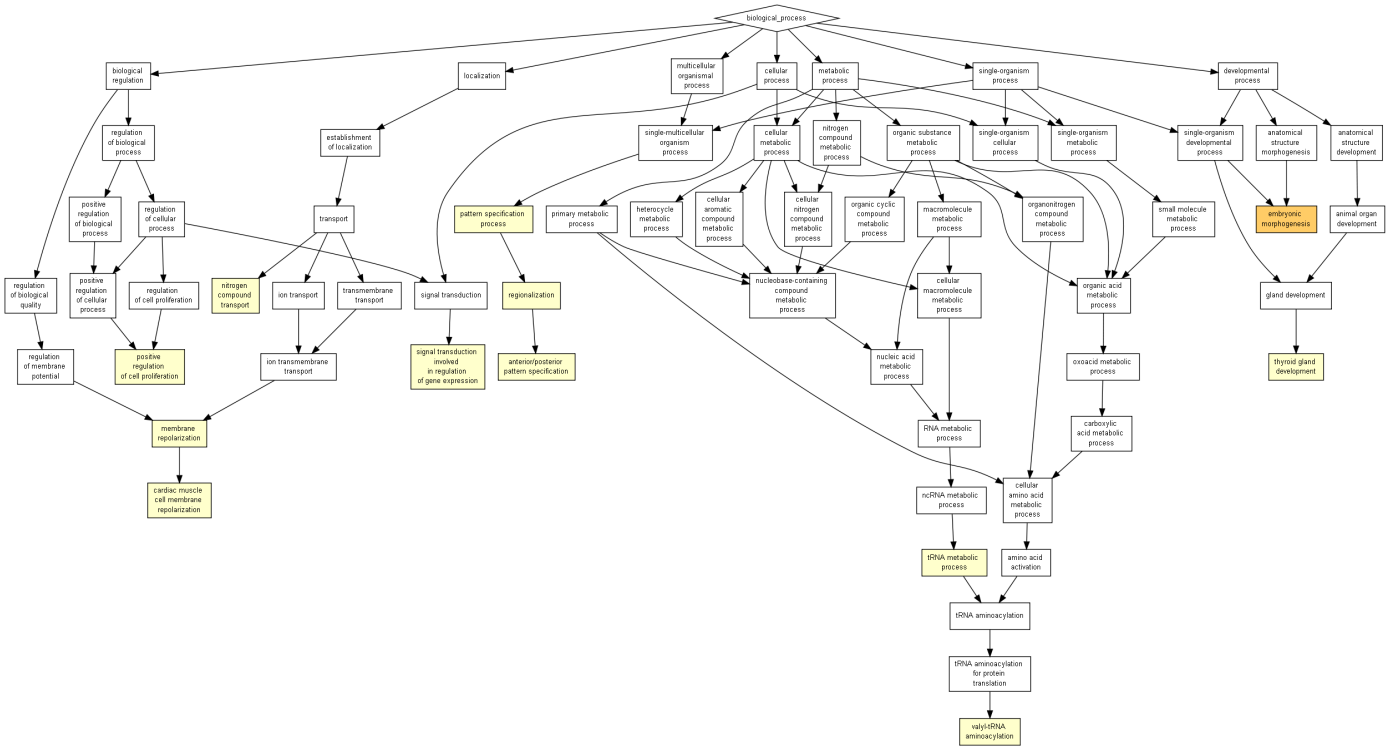


Fig 1: Downregulated genes visualised using Gorilla

GOrilla is a web-based application that identifies enriched GO terms in ranked lists of genes. This is particularly useful in many typical cases where genomic data may be naturally represented as a ranked list of genes (e.g. by level of expression or of differential expression). GOrilla computes an exact p-value for the observed enrichment, taking threshold multiple testing into account without the need for simulations. The output of the enrichment analysis is visualized as a hierarchical structure, providing a clear view of the relations between enriched GO terms.

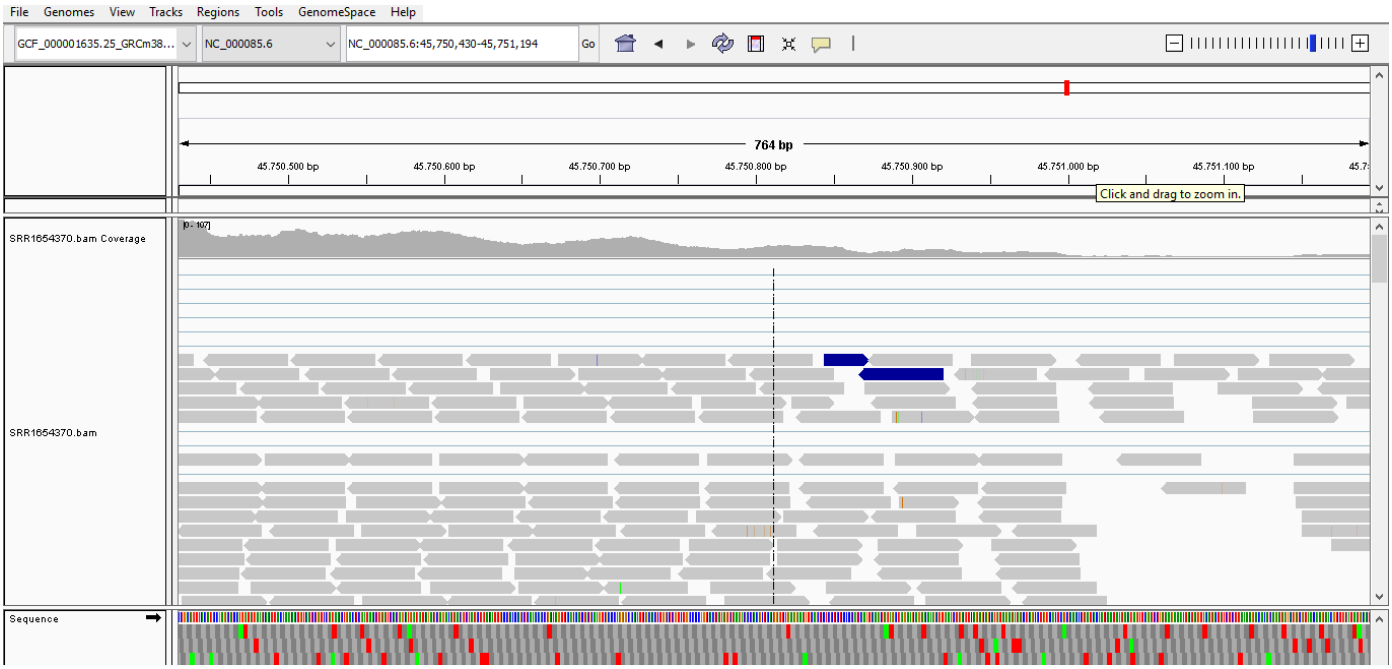
These results seem to indicate that PREP1 KO affects processes linked to embryonic morphogenesis, anterior/ posterior pattern specification and the regionalization.

GO term	Description	P-value	FDR q-value	Genes
GO:0048598	embryonic morphogenesis	5.16E-7	4.88E-3	**Fgf8 - fibroblast growth factor 8**
				Wnt3 - wingless-related mmtv
				Wnt3 - wingless-related mmtv
				Rspo3 - r-spondin 3 homolog (xenopus laevis)
				Hoxa3 - homeobox a3
				Sp8 - trans-acting transcription factor 8
				Hoxa10 - homeobox a10
				Otx1 - orthodenticle homolog 1 (drosophila)
GO:0009952	anterior/posterior pattern specification	2.2E-5	1.04E-1	**Wnt3 - wingless-related mmtv**
				Cdx2 - caudal type homeobox 2
				Hoxa3 - homeobox a3
GO:0003002	regionalization	6.78E-5	2.14E-1	**Fgf8 - fibroblast growth factor 8**
				Wnt3 - wingless-related mmtv
				Wnt8a - wingless-related mmtv
				Cdx2 - caudal type homeobox 2
				Hoxa3 - homeobox a3
				Sp8 - trans-acting transcription factor 8
				Otx1 - orthodenticle homolog 1

We were able to confirm three important downregulated transcription factors, Fgf8 (fibroblast growth factor 8), Wnt 3, Wnt9b (wingless-type mmtv integration site 3 and 9b). These confirms the results of the article where they also found that genes of the Fgf patway are affected in Prep1 knockout mice. It is claimed that depletion of PREP1 in ESC would induce a deficient embryoid bodies formation and differentiation. Re-establishment of the Prep1 level would rescue the phenotype.

IGV visualisation

To visualize the data we uploaded our reference genome (GCF_000001635.25_GRCm38.p5_genomic) and the files: SRR1654370.bam and SRR1654370.bam.bai into IGV. RNA-seq data showed that genes of the Fgf pathway are affected by the Prep1 deletion. As Fgf signaling is important for the first steps of the ES differentiation and primitive endoderm formation, we are going to take a look in IGV to one of the Fgf genes that GOrilla showed to be significant: Fgf8. We looked up where Fgf8 was located in the mouse genome and found the position: NC_000085.6 (45736798..45742938, complement) on NCBI. In IGV we see that around this position in the mouse genome, a couple of mutations occur.



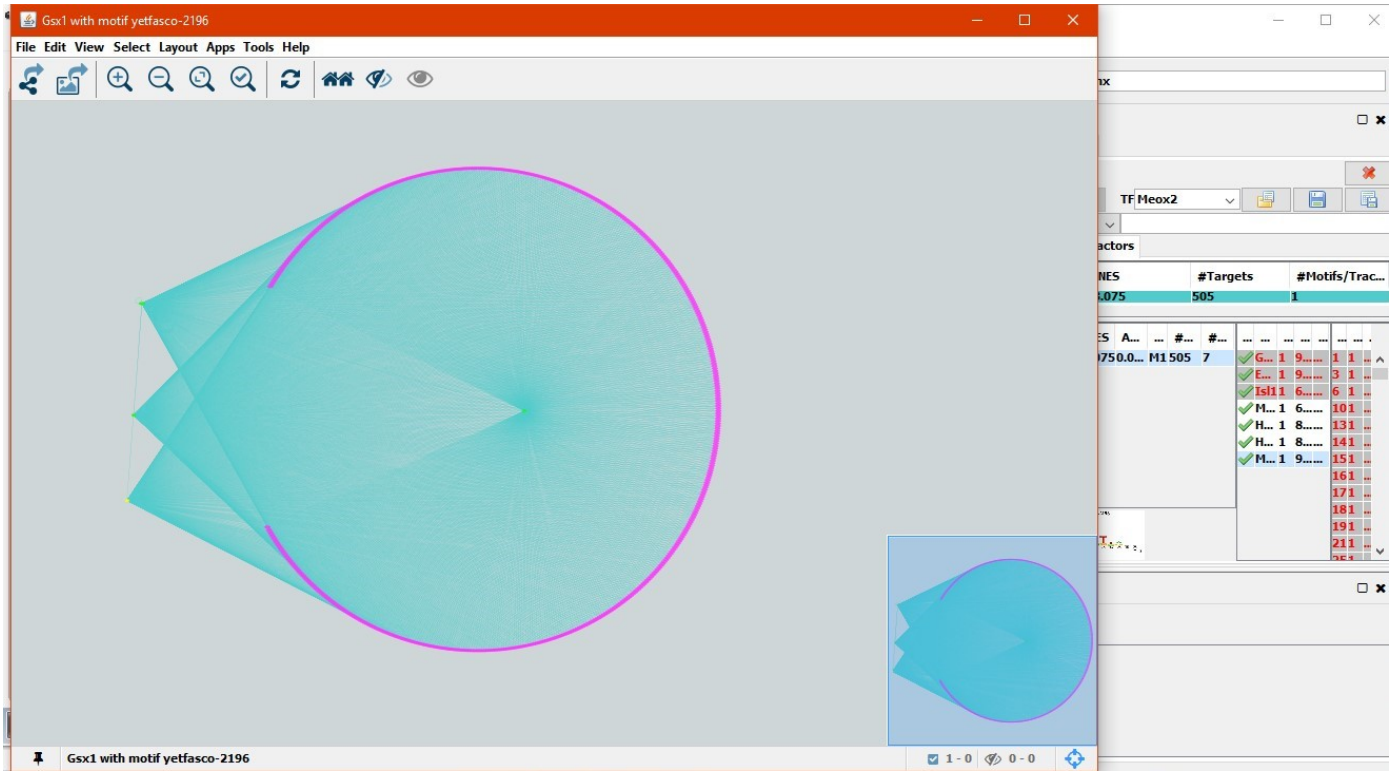
In the article RT-qPCR analysis of differentiation Prep1 KO EBs showed an great abberant expression of Fgf4. Fgf4 levels were about 30% lower in Prep1 KO ES cells compared to the WT. The mechanism by which Prep1 regulates the levels of Fgf4 is still unknown. If we look up the position of Fgf4 in the mouse genome we find that it is located on NC_000073.6 (144861386..144865243). This location was searched in IGV and the following figure shows the output. We see that on the position of the Fgf4 gene in the mouse, a lot of mutation occur compared the the reference genome of WT mice.

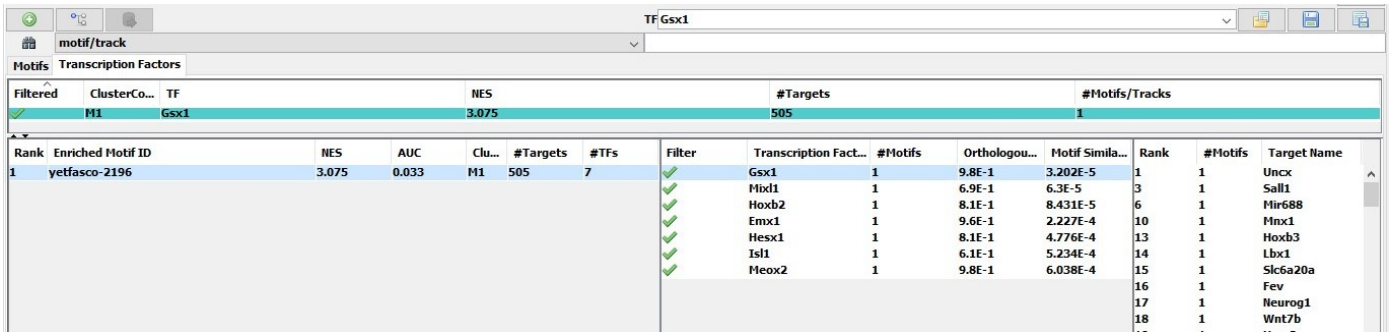


iRegulon

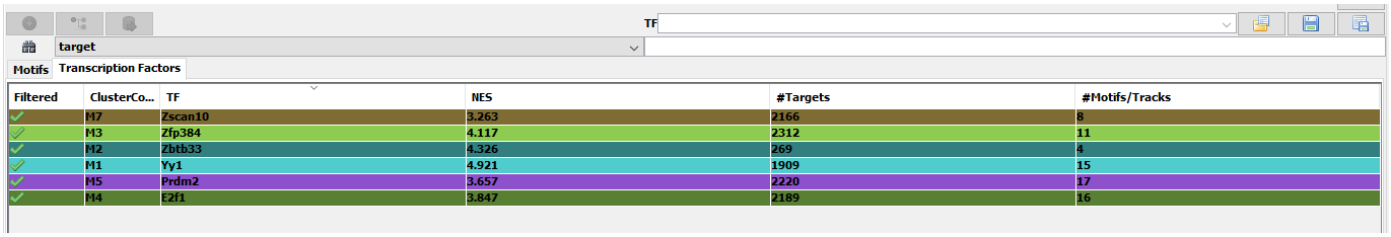
iRegulon: Sequence-based Discovery of Regulons and iRegulon detects the TF (Transcription Factors), the targets and the motifs/tracks from a set of genes. We load the list of genes (.txt file) in the iregulon app of cytoscape. A "network" of unconnected nodes will be created. Then, we use the Query iRegulon to predict the regulators and targets. We uploaded two different files in iRegulon the predict the network.

The first file contained the list of all genes that were or up- or downregulated. Here, we find that our main regulator the gene Gsx-1 is. Gsx-1 has 505 targets and is associated with 7 other TF. In the picture below the network is shown. The Gsx-1 or GS homeobox 1 has different important functions in the mouse. For example: the gene plays an important role in pituitary development. During the early phase of neurogenesis, Gsh1/2 coordinately regulates the expression of Tlx3, which is a critical postmitotic determinant for dorsal glytamatergic sensory interneurons. Other associated processes are: positive regulation of transcription from RNA polymerase II promoter and spinal cord association neuron differentiation.





The second file that was uploaded into iRegulon shows a complete different picture. Here, we selected the genes that were downregulated since we have seen that the most frequent effect was a down-regulation in the KO mouse. Instead of one main regulator, there are 6 regulators all associated with many targets. The network is not shown since it gives no clear image of which genes are connected to which genes and no information can be seen on it.



ChIPSeq

The two different ChIP seq samples (sra type) were downloaded from the GEO Accession viewer, and converted into “fastq” files by using the fastq-dump command. The first sample is the Prep1 (ChIP-Seq), and this will inform us about the function of the Prep1. Afterwards a star alignment was performed. We could have done it with the the reference genome we made ourselves (see RNAseq), but given that there is a new version available on the server (mm10) we decided to use this one for mapping the reads to the genome.

After the conversion of the ".sam file" into the ".bam file", the macs2 command was used to create the narrowpeak file. The peak calling was done against the reference genome of mm10. These files together with the "sorted.bam files" were loaded into IGV, and then compared with the results from the paper.

The ChIP-seq will allows us to see whether the downregulated genes found during the RNA seq are due to a direct effect of the PREP, which means that PREP1 directly binds the transcription site of this gene, or whether this is an indirect effect.

In [75]:

```
cd /mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq
```

In [76]:

```
pwd
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq
```

Fist step: creating a symbolic link to the raw fastq files. (a symbolic link instead of a full copy)

In [213]:

```
ln -sf /mnt/storage/r0308983/jupyternotebooks/examwork/~/data/ncbi/sra/SRR1652476_1.fastq /
ln -sf /mnt/storage/r0308983/jupyternotebooks/examwork/~/data/ncbi/sra/SRR1652477_1.fastq /
```

Quality control using FASTQC

After receiving the sequencing data, our first step is to check our data quality. This step helps to identify potential technical biases that might affect our analysis. code: fastqc fastqfile -o . outputdir

In [7]:

```
cd /mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq
pwd
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq
```

In [163]:

```
/usr/local/bin/FastQC/fastqc -o . SRR1652476_1.fastq
/usr/local/bin/FastQC/fastqc -o . SRR1652477_1.fastq
```

```
Started analysis of SRR1652476_1.fastq
Too many tiles (>500) so giving up trying to do per-tile qualities since w
e're probably parsing the file wrongly
Approx 5% complete for SRR1652476_1.fastq
Approx 10% complete for SRR1652476_1.fastq
Approx 15% complete for SRR1652476_1.fastq
Approx 20% complete for SRR1652476_1.fastq
Approx 25% complete for SRR1652476_1.fastq
Approx 30% complete for SRR1652476_1.fastq
Approx 35% complete for SRR1652476_1.fastq
Approx 40% complete for SRR1652476_1.fastq
Approx 45% complete for SRR1652476_1.fastq
Approx 50% complete for SRR1652476_1.fastq
Approx 55% complete for SRR1652476_1.fastq
Approx 60% complete for SRR1652476_1.fastq
Approx 65% complete for SRR1652476_1.fastq
Approx 70% complete for SRR1652476_1.fastq
Approx 75% complete for SRR1652476_1.fastq
Approx 80% complete for SRR1652476_1.fastq
Approx 85% complete for SRR1652476_1.fastq
Approx 90% complete for SRR1652476_1.fastq
Approx 95% complete for SRR1652476_1.fastq
Analysis complete for SRR1652476_1.fastq
Started analysis of SRR1652477_1.fastq
Too many tiles (>500) so giving up trying to do per-tile qualities since w
e're probably parsing the file wrongly
Approx 5% complete for SRR1652477_1.fastq
Approx 10% complete for SRR1652477_1.fastq
Approx 15% complete for SRR1652477_1.fastq
Approx 20% complete for SRR1652477_1.fastq
Approx 25% complete for SRR1652477_1.fastq
Approx 30% complete for SRR1652477_1.fastq
Approx 35% complete for SRR1652477_1.fastq
Approx 40% complete for SRR1652477_1.fastq
Approx 45% complete for SRR1652477_1.fastq
Approx 50% complete for SRR1652477_1.fastq
Approx 55% complete for SRR1652477_1.fastq
Approx 60% complete for SRR1652477_1.fastq
Approx 65% complete for SRR1652477_1.fastq
Approx 70% complete for SRR1652477_1.fastq
Approx 75% complete for SRR1652477_1.fastq
Approx 80% complete for SRR1652477_1.fastq
Approx 85% complete for SRR1652477_1.fastq
Approx 90% complete for SRR1652477_1.fastq
Approx 95% complete for SRR1652477_1.fastq
Analysis complete for SRR1652477_1.fastq
```

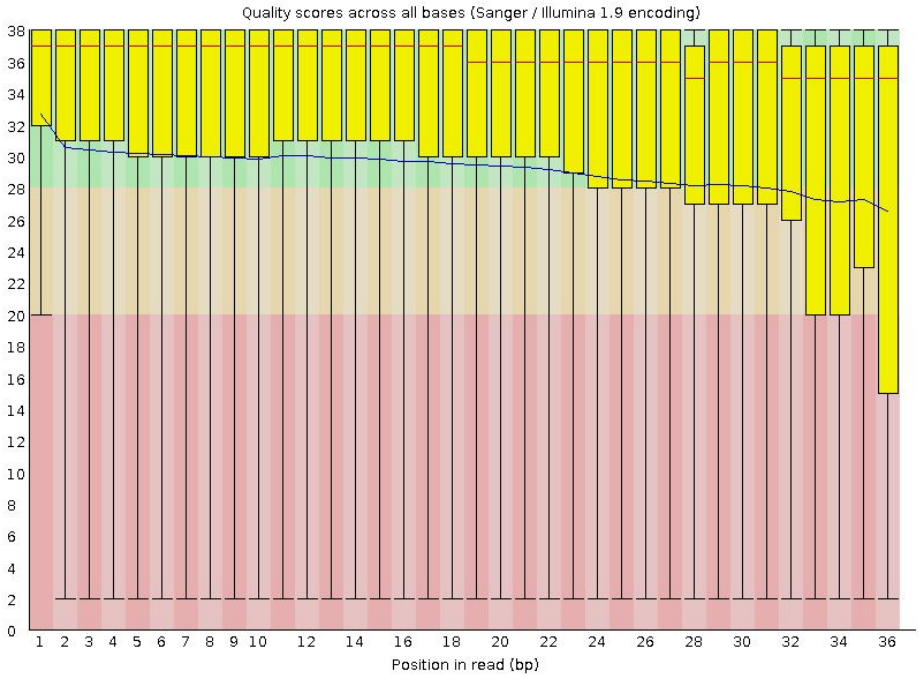
Quality control Prep1 (ChIP-Seq)_SRR1652476

FastQC Report

Summary

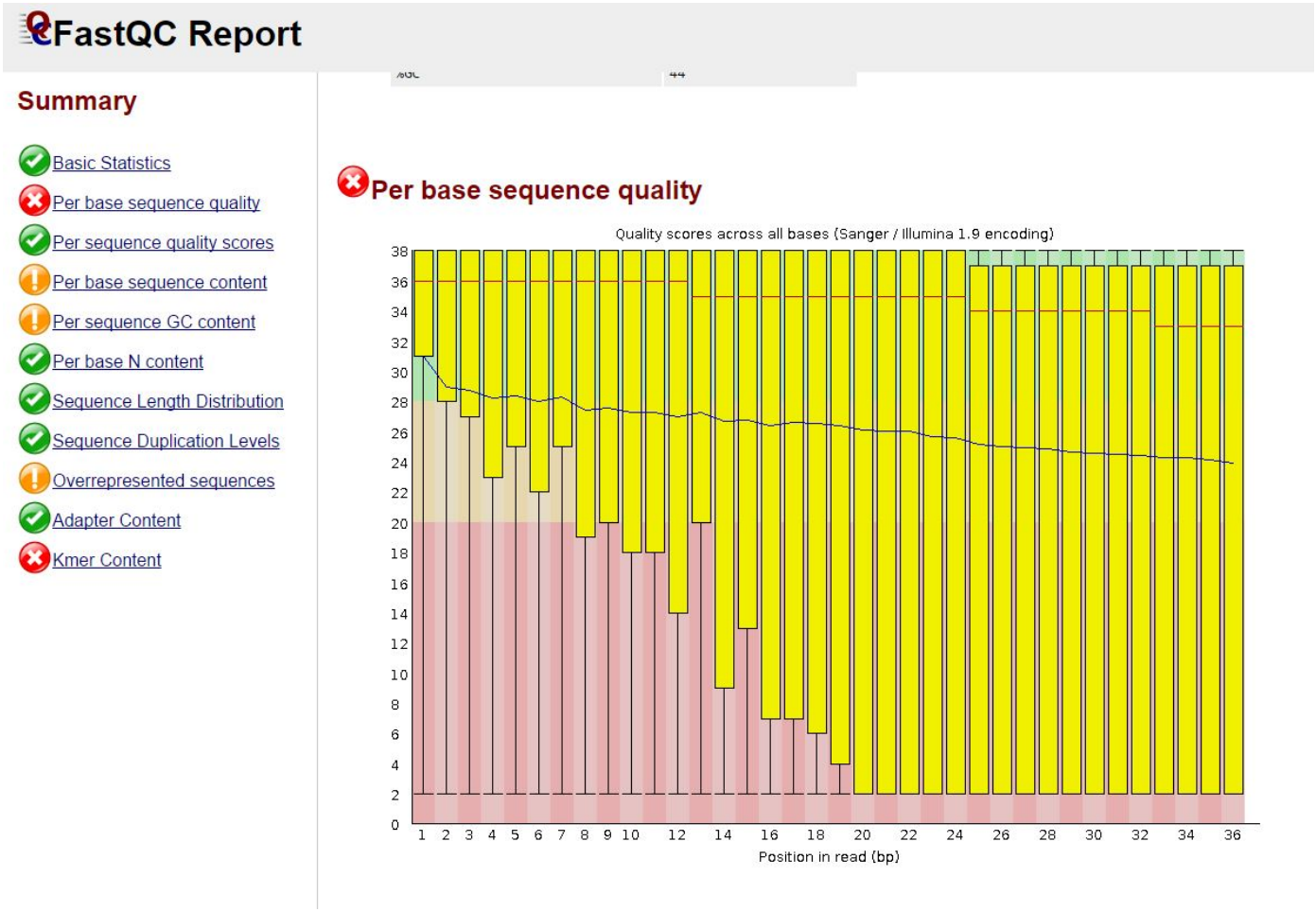
- Basic Statistics
- Per base sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content

Per base sequence quality



The overall quality of the reads seems good. The per base sequence quality is good. There is only a warning for the overrepresented sequences.

Quality control Input_DNA (ChIP-Seq control)_SRR1652477



Fastqc quality of the control file (input_DNA) is very bad (see picture above). With 'fastx toolkit' (clipper or trimmer) or with 'fastq quality filter', we could clean up this file. But then a lot of reads would be missed and we would have less peaks. Also this is just a control file.

In [164]:

```
cd /mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq
```

In [168]:

```
pwd
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq
```

In [8]:

```
ls -l
```

```
total 3644508
drwxr-xr-x 2 r0308983 domain users       4096 jan 24 11:44 macs2_Prep1
drwxr-xr-x 2 r0308983 domain users       4096 jan 10 16:31 macs2test
drwxr-xr-x 2 r0308983 domain users       4096 jan 10 19:20 quality control
lrwxrwxrwx 1 r0308983 domain users         82 jan 25 13:53 SRR1652476_1.fast
q -> /mnt/storage/r0308983/jupyternotebooks/examwork/~/data/ncbi/sra/SRR1652
476_1.fastq
-rw-r--r-- 1 r0308983 domain users  707924612 jan  8 16:17 SRR1652476.bam
-rw-r--r-- 1 r0308983 domain users       1850 jan  8 15:53 SRR1652476.Log.fi
nal.out
-rw-r--r-- 1 r0308983 domain users      24230 jan  8 15:53 SRR1652476.Log.ou
t
-rw-r--r-- 1 r0308983 domain users        364 jan  8 15:53 SRR1652476.Log.pr
ogress.out
-rw-r--r-- 1 r0308983 domain users     500063 jan  8 15:53 SRR1652476.SJ.ou
t.tab
-rw-r--r-- 1 r0308983 domain users  716385870 jan  8 16:21 SRR1652476.sorte
d.bam
-rw-r--r-- 1 r0308983 domain users    2615360 jan  8 16:24 SRR1652476.sorte
d.bam.bai
lrwxrwxrwx 1 r0308983 domain users         82 jan 25 13:53 SRR1652477_1.fast
q -> /mnt/storage/r0308983/jupyternotebooks/examwork/~/data/ncbi/sra/SRR1652
477_1.fastq
-rw-r--r-- 1 r0308983 domain users 1110544626 jan 10 22:18 SRR1652477.bam
-rw-r--r-- 1 r0308983 domain users       1853 jan 10 19:41 SRR1652477.Log.fi
nal.out
-rw-r--r-- 1 r0308983 domain users      24230 jan 10 19:41 SRR1652477.Log.ou
t
-rw-r--r-- 1 r0308983 domain users        364 jan 10 19:41 SRR1652477.Log.pr
ogress.out
-rw-r--r-- 1 r0308983 domain users     967739 jan 10 19:41 SRR1652477.SJ.ou
t.tab
-rw-r--r-- 1 r0308983 domain users 1190560025 jan 10 22:24 SRR1652477.sorte
d.bam
-rw-r--r-- 1 r0308983 domain users    2352800 jan 10 22:25 SRR1652477.sorte
d.bam.bai
```

Mapping to the genome

Next, we will align the reads to the reference genome. We will again use STAR here to align to the reference genome we made earlier in the RNA sequencing steps.

Mapping reads to the genome

In [9]:

```
pwd
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq
```

We will use STAR, which we made ourselves. genomeDir is in the folder:
/mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq

In [10]:

```
STAR --genomeDir /mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq \
     --runThreadN 32 \
     --readFilesIn SRR1652476_1.fastq \
     --outFileNamePrefix SRR1652476.
```

```
Jan 28 14:38:13 ..... started STAR run
Jan 28 14:38:13 ..... loading genome
Jan 28 14:42:47 ..... started mapping
Jan 28 14:44:31 ..... finished successfully
```

In [11]:

```
STAR --genomeDir /mnt/storage/r0308983/jupyternotebooks/examwork/RNASeq \
     --runThreadN 32 \
     --readFilesIn SRR1652477_1.fastq \
     --outFileNamePrefix SRR1652477.
```

```
Jan 28 14:44:51 ..... started STAR run
Jan 28 14:44:51 ..... loading genome
Jan 28 14:48:30 ..... started mapping
Jan 28 14:50:23 ..... finished successfully
```

In [12]:

```
ls -l SRR1652476.*
ls -l SRR1652477.*
```

```
-rw-r--r-- 1 r0308983 domain users 3012396036 jan 28 14:44 SRR1652476.Aligne
d.out.sam
-rw-r--r-- 1 r0308983 domain users  707924612 jan  8 16:17 SRR1652476.bam
-rw-r--r-- 1 r0308983 domain users       1849 jan 28 14:44 SRR1652476.Log.fi
nal.out
-rw-r--r-- 1 r0308983 domain users      24086 jan 28 14:44 SRR1652476.Log.ou
t
-rw-r--r-- 1 r0308983 domain users        364 jan 28 14:44 SRR1652476.Log.pr
ogress.out
-rw-r--r-- 1 r0308983 domain users     451780 jan 28 14:44 SRR1652476.SJ.ou
t.tab
-rw-r--r-- 1 r0308983 domain users  716385870 jan  8 16:21 SRR1652476.sorte
d.bam
-rw-r--r-- 1 r0308983 domain users    2615360 jan  8 16:24 SRR1652476.sorte
d.bam.bai
-rw-r--r-- 1 r0308983 domain users 5480560644 jan 28 14:50 SRR1652477.Aligne
d.out.sam
-rw-r--r-- 1 r0308983 domain users 1110544626 jan 10 22:18 SRR1652477.bam
-rw-r--r-- 1 r0308983 domain users       1852 jan 28 14:50 SRR1652477.Log.fi
nal.out
-rw-r--r-- 1 r0308983 domain users      24085 jan 28 14:50 SRR1652477.Log.ou
t
-rw-r--r-- 1 r0308983 domain users        364 jan 28 14:50 SRR1652477.Log.pr
ogress.out
-rw-r--r-- 1 r0308983 domain users     888942 jan 28 14:50 SRR1652477.SJ.ou
t.tab
-rw-r--r-- 1 r0308983 domain users 1190560025 jan 10 22:24 SRR1652477.sorte
d.bam
-rw-r--r-- 1 r0308983 domain users    2352800 jan 10 22:25 SRR1652477.sorte
d.bam.bai
```

In [13]:

```
ls SRR1652476.*
ls SRR1652477.*
```

```
SRR1652476.Aligned.out.sam  SRR1652476.Log.progress.out
SRR1652476.bam              SRR1652476.SJ.out.tab
SRR1652476.Log.final.out    SRR1652476.sorted.bam
SRR1652476.Log.out          SRR1652476.sorted.bam.bai
SRR1652477.Aligned.out.sam  SRR1652477.Log.progress.out
SRR1652477.bam              SRR1652477.SJ.out.tab
SRR1652477.Log.final.out    SRR1652477.sorted.bam
SRR1652477.Log.out          SRR1652477.sorted.bam.bai
```

SAMfile converting to BAMfile (compressed smaller file), followed by indexing to BAIfile for quick access

Next, we converted the sam files into bam files using samtools.

In [14]:

```
samtools view -S -b SRR1652476.Aligned.out.sam > SRR1652476.bam
samtools view -S -b SRR1652477.Aligned.out.sam > SRR1652477.bam
```

In [15]:

```
samtools sort -O bam -o SRR1652476.sorted.bam SRR1652476.bam
samtools sort -O bam -o SRR1652477.sorted.bam SRR1652477.bam
```

```
[bam_sort_core] merging from 5 files...
[bam_sort_core] merging from 9 files...
```

We also generated the index files for each bam file.

In [16]:

```
samtools index SRR1652476.sorted.bam
samtools index SRR1652477.sorted.bam
```

Peak calling

We will use macs2 (Model-based Analysis of ChIP-Seq) for peak calling. This is used for identifying transcript factor binding sites. MACS captures the influence of genome complexity to evaluate the significance of enriched ChIP regions, and MACS improves the spatial resolution of binding sites through combining the information of both sequencing tag position and orientation. MACS can be easily used for ChIP-Seq data alone, or with control sample with the increase of specificity.

In [17]:

```
pwd
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq
```

In [103]:

```
mkdir macs2_Prep1
```

In [18]:

```
ls -l
```

```
total 11938304
drwxr-xr-x 2 r0308983 domain users       4096 jan 24 11:44 macs2_Prep1
drwxr-xr-x 2 r0308983 domain users       4096 jan 10 16:31 macs2test
drwxr-xr-x 2 r0308983 domain users       4096 jan 10 19:20 quality control
lrwxrwxrwx 1 r0308983 domain users         82 jan 25 13:53 SRR1652476_1.fast
q -> /mnt/storage/r0308983/jupyternotebooks/examwork/~/data/ncbi/sra/SRR1652
476_1.fastq
-rw-r--r-- 1 r0308983 domain users 3012396036 jan 28 14:44 SRR1652476.Aligne
d.out.sam
-rw-r--r-- 1 r0308983 domain users  707927788 jan 28 14:53 SRR1652476.bam
-rw-r--r-- 1 r0308983 domain users       1849 jan 28 14:44 SRR1652476.Log.fi
nal.out
-rw-r--r-- 1 r0308983 domain users      24086 jan 28 14:44 SRR1652476.Log.ou
t
-rw-r--r-- 1 r0308983 domain users        364 jan 28 14:44 SRR1652476.Log.pr
ogress.out
-rw-r--r-- 1 r0308983 domain users     451780 jan 28 14:44 SRR1652476.SJ.ou
t.tab
-rw-r--r-- 1 r0308983 domain users  716366359 jan 28 14:59 SRR1652476.sorte
d.bam
-rw-r--r-- 1 r0308983 domain users    2615232 jan 28 15:09 SRR1652476.sorte
d.bam.bai
lrwxrwxrwx 1 r0308983 domain users         82 jan 25 13:53 SRR1652477_1.fast
q -> /mnt/storage/r0308983/jupyternotebooks/examwork/~/data/ncbi/sra/SRR1652
477_1.fastq
-rw-r--r-- 1 r0308983 domain users 5480560644 jan 28 14:50 SRR1652477.Aligne
d.out.sam
-rw-r--r-- 1 r0308983 domain users 1110551449 jan 28 14:56 SRR1652477.bam
-rw-r--r-- 1 r0308983 domain users       1852 jan 28 14:50 SRR1652477.Log.fi
nal.out
-rw-r--r-- 1 r0308983 domain users      24085 jan 28 14:50 SRR1652477.Log.ou
t
-rw-r--r-- 1 r0308983 domain users        364 jan 28 14:50 SRR1652477.Log.pr
ogress.out
-rw-r--r-- 1 r0308983 domain users     888942 jan 28 14:50 SRR1652477.SJ.ou
t.tab
-rw-r--r-- 1 r0308983 domain users 1190579589 jan 28 15:05 SRR1652477.sorte
d.bam
-rw-r--r-- 1 r0308983 domain users    2353624 jan 28 15:09 SRR1652477.sorte
d.bam.bai
```

In [19]:

```
macs2 callpeak
```

```
usage: macs2 callpeak [-h] -t TFILE [TFILE ...] [-c [CFILE [CFILE ...]]]
                      [-f {AUTO,BAM,SAM,BED,ELAND,ELANDMULTI,ELANDEXPORT,BOW
TIE,BAMPE}]
                      [-g GSIZE] [--keep-dup KEEPDUPLICATES]
                      [--buffer-size BUFFER_SIZE] [--outdir OUTDIR] [-n NAM
E]
                      [-B] [--verbose VERBOSE] [--trackline] [--SPMR]
                      [-s TSIZE] [--bw BW] [-m MFOLD MFOLD] [--fix-bimodal]
                      [--nomodel] [--shift SHIFT] [--extsize EXTSIZE]
                      [-q QVALUE | -p PVALUE] [--to-large] [--ratio RATIO]
                      [--down-sample] [--seed SEED] [--nolambda]
                      [--slocal SMALLLOCAL] [--llocal LARGELOCAL] [--broad]
                      [--broad-cutoff BROADCUTOFF] [--cutoff-analysis]
                      [--call-summits] [--fe-cutoff FECUTOFF]
macs2 callpeak: error: argument -t/--treatment is required
```

In [20]:

```
cd /mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq/
pwd
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq
```

In [21]:

```
macs2 callpeak -t SRR1652476.sorted.bam -c SRR1652477.sorted.bam --outdir macs2_Prep1 -f BA
```

```
INFO  @ Sat, 28 Jan 2017 15:10:04:
# Command line: callpeak -t SRR1652476.sorted.bam -c SRR1652477.sorted.bam -
-outdir macs2_Prep1 -f BAM -g mm -n Prep1 -p 0.000001
# ARGUMENTS LIST:
# name = Prep1
# format = BAM
# ChIP-seq file = ['SRR1652476.sorted.bam']
# control file = ['SRR1652477.sorted.bam']
# effective genome size = 1.87e+09
# band width = 300
# model fold = [5, 50]
# pvalue cutoff = 1.00e-06
# qvalue will not be calculated and reported as -1 in the final output.
# Larger dataset will be scaled towards smaller dataset.
# Range for calculating regional lambda is: 1000 bps and 10000 bps
# Broad region calling is off

INFO  @ Sat, 28 Jan 2017 15:10:04: #1 read tag files...
INFO  @ Sat, 28 Jan 2017 15:10:04: #1 read treatment tags...
INFO  @ Sat, 28 Jan 2017 15:10:08:  1000000
INFO  @ Sat, 28 Jan 2017 15:10:12:  2000000
INFO  @ Sat, 28 Jan 2017 15:10:15:  3000000
INFO  @ Sat, 28 Jan 2017 15:10:19:  4000000
INFO  @ Sat, 28 Jan 2017 15:10:23:  5000000
INFO  @ Sat, 28 Jan 2017 15:10:26:  6000000
INFO  @ Sat, 28 Jan 2017 15:10:30:  7000000
INFO  @ Sat, 28 Jan 2017 15:10:34:  8000000
INFO  @ Sat, 28 Jan 2017 15:10:37:  9000000
INFO  @ Sat, 28 Jan 2017 15:10:41:  10000000
INFO  @ Sat, 28 Jan 2017 15:10:45:  11000000
INFO  @ Sat, 28 Jan 2017 15:10:48:  12000000
INFO  @ Sat, 28 Jan 2017 15:10:52:  13000000
INFO  @ Sat, 28 Jan 2017 15:10:56:  14000000
INFO  @ Sat, 28 Jan 2017 15:10:59:  15000000
INFO  @ Sat, 28 Jan 2017 15:11:03:  16000000
INFO  @ Sat, 28 Jan 2017 15:11:07:  17000000
INFO  @ Sat, 28 Jan 2017 15:11:09:  18000000
INFO  @ Sat, 28 Jan 2017 15:11:11:  19000000
INFO  @ Sat, 28 Jan 2017 15:11:13: #1.2 read input tags...
INFO  @ Sat, 28 Jan 2017 15:11:17:  1000000
INFO  @ Sat, 28 Jan 2017 15:11:21:  2000000
INFO  @ Sat, 28 Jan 2017 15:11:24:  3000000
INFO  @ Sat, 28 Jan 2017 15:11:26:  4000000
INFO  @ Sat, 28 Jan 2017 15:11:29:  5000000
INFO  @ Sat, 28 Jan 2017 15:11:33:  6000000
INFO  @ Sat, 28 Jan 2017 15:11:37:  7000000
INFO  @ Sat, 28 Jan 2017 15:11:40:  8000000
INFO  @ Sat, 28 Jan 2017 15:11:44:  9000000
INFO  @ Sat, 28 Jan 2017 15:11:48:  10000000
INFO  @ Sat, 28 Jan 2017 15:11:51:  11000000
INFO  @ Sat, 28 Jan 2017 15:11:55:  12000000
INFO  @ Sat, 28 Jan 2017 15:11:58:  13000000
INFO  @ Sat, 28 Jan 2017 15:12:01:  14000000
INFO  @ Sat, 28 Jan 2017 15:12:04:  15000000
INFO  @ Sat, 28 Jan 2017 15:12:07:  16000000
INFO  @ Sat, 28 Jan 2017 15:12:11:  17000000
INFO  @ Sat, 28 Jan 2017 15:12:15:  18000000
INFO  @ Sat, 28 Jan 2017 15:12:19:  19000000
INFO  @ Sat, 28 Jan 2017 15:12:22:  20000000
INFO  @ Sat, 28 Jan 2017 15:12:25:  21000000
```

```
INFO  @ Sat, 28 Jan 2017 15:12:29:  22000000
INFO  @ Sat, 28 Jan 2017 15:12:33:  23000000
INFO  @ Sat, 28 Jan 2017 15:12:36:  24000000
INFO  @ Sat, 28 Jan 2017 15:12:40:  25000000
INFO  @ Sat, 28 Jan 2017 15:12:43:  26000000
INFO  @ Sat, 28 Jan 2017 15:12:46:  27000000
INFO  @ Sat, 28 Jan 2017 15:12:48:  28000000
INFO  @ Sat, 28 Jan 2017 15:12:51:  29000000
INFO  @ Sat, 28 Jan 2017 15:12:53:  30000000
INFO  @ Sat, 28 Jan 2017 15:12:55:  31000000
INFO  @ Sat, 28 Jan 2017 15:12:58:  32000000
INFO  @ Sat, 28 Jan 2017 15:13:00:  33000000
INFO  @ Sat, 28 Jan 2017 15:13:02:  34000000
INFO  @ Sat, 28 Jan 2017 15:13:04:  35000000
INFO  @ Sat, 28 Jan 2017 15:13:05: #1 tag size is determined as 36 bps
INFO  @ Sat, 28 Jan 2017 15:13:05: #1 tag size = 36
INFO  @ Sat, 28 Jan 2017 15:13:05: #1  total tags in treatment: 13693923
INFO  @ Sat, 28 Jan 2017 15:13:05: #1 user defined the maximum tags...
INFO  @ Sat, 28 Jan 2017 15:13:05: #1 filter out redundant tags at the same
 location and the same strand by allowing at most 1 tag(s)
INFO  @ Sat, 28 Jan 2017 15:13:06: #1  tags after filtering in treatment: 12
247342
INFO  @ Sat, 28 Jan 2017 15:13:06: #1  Redundant rate of treatment: 0.11
INFO  @ Sat, 28 Jan 2017 15:13:06: #1  total tags in control: 19174658
INFO  @ Sat, 28 Jan 2017 15:13:06: #1 user defined the maximum tags...
INFO  @ Sat, 28 Jan 2017 15:13:06: #1 filter out redundant tags at the same
 location and the same strand by allowing at most 1 tag(s)
INFO  @ Sat, 28 Jan 2017 15:13:06: #1  tags after filtering in control: 1493
5458
INFO  @ Sat, 28 Jan 2017 15:13:06: #1  Redundant rate of control: 0.22
INFO  @ Sat, 28 Jan 2017 15:13:06: #1 finished!
INFO  @ Sat, 28 Jan 2017 15:13:06: #2 Build Peak Model...
INFO  @ Sat, 28 Jan 2017 15:13:06: #2 looking for paired plus/minus strand p
eaks...
INFO  @ Sat, 28 Jan 2017 15:13:07: #2 number of paired peaks: 3260
INFO  @ Sat, 28 Jan 2017 15:13:07: start model_add_line...
INFO  @ Sat, 28 Jan 2017 15:13:08: start X-correlation...
INFO  @ Sat, 28 Jan 2017 15:13:08: end of X-cor
INFO  @ Sat, 28 Jan 2017 15:13:08: #2 finished!
INFO  @ Sat, 28 Jan 2017 15:13:08: #2 predicted fragment length is 131 bps
INFO  @ Sat, 28 Jan 2017 15:13:08: #2 alternative fragment length(s) may be
 131 bps
INFO  @ Sat, 28 Jan 2017 15:13:08: #2.2 Generate R script for model : macs2_
Prep1/Prep1_model.r
INFO  @ Sat, 28 Jan 2017 15:13:08: #3 Call peaks...
INFO  @ Sat, 28 Jan 2017 15:13:08: #3 Call peaks with given -log10pvalue cut
off: 6.00000 ...
INFO  @ Sat, 28 Jan 2017 15:13:08: #3 Pre-compute pvalue-qvalue table...
INFO  @ Sat, 28 Jan 2017 15:14:13: #3 Call peaks for each chromosome...
INFO  @ Sat, 28 Jan 2017 15:14:40: #4 Write output xls file... macs2_Prep1/P
rep1_peaks.xls
INFO  @ Sat, 28 Jan 2017 15:14:40: #4 Write peak in narrowPeak format fil
e... macs2_Prep1/Prep1_peaks.narrowPeak
INFO  @ Sat, 28 Jan 2017 15:14:40: #4 Write summits bed file... macs2_Prep1/
Prep1_summits.bed
INFO  @ Sat, 28 Jan 2017 15:14:40: Done!
```

In [22]:

```
cd /mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq
pwd
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq
```

In [23]:

```
head macs2_Prep1/Prep1_peaks.narrowPeak
```

```
NT_039252.1     249882  250148  Prep1_peak_1    347     .       16.14667
        34.77995        30.91858        137
NT_039382.4     169862  170003  Prep1_peak_2    89      .       4.91322 8.91
013     5.42821 72
NT_039382.4     527907  528337  Prep1_peak_3    214     .       11.30267
        21.46081        17.73807        311
NT_039634.4     452559  452910  Prep1_peak_4    1008    .       31.71304
        100.85107       96.60367        164
NT_039634.4     606805  606983  Prep1_peak_5    200     .       10.76444
        20.07122        16.36566        90
NT_039662.3     8820    9576    Prep1_peak_6    1922    .       60.81910
        192.23557       187.52682       324
NT_039662.3     59113   59511   Prep1_peak_7    1404    .       47.36355
        140.49121       136.04565       123
NT_039662.3     92072   92263   Prep1_peak_8    134     .       8.07333 13.4
6318    9.86621 72
NT_039662.3     229485  229634  Prep1_peak_9    122     .       7.53511 12.2
1894    8.64656 35
NT_039662.3     854305  854854  Prep1_peak_10   1670    .       54.36044
        167.06264       162.48657       296
```

Next, we checked the amount of peaks that have been called in the sample.

In [24]:

```
wc -l macs2_Prep1/Prep1_peaks.narrowPeak
```

```
2977 macs2_Prep1/Prep1_peaks.narrowPeak
```

In the article they use a threshold pvalue of <10^-6 and an FDR (false discovery rate) <20% in the immunoprecipitation v. input comparison, and they identified 3902 peaks bound by Prep1. We have 2977 peaks bound by Prep1.

In [25]:

```
cd /mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq/macs2_Prep1
pwd
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq/macs2_Prep1
```

In [26]:

```
ls
```

```
Prep1_model.r         Prep1_peaksgrep3.bed    Prep1_peaks.xls
Prep1_peaks.bed       Prep1_peaksgrep.bed     Prep1_summits.bed
Prep1_peaksgrep2.bed  Prep1_peaks.narrowPeak
```

In [27]:

```
cat Prep1_summits.bed | wc -l
```

```
2977
```

In [28]:

```
cd /mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq
pwd
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq
```

In [29]:

```
cd /mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq/macs2_Prep1
```

In [30]:

```
ls
```

```
Prep1_model.r         Prep1_peaksgrep3.bed    Prep1_peaks.xls
Prep1_peaks.bed       Prep1_peaksgrep.bed     Prep1_summits.bed
Prep1_peaksgrep2.bed  Prep1_peaks.narrowPeak
```

In [31]:

```
head Prep1_peaks.narrowPeak
```

```
NT_039252.1     249882  250148  Prep1_peak_1    347     .       16.14667
        34.77995        30.91858        137
NT_039382.4     169862  170003  Prep1_peak_2    89      .       4.91322 8.91
013     5.42821 72
NT_039382.4     527907  528337  Prep1_peak_3    214     .       11.30267
        21.46081        17.73807        311
NT_039634.4     452559  452910  Prep1_peak_4    1008    .       31.71304
        100.85107       96.60367        164
NT_039634.4     606805  606983  Prep1_peak_5    200     .       10.76444
        20.07122        16.36566        90
NT_039662.3     8820    9576    Prep1_peak_6    1922    .       60.81910
        192.23557       187.52682       324
NT_039662.3     59113   59511   Prep1_peak_7    1404    .       47.36355
        140.49121       136.04565       123
NT_039662.3     92072   92263   Prep1_peak_8    134     .       8.07333 13.4
6318    9.86621 72
NT_039662.3     229485  229634  Prep1_peak_9    122     .       7.53511 12.2
1894    8.64656 35
NT_039662.3     854305  854854  Prep1_peak_10   1670    .       54.36044
        167.06264       162.48657       296
```

In [32]:

```
tail Prep1_peaks.narrowPeak
```

```
chrX    169997606       169998644       Prep1_peak_2968 972     .       8.31
704     97.24997        93.02292        615
chrX    169998722       169998855       Prep1_peak_2969 123     .       4.23
086     12.32322        8.75037 86
chrX    170001398       170001569       Prep1_peak_2970 375     .       5.42
812     37.52510        33.64460        87
chrX    170009029       170009607       Prep1_peak_2971 1629    .       14.5
1812    162.92537       158.37044       256
chrX    170009797       170010248       Prep1_peak_2972 741     .       8.24
856     74.18668        70.07689        191
chrX    170674240       170674498       Prep1_peak_2973 166     .       7.84
019     16.62945        12.97658        53
chrY    90765138        90765658        Prep1_peak_2974 1259    .       15.5
4807    125.95316       121.57939       403
chrY    90765699        90765964        Prep1_peak_2975 646     .       10.0
2095    64.67738        60.61895        121
chrY    90771966        90772955        Prep1_peak_2976 1122    .       7.74
580     112.28421       107.97959       367
chrY    90775838        90776016        Prep1_peak_2977 761     .       8.92
333     76.16503        72.04437        76
```

Converting peaks.narrowPeak file to a .bed file

We then converted the peaks.narrowPeak file into a bed file. We need to conserve the first four columns. The first column is the name of the chromosome, the second column is the start of the peak, third column is the end position of the peak and the fourth column is the name of the peak.

In [33]:

```
cat Prep1_peaks.narrowPeak | cut -f 1-4 > Prep1_peaks.bed
```

In [34]:

```
ls
```

```
Prep1_model.r         Prep1_peaksgrep3.bed    Prep1_peaks.xls
Prep1_peaks.bed       Prep1_peaksgrep.bed     Prep1_summits.bed
Prep1_peaksgrep2.bed  Prep1_peaks.narrowPeak
```

In [35]:

```
head Prep1_peaks.bed
```

```
NT_039252.1     249882  250148  Prep1_peak_1
NT_039382.4     169862  170003  Prep1_peak_2
NT_039382.4     527907  528337  Prep1_peak_3
NT_039634.4     452559  452910  Prep1_peak_4
NT_039634.4     606805  606983  Prep1_peak_5
NT_039662.3     8820    9576    Prep1_peak_6
NT_039662.3     59113   59511   Prep1_peak_7
NT_039662.3     92072   92263   Prep1_peak_8
NT_039662.3     229485  229634  Prep1_peak_9
NT_039662.3     854305  854854  Prep1_peak_10
```

In [36]:

```
tail Prep1_peaks.bed
```

```
chrX    169997606       169998644       Prep1_peak_2968
chrX    169998722       169998855       Prep1_peak_2969
chrX    170001398       170001569       Prep1_peak_2970
chrX    170009029       170009607       Prep1_peak_2971
chrX    170009797       170010248       Prep1_peak_2972
chrX    170674240       170674498       Prep1_peak_2973
chrY    90765138        90765658        Prep1_peak_2974
chrY    90765699        90765964        Prep1_peak_2975
chrY    90771966        90772955        Prep1_peak_2976
chrY    90775838        90776016        Prep1_peak_2977
```

We now have a bed file that contains only the first four columns.

We will use this bed file to look at our peaks in IGV. But we will first perform a GREAT analysis of the peaks.

GREAT ANALYIS

GREAT allows to determine biological processes that are connected to the peaks determined in ChIP sequencing. We first have to remove the NT and NW lines of our bed files because GREAT does not recognize this and will not process the files.

In [37]:

```
pwd
ls
```

```
/mnt/storage/r0308983/jupyternotebooks/examwork/ChIPSeq/macs2_Prep1
Prep1_model.r         Prep1_peaksgrep3.bed    Prep1_peaks.xls
Prep1_peaks.bed       Prep1_peaksgrep.bed     Prep1_summits.bed
Prep1_peaksgrep2.bed  Prep1_peaks.narrowPeak
```

In [38]:

```
head Prep1_peaks.bed
```

```
NT_039252.1     249882  250148  Prep1_peak_1
NT_039382.4     169862  170003  Prep1_peak_2
NT_039382.4     527907  528337  Prep1_peak_3
NT_039634.4     452559  452910  Prep1_peak_4
NT_039634.4     606805  606983  Prep1_peak_5
NT_039662.3     8820    9576    Prep1_peak_6
NT_039662.3     59113   59511   Prep1_peak_7
NT_039662.3     92072   92263   Prep1_peak_8
NT_039662.3     229485  229634  Prep1_peak_9
NT_039662.3     854305  854854  Prep1_peak_10
```

In [39]:

```
grep -v "NT" Prep1_peaks.bed | grep -v "NW" > Prep1_peaksgrep.bed
```

In [40]:

```
head Prep1_peaksgrep.bed
```

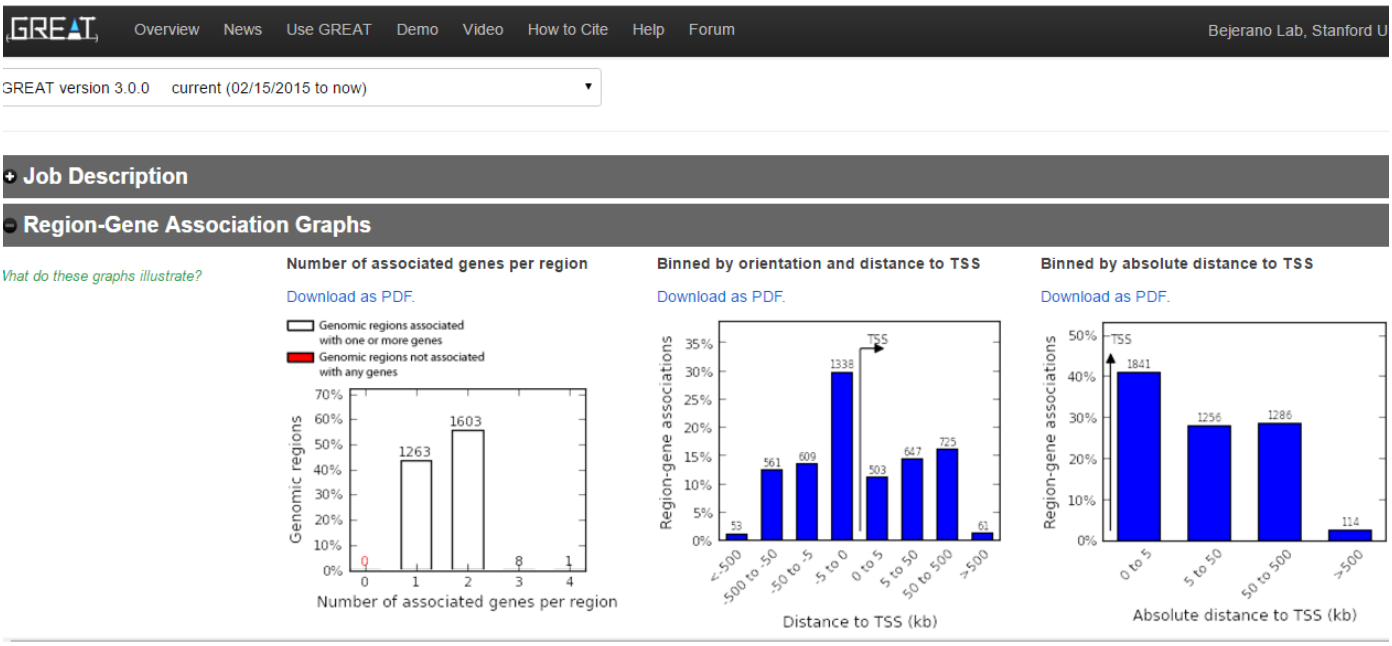
```
chr1    7088722 7089156 Prep1_peak_103
chr1    7206786 7206986 Prep1_peak_104
chr1    7397633 7398430 Prep1_peak_105
chr1    7777129 7777523 Prep1_peak_106
chr1    9967799 9968102 Prep1_peak_107
chr1    13372256        13372952        Prep1_peak_108
chr1    13660543        13660726        Prep1_peak_109
chr1    15805424        15805664        Prep1_peak_110
chr1    17013998        17014223        Prep1_peak_111
chr1    24245819        24245989        Prep1_peak_112
```

In [41]:

```
tail Prep1_peaksgrep.bed
```

```
chrX    169997606       169998644       Prep1_peak_2968
chrX    169998722       169998855       Prep1_peak_2969
chrX    170001398       170001569       Prep1_peak_2970
chrX    170009029       170009607       Prep1_peak_2971
chrX    170009797       170010248       Prep1_peak_2972
chrX    170674240       170674498       Prep1_peak_2973
chrY    90765138        90765658        Prep1_peak_2974
chrY    90765699        90765964        Prep1_peak_2975
chrY    90771966        90772955        Prep1_peak_2976
chrY    90775838        90776016        Prep1_peak_2977
```

As you can see, we now have the bed file that contain lines that GREAT will be able to read. We proceeded to upload the bed file to see which processes were linked to the peaks.



The "Number of associated genes per region" graph shows how many genes each genomic region is assigned as putatively regulating based on the association rule used. In the first figure we can see that 1263 genomic regions are associated with one gene. 1603 genomic regions were associated with 2 genes, 8 regions with 3 genes and 1 region with 4 genes. There were no regions not associated with any genes. The distance to TSS (Transcriptional Start Sites) is shown in the second figure in Kb. This graph shows the distance between input regions and their putatively regulated genes. Two graphs are displayed: one in which