Name: Hien Le

Student ID: [redacted]

Project work report

The project work focuses on ChIP-seq analysis, a method used to analyze proteins interaction with DNA. Based on analysis peak shifts, the results will show annotated peaks, binding motif discovery, and enriched pathways of cell signaling involved or stimulated under specific conditions e.g cancer stage, drug treatments, UV.

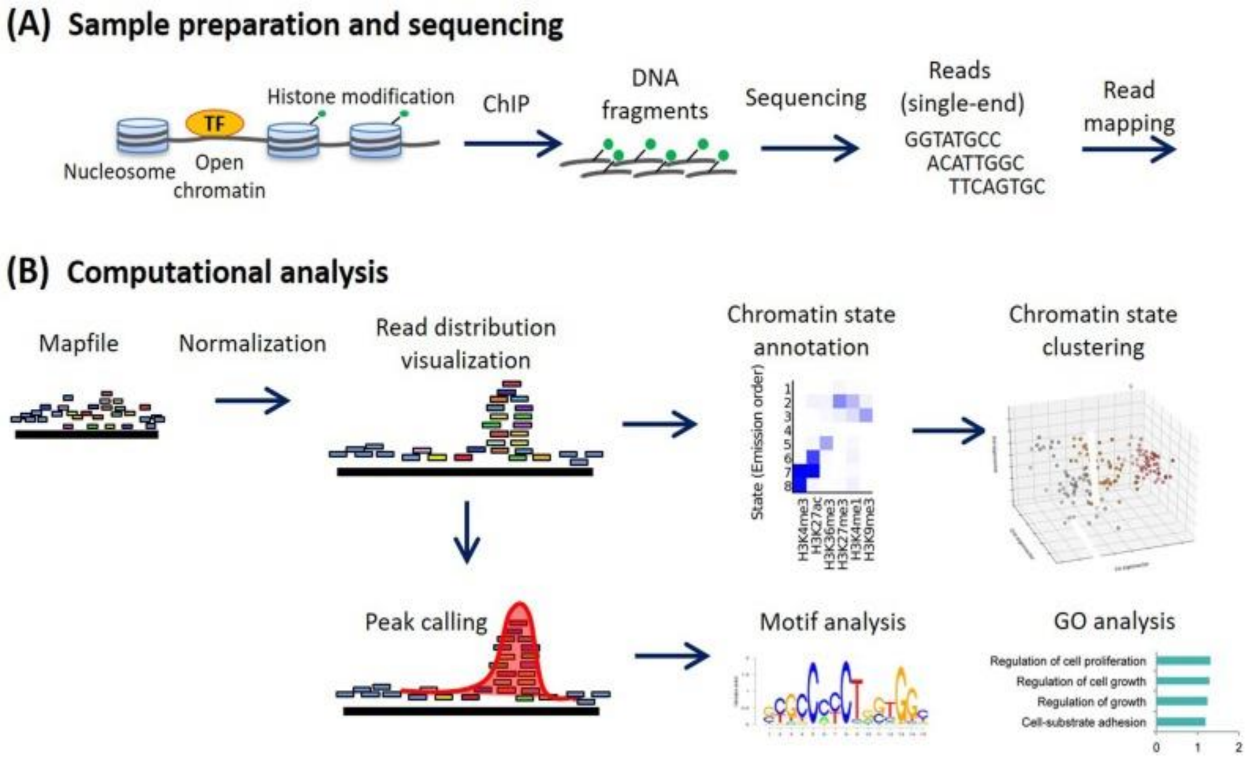


Fig.1: ChIP-seq analysis workflow (A) Sample preparation and sequencing. (B) Computational analysis in a canonical ChIP-seq analysis. Various analyses are implemented using normalized read distribution [1].

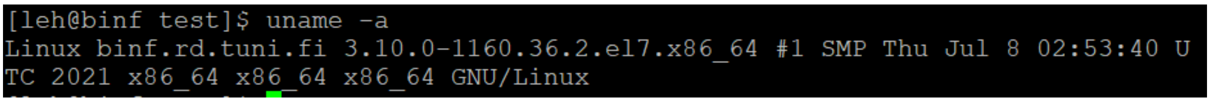
Analysis 1:

The first requirement is to analysis ATAC-seq results of two prostate cancer cell lines: EP156T and VCaP. The output of ATAC-seq experiments is stored as .BAM file in the directory /student_data/BBT.BI.201_2021/data/capstone_project/.

To call the peaks, it is worth to use a tool called MACS2. Running MACS program to identify transcription factor binding sites of sample. As a non-interactive command line tool, MACS takes input by setting proper command line parameters and no input is needed during running. The input should be mapped reads from ChIP-Seq experiments, and several widely used formats are accepted, while the control data is optional. The minimum output of MACS contains the called peaks and their summits, together with R scripts used to draw the shifting size model build by MACS. [2]

Necessary resources list:

Hardware and software: A computer with proper versions of Linux as below.



```
[leh@binf test]$ uname -a
Linux binf.rd.tuni.fi 3.10.0-1160.36.2.el7.x86_64 #1 SMP Thu Jul 8 02:53:40 U
TC 2021 x86_64 x86_64 x86_64 GNU/Linux
```

Files: To identify transcript factor binding sites, the required files must have the information of mapped genomic locations for sequencing reads in certain formats. Here, the data of EP156T and VCap is stored as .BAM and .BAM.BAI files.

Running environment: MACS version 2.2.6a2

Output directory: The result of MACS2 have some files. Thus, one directory is created to store the results of the macs2 callpeak command.

Made directory name: /student_data/BBT.BI.201_2021/students/leh/project_work

Setting directory before running commands:

Working directory is /student_data/BBT.BI.201_2021/data/capstone_project/

Execute MACS by the following command:

```
macs2 callpeak -f BAMPE -t <BAM files> --nomodel –broad -g hs -n <string names> --outdir
2> .log
```

In this case, the two commands are written following:

- EP156T cells:

```
macs2 callpeak -f BAMPE -t VCaP.bam -g hs -n VCaP --nomodel --broad --outdir
/student_data/BBT.BI.201_2021/students/leh/project_work 2>
/student_data/BBT.BI.201_2021/students/leh/project_work/test/VCaP.log
```

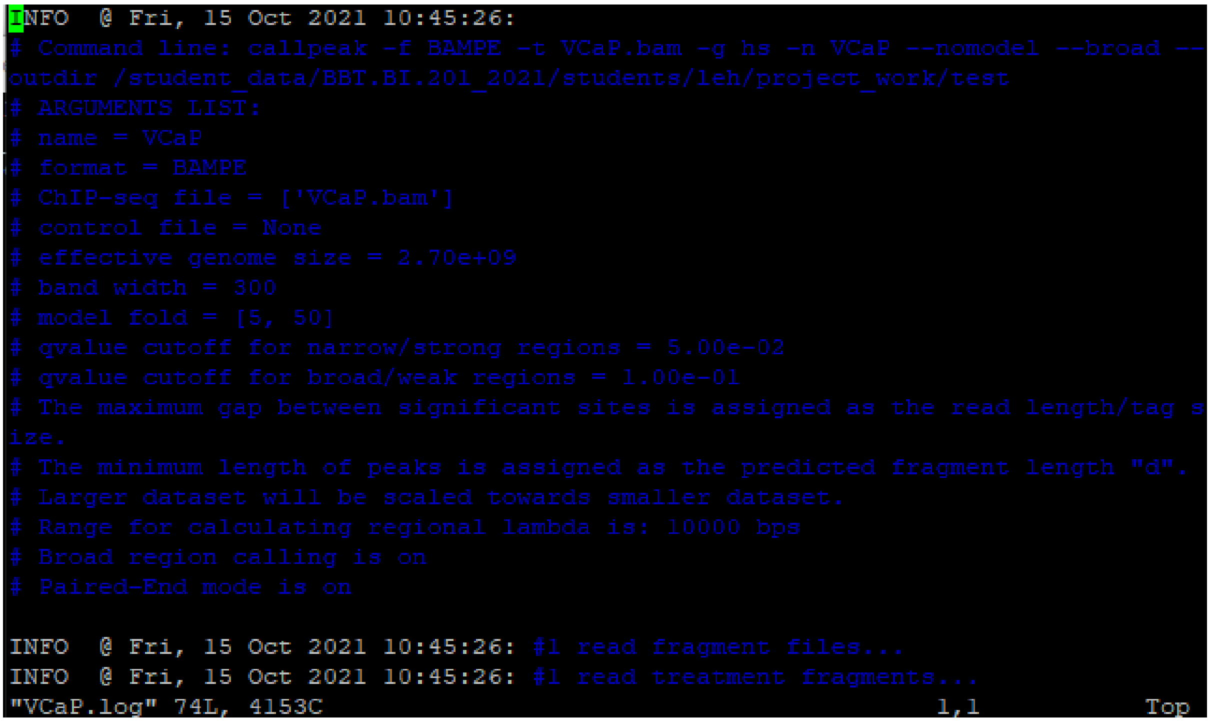
- VCaP cells:

```
macs2 callpeak -f BAMPE -t EP156T.bam -g hs -n EP156T --nomodel --broad --outdir
/student_data/BBT.BI.201_2021/students/leh/project_work/ 2>
/student_data/BBT.BI.201_2021/students/leh/project_work/test/EP156T.log
```

MACS progression:

Check the MACS critical and progress messages displayed in the terminal, shown as below.

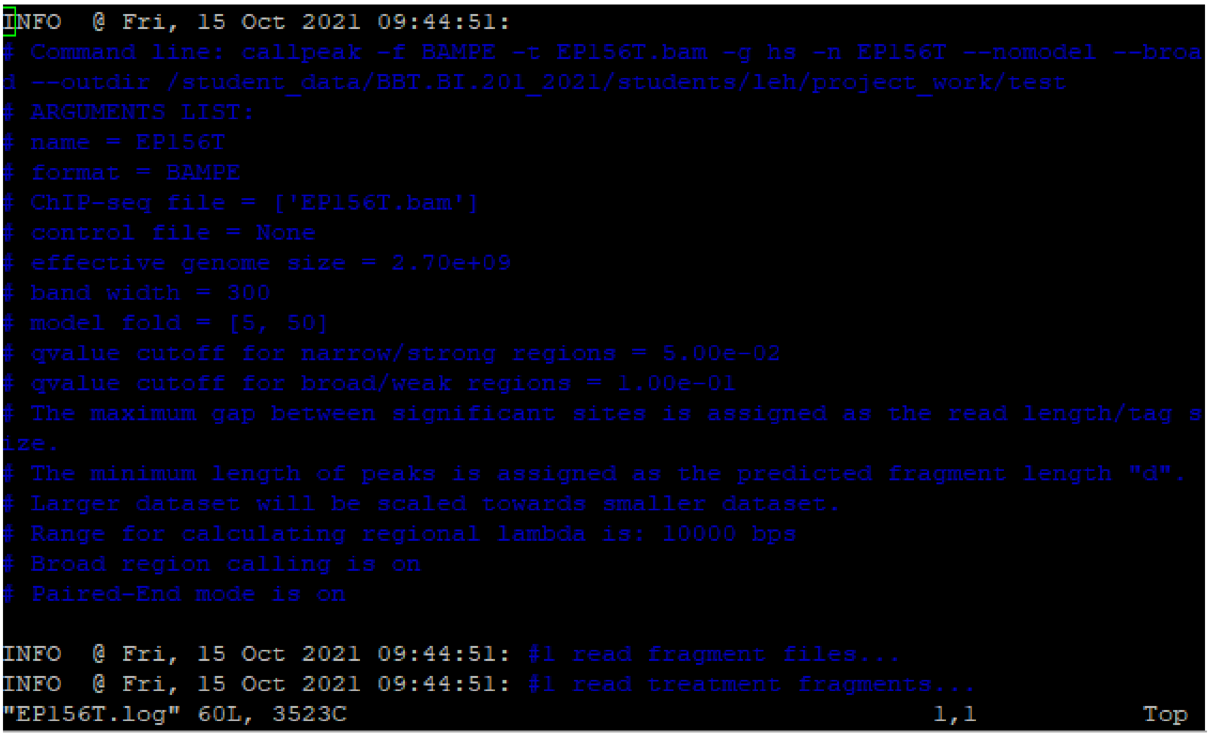
For example: in VCaP.bam file as input



```
INFO  @ Fri, 15 Oct 2021 10:45:26:
# Command line: callpeak -f BAMPE -t VCaP.bam -g hs -n VCaP --nomodel --broad --
outdir /student_data/BBT.BI.201_2021/students/leh/project_work/test
# ARGUMENTS LIST:
# name = VCaP
# format = BAMPE
# ChIP-seq file = ['VCaP.bam']
# control file = None
# effective genome size = 2.70e+09
# band width = 300
# model fold = [5, 50]
# qvalue cutoff for narrow/strong regions = 5.00e-02
# qvalue cutoff for broad/weak regions = 1.00e-01
# The maximum gap between significant sites is assigned as the read length/tag s
ize.
# The minimum length of peaks is assigned as the predicted fragment length "d".
# Larger dataset will be scaled towards smaller dataset.
# Range for calculating regional lambda is: 10000 bps
# Broad region calling is on
# Paired-End mode is on

INFO  @ Fri, 15 Oct 2021 10:45:26: #1 read fragment files...
INFO  @ Fri, 15 Oct 2021 10:45:26: #1 read treatment fragments...
"VCaP.log" 74L, 4153C                                          1,1           Top
```

EP156T file



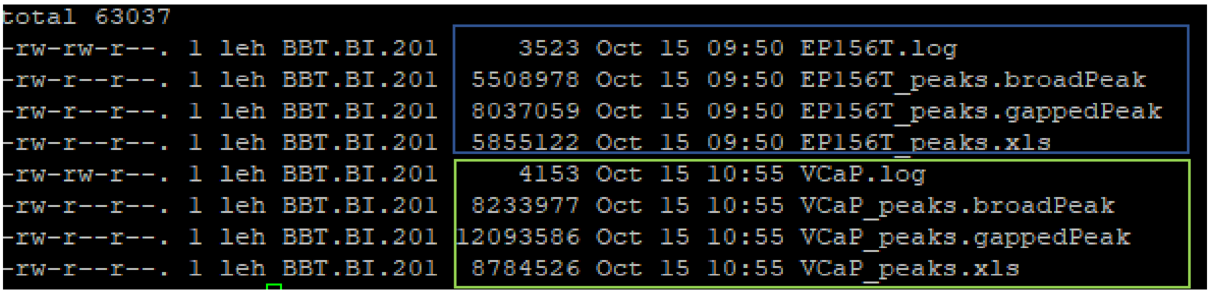
```
INFO  @ Fri, 15 Oct 2021 09:44:51:
# Command line: callpeak -f BAMPE -t EP156T.bam -g hs -n EP156T --nomodel --broa
d --outdir /student_data/BBT.BI.201_2021/students/leh/project_work/test
# ARGUMENTS LIST:
# name = EP156T
# format = BAMPE
# ChIP-seq file = ['EP156T.bam']
# control file = None
# effective genome size = 2.70e+09
# band width = 300
# model fold = [5, 50]
# qvalue cutoff for narrow/strong regions = 5.00e-02
# qvalue cutoff for broad/weak regions = 1.00e-01
# The maximum gap between significant sites is assigned as the read length/tag s
ize.
# The minimum length of peaks is assigned as the predicted fragment length "d".
# Larger dataset will be scaled towards smaller dataset.
# Range for calculating regional lambda is: 10000 bps
# Broad region calling is on
# Paired-End mode is on

INFO  @ Fri, 15 Oct 2021 09:44:51: #1 read fragment files...
INFO  @ Fri, 15 Oct 2021 09:44:51: #1 read treatment fragments...
"EP156T.log" 60L, 3523C                                        1,1           Top
```

Running time: it is taken 10 min for VP156T cells and 20 min for VCaP cells.

Check the ouput files generated by MACS:

Macs2 callpeak command generates 4 files for each cell line in the output directory. They consist log, broadPeak, gappedPeak, and xls files.



```
total 63037
-rw-rw-r--. 1 leh BBT.BI.201     3523 Oct 15 09:50 EP156T.log
-rw-r--r--. 1 leh BBT.BI.201  5508978 Oct 15 09:50 EP156T_peaks.broadPeak
-rw-r--r--. 1 leh BBT.BI.201  8037059 Oct 15 09:50 EP156T_peaks.gappedPeak
-rw-r--r--. 1 leh BBT.BI.201  5855122 Oct 15 09:50 EP156T_peaks.xls
-rw-rw-r--. 1 leh BBT.BI.201     4153 Oct 15 10:55 VCaP.log
-rw-r--r--. 1 leh BBT.BI.201  8233977 Oct 15 10:55 VCaP_peaks.broadPeak
-rw-r--r--. 1 leh BBT.BI.201 12093586 Oct 15 10:55 VCaP_peaks.gappedPeak
-rw-r--r--. 1 leh BBT.BI.201  8784526 Oct 15 10:55 VCaP_peaks.xls
```

Log format: present the progression of command.

BroadPeak format: use to provide called region of signal enrichment based on pooled, normalized interpreted data.

GappedPeak format: provide called regions of signal enrichment based on pooled, normalized data where the regions may be splice or incorporate gaps in the genomic sequence.

Xls format: contains full information of a called peak with a header.

Count the number of peaks called for each cell line in file. BroadPeak extension is executed to count peaks

The number of peaks in EP156T cells: 76362 peaks.

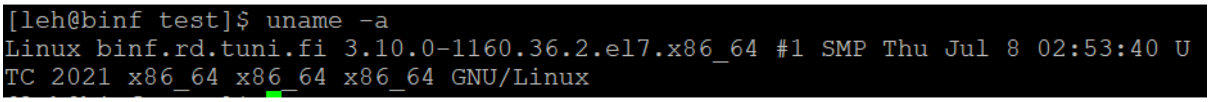
The number of peaks in VCaP cells: 116909 peaks.

Analysis 2:

Different data types have different peak shapes. Same transcription bindings may have different peak shapes reflecting differences in biological conditions. Replicates should have similar binding patterns. To call peaks, we use the common tool: macs2 callpeak and have 3 different formats as the above question. Broakpeaks are the best choice for histone modifications or histone modifiers since the regions can be much wider.

Necessary resources list:

Hardware and software: A computer with proper versions of Linux as below.



```
[leh@binf test]$ uname -a
Linux binf.rd.tuni.fi 3.10.0-1160.36.2.el7.x86_64 #1 SMP Thu Jul 8 02:53:40 U
TC 2021 x86_64 x86_64 x86_64 GNU/Linux
```

Files: To compare the peaks called for each of the cell lines and to check how similar the peaks are with each other, jaccard similarity coefficient – jaccard index was calculated to gauge the similarity and diversity of sample sets. Here, the data of EP156T and VCap is stored as .broad peak format.

File input: EP156T_peaks.broadPeak and VCap_peaks.broadPeaks

Running environment: installed bedtools jaccard (aka jaccard) version v2.29.2-1-g59bbfbdc.

Output directory: The result of bedtools jaccard include one file.

/student_data/BBT.BI.201_2021/students/leh/project_work

Setting directory before running commands:

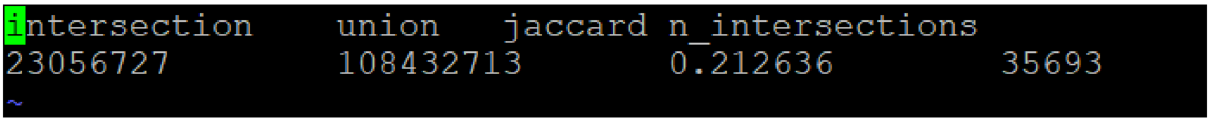
Working directory is student_data/BBT.BI.201_2021/students/leh/project_work

Execute bedtools jaccard by the following command:

bedtools jaccard -a <file 1> -b <file 2> > <output>

bedtools jaccard -a EP156T_peaks.broadPeak -b VCaP_peaks.broadPeak > EP156T_VCap_jaccard.tsv

Result:



```
intersection    union   jaccard n_intersections
23056727        108432713       0.212636        35693
~
```

By default, bedtools jaccard reports the length of intersection (23056727), the length of the union (108432713), the final Jaccard statistic reflecting the similarity of the two sets (0.212636), and the number of intersections (35693).

The Jaccard similarity index presents how similar the two set are:

- Two sets that share all members would be 100% similar, the closer to 100%, the more similarity e.g 80-90%.
- If they share no members, the index is closer to 0% similar.
- The midway point 50% means that the two sets share haft of the members.

In this case, the similar between EP156T normal prostate cells and VCaP prostate cancer cells is 21.2% similar of transcription factor binding.

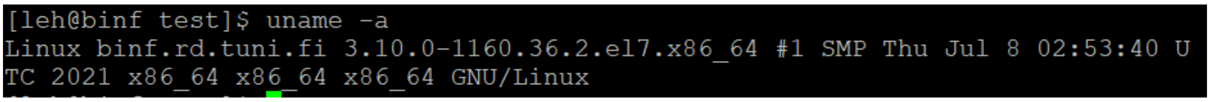
Analysis 3:

In here, the annotated peaks are found regarding their genomic location. This information will present where the peaks are in the genome and what their possible function could be. Here, Hypergeometric Optimization of Motif EnRichment (HOMER) is used to analysis and investigate the annotated peaks.

To call the annotated peaks per cell line, annotatePeaks.pl program is used to analysis the .broadPeak files.

Necessary resources list:

Hardware and software: A computer with proper versions of Linux as below.

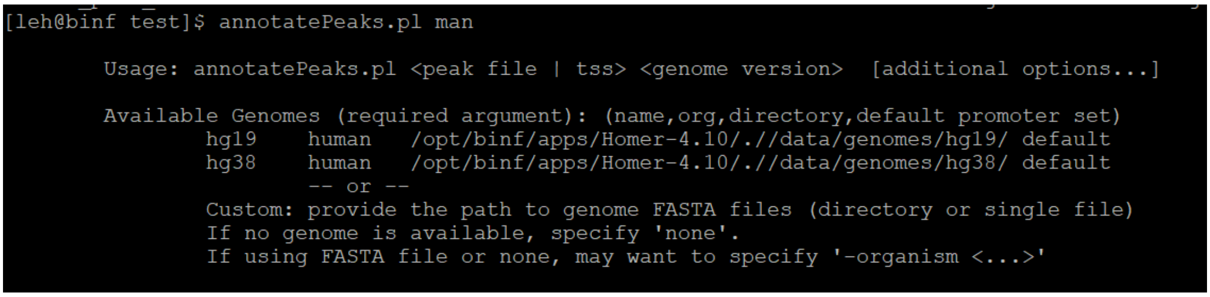


```
[leh@binf test]$ uname -a
Linux binf.rd.tuni.fi 3.10.0-1160.36.2.el7.x86_64 #1 SMP Thu Jul  8 02:53:40 U
TC 2021 x86_64 x86_64 x86_64 GNU/Linux
```

File input: EP156T_peaks.broadPeak and VCap_peaks.broadPeaks

Running environment: installed HOMER vrsion v4.11.

To associate peaks with nearby genes, HOMER contains a useful, all-in-one program for performing peak annotations, it is named as annotatePeaks.pl. This command can perform Gene Ontology Analysis, genomic feature association analysis (Genome Ontology), associates peaks with gene expression data, calculated based on the results of ChIP-seq analysis. AnnotatePeaks.pl is set up with available genomes hg19 and hg38.



```
[leh@binf test]$ annotatePeaks.pl man

        Usage: annotatePeaks.pl <peak file | tss> <genome version>  [additional options...]

        Available Genomes (required argument): (name,org,directory,default promoter set)
                hg19    human   /opt/binf/apps/Homer-4.10/.//data/genomes/hg19/ default
                hg38    human   /opt/binf/apps/Homer-4.10/.//data/genomes/hg38/ default
                        -- or --
                Custom: provide the path to genome FASTA files (directory or single file)
                If no genome is available, specify 'none'.
                If using FASTA file or none, may want to specify '-organism <...>'
```

Output directory: The result of annotatePeaks.pl include lists of annotated peaks per cell line.

student_data/BBT.BI.201_2021/students/leh/project_work

Setting directory before running commands:

Setting directory before running commands:

Working directory is student_data/BBT.BI.201_2021/students/leh/project_work

Execute annotatePeaks.pl by the following command:

annotatePeak.pl <peak/BED file> <genome> -annStats > <output file>

The option -annStats is counted to take the statistical analysis of annotated peaks.

Reference genome here is hg38.

- EP156T cells

annotatePeaks.pl EP156T_peaks.broadPeak hg38 -annStats EP156T_stats.tsv > EP156T_annotate_peak.tsvVCaP cells

- VCaP cells

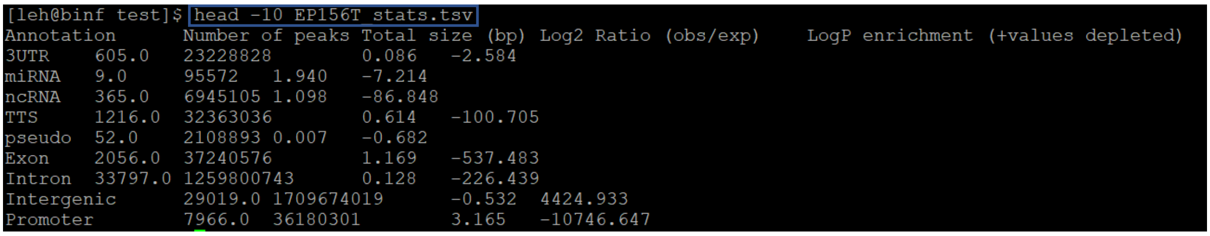
annotatePeaks.pl VCaP_peaks.broadPeak hg38 -annStats VCaP_stats.tsv > VCaP_annotate_peak.tsv

Result:

The results are presented in .tsv file

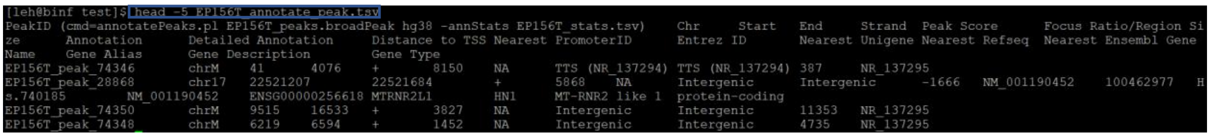
- The results of EP156T cells include 2 files:

EP156T_stats.tsv file includes Annotating results including gene names, number of peaks, total size, log2 ration, and loqP enrichment.



```
[leh@binf test]$ head -10 EP156T_stats.tsv
Annotation      Number of peaks Total size (bp) Log2 Ratio (obs/exp)    LogP enrichment (+values depleted)
3UTR    605.0   23228828        0.086   -2.584
miRNA   9.0     95572   1.940   -7.214
ncRNA   365.0   6945105 1.098   -86.848
TTS     1216.0  32363036        0.614   -100.705
pseudo  52.0    2108893 0.007   -0.682
Exon    2056.0  37240576        1.169   -537.483
Intron  33797.0 1259800743      0.128   -226.439
Intergenic      29019.0 1709674019      -0.532  4424.933
Promoter        7966.0  36180301        3.165   -10746.647
```

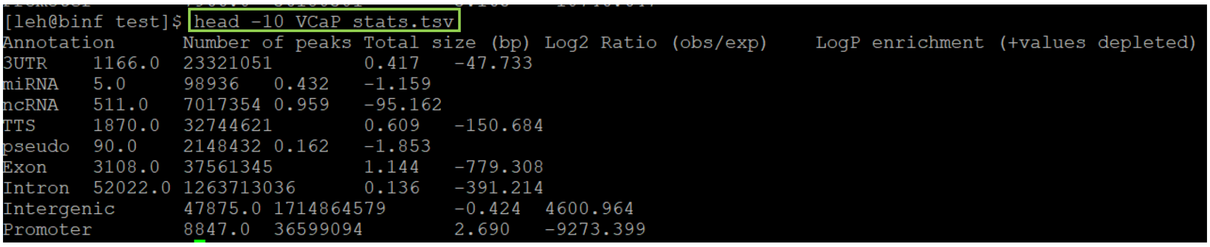
EP156T_annotate_peaks.tsv presents the detail information of annotated peaks e.g peak ID, chromosome, peak start position, peak end position, strand, peak score, FDR focuse rations/region size, etc.



```
[leh@binf test]$ head -5 EP156T_annotate_peak.tsv
PeakID (cmd=annotatePeaks.pl EP156T_peaks.broadPeak hg38 -annStats EP156T_stats.tsv)    Chr     Start   End     Strand  Peak Score      Focus Ratio/Region Si
ze      Annotation      Detailed Annotation     Distance to TSS Nearest PromoterID      Entrez ID       Nearest Unigene Nearest Refseq  Nearest Ensembl Gene
Name    Gene Alias      Gene Description        Gene Type
EP156T_peak_74346       chrM    41      4076    +       8150    NA      TTS (NR_137294) TTS (NR_137294) 387     NR_137295
EP156T_peak_28868       chr17   22521207        22521684        +       5868    NA      Intergenic      Intergenic      -1666   NM_001190452    100462977       H
s.740185        NM_001190452    ENSG00000256618 MTRNR2L1        HN1     MT-RNR2 like 1  protein-coding
EP156T_peak_74350       chrM    9515    16533   +       3827    NA      Intergenic      Intergenic      11353   NR_137295
EP156T_peak_74348       chrM    6219    6594    +       1452    NA      Intergenic      Intergenic      4735    NR_137295
```

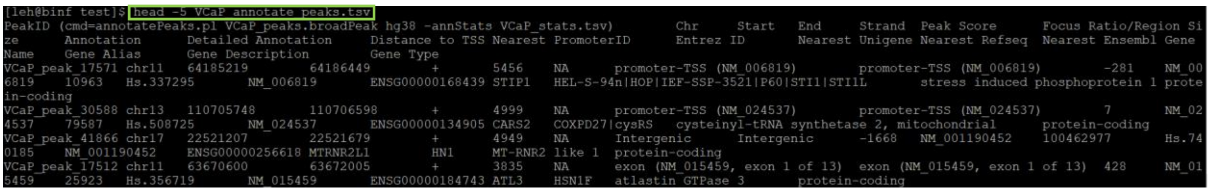
- Similar, the results of VCaP cells have 2 output files

VCaP_stats.tsv file includes Annotating results including gene names, number of peaks, total size, log2 ration, and loqP enrichment.



```
[leh@binf test]$ head -10 VCaP_stats.tsv
Annotation      Number of peaks Total size (bp) Log2 Ratio (obs/exp)    LogP enrichment (+values depleted)
3UTR    1166.0  23321051        0.417   -47.733
miRNA   5.0     98936   0.432   -1.159
ncRNA   511.0   7017354 0.959   -95.162
TTS     1870.0  32744621        0.609   -150.684
pseudo  90.0    2148432 0.162   -1.853
Exon    3108.0  37561345        1.144   -779.308
Intron  52022.0 1263713036      0.136   -391.214
Intergenic      47875.0 1714864579      -0.424  4600.964
Promoter        8847.0  36599094        2.690   -9273.399
```

VCaP_annotate_peaks.tsv presents the detail information of annotated peaks e.g peak ID, chromosome, peak start position, peak end position, strand, peak score, FDR focuse rations/region size, etc.



```
[leh@binf test]$ head -5 VCaP_annotate_peaks.tsv
PeakID (cmd=annotatePeaks.pl VCaP_peaks.broadPeak hg38 -annStats VCaP_stats.tsv)        Chr     Start   End     Strand  Peak Score      Focus Ratio/Region Si
ze      Annotation      Detailed Annotation     Distance to TSS Nearest PromoterID      Entrez ID       Nearest Unigene Nearest Refseq  Nearest Ensembl Gene
Name    Gene Alias      Gene Description        Gene Type
VCaP_peak_17571 chr11   64185219        64186449        +       5456    NA      promoter-TSS (NM_006819)        promoter-TSS (NM_006819)        -281    NM_00
6819    10963   Hs.337295       NM_006819       ENSG00000168439 STIP1   HEL-S-94n|HOP|IEF-SSP-3521|P60|STI1|STI1L       stress induced phosphoprotein 1 prote
in-coding
VCaP_peak_30588 chr13   110705748       110706598       +       4999    NA      promoter-TSS (NM_024537)        promoter-TSS (NM_024537)        7       NM_02
4537    79587   Hs.508725       NM_024537       ENSG00000134905 CARS2   COXPD27|cysRS   cysteinyl-tRNA synthetase 2, mitochondrial      protein-coding
VCaP_peak_41866 chr17   22521207        22521679        +       4949    NA      Intergenic      Intergenic      -1668   NM_001190452    100462977       Hs.74
0185    NM_001190452    ENSG00000256618 MTRNR2L1        HN1     MT-RNR2 like 1  protein-coding
VCaP_peak_17512 chr11   63670600        63672005        +       3835    NA      exon (NM_015459, exon 1 of 13)  exon (NM_015459, exon 1 of 13)  428     NM_01
5459    25923   Hs.356719       NM_015459       ENSG00000184743 ATL3    HSN1F   atlastin GTPase 3       protein-coding
```

Visualization task 1.

In R, load the first 14 rows (including the header) and the log2 Ratio (obs/exp) column of the files containing the annotation statistics and create a bar plot presenting the fold change expression of 13 first genes in the lists of two cell lines.

Necessary resources list:

Hardware and software: A computer with proper versions of R studio version 3.6.0.

Input file: EP156T_stats.tsv and VCaP_stats.tsv

Working session: To set up the working session where store the data files.

pwd() # to check where working session is now.

setwd("/student_data/BBT.BI.201_2021/students/leh/project_work") # the tsv files are stored in the project work directory.

Execute commands in R studio as below:

```
###SET UP WOKING SECTION
```

```
setwd("/student_data/BBT.BI.201_2021/students/leh/project_work”)

###LOAD DATA
EP156T_stats <- read.table(file = 'EP156T_stats.tsv', sep = '\t',
header = TRUE) # Because tsv file includes the header
EP156T_stats_visual <- EP156T_stats[c(1:13), c(1,4)] # To concern
the data from 1 to 13 rows only
View(EP156T_stats_visual)

VCaP_stats<- read.table(file = 'VCaP_stats.tsv', sep = '\t', header
= TRUE) #Because tsv file include the header
VCaP_stats_visual <- VCaP_stats[c(1:13), c(1,4)] # To concern the
data from 1 to 13 rows only
View(VCaP_stats_visual)

###DRAW BARPLOT
EP156T_vec <- as.numeric(EP156T_stats_visual$Log2.Ratio..obs.exp.)
#Because the original data is string, thus it is necessary to change
them to numeric consideration
VCaP_vec <- as.numeric(VCaP_stats_visual$Log2.Ratio..obs.exp.)

all_vec <- c(VCaP_vec, EP156T_vec)
data_matrix <- matrix(all_vec, nrow = 2, byrow = TRUE)
colnames(data_matrix) <- VCaP_stats_visual$Annotation
rownames(data_matrix) <- c("VCaP", "EP156T")
colours = c("green","darkgreen")

barplot(data_matrix,main='Results',ylab = "Log2 ration",ylim = c(-
2, 5), xlab = "Genomic regions",beside = TRUE,  col=colours)
legend('topright',fill=colours,legend=c('VCaP','EP156T'))
```

Results:

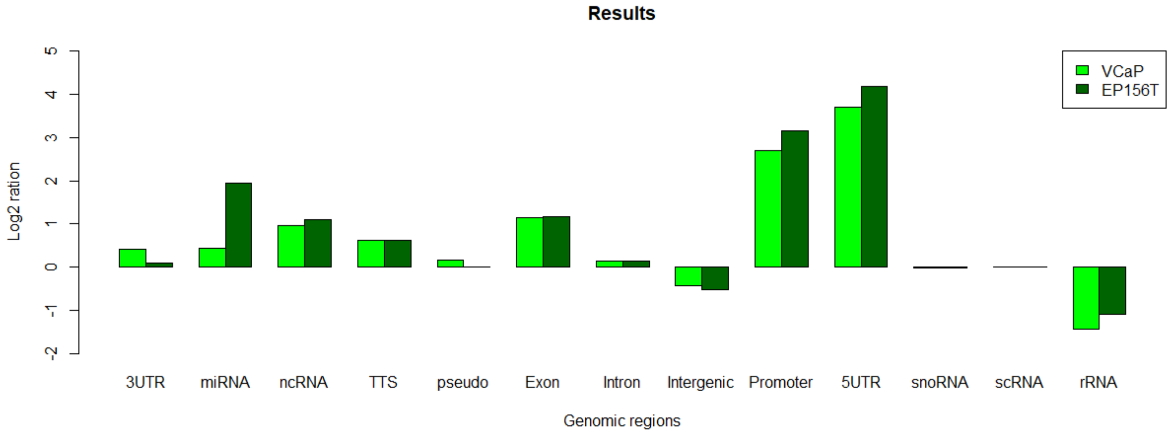


Fig. 2: The fold change expression of 13 first genes in the lists of VCaP and EP156T cells.

Analysis

VCaP cell are an adherent, epithelial cell line with high expression of androgen receptor (AR) and prostate specific antigen expression. Moreover, EP156T cells, primary cells, were isolated from healthy part of prostate cancer patients, which expresses genes consistent with a prostate basal epithelial phenotype. In the present report, it is reported that the significant different expression is observed in 3UTA, miRNA, pseudo, promoter, 5UTR, and rRNA in Fig. 2.

In detail, AR is known as an essential player in the development and progression of prostate cancer. Prostate cancer cells, here, VCaP cells were detected that the mRNA levels of AR were rapidly and substantially increased in response to androgen deprivation, which suggests that AR mediated negative regulation of AR gene expression may make a significant contribution to increasing mRNA levels of AR in cancer. The recent studies have reported that microRNA (miRNA) can target the function of AR promote a functional role of these non-coding RNAs (ncRNA) in the pathogenesis of prostate cancer. In addition, miRNAs bind to the 3' untranslated region (3UTR) of a mRNA by base-paring interactions and modulate translation either by destabilizing the message or by repression of protein synthesis in actively translation. The activation of miRNA has negative regulation with AR signaling. Here, cancer cells VCaP show the down-regulated expression of miRNA whereas 3UTR expression is upregulated in a comparison with non-cancer cells EP156T. Notably, the slight down-expression of ncRNA is determined in VCap cells when comparing to EP156T cells. These findings are in the line with previous studies. In oppositive side, five primer untranslated region (5UTR) has the lower expression in VCaP cells than in EP156T cells. The 5UTR region is the region of mRNA that directly links to initiation codon, which regulates to translational control. It could suggest that the differences of protein levels of two cell lines are affected by 5UTR region [3]

In addition, it is worth to know that non-coding pseudogenes are activated and involved as AR target genes in prostate cancer cells such as KLP4 and KLP1. In the present study, Fig.2 presents that the expression of pseudo in VCaP cells is higher in EP156T cells. Promoter is a sequence of DNA to which proteins binds that initiate transcription of a single RNA from the DNA downstream signaling. This RNA has the information of a protein or can have a function in and of itself such as tRNA, mRNA, or rRNA [4]. Notably, the expression of promoter in VCaP cells is lower than in EP156T cells. It is necessary to suggest the hypothesis that tumor suppressor genes maybe downregulated or inhibit in cancer cells in a comparison with normal cells. Furthermore, the higher expression of ribosomal RNA (rRNA) in VCaP cells is detected when comparing to EP156T cells. The reasons may be explained as alternations in nucleoli, consisting increased number, increased size, a change in altered architecture, and promoted function are known as makers for recognizing prostate cancer cells. Some studies have reported that overexpression of rRNA were determined in the majority of human primary prostate cancer specimens as compared with matched benign prostate tissues [5]. Taken together, the findings in annotated peaks between two cell lines are consisted with previous studies.

Analysis 4

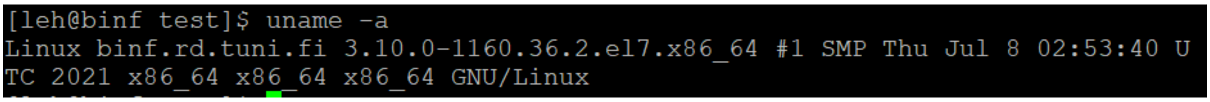
The final analysis that we do here is to check what fraction of the peaks that we have detected may harbor transcription factors (TFs) and in how many peaks a given TF can bind.

The experimental information about the TFs comes from the Gene transcription regulation databased (GTRD). In here, we use only a subset of this database. In general, the most common question asked of two sets of genomic features is whether or not any of the features in the two set "overlap" or similarity with one another. In this analysis, bedtool intersect is used to creen for overlaps between two set of genomic features.

Necessary resources list:

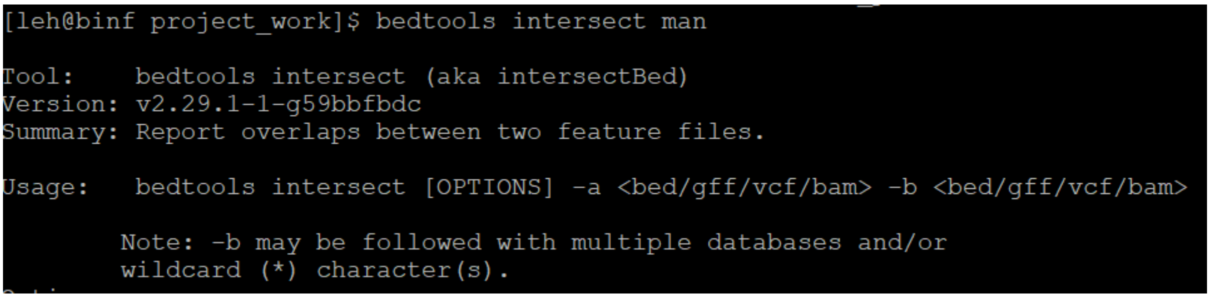
Hardware and software: A computer with proper versions of Linux and bedtools as below.

Linux



```
[leh@binf test]$ uname -a
Linux binf.rd.tuni.fi 3.10.0-1160.36.2.el7.x86_64 #1 SMP Thu Jul 8 02:53:40 U
TC 2021 x86_64 x86_64 x86_64 GNU/Linux
```

Bedtools intersect v2.29.1-1-g59bbfbdc



```
[leh@binf project_work]$ bedtools intersect man

Tool:    bedtools intersect (aka intersectBed)
Version: v2.29.1-1-g59bbfbdc
Summary: Report overlaps between two feature files.

Usage:   bedtools intersect [OPTIONS] -a <bed/gff/vcf/bam> -b <bed/gff/vcf/bam>

        Note: -b may be followed with multiple databases and/or
        wildcard (*) character(s).
```

File input: EP156T_peaks.broadPeak and VCap_peaks.broadPeaks

gtrd_prostate.bed (/student_data/references/gtrd/gtrd_prostate.bed)

Output directory: The result of the two overlapping features between two set files and the command will release one file. Therefore, the output file will be stored in

student_data/BBT.BI.201_2021/students/leh/project_work

Setting directory before running commands:

student_data/BBT.BI.201_2021/students/leh/project_work

Execute annotatePeaks.pl by the following command:

bedtools intersect [OPTIONS) -a <file 1> -b <file 2>

OPTIONs: -wa -wb

The flags -wa and -wb are added so that the features (columns) from the file adter -a and -b flag would also be in the output data.

- EP156T cells

bedtools intersect -wa -wb -a /student_data/references/gtrd/gtrd_prostate.bed -b EP156T_peaks.broadPeak > EP156T_intersect.tsv

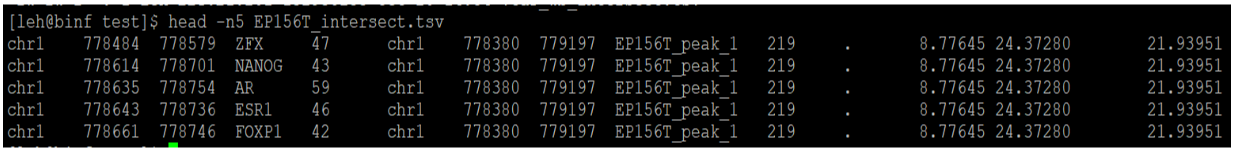
- VCaP cells

bedtools intersect -wa -wb -a /student_data/references/gtrd/gtrd_prostate.bed -b VCaP_peaks.broadPeak > VCaP_intersect.tsv

Results:

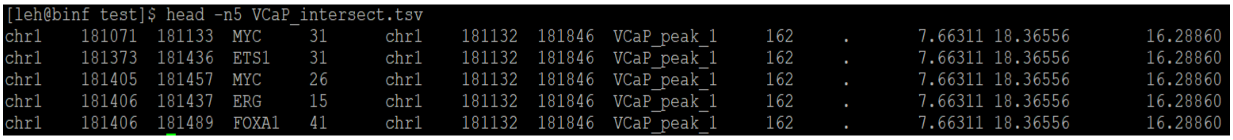
The results are shown in tsv file. After intersection, each peak are presented in 14 columns with detail information about overlap between the peaks and the sites in a comparison with GTRD.

- EP156T cells



```
[leh@binf test]$ head -n5 EP156T_intersect.tsv
chr1    778484  778579  ZFX     47      chr1    778380  779197  EP156T_peak_1   219     .       8.77645 24.37280        21.93951
chr1    778614  778701  NANOG   43      chr1    778380  779197  EP156T_peak_1   219     .       8.77645 24.37280        21.93951
chr1    778635  778754  AR      59      chr1    778380  779197  EP156T_peak_1   219     .       8.77645 24.37280        21.93951
chr1    778643  778736  ESR1    46      chr1    778380  779197  EP156T_peak_1   219     .       8.77645 24.37280        21.93951
chr1    778661  778746  FOXP1   42      chr1    778380  779197  EP156T_peak_1   219     .       8.77645 24.37280        21.93951
```

- VCaP cells



```
[leh@binf test]$ head -n5 VCaP_intersect.tsv
chr1    181071  181133  MYC     31      chr1    181132  181846  VCaP_peak_1     162     .       7.66311 18.36556        16.28860
chr1    181373  181436  ETS1    31      chr1    181132  181846  VCaP_peak_1     162     .       7.66311 18.36556        16.28860
chr1    181405  181457  MYC     26      chr1    181132  181846  VCaP_peak_1     162     .       7.66311 18.36556        16.28860
chr1    181406  181437  ERG     15      chr1    181132  181846  VCaP_peak_1     162     .       7.66311 18.36556        16.28860
chr1    181406  181489  FOXA1   41      chr1    181132  181846  VCaP_peak_1     162     .       7.66311 18.36556        16.28860
```

Here, after intersection of detected peaks with GTRD data using bedtool intersect, a peak is not reported if it does not have an onverlap with a GTRQ feature.

To create a sequence of Unix commands that cuts the first 3 columns (cut -f -1,2,3) of the intersection results, sorts these by the first and the second columns with the second column presented as numeric, find the unique peaks and count them. The sequence command includes cut (cut 3 first columns), sort (sort the first and the second column) and uniq (find the unique peaks), and wc (count the result).

Sequence command:

- EP156T cells

cut -f1-3 EP156T_intersect.tsv | sort -k1 -nk2 | uniq -u | wc -l

cut -f1-3 EP156T_intersect.tsv | sort -k1 -nk2 | uniq -u > EP156T_sort_intersection.tsv

- VCaP cells

cut -f1-3 VCaP_intersect.tsv | sort -k1 -nk2 | uniq -u | wc -l

cut -f1-3 VCaP_intersect.tsv | sort -k1 -nk2 | uniq -u > VCaP_sort_intersection.tsv

Unique peaks

The number of unique peaks in EP156 cells: 762753

The number of unique peaks in VCaP cells: 1287963

Fractions???

I am not sure that I do not misunderstand the question about fractions in here.

Do you mean the fractions between the number of unique peaks with the number of overlap peaks?

In this case, we can calculate based on the equation:

$$Fration = \frac{The\ number\ of\ unique\ peaks}{The\ number\ of\ overlap\ peaks} * 100\%$$

The number of overlap peaks in EP156T cells: 763469

The number of overlap peaks in VCaP cells : 1289123

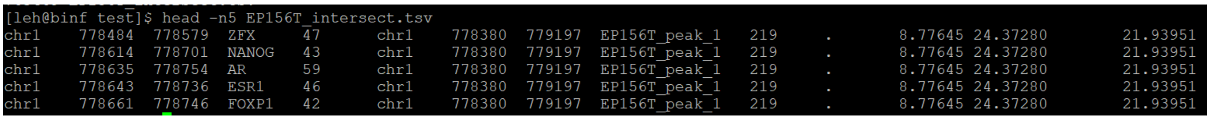
The faction of EP156T cells: 99.96%

The faction of VCaP cells: 99.91%

Visualization task 2:

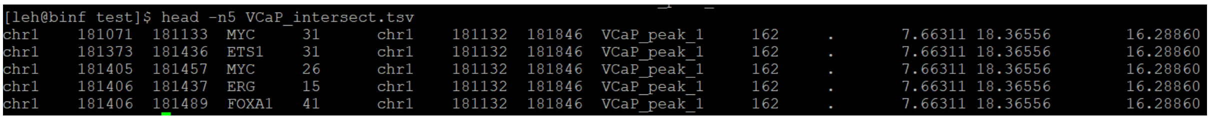
In this analysis 4, we have the intersection files of each cell line. The files include 14 columns to present the detail of overlapping peaks between EP156T cells or VCaP cells with database of subset of 5000 ChIP-seq experiments of prostate cancer samples.

- EP156 cells



```
[leh@binf test]$ head -n5 EP156T_intersect.tsv
chr1    778484  778579  ZFX     47      chr1    778380  779197  EP156T_peak_1   219     .       8.77645 24.37280        21.93951
chr1    778614  778701  NANOG   43      chr1    778380  779197  EP156T_peak_1   219     .       8.77645 24.37280        21.93951
chr1    778635  778754  AR      59      chr1    778380  779197  EP156T_peak_1   219     .       8.77645 24.37280        21.93951
chr1    778643  778736  ESR1    46      chr1    778380  779197  EP156T_peak_1   219     .       8.77645 24.37280        21.93951
chr1    778661  778746  FOXP1   42      chr1    778380  779197  EP156T_peak_1   219     .       8.77645 24.37280        21.93951
```

- VCaP cells

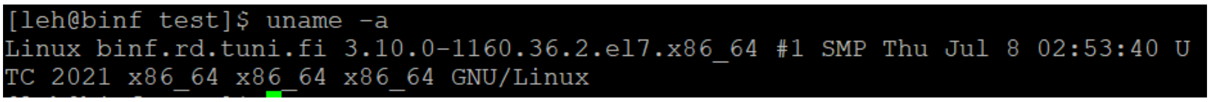


```
[leh@binf test]$ head -n5 VCaP_intersect.tsv
chr1    181071  181133  MYC     31      chr1    181132  181846  VCaP_peak_1     162     .       7.66311 18.36556        16.28860
chr1    181373  181436  ETS1    31      chr1    181132  181846  VCaP_peak_1     162     .       7.66311 18.36556        16.28860
chr1    181405  181457  MYC     26      chr1    181132  181846  VCaP_peak_1     162     .       7.66311 18.36556        16.28860
chr1    181406  181437  ERG     15      chr1    181132  181846  VCaP_peak_1     162     .       7.66311 18.36556        16.28860
chr1    181406  181489  FOXA1   41      chr1    181132  181846  VCaP_peak_1     162     .       7.66311 18.36556        16.28860
```

We want to count the number of different TFs that can bind to annotated peaks at different locations. To determine the numbers of peaks, we create a sequence of Unix commands that cuts the 4th column (Due to the analysis 4, gtrd_prostate.bed file is ordered as the file a in bedtools intersect command, therefore, the columns of annotated gene in here is presented as 4th columns, instead 13th columns in the question), sorts the data and counts the number of times a given TF has been seen and finally sorts the results by TF name.

Necessary resources list:

Hardware and software: A computer with proper versions of Linux as below.



```
[leh@binf test]$ uname -a
Linux binf.rd.tuni.fi 3.10.0-1160.36.2.el7.x86_64 #1 SMP Thu Jul 8 02:53:40 U
TC 2021 x86_64 x86_64 x86_64 GNU/Linux
```

File input: EP156T_peaks.broadPeak and VCap_peaks.broadPeaks

Execute command:

cut | sort | uniq | sort > file.tsv

- EP156T cells

cut -f4 EP156T_intersect.tsv | sort | uniq -c | sort -k2 > EP156T_visualization.csv

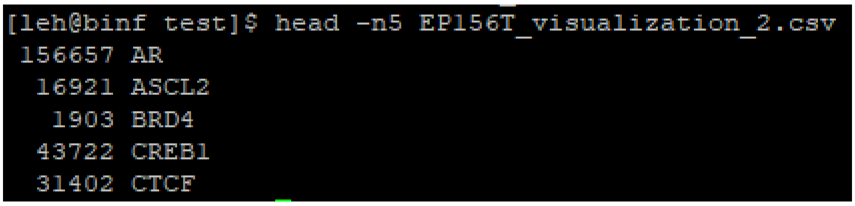
- VCaP cells

cut -f4 VCaP_intersect.tsv | sort | uniq -c | sort -k2 > VCaP_visualization.csv

Result:

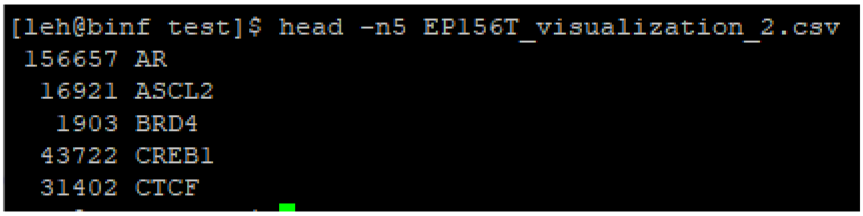
The results are presented as tsv files and include the information of genes.

- EP156 cells results



```
[leh@binf test]$ head -n5 EP156T_visualization_2.csv
 156657 AR
  16921 ASCL2
   1903 BRD4
  43722 CREB1
  31402 CTCF
```

- VCaP cells results



```
[leh@binf test]$ head -n5 EP156T_visualization_2.csv
 156657 AR
  16921 ASCL2
   1903 BRD4
  43722 CREB1
  31402 CTCF
```

Visualization of data by R studio

Hardware and software: A computer with proper versions of R studio version 3.6.0.

Input file: EP156T_visualization.csv and VCaP_visualization.csv

EP156T_intersection.tsv and VCaP_intersection.tsc (from analysis 4)

Working session: To set up the working session where store the data files.

pwd() # to check where working session is now.

setwd("/student_data/BBT.BI.201_2021/students/leh/project_work") # the tsv files are stored in the project work directory.

Execute commands in R studio as below:

```
###SET UP WORKING SECTION
pwd() # Check the location of working section
setwd("~/student_data/BBT.BI.201_2021/student/leh/project_work")    #Set
up the working section

### LOADING THE DATA
VCaP_intersect <- read.table(file = 'VCaP_intersect.tsv', sep = '\t',
header = FALSE) #Load tsv file of overload peaks in VCaP cells
View(VCaP_intersect)
EP156T_intersect <- read.table(file = 'EP156T_intersect.tsv', sep = '\t',
header = FALSE) #Load tsv file of overload peaks in EP156T cells
nrow(EP156T_intersect) #Determine the number of overload peaks in EP156T
cells
nrow(VCaP_intersect)  #Determine  the  number  of  overload  peaks  in  VCaP
cells
VCaP_visualization_2 <- read.csv(file = 'VCaP_visualization_2.csv', sep
= '', header = FALSE)#Load the uniq peaks in VCaP cells in csv file
View(VCaP_visualization_2)
EP156T_visualization_2 <- read.csv(file = 'EP156T_visualization_2.csv',
sep = '', header = FALSE)#Load the uniq peaks in EP156T cells in csv file
View(EP156T_visualization_2)
```

```
###NOMOLIZATION
VCaP_visualization_2$V1 <- VCaP_visualization_2$V1 *100 /
nrow(VCaP_intersect)
EP156T_visualization_2$V1 <- EP156T_visualization_2$V1 *100 /
nrow(EP156T_intersect)

###MERGE DATA
cell_lines_TF = merge(x=VCaP_visualization_2, y =
EP156T_visualization_2, by= 'V2', all = TRUE)
cell_lines_TF
setNames(cell_lines_TF, c("TF", "VCaP", "EP156T"))
cell_lines_TF

###SORT DATA
cell_lines_TF_Sorted <- cell_lines_TF[order(-cell_lines_TF$V1.x, -
cell_lines_TF$V1.y),]
View(cell_lines_TF_Sorted)
colnames(cell_lines_TF_Sorted) <- c("TF", "VCaP", "EP156T")
View(cell_lines_TF)
write.table(cell_lines_TF_Sorted, file='cell_lines_TF_Sorted.tsv',
quote=FALSE, sep='\t') #Write the data in the new file name:
cell_line_TF_Sorted.tsv and save it

## #DRAW BARPLOT
VCaP_value <- as.numeric(cell_lines_TF_Sorted$VCaP)
EP156T_value <- as.numeric(cell_lines_TF_Sorted$EP156T)
all_vec <- c(VCaP_value, EP156T_value)
data_matrix <- matrix(all_vec, nrow = 2, byrow = TRUE)
colnames(data_matrix) <- cell_line_TF_sorted$TF
rownames(data_matrix) <- c(“VCaP”, “EP156T”)

colours = c("green","darkgreen")
par(mar=c(7,4,4,4))
barplot(data_matrix, xlab = "Transcription factor", ylab =
"Percentages(%)", ylim = c(0,25), col = colours, beside = TRUE, las = 2)
legend('topright',fill=colours,legend=c('VCaP','EP156T'))

colnames(data_matrix) <- cell_lines_TF_Sorted$TF
labels <- as.vector(colnames(data_matrix))
text(1:100, par("usr")[1], labels=labels, srt=90, pos=1, xpd=TRUE)
```

Result

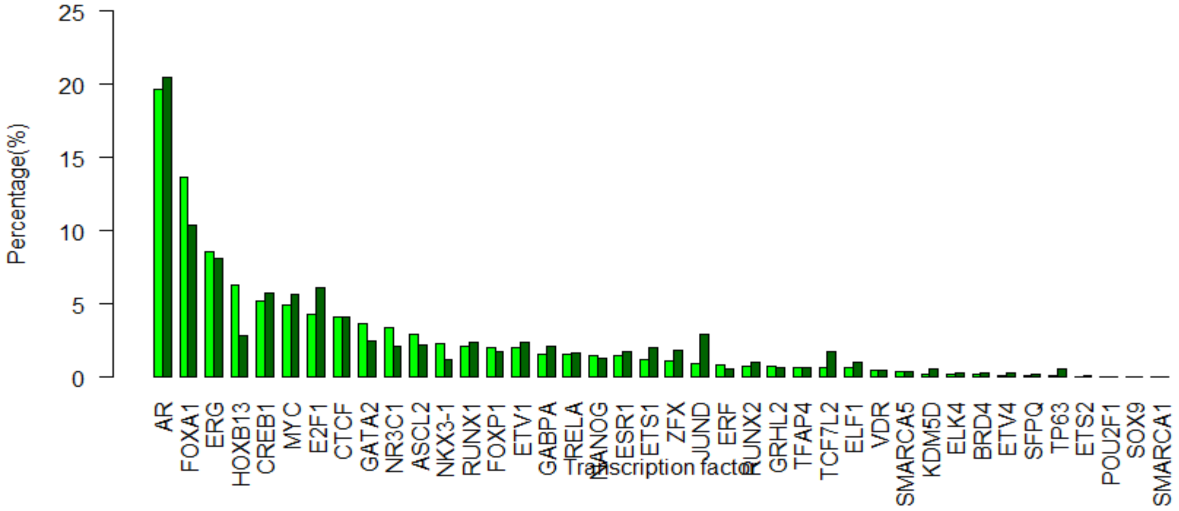


Fig. 3: Listed transcription factors in VCaP and EP156T cells.

Analysis

Identifying prostate cancer-driving transcription factors (TFs) in addition to the androgen receptor promise to improve our ability to effectively investigate and treat this disease. In here, ChIP-seq analysis of VCaP and EP156T cells is performed and analyzed regarding the subset of Gen transcription regulation databased (GTRD) which has collected 5,068 ChIP-seq experiments. In general, it is observed the presence of AR as the highest TF in both prostate cells. In the previous studies, the humane AR is known as a ligand-activated TF that regulates genes crucial for male sexual differentiation and developments, especially in prostate tissues. Thus, the finding about AR in here is in the line with previous conclusion.

In addition, FoxA1 (FOXA1) is a pioneering TF of the AR that is indispensable for the lineage-specific gene expression of the prostate. To date, there have been conflicting reports about the dual role functions of FoxA1 in cancer progression and prognosis. FoxA1 promote cell proliferation dependent on AR signaling pathway whereas the opposite side is reported that FoxA1 suppresses cell mobility and epithelial to mesenchymal transition (EMT) through AR-independent signaling. Importantly, recently identified FoxA1 expression is slightly up-regulated in localized prostate cancer wherein cell proliferation is the main character [6]. In the present study, FoxA1 is detected the remarkably expression in VCaP cells in comparison with the subset of prostate gene databased and EP156 cells. In the same line, it is reported the markable expression and detection of HOXB13-Homeobox B13, GATA-binding factor 2, NR3C1-nuclear receptor subfamily 3 group C, ASCL2-achaete scute complex homolog 2, and NKX3-1 in VCaP cells when comparing with the gene databased and EP156 cells. HOXB13was observed in multiple hereditary prostate cancer and involved in a number of biological functions consisting coactivation and localization of AR and FoxA1. GATA2 is a crucial component in the complex regulatory network of transcription factors that sustains prostate cancer growth in both AR-dependent prostate cancer and castration resistant prostate cancer. It is known that the function of GATA2 may work as a supporter and mediator for AR activation in cancer progression [7]. Additionally, Lin et al reported that compared to normal tissues, the inhibition of ASCL2 links directly to cellular proliferation and tumor growth in xenogralf tumor experiment [8] whereas NKX3-1 plays as an androgen-regulated prostate genes in maintaining prostate cell fate [9]. Notably, the present findings suggest that the activity of AR signaling is observed in VCaP cells more than in EP156T cells. However, AR is the potential target for prostate progression and cancer therapy.

In another side, the lower observation of MYC, GABPA, ESR1, ETS1, and TCF7L2 is determined in VCaP cells in a comparison with EP156T cells. However, the normalization of these TF is notable observed in both cell lines in a comparison with the gene databased of prostate cells. Koh et el revealed that MYC appears to be activated at the earliest phase of prostate cancer and links to disease initiation and progression by promoting an embryonic stem cell-like signature [10]. These insights pave the way to potential novel therapeutic concepts in primary prostate cells based on MYC biology. In addition, in begin and localized prostate cancer, GABPA may act as a tumor suppressor during cancer progression and metastasis and is a potential therapeutic target for chemotherapy [11]. Recently, the cross-talk between androgen and estrogen signaling pathways is determined in the prostate cancer progression. It is worth to add that ESR1 inhibitors can repress prostate cancer tumorigenicity and biological data have suggested that prostate cancer cells can use alternative nuclear receptors signaling pathway such as ESR1 instead of AR signaling to remark the tumor progression [12]. In here, ESR1 is determined in both cell lines and the normalization of ESR1 is higher in EP156T cells. It could be explained that in EP156T cells are determined either AR signaling or estrogen pathways.

Taken together, the results suggest that regarding to the different stage of prostate cancer, the binding motif discovery and enriched pathways could be different. In addition, these findings could be supported the database of prostate gene therapy for investigating the potential targets for cancer treatment.

References:

1. Nakato R, Sakata T (2021) Methods for ChIP-seq analysis: A practical workflow and advanced applications. Methods 187:44–53. https://doi.org/10.1016/J.YMETH.2020.03.005

2. Feng J, Liu T, Zhang Y (2011) Using MACS to Identify Peaks from ChIP-Seq Data. Curr Protoc Bioinformatics CHAPTER:Unit2.14. https://doi.org/10.1002/0471250953.BI0214S34

3. Mignone F, Gissi C, Liuni S, Pesole G (2002) Untranslated regions of mRNAs. Genome Biol 2002 33 3:1–10. https://doi.org/10.1186/GB-2002-3-3-REVIEWS0004

4. BV C, P D-A, S C, et al (2019) Pseudogene Associated Recurrent Gene Fusion in Prostate Cancer. Neoplasia 21:989–1002. https://doi.org/10.1016/J.NEO.2019.07.010

5. Uemura M, Zheng Q, Koh CM, et al (2011) Overexpression of ribosomal RNA in prostate cancer is common but not linked to rDNA promoter hypomethylation. Oncogene 2012 3110 31:1254–1263. https://doi.org/10.1038/onc.2011.319

6. Jin H-J, Zhao JC, Ogden I, et al (2013) Androgen receptor-independent function of FoxA1 in prostate cancer metastasis. Cancer Res 73:3725. https://doi.org/10.1158/0008-5472.CAN-12-3468

7. Chaytor L, Simcock M, Nakjang S, et al (2019) The Pioneering Role of GATA2 in Androgen Receptor Variant Regulation Is Controlled by Bromodomain and Extraterminal Proteins in Castrate-Resistant Prostate Cancer. Mol Cancer Res 17:1264–1278. https://doi.org/10.1158/1541-7786.MCR-18-1231

8. CY W, P S, JT H, et al (2017) Systematic analysis of the achaete-scute complex-like gene signature in clinical cancer patients. Mol Clin Oncol 6:7–18. https://doi.org/10.3892/MCO.2016.1094

9. Q X, ZA W (2017) Transcriptional regulation of the Nkx3.1 gene in prostate luminal stem cell specification and cancer initiation via its 3' genomic region. J Biol Chem 292:13521–13530. https://doi.org/10.1074/JBC.M117.788315

10. Koh CM, Bieberich CJ, Dang C V., et al (2010) MYC and Prostate Cancer. Genes Cancer 1:617. https://doi.org/10.1177/1947601910379132

11. Hollenhorst PC (2012) RAS/ERK pathway transcriptional regulation through ETS/AP-1 binding sites. http://dx.doi.org/104161/sgtp19630 3:154–158. https://doi.org/10.4161/SGTP.19630

12. Mishra S, Tai Q, Gu X, et al (2015) Estrogen and estrogen receptor alpha promotes malignancy and osteoblastic tumorigenesis in prostate cancer. Oncotarget 6:44388. https://doi.org/10.18632/ONCOTARGET.6317