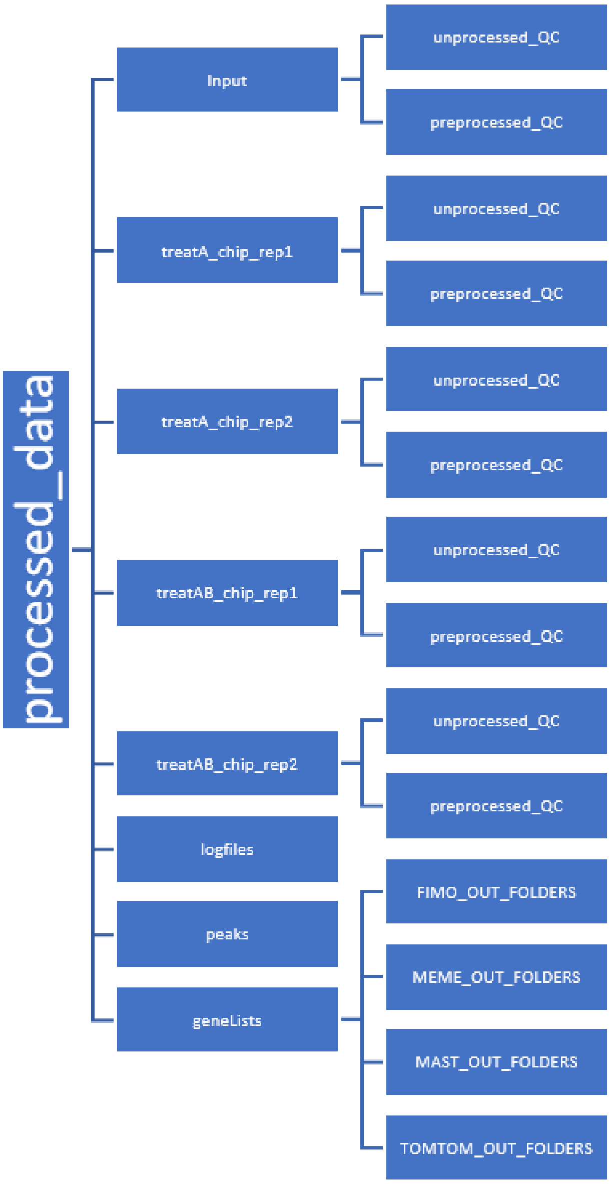
MCB 5430 midterm assignment

Bianca Mocanu



Tree folder structure of the script output

1 ChIP-seq processing, alignment and display

1.1 Pipeline

The shell script used in processing all of the data in this midterm can be found here.

1.2 Summary table

The mapping of the reads has been done with the -m 1 option. Therefore, the ”reads with at least one reported alignment” field in the log file refers to the number of uniquely mapped reads that fulfilled the mapping parameters. Table 1 displays the summary of the reads alignment step of the pipeline.

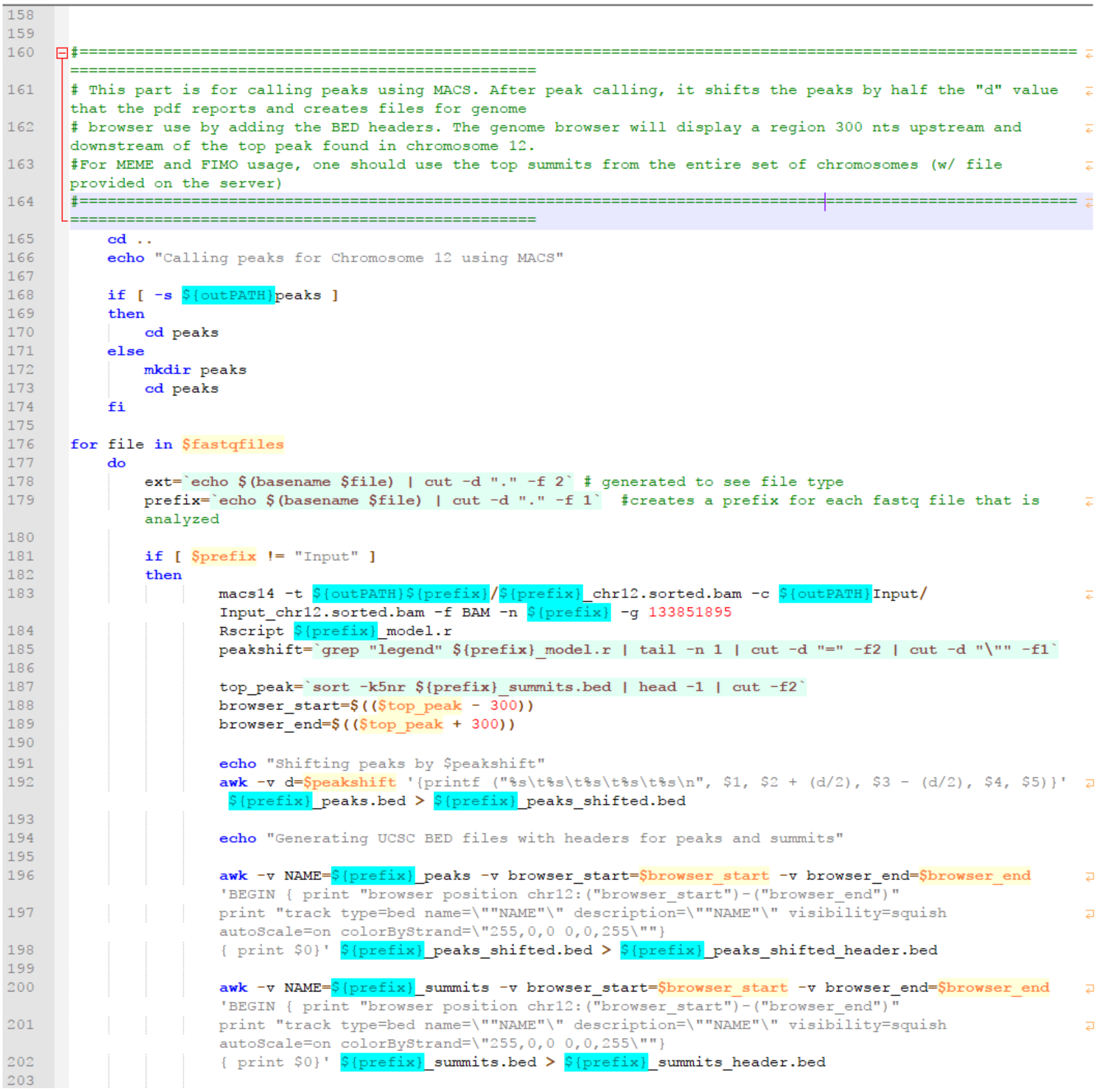
Table 1. Summary table for the alignment step of the pipeline. Reads total refers to the number of fastq reads that passed the preprocessing steps of the pipeline. ”# and % unique” display the number and the percentage of uniquely mapped reads. ”# and % multiple” display the number and the percentage of multiply mapped reads that got discarded due to the constrains of the mapping parameter -m 1. ”# and % unaligned” display the number and percentage of reads that did not align to the hg19 genome.

Sample	# reads total	# unique	# multiple	# unaligned	% unique	% multiple	% unaligned
Input	12,265,901	10,088,192	1,016,900	1,160,809	82.25	8.29	9.46
treatment A rep 1	13,164,770	10,716,882	921,317	1,526,571	81.41	7.00	11.60
treatment A rep 2	14,071,793	11,248,659	971,196	1,851,938	79.94	6.90	13.16
treatment AB rep 1	13,132,269	10,686,406	955,345	1,490,518	81.38	7.27	11.35
treatment AB rep 2	8,725,137	6,948,241	601,088	1,175,808	79.63	6.89	13.48

2 Peak calling and analysis

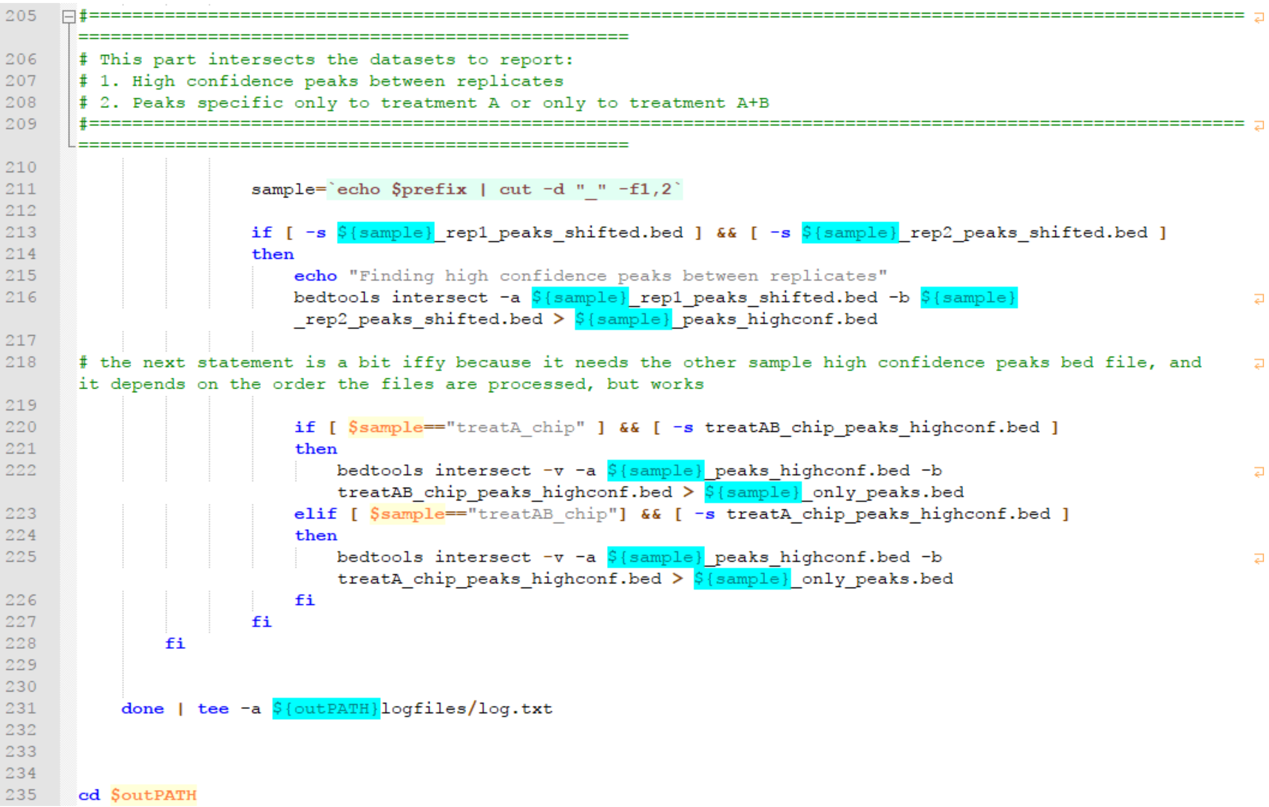
The following part of the script includes the peak calling step, high confidence peak files generation and getting the peaks unique to each treatment. Since they were all typed in a single for loop, they are displayed together below (therefore the for loop ends on next page).

2.1 MACS



```


#=============================================================================================================================================================
# This part is for calling peaks using MACS. After peak calling, it shifts the peaks by half the "d" value that the pdf reports and creates files for genome
# browser use by adding the BED headers. The genome browser will display a region 300 nts upstream and downstream of the top peak found in chromosome 12.
#For MEME and FIMO usage, one should use the top summits from the entire set of chromosomes (w/ file provided on the server)
#=============================================================================================================================================================
    cd ..
    echo "Calling peaks for Chromosome 12 using MACS"

    if [ -s ${outPATH}peaks ]
    then
        cd peaks
    else
        mkdir peaks
        cd peaks
    fi

for file in $fastqfiles
    do
        ext=`echo $(basename $file) | cut -d "." -f 2` # generated to see file type
        prefix=`echo $(basename $file) | cut -d "." -f 1`  #creates a prefix for each fastq file that is analyzed

        if [ $prefix != "Input" ]
        then
                macs14 -t ${outPATH}${prefix}/${prefix}_chr12.sorted.bam -c ${outPATH}Input/Input_chr12.sorted.bam -f BAM -n ${prefix} -g 133851895
                Rscript ${prefix}_model.r
                peakshift=`grep "legend" ${prefix}_model.r | tail -n 1 | cut -d "=" -f2 | cut -d "\"" -f1`

                top_peak=`sort -k5nr ${prefix}_summits.bed | head -1 | cut -f2`
                browser_start=$(($top_peak - 300))
                browser_end=$(($top_peak + 300))

                echo "Shifting peaks by $peakshift"
                awk -v d=$peakshift '{printf ("%s\t%s\t%s\t%s\t%s\n", $1, $2 + (d/2), $3 - (d/2), $4, $5)}' ${prefix}_peaks.bed > ${prefix}_peaks_shifted.bed

                echo "Generating UCSC BED files with headers for peaks and summits"

                awk -v NAME=${prefix}_peaks -v browser_start=$browser_start -v browser_end=$browser_end 'BEGIN { print "browser position chr12:("browser_start")-("browser_end")"
                print "track type=bed name=\""NAME"\" description=\""NAME"\" visibility=squish autoScale=on colorByStrand=\"255,0,0 0,0,255\""}
                { print $0}' ${prefix}_peaks_shifted.bed > ${prefix}_peaks_shifted_header.bed

                awk -v NAME=${prefix}_summits -v browser_start=$browser_start -v browser_end=$browser_end 'BEGIN { print "browser position chr12:("browser_start")-("browser_end")"
                print "track type=bed name=\""NAME"\" description=\""NAME"\" visibility=squish autoScale=on colorByStrand=\"255,0,0 0,0,255\""}
                { print $0}' ${prefix}_summits.bed > ${prefix}_summits_header.bed

```

2.2 High confidence peaks and 2.3 Peaks specific to each treatment



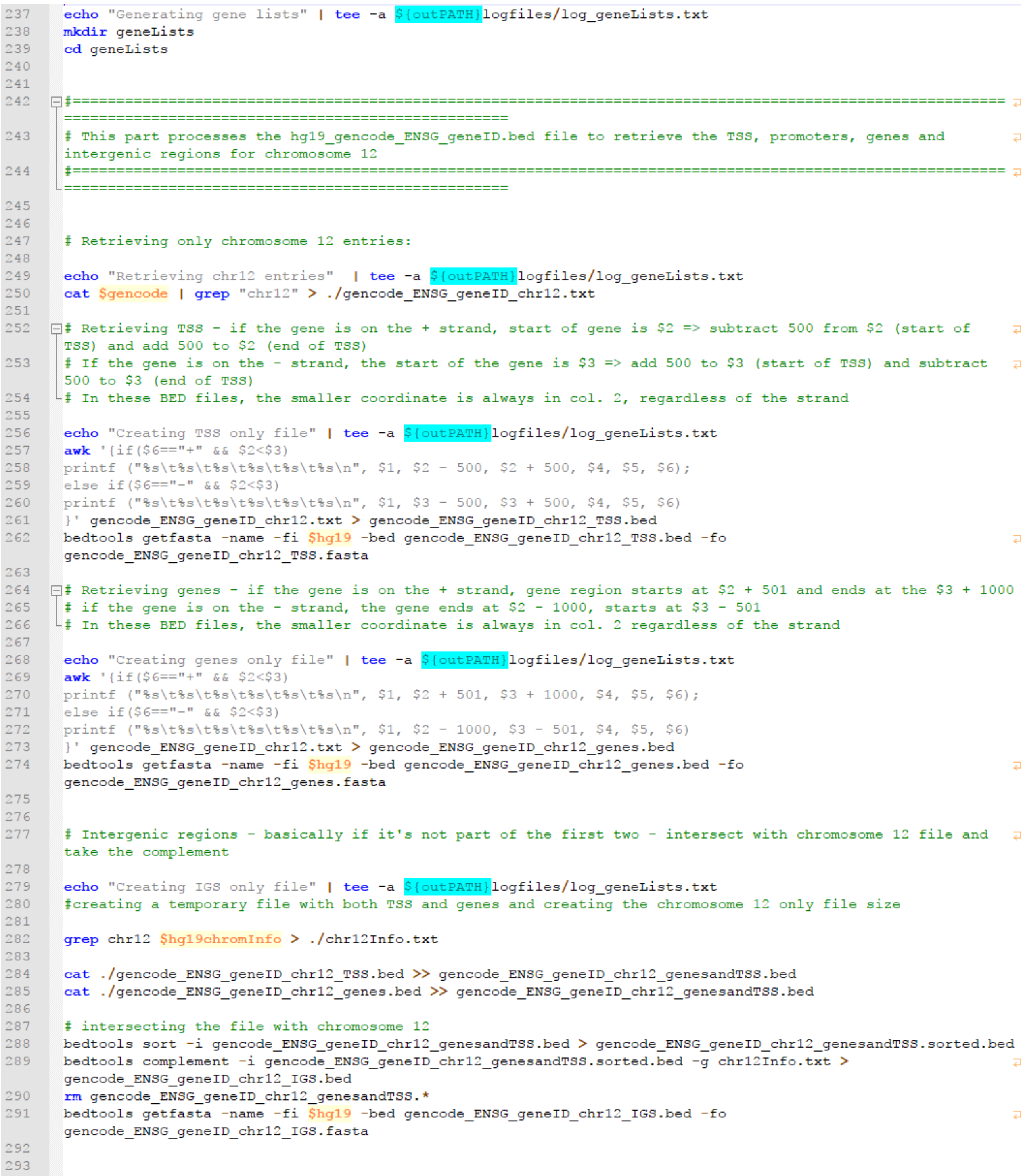
```
#====================================================================================================================================================
# This part intersects the datasets to report:
# 1. High confidence peaks between replicates
# 2. Peaks specific only to treatment A or only to treatment A+B
#====================================================================================================================================================

                sample=`echo $prefix | cut -d "_" -f1,2`

                if [ -s ${sample}_rep1_peaks_shifted.bed ] && [ -s ${sample}_rep2_peaks_shifted.bed ]
                then
                    echo "Finding high confidence peaks between replicates"
                    bedtools intersect -a ${sample}_rep1_peaks_shifted.bed -b ${sample}_rep2_peaks_shifted.bed > ${sample}_peaks_highconf.bed

# the next statement is a bit iffy because it needs the other sample high confidence peaks bed file, and it depends on the order the files are processed, but works

                    if [ $sample=="treatA_chip" ] && [ -s treatAB_chip_peaks_highconf.bed ]
                    then
                        bedtools intersect -v -a ${sample}_peaks_highconf.bed -b treatAB_chip_peaks_highconf.bed > ${sample}_only_peaks.bed
                    elif [ $sample=="treatAB_chip"] && [ -s treatA_chip_peaks_highconf.bed ]
                    then
                        bedtools intersect -v -a ${sample}_peaks_highconf.bed -b treatA_chip_peaks_highconf.bed > ${sample}_only_peaks.bed
                    fi
                fi
        fi


    done | tee -a ${outPATH}logfiles/log.txt



cd $outPATH
```

Observation: The file containing peaks unique to treatment AB only is an empty file. This does not seem to be a script error - upon analysis of the peaks common to both treatments and peaks unique to treatment A, it is likely that AB peaks are a subset of A peaks. This already might mean that whatever treatment AB is, it inhibits the binding of this transcription factor to DNA.

3 Distribution of TF binding sites

3.1 Bed files with promoter, gene and intergenic sequences

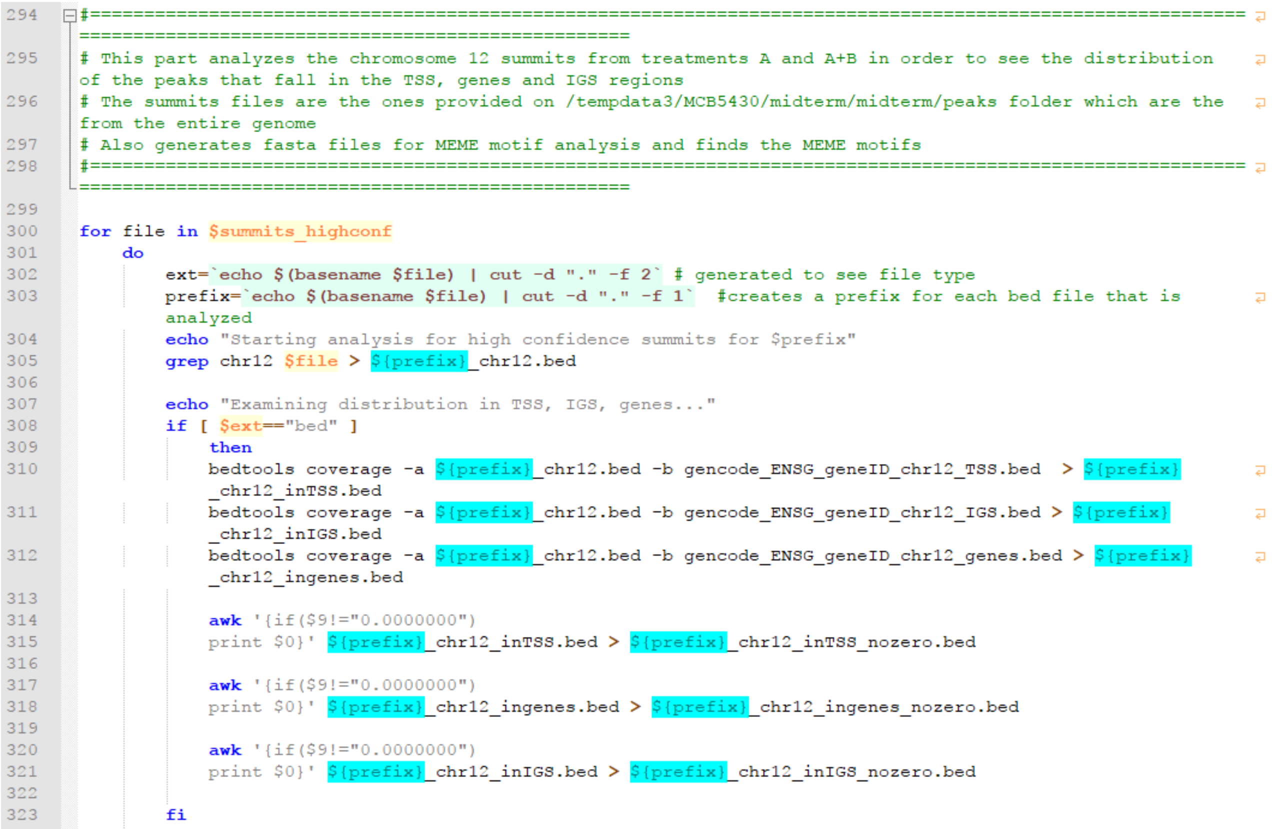
For this step, TSS abbreviation of the files stands for the promoters, genes represent the genes and IGS represent the intergenic sequences. In addition to .bed files for each region type, .fasta files have been created here as well.



```
echo "Generating gene lists" | tee -a ${outPATH}logfiles/log_geneLists.txt
mkdir geneLists
cd geneLists


#==============================================================================================================================================================
# This part processes the hg19_gencode_ENSG_geneID.bed file to retrieve the TSS, promoters, genes and intergenic regions for chromosome 12
#==============================================================================================================================================================


# Retrieving only chromosome 12 entries:

echo "Retrieving chr12 entries"  | tee -a ${outPATH}logfiles/log_geneLists.txt
cat $gencode | grep "chr12" > ./gencode_ENSG_geneID_chr12.txt

# Retrieving TSS - if the gene is on the + strand, start of gene is $2 => subtract 500 from $2 (start of TSS) and add 500 to $2 (end of TSS)
# If the gene is on the - strand, the start of the gene is $3 => add 500 to $3 (start of TSS) and subtract 500 to $3 (end of TSS)
# In these BED files, the smaller coordinate is always in col. 2, regardless of the strand

echo "Creating TSS only file" | tee -a ${outPATH}logfiles/log_geneLists.txt
awk '{if($6=="+" && $2<$3)
printf ("%s\t%s\t%s\t%s\t%s\t%s\n", $1, $2 - 500, $2 + 500, $4, $5, $6);
else if($6=="-" && $2<$3)
printf ("%s\t%s\t%s\t%s\t%s\t%s\n", $1, $3 - 500, $3 + 500, $4, $5, $6)
}' gencode_ENSG_geneID_chr12.txt > gencode_ENSG_geneID_chr12_TSS.bed
bedtools getfasta -name -fi $hg19 -bed gencode_ENSG_geneID_chr12_TSS.bed -fo gencode_ENSG_geneID_chr12_TSS.fasta

# Retrieving genes - if the gene is on the + strand, gene region starts at $2 + 501 and ends at the $3 + 1000
# if the gene is on the - strand, the gene ends at $2 - 1000, starts at $3 - 501
# In these BED files, the smaller coordinate is always in col. 2 regardless of the strand

echo "Creating genes only file" | tee -a ${outPATH}logfiles/log_geneLists.txt
awk '{if($6=="+" && $2<$3)
printf ("%s\t%s\t%s\t%s\t%s\t%s\n", $1, $2 + 501, $3 + 1000, $4, $5, $6);
else if($6=="-" && $2<$3)
printf ("%s\t%s\t%s\t%s\t%s\t%s\n", $1, $2 - 1000, $3 - 501, $4, $5, $6)
}' gencode_ENSG_geneID_chr12.txt > gencode_ENSG_geneID_chr12_genes.bed
bedtools getfasta -name -fi $hg19 -bed gencode_ENSG_geneID_chr12_genes.bed -fo gencode_ENSG_geneID_chr12_genes.fasta


# Intergenic regions - basically if it's not part of the first two - intersect with chromosome 12 file and take the complement

echo "Creating IGS only file" | tee -a ${outPATH}logfiles/log_geneLists.txt
#creating a temporary file with both TSS and genes and creating the chromosome 12 only file size

grep chr12 $hg19chromInfo > ./chr12Info.txt

cat ./gencode_ENSG_geneID_chr12_TSS.bed >> gencode_ENSG_geneID_chr12_genesandTSS.bed
cat ./gencode_ENSG_geneID_chr12_genes.bed >> gencode_ENSG_geneID_chr12_genesandTSS.bed

# intersecting the file with chromosome 12
bedtools sort -i gencode_ENSG_geneID_chr12_genesandTSS.bed > gencode_ENSG_geneID_chr12_genesandTSS.sorted.bed
bedtools complement -i gencode_ENSG_geneID_chr12_genesandTSS.sorted.bed -g chr12Info.txt > gencode_ENSG_geneID_chr12_IGS.bed
rm gencode_ENSG_geneID_chr12_genesandTSS.*
bedtools getfasta -name -fi $hg19 -bed gencode_ENSG_geneID_chr12_IGS.bed -fo gencode_ENSG_geneID_chr12_IGS.fasta


```

3.2 Determining the distribution of high confidence peaks (summits) that fall into promoter, genes and intergenic sequences

This entire code block is also under the same for loop which analyzes everything down to the TOMTOM step (therefore, it's far from the 'done' line)



```
#==============================================================================================================================================================
# This part analyzes the chromosome 12 summits from treatments A and A+B in order to see the distribution of the peaks that fall in the TSS, genes and IGS regions
# The summits files are the ones provided on /tempdata3/MCB5430/midterm/midterm/peaks folder which are the from the entire genome
# Also generates fasta files for MEME motif analysis and finds the MEME motifs
#==============================================================================================================================================================

for file in $summits_highconf
    do
        ext=`echo $(basename $file) | cut -d "." -f 2` # generated to see file type
        prefix=`echo $(basename $file) | cut -d "." -f 1`  #creates a prefix for each bed file that is analyzed
        echo "Starting analysis for high confidence summits for $prefix"
        grep chr12 $file > ${prefix}_chr12.bed

        echo "Examining distribution in TSS, IGS, genes..."
        if [ $ext=="bed" ]
            then
            bedtools coverage -a ${prefix}_chr12.bed -b gencode_ENSG_geneID_chr12_TSS.bed  > ${prefix}_chr12_inTSS.bed
            bedtools coverage -a ${prefix}_chr12.bed -b gencode_ENSG_geneID_chr12_IGS.bed > ${prefix}_chr12_inIGS.bed
            bedtools coverage -a ${prefix}_chr12.bed -b gencode_ENSG_geneID_chr12_genes.bed > ${prefix}_chr12_ingenes.bed

            awk '{if($9!="0.0000000")
            print $0}' ${prefix}_chr12_inTSS.bed > ${prefix}_chr12_inTSS_nozero.bed

            awk '{if($9!="0.0000000")
            print $0}' ${prefix}_chr12_ingenes.bed > ${prefix}_chr12_ingenes_nozero.bed

            awk '{if($9!="0.0000000")
            print $0}' ${prefix}_chr12_inIGS.bed > ${prefix}_chr12_inIGS_nozero.bed

        fi
```

3.3 Summary table for distribution in genomic regions

Table 2 contains a summary for the section 3.2 and for MAST and FIMO. The following example calculations show how the table was compiled:

1. Grey rows: determining the distribution of high confidence peaks (summits) in different genome regions

The numbers for overlapping summits for each genomic region are present in the columns starting with #. The Total represents the sum of all summits falling in promoters, genes and intergenic sequences, and the percentages are calculated by dividing the number for each genomic region by the total number of summits:

% in genes = (# in genes / # Total) * 100

e.g. For Treatment A:
% in genes = (# in genes / # Total) * 100
% in genes = (411 / 1032) * 100
% in genes = **39.82**.

2. MAST outputs: determining how many peaks have motifs (peaks being broken down by different genome regions)

Starting for example with *treatment A intergenic regions*, 117 peaks have motif 1 and 137 peaks have motif 2. The total number of peaks with motifs for treatment A is therefore 117+137 = 254. To obtain the number of peaks without motifs, 254 was subtracted from the *total peaks from intergenic regions* calculated in section 3.2 (in the grey bar):

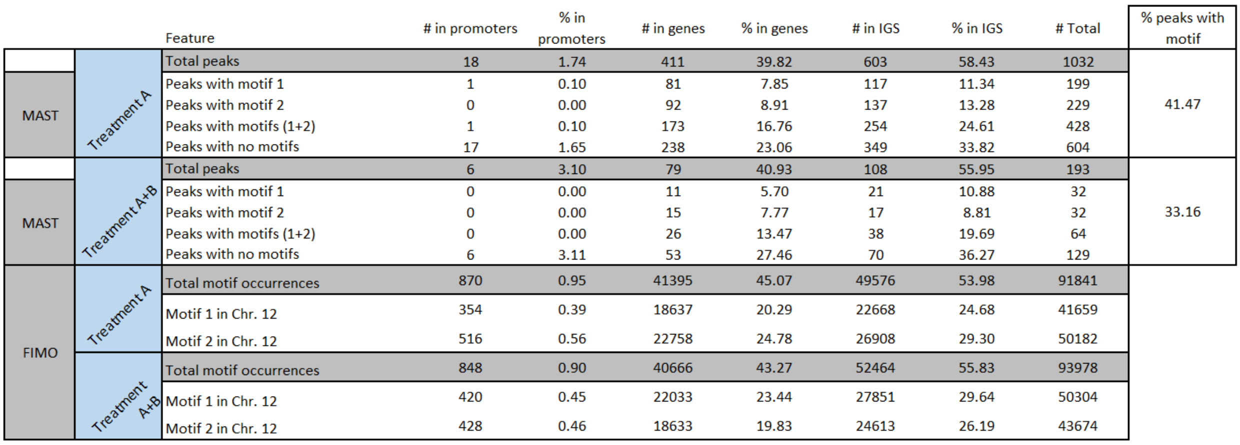
# peaks with no motifs in IGS = total # peaks in IGS - total # peaks w/ motifs in IGS
# peaks with no motifs = 603 - 254 = **349**

To further calculate the percentages, I divided the number of peaks (having a certain motif in a certain genomic region) by the total number of peaks, regardless of genomic region. For example, for the treatment A intergenic regions for motif 1, the calculations were as following:

% in IGS = (# in IGS / # Total) * 100
% in IGS = (117 / 1032) * 100
% in IGS = 11.34

This way, by looking at a genomic region percentages column, the white part of the table is essentially a breakdown of the grey row percentage. The table can be read, for example, like this: for treatment A, 58.43% of peaks are in intergenic regions and out of 58.43 percents 11.34 of them have motif 1, 13.28 of them have motif 2 and 33.82 of them display no motif.

Table 2. Summary of MAST, FIMO and distribution of summits in different genomic regions. Certain features determined with either MAST, FIMO or bedtools are broken down by intergenic regions. Raw number of peaks as well as calculated percentages of the total are displayed.



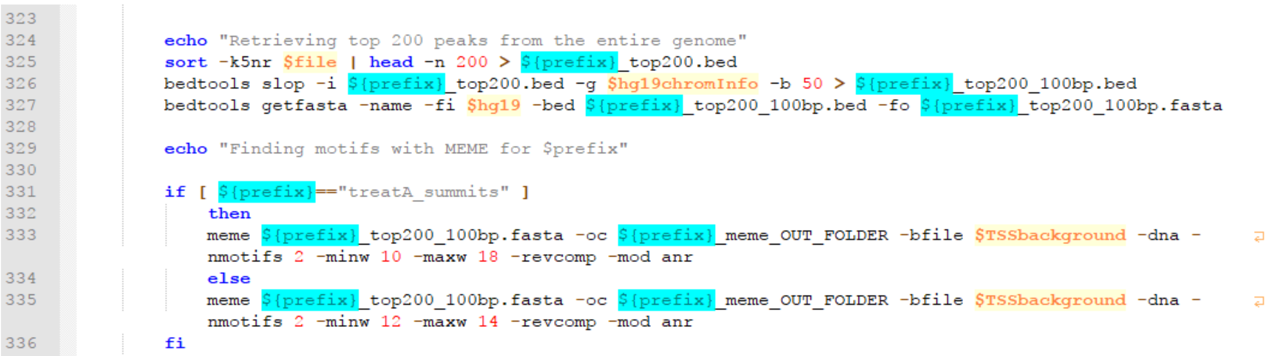
		Feature	# in promoters	% in promoters	# in genes	% in genes	# in IGS	% in IGS	# Total	% peaks with motif
	Treatment A	Total peaks	18	1.74	411	39.82	603	58.43	1032	41.47
MAST		Peaks with motif 1	1	0.10	81	7.85	117	11.34	199	
		Peaks with motif 2	0	0.00	92	8.91	137	13.28	229	
		Peaks with motifs (1+2)	1	0.10	173	16.76	254	24.61	428	
		Peaks with no motifs	17	1.65	238	23.06	349	33.82	604	
	Treatment A+B	Total peaks	6	3.10	79	40.93	108	55.95	193	33.16
MAST		Peaks with motif 1	0	0.00	11	5.70	21	10.88	32	
		Peaks with motif 2	0	0.00	15	7.77	17	8.81	32	
		Peaks with motifs (1+2)	0	0.00	26	13.47	38	19.69	64	
		Peaks with no motifs	6	3.11	53	27.46	70	36.27	129	
FIMO	Treatment A	Total motif occurrences	870	0.95	41395	45.07	49576	53.98	91841	
		Motif 1 in Chr. 12	354	0.39	18637	20.29	22668	24.68	41659	
		Motif 2 in Chr. 12	516	0.56	22758	24.78	26908	29.30	50182	
	Treatment A+B	Total motif occurrences	848	0.90	40666	43.27	52464	55.83	93978	
		Motif 1 in Chr. 12	420	0.45	22033	23.44	27851	29.64	50304	
		Motif 2 in Chr. 12	428	0.46	18633	19.83	24613	26.19	43674	

3.4 TF preferences

This transcription factor's preference for promoters or enhancers can be assessed by looking at the number of peaks falling in promoter regions and outside promoter regions. According to Table 2, for treatment A, 1.74% of the summits are in the promoter regions, while 98.25% are in genes and intergenic sequences. For treatment A+B, 3.10 % of the summits are in the promoter regions, while 96.89% of the summits are in genes and intergenic sequences. This means that the transcription factor has a strong preference to bind **enhancers**, rather than promoters.

4 Identifying motifs under peaks

4.1 MEME

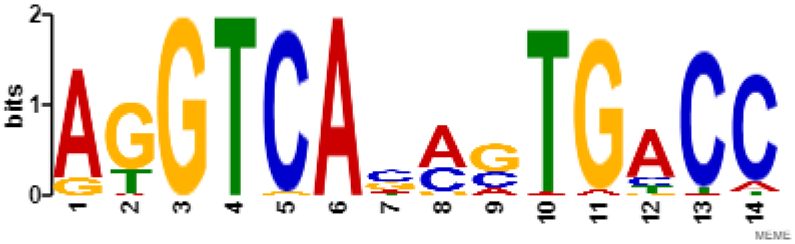


```
        echo "Retrieving top 200 peaks from the entire genome"
        sort -k5nr $file | head -n 200 > ${prefix}_top200.bed
        bedtools slop -i ${prefix}_top200.bed -g $hg19chromInfo -b 50 > ${prefix}_top200_100bp.bed
        bedtools getfasta -name -fi $hg19 -bed ${prefix}_top200_100bp.bed -fo ${prefix}_top200_100bp.fasta

        echo "Finding motifs with MEME for $prefix"

        if [ ${prefix}=="treatA_summits" ]
            then
            meme ${prefix}_top200_100bp.fasta -oc ${prefix}_meme_OUT_FOLDER -bfile $TSSbackground -dna -nmotifs 2 -minw 10 -maxw 18 -revcomp -mod anr
            else
            meme ${prefix}_top200_100bp.fasta -oc ${prefix}_meme_OUT_FOLDER -bfile $TSSbackground -dna -nmotifs 2 -minw 12 -maxw 14 -revcomp -mod anr
        fi
```

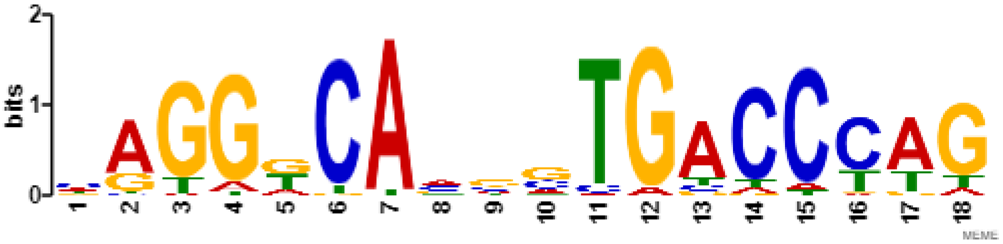
Using the code block above, the following motifs have been found (not displaying the reverse complements):

Treatment A

Motif 1:

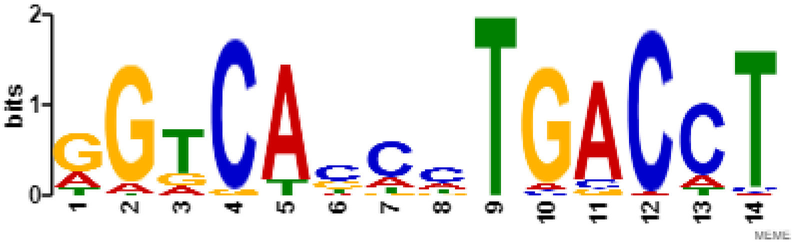


Motif 2:

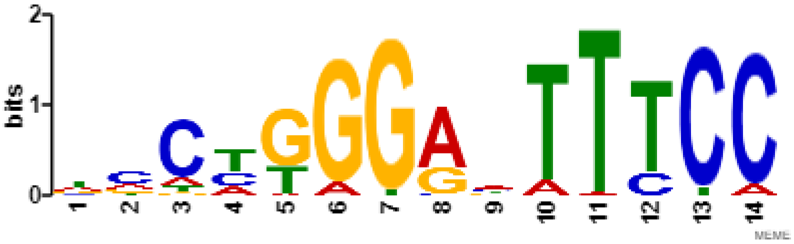


Treatment A+B

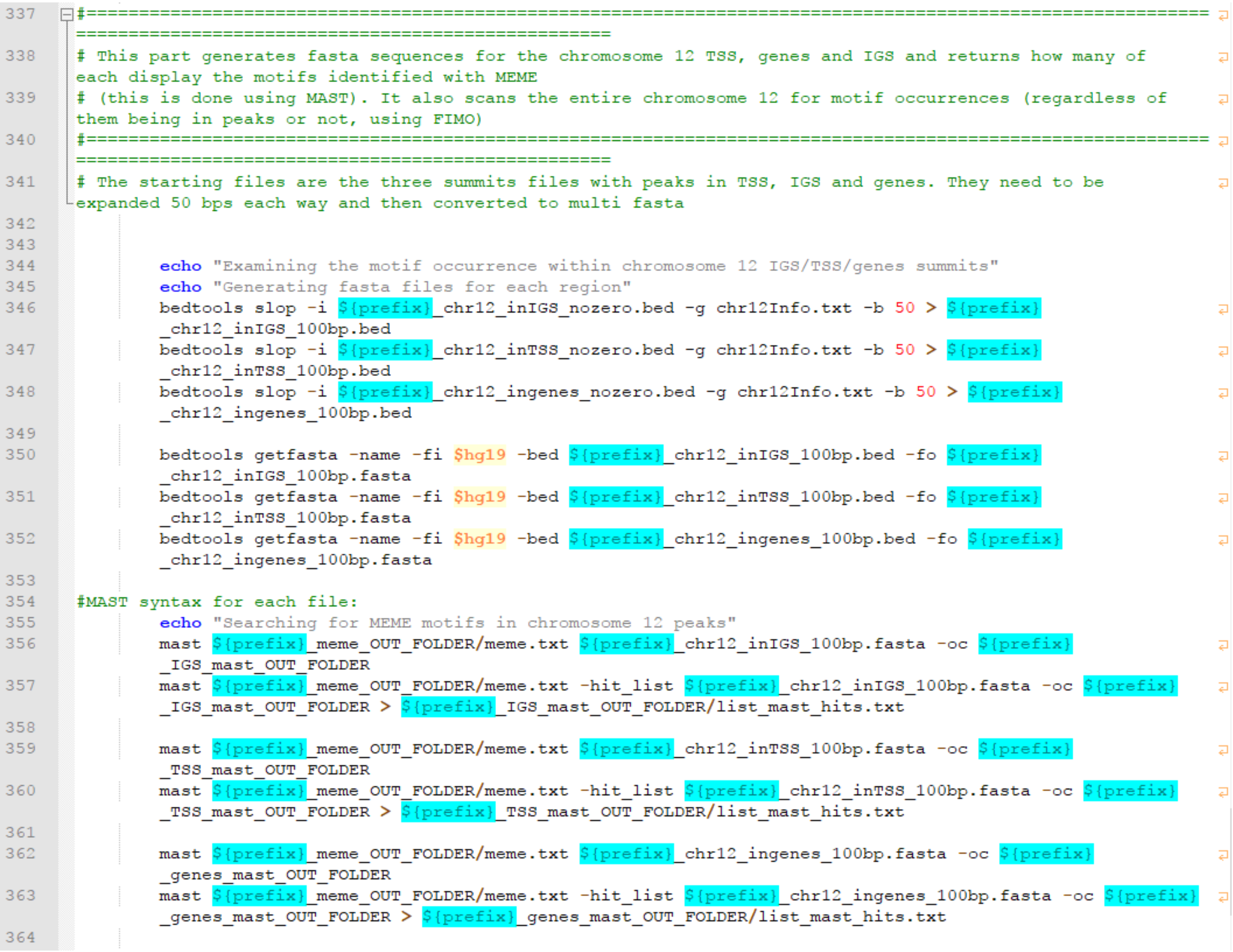
Motif 1:



Motif 2:

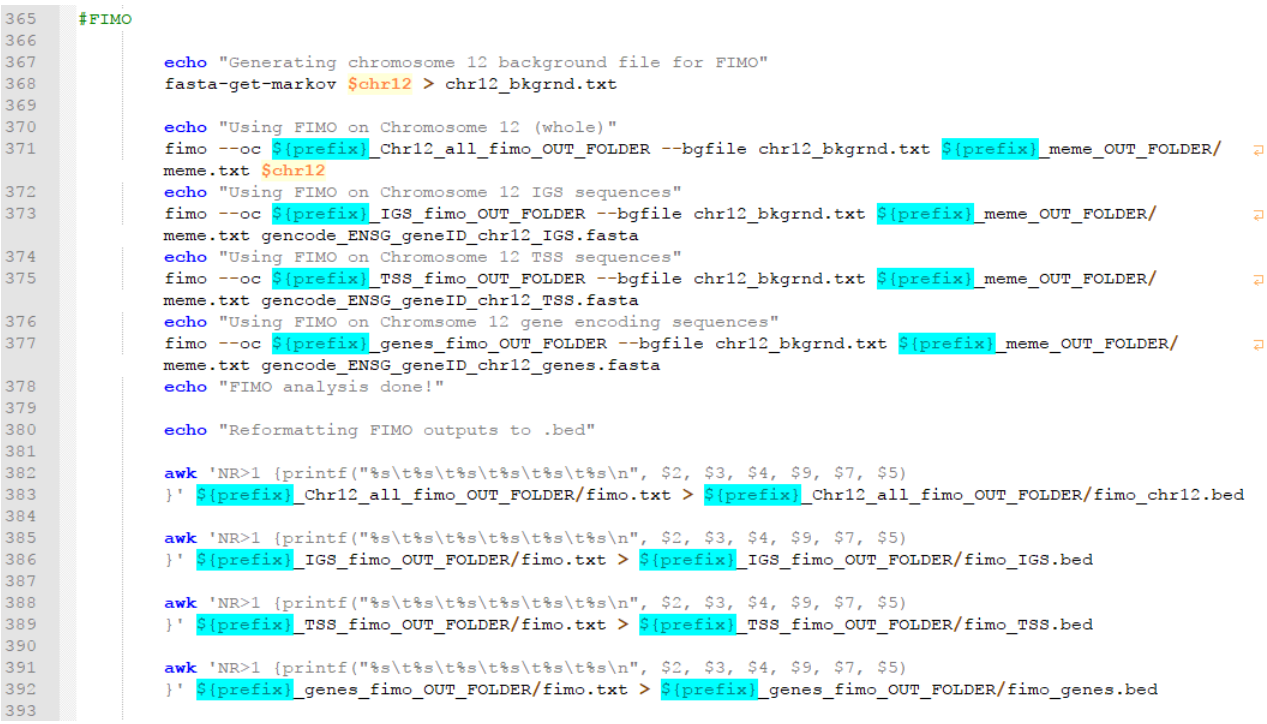


4.2 MAST



```
#=====================================================================================================================================================
# This part generates fasta sequences for the chromosome 12 TSS, genes and IGS and returns how many of each display the motifs identified with MEME
# (this is done using MAST). It also scans the entire chromosome 12 for motif occurrences (regardless of them being in peaks or not, using FIMO)
#=====================================================================================================================================================
# The starting files are the three summits files with peaks in TSS, IGS and genes. They need to be expanded 50 bps each way and then converted to multi fasta


        echo "Examining the motif occurrence within chromosome 12 IGS/TSS/genes summits"
        echo "Generating fasta files for each region"
        bedtools slop -i ${prefix}_chr12_inIGS_nozero.bed -g chr12Info.txt -b 50 > ${prefix}_chr12_inIGS_100bp.bed
        bedtools slop -i ${prefix}_chr12_inTSS_nozero.bed -g chr12Info.txt -b 50 > ${prefix}_chr12_inTSS_100bp.bed
        bedtools slop -i ${prefix}_chr12_ingenes_nozero.bed -g chr12Info.txt -b 50 > ${prefix}_chr12_ingenes_100bp.bed

        bedtools getfasta -name -fi $hg19 -bed ${prefix}_chr12_inIGS_100bp.bed -fo ${prefix}_chr12_inIGS_100bp.fasta
        bedtools getfasta -name -fi $hg19 -bed ${prefix}_chr12_inTSS_100bp.bed -fo ${prefix}_chr12_inTSS_100bp.fasta
        bedtools getfasta -name -fi $hg19 -bed ${prefix}_chr12_ingenes_100bp.bed -fo ${prefix}_chr12_ingenes_100bp.fasta

#MAST syntax for each file:
        echo "Searching for MEME motifs in chromosome 12 peaks"
        mast ${prefix}_meme_OUT_FOLDER/meme.txt ${prefix}_chr12_inIGS_100bp.fasta -oc ${prefix}_IGS_mast_OUT_FOLDER
        mast ${prefix}_meme_OUT_FOLDER/meme.txt -hit_list ${prefix}_chr12_inIGS_100bp.fasta -oc ${prefix}_IGS_mast_OUT_FOLDER > ${prefix}_IGS_mast_OUT_FOLDER/list_mast_hits.txt

        mast ${prefix}_meme_OUT_FOLDER/meme.txt ${prefix}_chr12_inTSS_100bp.fasta -oc ${prefix}_TSS_mast_OUT_FOLDER
        mast ${prefix}_meme_OUT_FOLDER/meme.txt -hit_list ${prefix}_chr12_inTSS_100bp.fasta -oc ${prefix}_TSS_mast_OUT_FOLDER > ${prefix}_TSS_mast_OUT_FOLDER/list_mast_hits.txt

        mast ${prefix}_meme_OUT_FOLDER/meme.txt ${prefix}_chr12_ingenes_100bp.fasta -oc ${prefix}_genes_mast_OUT_FOLDER
        mast ${prefix}_meme_OUT_FOLDER/meme.txt -hit_list ${prefix}_chr12_ingenes_100bp.fasta -oc ${prefix}_genes_mast_OUT_FOLDER > ${prefix}_genes_mast_OUT_FOLDER/list_mast_hits.txt

```

4.3 FIMO



```
#FIMO

	echo "Generating chromosome 12 background file for FIMO"
	fasta-get-markov $chr12 > chr12_bkgrnd.txt

	echo "Using FIMO on Chromosome 12 (whole)"
	fimo --oc ${prefix}_Chr12_all_fimo_OUT_FOLDER --bgfile chr12_bkgrnd.txt ${prefix}_meme_OUT_FOLDER/meme.txt $chr12
	echo "Using FIMO on Chromosome 12 IGS sequences"
	fimo --oc ${prefix}_IGS_fimo_OUT_FOLDER --bgfile chr12_bkgrnd.txt ${prefix}_meme_OUT_FOLDER/meme.txt gencode_ENSG_geneID_chr12_IGS.fasta
	echo "Using FIMO on Chromosome 12 TSS sequences"
	fimo --oc ${prefix}_TSS_fimo_OUT_FOLDER --bgfile chr12_bkgrnd.txt ${prefix}_meme_OUT_FOLDER/meme.txt gencode_ENSG_geneID_chr12_TSS.fasta
	echo "Using FIMO on Chromsome 12 gene encoding sequences"
	fimo --oc ${prefix}_genes_fimo_OUT_FOLDER --bgfile chr12_bkgrnd.txt ${prefix}_meme_OUT_FOLDER/meme.txt gencode_ENSG_geneID_chr12_genes.fasta
	echo "FIMO analysis done!"

	echo "Reformatting FIMO outputs to .bed"

	awk 'NR>1 {printf("%s\t%s\t%s\t%s\t%s\t%s\n", $2, $3, $4, $9, $7, $5)
	}' ${prefix}_Chr12_all_fimo_OUT_FOLDER/fimo.txt > ${prefix}_Chr12_all_fimo_OUT_FOLDER/fimo_chr12.bed

	awk 'NR>1 {printf("%s\t%s\t%s\t%s\t%s\t%s\n", $2, $3, $4, $9, $7, $5)
	}' ${prefix}_IGS_fimo_OUT_FOLDER/fimo.txt > ${prefix}_IGS_fimo_OUT_FOLDER/fimo_IGS.bed

	awk 'NR>1 {printf("%s\t%s\t%s\t%s\t%s\t%s\n", $2, $3, $4, $9, $7, $5)
	}' ${prefix}_TSS_fimo_OUT_FOLDER/fimo.txt > ${prefix}_TSS_fimo_OUT_FOLDER/fimo_TSS.bed

	awk 'NR>1 {printf("%s\t%s\t%s\t%s\t%s\t%s\n", $2, $3, $4, $9, $7, $5)
	}' ${prefix}_genes_fimo_OUT_FOLDER/fimo.txt > ${prefix}_genes_fimo_OUT_FOLDER/fimo_genes.bed

```

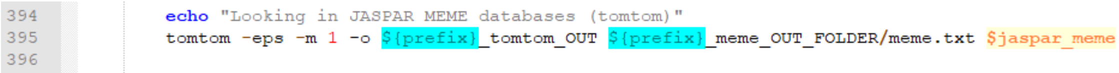
In order to generate Table 2 statistics for FIMO, I used pipes to process the fimo.txt file and to find out out of the two motifs, how many occurrences are for motif 1 and how many are for motif 2, for each treatment. I then checked that the sum of the two is consistent with how many motif occurrences FIMO reports in the html file. E.g.

```
cat fimo.txt | cut -f 1 | grep 1 | wc -l    # for motif 1

cat fimo.txt | cut -f 1 | grep 2 | wc -l # for motif 2
```

Provide an explanation for why not all of your peaks have identified motifs.
As observed in Table 2, only 41.47% of the treatment A peaks display one of the two motifs identified using MEME. Likewise, for treatment A+B, only 33.16% of the peaks have a motif. The reasons why we identify peaks without motif can be the following:
1. According to the chosen MEME options, I looked for the best two motifs only - there could be more than two and the percentages of peaks with a motif might be higher than these numbers.
2. Transcription factors, before any sequence specificity, have DNA binding domains, which causes them to be bound to DNA / chromatin even when not active.
3. Many transcription factors are ligand-dependent, which is a great way to finely modulate the regulation of a set of genes in a ligand concentration dependent manner. Different concentrations of ligands can result in various conformational changes for the transcription factors which can bind different DNA sequences in return.
What do the results tell about the likelihood of the TF finding its motif?
According to the FIMO outcomes (Table 2), there is an overwhelming amount of motif sequences the transcription factor could bind on Chromosome 12. By examining the percentages, for example in treatment A+B, it also appears that the proportion of peaks

found in e.g. intergenic sequences (55.95%) is about the same as the proportion of motifs found in intergenic sequences in the genome (53.98 and 55.83% of motifs found on Chromosome 12 are in intergenic sequences).
This is obviously different from a situation where there would be very few motifs in the genome and the transcription factor would find them against all odds. This transcription factor seems to be *statistically favored to encounter its motifs*, but the abundance of motifs makes it *unlikely that the transcription factor finds its way to perform its function, e.g. upregulate a certain gene by binding to a particular enhancer.* We indeed observe it does not bind very many of these motifs, despite their abundance.
The reasons are plenty - it could be a matter of DNA accessibility (perhaps these motifs are tightly bound by histones which have repressive marks, especially in intergenic sequences), or it could be that the transcription factor generally lies at the end of a signaling cascade and its binding to the targets depends on other effector molecules as well, and not on the DNA sequence alone. This adds another degree of complexity which FIMO cannot account for.

4.4 TomTom



```
    echo "Looking in JASPAR MEME databases (tomtom)"
    tomtom -eps -m 1 -o ${prefix}_tomtom_OUT ${prefix}_meme_OUT_FOLDER/meme.txt $jaspar_meme

```

According to the TomTom search, for treatment A, there are the following candidate transcription factors:

```
       Name MA0258.1                     Name MA0112.1
Alt. Name ESR2                      Alt. Name ESR1
Database jaspar.meme                Database jaspar.meme
p-value 5.63649e-09                 p-value 8.99539e-09
```

For treatment A+B, there are the following transcription factors:

```
       Name MA0112.2          Name MA0258.1          Name MA0066.1
Alt. Name ESR1           Alt. Name ESR2           Alt. Name PPARG
Database jaspar.meme     Database jaspar.meme     Database jaspar.meme
p-value 3.00279e-10      p-value 2.0445e-08       p-value 2.46812e-07
```

5 Final Question: What factor did Billy ChIP, and what were each of the treatments?

From the TomTom hits above, one can confidently say that the antibody Billy did was against the Estrogen Receptor protein ESR1 or ESR2. According to the uniprot database, the following are known about the Estrogen Receptor [3] :

ESR1 binds DNA as a homodimer and it can form heterodimers with ESR2 as well. Binding is followed by a phosphorylation event on both of the monomer subunits.

Generally, ESR is stabilized by phosphorylation (and protected from proteosomal degradation). ESR activity is modulated by signaling pathway kinases that phosphorylate ESR1 as well as its interacting partners.

ESR has three domains - a modulating N-terminal domain, a DNA binding domain (2 zinc fingers) and a C-terminal ligand binding domain. The N-terminal domain can transactivate in a ligand independent manner, while the C-terminal can transactivate in a ligand dependent manner. Transcription is canonically activated through the C-terminal domain by binding of estrogen. As a result of ligand binding, ESR1 associates with a network of coactivators and binds to estrogen responsive elements.

Various mutations of ESR C-terminal domain are associated with disease. Several mutations result in estrogen resistance disease, where the variants have greatly reduced canonical activity in the presence of elevated estrogen levels. The non-classical activity of the mutants (through the estrogen independent domain) is greatly enhanced and this promotes tumor development and progression.[1]

As seen in Table 2, the total number of peaks for treatment A is 1032, while the number of peaks for treatment A+B is 193. An observation described in section 2.3 shows that the peaks in treatment A+B are a subset of treatment A, which means that the addition of B reagent results in a decreased binding of ESR 1 to the responsive elements. All things considered, for Chromsome 12 peaks, this is a 81.2% decrease in binding.

Assuming that the cell lines Billy has used have the wild type estrogen receptor, treatment A could be estrogen itself, while B can be anything that inhibits ESR1.

One popular treatment for breast cancer cells with wild type ESR1 is an aromatase inhibitor. However, this is not a good choice because aromatase uses body levels of androgen hormones as a substrate and only the production of estrogen in a body context would be lowered.

Another candidate for treatment B is tamoxifen, which is a competitor of estrogen in binding to ESR1 and an antagonist. However, tamoxifen binding to ESR1 still results in binding to DNA and repression of transcription, which in a ChIP-seq experiment would not result in differential binding of ESR1 to DNA - the peaks would still come up[2].

It is safe to assume interfering with phosphorylation will result in destabilizing the ESR1 homodimers or ESR1/2 heterodimers that can bind to DNA. Treatment A could be estrogen, while treatment B could be a kinase inhibitor which would show that even in the presence of estrogen, ESR1 cannot bind to the responsive elements. In addition, repressing phosphorylation in general affects the entire signaling pathway and prevents ESR1 interaction with DNA via both ligand dependent and independent domains.

REFERENCES

[1] Jeselsohn R, Buchwalter G, De Angelis C, Brown M, Schiff R (2015) *ESR1 mutations as a mechanism for acquired endocrine resistance in breast cancer* Nat Rev Clin Oncol. 12(10): 573–583.

[2] Wang DY, Fulthorpe R, Liss SN, Edwards EA (2004). *Identification of Estrogen-Responsive Genes by Complementary Deoxyribonucleic Acid Microarray and Characterization of a Novel Early Estrogen-Induced Gene: EEIG1* Molecular Endocrinology, Volume 18, Issue 2, Pages 402–411.

Internet resources:
[3] www.uniprot.org/uniprot/P03372 (Retrieved on Nov. 5, 2017)