STAT 115 Homework 4

Jenna Landy

Due: Tuesday 03/23/2021 by 11:59 pm

Part I. Chromatin Modification (3.5/5 pts)

DNA methylation patterns are altered in many diseases, including cancer, which makes this epigenetic mark an attractive target for various studies. Genome-wide detection of 5mC by bisulfite sequencing is regarded as the current gold standard for DNA methylation detection. In the HW3, we have performed the gene regulation analysis on prostate cancer using transcription factor ChIP-seq data. To better understand the effect of methylation on gene expression, we can utilize BS-seq to detect the DNA methylation positions in the genome.

I.1 (3.5 pts) Reduced-representation bisulfite sequencing (RRBS-Seq) is a technique that uses one or multiple restriction enzymes on the genomic DNA to produce sequence-specific fragmentation. The fragmented genomic DNA is then treated with bisulfite and sequenced. RRBS-Seq is particularly effective on the sites with high methylation, such as in promoters and repeat regions. Given a subsampled RRBS-seq file(`data/bs_subreads.fastq`) in a prostate cancer cell line (LNCaP), perform the reads mapping with bismarker(https://github.com/FelixKrueger/Bismark/tree/master/Docs (https://github.com/FelixKrueger/Bismark/tree/master/Docs)). In this subsampled file, how many reads are unaligned and how many CpGs, CHGs, and CHHs are methylated?

1. (0.5 pt) Make your own working directory in your local home folder. Download bismarker and ‘install’ it https://github.com/FelixKrueger/Bismark#installation (https://github.com/FelixKrueger/Bismark#installation). Note that the latest version is written in perl language, so execute the program by either `/absolute/path/to/the/bismark-*/file` or `perl bismark-*` under the corresponding folder.

```
# install with anaconda
conda install -c bioconda bismark
```

2. Prepare the genome for Bismark. You can start by copying the hg38 from `/n/stat115/2021/HW1/index/hg38_raw/` to your local machine and build from there (1.5 pt graduate students only). One of the main jobs is to conduct the C-T or G-A conversion. Why are we doing this? (0.5 pt, all students)

```
# copy folder to home folder
mkdir hg38_raw
cp /n/stat115/2021/HW1/index/hg38_raw/* hg38_raw/

# run genome preparation
bismark_genome_preparation ./hg38_raw
```

Answer:

```
The samples considered underwent bisulfite conversion, where Uracils took the place of unmethylate
d Cytosines. Through PCR, these uracils became thymadines. This means that the samples will have T
s instead of Cs where the Cs were unmethylated. In their complementary strands, the samples will h
ave As instead of Gs where the Gs complement was unmethylated. Comparing samples to the reference
genome with C-T or G-A conversion lets us know which CpG sites are methylated versus unmethylated.
`bowtie-build` creates a bowtie index from the FASTA files of these G-A and C-T conversion sites.

Genome preparation adds the "Bisulfite_Genome" folder with CT and GA conversion files to the `hg38
_raw` directory. Now we can run bismark!
```

LOG OF `bismark_genome_preparation` : This is the end of the log. The rest of the log is in `bismark.log` , which I have also submitted.

```
Wrote 986164126 bytes to primary EBWT file: BS_CT.rev.1.bt2
Wrote 736462272 bytes to secondary EBWT file: BS_CT.rev.2.bt2
Re-opening _in1 and _in2 as input streams
Returning from Ebwt constructor
Headers:
    len: 2945849067
    bwtLen: 2945849068
    sz: 736462267
    bwtSz: 736462267
    lineRate: 6
    offRate: 4
    offMask: 0xfffffff0
    ftabChars: 10
    eftabLen: 20
    eftabSz: 80
    ftabLen: 1048577
    ftabSz: 4194308
    offsLen: 184115567
    offsSz: 736462268
    lineSz: 64
    sideSz: 64
    sideBwtSz: 48
    sideBwtLen: 192
    numSides: 15342964
    numLines: 15342964
    ebwtTotLen: 981949696
    ebwtTotSz: 981949696
    color: 0
    reverse: 1
```

3. (1 pt) Now that we prepared the genome for the question, follow the Bismark docs to run the main bismark functions, then deduplicate and extract the methylations. For undergraduate students, use directly the converted genome `/n/stat115/2021/HW4/p1_data/hg38/` or here `/n/home13/stat115u2102/hw4/setup/hg38_p1/` . What do you see from `perl bismark2report` ?

```
bismark --genome hg38_raw ../data/bs_subreads.fastq
deduplicate_bismark --bam bs_subreads_bismark_bt2.bam
bismark_methylation_extractor --gzip --bedGraph bs_subreads_bismark_bt2.deduplicated.bam
bismark2report
```

Bismark Report:

```
Bismark report for: ../data/bs_subreads.fastq (version: v0.23.0)
Option '--directional' specified (default mode): alignments to complementary strands (CTOT, CTOB)
were ignored (i.e. not performed)
Bismark was run with Bowtie 2 against the bisulfite genome of /n/home05/bst282u2114/hw4/q1/hg38_ra
w/ with the specified options: -q --score-min L,0,-0.2 --ignore-quals

Final Alignment report
======================
Sequences analysed in total:    10000
Number of alignments with a unique best hit from the different alignments:      4912
Mapping efficiency:     49.1%
Sequences with no alignments under any condition:       4176
Sequences did not map uniquely: 912
Sequences which were discarded because genomic sequence could not be extracted: 0

Number of sequences with unique best (first) alignment came from the bowtie output:
CT/CT:  2462    ((converted) top strand)
CT/GA:  2450    ((converted) bottom strand)
GA/CT:  0       (complementary to (converted) top strand)
GA/GA:  0       (complementary to (converted) bottom strand)

Number of alignments to (merely theoretical) complementary strands being rejected in total:     0

Final Cytosine Methylation Report
=================================
Total number of C's analysed:   40347

Total methylated C's in CpG context:    1369
Total methylated C's in CHG context:    21
Total methylated C's in CHH context:    103
Total methylated C's in Unknown context:        0

Total unmethylated C's in CpG context:  684
Total unmethylated C's in CHG context:  10053
Total unmethylated C's in CHH context:  28117
Total unmethylated C's in Unknown context:      1

C methylated in CpG context:    66.7%
C methylated in CHG context:    0.2%
C methylated in CHH context:    0.4%
C methylated in Unknown context (CN or CHN):    0.0%


Bismark completed in 0d 0h 0m 30s
~
```

Answer:

```
In the `deduplicate_bismark` step, 4 alignments were removed (0.08%), leaving 4908 deduplicated le
ftover sequences. The `bismark2report` step creates a html file as well as a text log file (report
ed below).

To answer the question **how many reads are unaligned**, about half of the sequnces were uniquely
aligned (49.1%), with 41.8% not aligned, and 9.12% had multiple alignments. **4176 sequences had n
o alignments** under any condition, and **912 sequences did not map uniquely**.

To answer the question **how many CpGs, CHGs, and CHHs are methylated**, there were **1369 methyla
ted Cs in CpGs, 21 methylated Cs in CHGs, and 103 methylated Cs in CHHs**. This report tells us th
at almost all methylated cytosines were in a CpG context. The rate of methylation is also greatest
in CpG contexts: bout 66.7% of C's in CpG context were methylated, compared to 0.2% of C's in CHG
contexts and 0.4% of Cs in CHH contexts.
```

I.2 (1.5 pts) Methylation in cytosine at promoter regions normally suppresses the gene expression, while H3K4me3 and H3K27ac histone modifications at promoter regions imply higher gene expression. We have processed RRBS-seq data on the prostater cancer cell line dataset(https://www.encodeproject.org/experiments/ENCSR859PDD/ (https://www.encodeproject.org/experiments/ENCSR859PDD/)) and report the high methylation signal sites in `data/Methylation.bed`. Draw violin plots of the expression level of genes with methylation, H3K4me3, and H3K27ac in their promoters. Could you find that the higher methylation signals repress the gene expression?

The `/n/stat115/2021/HW4/p1_data/ProstateCancer_H3K4me3_peaks.bed` and `/n/stat115/2021/HW4/p1_data/ProstateCancer_H3K27ac_peaks.bed` are the H3K4me3 and H3K27ac peaks files of prostate cancer. Try to find the intersection of loops and histone modification signal intervals. In the `/n/stat115/2021/HW4/p1_data/Expr_loc.txt` , the first three columns are chromosome coordinates, the fourth column is the gene expression score of a prostate cancer tissue, and the rest columns are the gene symbols.

Hint1: Use the preinstalled `bedtools` by calling `module load bedtools2` .

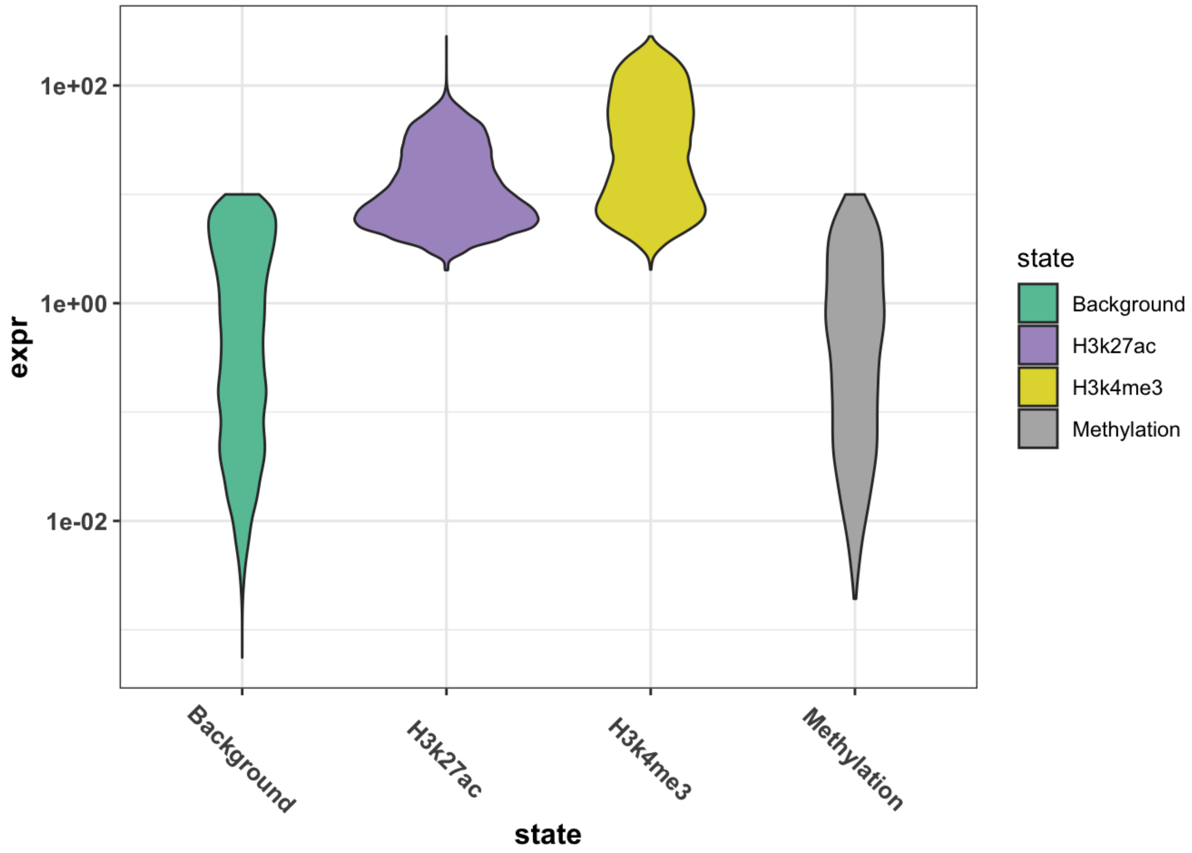
Hint2: Use `>` to write your results into an output file. Use `uniq` command to remove the duplicates from your intersections. e.g. `bedtools intersect {YOUR FLAGS} | uniq > OUTPUT.TXT`

Hint3: The methylation group might have a lower signal-noise ratio. So drop the genes with very low expression (< 0.0001) and use the log scale when you generate the violin plots for better visualization. You can use `ylim` and `scale_y_log10` from `ggplot` .

```
module load bedtools2

bedtools intersect -a ../data/Expr_loc.txt -b ../data/Methylation.bed -wa | uniq > Q1.2_Methyl.txt
```

```
H3k4me3 <- read.table('ProstateCancer_H3K4me3_peaks.bed')
H3k27ac <- read.table('ProstateCancer_H3k27ac_peaks.bed')
Methylation <- read.table('Q1.2_Methyl.txt')
Background <- read.table('Expr_loc.txt')

library(RColorBrewer)
library(ggplot2)
```

```
expr_df <- data.frame(
  expr = c(H3k4me3$V5, H3k27ac$V5, Methylation$V4, Background$V4),
  state = c(rep("H3k4me3", nrow(H3k4me3)),
            rep("H3k27ac", nrow(H3k27ac)),
            rep("Methylation", nrow(Methylation)),
            rep("Background", nrow(Background))
  )
)

getPalette <- colorRampPalette(brewer.pal(8, "Set2"))
ggplot(expr_df) +
  geom_violin(aes(x = state, y = expr, fill = state)) +
  theme_bw() +
  theme(
        axis.text.x = element_text(
          size = 10, face = "bold", hjust = 0.5, vjust = 0.5, angle = -45),
        axis.text.y = element_text(size = 10, face = "bold"),
        axis.title.x = element_text(size = 12, face = "bold"),
        axis.title.y = element_text(size = 12, face = "bold")
    ) +
    scale_fill_manual(values = getPalette(length(unique(expr_df$state)))) +
    ylim(0.0001, 10) +
    scale_y_log10()
```



```
summary(Methylation$V4)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## 0.00000 0.01009 0.15538 1.08271 1.15704 9.98209
```

```
summary(Background$V4)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## 0.00000 0.00000 0.03102 1.06601 0.81727 9.99870
```

Answer:

```
The higher methylation signals have slightly lower expression than the background, on average. The
y also have significantly lower expression than the H3k27ac and H3k4me3 locations.

Yes, it seems that higher methylation signals repress the gene expression.
```

Part II. HiC (5/6 pts)

Genome architecture plays a key role in nuclear functions. The spatial arrangement and proximity of genes have been linked to biological functions, such as gene replication, regulation, and transcription. The Hi-C technique allows researchers to extract the interaction frequency for all loci of a genome at high-throughput and at a genome-wide scale. In this part, we will learn how the HiC data integrates with other epigenetic data and genome architecture affects gene expression in prostate cancer.

II.1 (2 pts) Given subsampled example fastq files generated by Hi-C technique(`/n/stat115/2021/HW4/p2_data/HiC_fq`), perform reads mapping, filtering, and binning to this data with runHiC(http://xiaotaowang.github.io/HiC_pipeline/quickstart.html (http://xiaotaowang.github.io/HiC_pipeline/quickstart.html)). How about the quality of this subsampled data? Could you explain the QC criteria of Hi-C data?**

Instructions:

1. First load the prepared anaconda environment. Make sure use exactly the same version for python and the dependencies. Note the difference from doing the same on local machines, e.g. a laptop or PC.

```
module load Anaconda3 python/3.7.7-fasrc01
conda activate /n/stat115/2021/HW4/conda_env37
module load samtools min
imap2 bwa sratoolkit
```

2. [Optional] The runHiC script takes the bwa index as the input. You can either use the one we used for HW3 `/n/stat115/2021/HW3/bwa_hg38_index` or build your own index separately.

```
I will be using the one we used for homework 3.
```

3. Note that the index file is **NOT*** the same as the one we used for bismark. What is the main difference between bowtie and bwa in practice? (Hint: https://www.biostars.org/p/134418/ (https://www.biostars.org/p/134418/)). Copy the index to your local dir and build the index.

```
Both bowtie and bwa provide implementations of the Burrows-Wheeler algorithm. BWA performs a more
sensitive search than Bowtie by default, and so BWA is slower.
```

4. Prepare your working directory with the index files and your metadata file `datasets.tsv`. See http://xiaotaowang.github.io/HiC_pipeline/quickstart.html#create-the-meta-data-file (http://xiaotaowang.github.io/HiC_pipeline/quickstart.html#create-the-meta-data-file). In our case we would use these following setup configuration. Make sure you have more than one replica sample in your metadata to avoid future ambiguity. If you skipped step 2, copy the prepared index file `/n/stat115/2021/HW4/p2_data/hg38` or `/n/home13/stat115u2102/hw4/setup/hg38_p2` to your wdr directly. Also download `hg38.chrom.sizes` from canvas to your wdr.

```
mkdir setup results
cp /n/stat115/2021/HW4/p2_data/reference_files/hg38.chrom.sizes ./setup
cp -r /n/home13/stat115u2102/hw4/setup/hg38_p2 setup/hg38
cp -r /n/home13/stat115u2102/hw4/setup/HiC_subreads/* setup/HiC_subreads
vim setup/datasets.tsv

# paste following lines into datasets.tsv
# HiC_subreads GM06990 R1 HindIII
# HiC_subreads GM06990 R2 HindIII
```

5. Check your wdr, which should have at least the following folders.

```
.
├── hg38
│   ├── hg38.fa
│   ├── hg38.fa.amb
│   ├── hg38.fa.ann
│   ├── hg38.fa.bwt
│   ├── hg38.fa.fai
│   ├── hg38.fa.pac
│   └── hg38.fa.sa
├── HiC_subreads
│   ├── HiC_subreads
│   ├── HiC_subreads_1.fastq
│   ├── HiC_subreads_2.fastq
│   └── HiC_subreads.completed
├── hg38.chrom.sizes
└── datasets.tsv
```

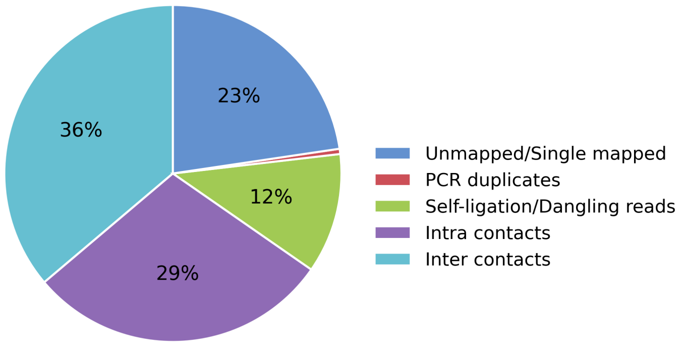
Run the runHiC commands and generate the stats table. Put here the output `*.stats` and plots `*.png` for the summary group `allReps` **ONLY**. What is the message you can read from the results.

```
# running from inside of setup
runHiC pileup -m datasets.tsv -p . -g hg38 -f HiC_subreads -F FASTQ -A bwa-mem -t 10 -O BAM --chro
msizes-file ./hg38.chrom.sizes --logFile ../results/runHiC.log

runHiC quality -m datasets.tsv -L filtered-hg38 --logFile ../results/runHiC.log
```

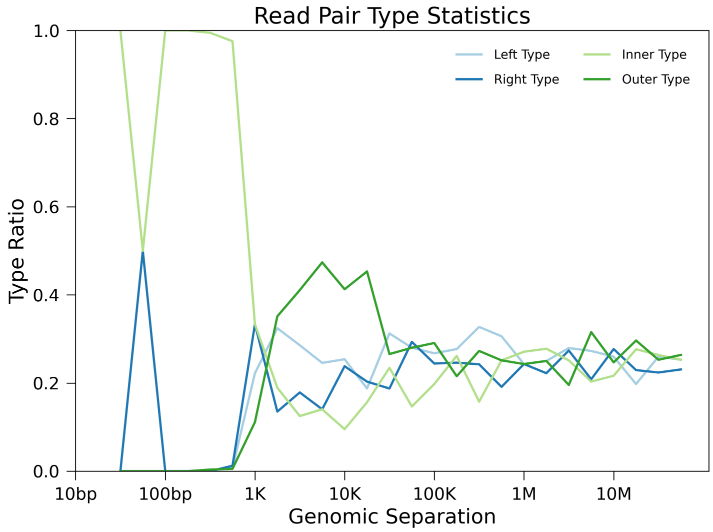
Log file:

```
000_SequencedReads:  30000
    010_DoubleSideMappedReads:  23204
    020_SingleSideMappedReads:  5600
    030_UnmappedReads:  1196
100_NormalPairs:  23204
    110_AfterFilteringReads:  19602
    120_SameFragmentReads:  3452
        122_SelfLigationReads:  468
        124_DanglingReads:  2964
        126_UnknownMechanism:  20
    130_DuplicateRemoved:  150
400_TotalContacts:  19602
    410_IntraChromosomalReads:  8748
        412_IntraLongRangeReads(>=20Kb):  5738
        412_IntraShortRangeReads(<20Kb):  3010
    420_InterChromosomalReads:  10854

Critical Indicators:
Double Unique Mapped Ratio = 23204 / 30000 = 0.7735
Self-Ligation Ratio = 468 / 30000 = 0.0156
Dangling-Reads Ratio = 2964 / 30000 = 0.0988
Long-Range Ratio = 5738 / 19602 = 0.2927
Data Usage = 19602 / 30000 = 0.6534
```

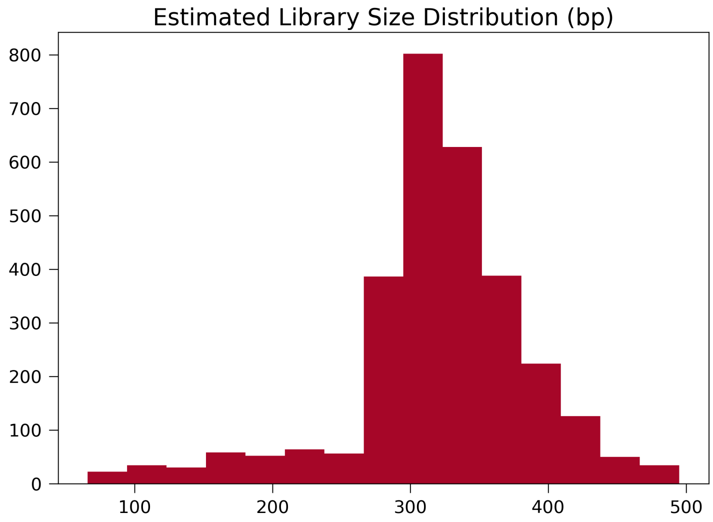
```
library(knitr)
knitr::include_graphics('stats.png')
```



```
knitr::include_graphics('pairtypes.png')
```



```
knitr::include_graphics('librarysize.png')
```



Answer:

```
The statistics plot shows that most contacts were ineterchromosomal, followed by intrachromosomal
contacts. There were much fewer self-ligation / dangling reads.

The pairtypes plot shows that with larger separation, the types seem to stabilize around 25%. We s
hould only consider contacts with genomic distances greater than 50Kb, because that is where the d
istribution starts to be relatively constant.

The library size plot looks very similar to the distribution of a typical 300-500bp library.
```

II.2 (2 pts) In the II.1, you have learned how to generate a genomic interaction matrix on the example file. Here we provided a genomic interaction data on chr21 in prostate cancer (`data/chr21.chr21_10000.cool`). Normalize this data at 10k resolution and perform loop calling with 10% CTCF model. How many loops with >0.9 confidence can you find? Then draw a genomic contact heatmap to show the interaction in chr21. Are there any highly interactive regions?

Hint: cooler(https://cooler.readthedocs.io/en/latest/ (https://cooler.readthedocs.io/en/latest/)) can perform normalization and peakachu(https://github.com/tariks/peakachu (https://github.com/tariks/peakachu)) can perform loop calling on Hi-C data. higlass(https://higlass.io/ (https://higlass.io/)) may help with visualization.

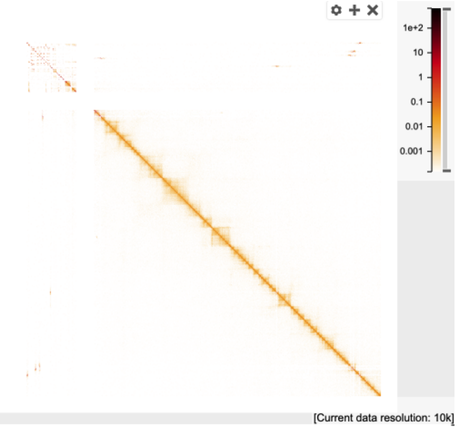
```
# load cool files and normalization
cooler balance ../data/chr21.chr21_10000.cool --max-iters 1000--force

# resolutions 10000 normalizes at 10k
hicConvertFormat --matrices ../data/chr21.chr21_10000.cool --inputFormat cool --outputFormat mcool
--outFileName chr21.chr21_10000.mcool --resolutions 10000
```

```
peakachu score_genome -r 10000 --balance -p ../data/chr21.chr21_10000.cool -m /n/stat115/2021/HW4/
p2_data/down10.ctcf.pkl -O .

peakachu pool -i chr21.bed -t 0.9 > chr21.loops.bedpe

cat chr21.loops.bedpe | wc -l # returns 78
```

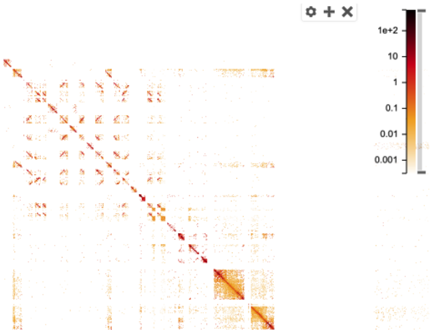
```
python3 -m pip install higlass-python
jupyter nbextension enable --py --sys-prefix widgetsnbextension
jupyter nbextension install --py --sys-prefix higlass
jupyter nbextension enable --py --sys-prefix higlass
jupyter notebook
```

```
%load_ext autoreload
%autoreload 2

import higlass
from higlass.client import Track, View

tsl = higlass.tilesets.cooler('chr21.chr21_10000.mcool')
srl = Track('heatmap', tileset = tsl)
view1 = View([srl])
display, server, viewconf = higlass.display([view1])
display
```

```
knitr::include_graphics("higlass-full.png")
```



```
knitr::include_graphics("higlass-topleft.png")
```



Answer:

```
There are 78 loops with over 0.9 confidence.

Highly interactive regions are marked in darker red in the higlass plot. These regions seem to be
concentrated in the top left corner (zoomed in in the second image), as well as along the diagona
l. This indicates that distanced interactions may be more frequent towards the start of the chromo
some (top left), and that most interactions outside of that region are between sections very close
together (nar the diagonal).
```

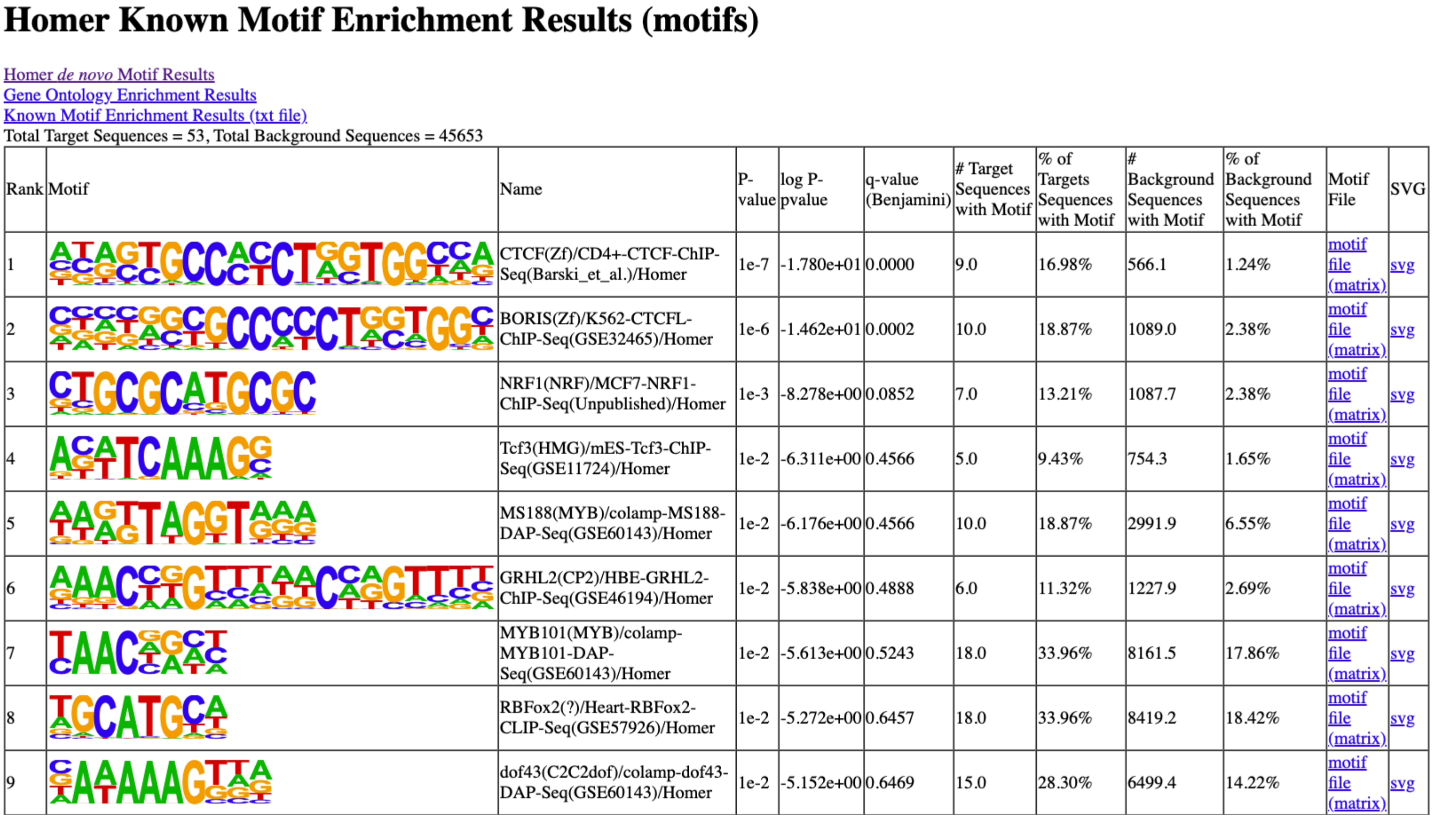
II.3 (1 pt) Transcription factors help construct the chromatin loops. With the provided prostate cancer ATAC-seq peaks file(`data/tumor_ATAC_peaks.bed`), could you find the open regions in the loop anchors? What factors bind in the loop anchors? What potential roles may these factors play?

ATAC-seq identifies open DNA regions. The `bedpe` output from the previous question provides anchor locations. To find open regions in the loop anchors, I intersect the ATAC-seq output with the locations in the `bedpe` file.

```
awk 'BEGIN {OFS="\t"}{print $1,$2,$3}' chr21.loops.bedpe > anchor1.txt
awk 'BEGIN {OFS="\t"}{print $4,$5,$6}' chr21.loops.bedpe > anchor2.txt
cat anchor1.txt anchor2.txt > anchor.txt

bedtools intersect -a tumor_ATAC_peaks.bed -b anchor.txt | uniq > intersect.bed
cat intersect.bed | wc -l # returns 53
```

```
source /n/stat115/2021/HW4/miniconda3/bin/activate
conda activate /n/stat115/2021/HW4/homer_env
findMotifsGenome.pl intersect.bed setup/hg38/hg38.fa motifs
```

```
knitr::include_graphics('motifs.png')
```



Homer Known Motif Enrichment Results (motifs)

Homer *de novo* Motif Results
Gene Ontology Enrichment Results
Known Motif Enrichment Results (txt file)
Total Target Sequences = 53, Total Background Sequences = 45653

Rank	Motif	Name	P-value	log P-pvalue	q-value (Benjamini)	# Target Sequences with Motif	% of Targets Sequences with Motif	# Background Sequences with Motif	% of Background Sequences with Motif	Motif File	SVG
1		CTCF(Zf)/CD4+-CTCF-ChIP-Seq(Barski_et_al.)/Homer	1e-7	-1.780e+01	0.0000	9.0	16.98%	566.1	1.24%	motif file (matrix)	svg
2		BORIS(Zf)/K562-CTCFL-ChIP-Seq(GSE32465)/Homer	1e-6	-1.462e+01	0.0002	10.0	18.87%	1089.0	2.38%	motif file (matrix)	svg
3		NRF1(NRF)/MCF7-NRF1-ChIP-Seq(Unpublished)/Homer	1e-3	-8.278e+00	0.0852	7.0	13.21%	1087.7	2.38%	motif file (matrix)	svg
4		Tcf3(HMG)/mES-Tcf3-ChIP-Seq(GSE11724)/Homer	1e-2	-6.311e+00	0.4566	5.0	9.43%	754.3	1.65%	motif file (matrix)	svg
5		MS188(MYB)/colamp-MS188-DAP-Seq(GSE60143)/Homer	1e-2	-6.176e+00	0.4566	10.0	18.87%	2991.9	6.55%	motif file (matrix)	svg
6		GRHL2(CP2)/HBE-GRHL2-ChIP-Seq(GSE46194)/Homer	1e-2	-5.838e+00	0.4888	6.0	11.32%	1227.9	2.69%	motif file (matrix)	svg
7		MYB101(MYB)/colamp-MYB101-DAP-Seq(GSE60143)/Homer	1e-2	-5.613e+00	0.5243	18.0	33.96%	8161.5	17.86%	motif file (matrix)	svg
8		RBFox2(?)/Heart-RBFox2-CLIP-Seq(GSE57926)/Homer	1e-2	-5.272e+00	0.6457	18.0	33.96%	8419.2	18.42%	motif file (matrix)	svg
9		dof43(C2C2dof)/colamp-dof43-DAP-Seq(GSE60143)/Homer	1e-2	-5.152e+00	0.6469	15.0	28.30%	6499.4	14.22%	motif file (matrix)	svg

Answer:

```
The first bedtools intersection shows that there are 53 open regions in the loop anchors. To find
what factors bind in the loop anchors, I ran `findMotifsGenome.pl` to identify the motifs within t
he open regions in the loop anchors.

The findMotifsGenome function shows that the top 9 known Motifs are in the screenshot above. There
were many de novo motif results as well, but all were marked as possible false positive and we ca
n't look up their functions because they are unknown.

To find what role these factors play, I looked through the [Homor Motifs](http://homer.ucsd.edu/ho
mer/motif/HomerMotifDB/homerResults.html) info page for each of the top 3 motifs.

CTCF(Zf)/CD4+-CTCF-ChIP-Seq(Barski_et_al.)/Homer(1.000) and BORIS(Zf)/K562-CTCFL-ChIP-Seq(GSE3246
5)/Homer(1.000) are in the same family. According to gene cards, this is a "member of the BORIS +
CTCF gene family... Mutations in this gene have been associated with invasive breast cancers, pros
tate cancers, and Wilms' tumors."

The NRF1(NRF)/MCF7-NRF1-ChIP-Seq(Unpublished)/Homer(1.000) gene encodes a protein that functions a
s a transcription factor for "some key metabolic genes regulating cellular growth and nuclear gene
s required for respiration, heme biosynthesis, and mitochondrial DNA transcription and replicatio
n". It has been linked to Skin Carcinoma In Situ and Oncocytoma.
```

II.4 (1 pt, graduates only) Normally histone modification H3K27ac marks the enhancer regions on the genome, and H3K4me3 marks the promoter regions. Use the prostate tumor H3K27ac and H3K4me3 peaks files in I.2 here. Based on the loop file, could you find the loops that contact an enhancer region with a gene promoter region on chr21? Do these target genes express higher than the genes without loops structure?

Method 1: find loops where one anchor is an enhancer and the other is a promoter

```
bedtools intersect -c -a anchor1.txt -b /n/stat115/2021/HW4/p1_data/ProstateCancer_H3K27ac_peaks.b
ed > left_enhancers.bed
bedtools intersect -c -a anchor1.txt -b /n/stat115/2021/HW4/p1_data/ProstateCancer_H3K4me3_peaks.b
ed > left_promoters.bed

bedtools intersect -c -a anchor2.txt -b /n/stat115/2021/HW4/p1_data/ProstateCancer_H3K27ac_peaks.b
ed > right_enhancers.bed
bedtools intersect -c -a anchor2.txt -b /n/stat115/2021/HW4/p1_data/ProstateCancer_H3K4me3_peaks.b
ed > right_promoters.bed
```

```
left_promoters <- read.table('left_promoters.bed')
left_enhancers <- read.table('left_enhancers.bed')
right_promoters <- read.table('right_promoters.bed')
right_enhancers <- read.table('right_enhancers.bed')
loops <- read.table('chr21.loops.bedpe')

out <- ((left_promoters$V4 > 0) & (right_enhancers$V4 > 0)) | ((left_enhancers$V4 > 0) & (right_pr
omoters$V4 > 0))

loops[out,]
```

```
##       V1       V2       V3    V4       V5       V6        V7
## 17 chr21 39180000 39190000 chr21 39380000 39390000 0.9328525
```

Answer:

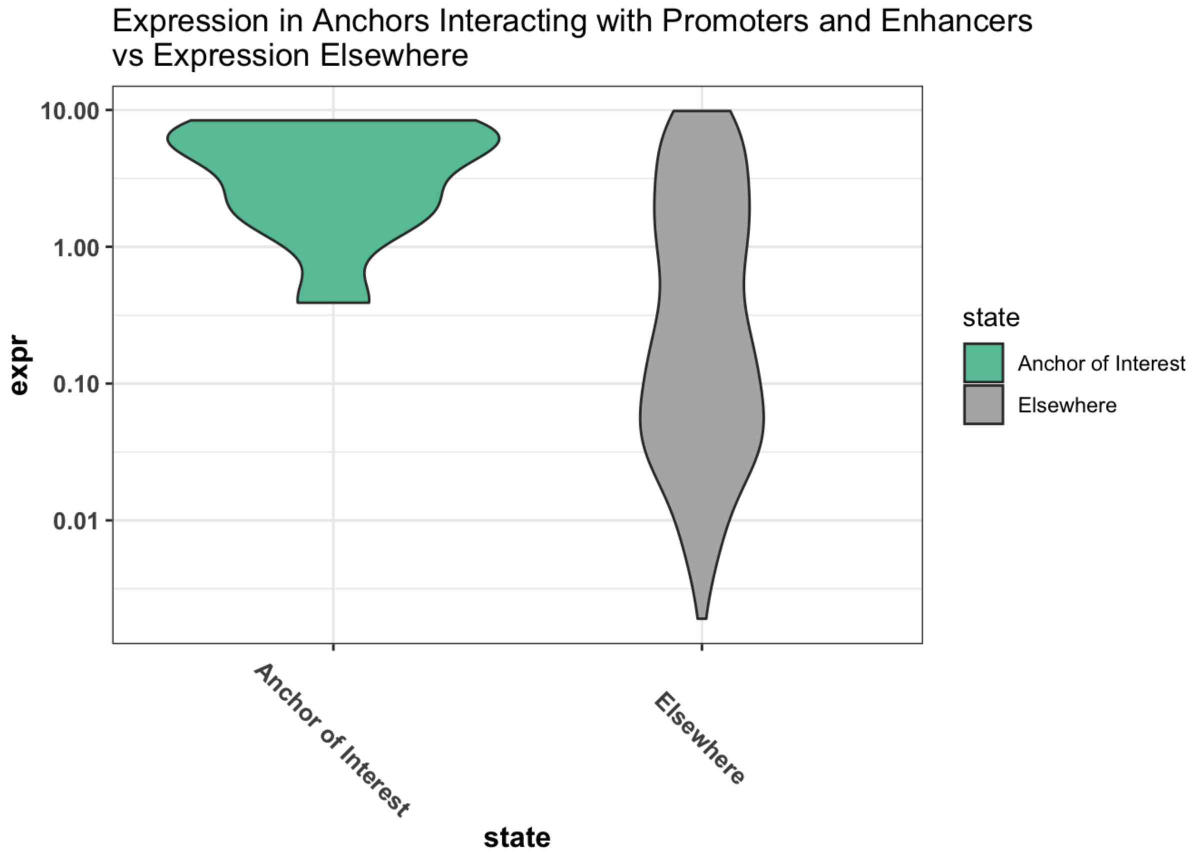
```
We want to find loops where one anchor is an enhancer and the other is a promoter. First, I found
which anchors are enhancers and which anchors are promoters, and I kept track of which anchors are
on the left or the right (with respec to the structure of the `bedpe` file).

Using the `-c` flag with `bedtools intersect` means each entry in A will be in the resulting file,
and a new column will be created that reports the number of overlaps with each B file on a distinc
t line. This reports 0 for A entries that have no overlap with B. Note that because I'm using `anc
hor1.txt` and `anchor2.txt`, which represent the left and right sides of the loops, and are lined
up in order. Because all rows are retained with the `-c` flag, the resulting files will still line
up.

Our goal is to either have a left enhancer line up with a right promoter, or a left promoter line
up with a right enhancer. The new row in each of these four files, row 4, tells us whether the anc
hor is what we are looking for (value >= 1) or not (value = 0).

With this method, there is only one loop that that contact an enhancer region with a gene promoter
region on chr21. It has the locations shown above.
```

Method 2: Find anchor regions that overlap with a promoter section and an enhancer section

```
# another option mentioned in the slack is to consider anchor regions that overlap with a promoter
section and an enhancer section
# directly bedtools between the concatenated anchor.txt with each bed file and find the intersecti
on afterwards

bedtools intersect -wa -a anchor.txt -b /n/stat115/2021/HW4/p1_data/ProstateCancer_H3K4me3_peaks.b
ed | uniq > promoters.bed
bedtools intersect -wa -a anchor.txt -b /n/stat115/2021/HW4/p1_data/ProstateCancer_H3K27ac_peaks.b
ed | uniq > enhancers.bed

bedtools intersect -a promoters.bed -b enhancers.bed | uniq > promoters_enhancers.bed

wc -l promoters_enhancers.bed # returns 14
```

```
bedtools intersect -wa -a ../data/Expr_loc_chr21.txt -b promoters_enhancers.bed > loop_expr.txt
bedtools subtract -a ../data/Expr_loc_chr21.txt -b promoters_enhancers.bed > non_loop_expr.txt
```

```
loop_expr <- read.table('loop_expr.txt')
non_loop_expr <- read.table('non_loop_expr.txt')

expr_df <- data.frame(
  expr = c(loop_expr$V4, non_loop_expr$V4),
  state = c(rep("Anchor of Interest", nrow(loop_expr)),
            rep("Elsewhere", nrow(non_loop_expr))
  )
)

getPalette <- colorRampPalette(brewer.pal(8, "Set2"))
ggplot(expr_df) +
  geom_violin(aes(x = state, y = expr, fill = state)) +
  theme_bw() +
  theme(
        axis.text.x = element_text(
          size = 10, face = "bold", hjust = 0.5, vjust = 0.5, angle = -45),
        axis.text.y = element_text(size = 10, face = "bold"),
        axis.title.x = element_text(size = 12, face = "bold"),
        axis.title.y = element_text(size = 12, face = "bold")
    ) +
    scale_fill_manual(values = getPalette(length(unique(expr_df$state)))) +
    ylim(0.0001, 10) +
    scale_y_log10() +
  ggtitle('Expression in Anchors Interacting with Promoters and Enhancers\nvs Expression Elsewher
e')
```



Answer:

```
Here, we want to find anchors that overlap with both an enhancer and promoter section. First, I fo
und the anchors that overlap with an enhancer, and separately the anchors that overlap with a prom
oter. Then, I intersected the two lists of anchors to find regions of interest.

With this method, there are 14 anchors that overlap with both a promoter and an enhancer.

To determine **whether these target genes express higher than the genes without loops structure**,
we need the expression for regions in the 14 anchors of interest, as well as the expression for ev
erything besides those 14 anchors of interest. For the first, I intersected expression with the an
chors of interest. For the second, I subtracted the anchors file from the expression file.

There were 11 genes overlapping with the 14 anchors that interact with a promoter and enhancer.

The plot shows that genes in the anchors that have interactions with promoters and enhancers have
higher expression levels than genes elsewhere on chromosome 21.
```

Part III. Hidden Markov Model and TAD boundaries (2 pts)

Topologically associating domains (TADs) define genomic intervals, where sequences within a TAD physically interact more frequently with each other than with sequences outside the TAD. TADs are often defined by HiC (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3149993/ (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3149993/)), an experimental technique designed to study the three-dimensional architecture of genomes. HiC generates PE sequenced data, where the two mate pairs indicate two genomic regions that are might be far apart in the genome, but physically interact with each other. If we look across the genome in bins (40kb in the early paper, but now can go down to 5-10kb with deeper sequencing), we could find reads that are mapped there and check whether their interacting mate pairs are mapped upstream or downstream. In each bin, we can calculate a directional index (DI) to quantify the degree of upstream or downstream bias of a given bin (for more details, see the supplement- `Supplement_10.1038_nature11082.pdf`). For this HW, we ask you to implement a hidden Markov Model (Viterbi) to find regions with upstream bias (DI < 0) and those with downstream bias (DI > 0), even though the DI in individual bins might have some noise. This way, TAD boundaries could be discovered as clusters of bins from negative DIs to positive DIs (see Supplementary Figure 12b).

For simplicity, we will only have two hidden states (upstream, downstream), and use the following HMM parameters (these do not necessarily capture the real data distribution, but just to help your implementation):

```
Initial probability: upstream = 0.5, downstream = 0.5
Transition probability: Pb(up to up) = Pb(dn to dn) = 0.9, Pb(up to dn) = Pb(dn to up) = 0.1

Emission probabilities:
P{<-1200, [-1200,-800), [-800,-500), [-500,0), [0,500), [-500,800), [800, 1200), >= 1200 | upstrea
m} = (0.01, 0.01, 0.02, 0.04, 0.65, 0.15, 0.08, 0.04)
P{<-1200, [-1200,-800), [-800,-500), [-500,0), [0,500), [-500,800), [800, 1200), >= 1200 | downstr
eam} = (0.04, 0.08, 0.15, 0.65, 0.04, 0.02, 0.01, 0.01)
```

Given the DI file (`data/ESC.Dixon_2015.DI.chr21.txt`), implement and utilize the Viterbi algorithm to predict the hidden states of the Hi-C data. Visualize your result with a graph utilizing the following: midpoint of genomic bin on the x-axis; DI score per bin on the y-axis; color: hidden state of the HMM.

Hint1: Examples HMM code can be found at: http://www.adeveloperdiary.com/data-science/machine-learning/implement-viterbi-algorithm-in-hidden-markov-model-using-python-and-r/ (http://www.adeveloperdiary.com/data-science/machine-learning/implement-viterbi-algorithm-in-hidden-markov-model-using-python-and-r/)

Hint2: The observations are continuous or have too many discrete values. Try binning them into a few discrete regions. Use `cut` function built in R.

```
Viterbi=function(v,a,b,initial_distribution) {

  T = length(v)
  M = nrow(a)
  prev = matrix(0, T-1, M)
  omega = matrix(0, M, T)

  omega[, 1] = log(initial_distribution * b[, v[1]])
  for(t in 2:T){
    for(s in 1:M) {
      probs = omega[, t - 1] + log(a[, s]) + log(b[s, v[t]])
      prev[t - 1, s] = which.max(probs)
      omega[s, t] = max(probs)
    }
  }

  S = rep(0, T)
  last_state=which.max(omega[,ncol(omega)])
  S[1]=last_state

  j=2
  for(i in (T-1):1){
    S[j]=prev[i,last_state]
    last_state=prev[i,last_state]
    j=j+1
  }

  S[which(S==1)]='A'
  S[which(S==2)]='B'

  S=rev(S)

  return(S)

}
```

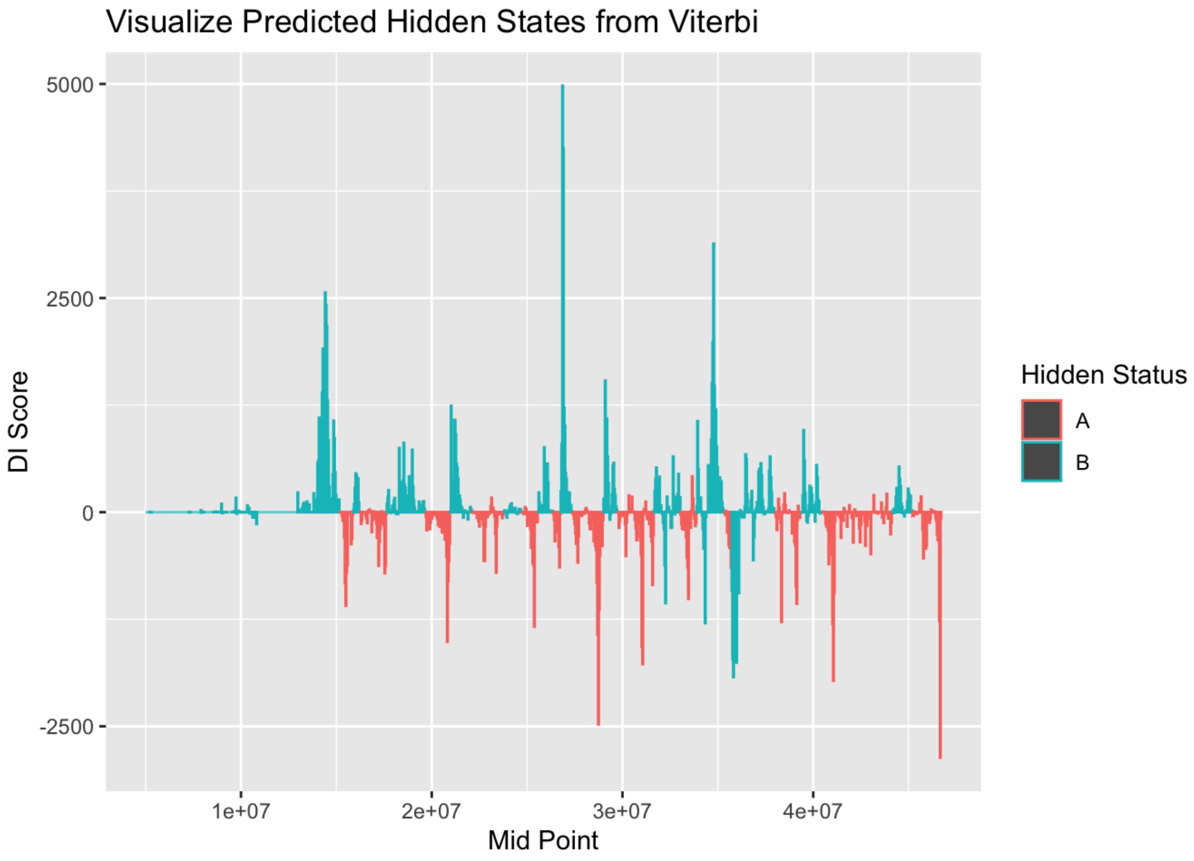
```
dat <- read.table('data/ESC.Dixon_2015.DI.chr21.txt')
dat$mid <- (dat$V3 + dat$V2)/2
dat$V4_discrete <- cut(dat$V4, c(min(dat$V4)-1, -1200, -800, -500, 0, 500, 800, 1200, max(dat$V4)
 + 1), right = FALSE)
table(dat$V4_discrete)
```

```
## 
## [-2.87e+03,-1.2e+03)      [-1.2e+03,-800)          [-800,-500) 
##                   16                   16                   24 
##             [-500,0)              [0,500)            [500,800) 
##                  392                  534                   28 
##        [800,1.2e+03)   [1.2e+03,4.98e+03) 
##                   17                   16
```

```
pr_upstream <- c(0.01, 0.01, 0.02, 0.04, 0.65, 0.15, 0.08, 0.04)
pr_downstream <- c(0.04, 0.08, 0.15, 0.65, 0.04, 0.02, 0.01, 0.01)
initial_distribution = c(1/2, 1/2)
```

```
M = 2
K = length(unique(dat$V4_discrete)) # = 8 bins
A = matrix(c(0.9,0.1,0.1,0.9), M, M)
B = matrix(c(pr_upstream, pr_downstream), M, K)
initial_distribution = c(1/2, 1/2)

myout = Viterbi(
  dat$V4_discrete, A, B,
  initial_distribution
)
```

```
plot_dat <- data.frame(
  'mid' = dat$mid,
  'DI' = dat$V4,
  'state' = myout
)

ggplot(plot_dat, aes(x = mid, y = DI, color = as.factor(state))) +
  geom_col() +
  ggtitle("Visualize Predicted Hidden States from Viterbi") +
  xlab("Mid Point") +
  ylab("DI Score") +
  labs(color = "Hidden Status")
```



Part IV: GWAS Followup (3 pts)

The NHGRI-EBI GWAS Catalog is a curated dataset of trait-associated genetic variants for human. While it provides association between single-nucleotide polymorphisms (SNPs) and trait (i.e. cancer), the genetic variants in GWAS catalog are not necessarily causative or functional for a trait, since SNPs can be highly correlated measured by linkage disequilibrium

(LD). To learn the potential functional effect of a certain SNP, especially the non-coding variants, we can use RegulomeDB to explore the potential function of the SNP.

You will explore the following online resources: The NHGRI-EBI GWAS catalog (https://www.ebi.ac.uk/gwas/ (https://www.ebi.ac.uk/gwas/)), dbSNP (https://www.ncbi.nlm.nih.gov/snp/ (https://www.ncbi.nlm.nih.gov/snp/)), LDLink (https://ldlink.nci.nih.gov/ (https://ldlink.nci.nih.gov/)), and RegulomeDB (the beta version http://regulomedb.org (http://regulomedb.org) or the more stable older version http://legacy.regulomedb.org/ (http://legacy.regulomedb.org/)).

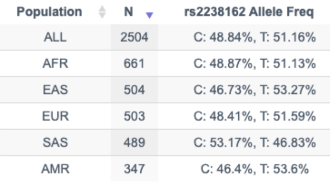
IV.1 Explore whether there are genetic variants within the gene BRCA2 which are associated with any traits. What traits are associated with the BRCA2 variants? Which SNP has the smallest p-value related to breast cancer? What is the risk allele?

```
On the GWAS catalog result for Gene: BRCA2 (https://www.ebi.ac.uk/gwas/genes/BRCA2), there were 26
studies that found 29 associations with BRCA2. 14 specific traits were reported, including LDL cho
lesterol, multiple types of breast carcinoma, generic cancer, and multiple types of lung cancer.

Looking at the list of associations, the variant with the smallest p-value is rs2238162, and the r
isk allele is C, with a p-value of 3 x 10E-27.
```

IV.2 For the BRCA2 SNP with the most significant association with breast cancer, what consequence does the risk allele have on the BRCA2 protein sequence? Based on 1000 Genomes in LDLink, what is the allele frequency of the risk allele among the 5 ethnicities. In the population with the highest risk in the resource, what is the expected number of people with heterozygous genotype at this SNP, assuming linkage disequilibrium?

```
The most severe consequence is a noncoding transcript exon variant.

The allel frequencies for the five ethnicities are in the table below.
```

```
knitr::include_graphics("allele_freq.png")
```



Population	N	rs2238162 Allele Freq
ALL	2504	C: 48.84%, T: 51.16%
AFR	661	C: 48.87%, T: 51.13%
EAS	504	C: 46.73%, T: 53.27%
EUR	503	C: 48.41%, T: 51.59%
SAS	489	C: 53.17%, T: 46.83%
AMR	347	C: 46.4%, T: 53.6%

```
The population with the highest risk is SAS, south asian. This is because the variant that has an
association with breast cancer has a C in the specific location of the mutation, and south asians
have the highest C allele frequency.

The expected number of people with the heterozygous genotype at this SNP, assuming linkage disequi
librium, is n*2*p*q = 489*2*(0.5317)*(0.4683) = 243.5, using the hardy weinberg equation.
```

IV.3 Explore a certain SNP, rs4784227, that was reported to be associated with breast cancer. Is it an intergenic, exonic or intronic variant? What gene does it fall in?

```
rs4784227 is an intronic variant in the CASC16 gene.
```

IV.4 Explore the SNP rs4784227 in RegulomeDB. What functional category does the rank score (or Regulome DB Score) implicate? What factors does RegulomeDB take into consideration while scoring the potential function of SNPs?

```
The rank score is 2b, which corresponds to "likely to affect binding". To calculate this score, Re
gulomeDB takes into account what kind of supporting data is available for a single coordinate. The
se data types include eQTL, TF binding, matched TF motif, matched DNase Footprint, DNase peak, et
c.
```

IV.5 Describe the evidence that implicates the regulatory potential of rs4784227, for example, list several transcription factors with binding peaks overlapping this SNP; report the cell types with open chromatin regions overlapping this SNP.

```
Specifically for this score, 2b, a SNP must have TF binding, any motif, DNase Footprint, and DNase
peak.

A transcription factor with binding peaks overlapping rs4784227 include FOXA1 (HepG2 cell type), T
CF12 (A549 cell type), and TEAD4 (HepG2 cell type).
```

IV.6 [Graduate Students] Read the paper by Cowper-Sal et al. (PMID 23001124) and summarize the potential mechanisms of the above SNP's function in terms of affecting transcription factor-DNA interaction and regulating genes.

```
TOX3 expression decreases with the risk allele T for the SNP rs4784227.

The SNP rs4784227 is in an enhancer region for the TOX3 gene. The SNP also overlaps a motif within
a FOXA1 binding site. With the T risk allele as opposed to the reference C allele, FOXA1 binding i
s more common. The T risk allele is also associated with increased binding of TLE proteins over th
e reference C allele. Even though FOXA2 typically increases gene expression (as it is a transcript
ion factor), co-binding with TLE proteins causes a decrease in transcription.
```