seqOutBias: Universal correction of enzymatic sequence bias

Michael Guertin

June 7, 2017

A step-by-step guide to correcting sequence biases resulting from enzymatic sequence preferences using seqOutBias. [1]

Contents

[1] address questions to Michael Guertin, guertin@virginia.edu

List of Figures

1 Processing of DNase-seq data from ENCODE

Enzymatic digestion sequence preference was characterized on the genome-wide scale in a joint publication from Shirley Liu's and Myles Brown's groups using DNase-seq data (He *et al.*, 2014). We use publicly available DNase-seq data from ENCODE (Stamatoyannopoulos Lab) and data deposited into GEO (Lazarovici *et al.*, 2013) as examples of how to use `seqOutBias` to correct enzymatic accessibility data.

1.1 Retrieving raw data from ENCODE

Download the fastq files directly from ENCODE (`http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeUwDnase/`).
Use `wget` to retrieve the raw fastq files from ENCODE, `tar` the downloaded files, combine all the replicates for each condition and compress the resultant file. Note that we are combining all the data sets for the purpose of having more sequencing depth of coverage and for ease of analysis downstream. These files result from DNase-nicking of crude nuclei isolations, so peaks represent regions of open chromatin *in vivo*. A recent comprehensive review of outlines the molecular biology details of DNase-seq (Vierstra and Stamatoyannopoulos, 2016). It is noteworthy that DNase does not cleave double stranded DNA, instead DNase nicks the phosphodiester backbone of DNA and four nicking events are needed to detect a DNA fragment by DNase-seq, although only two nicking events are detected per fragment (Vierstra and Stamatoyannopoulos, 2016; Thomas, 1956).

```
mkdir ~/DNase_ENCODE
cd ~/DNase_ENCODE
wget http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeUwDnase/wgEncodeUwDnaseMcf7Est100nm1hRawDataRep1.fastq.tgz
wget http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeUwDnase/wgEncodeUwDnaseMcf7Est100nm1hRawDataRep2.fastq.tgz
wget http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeUwDnase/wgEncodeUwDnaseMcf7Estctrl0hRawDataRep1.fastq.tgz
wget http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeUwDnase/wgEncodeUwDnaseMcf7Estctrl0hRawDataRep2.fastq.tgz
tar -xvf wgEncodeUwDnaseMcf7Estctrl0hRawDataRep2.fastq.tgz
tar -xvf wgEncodeUwDnaseMcf7Estctrl0hRawDataRep1.fastq.tgz
tar -xvf wgEncodeUwDnaseMcf7Est100nm1hRawDataRep2.fastq.tgz
tar -xvf wgEncodeUwDnaseMcf7Est100nm1hRawDataRep1.fastq.tgz
cat UwStam_MCF7-*fastq > UW_MCF7_both.fastq
gzip UW_MCF7_both.fastq
rm *.fastq.tgz
rm *fastq
```

Download short read archive data set (SRA accession SRX247626) of DNase-seq data from DNA purfied from IMR90 cells (Lazarovici *et al.*, 2013), convert the *sra* to a *fastq* file using `fastq-dump` (herein we use version: 2.7.0), change the name to be descriptive, and compress the file. Note that this is DNase-seq data from naked DNA digestion (i.e. no bound proteins), which provided the most compeling evidence that DNase signatures at the site of transcription factor (TF) binding are not a result of protein binding (He *et al.*, 2014).

```
wget ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByRun/sra/SRR/SRR769/SRR769954/SRR769954.sra
fastq-dump SRR769954.sra
mv SRR769954.fastq IMR90_Naked_DNase.fastq
gzip IMR90_Naked_DNase.fastq
rm *sra
```

1.2 Index the Appropriate Genome File

Retrieve the relevant genome from UCSC (Karolchik *et al.*, 2014), we will use the latest assembly, hg38. This is a zipped *fasta* file of the entire human genome. Bowtie 2 is an efficient tool for aligning sequencing reads to long reference sequences. For this execution we used Bowtie2 version 2.2.6. The first task is to build the genome index with bowtie2 (Langmead *et al.*, 2009) `http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml#the-bowtie2-build-indexer`. This only has to be performed once per genome.

```
wget http://hgdownload.soe.ucsc.edu/goldenPath/hg38/bigZips/hg38.fa.gz
gunzip hg38.fa.gz
bowtie2-build hg38.fa hg38
```

1.3 Align *fastq.gz* files with bowtie2

This code will loop through files in a directory. The file name is split on the ’.fastq.gz’ string and the variable ’name’ is assigned to the first string.

```
for fq in *.fastq.gz
do
  name=$(echo $fq | awk -F".fastq.gz" '{print $1}')
  echo $name
  bowtie2 -x hg38 -U $fq -S $name.sam
done
```

1.4 Convert to *bam* file format

Sam files retain all the information from the *fastq* files, but include additional information, including alignment coordinates (`http://samtools.github.io/hts-specs/SAMv1.pdf`). The header has all the chromosome size information from the hg38.fa file.
Next convert the sam file to the compressed and **sorted** BAM format using samtools (version 1.2 used herein) (Li *et al.*, 2009).

```
for sam in *.sam
do
  name=$(echo $sam | awk -F".sam" '{print $1}')
  echo $name
  samtools view -b $sam | samtools sort - $name
done
rm *sam
```

2 `seqOutBias` to generate scaled *bigWig* files

The software `seqOutBias` will scale aligned *bam* read counts by the ratio of genome-wide observed read counts to the sequence based counts for each k-mer. The k-mer counts take into account the mappability at a given read length. The `seqOutBias` program allows for flexibility in specifying k-mer size, strand-specific offsets, and spaced k-mers.

2.1 Using `seqOutBias` to scale DNase-seq files by 6-mer nick preference

The specificity of DNase is strongly influenced by the three bases that flank each side of the DNase cut site (Figure 1) (He *et al.*, 2014; Yardımcı *et al.*, 2014).

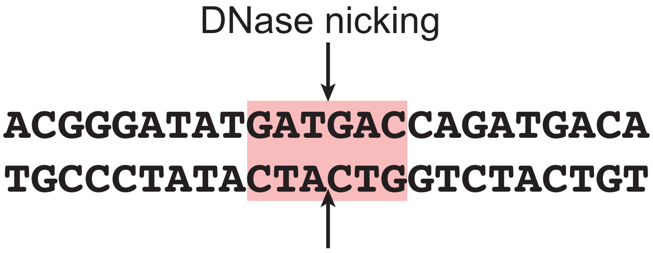


Figure 1: The six base pair window centered on the DNase nick dictates cleavage preference. (He *et al.*, 2014)

`seqOutBias` will calculate the genome-wide occurences of each specified k-mer centered on the DNase nick site accounting for the mappability of the specified read length (note the default is `-read-size=36`). For each case below the offsets are half the value of the `kmer-size` parameter, which is the sequence length (k-mer) that surronds the nick-site and influences specificity, therefore the program will calculate the frequency of k-mers centered on the nick-site. Experimentally, we assume that we are equally likely to sequence either end of a DNase nick site, so the `-shift-counts` parameter is used to shift the Crick strand alignments in line with the Watson strand alignments (Figure 2). DNase nicks can be offset or in line, as shown. Note that generating the mappability files for a given genome and read length is time-consuming, but once these files are made, `seqOutBias` will recognize the existence of these files and avoid timely recomputing and regeneration of these files.

```
bam=UW_MCF7_both.bam
seqOutBias hg38.fa $bam --no-scale --bw=MCF7_0-mer.bigWig --shift-counts --skip-bed
seqOutBias hg38.fa $bam --kmer-size=6 --bw=MCF7_6-mer.bigWig --plus-offset=3 --minus-offset=3 --shift-counts --skip-bed
seqOutBias hg38.fa $bam --kmer-size=10 --bw=MCF7_10-mer.bigWig --plus-offset=5 --minus-offset=5 --shift-counts --skip-bed

bam=IMR90_Naked_DNase.bam
seqOutBias hg38.fa $bam --no-scale --bw=Naked_0-mer.bigWig --shift-counts --skip-bed
seqOutBias hg38.fa $bam --kmer-size=6 --bw=Naked_6-mer.bigWig --plus-offset=3 --minus-offset=3 --shift-counts --skip-bed
seqOutBias hg38.fa $bam --kmer-size=10 --bw=Naked_10-mer.bigWig --plus-offset=5 --minus-offset=5 --shift-counts --skip-bed
```

2.2 Visualizing the single-nucleotide cut files in UCSC

Convert the `bigWig` files to `bedGraph` and add a header to the files for loading into the UCSC genome browser. First use the UCSC tool `bigWigToBedGraph` (`http://hgdownload.cse.ucsc.edu/admin/exe/`) to convert the *bigWig* files to *bedGraph* files, then add a header to the files, and compress.

```
for wig in *bigWig
do
    name=$(echo $wig | awk -F".bigWig" '{print $1}')
    echo $name
    touch temp.txt
    echo "track type=bedGraph name=$name" >> temp.txt
    bigWigToBedGraph $wig $name.bdg
    cat temp.txt $name.bdg > $name.bedGraph
    rm temp.txt
```

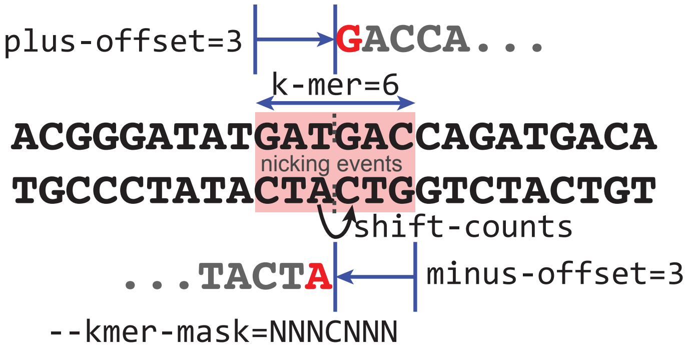
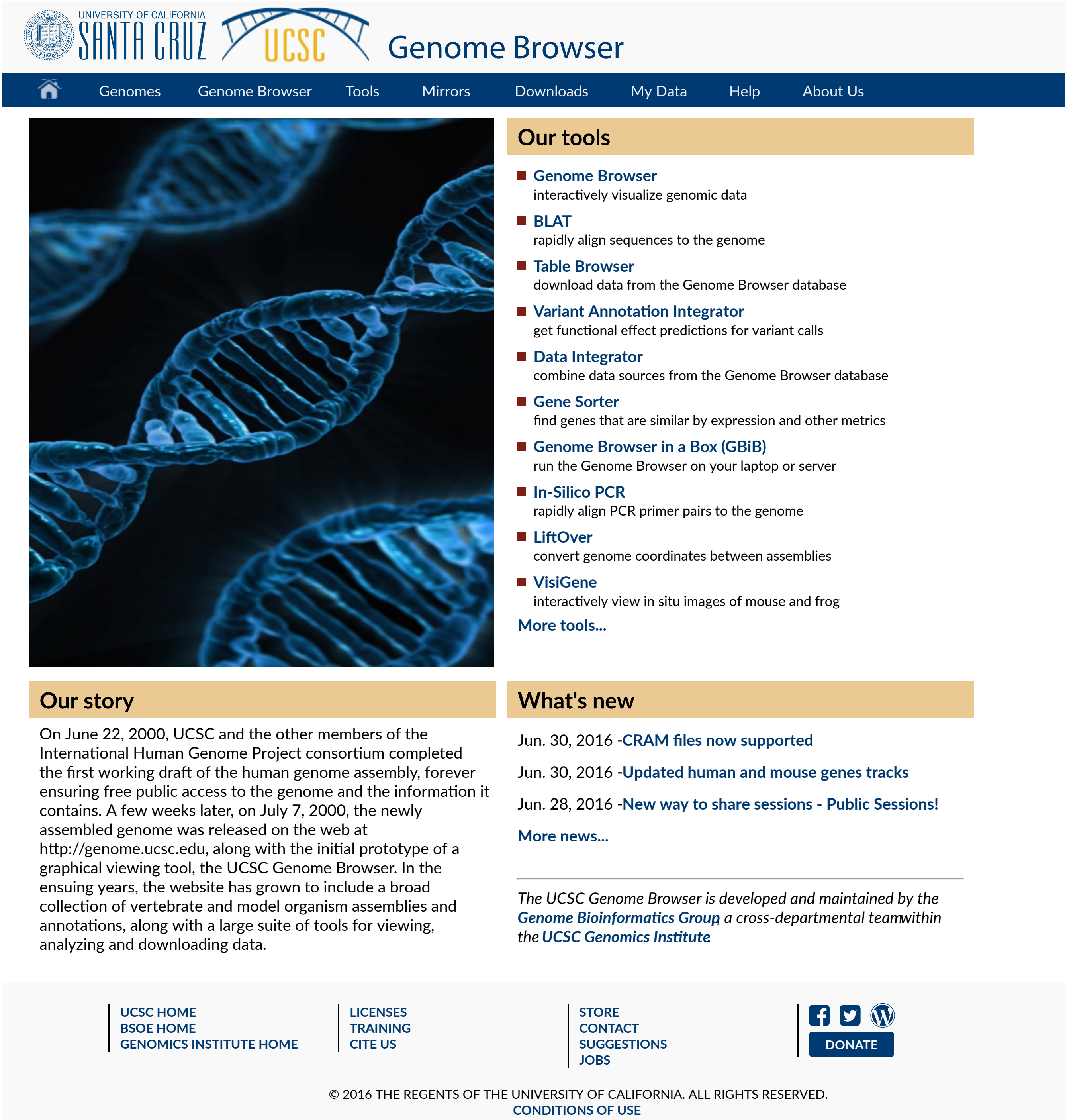


Figure 2: DNase nicking occurs as marked between the two centered base pairs. DNase's specificity is conferred by the hexamer sequence centered (red block) on the nick sites (dotted vertical lines); this parameter is referred to as the `k-mer`. For the purposes of this illustration, the two nicks that result in liberation of the DNA ends are in line. We explore the scenarios where the nicks are offset and result in overhangs in Section 8. The `plus-offset` and `minus-offset` specify the nick site relative to the first position and last position of the `k-mer`. During the library preparation, we assume that the plus and minus strand are equally likely to be sequenced (either red nucleotide will be the first base sequenced). This assumption, however, is not true and the DNA end-repair and ligation have inherent biases. As opposed to specifying the immediate upstream base for the minus strand, we arbitrarily shift the base position by +1 to match the position of the immediate upstream base from the plus aligned read; note that the actual shift amounts will differ depending on the relative positions dictated by the plus/minus-offset values.

```
    rm $name.bdg
    gzip $name.bedGraph
done
mkdir Naked
mkdir MCF7
mv Naked*bigWig Naked
mv MCF7*bigWig MCF7
```

Use the UCSC browser (`https://genome.ucsc.edu`) to visualize the normalized and unnormalized files (Karolchik *et al.*, 2014). Click *Genomes* in the upper left corner (Figure 3). Make sure you have the correct assembly, we are using *hg38*. Next click *add custom tracks* (Figure 4). Use the GUI to navigate to the **.bedGraph.gz* file-containing directory and upload each file individually. You will want to register and save sessions and you will only need to upload the data once.



Our tools

- **Genome Browser**
  interactively visualize genomic data
- **BLAT**
  rapidly align sequences to the genome
- **Table Browser**
  download data from the Genome Browser database
- **Variant Annotation Integrator**
  get functional effect predictions for variant calls
- **Data Integrator**
  combine data sources from the Genome Browser database
- **Gene Sorter**
  find genes that are similar by expression and other metrics
- **Genome Browser in a Box (GBiB)**
  run the Genome Browser on your laptop or server
- **In-Silico PCR**
  rapidly align PCR primer pairs to the genome
- **LiftOver**
  convert genome coordinates between assemblies
- **VisiGene**
  interactively view in situ images of mouse and frog

More tools...

Our story

On June 22, 2000, UCSC and the other members of the International Human Genome Project consortium completed the first working draft of the human genome assembly, forever ensuring free public access to the genome and the information it contains. A few weeks later, on July 7, 2000, the newly assembled genome was released on the web at http://genome.ucsc.edu, along with the initial prototype of a graphical viewing tool, the UCSC Genome Browser. In the ensuing years, the website has grown to include a broad collection of vertebrate and model organism assemblies and annotations, along with a large suite of tools for viewing, analyzing and downloading data.

What's new

Jun. 30, 2016 -**CRAM files now supported**

Jun. 30, 2016 -**Updated human and mouse genes tracks**

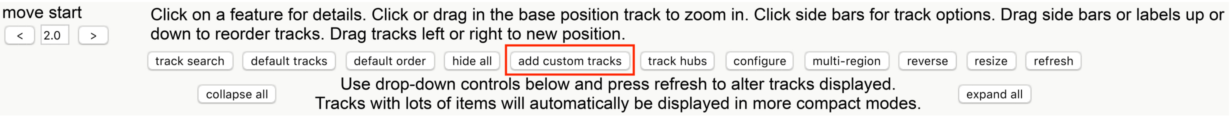
Jun. 28, 2016 -**New way to share sessions - Public Sessions!**

More news...

The UCSC Genome Browser is developed and maintained by the ***Genome Bioinformatics Group****, a cross-departmental teamwithin the* ***UCSC Genomics Institute***

© 2016 THE REGENTS OF THE UNIVERSITY OF CALIFORNIA. ALL RIGHTS RESERVED.
CONDITIONS OF USE

Figure 3: The UCSC homepage (`https://genome.ucsc.edu`).



Click on a feature for details. Click or drag in the base position track to zoom in. Click side bars for track options. Drag side bars or labels up or down to reorder tracks. Drag tracks left or right to new position.

Use drop-down controls below and press refresh to alter tracks displayed.
Tracks with lots of items will automatically be displayed in more compact modes.

Figure 4: Below the browser, click the **add custom tracks** icon.

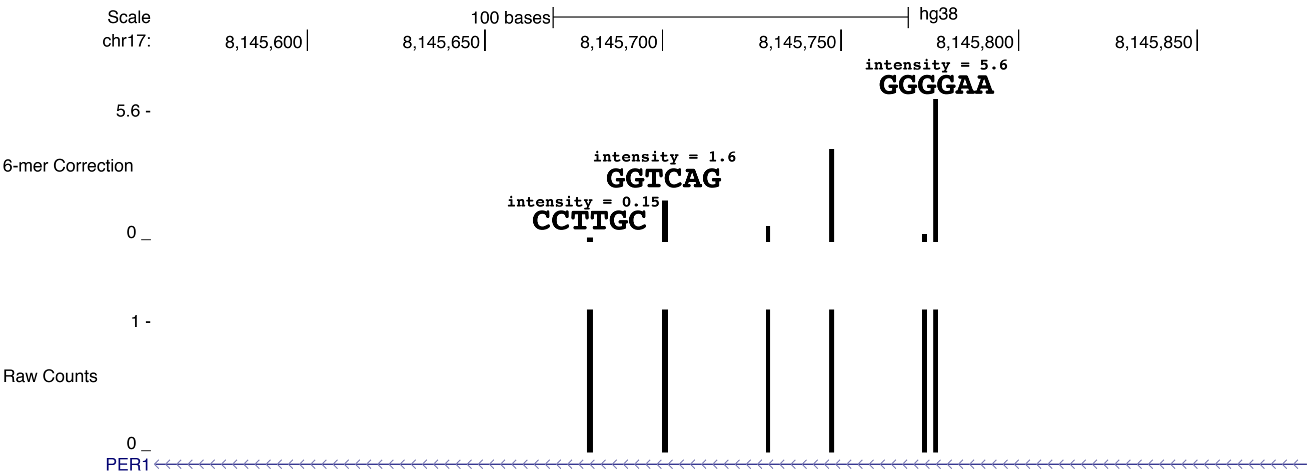


Figure 5: Note that each bar is scaled inversely with DNase sequence preference.

2.3 Generating and analyzing k-mer count tables

Next you can use `seqOutBias table` to generate a table that contains the k-mer index, k-mer string, plus strand count, minus strand count, observed plus strand reads, and observed minus strand reads.

```
seqOutBias table hg38_36.6.3.3.tbl IMR90_Naked_DNase.bam > hg38_36.6.3.3.IMR90_Naked_DNase.txt
```

Compare the frequency of the 4096 hexamers in the genome with the observed cut frequency of DNase using `R`.

```r
setwd('~/DNase_ENCODE')

counts.table = read.table('hg38_36.6.3.3.IMR90_Naked_DNase.txt')
totals = colSums(counts.table[,3:6])
scale.table = data.frame(counts.table[,1:2], t(apply(counts.table[,3:6], 1,
    function(row) c((row[1]/totals[1]) / (row[3] / totals[3]), (row[2] / totals[2]) / (row[4] / totals[4])))))

scale.table[scale.table[,2] == 'CCTTGC',]
scale.table[scale.table[,2] == 'GGTCAG',]
scale.table[scale.table[,2] == 'GGGGAA',]
```

2.4 Retrieving ChIP-seq binding and sequence motif data

To look at composite footprints that result from transcription factor binding to DNA in the context of chromatin, we need to first find all the regions bound by the factor. We get these from processed ENCODE data; we could merge or intersect the replicate files using software like `bedtools` (Quinlan and Hall, 2010), but for the purposes of this vignette we will keep it simple and look at the first replicate *broadPeak* file for three factors. We need to convert these files from hg19 to hg38 coordinates using UCSC `liftOver` `http://hgdownload.cse.ucsc.edu/admin/exe/` and retrieve the sequence associated with each genome coordinate using `fastaFromBed` from `bedtools` (Quinlan and Hall, 2010). Note that we use MAST (Bailey *et al.*, 2009) to identify TF binding sites within ChIP-seq peaks to infer the site of TF binding precisely using traditional DNase-seq data. However, since the naked DNA DNase-seq is lower coverage and the DNA was stripped of proteins, we use FIMO (Grant *et al.*, 2011) to identify all potential TF binding sites in the genome for our composite profiles.

```
url=http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeHaibTfbs/
wget ${url}wgEncodeHaibTfbsMcf7Elf1V0422111PkRep1.broadPeak.gz
wget ${url}wgEncodeHaibTfbsMcf7Gata3V0422111PkRep1.broadPeak.gz
wget ${url}wgEncodeHaibTfbsMcf7MaxV0422111PkRep1.broadPeak.gz
wget ${url}wgEncodeHaibTfbsMcf7CtcfcV0422111PkRep1.broadPeak.gz
wget http://hgdownload.cse.ucsc.edu/goldenPath/hg19/liftOver/hg19ToHg38.over.chain.gz
gunzip hg19ToHg38.over.chain.gz

for peak in *Rep1.broadPeak.gz
do
    name=$(echo $peak | awk -F"wgEncodeHaibTfbsMcf7" '{print $NF}' | awk -F"V0422111PkRep1.broadPeak.gz" '{print $1}')
    unz=$(echo $peak | awk -F".gz" '{print $1}')
    echo $name
    gunzip $peak
    echo $unz
    liftOver $unz hg19ToHg38.over.chain $name.hg38.broadPeak $name.hg38.unmapped.txt -bedPlus=6
    fastaFromBed -fi hg38.fa -bed $name.hg38.broadPeak -fo $name.hg38.fasta
    gzip *broadPeak
done
mv Ctcfc.hg38.fasta CTCF.hg38.fasta
```

We are interested in those factor binding events that are direct and we will use the presence of a strong consensus binding motif as an indicator of direct binding. There are many potential sources for position specific weight matrices, but we will use MEME (Bailey *et al.*, 2006).

```
wget http://meme-suite.org/meme-software/Databases/motifs/motif_databases.12.12.tgz
tar -xvf motif_databases.12.12.tgz
head -9 motif_databases/JASPAR/JASPAR_CORE_2016_vertebrates.meme > header_meme_temp.txt
grep -i -A 14 'MOTIF MA0058.3 MAX' motif_databases/JASPAR/JASPAR_CORE_2016.meme > max_temp.txt
grep -i -A 16 'MOTIF MA0473.2 ELF1' motif_databases/JASPAR/JASPAR_CORE_2016.meme > elf1_temp.txt
grep -i -A 12 'MOTIF MA0037.2 GATA3' motif_databases/JASPAR/JASPAR_CORE_2016.meme > gata3_temp.txt
grep -i -A 23 'MOTIF MA0139.1 CTCF' motif_databases/JASPAR/JASPAR_CORE_2016.meme > ctcf_temp.txt
cat header_meme_temp.txt ctcf_temp.txt > CTCF_minimal_meme.txt
cat header_meme_temp.txt max_temp.txt > Max_minimal_meme.txt
cat header_meme_temp.txt elf1_temp.txt > Elf1_minimal_meme.txt
cat header_meme_temp.txt gata3_temp.txt > Gata3_minimal_meme.txt
rm *temp.txt

for meme in *.hg38.fasta
do
    name=$(echo $meme | awk -F".hg38.fasta" '{print $1}')
    echo $name
    mast ${name}_minimal_meme.txt $meme -hit_list -mt 0.0005 > ${name}_mast.txt
    fimo --thresh 0.0001 --text ${name}_minimal_meme.txt hg38.fa > ${name}_fimo.txt
    ceqlogo -i1 ${name}_minimal_meme.txt -o ${name}_logo.eps -N -Y
done
```

2.5 Use R to plot composite DNase profiles at TF binding sites

First you need to install the bigWig library from André Martins (https://github.com/andrelmartins/bigWig). The lattice and latticeExtra libraries can be installed from the CRAN repository. Recall we process the Naked DNA DNase-seq and conventional DNase-seq separately and the input motifs are distinct for each.

```
source('https://raw.githubusercontent.com/guertinlab/seqOutBias/master/docs/R/seqOutBias_functions.R')

#note that the full path is needed to the directory containing the bigWigs
all.composites.dnase.naked = cycle.fimo.new.not.hotspots(path.dir.fimo = '~/DNase_ENCODE/',
    path.dir.bigWig = '/Users/guertinlab/DNase_ENCODE/Naked/', window = 30, exp = 'Naked_DNase')
all.composites.dnase.mcf7 = cycle.fimo.new.not.hotspots(path.dir.mast = '~/DNase_ENCODE/',
    path.dir.bigWig = '/Users/guertinlab/DNase_ENCODE/MCF7/', window = 30, exp = 'MCF7_DNase')

composites.func.panels.naked.chromatin(all.composites.dnase.mcf7[(all.composites.dnase.mcf7$cond == 'MCF7_0-mer' |
             all.composites.dnase.mcf7$cond == 'MCF7_6-mer') & (all.composites.dnase.mcf7$grp != 'CTCF') ,],
                                       fact = 'MCF7 DNase', summit = 'Motif', num = 24,
                                       col.lines = c(rgb(0,0,1,1/2), rgb(0,0,0,1/2)),
                                        fill.poly = c(rgb(0,0,1,1/4), rgb(0,0,0,1/4)))

composites.func.panels.naked.chromatin(all.composites.dnase.naked[(all.composites.dnase.naked$cond == 'Naked_0-mer' |
             all.composites.dnase.naked$cond == 'Naked_6-mer') & (all.composites.dnase.naked$grp != 'CTCF'),],
                                       fact= "Naked DNase", summit= "Motif",num = 24,
                                       col.lines = c(rgb(0,0,1,1/2), rgb(0,0,0,1/2)),
                                       fill.poly = c(rgb(0,0,1,1/4), rgb(0,0,0,1/4)))

save(all.composites.dnase.naked, all.composites.dnase.mcf7, '~/DNase_ENCODE/MCF7_composites.Rdata')
```

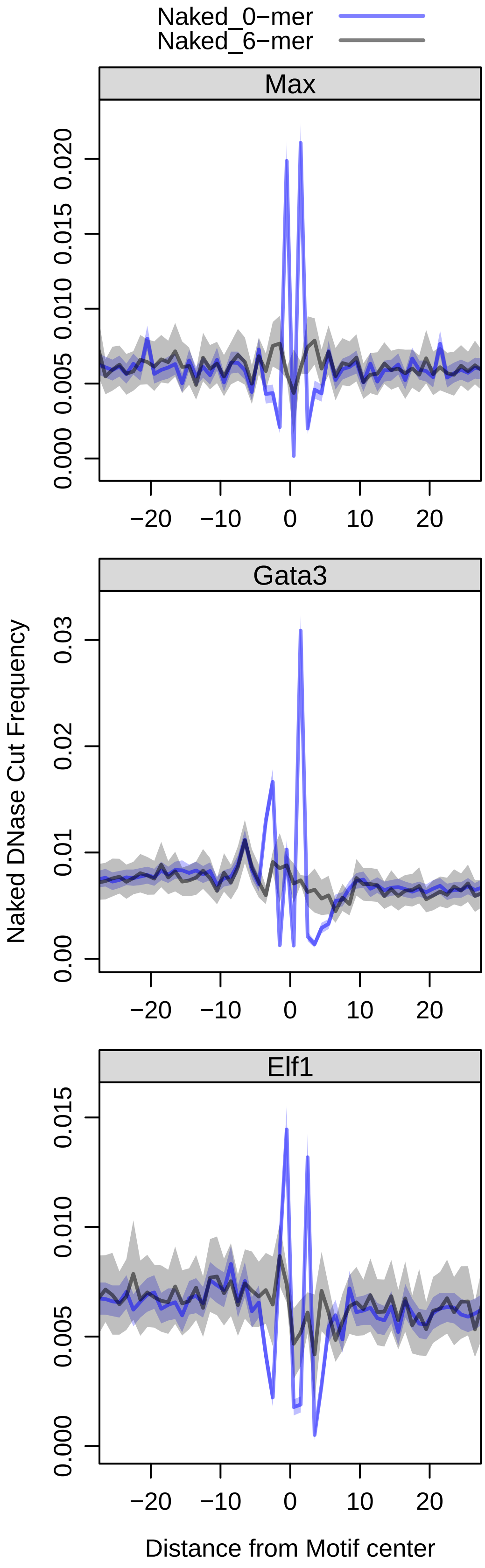


Figure 6: The DNase nick bias is abrogated in an deproteinized DNA (Naked) DNase experiment (Lazarovici *et al.*, 2013) as illustrated by these composite profiles of DNase cut-frequency for three distinct transcription factor motifs.

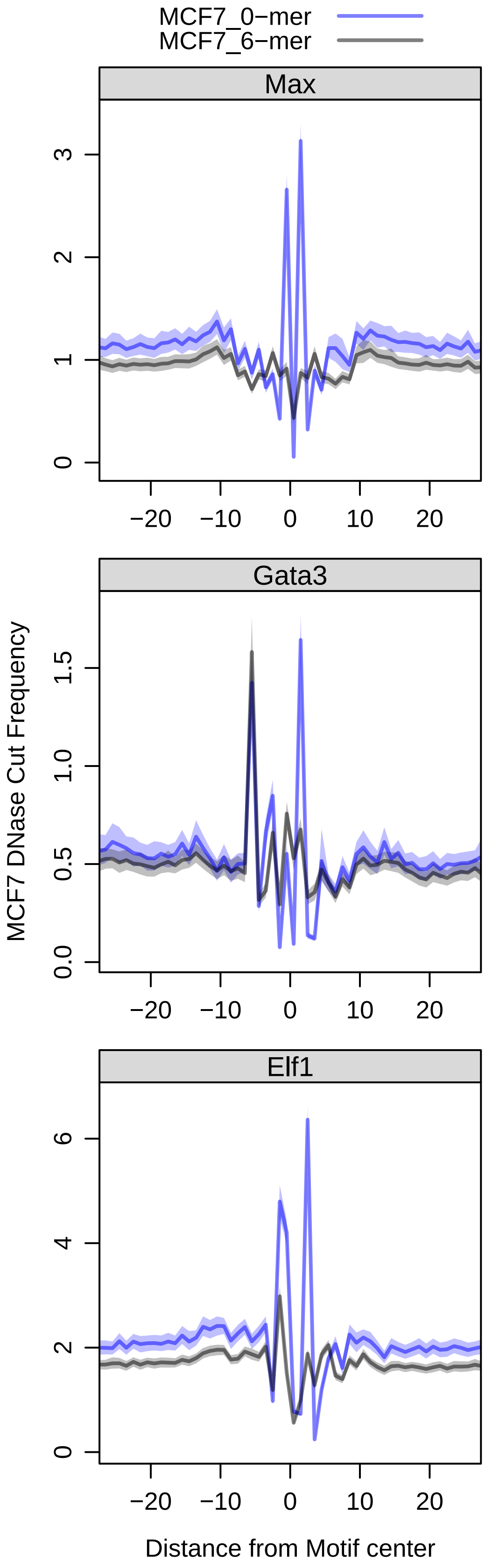


Figure 7: Upon correcting for DNase nick bias, we observe true signatures that may be a result of TF/protein interactions, which we do not observe with the Naked DNase composites. Note the sharp peak upstream of the GATA3 motif; this sharp signature peak is one base-pair downstream (position x = -5.5) of the broader and less intense signature peak observed in Figure 6 (position x = -6.5). Max exhibits a modest *composite* footprint, which is caused by protection from DNase activity mediated by TF/protein interaction.

3 Correction of Tn5 sequence bias from ATAC-seq data

Next we will correct paired-end ATAC-seq data from the Greenleaf group (Buenrostro *et al.*, 2013) and naked DNA ATAC-seq generated by our group. Note that ATAC-seq uses Illumina's Nextera kit to directly transpose sequencing adapters into accessible chromatin. ATAC-seq is unique among enzymatic accessibility assays because each transposition event inserts two adapters into the chromatin. Each Tn5 molecule can be pre-loaded with any combination of the paired-end 1 and paired-end 2 adapter.

3.1 Downloading and processing ATAC-seq data

First we will download SRA files from naked ATAC-seq, which is a genomic DNA isolation followed by standard ATAC-seq protocols. These data are paired-end, which necessitates splitting the `SRA` file into two `fastq` files. Next we exclude all instances of reads that align to chrM–this is not a problem with these data, but typical ATAC-seq on chromatin can yield a high fraction of reads aligning to chrM. We will treat the reads that align to the plus and minus strand differently, because of how the Tn5 recognition site is distinct for plus and minus reads. Note that we optimized the k-mer mask for ATAC-seq.

```
mkdir ~/ATAC_Walavalkar
cd ~/ATAC_Walavalkar

#ATAC Naked
wget ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX243/SRX2438155/SRR5123141/SRR5123141.sra
wget ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX243/SRX2438156/SRR5123142/SRR5123142.sra

fastq-dump --split-3 SRR5123141.sra
fastq-dump --split-3 SRR5123142.sra

mv SRR5123141_1.fastq C1_gDNA_PE1_rep1.fastq
mv SRR5123141_2.fastq C1_gDNA_PE2_rep1.fastq
mv SRR5123142_1.fastq C1_gDNA_PE1_rep2.fastq
mv SRR5123142_2.fastq C1_gDNA_PE2_rep2.fastq

gzip *fastq

wget https://raw.githubusercontent.com/igvteam/igv/master/genomes/sizes/hg38.chrom.sizes

plus_mask=NXNXXXCXXNNXNNNXXN
minus_mask=NXXNNNXNNXXCXXXNXN
for fq in *_PE1_rep1.fastq.gz
do
  name=$(echo $fq | awk -F"_PE1_rep1.fastq.gz" '{print $1}')
  echo $name
  bowtie2 -x ~/DNase_ENCODE/hg38 -1 $fq,${name}_PE1_rep2.fastq.gz -2 ${name}_PE2_rep1.fastq.gz,${name}_PE2_rep2.fastq.gz -S $name.sam
  grep -v '\tchrM\t' $name.sam > $name.chrM.sam
  samtools view -b $name.chrM.sam | samtools sort - $name
  rm *sam
  samtools view -bh -F 20 ${name}.bam > ${name}_plus.bam
  samtools view -bh -f 0x10 ${name}.bam > ${name}_minus.bam
  seqOutBias ~/DNase_ENCODE/hg38.fa ${name}_plus.bam --kmer-mask ${plus_mask} --bw=${name}_plus_${plus_mask}-mer.bigWig --shift-counts --read-size=75
  seqOutBias ~/DNase_ENCODE/hg38.fa ${name}_minus.bam --kmer-mask ${minus_mask} --bw=${name}_minus_${minus_mask}-mer.bigWig --shift-counts --read-size=75
  seqOutBias ~/DNase_ENCODE/hg38.fa ${name}_minus.bam --no-scale --bw=${name}_no_scale_minus.bigWig --shift-counts --read-size=75
  seqOutBias ~/DNase_ENCODE/hg38.fa ${name}_plus.bam --no-scale --bw=${name}_no_scale_plus.bigWig --shift-counts --read-size=75
  bigWigMerge ${name}_plus_${plus_mask}-mer.bigWig ${name}_minus_${minus_mask}-mer.bigWig ${name}_${plus_mask}_${minus_mask}_merged.bedGraph
  bigWigMerge ${name}_no_scale_plus.bigWig ${name}_no_scale_minus.bigWig ${name}_no_scale_merged.bedGraph
  sort -k1,1 -k2,2n ${name}_${plus_mask}_${minus_mask}_merged.bedGraph > ${name}_${plus_mask}_${minus_mask}_merged.sorted.bedGraph
  sort -k1,1 -k2,2n ${name}_no_scale_merged.bedGraph > ${name}_no_scale_merged.sorted.bedGraph
  bedGraphToBigWig ${name}_${plus_mask}_${minus_mask}_merged.sorted.bedGraph hg38.chrom.sizes ${name}_${plus_mask}_${minus_mask}_merged.bigWig
  bedGraphToBigWig ${name}_no_scale_merged.sorted.bedGraph hg38.chrom.sizes ${name}_no_scale_merged.bigWig
done

mkdir Naked
mv C1*merged.bigWig Naked
```

3.2 Processing GM12878 ATAC-seq and TF binding data for GM12878 cells

Perform the same processes for the original ATAC-seq data (Buenrostro *et al.*, 2013). As we did in Section 2.4, we want to retrieve ChIP-seq data for GM12878 cells to plot composite profiles of ATAC signal.

```
#GM12878 Greenleaf
wget ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX298/SRX298000/SRR891268/SRR891268.sra
fastq-dump --split-3 SRR891268.sra
mv SRR891268_1.fastq ATAC_GM12878_PE1.fastq
mv SRR891268_2.fastq ATAC_GM12878_PE2.fastq

gzip ATAC_GM12878_PE1.fastq
gzip ATAC_GM12878_PE2.fastq

plus_mask=NXNXXXCXXNNXNNNXXN
minus_mask=NXXNNNXNNXXCXXXNXN
for fq in *_PE1.fastq.gz
do
  name=$(echo $fq | awk -F"_PE1.fastq.gz" '{print $1}')
  echo $name
  bowtie2 -x ~/DNase_ENCODE/hg38 -1 $fq -2 ${name}_PE2.fastq.gz -S $name.sam
  grep -v '\tchrM\t' $name.sam > $name.chrM.sam
  samtools view -b $name.chrM.sam | samtools sort - $name
  samtools view -bh -F 20 ${name}.bam > ${name}_plus.bam
  samtools view -bh -f 0x10 ${name}.bam > ${name}_minus.bam
  seqOutBias ~/DNase_ENCODE/hg38.fa ${name}_plus.bam --kmer-mask ${plus_mask} --bw=${name}_plus_${plus_mask}-mer.bigWig --shift-counts --read-size=50
  seqOutBias ~/DNase_ENCODE/hg38.fa ${name}_minus.bam --kmer-mask ${minus_mask} --bw=${name}_minus_${minus_mask}-mer.bigWig --shift-counts --read-size=50
  seqOutBias ~/DNase_ENCODE/hg38.fa ${name}_minus.bam --no-scale --bw=${name}_no_scale_minus.bigWig --shift-counts --read-size=50
  seqOutBias ~/DNase_ENCODE/hg38.fa ${name}_plus.bam --no-scale --bw=${name}_no_scale_plus.bigWig --shift-counts --read-size=50
  bigWigMerge ${name}_plus_${plus_mask}-mer.bigWig ${name}_minus_${minus_mask}-mer.bigWig ${name}_${plus_mask}_${minus_mask}_merged.bedGraph
  bigWigMerge ${name}_no_scale_plus.bigWig ${name}_no_scale_minus.bigWig ${name}_no_scale_merged.bedGraph
  sort -k1,1 -k2,2n ${name}_${plus_mask}_${minus_mask}_merged.bedGraph > ${name}_${plus_mask}_${minus_mask}_merged.sorted.bedGraph
  sort -k1,1 -k2,2n ${name}_no_scale_merged.bedGraph > ${name}_no_scale_merged.sorted.bedGraph
  bedGraphToBigWig ${name}_${plus_mask}_${minus_mask}_merged.sorted.bedGraph hg38.chrom.sizes ${name}_${plus_mask}_${minus_mask}_merged.bigWig
  bedGraphToBigWig ${name}_no_scale_merged.sorted.bedGraph hg38.chrom.sizes ${name}_no_scale_merged.bigWig
done

mkdir gm12878
mv *GM12878*merged.bigWig gm12878

url=http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeSydhTfbs/
wget ${url}wgEncodeSydhTfbsGm12878Corestsc30189IggmusPk.narrowPeak.gz
wget ${url}wgEncodeSydhTfbsGm12878Ebf1sc137065StdPk.narrowPeak.gz
wget ${url}wgEncodeSydhTfbsGm12878Ctcfsc15914c20StdPk.narrowPeak.gz
url=http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeHaibTfbs/
wget ${url}wgEncodeHaibTfbsGm12878Sp1Pcr1xPkRep1.broadPeak.gz

grep -i -A 25 'MOTIF MA0138.2 REST' ~/DNase_ENCODE/motif_databases/JASPAR/JASPAR_CORE_2016.meme > REST_meme_temp.txt
grep -i -A 18 'MOTIF MA0154.3 EBF1' ~/DNase_ENCODE/motif_databases/JASPAR/JASPAR_CORE_2016.meme > EBF1_meme_temp.txt
grep -i -A 15 'MOTIF MA0079.3 SP1' ~/DNase_ENCODE/motif_databases/JASPAR/JASPAR_CORE_2016.meme > SP1_meme_temp.txt
grep -i -A 23 'MOTIF MA0139.1 CTCF' ~/DNase_ENCODE/motif_databases/JASPAR/JASPAR_CORE_2016.meme > CTCF_meme_temp.txt

head -9 ~/DNase_ENCODE/motif_databases/JASPAR/JASPAR_CORE_2016_vertebrates.meme > header_meme_temp.txt

for i in *meme_temp.txt
do
    name=$(echo $i | awk -F"_meme" '{print $1}')
    cat header_meme_temp.txt $i > ${name}_minimal_meme.txt
done

rm *temp.txt

for peak in *Gm12878*Peak.gz
do
    name=$(echo $peak | awk -F"TfbsGm12878" '{print $NF}' | awk -F"." '{print $1}')
    unz=$(echo $peak | awk -F".gz" '{print $1}')
    echo $name
    gunzip $peak
    echo $unz
    liftOver $unz ~/DNase_ENCODE/hg19ToHg38.over.chain $name.hg38.narrowPeak $name.hg38.narrow.unmapped.txt -bedPlus=6
    fastaFromBed -fi ~/DNase_ENCODE/hg38.fa -bed $name.hg38.narrowPeak -fo $name.hg38.fasta
    gzip ${name}*Peak
done

mv Ctcfsc15914c20StdPk.hg38.fasta CTCF.hg38.fasta
mv Sp1Pcr1xPkRep1.hg38.fasta SP1.hg38.fasta
mv Ebf1sc137065StdPk.hg38.fasta EBF1.hg38.fasta
mv Corestsc30189IggmusPk.hg38.fasta REST.hg38.fasta

for meme in *.hg38.fasta
do
    name=$(echo $meme | awk -F".hg38.fasta" '{print $1}')
    echo $name
    mast ${name}_minimal_meme.txt $meme -hit_list -mt 0.0005 > ${name}_mast.txt
done

for meme in *.hg38.fasta
do
    name=$(echo $meme | awk -F".hg38.fasta" '{print $1}')
    echo $name
    fimo --thresh 0.0001 --text ${name}_minimal_meme.txt ~/DNase_ENCODE/hg38.fa > ${name}_fimo.txt
    grep -v chrM ${name}_fimo.txt > ${name}_noM_fimo.txt
    rm ${name}_fimo.txt
    mv ${name}_noM_fimo.txt ${name}_fimo.txt
done
```

3.3 Plotting ATAC composites using R

For each transcription factor motif we will plot the ATAC signal using the GM12878 and Naked DNA data as we did in Section 2.5 for DNase data. This section necessitates the loading of functions from Section 2.5.

```
source('https://raw.githubusercontent.com/guertinlab/seqOutBias/master/docs/R/seqOutBias_functions.R')

setwd('~/ATAC_Walavalkar/')

all.composites.ATAC = cycle.fimo.new.not.hotspots(path.dir.mast = '~/ATAC_Walavalkar/',
    path.dir.bigWig = '/Users/guertinlab/ATAC_Walavalkar/gm12878/', window = 30, exp = 'GM12878_ATAC')
all.composites.ATAC.naked = cycle.fimo.new.not.hotspots(path.dir.fimo = '~/ATAC_Walavalkar/',
    path.dir.bigWig = '/Users/guertinlab/ATAC_Walavalkar/Naked/', window = 30, exp = 'Naked_ATAC')

save(all.composites.ATAC, all.composites.ATAC.naked, file = 'ATAC_naked_composites.Rdata')

all.composites.ATAC.naked$cond = gsub("C1_gDNA_no_scale_merged", "Raw", all.composites.ATAC.naked$cond)
all.composites.ATAC.naked$cond = gsub("C1_gDNA_NXNXXXCXXNNXNNNXXN_NXXNNNXNNXXCXXXNXN_merged", "Corrected",
    all.composites.ATAC.naked$cond)

composites.func.panels.naked.chromatin(all.composites.ATAC.naked[(all.composites.ATAC.naked$grp != 'CTCF'),],
                                           fact = paste('ATAC Naked', sep= ' '), summit = 'Motif', num = 24,
                                       col.lines = rev(c(rgb(0,0,1,1/2), rgb(0,0,0,1/2))),
                                        fill.poly = rev(c(rgb(0,0,1,1/4), rgb(0,0,0,1/4))))
```

```
all.composites.ATAC$cond = gsub("ATAC_GM12878_no_scale_merged", "Raw", all.composites.ATAC$cond)
all.composites.ATAC$cond = gsub("ATAC_GM12878_NXNXXXCXXNNXNNNXXN_NXXNNNXNNXXCXXXNXN_merged", "Corrected", all.composites.ATAC$cond)

composites.func.panels.naked.chromatin(all.composites.ATAC[all.composites.ATAC$grp != 'CTCF',],
                                           fact = paste('ATAC GM12878', sep= ' '), summit = 'Motif', num = 24,
                                       col.lines = rev(c(rgb(0,0,1,1/2), rgb(0,0,0,1/2))),
                                        fill.poly = rev(c(rgb(0,0,1,1/4), rgb(0,0,0,1/4))))
```

4 Correction of MNase sequence bias from MNase-seq data

We use the same process to correct MNase-seq data.

4.1 Processing MNase-seq data with `seqOutBias`

The MNase-seq data is paired-end. Note the use of the `pdist=100:400` to specifically process reads that have insert sizes between 100 and 400 base pairs.

```
#completely new
mkdir MNase_Zhang
cd MNase_Zhang
wget ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX564/SRX564203/SRR1323041/SRR1323041.sra
wget ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX564/SRX564204/SRR1323042/SRR1323042.sra
fastq-dump --split-3 SRR1323041.sra
fastq-dump --split-3 SRR1323042.sra
mv SRR1323041_1.fastq MNase_MCF7_PE1_rep1.fastq
mv SRR1323042_1.fastq MNase_MCF7_PE1_rep2.fastq
mv SRR1323041_2.fastq MNase_MCF7_PE2_rep1.fastq
mv SRR1323042_2.fastq MNase_MCF7_PE2_rep2.fastq
gzip *fastq

for fq in *_PE1_rep1.fastq.gz
do
  name=$(echo $fq | awk -F"_PE1_rep1.fastq.gz" '{print $1}')
  echo $name
  bowtie2 -x ~/DNase_ENCODE/hg38 -1 $fq,${name}_PE1_rep2.fastq.gz -2 ${name}_PE2_rep1.fastq.gz,${name}_PE2_rep2.fastq.gz -S $name.sam
  samtools view -b $name.sam | samtools sort - $name
  seqOutBias ~/DNase_ENCODE/hg38.fa $name.bam --no-scale --bw=${name}_0-mer.bigWig --shift-counts --read-size=101 --pdist=100:400
  seqOutBias ~/DNase_ENCODE/hg38.fa $name.bam --kmer-mask=NNNNCNNNN --bw=${name}_8-mer.bigWig --shift-counts --read-size=101 --pdist=100:400
done

mkdir MNase_final
mv MNase_MCF7_0-mer.bigWig MNase_0-mer.bigWig
mv MNase_MCF7_8-mer.bigWig MNase_8-mer.bigWig
mv MNase_0-mer.bigWig MNase_final
mv MNase_8-mer.bigWig MNase_final
```

4.2 Plotting MNase-seq composites using `R`

Plot the composite MNase profile using the MCF7 ChIP-seq peaks from Section 2.4. These MNase-seq data are relatively low coverage and MNase-seq reads are not enriched at TF binding sites, as in DNase-seq. Therefore, the sequence bias correction is more apparent when you average over all the motif instances in the genome that we identified by FIMO in Section 2.4.

```
source('https://raw.githubusercontent.com/guertinlab/seqOutBias/master/docs/R/seqOutBias_functions.R')

all.composites.mnase.mcf7 = cycle.fimo.new.not.hotspots(path.dir.mast =  '~/DNase_ENCODE/',
    path.dir.bigWig = '/Users/guertinlab/MNase_Zhang/MNase_final', window = 30, exp = 'MCF7_MNase')

all.composites.mnase = cycle.fimo.new.not.hotspots(path.dir.fimo = '~/DNase_ENCODE/',
    path.dir.bigWig = '/Users/guertinlab/MNase_Zhang/MNase_final', window = 30, exp = 'MNase')

save(all.composites.mnase.mcf7, file = '~/MNase_Zhang/all.composites.mnase.mcf7.Rdata')
save(all.composites.mnase, file = '~/MNase_Zhang/all.composites.mnase.Rdata')
```

5 Scaling Tissue Accessible Chromatin (TACh) Benzonase and Cyanase Digested DNA

Many enzymes nick DNA with distinct sequence biases. Here we characterize the biases of Benzonase and Cyanase and show that `seqOutBias` can scale DNA digestion data resulting from Benzonase and Cyanase treatment (Grøntved *et al.*, 2012).

5.1 Retrieving TACh-seq data from mouse liver

```
mkdir ~/TACh_Grontved
cd ~/TACh_Grontved
url=ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX174/
wget ${url}SRX174756/SRR535737/SRR535737.sra
wget ${url}SRX174757/SRR535738/SRR535738.sra
wget ${url}SRX174757/SRR535739/SRR535739.sra
wget ${url}SRX174758/SRR535740/SRR535740.sra
wget ${url}SRX174761/SRR535744/SRR535744.sra
wget ${url}SRX174760/SRR535742/SRR535742.sra
wget ${url}SRX174760/SRR535743/SRR535743.sra
wget ${url}SRX174759/SRR535741/SRR535741.sra
wget ${url}SRX174755/SRR535735/SRR535735.sra
wget ${url}SRX174755/SRR535736/SRR535736.sra

for i in *sra
do
   fastq-dump $i
done

mv SRR535737.fastq mm10_liver_Benzonase0.25U.fastq
mv SRR535738.fastq mm10_liver_Benzonase1U_1.fastq
mv SRR535739.fastq mm10_liver_Benzonase1U_2.fastq
mv SRR535740.fastq mm10_liver_Benzonase4U.fastq
mv SRR535741.fastq mm10_liver_Cyanase0.25U.fastq
mv SRR535742.fastq mm10_liver_Cyanase1U_1.fastq
mv SRR535743.fastq mm10_liver_Cyanase1U_2.fastq
mv SRR535744.fastq mm10_liver_Cyanase4U.fastq
mv SRR535735.fastq DNaseI_a.fastq
mv SRR535736.fastq DNaseI_b.fastq

cat *Benz* > mm10_liver_Benzonase.fastq
cat *Cyan* > mm10_liver_Cyanase.fastq
cat DNaseI_*.fastq > mm10_liver_DNase.fastq
gzip *ase.fastq
rm *fastq
rm *sra
```

5.2 Index the mm10 genome file and align to mm10

Retrieve the compressed mm10 genome from UCSC (Karolchik *et al.*, 2014). This is a 2bit compressed file and needs to be converted to a *fasta* using `twoBitToFa` from `http://hgdownload.soe.ucsc.edu/admin/exe/`.

```
wget http://hgdownload.cse.ucsc.edu/goldenPath/mm10/bigZips/mm10.2bit
twoBitToFa mm10.2bit mm10.fa
bowtie2-build mm10.fa mm10

for fq in *.fastq.gz
do
  name=$(echo $fq | awk -F".fastq.gz" '{print $1}')
  echo $name
  bowtie2 -x mm10 -U $fq -S $name.sam
  samtools view -b $name.sam | samtools sort - $name
  rm $name.sam
done
```

5.3 Using `seqOutBias` to determine the sequence preference for Cyanase and Benzonase

```
for bam in mm10_liver*.bam
do
    name=$(echo $bam | awk -F"mm10_liver_" '{print $NF}' | awk -F".bam" '{print $1}')
    echo $name
    seqOutBias mm10.fa $bam --no-scale --bw=${name}_0-mer.bigWig --shift-counts --skip-bed --read-size=35
    seqOutBias mm10.fa $bam --kmer-mask=NNNCNNN --bw=${name}_6-mer.bigWig --shift-counts --read-size=35
    seqOutBias mm10.fa $bam --kmer-mask=NNNNCNNNN --bw=${name}_8-mer.bigWig --shift-counts --read-size=35
```

```
done

mv DNase_0-mer.bigWig DNase_mouse_0-mer.bigWig
mv DNase_6-mer.bigWig DNase_mouse_6-mer.bigWig
mv DNase_8-mer.bigWig DNase_mouse_8-mer.bigWig

mkdir dnase
mkdir benzonase
mkdir cyanase
mv Benzonase*bigWig benzonase
mv Cyanase*bigWig cyanase
mv DNase*bigWig dnase
```

5.4 Retrieving ChIP-seq binding and sequence motif data for mouse liver

To look at composite footprints that are the result of transcription factors binding to DNA in the context of chromatin, we need to first find all the regions bound by the factor. We will retrieve TF binding data from several sources (Seo *et al.*, 2009; Stamatoyannopoulos *et al.*, 2012; Grøntved *et al.*, 2013). Then we convert the peak files to the latest genome assembly.

```
#CTCF
wget https://www.encodeproject.org/files/ENCFF001YAM/@@download/ENCFF001YAM.bed.gz
mv ENCFF001YAM.bed.gz CTCF.mm9.bed.gz
#FOXA2
url=ftp://ftp.ncbi.nlm.nih.gov/geo/series/
wget ${url}GSE25nnn/GSE25836/suppl/GSE25836_Mouse_Liver_FOXA2_GLITR_1p5_FDR.bed.gz
mv GSE25836_Mouse_Liver_FOXA2_GLITR_1p5_FDR.bed.gz FOXA2.mm8.bed.gz
#CEBP-beta
wget ftp://ftp.ncbi.nlm.nih.gov/geo/series/GSE46nnn/GSE46047/suppl/GSE46047%5FCEBPb%5Fpeaks%5Fveh%5Fmouse%5Fliver%5Fmm9%2Etxt%2Egz
mv GSE46047_CEBPb_peaks_veh_mouse_liver_mm9.txt.gz CEBP-beta_temp.mm9.bed.gz

wget http://hgdownload.cse.ucsc.edu/goldenPath/mm9/liftOver/mm9ToMm10.over.chain.gz
wget http://hgdownload.cse.ucsc.edu/goldenPath/mm8/liftOver/mm8ToMm10.over.chain.gz

gunzip *.over.chain.gz
gunzip *mm*bed.gz
tail +2 CEBP-beta_temp.mm9.bed > CEBP-beta.mm9.bed
rm CEBP-beta_temp.mm9.bed

for peak in *mm8.bed
do
    name=$(echo $peak | awk -F".mm8" '{print $1}')
    echo $name
    liftOver $peak mm8ToMm10.over.chain $name.mm10.bed $name.mm10.unmapped.txt -bedPlus=6
    fastaFromBed -fi mm10.fa -bed $name.mm10.bed -fo $name.mm10.fasta
done

for peak in *mm9.bed
do
    name=$(echo $peak | awk -F".mm9" '{print $1}')
    echo $name
    liftOver $peak mm9ToMm10.over.chain $name.mm10.bed $name.mm10.unmapped.txt -bedPlus=6
    fastaFromBed -fi mm10.fa -bed $name.mm10.bed -fo $name.mm10.fasta
done

head -9  ~/DNase_ENCODE/motif_databases/JASPAR/JASPAR_CORE_2016_vertebrates.meme > header_meme_temp.txt

grep -i -A 16 'MOTIF MA0047.2 FOXA2' ~/DNase_ENCODE/motif_databases/JASPAR/JASPAR_CORE_2016.meme > foxa2_temp.txt
grep -i -A 23 'MOTIF MA0139.1 CTCF' ~/DNase_ENCODE/motif_databases/JASPAR/JASPAR_CORE_2016.meme > ctcf_temp.txt
grep -i -A 15 'MOTIF MA0466.2 CEBPB' ~/DNase_ENCODE/motif_databases/JASPAR/JASPAR_CORE_2016.meme > cebpb_temp.txt
cat header_meme_temp.txt foxa2_temp.txt > FOXA2_minimal_meme.txt
cat header_meme_temp.txt ctcf_temp.txt > CTCF_minimal_meme.txt
cat header_meme_temp.txt cebpb_temp.txt > CEBP-beta_minimal_meme.txt
rm *temp.txt

for meme in *.mm10.fasta
do
    name=$(echo $meme | awk -F".mm10.fasta" '{print $1}')
    echo $name
    mast ${name}_minimal_meme.txt $meme -hit_list -mt 0.0001 > ${name}_mast.txt
    fimo --thresh 0.0001 --text ${name}_minimal_meme.txt mm10.fa > ${name}_fimo.txt
    ceqlogo -i1 ${name}_minimal_meme.txt -o ${name}_logo.eps -N -Y
done
```

5.5 Plotting Benzonase and Cyanase composites using R

```
source('https://raw.githubusercontent.com/guertinlab/seqOutBias/master/docs/R/seqOutBias_functions.R')

#note that the full path is needed to the directory containing the bigWigs
all.composites.cyanase = cycle.fimo.new.not.hotspots(path.dir.mast = '~/TACh_Grontved/',
    path.dir.bigWig = '/Users/guertinlab/TACh_Grontved/cyanase/', window = 30, exp = 'Cyanase')

all.composites.benzonase = cycle.fimo.new.not.hotspots(path.dir.mast = '~/TACh_Grontved/',
    path.dir.bigWig = '/Users/guertinlab/TACh_Grontved/benzonase/', window = 30, exp = 'Benzonase')

all.composites.dnase = cycle.fimo.new.not.hotspots(path.dir.mast = '~/TACh_Grontved/',
    path.dir.bigWig = '/Users/guertinlab/TACh_Grontved/dnase/', window = 30, exp = 'DNase_mm10')

save(all.composites.cyanase, all.composites.benzonase, all.composites.dnase, file = 'MOUSE_composites.Rdata')


composites.func.panels.naked.chromatin(all.composites.benzonase[all.composites.benzonase$cond == 'Benzonase_0-mer' |
             all.composites.benzonase$cond == 'Benzonase_8-mer',], fact= "Benzonase8", summit= "Motif",num = 24,
                                       col.lines = c(rgb(0,0,1,1/2), rgb(0,0,0,1/2)),
                                       fill.poly = c(rgb(0,0,1,1/4), rgb(0,0,0,1/4)))

composites.func.panels.naked.chromatin(all.composites.cyanase[all.composites.cyanase$cond == 'Cyanase_0-mer' |
             all.composites.cyanase$cond == 'Cyanase_8-mer',], fact= "Cyanase8", summit= "Motif",num = 24,
                                       col.lines = c(rgb(0,0,1,1/2), rgb(0,0,0,1/2)),
                                       fill.poly = c(rgb(0,0,1,1/4), rgb(0,0,0,1/4)))

composites.func.panels.naked.chromatin(all.composites.dnase[all.composites.dnase$cond == 'DNase_mouse_0-mer' |
             all.composites.dnase$cond == 'DNase_mouse_6-mer',], fact= "Dnase6", summit= "Motif",num = 24,
                                       col.lines = c(rgb(0,0,1,1/2), rgb(0,0,0,1/2)),
                                       fill.poly = c(rgb(0,0,1,1/4), rgb(0,0,0,1/4)))
```

6 PRO-seq T4 RNA ligase correction and analysis

We will explore the sequence bias assocated with single nucleotide resolution GRO-seq (Core *et al.*, 2008): PRO-seq (Kwak *et al.*, 2013). All PRO-seq data is from K562 cells (Core *et al.*, 2014).

6.1 Retrieving and processing PRO-seq data

```
mkdir Core_PRO
wget ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX/SRX683/SRX683602/SRR1554311/SRR1554311.sra
fastq-dump SRR1554311.sra
mv SRR1554311.fastq K562_pro.fastq
rm SRR1554311.sra
fastx_clipper -Q 33 -i K562_pro.fastq -o K562_pro.clipped.fastq -a TGGAATTCTCGGGTGCCAAGG -l 15
rm K562_pro.fastq
fastx_trimmer -Q 33 -l 30 -i K562_pro.clipped.fastq -o K562_pro.trimmed.fastq
rm K562_pro.clipped.fastq
fastx_reverse_complement -Q 33 -z -i K562_pro.trimmed.fastq -o K562_pro.rc.fastq.gz
rm K562_pro.trimmed.fastq
bowtie2 -x ~/DNase_ENCODE/hg38 -U K562_pro.rc.fastq.gz -S K562_pro.sam
samtools view -b K562_pro.sam | samtools sort - K562_pro
rm K562_pro.sam
#process plus and minus aligned reads separately
samtools view -bh -F 20 K562_pro.bam > K562_pro_plus.bam
samtools view -bh -f 0x10 K562_pro.bam > K562_pro_minus.bam
```

6.2 Using `seqOutBias` to correct PRO-seq data

We will not perform genomic k-mer correction, instead we will will look exclusively at gene annotations. The vast majority of transcription occurs in annotated genes, although lower levels of transcription are pervasive in the genome. The reason we are looking at genes is because it is conceivable that the k-mer counts are different between the genome and the transcribed units.

```
#gene annotations
wget ftp://ftp.ensembl.org/pub/release-87/gtf/homo_sapiens/Homo_sapiens.GRCh38.87.gtf.gz
gunzip gunzip Homo_sapiens.GRCh38.87.gtf.gz
awk '$3 == "gene"' Homo_sapiens.GRCh38.87.gtf | sed 's/^/chr/' | awk '{OFS="\t";} {print $1,$4,$5,$2,$6,$7}' > Homo_sapiens.GRCh38.87.bed
awk '$3 == "gene"' Homo_sapiens.GRCh38.87.gtf | sed 's/^/chr/' | awk '{OFS="\t";} {print $1,$4,$5,$2,$6,$7}' > Homo_sapiens.GRCh38.87.bed
awk '$6 == "+"' Homo_sapiens.GRCh38.87.bed | awk '{OFS="\t";} {print $1,$2,$2+100,$4,$5,$6}' > Homo_sapiens.GRCh38.87.plus.dsTSS.bed
awk '$6 == "-"' Homo_sapiens.GRCh38.87.bed | awk '{OFS="\t";} {print $1,$3-100,$3,$4,$5,$6}' > Homo_sapiens.GRCh38.87.minus.dsTSS.bed
cat Homo_sapiens.GRCh38.87.plus.dsTSS.bed Homo_sapiens.GRCh38.87.minus.dsTSS.bed > Homo_sapiens.GRCh38.87.dsTSS.bed
subtractBed -s -a Homo_sapiens.GRCh38.87.bed -b Homo_sapiens.GRCh38.87.dsTSS.bed > Homo_sapiens.GRCh38.87.body.bed
awk '$6 == "+"' Homo_sapiens.GRCh38.87.body.bed > Homo_sapiens.GRCh38.87.body.plus.bed
awk '$6 == "-"' Homo_sapiens.GRCh38.87.body.bed > Homo_sapiens.GRCh38.87.body.minus.bed

sort -k1,1 -k2,2n Homo_sapiens.GRCh38.87.body.plus.bed > Homo_sapiens.GRCh38.87.body.plus.sorted.bed
sort -k1,1 -k2,2n Homo_sapiens.GRCh38.87.body.minus.bed > Homo_sapiens.GRCh38.87.body.minus.sorted.bed
```

```
rm Homo_sapiens.GRCh38.87.body.plus.bed
rm Homo_sapiens.GRCh38.87.body.minus.bed

mergeBed -s -i Homo_sapiens.GRCh38.87.body.plus.sorted.bed  > Homo_sapiens.GRCh38.87.body.plus.bed
mergeBed -s -i Homo_sapiens.GRCh38.87.body.minus.sorted.bed  > Homo_sapiens.GRCh38.87.body.minus.bed
```

We will process the reads that align to the plus and minus strand separately and implement the `tail-edge` option to output the 3′ end of the sequence read.

```
#correct based on k-mers observed in gene bodies
reg=Homo_sapiens.GRCh38.87.body
bam=K562_pro_plus.bam
seqOutBias ~/DNase_ENCODE/hg38.fa $bam --regions=${reg}.plus.bed --no-scale --bw=PRO_plus_body_0-mer.bigWig --tail-edge --read-size=30
bam=K562_pro_minus.bam
seqOutBias ~/DNase_ENCODE/hg38.fa $bam --regions=${reg}.minus.bed --no-scale --bw=PRO_minus_body_0-mer.bigWig --tail-edge --read-size=30

bam=K562_pro_plus.bam
seqOutBias ~/DNase_ENCODE/hg38.fa $bam --regions=${reg}.plus.bed --kmer-mask=NNNCNNN --bw=PRO_plus_body_NNNCNNN-mer.bigWig --tail-edge --read-size=30

bam=K562_pro_minus.bam
seqOutBias ~/DNase_ENCODE/hg38.fa $bam --regions=${reg}.minus.bed --kmer-mask=NNNCNNN --bw=PRO_minus_body_NNNCNNN-mer.bigWig --tail-edge --read-size=30

#for loading into UCSC
for bw in *plus*-mer.bigWig
do
    name=$(echo $bw | awk -F"/" '{print $NF}' | awk -F".bigWig" '{print $1}')
    echo $name
    bigWigToBedGraph $bw $name.bg
    touch temp.txt
    echo "track type=bedGraph name=$name color=255,0,0 alwaysZero=on visibility=full" >> temp.txt
    cat temp.txt $name.bg > $name.bedGraph
    rm temp.txt
    rm $name.bg
    gzip $name.bedGraph
done

for bw in *minus*mer.bigWig
do
    name=$(echo $bw | awk -F"/" '{print $NF}' | awk -F".bigWig" '{print $1}')
    echo $name
    bigWigToBedGraph $bw $name.bg
    touch temp.txt
    echo "track type=bedGraph name=$name color=0,0,255 alwaysZero=on visibility=full" >> temp.txt
    cat temp.txt $name.bg > $name.bedGraph
    rm temp.txt
    rm $name.bg
    gzip $name.bedGraph
done

mkdir plus
mkdir minus
mv *minus*bigWig minus
mv *plus*bigWig plus
```

6.3 Plotting PRO-seq density surrounding TF binding sites

Next we will look at PRO-seq signal centered around CTCF binding sites. We will consider the orientation of the CTCF motif and the original alignment strand of the sequence read.

```
wget http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeUwTfbs/wgEncodeUwTfbsK562CtcfStdPkRep1.narrowPeak.gz

for peak in *narrowPeak.gz
do
    name=$(echo $peak | awk -F"wgEncodeUwTfbsK562" '{print $NF}' | awk -F"sc" '{print $1}' | awk -F"Std" '{print $1}')
    unz=$(echo $peak | awk -F".gz" '{print $1}')
    echo $name
    gunzip $peak
    echo $unz
    liftOver $unz ~/DNase_ENCODE/hg19ToHg38.over.chain $name.hg38.narrowPeak $name.hg38.narrow.unmapped.txt -bedPlus=6
    fastaFromBed -fi ~/DNase_ENCODE/hg38.fa -bed $name.hg38.narrowPeak -fo $name.hg38.fasta
    gzip *narrowPeak
done

#specifically find peaks in the gene bodies of genes on the + and - strands
#specifically look at CTCF binding sites in both orientations

mv Ctcf.hg38.fasta CTCF.hg38.fasta
mv Ctcf.hg38.narrowPeak.gz CTCF.hg38.k562.narrowPeak.gz
head -9 ~/DNase_ENCODE/motif_databases/JASPAR/JASPAR_CORE_2016_vertebrates.meme > header_meme_temp.txt
grep -i -A 23 'MOTIF MA0139.1 CTCF' ~/DNase_ENCODE/motif_databases/JASPAR/JASPAR_CORE_2016.meme > ctcf_temp.txt
cat header_meme_temp.txt ctcf_temp.txt > CTCF_minimal_meme.txt
rm *temp.txt

gunzip CTCF.hg38.k562.narrowPeak
intersectBed -a CTCF.hg38.k562.narrowPeak -b Homo_sapiens.GRCh38.87.body.plus.bed > CTCF.hg38.k562.gb.plus.peaks.bed
fastaFromBed -fi ~/DNase_ENCODE/hg38.fa -bed CTCF.hg38.k562.gb.plus.peaks.bed -fo CTCF.hg38.k562.gb.plus.peaks.fasta
intersectBed -a CTCF.hg38.k562.narrowPeak -b Homo_sapiens.GRCh38.87.body.minus.bed > CTCF.hg38.k562.gb.minus.peaks.bed
fastaFromBed -fi ~/DNase_ENCODE/hg38.fa -bed CTCF.hg38.k562.gb.minus.peaks.bed -fo CTCF.hg38.k562.gb.minus.peaks.fasta

for meme in *.peaks.fasta
do
    name=$(echo $meme | awk -F".hg38" '{print $1}')
    nm=$(echo $meme | awk -F".peaks" '{print $1}')
    echo $name
    mast ${name}_minimal_meme.txt $meme -hit_list -mt 0.0005 > ${nm}_mast.txt
done

for i in *_mast.txt
do
   name=$(echo $i | awk -F"_mast" '{print $1}')
   grep ' +1 ' $i > ${name}_plus_mast.txt
   grep ' -1 ' $i > ${name}_minus_mast.txt
done
```

```
source('https://raw.githubusercontent.com/guertinlab/seqOutBias/master/docs/R/seqOutBias_functions.R')

all.composites.plus.pro = cycle.fimo.new.not.hotspots(path.dir.mast = '~/Core_PRO/',
    path.dir.bigWig = '/Users/guertinlab/Core_PRO/plus/', window = 30, exp = 'PRO plus')

all.composites.minus.pro = cycle.fimo.new.not.hotspots(path.dir.mast = '~/Core_PRO/',
    path.dir.bigWig = '/Users/guertinlab/Core_PRO/minus/', window = 30, exp = 'PRO minus')

save(all.composites.minus.pro, all.composites.plus.pro, file = 'PRO_composites.Rdata')

composites.func.pro(all.composites.plus.pro, fact= "PRO plus", summit= "CTCF motif",num =24)
composites.func.pro(all.composites.minus.pro, fact= "PRO minus", summit= "CTCF motif",num =24)
```

6.4 Plotting PRO-seq density surrounding splice sites

Previous work in *Drosophila* has shown that RNA Polymerase density decreases directly upstream of the 5′ end of exons, at the site of the 3′ splice site (Kwak *et al.*, 2013). To determine whether the run on experiment or the library preparation exhibit sequence biases, we plot the PRO-seq density at the 5′ exon boundary. We will exclude the first exon from our analysis, because RNA Polymerase II pausing is a common feature of most genes. We will process the plus and minus strand genes separately. Additionally, we will plot the composites for distinct splice acceptor sequences. Positions -3 relative to the exon start tolerates all nucleotides, but C and T are prefered. As expected, the span of the confidence intervals correlates with the number of motif instances in each category (Figure 8). Therefore, we randomly selected 38358 CAG rows to match the 38358 TAG rows–there are many fewer instances of the AAG 3′ splice acceptor, so we excluded these. The composite profiles show that the CAG consensus splice site (compared to TAG) promotes slower elongation rate in the 5´end of exons (Figure 9).

```
awk '$3 == "exon"' Homo_sapiens.GRCh38.87.gtf | sed 's/^/chr/' | grep -v 'exon_number "1"' | awk '{OFS="\t";} {print $1,$4,$5,$2,$6,$7}' > Homo_sapiens.GRCh38.87.exon.bed

awk '$6 == "+"' Homo_sapiens.GRCh38.87.exon.bed | awk -F"\t" '!seen[$1, $2]++' | grep -v '\.1' > Homo_sapiens.GRCh38.87.exon.plus.bed
awk '$6 == "-"' Homo_sapiens.GRCh38.87.exon.bed | awk -F"\t" '!seen[$1, $3]++' | grep -v '\.1' | grep -v '\.2'  > Homo_sapiens.GRCh38.87.exon.minus.bed

exFile=Homo_sapiens.GRCh38.87.exon.plus.bed
awk '{$2 = $2 - 21; print}' $exFile | awk '{OFS="\t";} {$3 = $2 + 20; print}' | fastaFromBed -fi ~/DNase_ENCODE/hg38.fa -s -bed stdin -fo exon.hg38.20.fasta

declare -a arr=("cag" "tag" "gag" "aag")

for i in "${arr[@]}"
do
    echo $i
    grep -B 1 -i $i exon.hg38.fasta | grep '>' > exon.${i}.hg38.fasta
    cat exon.${i}.hg38.fasta | awk -F">" '{print $NF}' | awk -F":" '{print $1}' > exon.${i}.chr.txt
    cat exon.${i}.hg38.fasta | awk -F":" '{print $NF}' | awk -F"-" '{print $1}' > exon.${i}.start.txt
    cat exon.${i}.hg38.fasta | awk -F"-" '{print $NF}' | awk -F"(" '{print $1}' > exon.${i}.end.txt
    cat exon.${i}.hg38.fasta | awk -F"(" '{print $NF}' | awk -F")" '{print $1}' > exon.${i}.strand.txt
    paste exon.${i}.chr.txt exon.${i}.start.txt exon.${i}.end.txt exon.${i}.strand.txt exon.${i}.strand.txt exon.${i}.strand.txt > exon.${i}.hg38.bed
    fastaFromBed -fi ~/DNase_ENCODE/hg38.fa -s -bed exon.${i}.hg38.bed -fo exon.${i}.hg38.3nuc.fasta
    rm exon.*.hg38.*.txt
done
```

```
source('https://raw.githubusercontent.com/guertinlab/seqOutBias/master/docs/R/seqOutBias_functions.R')

exon.plus = read.table('~/Core_PRO/Homo_sapiens.GRCh38.87.exon.plus.bed')
exon.plus[,3] = exon.plus[,2] + 149
exon.plus[,2] = exon.plus[,2] - 151
exon.minus = read.table('~/Core_PRO/Homo_sapiens.GRCh38.87.exon.minus.bed')
exon.minus[,2] = exon.minus[,3] - 150
exon.minus[,3] = exon.minus[,3] + 150

plus.exon.pro = composites.test.naked('/Users/guertinlab/Core_PRO/plus', exon.plus, region = 150, grp = 'PRO-seq')
plus.exon.pro[[1]]$x = plus.exon.pro[[1]]$x -0.5
composites.func.panels.naked.chromatin(plus.exon.pro[[1]], fact = 'PolII', summit = 'Exon plus', num=30)

minus.exon.pro = composites.test.naked('/Users/guertinlab/Core_PRO/minus', exon.minus, region = 150, grp = 'PRO-seq')
minus.exon.pro[[1]]$x = minus.exon.pro[[1]]$x -0.5
```

```r
composites.func.panels.naked.chromatin(minus.exon.pro[[1]], fact = 'PolII', summit = 'Exon minus -', num=30)

save(minus.exon.pro, plus.exon.pro, file = '~/Core_PRO/exon.pro.Rdata')
load('~/Core_PRO/exon.pro.Rdata')

#acceptor seqLogo
exonjunc = read.table('exon.hg38.20.fasta', comment.char = '>')
exonjunc[,1] = as.character(exonjunc[,1])
exonjunc = data.frame(lapply(exonjunc, function(v) {
  if (is.character(v)) return(toupper(v))
  else return(v)
}))

pswm.func(exonjunc[,1], 'splice_acceptor', positions = 20)

#subdividing sequences at acceptor site
exon.aag = read.table('~/Core_PRO/exon.aag.hg38.bed')
exon.cag = read.table('~/Core_PRO/exon.cag.hg38.bed')
exon.gag = read.table('~/Core_PRO/exon.gag.hg38.bed')
exon.tag = read.table('~/Core_PRO/exon.tag.hg38.bed')

#selecting the same number of coordinates to generate coparable confidence interval estimates
exon.cag = randomRows(exon.cag, nrow(exon.tag))

exon.cag[,3] = exon.cag[,2] + 153
exon.cag[,2] = exon.cag[,2] - 147
plus.exon.cag.pro = composites.test.naked('/Users/guertinlab/Core_PRO/plus', exon.cag, region = 150, grp = 'PRO-seq')
plus.exon.cag.pro[[1]]$x = plus.exon.cag.pro[[1]]$x -0.5
composites.func.panels.naked.chromatin(plus.exon.cag.pro[[1]], fact = 'PolII', summit = 'Exon CAG plus', num=30)

exon.tag[,3] = exon.tag[,2] + 153
exon.tag[,2] = exon.tag[,2] - 147
plus.exon.tag.pro = composites.test.naked('/Users/guertinlab/Core_PRO/plus', exon.tag, region = 150, grp = 'PRO-seq')
plus.exon.tag.pro[[1]]$x = plus.exon.tag.pro[[1]]$x -0.5
composites.func.panels.naked.chromatin(plus.exon.tag.pro[[1]], fact = 'PolII', summit = 'Exon TAG plus', num=30)

exon.aag[,3] = exon.aag[,2] + 153
exon.aag[,2] = exon.aag[,2] - 147
plus.exon.aag.pro = composites.test.naked('/Users/guertinlab/Core_PRO/plus', exon.aag, region = 150, grp = 'PRO-seq')
plus.exon.aag.pro[[1]]$x = plus.exon.aag.pro[[1]]$x -0.5
composites.func.panels.naked.chromatin(plus.exon.aag.pro[[1]], fact = 'PolII', summit = 'Exon AAG plus', num=30)

save(minus.exon.pro, plus.exon.pro, plus.exon.aag.pro, plus.exon.cag.pro, plus.exon.tag.pro, file = '~/Core_PRO/exon.pro.Rdata')
load('~/Core_PRO/exon.pro.Rdata')
```

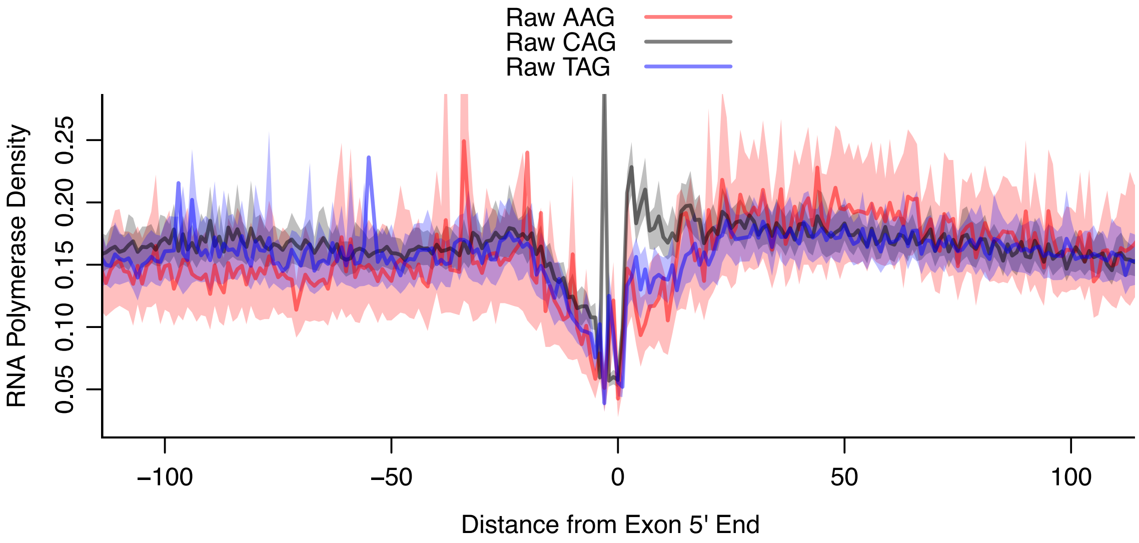


Figure 8: We observe a sharp skipe in position -3 only at CAG 3′ splice sites. This indicates that cytosine is preferentially incorporated during the run-on or preferentially ligated.

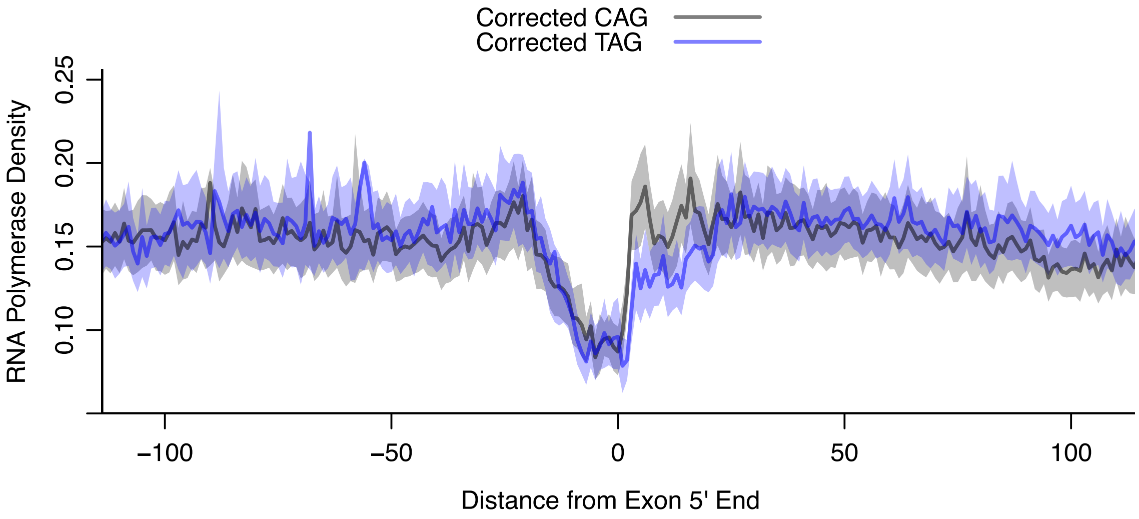


Figure 9: We examined the composite profiles at corrected CAG and TAG splice acceptor sites and we observe that RNA polymerase density is higher following CAG splice acceptor sites, which indicates that the Polymerase proceeds into the exon more slowly following a CAG splice acceptor site.

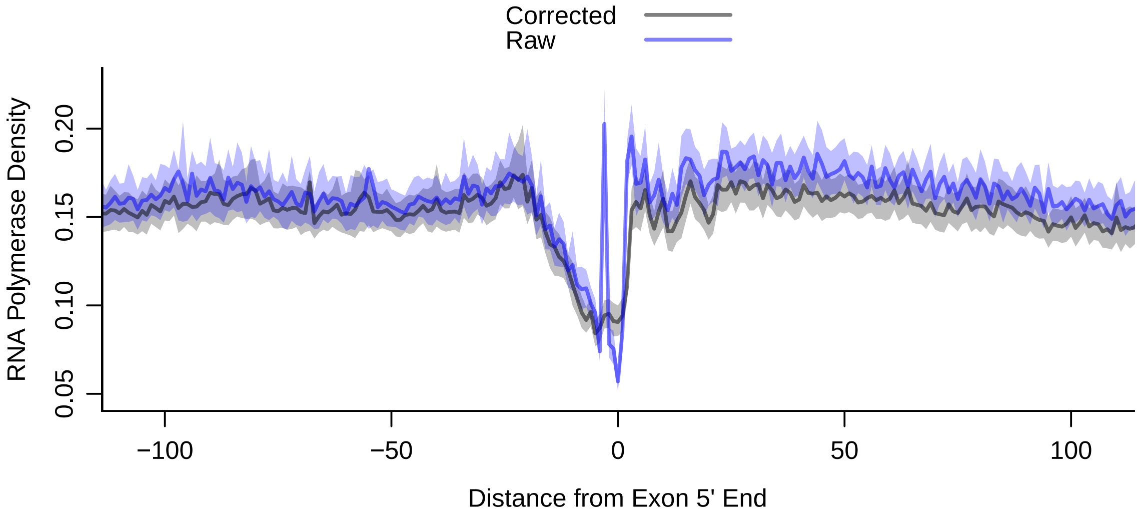


Figure 10: Upon correcting for enzymatic sequence bias, the signature at the 3′ splice site is abrogated. The first base of the exon spans position 0-1 on the x-axis. The position -3 upstream from the exon start results from T4 RNA ligase sequence bias and this sequence bias is corrected by `seqOutBias`.

6.5 Plotting `seqOutBias` correction of DNase, MNase, ATAC, TACh, and PRO-seq data at CTCF binding sites.

The only factor with ChIP-seq data in MCF7, GM12878, K562, and mouse liver is CTCF. `SeqOutBias` corrects the sequence bias for CTCF reasonably well. Although, the Tn5 bias seems to span a wide domain and a k-mer correction is likely not optimal, as sequence features that span this domain likely influence Tn5 recognition.

```r
source('https://raw.githubusercontent.com/guertinlab/seqOutBias/master/docs/R/seqOutBias_functions.R')

load('~/DNase_ENCODE/MCF7_composites.Rdata')
load('~/TACh_Grontved/MOUSE_composites.Rdata')
load('~/MNase_Zhang/all.composites.mnase.mcf7.Rdata')
load('~/ATAC_Walavalkar/ATAC_naked_composites.Rdata')
load('~/Core_PRO/PRO_composites.Rdata')

#Comparing correction of sequence bias dictated by the CTCF motif
#all.composites.ATAC$grp = gsub("CTCF_GM12878", "CTCF", all.composites.ATAC$grp)

all.composites.ATAC$cond = gsub("ATAC_GM12878_no_scale_merged", "ATACgm_0-mer", all.composites.ATAC$cond)
all.composites.ATAC$cond = gsub("ATAC_GM12878_NXNXXXCXXNNXNNNXXN_NXXNNNXNNXXCXXXNXN_merged", "ATACgm_NXNXXXCXXNNXNNNXXN-mer",
    all.composites.ATAC$cond)

all.composites.ATAC.naked$cond = gsub("C1_gDNA_no_scale_merged", "ATACnk_0-mer", all.composites.ATAC.naked$cond)
all.composites.ATAC.naked$cond = gsub("C1_gDNA_NXNXXXCXXNNXNNNXXN_NXXNNNXNNXXCXXXNXN_merged", "ATACnk_NXNXXXCXXNNXNNNXXN-mer",
    all.composites.ATAC.naked$cond)

all.composites.plus.pro$cond = gsub("PRO_plus_body_0-mer", "PRO_0-mer", all.composites.plus.pro$cond)
all.composites.plus.pro$cond = gsub("PRO_plus_body_NNNCNNN-mer", "PRO_6-mer", all.composites.plus.pro$cond)

alldf = rbind(all.composites.cyanase, all.composites.benzonase, all.composites.dnase.mcf7,
    all.composites.dnase.naked, all.composites.ATAC, all.composites.ATAC.naked, all.composites.mnase.mcf7,
    all.composites.plus.pro)
#colnames(alldf) = c('est', 'x', 'grp', 'upper', 'lower', 'cond')
ctcf.df = alldf[alldf$grp == 'CTCF',]
ctcf.df = ctcf.df[ctcf.df$cond == 'Cyanase_0-mer' | ctcf.df$cond == 'Cyanase_10-mer' | ctcf.df$cond == 'Benzonase_0-mer' |
                      ctcf.df$cond == 'Benzonase_10-mer' | ctcf.df$cond == 'MCF7_0-mer' | ctcf.df$cond == 'MCF7_6-mer' |
                          ctcf.df$cond == 'Naked_0-mer' | ctcf.df$cond == 'Naked_6-mer' | ctcf.df$cond == 'ATACgm_0-mer' |
                              ctcf.df$cond == 'ATACgm_NXNXXXCXXNNXNNNXXN-mer' | ctcf.df$cond == 'ATACnk_0-mer' |
                                 ctcf.df$cond == 'ATACnk_NXNXXXCXXNNXNNNXXN-mer' | ctcf.df$cond == 'MNase_0-mer' |
                                     ctcf.df$cond == 'MNase_8-mer' | ctcf.df$cond == 'PRO_0-mer' |
                                         ctcf.df$cond == 'PRO_6-mer',]
ctcf.df$grp = sapply(strsplit(as.character(ctcf.df$cond),'_'), "[", 1)
ctcf.df$cond = sapply(strsplit(as.character(ctcf.df$cond),'_'), "[", 2)

ctcf.df[ctcf.df=="ATACgm"] = 'ATAC Chromatin'
ctcf.df[ctcf.df=="MCF7"] = 'DNase Chromatin'
#ctcf.df[ctcf.df=="MNase"] = 'DNase Chromatin'

ctcf.df[ctcf.df=="Naked"] = 'DNase Naked DNA'
ctcf.df[ctcf.df=="ATACnk"] = 'ATAC Naked DNA'
ctcf.df[ctcf.df=="ATACnk"] = 'ATAC Naked DNA'
ctcf.df[ctcf.df=="PRO"] = 'Precision Run-On'

ctcf.df[ctcf.df=='0-mer'] = 'Raw'
ctcf.df[ctcf.df=="10-mer"] = 'Corrected'
ctcf.df[ctcf.df=="6-mer"] = 'Corrected'
ctcf.df[ctcf.df=="8-mer"] = 'Corrected'
ctcf.df[ctcf.df=="NXNXXXCXXNNXNNNXXN-mer"] = 'Corrected'


composites.func(ctcf.df, fact= "Experimental", summit= "CTCF motif",num = 24,
                                       col.lines = rev(c(rgb(0,0,1,1/2), rgb(0,0,0,1/2))),
                                       fill.poly = rev(c(rgb(0,0,1,1/4), rgb(0,0,0,1/4))))
```

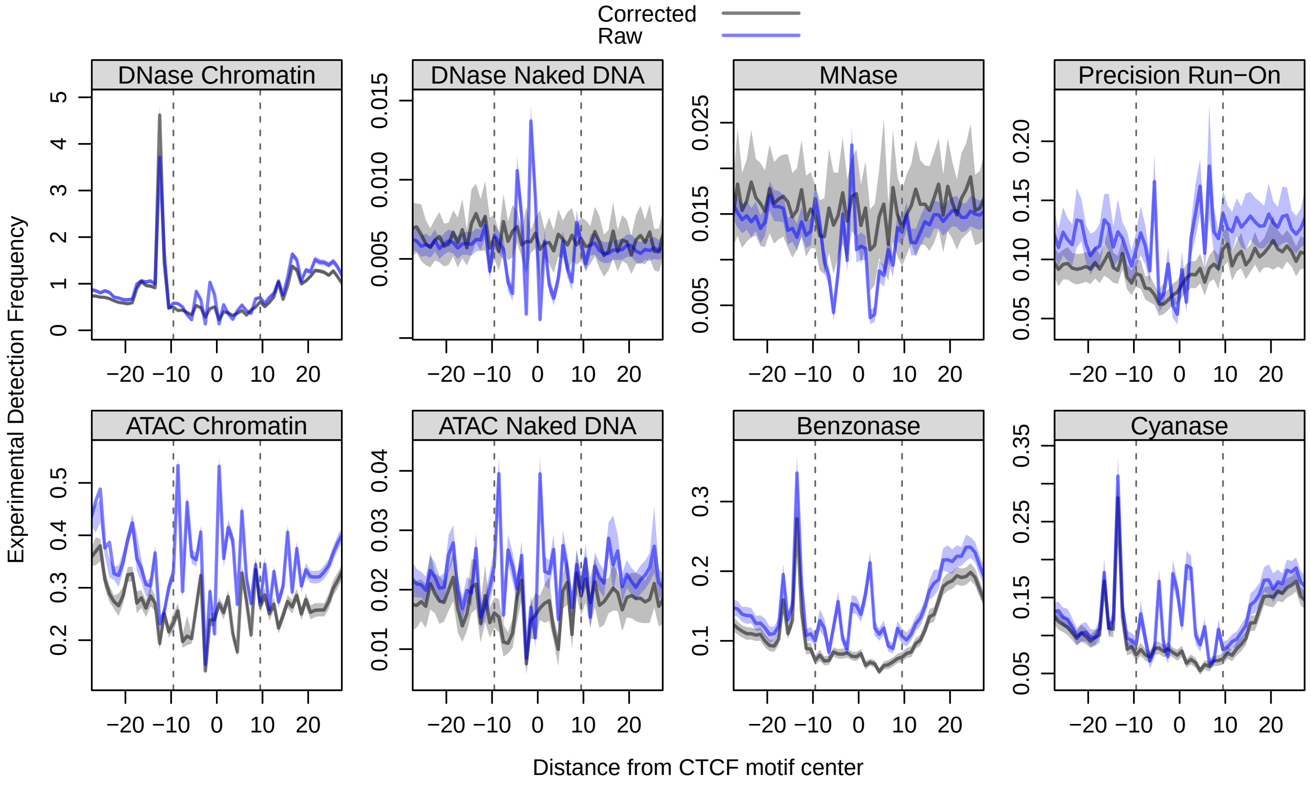


Figure 11: Upon correcting for enzymatic sequence bias, the signature at the site of CTCF binding is abrogated in each molecular genomics data set we tested. However, in cases of CTCF binding to chromatin, we observe protection that results in a footprint; note that MNase is not expected to result in a composite footprint. We observe the previously characterized sharp peak upstream of the CTCF motif; this signature is likely caused by CTCF-mediated enhancement of cleavage activity. This upstream peak signature and the ATAC footprint is less pronounced than previously reported (Buenrostro *et al.*, 2013).

7 k-mer mask optimization

Optimizing a k-mer mask is a balance between k-mer size and the degree to which enzymatic sequence bias is corrected. Choosing masks with more than 8 masked positions can result in an insufficient number of k-mers to accurately correct biases, because few sequences may exisit in some k-mer sequence bins. When more positions are included there is often has very little affect on the correction factor, so more information results in diminishing returns. Herein we describe two orthogonal methods to optimize k-mer masks using DNase and ATAC data as examples.

7.1 Hill-Climbing k-mer mask optimization of ATAC-seq data

The method takes a starting kmer-mask, a set of site tables (one table per TF), and at each step turns an masked position into an unmasked position, choosing the position that results in the lowest score. It iterates until there are no more unmasked positions. This approach requires many kmer-mask evaluations, which correspond to full runs of seqOutBias. It can run multiple instances of seqOutBias in parallel (see mc.cores parameter) if one has a machine with sufficient resources. At each step all possible positions to change an X to an N are evaluated and the one that results in the smallest score is chosen. For each PSWM, the standard deviation is computed for the profile obtained by summing, at each position in the PSWM, the scaled read counts across all sites. This scoring metric, which is the sum of these standard deviations across the set of PSWMs, is used to define the next position in the mask. Herein, we exclusively use the plus strand aligned reads.

```
load.sites <- function(filenames) {
    lapply(filenames, function(filename) {
        fimo = read.table(filename)
        bed6 = fimo[, c(2,3,4,1,6,5)]
        bed6
    })
}

sites = load.sites(c("~/ATAC_Walavalkar/CTCF_fimo.txt",
    "~/DNase_ENCODE/Elf1_fimo.txt",
    "~/DNase_ENCODE/Gata3_fimo.txt",
    "~/DNase_ENCODE/Max_fimo.txt",
    "~/ATAC_Walavalkar/SP1_fimo.txt",
    "~/ATAC_Walavalkar/EBF1_fimo.txt",
    "~/ATAC_Walavalkar/REST_fimo.txt"))

names(sites) = c("CTCF", "Elf1", "Gata3", "Max", "SP1", "EBF1", "REST")

source("https://raw.githubusercontent.com/guertinlab/seqOutBias/master/docs/R/seqOutBias_hcsearch.R")

seqOutBias.args = "--read-size=76 ~/DNase_ENCODE/hg38.fa ~/ATAC_Walavalkar/C1_gDNA_PE1_plus.bam ~/ATAC_Walavalkar/C1_gDNA_PE2_plus.bam"
seqOutBias.cmd = "seqOutBias"

initial.mask = "XXXXXXXXXXXXCXXXXXXXXXXXX"

#this command can be interrupted when the user is satisfied with the mask:
result.table = hc.search(sites, initial.mask, seqOutBias.args, prefix = "runhc_", sqcmd = seqOutBias.cmd, mc.cores = 4)

system(paste('seqOutBias ', seqOutBias.args, ' --no-scale --out=runhc_XXXXXXXXXXXXCXXXXXXXXXXXX.tbl --bw=runhc_XXXXXXXXXXXXCXXXXXXXXXXXX.bw'))

hc.atac.cutmasks = c('XXXXXXXXXXXXCXXXXXXXXXXXX',
    'XXXXXXXXXXXXCXXXXXXNXXXXX',
    'XXXXXXXXXXXXCXXNXXXNXXXXX',
    'XXXXXXXXNXXXCXXNXXXNXXXXX',
    'XXXXXXXXNXXXCXXNXXXNNXXXX',
    'XXXXXXXXNXXXCXXNNXXNNXXXX',
    'XXXXXXNXNXXXCXXNNXXNNXXXX',
    'XXXXXXNXNXXXCXXNNXXNNXXNX',
    'XXXXXXNXNXXXCXXNNXNNNXXNX',
    'XXXXXXNNNXXXCXXNNXNNNXXNX',
    'XXXXXXNNNXNXCXXNNXNNNXXNX',
    'XXXXXXNNNXNXCXXNNXNNNNXNX')


em.scores.atac = mclapply(hc.atac.cutmasks, function(cutmask) {
    bw.paths = run.cutmask(cutmask, seqOutBias.args, sqcmd=seqOutBias.cmd, clean = FALSE,  prefix = "runhc_")
    bw.plus = load.bigWig(bw.paths[1])
    bw.minus = load.bigWig(bw.paths[2])
    eval.cutmask(sites, bw.plus, bw.minus)
}, mc.cores = 4)

save(em.scores.atac, file="em.scores.atac.hc.Rdata")

require(lattice)
```

```
hc.atac.cutmasks = factor(hc.atac.cutmasks, levels=hc.atac.cutmasks)
pdf("ATAC-kmer_optimization.pdf", useDingbats=FALSE, width=4, height=6)
dotplot(as.numeric(em.scores.atac) ~ hc.atac.cutmasks,
        pch = 19,
        cex =1,
        col = 'black',
        main = "Hill Climbing derived k-mer masks",
        xlab = 'Masked Positions',
        ylim = c(0, 90000),
        scales=list(x=list(rot=45)),
        ylab = expression(paste(Sigma, ' SDs between PSWM positions')))
#for each set of TF PSWMs we sum the intensity of signal at each position,
#then we take the standard deviation between positions.
#The final metric is a sum of these standard deviations.
dev.off()
```

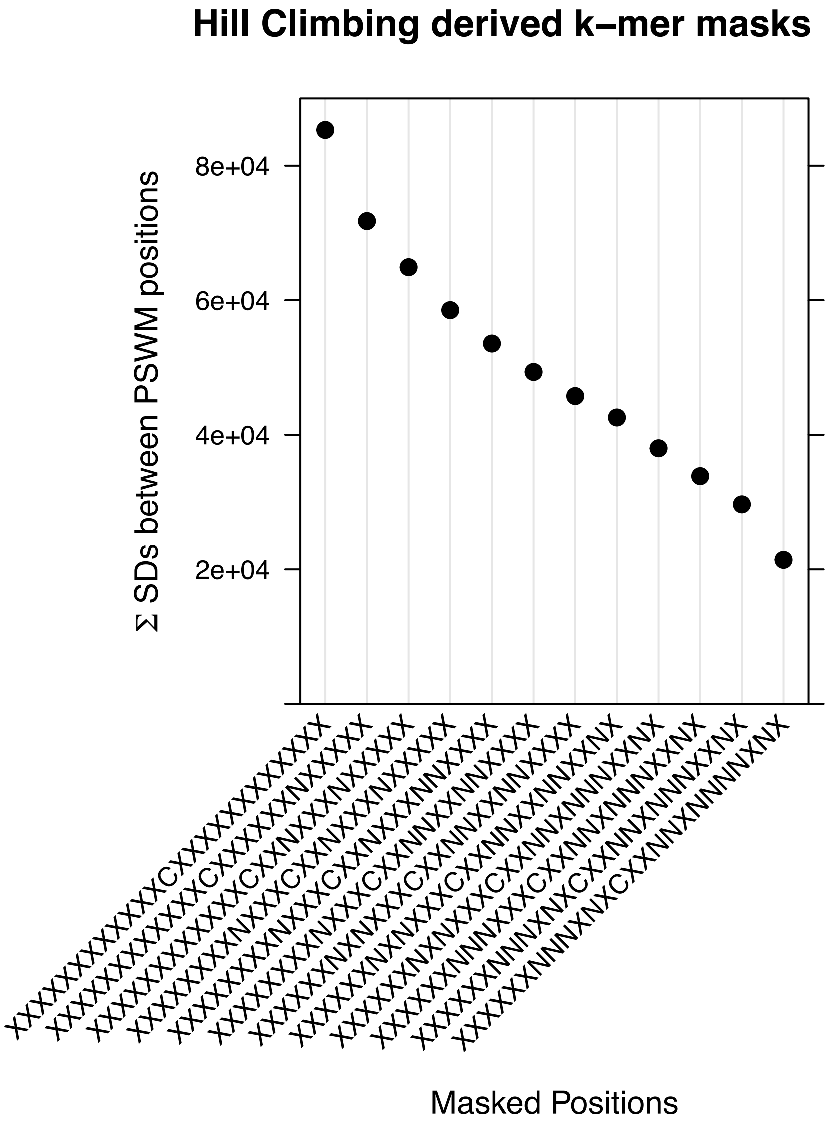


Figure 12: For each set of TF PSWMs we sum the intensity of signal at each position, then we take the standard deviation between positions. The final metric is a sum of these standard deviations. We chose to use the top 8 positions for the mask.

7.2 Expectation Maximization k-mer mask optimization of DNase-seq data

We developed a program that takes as input the k-mer count output of `seqOutBias table` and models the data as a set of binding sites, sharing a common motif, each with it's unknown orientation. This is a constrained version of MEME (Bailey *et al.*, 2006). This program will output a table of alternatives, where each row has more unmasked positions than the preceding rows. Unmasked positions are chosen by thresholding the information content of the resulting motif matrix. Running this with the "–verbose" flag, it will also output the PSWM. The resulting table has a column for the mask in the "forward" orientation and one where the mask in both directions is combined. The combined mask should be used if the data is not pre-split by strand. This approach is better suited for assays where a single enzyme with a preferred orientation cuts a particular site, producing reads in both directions, and one is interested in determining in more detail about what positions are influencing the choice of site.

The main disadvantages of this approach are: 1) it assumes that the mask is symmetric; 2) it requires a full counts table (all positions unmasked) as input; and 3) it requires multiple runs (automatically done in parallel) of the same computation, with random starting sites, to ensure a reasonably good global optimum for the motif PSWM.

The `kmer_mask_em` software is available at `https://github.com/guertinlab/kmer_mask_em`.

```
setwd('~/DNase_ENCODE')
system('seqOutBias table hg38_36.10.5.5.tbl UW_MCF7_both.bam > UW_MCF7_both_10.5.5.txt')
system('kmer_mask_em --verbose UW_MCF7_both_10.5.5.txt > UW_MCF7_both_10.5.5.verbose.txt')

read.cutmask <- function(filename) {
read.pwm <- function(lines, startMarker, mask = 10) {
    skipCount = which(lines == startMarker) + 1
    read.table(filename, skip=skipCount, nrows=mask, sep=' ',
               colClasses = c("character", "numeric", "numeric", "numeric", "numeric", "numeric"))
}

lines = readLines(filename)
pwm.simple = read.pwm(lines, "simple pwm:")
pwm.em = read.pwm(lines, "EM pwm:")

read.ic <- function(lines, startMarker) {
    skipCount = which(lines == startMarker)[2]
    read.table(filename, skip=skipCount, nrows=mask, sep=' ')
}

ic.mask = read.ic(lines, "IC    fwd_mask   merged_mask")

list(simple = pwm.simple, em = pwm.em, ic = ic.mask)
}

plot.pwm <- function(pwm) {
require(seqLogo)
seqLogo(makePWM(t(pwm[,2:5])))
}

pwms = read.cutmask("UW_MCF7_both_10.5.5.verbose.txt")

plot.pwm(pwms$em)

em.table = pwms$ic


ic = em.table[,1]

mask.ic = ic[seq(1,10,2)] + ic[seq(2,10,2)]

pdf("DNase-IC-kmer_EM.pdf", width=4, height=4)
dotplot(mask.ic ~ c('NCN','NXCXN','NXXCXXN','NXXXCXXXN','NXXXXCXXXXN'),
        pch = 19,
        cex =1,
        col = 'black',
        xlab = 'Masked Positions',
        scales=list(x=list(rot=30)),
        ylab = 'Information Content of N positions in mask')
dev.off()


pdf("DNase-IC-kmer_EM_cumsum.pdf", width=4, height=4)
dotplot(cumsum(mask.ic) ~ c('NCN','NNCNN','NNNCNNN','NNNNCNNNN','NNNNNCNNNNN'),
        pch = 19,
        cex =1,
        col = 'black',
        xlab = 'Masked Positions',
        scales=list(x=list(rot=30)),
        ylab = 'cumulative IC of N positions in mask')
dev.off()

em.cutmasks = as.character(em.table[,3])[seq(2,10,2)]

seqOutBias.args = "--read-size=36 ~/DNase_ENCODE/hg38.fa ~/DNase_ENCODE/UW_MCF7_both.bam"
seqOutBias.cmd = "seqOutBias"

em.scores.dnase.test = mclapply(em.cutmasks, function(cutmask) {
    bw.paths = run.cutmask(cutmask, seqOutBias.args, sqcmd=seqOutBias.cmd,
        prefix="run_", cleanup = FALSE)
    bw.plus = load.bigWig(bw.paths[1])
    bw.minus = load.bigWig(bw.paths[2])
    eval.cutmask(sites, bw.plus, bw.minus)
}, mc.cores = 4)

em.scores.dnase.10mer = em.scores.dnase
save(em.scores.dnase.10mer, file = 'em.scores.dnase.10mer.Rdata')

load("em.scores.dnase.10mer.Rdata")

system('seqOutBias ~/DNase_ENCODE/hg38.fa ~/DNase_ENCODE/UW_MCF7_both.bam --no-scale --read-size=36 --bw=runhc_XXXXXCXXXXX.bw')
tmp2 = eval.cutmask(sites,  load.bigWig('runhc_XXXXXCXXXXX.bw'), load.bigWig('runhc_XXXXXCXXXXX.bw'))


pdf("DNase-kmer_optimization.pdf", width=4, height=4)
dotplot(c(as.numeric(tmp2), as.numeric(em.scores.dnase)) ~ c('  uncorrected',' NCN','NNCNN','NNNCNNN','NNNNCNNNN','NNNNNCNNNNN'),
        pch = 19,
        cex =1,
        col = 'black',
        main = "EM derived k-mer masks",
        ylim = c(0, 410000),
        xlab = 'Masked Positions',
        scales=list(x=list(rot=30)),
        ylab = expression(paste(Sigma, ' SDs between PSWM positions')))
dev.off()
```

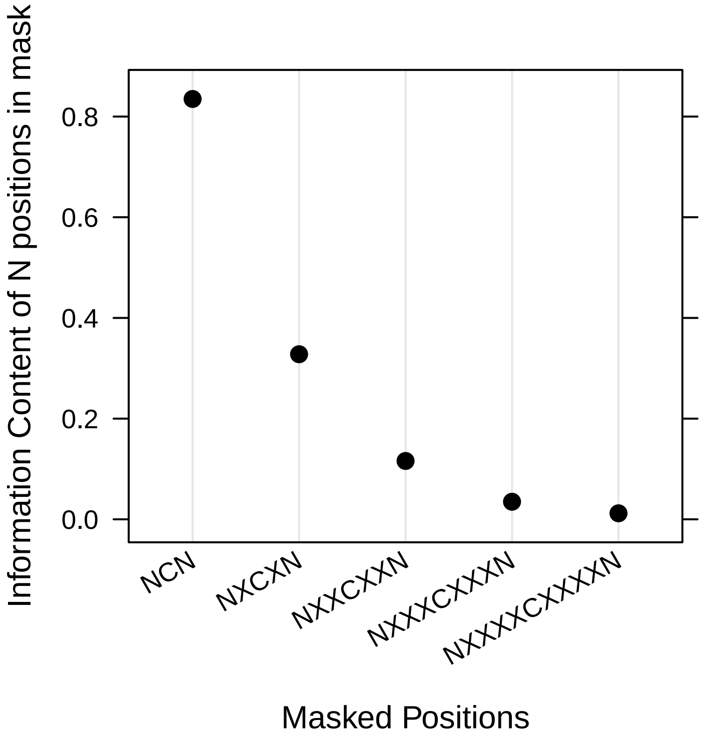


Figure 13: The positions directly flanking the DNase nick site have the most sequence information content and IC decreases moving away from the nick site.

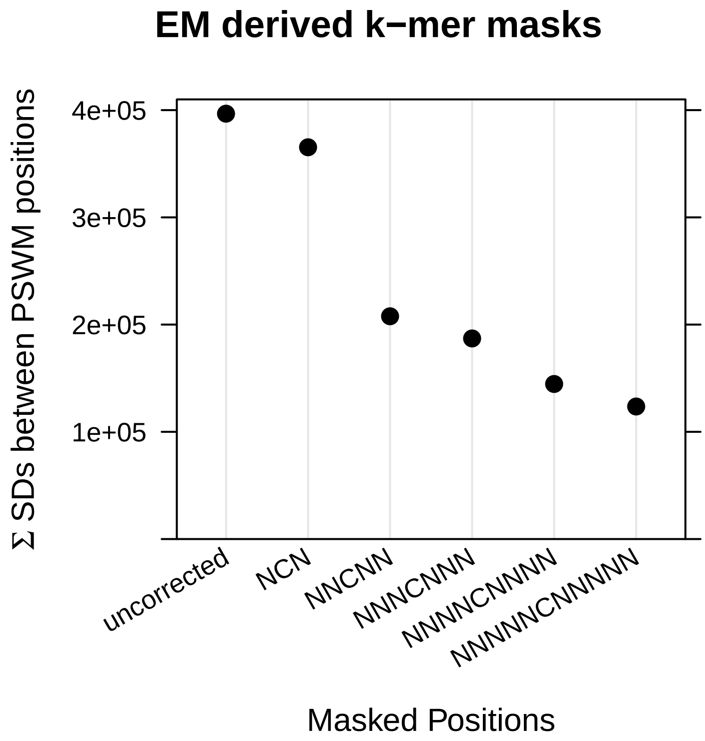


Figure 14: Using the same metric as described in Figure 12, we show that using more than two base on either side of the DNase nick site has minimal added advantage, as previously described (Sung *et al.*, 2014).

8 Characterizing the enzymatic clean up and ligation sequence bias

We had previously assumed that the three bases upstream and downstream of a DNase-nick site are equally likely to have adapters ligated and to be sequenced (Figure 2). However, we find that this is not the case and there is a bias in which 3-mer is ultimately detected by sequencing. Note that there is no inbalance for reverse palindromic 6-mers, for example: GCATGC

8.1 Plotting post-nicking enzymatic sequence biases

For each DNA nick we tally the number of times a plus strand and minus strand read detects the nick event. For simplicity, we assume that the mappability of plus and minus strand-aligned reads are the same.
The analysis below is exclusively for the plus strand analysis, but a minus or combined strand analysis gives the same results.

```
source('https://raw.githubusercontent.com/guertinlab/seqOutBias/master/docs/R/seqOutBias_functions.R')
counts.table = read.table('~/DNase_ENCODE/hg38_36.6.3.3.IMR90_Naked_DNase.txt')

ligation = cbind(counts.table, substring(counts.table[,2],1,3))
ligation[,8] = apply(ligation, 1, function(row) revcomp(row[7]))
ligation[,9] = substring(counts.table[,2],4,6)
colnames(ligation) = c(colnames(counts.table), 'V7', 'rcKmerUp','KmerDown')

mat = data.frame(matrix(nrow=64, ncol= 64))
count = 0
for (mer in unique(ligation$KmerDown)) {
    count = count + 1
    temp = ligation[ligation$KmerDown == mer,]
    rto = temp[,6]/temp[,5]
    mat[,count] = rto
    }
colnames(mat) = unique(ligation$KmerDown)
mat = do.call(data.frame,lapply(mat, function(x) replace(x, is.infinite(x),NA)))
rownames(mat) = unique(temp[,8])
mat = mat[order(rownames(mat)) , order(colnames(mat))]
mat = as.matrix(mat)

pdf('lig_bias_matrix.pdf', width=10.4, height=9.5)
heatmap.2(log(mat, base=10), col=colorpanel(30, "blue", "white","red"),
          symbreaks=T,scale="none",na.rm=TRUE,dendrogram = 'none', symm =TRUE,
          density.info=c("none"), key.xlab = expression('log'[10]*' ratio of bias (x-axis/y-axis)'),
          key.title= '',trace=c("none"),Rowv = FALSE, lhei=c(0.75,4), lwid = c(1.2, 4))
dev.off()

pdf('lig_bias_matrix_row.pdf', width=10.4, height=9.5)
heatmap.2(log(mat, base=10), col=colorpanel(30, "blue", "white","red"),
          symbreaks=T,scale="none",na.rm=TRUE,dendrogram = 'row', symm =TRUE,
          density.info=c("none"), key.xlab = expression('log'[10]*' ratio of bias (x-axis/y-axis)'),
          key.title= '',trace=c("none"), Rowv = TRUE, lhei=c(0.75,4), lwid = c(1.2, 4))
dev.off()

n.diag = mat
diag(n.diag) = NA
df = data.frame(x=c(n.diag, diag(mat)), group=factor(c(rep("non-Palindromic", length(n.diag)),
                                            rep("Palindromic", length(diag(mat))))))



pdf('lig_bias_bwplot.pdf', width=3, height=4)
trellis.par.set(box.umbrella = list(lty = 1, col="black", lwd=2),
                box.rectangle = list(col = 'black', lwd=1.6),plot.symbol = list(col='black', lwd=1.6, pch ='.'))
bwplot(log(x, base = 2) ~ group, df,
       scales=list(x=list(relation = "free", rot = 45)),
       ylab = expression('log'[2]*' ratio of detection bias'),
        pch = '|',
       col= 'black'
       )
dev.off()


pdf('avg_ligation_bias.pdf', width=12, height=4)
print(barchart((colMeans(mat, na.rm =TRUE)~colnames(mat)),
               col='grey85',
               ylim = c(0, max(colMeans(mat, na.rm =TRUE))+ 0.01 * max(colMeans(mat, na.rm =TRUE))),
               ylab=paste('relative ligation preference of each 3-mer', sep = ' '),
               xlab = '3-mer',
               origin = 0,
               scales=list(x=list(rot=45)),
               panel=function(...) {
                   panel.barchart(...)
                   panel.abline(h=1, lty= 2, col = 'grey40')
               }
               ))
dev.off()

#for (mer in unique(ligation$KmerDown)) {
for (mer in c('AAA', 'ATA', 'ATC', 'GAT','GAC')) {
    plot.barchart.lig(ligation[ligation$KmerDown == mer,],
                      filename=paste('ligation_bias_',mer,'.pdf', sep = ''), w = 12, h = 4)
}

ligation$ratio = ligation[,6]/ligation[,5]
x.ligation = ligation[with(ligation, order(ratio)), ]

pswm.func(head(x.ligation[,2], 205), out = 'low_205.txt', positions = 6)
pswm.func(tail(x.ligation[,2], 205), out = 'high_205.txt', positions = 6)
```

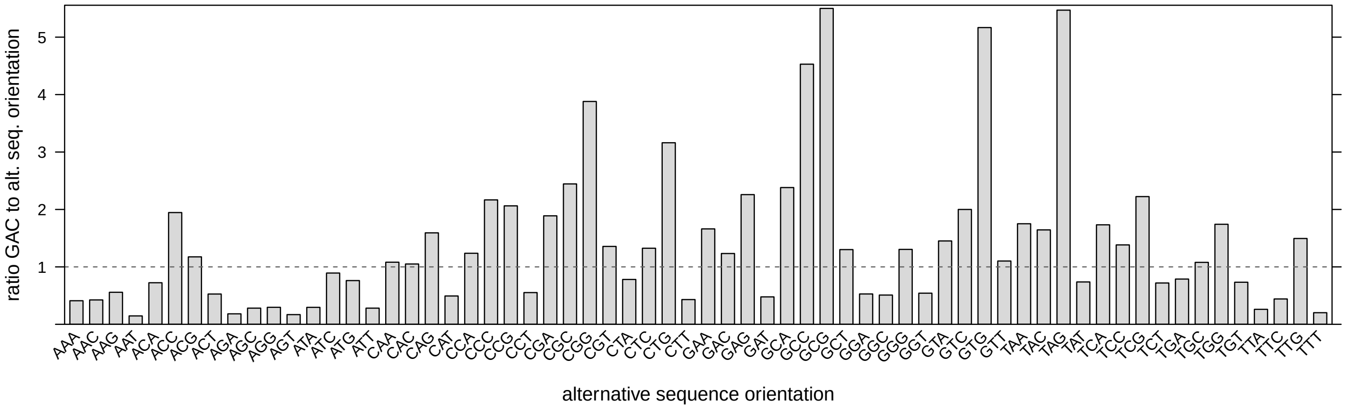


Figure 15: For all sequence-detected DNase-nicked 6-mers that end in 'GAC' we compare the ratio of sequence reads that start with 'GAC' ('**GAC**CAGATGACA' in Figure 2) to the oppositely oriented 3-mer ('**ATC**ATATCCCGT' in Figure 2).

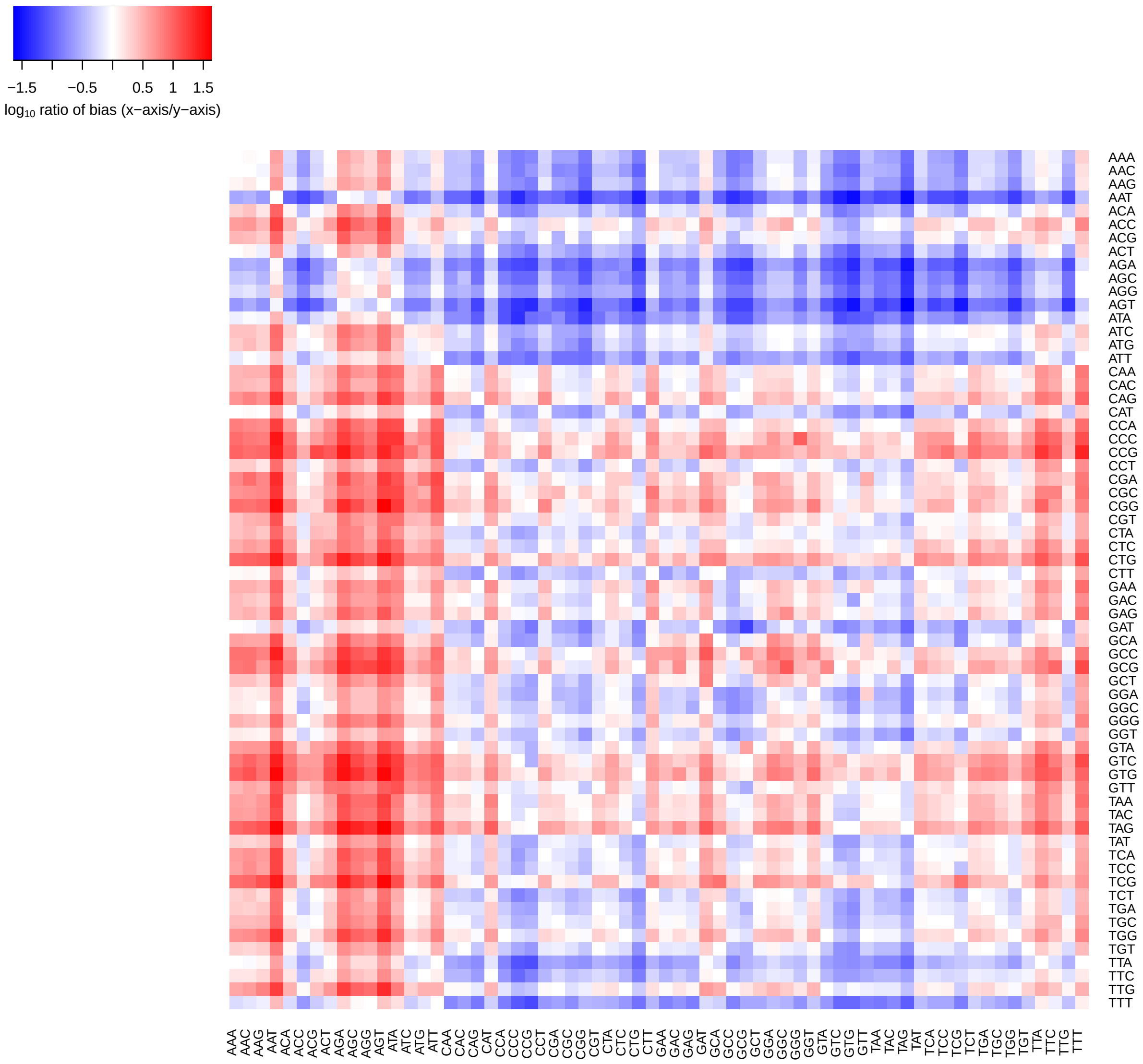


Figure 16: The relative bias of all 3-mers sequenced (the ratio of x-axis 3-mer to y-axis 3-mer). This bias results from enzymatic end repair and ligation sequence preference during the library preparation.

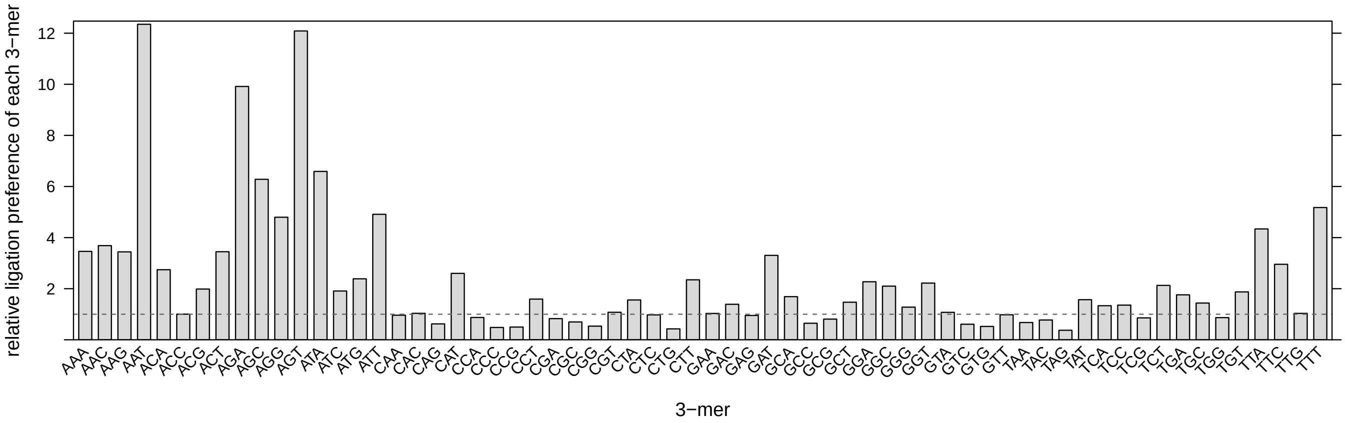


Figure 17: The ligation preference for each 3-mer relative to all 64 3-mers shows that ′AAT′ is the most preferred 3-mer relative to all others and ′TAG′ is least preferred. Note that this bar chart is the average of the exponentiation of each column in Figure 16.

8.2 Testing whether 3′ and 5′ ssDNA overhangs contribute to enzymatic sequence bias

Preparing digested DNA for Illumina high throughput sequencing requires several enzymatic treatments. T4 DNA Polymerase treatment blunts ends by 3′ overhang removal and 3′ recessed (5′ overhang) end fill-in. T4 Polynucleotide kinase phosphorylates the 5′ end and Klenow Fragment (3′ to 5′ exo-) adds an A 5′ overhang. We hypothesized that the ligation preference for each 3-mer relative to all other 3-mers is dictated by the overhanging sequence. Although 4 nick events are necessary to sequence a DNA molecule, we only detect one nick on each end of the molecule and it is impossible to determine the precise location of the other nicks. By assuming that two enzymes with similar nick specificity (Figure 18) will have comparable distribution of sequence overhangs, we can test the hypothesis that the overhang sequences are contribute to post-nicking enzymatic treatment biases. Therefore, we compared this post-nicking bias using DNase-seq data from two different labs and two different organisms (Figure 19 and Figure 20). We also compared the biases of Cyanase and Benzonase (Figure 19 and Figure 20), which have similar sequence preferences (Figure 18), although Cyanase and Benzonase are distinct enzymes. Benzonase is an endonuclease cloned from Serratia marcescens. Cyanase is within the same evolutionary family of alpha/alpha/beta folded nucleases as Benzonase, but Cyanase is cloned from a non-Serratia species. Cyanase is active as a monomer and Benzonase is active as a dimer.

```
seqOutBias table mm10_35.6.3.3.tbl mm10_liver_Cyanase.bam > mm10_35.6.3.3.liver_Cyanase.txt
seqOutBias table mm10_35.6.3.3.tbl mm10_liver_Benzonase.bam > mm10_35.6.3.3.liver_Benzonase.txt
seqOutBias table mm10_35.6.3.3.tbl mm10_liver_DNase.bam > mm10_35.6.3.3.liver_DNase.txt
```

```r
source('https://raw.githubusercontent.com/guertinlab/seqOutBias/master/docs/R/seqOutBias_functions.R')

setwd('~/TACh_Grontved')

counts.table.cyanase = read.table('mm10_35.6.3.3.liver_Cyanase.txt')
counts.table.benzonase = read.table('mm10_35.6.3.3.liver_Benzonase.txt')
counts.table.mm10.dnase = read.table('mm10_35.6.3.3.liver_DNase.txt')

totals.cyanase = colSums(counts.table.cyanase[,3:6])
scale.table.cyanase = data.frame(counts.table.cyanase[,1:2], t(apply(counts.table.cyanase[,3:6], 1,
    function(row) c((row[1]/totals[1]) / (row[3] / totals[3]), (row[2] / totals[2]) / (row[4] / totals[4])))))

totals.benzonase = colSums(counts.table.benzonase[,3:6])
scale.table.benzonase = data.frame(counts.table.benzonase[,1:2], t(apply(counts.table.benzonase[,3:6], 1,
    function(row) c((row[1]/totals[1]) / (row[3] / totals[3]), (row[2] / totals[2]) / (row[4] / totals[4])))))

totals.dnase = colSums(counts.table.mm10.dnase[,3:6])
scale.table.mm10.dnase = data.frame(counts.table.mm10.dnase[,1:2], t(apply(counts.table.mm10.dnase[,3:6], 1,
    function(row) c((row[1]/totals[1]) / (row[3] / totals[3]), (row[2] / totals[2]) / (row[4] / totals[4])))))

pdf('Cyanase_Benzonase_scale.pdf', width=4.5, height=4.5)
xyplot(log(scale.table.cyanase[,3], base = 10) ~ log(scale.table.benzonase[,3], base = 10),
       ylab = expression('log'[10]*'(Cyanase Scale Factor)'),
       xlab = expression('log'[10]*'(Benzonase Scale Factor)'),
       panel = function(x, y) {
           panel.xyplot(x, y,pch= 16, cex =0.5, col = 'black')
           panel.text(0.2*max(x), 0.95*min(y), label=paste('R = ', round(cor(x, y),2), sep =''))
       })
dev.off()

scale.table.dnase = do.call(data.frame,lapply(scale.table, function(x) replace(x, is.infinite(x),NA)))

pdf('DNase_Benzonase_scale.pdf', width=4.5, height=4.5)
xyplot(log(scale.table.dnase[,3], base = 10) ~ log(scale.table.benzonase[,3], base = 10),
       ylab = expression('log'[10]*'(MCF7 DNase Scale Factor)'),
       xlab = expression('log'[10]*'(Benzonase Scale Factor)'),
       panel = function(x, y) {
           panel.xyplot(x, y,pch= 16, cex =0.5, col = 'black')
           panel.text(-1.7, -1, label=paste('R = ', round(cor(x, y, use = 'complete.obs'),2), sep =''))

       })
dev.off()

pdf('DNase_DNase_mm10_scale.pdf', width=4.5, height=4.5)
xyplot(log(scale.table.dnase[,3], base = 10) ~ log(scale.table.mm10.dnase[,3], base = 10),
       ylab = expression('log'[10]*'(MCF7 DNase Scale Factor)'),
       xlab = expression('log'[10]*'(mouse liver DNase Scale Factor)'),
       panel = function(x, y) {
           panel.xyplot(x, y,pch= 16, cex =0.5, col = 'black')
           panel.text(0.4, -1, label=paste('R = ', round(cor(x, y, use = 'complete.obs'),2), sep =''))

       })
```

```
dev.off()

cyanase.mat = matrix.func(counts.table.cyanase)
benzonase.mat = matrix.func(counts.table.benzonase)
dnase.mcf7.mat = matrix.func(counts.table)
dnase.mm10.mat = matrix.func(counts.table.mm10.dnase)

pdf('lig_bias_matrix_benzonase.pdf', width=10.4, height=9.5)
heatmap.2(log(benzonase.mat, base=10), col=colorpanel(100, "blue", "white","red"),
          symbreaks=T,scale="none",na.rm=TRUE,dendrogram = 'none', symm =TRUE,
          density.info=c("none"), key.xlab = expression('log'[10]*' ratio of bias (x-axis/y-axis)'),
          key.title= '',trace=c("none"),Rowv =FALSE, lhei=c(0.75,4), lwid = c(1.2, 4))
dev.off()
pdf('lig_bias_matrix_cyanase.pdf', width=10.4, height=9.5)
heatmap.2(log(cyanase.mat, base=10), col=colorpanel(100, "blue", "white","red"),
          symbreaks=T,scale="none",na.rm=TRUE,dendrogram = 'none', symm =TRUE,
          density.info=c("none"), key.xlab = expression('log'[10]*' ratio of bias (x-axis/y-axis)'),
          key.title= '',trace=c("none"),Rowv =FALSE, lhei=c(0.75,4), lwid = c(1.2, 4))
dev.off()
pdf('lig_bias_matrix_dnase_mcf7.pdf', width=10.4, height=9.5)
heatmap.2(log(dnase.mcf7.mat, base=10), col=colorpanel(100, "blue", "white","red"),
          symbreaks=T,scale="none",na.rm=TRUE,dendrogram = 'none', symm =TRUE,
          density.info=c("none"), key.xlab = expression('log'[10]*' ratio of bias (x-axis/y-axis)'),
          key.title= '',trace=c("none"),Rowv =FALSE, lhei=c(0.75,4), lwid = c(1.2, 4))
dev.off()
pdf('lig_bias_matrix_dnase_mm_liver.pdf', width=10.4, height=9.5)
heatmap.2(log(dnase.mm10.mat, base=10), col=colorpanel(100, "blue", "white","red"),
          symbreaks=T,scale="none",na.rm=TRUE,dendrogram = 'none', symm =TRUE,
          density.info=c("none"), key.xlab = expression('log'[10]*' ratio of bias (x-axis/y-axis)'),
          key.title= '',trace=c("none"),Rowv =FALSE, lhei=c(0.75,4), lwid = c(1.2, 4))
dev.off()


pdf('DNase_DNase_comparison_post_nick.pdf', width=4.5, height=4.5)
xyplot(as.numeric(unlist(log(dnase.mm10.mat, base = 10))) ~ as.numeric(unlist(log(dnase.mcf7.mat, base = 10))),
       ylab = expression('log'[10]*'(mouse liver DNase ratio bias)'),
       xlab = expression('log'[10]*'(MCF7 DNase ratio bias)'),
       panel = function(x, y) {
           panel.xyplot(x, y,pch= 16, cex =0.5, col = 'black')
           panel.text(1, -1.5, label=paste('R = ', round(cor(x, y, use = 'complete.obs'),2), sep =''))

       })
dev.off()


pdf('Cyanase_Benzoase_comparison_post_nick.pdf', width=4.5, height=4.5)
xyplot(as.numeric(unlist(log(cyanase.mat, base = 10))) ~ as.numeric(unlist(log(benzonase.mat, base = 10))),
       ylab = expression('log'[10]*'(Cyanase ratio bias)'),
       xlab = expression('log'[10]*'(Benzonase ratio bias)'),
       panel = function(x, y) {
           panel.xyplot(x, y,pch= 16, cex =0.5, col = 'black')
           panel.text(1, -1.5, label=paste('R = ', round(cor(x, y, use = 'complete.obs'),2), sep =''))

       })
dev.off()

pdf('DNase_Benzoase_comparison_post_nick.pdf', width=4.5, height=4.5)
xyplot(as.numeric(unlist(log(dnase.mcf7.mat, base = 10))) ~ as.numeric(unlist(log(benzonase.mat, base = 10))),
       ylab = expression('log'[10]*'(MCF7 DNase ratio bias)'),
       xlab = expression('log'[10]*'(Benzonase ratio bias)'),
       panel = function(x, y) {
           panel.xyplot(x, y,pch= 16, cex =0.5, col = 'black')
           panel.text(1, -1.5, label=paste('R = ', round(cor(x, y, use = 'complete.obs'),2), sep =''))

       })
dev.off()
```

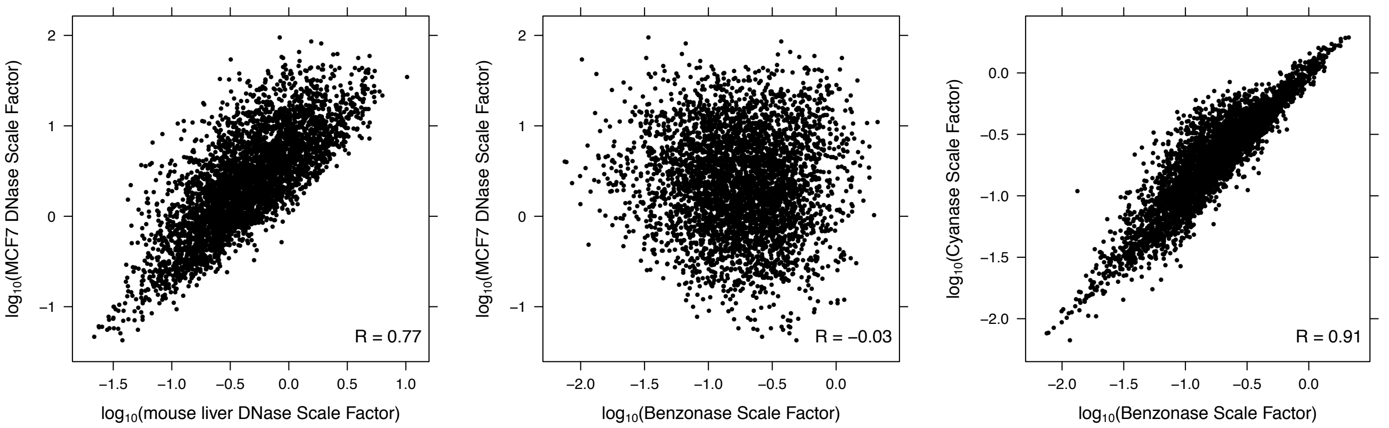


Figure 18: These indicate that the enzymatic nick biases, as measured by the `seqOutBias` scale factor, are correlated between DNase experiments and correlated between Cyanase and Benzonase (Grøntved *et al.*, 2012). Dr. John Stamatoyannopoulos' lab generated the MCF7 DNase-seq data (Neph *et al.*, 2012) and Dr. Gordon Hager's lab generated the mouse liver data (Grøntved *et al.*, 2012))

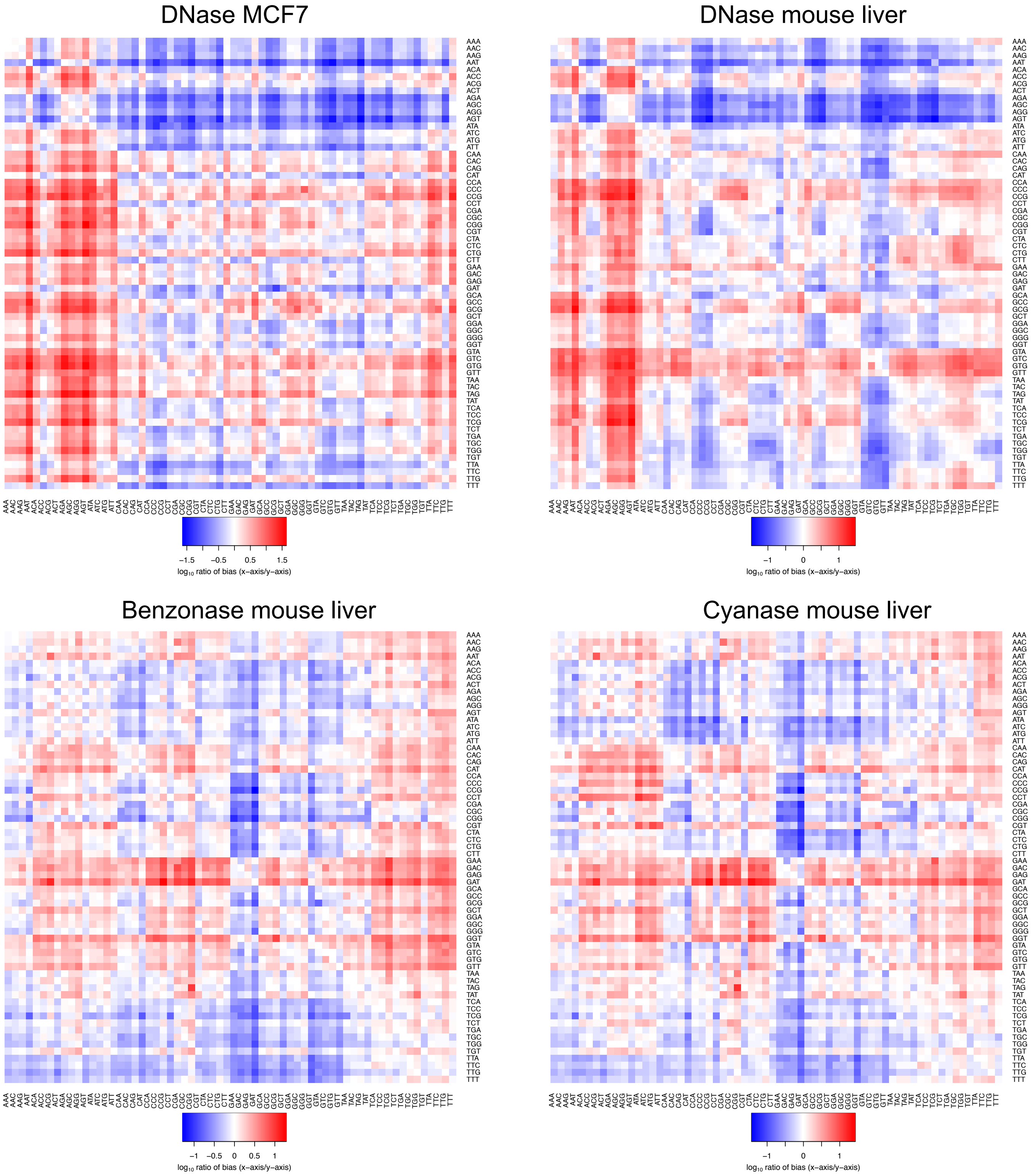


Figure 19: These heatmaps illustrate that the post-nick enzymatic processing biases of DNase are very similar between two labs (MCF7 data from John Stamatoyannopoulos' lab and mouse liver data is from Gordon Hager's lab). Likewise, the post-nick biases of Cyanase and Benzonase have similar patterns.

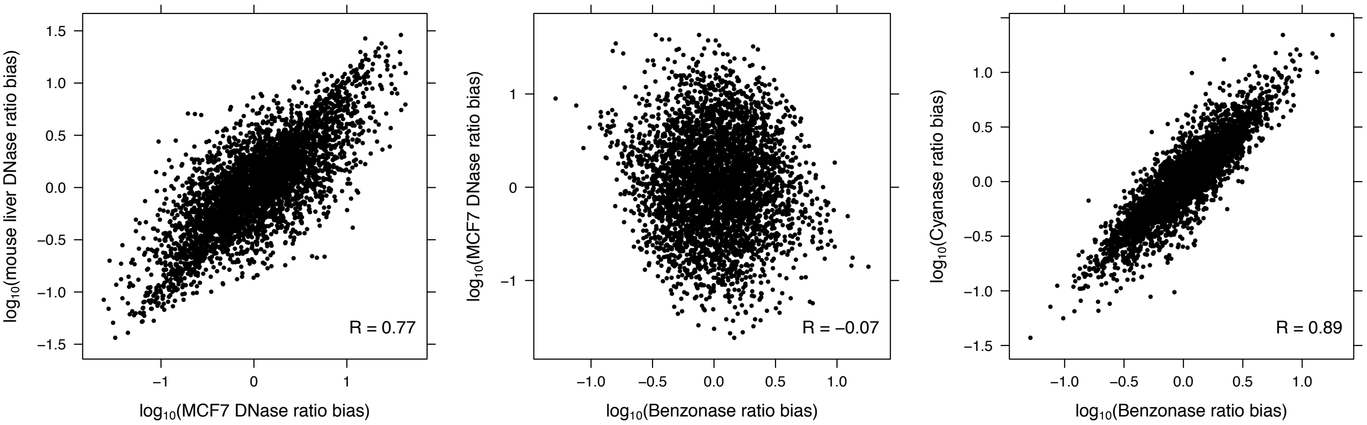


Figure 20: This figure plots the values from Figure 19. The post-nick sequence preferences are highly correlated between DNase-seq experiments and between Benzonase and Cyanase experiments, but not between DNase and Benzonase.

In conclusion, we show that `seqOutBias` successfully corrects sequence bias associated with many molecular genomics techniques and our analysis indicate that the enzymes that are common to many library prepartion protocols exhibit previously uncharacterized biases.

References

Bailey TL, Boden M, Buske FA, Frith M, Grant CE, Clementi L, Ren J, Li WW, Noble WS (2009). "MEME SUITE: tools for motif discovery and searching." *Nucleic acids research*, p. gkp335.

Bailey TL, Williams N, Misleh C, Li WW (2006). "MEME: discovering and analyzing DNA and protein sequence motifs." *Nucleic acids research*, **34**(suppl 2), W369–W373.

Buenrostro JD, Giresi PG, Zaba LC, Chang HY, Greenleaf WJ (2013). "Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position." *Nature methods*, **10**(12), 1213–1218.

Core LJ, Martins AL, Danko CG, Waters CT, Siepel A, Lis JT (2014). "Analysis of nascent RNA identifies a unified architecture of initiation regions at mammalian promoters and enhancers." *Nature genetics*, **46**(12), 1311–1320.

Core LJ, Waterfall JJ, Lis JT (2008). "Nascent RNA sequencing reveals widespread pausing and divergent initiation at human promoters." *Science*, **322**(5909), 1845–1848.

Grant CE, Bailey TL, Noble WS (2011). "FIMO: scanning for occurrences of a given motif." *Bioinformatics*, **27**(7), 1017–1018.

Grøntved L, Bandle R, John S, Baek S, Chung HJ, Liu Y, Aguilera G, Oberholtzer C, Hager GL, Levens D (2012). "Rapid genome-scale mapping of chromatin accessibility in tissue." *Epigenetics & chromatin*, **5**(1), 1.

Grøntved L, John S, Baek S, Liu Y, Buckley JR, Vinson C, Aguilera G, Hager GL (2013). "C/EBP maintains chromatin accessibility in liver and facilitates glucocorticoid receptor recruitment to steroid response elements." *The EMBO journal*, **32**(11), 1568–1583.

He HH, Meyer CA, Chen MW, Zang C, Liu Y, Rao PK, Fei T, Xu H, Long H, Liu XS, *et al.* (2014). "Refined DNase-seq protocol and data analysis reveals intrinsic bias in transcription factor footprint identification." *Nature methods*, **11**(1), 73–78.

Karolchik D, Barber GP, Casper J, Clawson H, Cline MS, Diekhans M, Dreszer TR, Fujita PA, Guruvadoo L, Haeussler M, *et al.* (2014). "The UCSC genome browser database: 2014 update." *Nucleic acids research*, **42**(D1), D764–D770.

Kwak H, Fuda NJ, Core LJ, Lis JT (2013). "Precise maps of RNA polymerase reveal how promoters direct initiation and pausing." *Science*, **339**(6122), 950–953.

Langmead B, Trapnell C, Pop M, Salzberg S (2009). "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome." *Genome Biology*, **10**(3), R25. ISSN 1465-6906. doi: 10.1186/gb-2009-10-3-r25. URL `http://genomebiology.com/2009/10/3/R25`.

Lazarovici A, Zhou T, Shafer A, Machado ACD, Riley TR, Sandstrom R, Sabo PJ, Lu Y, Rohs R, Stamatoyannopoulos JA, *et al.* (2013). "Probing DNA shape and methylation state on a genomic scale with DNase I." *Proceedings of the National Academy of Sciences*, **110**(16), 6376–6381.

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, *et al.* (2009). "The sequence alignment/map format and SAMtools." *Bioinformatics*, **25**(16), 2078–2079.

Neph S, Vierstra J, Stergachis AB, Reynolds AP, Haugen E, Vernot B, Thurman RE, John S, Sandstrom R, Johnson AK, Maurano MT, Humbert R, Rynes E, Wang H, Vong S, Lee K, Bates D, Diegel M, Roach V, Dunn D, Neri J, Schafer A, Hansen RS, Kutyavin T, Giste E, Weaver M, Canfield T, Sabo P, Zhang M, Balasundaram G, Byron R, MacCoss MJ, Akey JM, Bender MA, Groudine M, Kaul R, Stamatoyannopoulos JA (2012). "An expansive human regulatory lexicon encoded in transcription factor footprints." *Nature*, **489**(7414), 83–90.

Quinlan AR, Hall IM (2010). "BEDTools: a flexible suite of utilities for comparing genomic features." *Bioinformatics*, **26**(6), 841–842.

Seo YK, Chong HK, Infante AM, Im SS, Xie X, Osborne TF (2009). "Genome-wide analysis of SREBP-1 binding in mouse liver chromatin reveals a preference for promoter proximal binding to a new motif." *Proceedings of the National Academy of Sciences*, **106**(33), 13765–13769.

Stamatoyannopoulos JA, Snyder M, Hardison R, Ren B, Gingeras T, Gilbert DM, Groudine M, Bender M, Kaul R, Canfield T, *et al.* (2012). "An encyclopedia of mouse DNA elements (Mouse ENCODE)." *Genome biology*, **13**(8), 1.

Sung MH, Guertin MJ, Baek S, Hager GL (2014). "DNase footprint signatures are dictated by factor dynamics and DNA sequence." *Molecular cell*, **56**(2), 275–285.

Thomas CA (1956). "The Enzymatic Degradation of Desoxyribose Nucleic Acid." *J. Am. Chem. Soc.*, **78**(9), 1861–1868.

Vierstra J, Stamatoyannopoulos JA (2016). "Genomic footprinting." *Nat. Methods*, **13**(3), 213–221.

Yardımcı GG, Frank CL, Crawford GE, Ohler U (2014). "Explicit DNase sequence bias modeling enables high-resolution transcription factor footprint detection." *Nucleic acids research*, p. gku810.