seqOutATACBias: Rule Ensemble Correction of ATAC-seq Data Vignette

Jacob B. Wolpe* Michael J. Guertin†

Abstract

This vignette shows an example workflow which applies a 12 input mask rule ensemble model to correct Tn5 insertion sequence bias in ATAC-seq data. To start, example reads and reference genome are downloaded from cyverse. seqOutBias is next ran 13 times to generate unscaled read depth and input for the modeling. The rule ensemble model is then applied to the input data. Finally, the signal at ESR1 motifs in the test chromosome (chromosome 21) are plotted and compared with unscaled and seqOutBias output.

Contents

*Department of Biochemistry and Molecular Genetics, University of Virginia, Charlottesville, Virginia, United States of America

†Department of Genetics and Genome Sciences, University of Connecticut, Farmington, Connecticut, United States of America

1 Foreword

The analysis should take about 15 minutes to complete using chromosome 21 and require about 650 Mb of disk space.

This workflow shows the method of correcting ATAC-seq bias using a 12 input mask rule ensemble model. Input is a pre-aligned BAM file containing reads mapping to chromosome 21 from the SRR5123141 data set and hg38 reference genome chromosome 21. We recommend a fresh install of seqOutBias if it has not been installed since 07/02/22. This model was trained and created using the methods in the accompanying paper. Because this is a workflow, it requires several prerequisites to be in PATH or installed as a package in R.

2 Installations

In order to run this vignette, you must have the following installed and added to PATH:
seqOutBias (https://github.com/guertinlab/seqOutBias/archive/refs/heads/master.zip)
Rust >= 1.32.0
genometools >= 1.6.1
pyfaidx >= 0.7.1
GNU parallel >= 20220722
GNU wget >= 1.21.3
bedtools >= 2.30.0
bigWigToBedGraph >= 438
bedGraphToBigWig >= 2.9
wigToBigWig >= 2.8 Pandoc >= 2.19.2
R >= 4.2.1
R Packages:
- R data.table package >= 1.14.2
- bigWig R package

Check to see if you have the required dependencies in PATH. The following will print a message if a dependency cannot be called:

```
if ! command -v wget &> /dev/null
then
    echo "wget could not be found"
elif ! command -v faidx &> /dev/null
then
    echo "faidx could not be found"
elif ! command -v parallel &> /dev/null
then
    echo "GNU parallel could not be found"
elif ! command -v bigWigToBedGraph &> /dev/null
then
    echo "bigWigToBedGraph could not be found"
elif ! command -v bedGraphToBigWig &> /dev/null
then
    echo "bedGraphToBigWig could not be found"
elif ! command -v gt &> /dev/null
then
    echo "Genome tools could not be found"
elif ! command -v rustc &> /dev/null
then
    echo "Rust could not be found"
elif ! command -v seqOutBias &> /dev/null
then
    echo "seqOutBias could not be found"
elif ! command -v wigToBigWig &> /dev/null
then
    echo "wigToBigWig could not be found"
else
    echo "Checked dependencies installed"
fi
```

```
## Checked dependencies installed
```

If you find that any of these dependencies are not in PATH, you may install them from the following:

seqOutBias: https://github.com/guertinlab/seqOutBias/archive/refs/heads/master.zip
Rust: https://www.rust-lang.org/
genometools: http://genometools.org/
R: https://rstudio-education.github.io/hopr/starting.html
pyfaidx: https://pypi.org/project/pyfaidx/
GNU parallel: https://www.gnu.org/software/parallel/
bedtools: https://bedtools.readthedocs.io/en/latest/
bigWigToBedGraph: http://hgdownload.soe.ucsc.edu/admin/exe/
bedGraphToBigWig: http://hgdownload.soe.ucsc.edu/admin/exe/
bigWig R package: https://github.com/guertinlab/bigWig
wigToBigWig: https://anaconda.org/bioconda/ucsc-wigtobigwig
GNU wget: https://www.gnu.org/software/wget/

2.1 Auto-install R packages

Install the `data.table`, `bigWig`, and `devtools` R packages, if necessary:

```
tabletest = require(data.table)
```

```
## Loading required package: data.table
```

```
if(tabletest==FALSE){
  install.packages('data.table')
}
bigWigtest = require(bigWig)
```

```
## Loading required package: bigWig
```

```
if(bigWigtest==FALSE){
  install.packages('devtools')
  devtools::install_github("andrelmartins/bigWig", subdir="bigWig")
}
```

3 Generating scaled seqOutBias output for rule ensemble implementation

This section prepares the input data necessary for rule ensemble scaling of seqOutBias output. The first section downloads the chromosome 21 reference genome (hg38) and aligned, unscaled chromosome 21 read files in BAM format, from cyverse. Next, run seqOutBias with no scaling in order to generate the necessary suffixerator and tallymer reference files for future runs in addition to getting a raw read depth count.

3.1 Downloading reference genome and read data.

Download the reference genome (hg38) and aligned, deproteinized ATAC-seq read file from cyverse.

```
#To test this vignette with a subset (chr 21) genome and reads:
wget -nv https://data.cyverse.org/dav-anon/iplant/home/jacobwolpe/C1_gDNA_rep1_chr21.bam
wget -nv https://data.cyverse.org/dav-anon/iplant/home/jacobwolpe/hg38_chr21.fa
```

```
## 2022-12-13 13:16:17 URL:https://data.cyverse.org/dav-anon/iplant/home/jacobwolpe/C1_gDNA_rep1_chr21.bam [20970
## 2022-12-13 13:16:33 URL:https://data.cyverse.org/dav-anon/iplant/home/jacobwolpe/hg38_chr21.fa [47488490/47488
```

3.2 Initial run of seqOutBias to generate suffix and tallymer files

This initial run of seqOutBias will take some time, as it generates all suffix (.sft) and tallymer (.tal) reference files necessary for bias correction using this reference genome and data set. Subsequent runs are faster and may be done in parallel, using the same reference files. To preserve space, we delete the output .tbl file once the run is complete, as it is no longer needed and very large.

```
#Run seqOutBias with unscaled parameters to generate reference files and unscaled read depth:
seqOutBias hg38_chr21.fa C1_gDNA_rep1_chr21.bam --read-size=76 --no-scale \
            --strand-specific --custom-shift=4,-4 --bed=C1_gDNA_rep1_chr21_scaling_test.bed \
            --bw=C1_gDNA_rep1_chr21_unscaled.bigWig
#Remove large .tbl file:
rm hg38_chr21_76.4.2.2.tbl
```

```
## ################### Creating mappability file using tallymer ###################
## # dna=yes
## # indexname="hg38_chr21.sft"
## # prefixlength=automatic
## # storespecialcodes=false
## # inputfile[0]=hg38_chr21.fa
## # indexname=hg38_chr21.sft
## # outtistab=true,outsuftab=true,outlcptab=true,outbwttab=false,outbcktab=false,outdestab=true,outsdstab=true,o
## # parts=4
## # maxinsertionsort=3
## # maxbltriesort=1000
## # maxcountingsort=4000
## # lcpdist=false
## # sizeof (GtUword)=64
## # wildcardranges of length 1=4
## # wildcardranges of length 10=4
## # wildcardranges of length 20=1
## # wildcardranges of length 100=13
## # wildcardranges of length 10000=1
## # wildcardranges of length 50000=25
## # wildcardranges of length 100000=2
## # wildcardranges of length 150000=1
## # wildcardranges of length 5010000=1
## # init character encoding (uint32, 11678192 bytes, 2.00 bits/symbol)
## # totallength=46709983
## # numofsequences=1
## # specialcharacters=6621364
## # specialranges=52
## # realspecialranges=52
## # wildcards=6621364
## # wildcardranges=52
## # realwildcardranges=52
## # occurrences(a)=11820664
## # occurrences(c)=8185244
## # occurrences(g)=8226381
## # occurrences(t)=11856330
## # automatically determined prefixlength=10
## # maxinsertionsort=3
## # maxbltriesort=1000
## # maxcountingsort=4000
## # storespecialcodes=false
## # cmpcharbychar=false
## # totallength=46709983
## # sizeof (leftborder)=4194308 bytes
## # sizeof (countspecialcodes)=1048576 bytes
## # sizeof (distpfxidx)=349520 bytes
## # sizeof (bcktab)=5592404 bytes
## # largest bucket size=35844
## # widthofpart[0]=10022175
## # widthofpart[1]=10022207
## # widthofpart[2]=10022099
## # widthofpart[3]=10022138
## # create suffix_sort_space: suftab uses 64bit values: maxvalue=46709983,numofentries=10022207
## # compute part 0: 10022175 suffixes,228270 buckets from 0..228269
## # used workspace for sorting: 0.11 MB
## # countinsertionsort=20391
## # countbltriesort=200224
## # countcountingsort=403
## # countshortreadsort=0
## # countradixsort=0
## # counttqsort=36
## # compute part 1: 10022207 suffixes,296263 buckets from 228270..524532
## # countinsertionsort=33162
## # countbltriesort=241342
## # countcountingsort=353
## # countshortreadsort=0
## # countradixsort=0
## # counttqsort=22
## # compute part 2: 10022099 suffixes,312559 buckets from 524533..837091
## # countinsertionsort=32166
## # countbltriesort=263259
## # countcountingsort=337
## # countshortreadsort=0
## # countradixsort=0
## # counttqsort=21
## # compute part 3: 10022138 suffixes,211484 buckets from 837092..1048575
## # countinsertionsort=19282
## # countbltriesort=186005
## # countcountingsort=435
## # countshortreadsort=0
## # countradixsort=0
## # counttqsort=35
## # construct mer buckets for prefixlength 7
## # numofcodes = 16384
## # indexfilename = hg38_chr21.tal_76
## # alphasize = 4
## # mersize = 76
## # numofmers = 1099551
## # merbytes = 19
## ################################################################################
## # tallymer produced/found hg38_chr21.tal_76.gtTxt.gz
## # kmer-size: 4
## # plus-offset: 2
## # minus-offset: 2
## # chrom: "chr21"
## # - 46709984 bases
## # seqtable produced hg38_chr21_76.4.2.2.tbl
## # tabulate C1_gDNA_rep1_chr21.bam
## # scale C1_gDNA_rep1_chr21.bam
## # scale produced C1_gDNA_rep1_chr21_scaling_test_not_scaled.bed
## # scale produced C1_gDNA_rep1_chr21_unscaled.bigWig
```

3.3 Multiple runs of seqOutBias to generate rule ensemble modeling input

Now run seqOutBias in parallel to generate the 12 input masks for rule ensemble scaling. Once each mask is generated, the large .tbl files are also deleted.

```
#Masks necessary for rule ensemble implementation
masks=("XXXXXXXXXXXXXXXXXNNNNNNCNXXXXXXXXXXXXXXXXXXXXXX"
"XXXXXXXXXXXXXXXXXXNNNNNCNNXXXXXXXXXXXXXXXXXXXXX"
"XXXXXXXXXXXXXXXXXXXNNNNCNNNXXXXXXXXXXXXXXXXXXXX"
"XXXXXXXXXXXXXXXXXXXXNNNCNNNNXXXXXXXXXXXXXXXXXXX"
"XXXXXXXXXXXXXXXXXXXXXNNCNNNNNXXXXXXXXXXXXXXXXXX"
"XXXXXXXXXXXXXXXXXXXXXXXCNNNNNNNXXXXXXXXXXXXXXXX"
"XXXXXXXXXXXXXXXXXXXXXXXCXXNNNNNNNXXXXXXXXXXXXXX"
"XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXNNNNNNNXXXXXXXXX"
"XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXNNNNNNNXXXXXXXX"
"XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXNNNNNNNXXXXXXX"
"XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXXNNNNNNNXXXXXX"
"XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXXXXNNNNNNNXXXX")
printf "%s\n" "${masks[@]}" > masks.txt
#Run seqOutBias on the rest of the masks, in parallel and remove .tbl files to conserve space
parallel -j3 'seqOutBias hg38_chr21.fa C1_gDNA_rep1_chr21.bam --read-size=76 \
      --strand-specific --custom-shift=4,-4 \
      --kmer-mask={} --bed=C1_gDNA_rep1_chr21_{}.bed \
      --out=C1_gDNA_rep1_chr21_{}.tbl --bw=C1_gDNA_rep1_chr21_{}.bigWig
      echo Cleaning up C1_gDNA_rep1_chr21_{}
      rm C1_gDNA_rep1_chr21_{}.tbl
      rm C1_gDNA_rep1_chr21_{}_scaled.bed' ::: ${masks[@]}
```

```
## # tallymer produced/found hg38_chr21.tal_76.gtTxt.gz
## # kmer-size: 46
## # plus-offset: 23
## # minus-offset: 23
## # chrom: "chr21"
## # - 46709984 bases
## # seqtable produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXNNNNCNNNXXXXXXXXXXXXXXXXXXXX.tbl
## # tabulate C1_gDNA_rep1_chr21.bam
## # scale C1_gDNA_rep1_chr21.bam
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXNNNNCNNNXXXXXXXXXXXXXXXXXXXX_scaled.bed
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXNNNNCNNNXXXXXXXXXXXXXXXXXXXX.bigWig
## Cleaning up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXNNNNCNNNXXXXXXXXXXXXXXXXXXXX
## # tallymer produced/found hg38_chr21.tal_76.gtTxt.gz
## # kmer-size: 46
## # plus-offset: 23
## # minus-offset: 23
## # chrom: "chr21"
## # - 46709984 bases
## # seqtable produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXNNNNNNCNXXXXXXXXXXXXXXXXXXXXXX.tbl
## # tabulate C1_gDNA_rep1_chr21.bam
## # scale C1_gDNA_rep1_chr21.bam
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXNNNNNNCNXXXXXXXXXXXXXXXXXXXXXX_scaled.bed
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXNNNNNNCNXXXXXXXXXXXXXXXXXXXXXX.bigWig
## Cleaning up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXNNNNNNCNXXXXXXXXXXXXXXXXXXXXXX
## # tallymer produced/found hg38_chr21.tal_76.gtTxt.gz
## # kmer-size: 46
## # plus-offset: 23
## # minus-offset: 23
## # chrom: "chr21"
## # - 46709984 bases
## # seqtable produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXNNNNNCNNXXXXXXXXXXXXXXXXXXXXX.tbl
## # tabulate C1_gDNA_rep1_chr21.bam
## # scale C1_gDNA_rep1_chr21.bam
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXNNNNNCNNXXXXXXXXXXXXXXXXXXXXX_scaled.bed
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXNNNNNCNNXXXXXXXXXXXXXXXXXXXXX.bigWig
## Cleaning up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXNNNNNCNNXXXXXXXXXXXXXXXXXXXXX
## # tallymer produced/found hg38_chr21.tal_76.gtTxt.gz
## # kmer-size: 46
## # plus-offset: 23
## # minus-offset: 23
## # chrom: "chr21"
## # - 46709984 bases
## # seqtable produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCNNNNNNNXXXXXXXXXXXXXXXX.tbl
## # tabulate C1_gDNA_rep1_chr21.bam
## # scale C1_gDNA_rep1_chr21.bam
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCNNNNNNNXXXXXXXXXXXXXXXX_scaled.bed
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCNNNNNNNXXXXXXXXXXXXXXXX.bigWig
## Cleaning up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCNNNNNNNXXXXXXXXXXXXXXXX
## # tallymer produced/found hg38_chr21.tal_76.gtTxt.gz
## # kmer-size: 46
## # plus-offset: 23
## # minus-offset: 23
## # chrom: "chr21"
## # - 46709984 bases
## # seqtable produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXNNCNNNNNXXXXXXXXXXXXXXXXXX.tbl
## # tabulate C1_gDNA_rep1_chr21.bam
## # scale C1_gDNA_rep1_chr21.bam
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXNNCNNNNNXXXXXXXXXXXXXXXXXX_scaled.bed
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXNNCNNNNNXXXXXXXXXXXXXXXXXX.bigWig
## Cleaning up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXNNCNNNNNXXXXXXXXXXXXXXXXXX
## # tallymer produced/found hg38_chr21.tal_76.gtTxt.gz
## # kmer-size: 46
## # plus-offset: 23
## # minus-offset: 23
## # chrom: "chr21"
## # - 46709984 bases
## # seqtable produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXNNNCNNNNXXXXXXXXXXXXXXXXXXX.tbl
## # tabulate C1_gDNA_rep1_chr21.bam
## # scale C1_gDNA_rep1_chr21.bam
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXNNNCNNNNXXXXXXXXXXXXXXXXXXX_scaled.bed
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXNNNCNNNNXXXXXXXXXXXXXXXXXXX.bigWig
## Cleaning up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXNNNCNNNNXXXXXXXXXXXXXXXXXXX
## # tallymer produced/found hg38_chr21.tal_76.gtTxt.gz
## # kmer-size: 46
## # plus-offset: 23
## # minus-offset: 23
## # chrom: "chr21"
## # - 46709984 bases
## # seqtable produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXNNNNNNNXXXXXXXXX.tbl
## # tabulate C1_gDNA_rep1_chr21.bam
## # scale C1_gDNA_rep1_chr21.bam
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXNNNNNNNXXXXXXXXX_scaled.bed
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXNNNNNNNXXXXXXXXX.bigWig
## Cleaning up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXNNNNNNNXXXXXXXXX
## # tallymer produced/found hg38_chr21.tal_76.gtTxt.gz
## # kmer-size: 46
## # plus-offset: 23
## # minus-offset: 23
## # chrom: "chr21"
## # - 46709984 bases
## # seqtable produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXNNNNNNNXXXXXXXXXXXXXX.tbl
## # tabulate C1_gDNA_rep1_chr21.bam
## # scale C1_gDNA_rep1_chr21.bam
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXNNNNNNNXXXXXXXXXXXXXX_scaled.bed
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXNNNNNNNXXXXXXXXXXXXXX.bigWig
## Cleaning up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXNNNNNNNXXXXXXXXXXXXXX
## # tallymer produced/found hg38_chr21.tal_76.gtTxt.gz
## # kmer-size: 46
## # plus-offset: 23
## # minus-offset: 23
## # chrom: "chr21"
## # - 46709984 bases
## # seqtable produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXNNNNNNNXXXXXXXX.tbl
## # tabulate C1_gDNA_rep1_chr21.bam
## # scale C1_gDNA_rep1_chr21.bam
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXNNNNNNNXXXXXXXX_scaled.bed
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXNNNNNNNXXXXXXXX.bigWig
## Cleaning up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXNNNNNNNXXXXXXXX
## # tallymer produced/found hg38_chr21.tal_76.gtTxt.gz
## # kmer-size: 46
## # plus-offset: 23
## # minus-offset: 23
## # chrom: "chr21"
## # - 46709984 bases
## # seqtable produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXNNNNNNNXXXXXXX.tbl
## # tabulate C1_gDNA_rep1_chr21.bam
## # scale C1_gDNA_rep1_chr21.bam
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXNNNNNNNXXXXXXX_scaled.bed
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXNNNNNNNXXXXXXX.bigWig
## Cleaning up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXNNNNNNNXXXXXXX
## # tallymer produced/found hg38_chr21.tal_76.gtTxt.gz
## # kmer-size: 46
## # plus-offset: 23
## # minus-offset: 23
## # chrom: "chr21"
## # - 46709984 bases
## # seqtable produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXXNNNNNNNXXXXXX.tbl
## # tabulate C1_gDNA_rep1_chr21.bam
## # scale C1_gDNA_rep1_chr21.bam
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXXNNNNNNNXXXXXX_scaled.bed
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXXNNNNNNNXXXXXX.bigWig
## Cleaning up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXXNNNNNNNXXXXXX
## # tallymer produced/found hg38_chr21.tal_76.gtTxt.gz
## # kmer-size: 46
## # plus-offset: 23
## # minus-offset: 23
## # chrom: "chr21"
## # - 46709984 bases
## # seqtable produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXXXXNNNNNNNXXXX.tbl
## # tabulate C1_gDNA_rep1_chr21.bam
## # scale C1_gDNA_rep1_chr21.bam
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXXXXNNNNNNNXXXX_scaled.bed
## # scale produced C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXXXXNNNNNNNXXXX.bigWig
## Cleaning up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXXXXNNNNNNNXXXX
```

3.4 Convert bigWigs into bedGraph format

Convert bigWig files into bedGraph format using bigWigToBedGraph. This ensures proper formatting of our input for later use with unionbedgraph.

```
#Read in masks
while IFS= read -r line; do masks+=( "$line" ); done < masks.txt
#Convert bigwigs to bedGraph format
parallel -j3 'bigWigToBedGraph C1_gDNA_rep1_chr21_{}.bigWig C1_gDNA_rep1_chr21_{}.bedGraph' ::: ${masks[@]}
```

3.5 Combine all bedGraph files into a single file

Combine all bedGraph files into a single file using unionbedgraph and delete the individual bedGraph files.

```
#Read in masks
while IFS= read -r line; do masks+=( "$line" ); done < masks.txt
#Make array of bedGraph output
beds=( "${masks[@]/%/.bedGraph}" )
beds=( "${beds[@]/#/C1_gDNA_rep1_chr21_}" )
printf '%s\n' "${beds[@]}"
```

```
#Combine all bedGraph files into a single file
bedtools unionbedg -i ${beds[@]} > C1_gDNA_rep1_chr21_union.bedGraph
#Clean up all individual bedGraph files
for pos in ${beds[@]}; do
echo "Clean up" ${pos}
rm ${pos}
done
```

```
## C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXNNNNNNCNXXXXXXXXXXXXXXXXXXXXXX.bedGraph
## C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXNNNNNCNNXXXXXXXXXXXXXXXXXXXXX.bedGraph
## C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXNNNNCNNNXXXXXXXXXXXXXXXXXXXX.bedGraph
## C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXNNNCNNNNXXXXXXXXXXXXXXXXXXX.bedGraph
## C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXNNCNNNNNXXXXXXXXXXXXXXXXXX.bedGraph
## C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCNNNNNNNXXXXXXXXXXXXXXXX.bedGraph
## C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXNNNNNNNXXXXXXXXXXXXXX.bedGraph
## C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXNNNNNNNXXXXXXXXX.bedGraph
## C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXNNNNNNNXXXXXXXX.bedGraph
## C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXNNNNNNNXXXXXXX.bedGraph
## C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXXNNNNNNNXXXXXX.bedGraph
## C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXXXXNNNNNNNXXXX.bedGraph
## Clean up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXNNNNNNCNXXXXXXXXXXXXXXXXXXXXXX.bedGraph
## Clean up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXNNNNNCNNXXXXXXXXXXXXXXXXXXXXX.bedGraph
## Clean up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXNNNNCNNNXXXXXXXXXXXXXXXXXXXX.bedGraph
## Clean up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXNNNCNNNNXXXXXXXXXXXXXXXXXXX.bedGraph
## Clean up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXNNCNNNNNXXXXXXXXXXXXXXXXXX.bedGraph
## Clean up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCNNNNNNNXXXXXXXXXXXXXXXX.bedGraph
## Clean up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXNNNNNNNXXXXXXXXXXXXXX.bedGraph
## Clean up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXNNNNNNNXXXXXXXXX.bedGraph
## Clean up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXNNNNNNNXXXXXXXX.bedGraph
## Clean up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXNNNNNNNXXXXXXX.bedGraph
## Clean up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXXNNNNNNNXXXXXX.bedGraph
## Clean up C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXXXXCXXXXXXXXXXXXNNNNNNNXXXX.bedGraph
```

4 Rule ensemble implementation

In this section, take the seqOutBias scaled output (in bedGraph format) and apply the pre-trained rule ensemble model. Then scale this output to the original read depth. Read depth scaled rule ensemble output is then written into a bedGraph file. Finally, convert this bedGraph file into bigWig format for further use and analysis.

4.1 Rule ensemble implementation

Implement the rule ensemble modeling using the pre-trained model and single bedgraph file. Then scale this output to the unscaled read depth. Lastly, write this to a bedGraph output.

```
library(data.table)
options(scipen = 100)
#Read the unscaled bed file for read depth
print('Reading unscaled bed file...')
```

```
## [1] "Reading unscaled bed file..."
```

```
unscaled_bed = fread("C1_gDNA_rep1_chr21_scaling_test_not_scaled.bed")
unscaled_bed = sum(unscaled_bed$V5)
#Read the unionbedGraph file
print('Reading bed file C1_gDNA_rep1_chr21_union.bedGraph')
```

```
## [1] "Reading bed file C1_gDNA_rep1_chr21_union.bedGraph"
```

```
x <- fread('C1_gDNA_rep1_chr21_union.bedGraph')
#Set column names based on masks.txt file
masknames = read.table('masks.txt', header = FALSE)
masknames = masknames[,1]
masknames = c('chr', 'start', 'stop', masknames)
colnames(x) = masknames
#Retrieve 12 mask rule ensemble model:
source('https://raw.githubusercontent.com/guertinlab/Tn5bias/master/seqOutATACBias_workflow_Vignette/12mask_RuleE


#Implement rule ensemble model on union bedgraph
print('Applying rule ensemble model')
```

```
## [1] "Applying rule ensemble model"
```

```
x = RE_scale_12mask(x)

#Scale output to read depth
pre_read_depth = sum(x$RuleEnsemble)
RDS = unscaled_bed/pre_read_depth
x[, RuleEnsemble_RDS := RuleEnsemble*RDS ]

print('Writing rule ensemble scaled bed file...')
```

```
## [1] "Writing rule ensemble scaled bed file..."
```

```
write.table(x[,c(1:3,17)], file = 'C1_gDNA_rep1_chr21_RE_scaled.bedGraph',
            col.names = FALSE, row.names=FALSE, sep = '\t', quote=FALSE)
```

4.2 Convert the bedGraph rule ensemble output to bigWig format

To convert the bedGraph rule ensemble scaled output to bigWig format, we must first create a chrom.sizes file from our reference genome.

```
faidx hg38_chr21.fa -i chromsizes > hg38_chr21.fa.chrom.sizes
```

Next, we convert our bedGraph output into bigWig format

```
bedGraphToBigWig C1_gDNA_rep1_chr21_RE_scaled.bedGraph \
hg38_chr21.fa.chrom.sizes C1_gDNA_rep1_chr21_RE_scaled.bigWig
```

5 Bias correction analysis

This section verifies that the rule ensemble model has corrected the Tn5 sequence bias by plotting the composite signal at the ESR1 motif against the seqOutBias output and unscaled output.

5.1 Downloading ESR1 FIMO motifs

Download chromosome 21 plus and minus ESR1 FIMO motifs from cyverse.

```
wget https://data.cyverse.org/dav-anon/iplant/home/jacobwolpe/ESR1_rm_chr21_fimo.txt
```

```
## --2022-12-13 13:26:32--  https://data.cyverse.org/dav-anon/iplant/home/jacobwolpe/ESR1_rm_chr21_fimo.txt
## Resolving data.cyverse.org (data.cyverse.org)... 206.207.252.35
## Connecting to data.cyverse.org (data.cyverse.org)|206.207.252.35|:443... connected.
## HTTP request sent, awaiting response... 200 OK
## Length: 924648 (903K) [application/octet-stream]
## Saving to: ‘ESR1_rm_chr21_fimo.txt’
##
##      0K .......... .......... .......... .......... ..........  5%  326K 3s
##     50K .......... .......... .......... .......... .......... 11%  706K 2s
##    100K .......... .......... .......... .......... .......... 16% 6.17M 1s
##    150K .......... .......... .......... .......... .......... 22% 38.8M 1s
##    200K .......... .......... .......... .......... .......... 27%  701K 1s
##    250K .......... .......... .......... .......... .......... 33% 44.1M 1s
##    300K .......... .......... .......... .......... .......... 38% 53.8M 0s
##    350K .......... .......... .......... .......... .......... 44% 30.7M 0s
##    400K .......... .......... .......... .......... .......... 49%  796K 0s
##    450K .......... .......... .......... .......... .......... 55% 6.65M 0s
##    500K .......... .......... .......... .......... .......... 60% 41.2M 0s
##    550K .......... .......... .......... .......... .......... 66% 51.3M 0s
##    600K .......... .......... .......... .......... .......... 71% 47.3M 0s
##    650K .......... .......... .......... .......... .......... 77% 53.2M 0s
##    700K .......... .......... .......... .......... .......... 83% 51.7M 0s
##    750K .......... .......... .......... .......... .......... 88% 66.3M 0s
##    800K .......... .......... .......... .......... .......... 94% 16.2M 0s
##    850K .......... .......... .......... .......... .......... 99%  815K 0s
##    900K ..                                                     100% 59.5K=0.4s
##
## 2022-12-13 13:26:34 (1.97 MB/s) - ‘ESR1_rm_chr21_fimo.txt’ saved [924648/924648]
```

5.2 Plotting the ESR1 composite profile

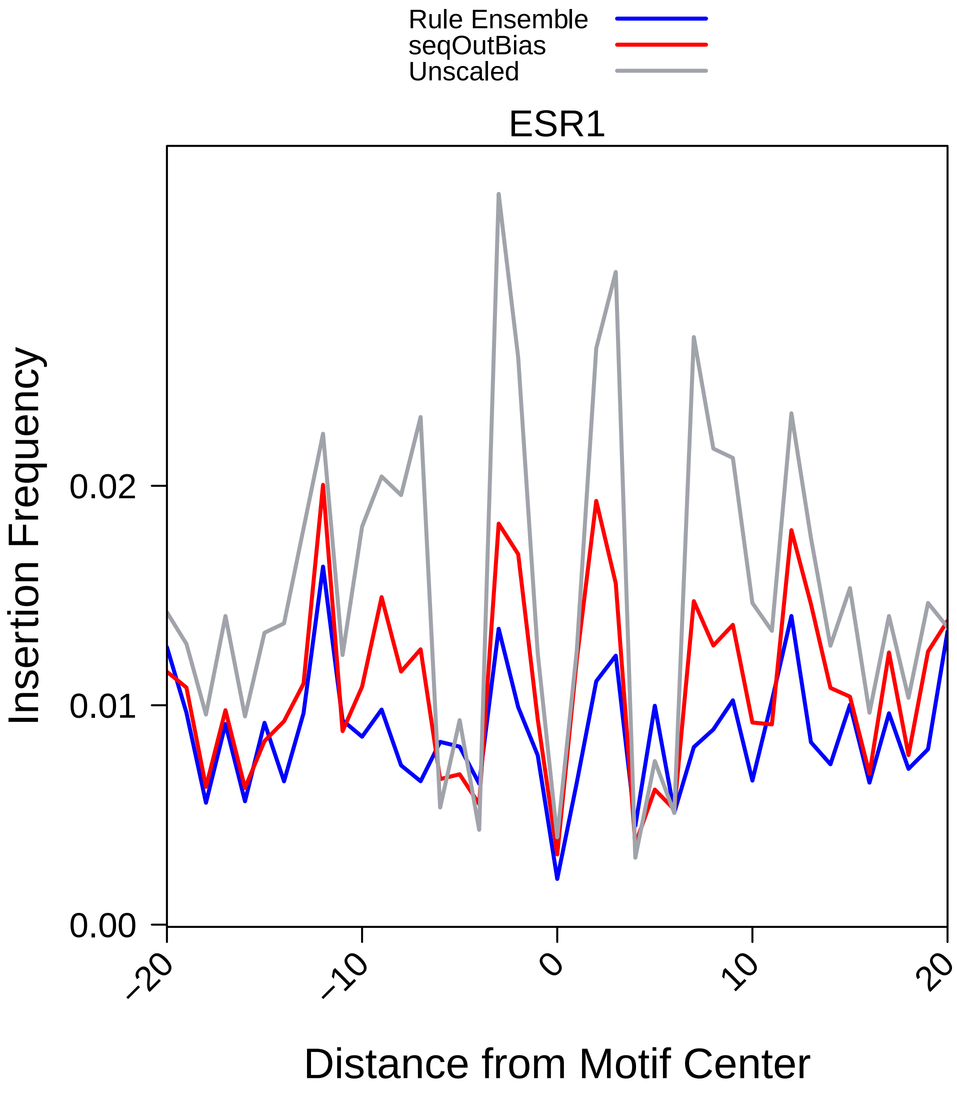
Plot the rule ensemble scaled output, seqOutBias scaled output and unscaled output at the ESR1 motif. First, coordinates are converted into bed format from input FIMO format. Next, the signal at these genomic locations is retrieved and averaged. Finally, we overlay these plotted values for comparison.

```r
source('https://raw.githubusercontent.com/guertinlab/Tn5bias/master/Manuscript_Vignette/Vignette_Scripts/Tn5_Bias
###################################################################################

#Load in the ESR1 motif region set
Motiflist <- vector('list', 1)
Motiflist[[1]] <- FIMO.to.BED('ESR1_rm_chr21_fimo.txt')

#Determine signal at the ESR1 motif in unscaled data
unscaled_compositelist = vector('list', 1)
unscaled_compositelist[[1]] = BED.query.bigWig(Motiflist[[1]],
                              'C1_gDNA_rep1_chr21_unscaled.bigWig',
                              'C1_gDNA_rep1_chr21_unscaled.bigWig',
                                        upstream = 20, downstream = 20,
                                        factor = 'ESR1',
                                        group = 'Unscaled', ATAC = TRUE)
```

```
## Loading required package: matrixStats
```

```r
#Determine signal at the ESR1 motif in seqOutBias data
seqOutBias_compositelist = vector('list', 1)
seqOutBias_compositelist[[1]] = BED.query.bigWig(Motiflist[[1]],
                              'C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXNNNCNNNNXXXXXXXXXXXXXXXXXXX.bigWig',
                              'C1_gDNA_rep1_chr21_XXXXXXXXXXXXXXXXXXXXNNNCNNNNXXXXXXXXXXXXXXXXXXX.bigWig',
                                        upstream = 20, downstream = 20,
                                        factor = 'ESR1',
                                        group = 'seqOutBias', ATAC = TRUE)
#Determine signal at the ESR1 motif in Rule Ensemble data
RE_compositelist = vector('list', 1)
RE_compositelist[[1]] = BED.query.bigWig(Motiflist[[1]],
                              'C1_gDNA_rep1_chr21_RE_scaled.bigWig',
                              'C1_gDNA_rep1_chr21_RE_scaled.bigWig',
                                        upstream = 20, downstream = 20,
                                        factor = 'ESR1',
                                        group = 'Rule Ensemble', ATAC = TRUE)


composite_plot = rbind(do.call(rbind, RE_compositelist),
                       do.call(rbind, unscaled_compositelist),
                       do.call(rbind, seqOutBias_compositelist))

composite_plot$group = factor(composite_plot$group)

plot.composites(composite_plot, legend = TRUE,
                pdf_name = 'Rule_ensemble_scaling_ESR1_chr21_composite',
                ylabel = 'Insertion Frequency',
                xlabel = 'Distance from Motif Center',
                motifline = FALSE, Motiflen = 0, figwidth = 6, figheight = 6, x_axis_range = -20:20)
```

```
## Loading required package: lattice

## pdf
##   2
```


Rule Ensemble
seqOutBias
Unscaled
ESR1
Insertion Frequency
Distance from Motif Center