Comparison of seqOutATACBias and ATACorrect bias correction output

Jacob B. Wolpe[*] Michael J. Guertin[†]

Abstract

This vignette outlines an analysis which compares output from ATACorrect and seqOutATACBias/rule ensemble scaling. We find that ATACorrect scales a subset of individual reads, in addition to removing all reads outside of input peak regions, optimizing data for later footprinting and downstream analysis of a defined region set. ATACorrect also creates simulated read data within input peak regions to further offset Tn5 bias within these defined genomic locations. ATACorrect individual read scaling is consistent between input peak regions, and slightly changes when these regions change. Because ATACorrect requires these input peak regions, it is a tool highly optimized for footprinting analysis, but does not broadly correct observed Tn5 sequence bias in a data set. In contrast, seqOutATACBias operates by scaling all individual reads of a data set to correct Tn5 local and regional bias. In order to compare output between the two we ran ATACorrect on all input reads, divided into 200kb segments, removed all synthetic reads created by ATACorrect, and concatenated all runs together to generate a set of ATACorrect scaled reads. We then set the read depth of ATACorrect scaled reads equal to that of rule ensemble output read depth. Comparison of these individually scaled reads reveals the incompatibility of comparing the output from these two software packages.

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

This vignette first examines the differences between seqOutATACBias and ATACorrect (a TOBIAS tool) output from a test set of read data on chromosome 21. Once this examination is complete, a series of modifications to ATACorrect output is performed in order to facilitate comparison with rule ensemble scaled output. These analyses require that seqOutATACBias and its dependencies are in path. Because seqOutATACBias and TOBIAS are written in different python versions, a conda virtual environment (python 3.7, named 'TOBIAS_venv') is used to install and run TOBIAS, which must also be in path using this method. If you wish to reproduce this analysis, you **must** have a conda virtual environment named 'TOBIAS_venv' with TOBIAS installed. This vignette is split into 5 sections: a check to make sure software dependencies are installed, running seqOutATACBias and ATACorrect on identical input reads and reference genome, analysis of their output, preparation of ATACorrect output for comparison with rule ensemble scaled reads, and finally analysis of bias correction between the two. Results from this analysis reveal that while both tools operate on similar input, their output is drastically different, as is their intended use.

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
conda 22.11.1
TOBIAS 0.15.1
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"
  elif ! command -v seqOutATACBias &> /dev/null
  then
      echo "seqOutATACBias could not be found"
  elif ! command -v conda &> /dev/null
  then
      echo "conda could not be found"
  else
      source activate TOBIAS_venv
  fi
  if ! command -v TOBIAS &> /dev/null
  then
      echo "TOBIAS 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 seqOutATACBias: https://github.com/guertinlab/Tn5bias/tree/master/seqOutATACBias_setup
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/
conda: https://docs.conda.io/projects/conda/en/latest/user-guide/install/macos.html
TOBIAS: https://github.com/loosolab/TOBIAS

2.1 Auto-install R packages

Install the `data.table`, `bigWig`, `devtools`, and `ggplot2` 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")
}
ggplottest = require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
if(ggplottest==FALSE){
  install.packages('ggplot2')
}
```

3 Generating output from seqOutATACBias and ATACorrect

This section prepares the data to compare the output structure from seqOutATACBias and ATACorrect. The first section downloads the chromosome 21 reference genome (hg38), aligned unscaled chromosome 21 read files in BAM format, and FIMO results for ESR1 on chromosome 21 from cyverse. Next, we use each model to generate the output that will be analyzed and compared.

3.1 Downloading reference genome and read data.

Download the reference genome for chromosome 21 (hg38_chr21.fa), aligned deproteinized ATAC-seq read file from cyverse (C1_gDNA_rep1_chr21.bam), and ESR1 motifs for chromosome 21 (ESR1_rm_chr21_fimo.txt).

```
#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
wget -nv https://data.cyverse.org/dav-anon/iplant/home/jacobwolpe/ESR1_rm_chr21_fimo.txt
wget -nv https://data.cyverse.org/dav-anon/iplant/home/jacobwolpe/ESR1_REST_chr21_fimo.txt
```

```
## 2023-03-13 11:37:46 URL:https://data.cyverse.org/dav-anon/iplant/home/jacobwolpe/C1_gDNA_rep1_chr21.bam [2
## 2023-03-13 11:37:48 URL:https://data.cyverse.org/dav-anon/iplant/home/jacobwolpe/hg38_chr21.fa [47488490/4
## 2023-03-13 11:37:49 URL:https://data.cyverse.org/dav-anon/iplant/home/jacobwolpe/ESR1_rm_chr21_fimo.txt [9
## 2023-03-13 11:37:51 URL:https://data.cyverse.org/dav-anon/iplant/home/jacobwolpe/ESR1_REST_chr21_fimo.txt
```

3.2 Run seqOutATACBias to generate output

Here we run seqOutATACBias to scale ATAC-seq reads using our rule ensemble model.

```
seqOutATACBias masks -i=C1_gDNA_rep1_chr21.bam -g=hg38_chr21.fa -p=3 -r=72

seqOutATACBias scale -i=C1_gDNA_rep1_chr21_union.bedGraph -g=hg38_chr21.fa
```

3.3 Run ATACorrect to generate output

We now use ATACorrect to generate output for later comparison. Because ATACorrect requires the genomic coordinates of interest as an input field ('peaks' file), we first convert the downloaded coordinates from FIMO format to bed.

```
options(scipen = 100)
source('https://raw.githubusercontent.com/guertinlab/Tn5bias/master/Manuscript_Vignette/Vignette_Scripts/Tn5_
library(data.table)
```

```
bed_peaks = FIMO.to.BED('ESR1_rm_chr21_fimo.txt')
write.table(bed_peaks[,1:4],file= 'ESR1_chr21.bed',
              sep = '\t', quote = FALSE, row.names = FALSE, col.names = FALSE)

bed_peaks = FIMO.to.BED('ESR1_REST_chr21_fimo.txt')
write.table(bed_peaks[,1:4],file= 'ESR1_REST_chr21.bed',
              sep = '\t', quote = FALSE, row.names = FALSE, col.names = FALSE)
```

Next, we run ATACorrect using the newly generated bed files and previously downloaded data. The first run of ATACorrect uses all identified ESR1 motifs on chromosome 21. The second run includes REST motifs in the peak file, in addition to the ESR1 motifs. Once ATACorrect has finished running, we convert the bigwig output into bedgraph format for comparison.

```
source activate TOBIAS_venv
TOBIAS ATACorrect --bam C1_gDNA_rep1_chr21.bam --genome hg38_chr21.fa --peaks ESR1_chr21.bed --outdir ATACorr
bigWigToBedGraph ATACorrect_ESR1_output/C1_gDNA_rep1_chr21_corrected.bw ATACorrect_ESR1_output/C1_gDNA_rep1_c
bigWigToBedGraph ATACorrect_ESR1_output/C1_gDNA_rep1_chr21_uncorrected.bw ATACorrect_ESR1_output/C1_gDNA_rep1

TOBIAS ATACorrect --bam C1_gDNA_rep1_chr21.bam --genome hg38_chr21.fa --peaks ESR1_REST_chr21.bed --outdir AT
bigWigToBedGraph ATACorrect_ESR1_REST_output/C1_gDNA_rep1_chr21_corrected.bw ATACorrect_ESR1_REST_output/C1_g
bigWigToBedGraph ATACorrect_ESR1_REST_output/C1_gDNA_rep1_chr21_uncorrected.bw ATACorrect_ESR1_REST_output/C1
```

4 Output analysis

This section compares the output structure and potential uses of both seqOutATACBias and ATACorrect.

4.1 ATACorrect scaling and read depth is only applied to supplied peak regions

To illustrate the differences between seqOutATACBias scaling and ATACorrect footprinting correction, we first download an illustrative browser snapshot from the following UCSC browser session:
https://genome.ucsc.edu/s/JacobWolpe/sOAB%20ATACorrect%20Raw%20Comparison

```
wget -nv https://github.com/guertinlab/Tn5bias/raw/master/Manuscript_Vignette/figs/sOAB_ATACorrect_comparison
```

```
## 2023-03-13 11:51:52 URL:https://raw.githubusercontent.com/guertinlab/Tn5bias/master/Manuscript_Vignette/fi
```

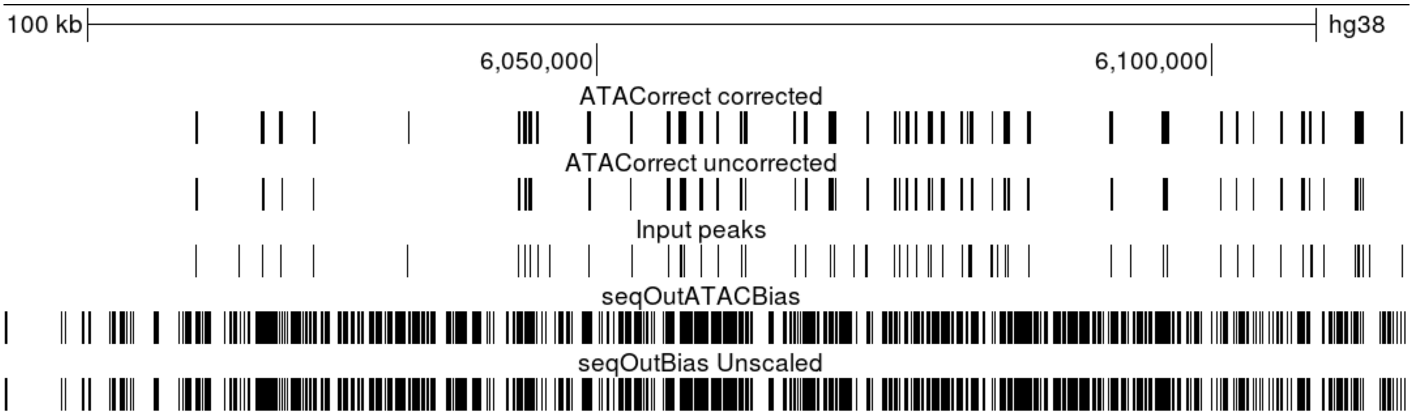

Figure 1: Browser snapshot of unscaled seqOutBias output, seqOutATACBias output, uncorrected ATACorrect output, corrected ATACorrect output, and the ‘peaks‘ file (Input peaks) used as input for the two ATACorrect output files

The browser snapshot seen in Figure 1 exemplifies the incompatibility between seqOutATACBias and ATACorrect outputs. In this image, we see that ATACorrect corrected and uncorrected output is only present at genomic locations contained in the input peaks file (ESR1_chr21). This is in contrast to seqOutATACBias scaled output, which is present at all read locations in the original input BAM file. seqOutATACBias was designed as a general method used to scale individual reads of a given data set to reduce the Tn5 sequence bias of each read. ATACorrect refines input data to the specific regions of interest, removes all other reads and adds theoretically unbiased reads not in the original data set to optimize footprinting algorithms. This is why the evaluation of both methods is so drastically different: ATACorrect was measured by improving correlation with ChIP-seq data and improvement of footprint depth; seqOutATACBias was measured for its ability to return average signal to theoretically random cleavage. Next, we show that simulated ATACorrect reads are confined to the regions nearby the input peaks file.

4.2 ATACorrect simulated data is within 175 base pairs of the peaks file regions

To determine what percent of simulated data produced by ATACorrect is within peak regions, we first expand the peak regions by 175 base pairs in both directions.

```
peaks = fread('ESR1_chr21.bed')
peaks$V2 = peaks$V2 - 175
peaks$V3 = peaks$V3 + 175
write.table(peaks[,1:4],file= 'ESR1_chr21_175bp.bed',
              sep = '\t', quote = FALSE, row.names = FALSE, col.names = FALSE)
```

Next, we use bedtools intersect to remove all uncorrected reads from the corrected reads file.

```
bedtools intersect -v -a ATACorrect_ESR1_output/C1_gDNA_rep1_chr21_corrected.bedgraph -b ATACorrect_ESR1_outp
```

Now, we determine which of these simulated reads are within the bounds of the peaks file we generated.

```
bedtools intersect -v -a C1_gDNA_rep1_chr21_corrected_NO_UNCOR_READS.bedgraph -b ESR1_chr21_175bp.bed > ATACo
ls -l ATACorrect_175bp_simulated_peaks.bed
```

```
## -rw-r--r--  1 guertinlab  staff  0 Mar 13 11:51 ATACorrect_175bp_simulated_peaks.bed
```

As this file is empty, no ATACorrect simulated reads are farther than 175 base pairs from a region in the original peaks file. This result shows that ATACorrect simulated output is entirely based on the input peak regions, to the exclusion of all other data within the input BAM file.

We can visualize this by zooming in on the previous browser snap shot. First, we download the image.

```
wget -nv https://github.com/guertinlab/Tn5bias/raw/master/Manuscript_Vignette/figs/sOAB_ATACorrect_SimReads.p
```

```
## 2023-03-13 11:51:53 URL:https://raw.githubusercontent.com/guertinlab/Tn5bias/master/Manuscript_Vignette/fi
```

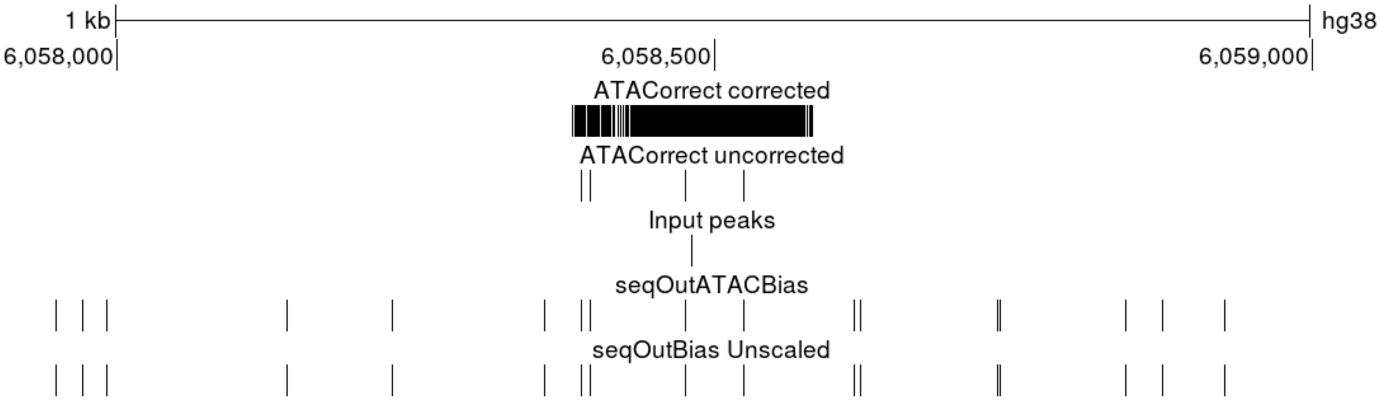

Figure 2: Zoomed in browser snapshot of unscaled seqOutBias output, seqOutATACBias output, uncorrected ATACorrect output, corrected ATACorrect output, and the 'peaks' file (Input peaks) used as input for the two ATACorrect output files

This shows that ATACorrect corrected output within 175bp of the coordinates within the peaks file is heavily dependent on simulated data, which is not found in either the ATACorrect unscaled output, seqOutATACBias, or seqOutBias unscaled output.

4.3 ATACorrect read scaling is consistent, regardless of which peaks are input

To show that ATACorrect read scaling is consistent, regardless of the input peaks file, we first remove all simulated data from the ATACorrect output. Next, we only examine reads present in the ESR1 peak file, as regions in the REST peak file will not be present in both.

```
bedtools intersect -a ATACorrect_ESR1_REST_output/C1_gDNA_rep1_chr21_corrected.bedgraph -b ATACorrect_ESR1_RE

bedtools intersect -a ATACorrect_ESR1_output/C1_gDNA_rep1_chr21_corrected.bedgraph -b ATACorrect_ESR1_output/

bedtools intersect -a ESR1_REST_C1_gDNA_rep1_chr21_corrected_NO_SIM_READS.bedgraph -b ESR1_C1_gDNA_rep1_chr21

bedtools intersect -a ESR1_C1_gDNA_rep1_chr21_corrected_NO_SIM_READS.bedgraph -b ESR1_REST_C1_gDNA_rep1_chr21
```

Finally, we plot these scaled reads.

```
ESR1_reads = fread('ESR1_C1_gDNA_rep1_chr21_corrected_NO_SIM_REST_READS.bedgraph')
ESR1_REST_reads = fread('ESR1_REST_C1_gDNA_rep1_chr21_corrected_NO_SIM_REST_READS.bedgraph')

pdf(file = 'sOAB_ATACorrect_scaled_read_comparison.pdf')
plot(ESR1_reads$V4, ESR1_REST_reads$V4, xlab = 'ESR1 Scaled Reads', ylab = 'ESR1 REST Scaled Reads', pch = 16
dev.off()
```

```
## pdf
##   2
```

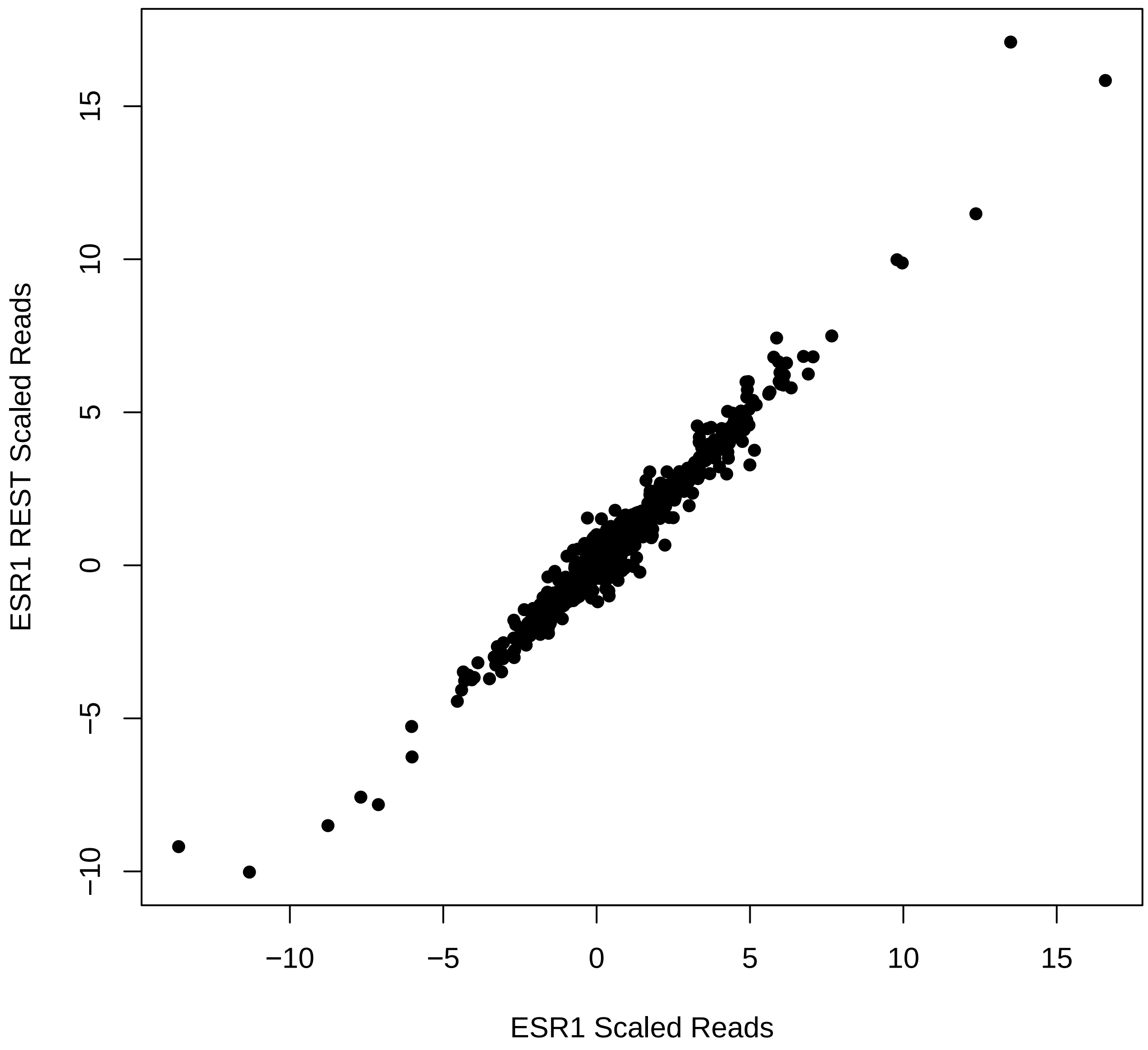

Figure 3: Dot plot of ATACorrect scaled read signal in the ESR1 peak regions from both ESR1 input and combined ESR1 and REST regions.

We observe that ATACorrect read scaling is highly correlated between runs using different input peaks files. This leads us to conclude that ATACorrect scaled reads are consistently scaled in the same manner, regardless of which regions are used as input peaks. In order to compare output between the two methods, simulated reads must be removed from ATACorrect output, as they make read scaling comparison between the two methods impossible. We leverage this understanding to create a set of ATACorrect scaled reads for comparison with rule ensemble scaled reads in the next section.

5 ATACorrect scaled read preparation

Because preparation of the ATACorrect scaled reads required nearly 100 runs of the software and generated several hundred GB of data, these computations were conducted in an HPC environment and not in the present vignette. However, we include the scripts and code used to produce this data below, so that a reader may understand the processing steps or recapitulate these findings.

We first attempted to run ATACorrect with a peak file in which the coordinates were equal to chromosome size. This resulted in no bias correction. Next, we attempted to input all test motif coordinates in our data set as a peak file. This resulted in a computation which lasted more than four days, using several processors and 60GB of RAM, with no output. Finally, we split a bed file containing all read positions in our data set into separate files containing 200k read positions each and ran them as separate computations.

We first split our bed file containing all read positions into files containing 200k read coordinates each.

```
split -l 200000 C1_gDNA_rep1.bedGraph C1_gDNA_rep1_
```

Next, we generated a script for each split file to run them independently. Below is the `aa` split file, used as an example.

```
TOBIAS ATACorrect --bam C1_gDNA_rep1.bam --genome hg38.fa \
--peaks C1_gDNA_rep1_aa --outdir split_output_aa --cores 1
```

Once all runs were finished, ATACorrect output corrected bigWig files were extracted from output folders and given names corresponding to their specific splits.

```
for folder in *
do
    name=$(echo $folder | awk -F"split_output_" '{print $2}')
    echo $name
    cd $folder
    cp C1_gDNA_rep1_corrected.bw ../../corrected_bw/${name}_corrected.bw
    cd ../
done
```

BigWig files were then converted into bedGraph format and pseudo reads removed, leaving only the original 200k scaled reads from the data set for each split. Once this had been completed for all splits, these bedGraphs were concatenated together.

```
for file in *.bw
do
    echo $file
    name=$(echo $file | awk -F"_corrected.bw" '{print $1}')
    bigWigToBedGraph $file ${name}.bedGraph
    bedtools intersect -a ${name}.bedGraph -b C1_gDNA_rep1.bedGraph > ${name}_NOSIM.bedGraph
done

cat *_NOSIM.bedGraph > ATACorrect_SequentialPeaks_NOSIM.bedGraph
```

Starting with the concatenated bedGraph file, we first remove duplicated reads. These read duplications are a function of ATACorrect expanding the region of interest by 175bp in both directions, thus capturing some reads which are within 175bp of the split. We find that these duplications account for 0.0022% of reads. We then scale the read depth of the concatenated bedGraph to that of our unscaled output. This read depth scaled output is then written to a bedGraph, sorted and converted back to bigWig format for plotting.

```
ATACorrect_bedgraph = fread('ATACorrect_SequentialPeaks_NOSIM.bedGraph')
ATACorrect_bedgraph = ATACorrect_bedgraph[!c(duplicated(ATACorrect_bedgraph[,1:2])),]

unscaled_read_depth = fread('C1_gDNA_rep1.bedGraph')
unscaled_read_depth = sum(unscaled_read_depth$V4)

ATACorrect_RD = sum(ATACorrect_bedgraph$V4)
RDS_factor = unscaled_read_depth/ATACorrect_RD
```

```
ATACorrect_bedgraph$RDS = ATACorrect_bedgraph$V4*RDS_factor

write.table(ATACorrect_bedgraph[,c(1:3,5)],file= 'ATACorrect_SequentialPeaks_NOSIM_RDS.bedGraph',
            sep = '\t', quote = FALSE, row.names = FALSE, col.names = FALSE)

system('bedSort ATACorrect_SequentialPeaks_NOSIM_RDS.bedGraph ATACorrect_SequentialPeaks_NOSIM_RDS.bedGraph')

system('bedgraphtobigwig ATACorrect_SequentialPeaks_NOSIM_RDS.bedGraph hg38.chrom.sizes ATACorrect_Sequential
```

Using the ATACorrect scaled reads, we evaluate signal at the same test set composite motif positions as used in the manuscript for comparison and analysis.

```
BWs <- c('ATACorrect_SequentialPeaks_NOSIM_RDS.bw')
ATACorrect_compositelist <- vector('list', 18)
for (i in 1:18) {
  ATACorrect_compositelist[[i]] = BED.query.bigWig(Motiflist[[i]], paste(BWs), paste(BWs),
                                      upstream = 100, downstream = 100,
                                      factor = names(Motiflist[i]),
                                      group = 'ATACorrect', ATAC = TRUE)
}
save(ATACorrect_compositelist, file = 'ATACorrect_compositelist.Rdata')
```

6 ATACorrect scaled read analysis

To compare the bias correction of ATACorrect read scaling against rule ensemble scaling, we first download all data necessary. This data contains all of the values as shown in the submission manuscript, in addition to the added ATACorrect plot values.

```
wget https://github.com/guertinlab/Tn5bias/raw/master/Manuscript_Vignette/Output_Vignette/ATACorrect_sOAB_com
```

We now use this downloaded data to plot composites of rule ensemble scaling and ATACorrect output at all test motifs evaluated in the manuscript.

```
source('https://raw.githubusercontent.com/guertinlab/Tn5bias/master/Manuscript_Vignette/Vignette_Scripts/Tn5_
load('ATACorrect_sOAB_compositelist.Rdata')

plot.composites(ATACorrect_sOAB_compositelist[[1]], legend = TRUE,
                pdf_name = 'ATACorrect_RuleEnsemble_Tn5_composite_comparison',
                figwidth = 16, figheight = 12,
                ylabel = '',
                xlabel = '',
                motifline = TRUE, Motiflen = ATACorrect_sOAB_compositelist[[2]], layoutgrid = c(6,3),
                x_axis_range = -20:20, X_axis_ticks = seq(-20,20,10),
                Y_axis_ticks = seq(0,0.05,0.01), nYaxisdigits = 2,
                hline_val = 0.006140407, y_axis_range = seq(0,0.02,0.01),
                y_axis = FALSE, hline = TRUE, Y_ticks = FALSE, labsize = 0.85,
                col.lines = c("#A1A3AB","#FF0000", "#0000FF"))
```

```
## Loading required package: lattice
```

```
## pdf
##   2
```

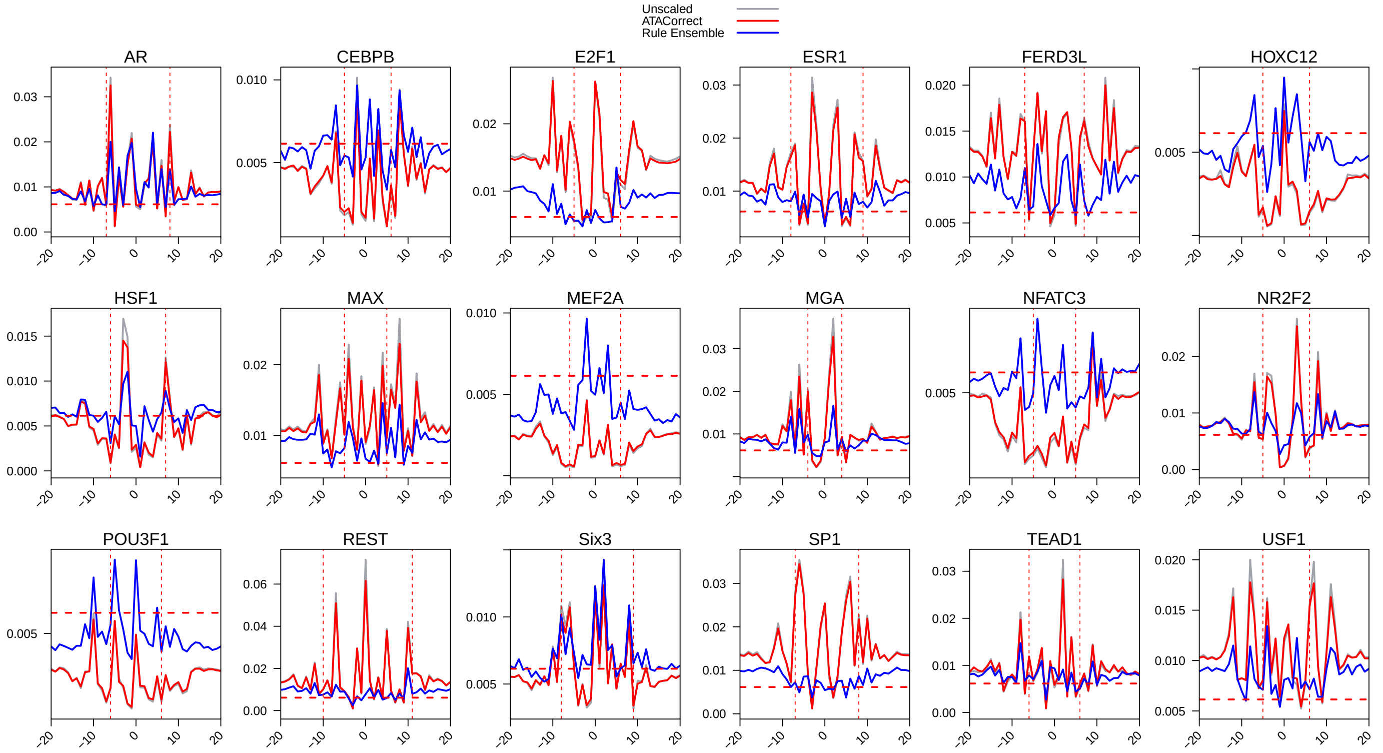

Figure 4: Test set composites of rule ensemble and ATACorrect scaling.

Bias correction observed in Figure 4 shows that rule ensemble scaling outperforms ATACorrect at scaling individual reads. To quantify the difference in bias correction, we next plot the absolute difference between rule ensemble and ATACorrect treatments from theoretical average cleavage for each position within 10 base pairs of a given motif. This can be thought of as the improvement (when above 0) of rule ensemble scaling over ATACorrect scaling.

```
singlenuc_frame = NULL
for (i in 1:nrow(ATACorrect_sOAB_compositelist[[2]])) {
  singlenuc_store =
    ATACorrect_sOAB_compositelist[[1]][which(ATACorrect_sOAB_compositelist[[1]]$factor ==
                    rownames(ATACorrect_sOAB_compositelist[[2]])[i] &
                    ATACorrect_sOAB_compositelist[[1]]$x <
                    ((ATACorrect_sOAB_compositelist[[2]][i,]/2)+10) &
                    ATACorrect_sOAB_compositelist[[1]]$x >
                    - ((ATACorrect_sOAB_compositelist[[2]][i,]/2)+10)),]
  singlenuc_frame = rbind(singlenuc_frame, singlenuc_store)
}

improvement_frame = singlenuc_frame[which(singlenuc_frame$group == 'Unscaled'),]
improvement_frame = improvement_frame[,-c(5)]
colnames(improvement_frame) = c('Unscaled_x', 'Unscaled_est', 'Unscaled_factor', 'Unscaled_group')

improvement_frame =
  cbind(improvement_frame, singlenuc_frame[which(singlenuc_frame$group == 'Rule Ensemble'),1:4])
colnames(improvement_frame)[5:8] = c('RE_x', 'RE_est', 'RE_factor', 'RE_group')
improvement_frame =
  cbind(improvement_frame, singlenuc_frame[which(singlenuc_frame$group == 'ATACorrect'),1:4])
colnames(improvement_frame)[9:12] =
  c('ATACorrect_x', 'ATACorrect_est', 'ATACorrect_factor', 'ATACorrect_group')

improvement_frame = improvement_frame[,-c(4,5,7,8,9,11,12)]
improvement_frame$calc_avg = 0.006140407
improvement_frame$RE_absdiff = abs(improvement_frame$RE_est - improvement_frame$calc_avg)
improvement_frame$ATACorrect_absdiff =
  abs(improvement_frame$ATACorrect_est - improvement_frame$calc_avg)
rownames(improvement_frame) = 1:nrow(improvement_frame)

improvement_frame$ATACorrect_sub_RE =
  improvement_frame$ATACorrect_absdiff - improvement_frame$RE_absdiff
improvement_frame$plot_col  = 'Rule Ensemble Improvement over ATACorrect'

length(which(improvement_frame$ATACorrect_absdiff >
               improvement_frame$RE_absdiff))/ nrow(improvement_frame)
```

```
## [1] 0.8681507
```

```
pdf('ATACorrect_RE_improvement.pdf', width=10, height=12)
ggplot(improvement_frame, aes(x=plot_col, y = ATACorrect_sub_RE)) +
  geom_violin(trim = FALSE, color = 'black', fill = 'light blue') +
  xlab("") + ylab("Rule Ensemble Improvement") + theme_classic() +
  geom_jitter(shape=16, position=position_jitter(0.2), alpha = 0.4) +
  theme(axis.text.x=element_blank(), axis.ticks.x=element_blank(),
        axis.text = element_text(size = 36), axis.title = element_text(size = 42,
        family = 'sans', face = 'bold'),
        axis.text.y = element_text(colour = "black", family = 'sans')) +
        geom_abline(slope = 0, lwd = 2, color = 'red', linetype = "dashed")
dev.off()
```

```
## pdf
##   2
```

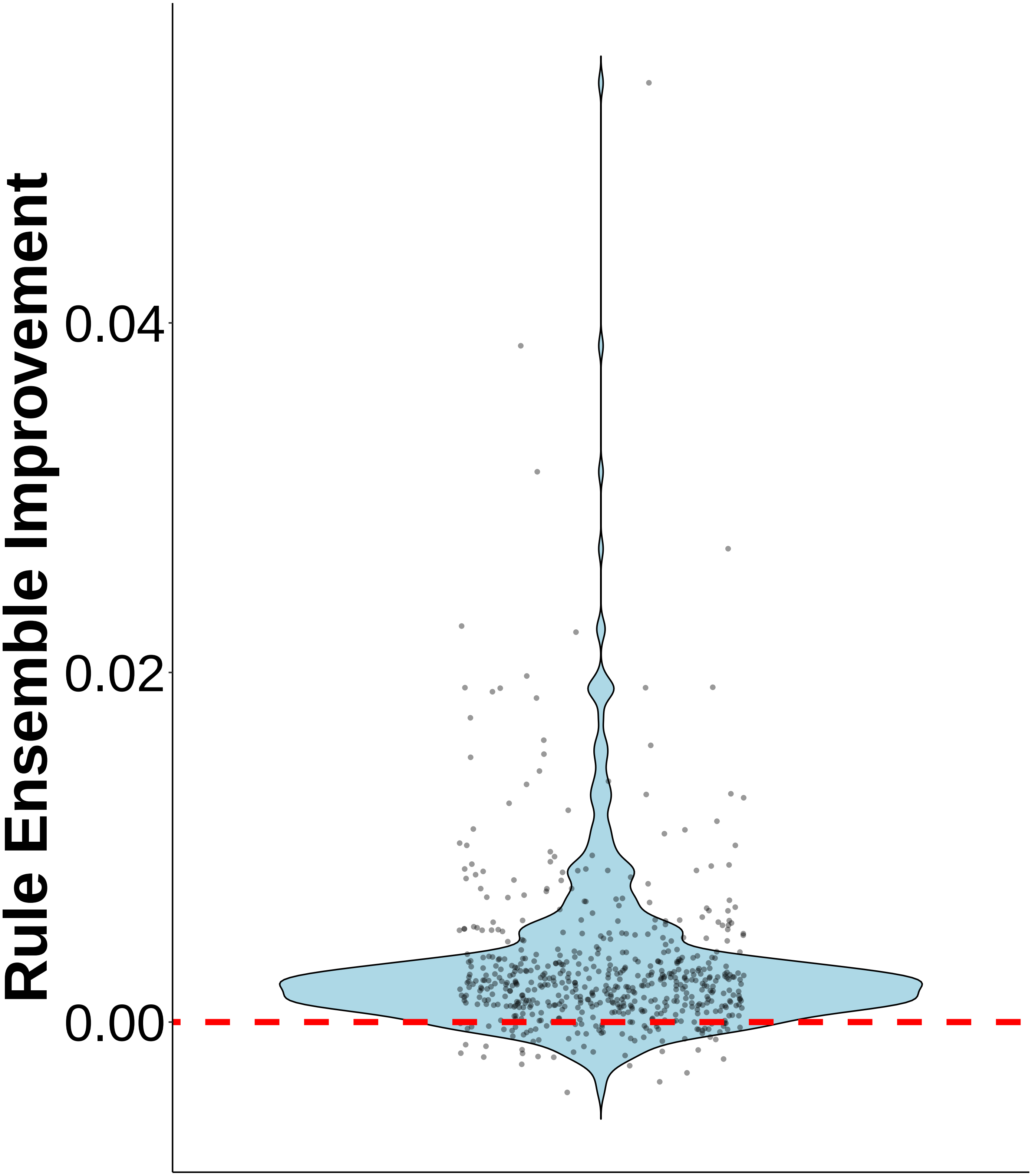

Figure 5: Violin plot of rule ensemble improvement over ATACorrect single nucleotide bias correction.

Figure 5 shows that 86.8% of positions within 10bp of a motif are improved by rule ensemble scaling over ATACorrect scaling. While this number is impressive, because of the number of modifications necessary to compare output from both methods we do not believe this is a meaningful comparison. We do however find that this comparison highlights the incompatibility between these two methods. We hope that this analysis serves as a satisfactory explanation as to why benchmarking these two methods is not possible.