Analysis of NOMe-seq data using the aaRon R package

Aaron Statham a.statham@garvan.org.au

November 17, 2014

Contents

1 Introduction

The NOMe-seq technique[1] is an extension of whole genome bisulfite sequencing[2] (WGBS) which enables the simultaneous assaying of DNA methylation and nucleosome occupancy at single nucleotide resolution. This vignette demonstrates tools to analyse and visualise the NOMe-seq profiles. This is illustrated using 4 human cell lines, work which was recently published in Genome Research[3].

1.1 The NOMe-seq assay

NOMe-seq stands for *Nucleosome Occupancy and Methylation sequencing* and enables the simultaneous interrogation of endogenous DNA methylation and nucleosome occupancy/DNA accessibility at a single nucleotide level. Endogenous methylcytosine in mammalian cells occurs almost exclusively in a CpG dinucleotide context. The NOMe-seq technique leverages this through the treatment of isolated cell nuclei with the GpC methyltransferase *M.CviPI*, adding a methyl group to the cytosines in GpC dinucleotides that are accessible to the enzyme, providing a methylation footprint of non-nucleosomal DNA. This treated DNA is then isolated and undergoes WGBS to determine the methylation status of every cytosine in the genome at single basepair resolution.

The endogenous and exogenous, *M.CviPI* added methylation is separable post-WGBS by whether the methylcysotines occur in a CpG or GpC sequence context. GpCpG sites are the exception, as the middle cytosine methylation is in both a CpG and GpC context; these sites are removed from the analysis. Additionally *M.CviPI* has a low ability to methylate CpCpG sites; these are also eliminated from the analysis.

This leaves **GCH** sites (a GpC *not* followed by a G; 123,015,862 present in the human genome hg19) available for the nucleosome occupancy/accessibility readout, and **WCG** sites (a CpG *not* preceded by a C or G; 20,475,403 present in the human genome hg19) available for endogenous methylation analysis.

1.2 How to use this manual

This document is split into two parts:

- Section 2 - the procedure to download, align, extract methylation/accessibility data and then join multiple samples into 'bigTable's of **GCH** and **WCG** methylation. These chunks of code are to be executed in *bash* and are highlighted in blue. If desired, this may be bypassed and previously analysed tables downloaded[4] as described in Section 2.7.
- Section 3 - the importation and analysis of processed NOMe-seq data, to be performed in *R*. These chunks of code are highlighteed in grey.

2 Processing of raw NOMe-seq data into tables of CpG and GpC methylation

This section details the pre-processing of raw NOMe-seq data. The sequencing data we are using has been deposited at the Sequence Read Archive (SRA). We will download the raw data from the SRA, convert to the 'fastq' format, align the reads to the human genome (build hg19) using *bwameth*[5, 6], extract methylation counts at **GCH** and **WCG** sites using *BisSNP*[7] and construct the final 'bigTable's ready for downstream analysis in *R*.

2.1 Installation of prerequisites

Firstly we must install the required tools for the analysis. These instructions have been tested on two variants of linux (Ubuntu 14.04 and CentOS 6.2) and require git and the python package manager pip.

```
# NOMe-seq-analysis - files required for this example analysis
# We will use this repository as the working directory for all downstream analysis
git clone https://github.com/astatham/NOMe-seq-analysis
cd NOMe-seq-analysis

# bwameth and its dependency toolshed - from https://github.com/brentp/bwa-meth
pip install toolshed
git clone https://github.com/brentp/bwa-meth.git

# BisSNP
wget http://ufpr.dl.sourceforge.net/project/bissnp/BisSNP-0.82.2/BisSNP-0.82.2.jar

# bwa
wget http://softlayer-sng.dl.sourceforge.net/project/bio-bwa/bwa-0.7.10.tar.bz2
tar -jxvf bwa-0.7.10.tar.bz2
rm bwa-0.7.10.tar.bz2
cd bwa-0.7.10; make; cd ..
```

```
# samtools
wget http://downloads.sourceforge.net/project/samtools/samtools/1.1/samtools-1.1.tar.bz2
tar -jxvf samtools-1.1.tar.bz2
rm samtools-1.1.tar.bz2
cd samtools-1.1; make; cd ..

# picard
wget -O picard-tools-1.122.zip \
    https://github.com/broadinstitute/picard/releases/download/1.122/picard-tools-1.122.zip
unzip picard-tools-1.122.zip
rm picard-tools-1.122.zip

# R and required packages
wget http://cran.r-project.org/src/base/R-3/R-3.1.1.tar.gz
tar -zxvf R-3.1.1.tar.gz
rm R-3.1.1.tar.gz
cd R-3.1.1
./configure
make
bin/R -e 'source("http://bioconductor.org/biocLite.R"); biocLite(c("GenomicRanges",
  "BSgenome.Hsapiens.UCSC.hg19","TxDb.Hsapiens.UCSC.hg19.knownGene", "AnnotationHub",
  "Repitools", "devtools", "ggplot2", "zoo", "minfi", "reshape2", "data.table", "R.utils"))'
bin/R -e 'devtools::install_github("astatham/aaRon")'
cd ..

# Set environmental variables & paths
BWAMETH="$PWD"/bwa-meth
PICARD="$PWD"/picard-tools-1.122
PATH="$PWD"/R-3.1.1/bin:"$PWD"/bwa-0.7.10:"$PWD"/samtools-1.1:$PATH
```

2.2 Creating an index of hg19 for alignment

We need to download a copy of the hg19 genome sequence from UCSC and create a *bwameth* index so that we can align our NOMe-seq reads to it.

```
mkdir index
cd index
wget http://hgdownload.cse.ucsc.edu/goldenPath/hg19/bigZips/chromFa.tar.gz
tar -zxvf chromFa.tar.gz
rm chromFa.tar.gz
for i in `seq 1 22` X Y M # Only include chr1-22, X, Y and M in index
do
    cat chr"$i".fa >> hg19.fa
done
samtools faidx hg19.fa
"$BWAMETH"/bwameth.py index hg19.fa
cd ..
```

2.3 Downloading raw data from the Short Read Archive

Now we download the raw data from the SRA. As each sample was sequenced over multiple lanes, there are more than one 'sra' file per cell line. The mapping between 'sra' files and cell lines is in the supplied 'files.txt'. The code below downloads the 'sra's with filenames corresponding to the cell line, converts 'sra's to 'fastq' pairs and concatenates the replicate lanes into a single 'fastq' pair per cell line, ready for alignment.

```
mkdir fastq
cd fastq
# download sra files using gnu parallel to perform downloads simultaneously
# can adjust the '-j' parameter to change the number of commands executed in parallel
cat files.txt | parallel --gnu --colsep ' ' -j 10 wget -O {2} {1}

# extract out fastq.gzs, and concatenate by cell line
ls *.sra | parallel --gnu -j 10 fastq-dump --split-files --gzip -F {}

parallel --gnu -j 4 "cat {}_?_1.fastq.gz > {}_1.fastq.gz" ::: HMEC MCF7 PrEC PC3
parallel --gnu -j 4 "cat {}_?_2.fastq.gz > {}_2.fastq.gz" ::: HMEC MCF7 PrEC PC3

# remove unneeded files
rm *.sra *_?_?.fastq.gz
cd ..
```

2.4 Alignment of reads to the human genome

Next we align the 'fastq' pairs of each sample to hg19 using *bwameth*. This step may take an extended amount of time as we are aligning 1.1 billion read pairs, so increasing the number of alignment threads (–threads parameter) or performing the alignment for each sample in parallel on a computational cluster is advisable. After alignment is complete, PCR duplicate reads are marked by *Picard MarkDuplicates*.

```
mkdir bams
for sample in HMEC MCF7 PrEC PC3
do
  # align
  "$BWAMETH"/bwameth.py --threads 32 --prefix bams/"$sample".align \
    --read-group @RG\\tID:"$sample"\\tSM:"$sample" --reference index/hg19.fa \
    fastq/"$sample"_1.fastq.gz fastq/"$sample"_2.fastq.gz

  # mark duplicates
  java -jar "$PICARD"/MarkDuplicates.jar I=bams/"$sample".align.bam O=bams/"$sample".bam  \
    M=bams/"$sample".metrics REMOVE_DUPLICATES=FALSE AS=TRUE CREATE_INDEX=TRUE
done
```

2.5 Extraction of methylation counts

Now we call *BisSNP* via *bwameth tabulate* once per cell line 'bam' file. *BisSNP* is a bisulfite aware genotyper extension to *GATK*[8], and exports methylation and genotype results in 'vcf' format, which *bwameth tabulate* then converts into a simpler 'bed'-like format. Again, this process can take an extended amount of time, so performing this step for all samples in parallel on a computational cluster is recommended. We then use the *R* script 'bissnp_to_tsv_nome.R' to separate the **GCH** and **WCG** methylation calls into separate files.

```
for sample in HMEC MCF7 PrEC PC3
do
  # extract methylation calls
  "$BWAMETH"/bwameth.py tabulate --bissnp BisSNP-0.82.2.jar --trim 4,4 --map-q 60 \
    --threads 32 --prefix "$sample".bissnp --reference index/hg19.fa \
    --nome bams/"$sample".bam
  # tabulate methylation calls
  R -f bissnp_to_tsv_nome.R --args "$sample" hg19
  gzip *.vcf *.bed
done
```

2.6 Combining samples into big tables

Finally we combine the individual samples **GCH** and **WCG** tables into a single 'bigTable' per type of methylation.

```
paste <(gunzip -c HMEC.GCH.tsv.gz) <(gunzip -c MCF7.GCH.tsv.gz | cut -f3,4) \
                                   <(gunzip -c PrEC.GCH.tsv.gz | cut -f3,4) \
                                   <(gunzip -c PC3.GCH.tsv.gz | cut -f3,4) \
      > bigTable.GCH.tsv
paste <(gunzip -c HMEC.CpGs.tsv.gz) <(gunzip -c MCF7.CpGs.tsv.gz | cut -f3,4) \
                                    <(gunzip -c PrEC.CpGs.tsv.gz | cut -f3,4) \
                                    <(gunzip -c PC3.CpGs.tsv.gz | cut -f3,4) \
      > bigTable.WCG.tsv
```

This completes the initial processing of raw NOMe-seq data into tables of methylation and accessibility for four cell lines. Alternatively, already processed tables from this same starting data have been processed and are downloadable[4].

2.7 Downloading of previously analysed tables for this experiment

```
wget http://zenodo.org/record/12454/files/bigTable.GCH.tsv.gz
wget http://zenodo.org/record/12454/files/bigTable.WCG.tsv.gz
gunzip bigTable.GCH.tsv.gz bigTable.WCG.tsv.gz
```

These 'bigTable's are now ready to be imported into *R* and analysed using *aaRon* in the next section.

3 Analysis of processed NOMe-seq data using aaRon

This section details the downstream analysis of the NOMe-seq data we just finished processing (or downloading). All of the following instructions are to be performed within the *R* environment.

3.1 Importing NOMe-seq data

The package *aaRon* may be downloaded directy from github using `devtools` if it needs to be installed or updated.

```
library(devtools)
install_github("astatham/aaRon")
```

Firstly we load the *R* package *aaRon*, and others required for the importation and analysis of NOMe-seq data.

```
library(aaRon)
library(data.table)
library(BSgenome.Hsapiens.UCSC.hg19)
```

Next we create a `samples` *data.frame* containing the names of the cell lines analysed, and which columns of the NOMe-seq tables contain methylation and coverage data.

```
samps <- c("HMEC", "MCF7", "PrEC", "PC3")
samples <- data.frame(Sample=samps,
                           C=paste0(samps, ".C"),
                         cov=paste0(samps, ".cov"), stringsAsFactors=FALSE, row.names=samps)
samples
```

```
##      Sample      C      cov
## HMEC   HMEC HMEC.C HMEC.cov
## MCF7   MCF7 MCF7.C MCF7.cov
## PrEC   PrEC PrEC.C PrEC.cov
## PC3     PC3  PC3.C  PC3.cov
```

Now we use the `fread` function from the *data.table* package to read in the two NOMe-seq 'bigTable's (`GCH` containing the accessibility data and `WCG` containing methylation data) and convert them into *GRanges* objects.

```
tab <- fread("bigTable.GCH.tsv", showProgress=FALSE)
GCH <- GRanges(tab$chr, IRanges(tab$position, width=1))
seqlengths(GCH) <- seqlengths(Hsapiens)[seqlevels(GCH)]
values(GCH) <- as.data.frame(tab)[,-c(1:2)]
head(GCH)
```

```
## GRanges object with 6 ranges and 8 metadata columns:
##       seqnames          ranges strand |    HMEC.C  HMEC.cov    MCF7.C  MCF7.cov    PrEC.C
##          <Rle>       <IRanges>  <Rle> | <integer> <integer> <integer> <integer> <integer>
##   [1]     chr1 [10482, 10482]      * |         0         0         0         1         1
##   [2]     chr1 [10486, 10486]      * |         0         0         0         1         0
##   [3]     chr1 [10490, 10490]      * |         0         0         0         0         0
##   [4]     chr1 [10494, 10494]      * |         0         0         0         0         0
##   [5]     chr1 [10521, 10521]      * |         0         1         0         0         0
##   [6]     chr1 [10526, 10526]      * |         0         0         0         0         0
##        PrEC.cov    PC3.C   PC3.cov
##       <integer> <integer> <integer>
##   [1]         4         1        10
##   [2]         3         0         0
##   [3]         0         0         0
##   [4]         0         0         0
##   [5]         0         1         7
##   [6]         0         0         0
##   -------
##   seqinfo: 93 sequences from an unspecified genome
```

```
tab <- fread("bigTable.WCG.tsv", showProgress=FALSE)
WCG <- GRanges(tab$chr, IRanges(tab$position, width=1))
seqlengths(WCG) <- seqlengths(Hsapiens)[seqlevels(WCG)]
values(WCG) <- as.data.frame(tab)[,-c(1:2)]
rm(tab)
head(WCG)
```

```
## GRanges object with 6 ranges and 8 metadata columns:
##       seqnames          ranges strand |    HMEC.C  HMEC.cov    MCF7.C  MCF7.cov    PrEC.C
##          <Rle>       <IRanges>  <Rle> | <integer> <integer> <integer> <integer> <integer>
##   [1]     chr1 [10469, 10469]      * |         0         0         0         0         0
##   [2]     chr1 [10542, 10542]      * |         1         1         0         0         0
##   [3]     chr1 [10563, 10563]      * |         0         0         0         0         0
##   [4]     chr1 [10577, 10577]      * |         0         0         0         0         0
##   [5]     chr1 [10609, 10609]      * |         0         0         0         0         0
##   [6]      chr1 [10620, 10620]      * |         0         0         0         0         0
##        PrEC.cov     PC3.C   PC3.cov
##       <integer> <integer> <integer>
##   [1]         0         0         0
##   [2]         0         1         2
##   [3]         0         0         0
##   [4]         0         0         0
##   [5]         0         0         0
##   [6]         0         0         0
##   -------
##   seqinfo: 93 sequences from an unspecified genome
```

As a check we confirm that all of the `C` and `cov` columns specified in `samples` are actually present in the `GCH` and `WCG` tables.

```
all(c(samples$C, samples$cov) %in% names(values(GCH)))

## [1] TRUE

all(c(samples$C, samples$cov) %in% names(values(WCG)))

## [1] TRUE
```

3.2 Detection of nucleosome depleted regions (NDRs)

The function `findNDRs` implements a sliding window chi-squared test to search for regions of statistically significantly increased GCH methylation, indicating a nucleosome depleted region (NDR). The default parameters are to test windows of 100bp in size every 20bp along each chromosome and to return regions of a minimum 140bp in size with a p-value less than 10e-15; these settings can be modified to suit the requirements of different studies.

warning: This procedure may take an extended amount of computation time when run on a complete datatset, so verbose progress is printed.

```
NDRs <- findNDRs(GCH, samples)

## Processing HMEC
##  - Counting Cs
##  - Counting Ts
##  - Calculating chisq pvalues
##  - Finding significant regions
## Processing MCF7
##  - Counting Cs
##  - Counting Ts
##  - Calculating chisq pvalues
##  - Finding significant regions
## Processing PrEC
##  - Counting Cs
##  - Counting Ts
##  - Calculating chisq pvalues
##  - Finding significant regions
## Processing PC3
##  - Counting Cs
##  - Counting Ts
##  - Calculating chisq pvalues
##  - Finding significant regions

NDRs
```

```
## GRangesList object of length 4:
## $HMEC
## GRanges object with 29125 ranges and 1 metadata column:
##           seqnames                ranges strand   |          p.mean
##              <Rle>             <IRanges>  <Rle>   |       <numeric>
##       [1]     chr1     [901441, 901640]       *   | 12.6684037861397
##       [2]     chr1     [948721, 948980]       *   | 13.9834584345369
##       [3]     chr1     [960061, 960220]       *   | 10.7333675228193
##       [4]     chr1     [976141, 976340]       *   | 12.7987391225026
##       [5]     chr1     [999341, 999520]       *   | 11.4601048241568
##       ...      ...                  ...     ... ...              ...
##   [29121]     chrY [10028701, 10028880]       *   | 10.8408976586317
##   [29122]     chrY [10036501, 10036680]       *   | 18.5755970398289
##   [29123]     chrY [10037721, 10038060]       *   | 21.1278491739742
##   [29124]     chrY [58977921, 58978060]       *   | 12.4076054345822
##   [29125]     chrM [     421,      600]       *   | 47.1296908846104
##
## ...
## <3 more elements>
## -------
## seqinfo: 93 sequences from an unspecified genome
```

`findNDRs` returns a *GRangesList* of detected NDRs. We can explore the number of NDRs in each sample and the overlaps of NDRs between samples using `elementLengths` and `grangesVenn`, the results of which are displayed in *Figure 1*.

```
elementLengths(NDRs)

##  HMEC  MCF7  PrEC   PC3
## 29125 28543 69127 28892

grangesVenn(NDRs)
```

3.3 Plotting aggregate NOMe-seq profiles around regions of interest

aaRon includes three functions for the plotting of averaged NOMe-seq profiles around regions of interest:

- `methylationBiPlot` plots **both** methylation and accessibility profiles surrounding a *GRanges* of supplied regions for a **single** sample.
- `methylationPlotRegions` plots **either** methylation or accessibility profiles surrounding a *GRangesList* of multiple sets of supplied regions for a **single** sample.
- `methylationPlotSamples` plots **either** methylation or accessibility profiles surrounding a *GRanges* of supplied regions for **multiple** samples.

Figure 2 is a `methylationBiPlot` showing both the methylation and accessibility profiles surrounding 51,385 unique transcription start sites (TSSs) as defined by the UCSC knownGene database.

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)

# Get the first transcribed base of every knownGene transcript
TSS <- promoters(transcripts(TxDb.Hsapiens.UCSC.hg19.knownGene), 0, 1)
# Remove redundant TSSs
TSS <- unique(TSS)
methylationBiPlot(GCH, WCG, TSS, samples["HMEC", ], up=3000, down=3000, addN=FALSE)
```

Next, let's use `methylationPlotRegions` to plot profiles around different classes of TSSs in MCF7. We use the *AnnotationHub* to obtain regions of the MCF7 genome that are marked by H3K4me3, a histone modification associated

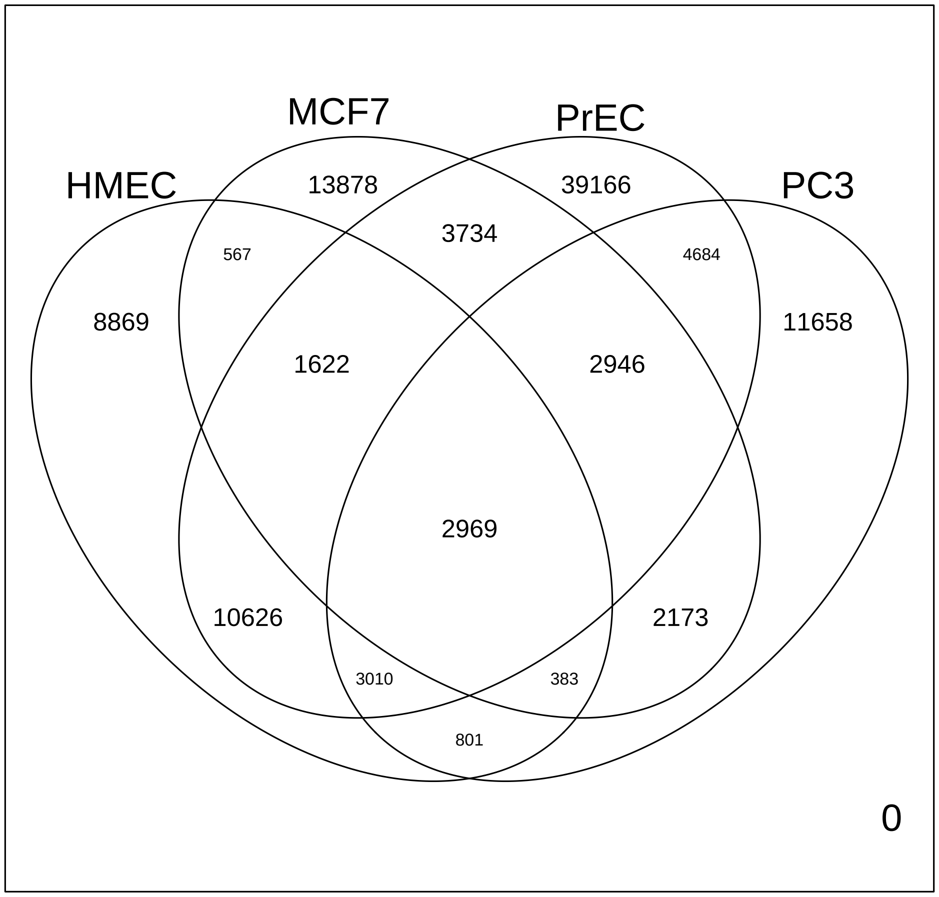


Figure 1: Overlap of NDRs between the four cell lines

with active TSSs. We then split the TSSs into whether they are close to a MCF7 H3K4me3 peak (within 1kb) or not, and then plot the profiles of these two sets of TSSs on the same graph for a direct comparison. *Figure 3* shows a comparison of accessibility and methylation between TSSs marked with H3K4me3 or unmarked, demonstrating the correlation between the H3K4me3 modification, loss of nucleosome occupancy and decreased DNA methylation.

```
library(AnnotationHub)
ah = AnnotationHub()
# download ENCODE MCF7 H3K4me3 peaks
MCF7.K4me3 <-
  ah$goldenpath.hg19.encodeDCC.wgEncodeUwHistone.wgEncodeUwHistoneMcf7H3k4me3StdPkRep1.narrowPeak_0.0.1.RD
head(MCF7.K4me3)
```

```
## GRanges object with 6 ranges and 6 metadata columns:
##       seqnames             ranges strand |        name     score signalValue    pValue
##          <Rle>          <IRanges>  <Rle> | <character> <integer>   <numeric> <numeric>
##   [1]     chr1 [540640, 540790]      * |           .         0          23  19.89520
##   [2]     chr1 [569780, 569930]      * |           .         0          19   6.07811
##   [3]     chr1 [713300, 713450]      * |           .         0          54  49.11670
```

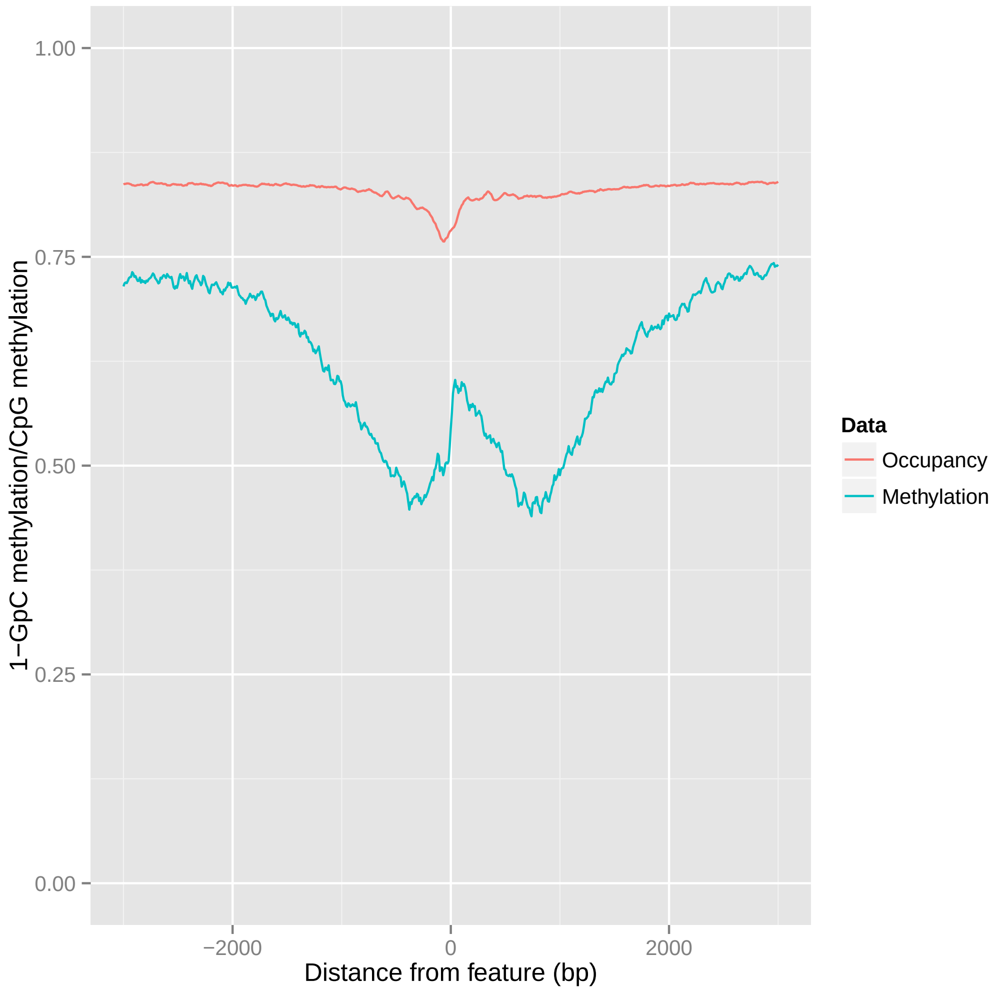
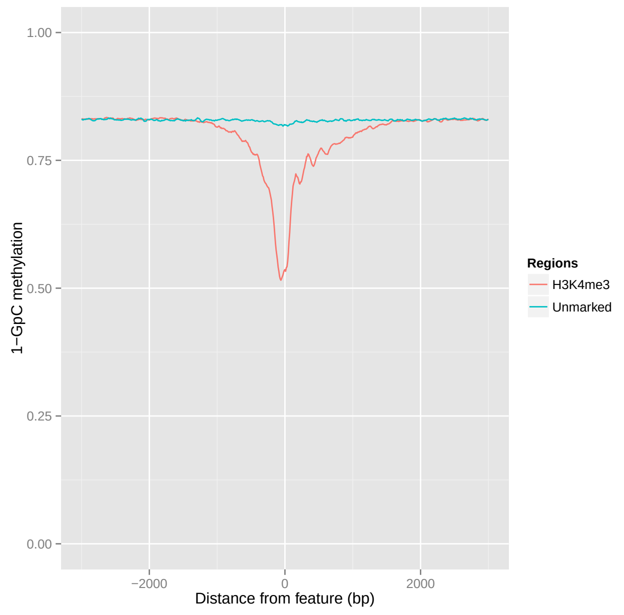
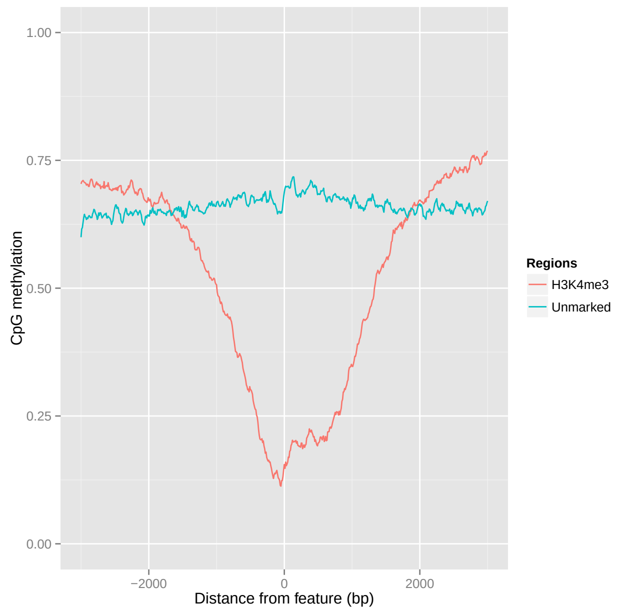


Figure 2: HMEC Methylation and accessibility at knownGene TSSs

```
##   [4]     chr1 [713520, 713670]      * |           .         0          26  44.82250
##   [5]     chr1 [713900, 714050]      * |           .         0          94  67.71310
##   [6]     chr1 [714180, 714330]      * |           .         0          27  67.71310
##          qValue      peak
##       <numeric> <integer>
##   [1]        -1        -1
##   [2]        -1        -1
##   [3]        -1        -1
##   [4]        -1        -1
##   [5]        -1        -1
##   [6]        -1        -1
##   -------
##   seqinfo: 23 sequences from hg19 genome

# Split TSSs by whether they are close (within 1kb) of a H3K4me3 peak
TSS.distance <- distanceToNearest(TSS, MCF7.K4me3)
TSS.split <- split(TSS, ifelse(values(TSS.distance)$distance<1000, "H3K4me3", "Unmarked"))
```



(a) Accessibility



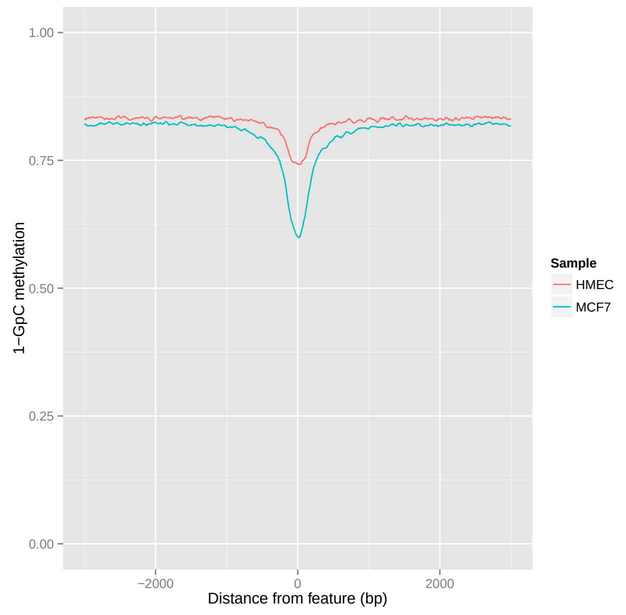
(b) Methylation

Figure 3: MCF7 TSS profiles split by H3K4me3 status

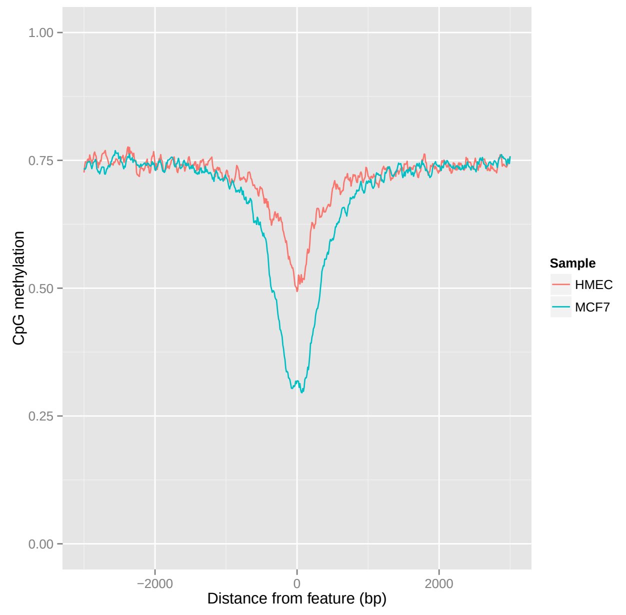
```
elementLengths(TSS.split)

##  H3K4me3 Unmarked
##    21148    27690

methylationPlotRegions(GCH, TSS.split, samples["MCF7", ], GpC=TRUE, up=3000, down=3000, addN=FALSE)
methylationPlotRegions(WCG, TSS.split, samples["MCF7", ], GpC=FALSE, up=3000, down=3000, addN=FALSE)
```

Finally, we will use `methylationPlotSamples` to compare the profiles around a tissue specific transcription factor - estrogen receptor alpha (ER) - which is overexpressed in MCF7 versus HMEC. We will first download the sites where ER has been shown to be bound in MCF7[9], however this data was analysed using version hg18 of the human genome, so first we must use `liftOver` to convert the ER peak co-ordinates to hg19. This procedure has already been performed and the resulting file 'MCF7.ER.hg19.bed' included in this github repository.

```
# Download UCSC liftover chain for hg18 to hg19 conversion
download.file("http://hgdownload.cse.ucsc.edu/goldenPath/hg18/liftOver/hg18ToHg19.over.chain.gz",
  "hg18ToHg19.over.chain.gz")

# The chain needs to be gunzip'd prior to being imported
library(R.utils)
gunzip("hg18ToHg19.over.chain.gz")
chain <- import.chain("hg18ToHg19.over.chain")

# Import the published set of MCF7 ER peaks and convert coordinates to hg19
MCF7.ER.hg18 <- import(
  "http://www.carroll-lab.org.uk/FreshFiles/Data/Data_Sheet_3/MCF7_ER_binding.bed")
MCF7.ER <- liftOver(MCF7.ER.hg18, chain)
# Only keep sites that are not split during the liftOver (ie are of length 1)
MCF7.ER <- unlist(MCF7.ER[elementLengths(MCF7.ER)==1])
```



(a) Accessibility



(b) Methylation

Figure 4: MCF7 ER binding sites in both breast cell lines

```
export(MCF7.ER, "MCF7.ER.hg19.bed")
```

Figure 4 displays the methylation and accessibility around these MCF7 ER binding sites in both HMEC and MCF7. ER is overexpressed in MCF7, and as expected we observe that these sites are much more accessible and show a decrease in DNA methylation compared to HMEC.

```
MCF7.ER <- import("MCF7.ER.hg19.bed")
methylationPlotSamples(GCH, MCF7.ER, samples[c("HMEC", "MCF7"),], GpC=TRUE, up=3000, down=3000, addN=FALSE
methylationPlotSamples(WCG, MCF7.ER, samples[c("HMEC", "MCF7"),], GpC=FALSE, up=3000, down=3000, addN=FALS
```

4 Conclusions

This manual details the step-by-step processing of raw NOMe-seq data into 'bigTable's, demonstrates some of the downstream analyses that can be performed using the *aaRon* package and is a resource that facilitates the reproducible analysis and interpretation of NOMe-seq data.

5 Session info

```
sessionInfo()

## R version 3.1.1 (2014-07-10)
## Platform: x86_64-unknown-linux-gnu (64-bit)
##
## locale:
##  [1] LC_CTYPE=en_AU.UTF-8       LC_NUMERIC=C               LC_TIME=en_AU.UTF-8
##  [4] LC_COLLATE=en_AU.UTF-8     LC_MONETARY=en_AU.UTF-8    LC_MESSAGES=en_AU.UTF-8
##  [7] LC_PAPER=en_AU.UTF-8       LC_NAME=C                  LC_ADDRESS=C
## [10] LC_TELEPHONE=C             LC_MEASUREMENT=en_AU.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats4    parallel  stats     graphics  grDevices utils     datasets  methods
## [9] base
##
## other attached packages:
##  [1] AnnotationHub_1.6.0                     TxDb.Hsapiens.UCSC.hg19.knownGene_3.0.0
##  [3] GenomicFeatures_1.18.1                  AnnotationDbi_1.28.0
##  [5] Biobase_2.26.0                          BSgenome.Hsapiens.UCSC.hg19_1.4.0
##  [7] BSgenome_1.34.0                         rtracklayer_1.26.1
##  [9] Biostrings_2.34.0                       XVector_0.6.0
## [11] data.table_1.9.4                        aaRon_0.8.3
## [13] GenomicRanges_1.18.1                    GenomeInfoDb_1.2.0
## [15] IRanges_2.0.0                           S4Vectors_0.4.0
## [17] BiocGenerics_0.12.0                     knitr_1.7
##
## loaded via a namespace (and not attached):
##   [1] affy_1.44.0                  affyio_1.34.0
##   [3] annotate_1.44.0              aroma.affymetrix_2.12.0
##   [5] aroma.apd_0.5.0              aroma.core_2.12.1
##   [7] base64_1.1                   base64enc_0.1-2
##   [9] BatchJobs_1.4                BBmisc_1.7
##  [11] beanplot_1.2                 BiocInstaller_1.16.0
##  [13] BiocParallel_1.0.0           BiocStyle_1.4.1
##  [15] biomaRt_2.22.0               bitops_1.0-6
##  [17] brew_1.0-6                   bumphunter_1.6.0
##  [19] Category_2.32.0              caTools_1.17.1
##  [21] checkmate_1.5.0              chron_2.3-45
##  [23] cluster_1.15.3               codetools_0.2-9
##  [25] colorspace_1.2-4             DBI_0.3.1
##  [27] digest_0.6.4                 DNAcopy_1.40.0
##  [29] doRNG_1.6                    edgeR_3.8.2
##  [31] evaluate_0.5.5               fail_1.2
##  [33] foreach_1.4.2                formatR_1.0
##  [35] gdata_2.13.3                 genefilter_1.48.1
##  [37] GenomicAlignments_1.2.0      ggplot2_1.0.0
##  [39] gplots_2.14.2                graph_1.44.0
##  [41] grid_3.1.1                   gridSVG_1.4-0
##  [43] GSEABase_1.28.0              gsmoothr_0.1.7
##  [45] gtable_0.1.2                 gtools_3.4.1
##  [47] highr_0.3                    htmltools_0.2.6
##  [49] httpuv_1.3.0                 httr_0.5
##  [51] illuminaio_0.8.0             interactiveDisplay_1.4.0
##  [53] interactiveDisplayBase_1.4.0 iterators_1.0.7
##  [55] KernSmooth_2.23-13           labeling_0.3
##  [57] lattice_0.20-29              limma_3.22.1
##  [59] locfit_1.5-9.1               MASS_7.3-35
##  [61] Matrix_1.1-4                 matrixStats_0.10.3
##  [63] mclust_4.4                   mime_0.2
##  [65] minfi_1.12.0                 multtest_2.22.0
##  [67] munsell_0.4.2              nlme_3.1-118
##  [69] nor1mix_1.2-0              pkgmaker_0.22
##  [71] plyr_1.8.1                 preprocessCore_1.28.0
##  [73] proto_0.3-10               PSCBS_0.43.0
##  [75] quadprog_1.5-5             R6_2.0
##  [77] RBGL_1.42.0                R.cache_0.10.0
##  [79] RColorBrewer_1.0-5         Rcpp_0.11.3
##  [81] RCurl_1.95-4.3             R.devices_2.12.0
##  [83] registry_0.2               Repitools_1.12.0
##  [85] reshape_0.8.5              reshape2_1.4
##  [87] R.filesets_2.6.0           R.huge_0.8.0
##  [89] Ringo_1.30.0               rjson_0.2.14
##  [91] RJSONIO_1.3-0              R.methodsS3_1.6.1
##  [93] rngtools_1.2.4             R.oo_1.18.0
##  [95] R.rsp_0.19.0               Rsamtools_1.18.0
##  [97] Rsolnp_1.14                RSQLite_0.11.4
##  [99] R.utils_1.34.0             scales_0.2.4
## [101] sendmailR_1.2-1            shiny_0.10.2.1
## [103] siggenes_1.40.0            splines_3.1.1
## [105] stringr_0.6.2              survival_2.37-7
## [107] tools_3.1.1                truncnorm_1.0-7
## [109] vsn_3.34.0                 XML_3.98-1.1
## [111] xtable_1.7-4               zlibbioc_1.12.0
## [113] zoo_1.7-11
```

References

[1] Theresa K Kelly, Yaping Liu, Fides D Lay, Gangning Liang, Benjamin P Berman, and Peter A Jones. Genome-wide mapping of nucleosome positioning and dna methylation within individual dna molecules. *Genome research*, 22(12):2497–2506, 2012.

[2] Ryan Lister, Mattia Pelizzola, Robert H Dowen, R David Hawkins, Gary Hon, Julian Tonti-Filippini, Joseph R Nery, Leonard Lee, Zhen Ye, Que-Minh Ngo, et al. Human dna methylomes at base resolution show widespread epigenomic differences. *nature*, 462(7271):315–322, 2009.

[3] Phillippa C Taberlay, Aaron L Statham, Theresa K Kelly, Susan J Clark, and Peter A Jones. Reconfiguration of nucleosome-depleted regions at distal regulatory elements accompanies dna methylation of enhancers and insulators in cancer. *Genome research*, 24(9):1421–1432, 2014.

[4] Aaron Luke Statham. Processed NOMe-seq data for four human cell lines. Oct 2014. URL: `http://dx.doi.org/10.5281/zenodo.12454`, `doi:{10.5281/zenodo.12454}`.

[5] Brent S. Pedersen, Kenneth Eyring, Subhajyoti De, Ivana V. Yang, and David A. Schwartz. Fast and accurate alignment of long bisulfite-seq reads, 2014. `arXiv:arXiv:1401.1129`.

[6] Brent S. Pedersen. align bs-seq reads and extract methylation without intermediate temp files. `https://github.com/brentp/bwa-meth`, 2014.

[7] Yaping Liu, Kimberly D Siegmund, Peter W Laird, and Benjamin P Berman. Bis-snp: Combined dna methylation and snp calling for bisulfite-seq data. *Genome Biol*, 13(7):R61, 2012.

[8] Mark A DePristo, Eric Banks, Ryan Poplin, Kiran V Garimella, Jared R Maguire, Christopher Hartl, Anthony A Philippakis, Guillermo del Angel, Manuel A Rivas, Matt Hanna, et al. A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nature genetics*, 43(5):491–498, 2011.

[9] Antoni Hurtado, Kelly A Holmes, Caryn S Ross-Innes, Dominic Schmidt, and Jason S Carroll. Foxa1 is a key determinant of estrogen receptor function and endocrine response. *Nature genetics*, 43(1):27–33, 2011.