Predicting the Chromatin Conformation From ChIPseq CebpE: Peak Calls

Ricky Lim[1], Samuel Collombet[2], Agus Salim[3], Touati Benoukraf[1]

[1]Cancer Science Institute of Singapore, National University of Singapore

[2]Institut de Biologie de l'Ecole Normale Superieur de Paris

[3]Department of Mathematics and Statistics

benoukraf@nus.edu.sg

```
work_dir = '/home/ricky/Rlim/ChromatinConformation/PeakCalls/CebpE/'
source('/home/ricky/Rlim/ChromatinConformation/PeakCalls/PeakCalls.R')
```

1 Peak Calls using jahmm

jahmm (just another hidden markov model) package is available on github. jahmm is a discretizer to call peaks from ChIPseq data.

1.1 Data Preparation

The input of jahmm is the binned-reads from aligned ChIPseq data.

In this experiment, the dataset is CEBP-Epsilon dataset from Koeffler lab coupled with its input sample. This dataset was binned into 300bp and 3000bp bins as the resolution of the ChIPseq peaks. The input data was preprocessed (binning) as follows:

```
#1. Following read alignment using bowtie2, bam format is converted into bed
bedtools bamtobed -i <bam> > *.bed
ls CebpE/Input/*.bam | parallel ``bedtools bamtobed -i {} > {.}.bed''

#2. The converted bed files are then binned into 300bp and 3000bp bins
ls CebpE/Input/KoefflerLab_BM_*.bed |
parallel ``./binitBed.py -b 300 -l 'mm10' -F 'bed' -n 2 -od 'CebpE/Input/' {} '' &
ls CebpE/Input/KoefflerLab_BM_*.bed |
parallel ``./binitBed.py -b 3000 -l 'mm10' -F 'bed' -n 2 -od 'CebpE/Input/' {} '' &

# add header
sed -i '1 i\chr\tstart\tend\tInput'
# combined the sample with input for ChIPseq profile
cut -f4 3000bin-KoefflerLab_BM_ChIPseq_CebpE_mm10_q10rmdup.bed |
paste 3000bin-KoefflerLab_BM_ChIPseq_Input_mm10_q10rmdup.bed - >
3000bin-KoefflerLab_BM_ChIPseq_CebpE_Input.bed
cut -f4 300bin-KoefflerLab_BM_ChIPseq_CebpE_mm10_q10rmdup.bed |
paste 300bin-KoefflerLab_BM_ChIPseq_Input_mm10_q10rmdup.bed - >
300bin-KoefflerLab_BM_ChIPseq_CebpE_Input.bed
```

```
KoefflerLab_BM_ChIPseq_CebpE_3000 <- read.table(paste0(work_dir,
                          'Input/3000bin-KoefflerLab_BM_ChIPseq_CebpE_Input.bed'),
                                                header=T)
KoefflerLab_BM_ChIPseq_CebpE_300 <- read.table(paste0(work_dir,
                          'Input/300bin-KoefflerLab_BM_ChIPseq_CebpE_Input.bed'),
                                               header=T)

head(KoefflerLab_BM_ChIPseq_CebpE_3000)

##    chr start   end Input CebpE
## 1 chr1     1  3000     0     0
## 2 chr1  3001  6000     0     0
## 3 chr1  6001  9000     0     0
## 4 chr1  9001 12000     0     0
## 5 chr1 12001 15000     0     0
## 6 chr1 15001 18000     0     0

KoefflerLab_BM_ChIPseq_CebpE_300_hmm <- KoefflerLab_BM_ChIPseq_CebpE_300[, c(1,4,5)]
KoefflerLab_BM_ChIPseq_CebpE_3000_hmm <- KoefflerLab_BM_ChIPseq_CebpE_3000[, c(1,4,5)]
```

1.2 Peak Calling: jahmm fit

In order to fit the ChIPseq read distributions using hidden markov model, three states were assumed. State 0 and 1 corresponds to non-targets and state 2 denotes the targets or ChIPseq peaks.

The fittings of jahmm on this dataset were shown in tables (see table below) and visualized in figures 1 and 2 for 300bp and 3000bp peak resolutions, respectively.

```
set.seed(12345)
fit_300 <- jahmm(KoefflerLab_BM_ChIPseq_CebpE_300_hmm)
fit_3000 <- jahmm(KoefflerLab_BM_ChIPseq_CebpE_3000_hmm)

plotFit(KoefflerLab_BM_ChIPseq_CebpE_300_hmm, fit_300, 1:100000,
        paste0(work_dir, 'Output/KoefflerLab_BM_ChIPseq_CebpE_300_Rjahmm.pdf'))

## pdf
##   2

plotFit(KoefflerLab_BM_ChIPseq_CebpE_3000_hmm, fit_3000, 1:10000,
        paste0(work_dir, 'Output/KoefflerLab_BM_ChIPseq_CebpE_3000_Rjahmm.pdf'))

## pdf
##   2

no_fit_300 <- table(fit_300$path)
no_fit_300 <- as.data.frame(no_fit_300)
colnames(no_fit_300) <- c('state', 'freq')
no_fit_3000 <- table(fit_3000$path)
no_fit_3000 <- as.data.frame(no_fit_3000)
colnames(no_fit_3000) <- c('state', 'freq')
```

State Frequency of the Fitted ChiPseq

	state	freq
1	0	4703674
2	1	4317233
3	2	64154

Table 1: Frequency of States in 300bp Peak Resolution

	state	freq
1	0	498408
2	1	393600
3	2	16490

Table 2: Frequency of States in 3000bp Peak Resolution

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

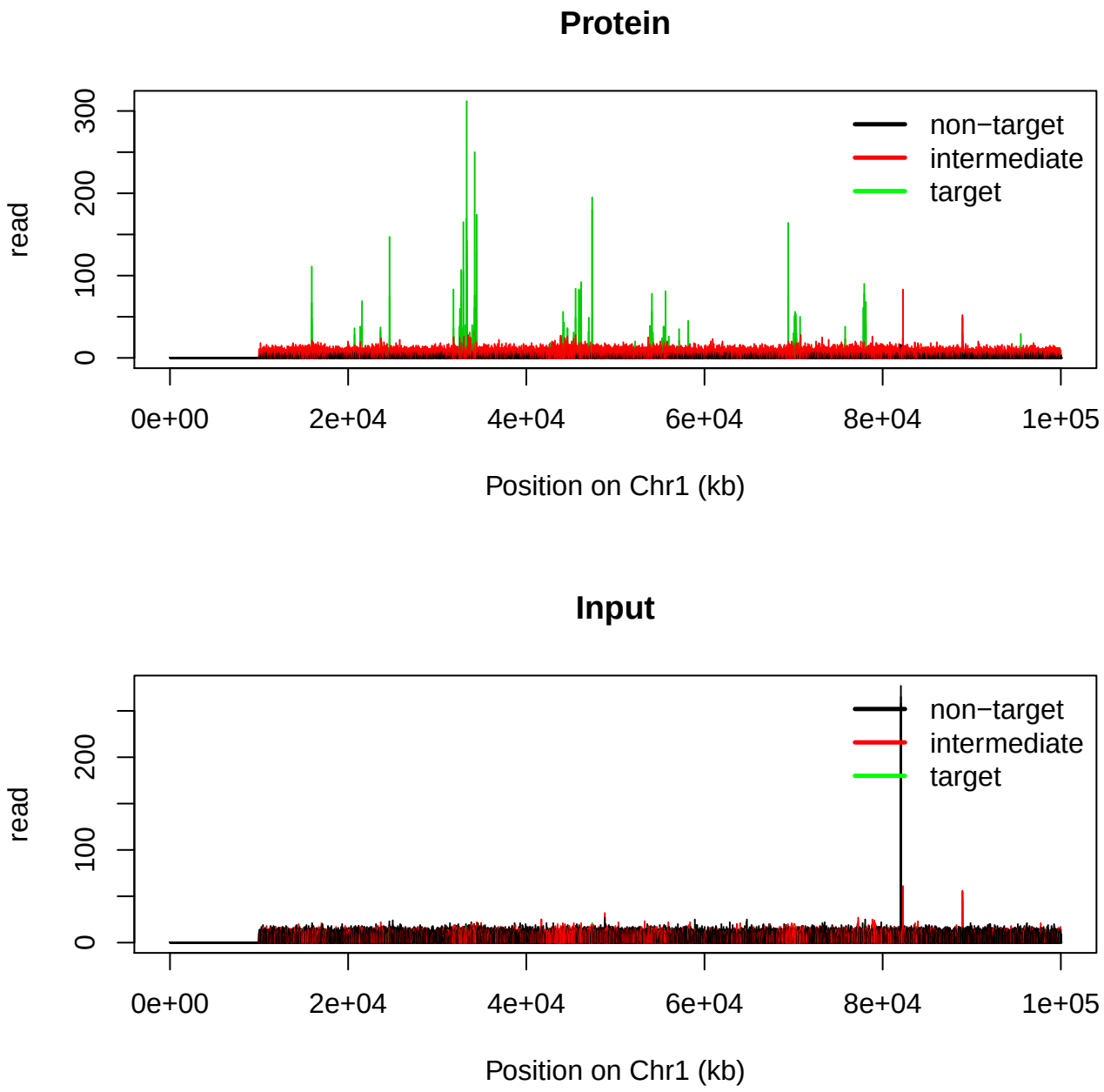


Figure 1: ChipSeq Peak Calls of CebpE: 300bp

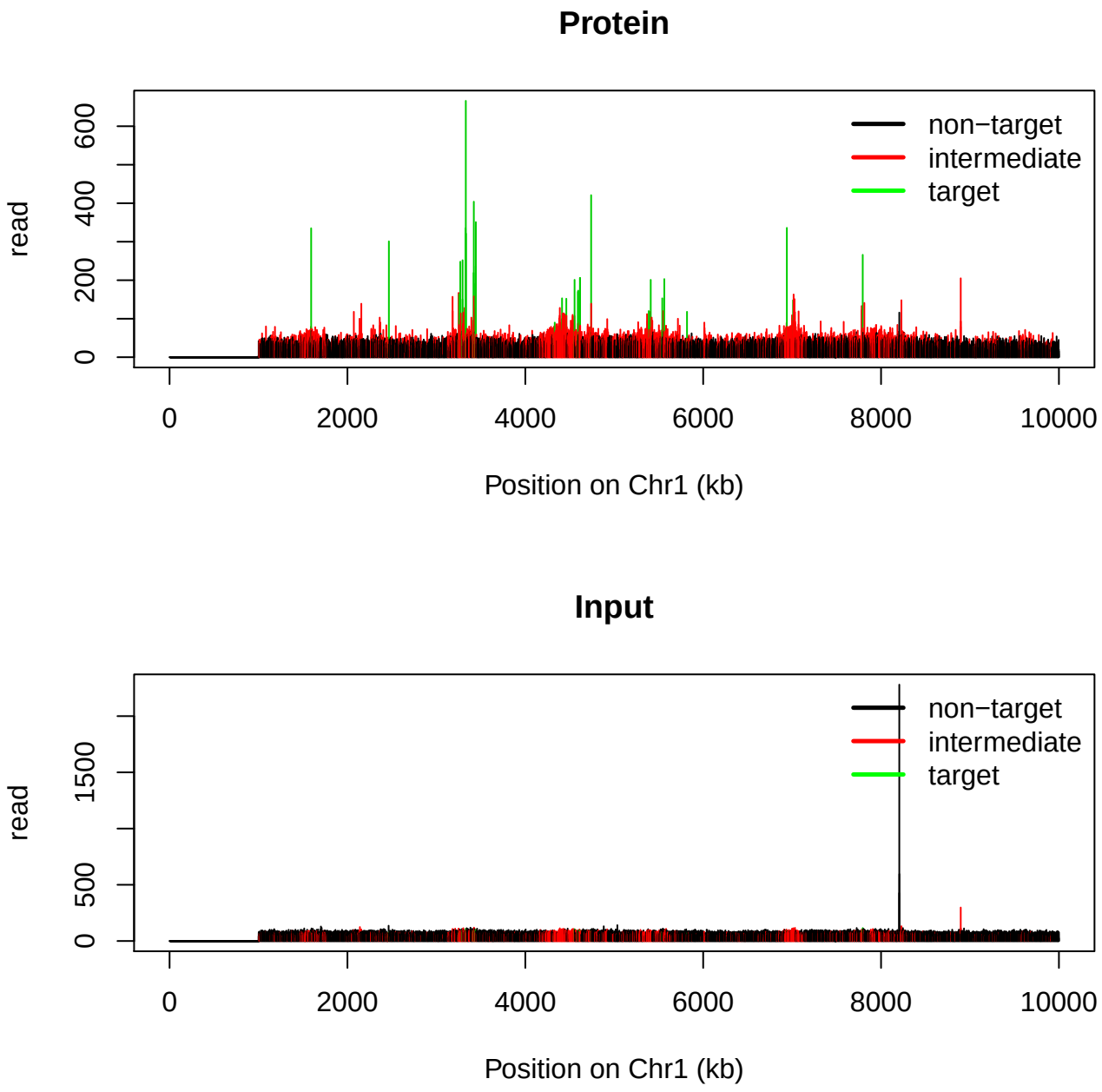


Figure 2: ChipSeq Peak Calls of CebpE: 3000bp

1.3 PeakCalls Output

The identified targets which are the ChIPseq peaks fitted with state 2 were produced as outputs.

```
KoefflerLab_BM_ChIPseq_CebpE_300$path <- fit_300$path
KoefflerLab_BM_ChIPseq_CebpE_3000$path <- fit_3000$path
head(KoefflerLab_BM_ChIPseq_CebpE_300)

##    chr start  end Input CebpE path
## 1 chr1     1  300     0     0    0
## 2 chr1   301  600     0     0    0
## 3 chr1   601  900     0     0    0
## 4 chr1   901 1200     0     0    0
## 5 chr1  1201 1500     0     0    0
## 6 chr1  1501 1800     0     0    0

KoefflerLab_BM_ChIPseq_CebpE_300_targets <- with(KoefflerLab_BM_ChIPseq_CebpE_300,
                                subset(KoefflerLab_BM_ChIPseq_CebpE_300, path == 2))
KoefflerLab_BM_ChIPseq_CebpE_3000_targets<- with(KoefflerLab_BM_ChIPseq_CebpE_3000,
                                subset(KoefflerLab_BM_ChIPseq_CebpE_3000, path == 2))

targets_300 <- nrow(KoefflerLab_BM_ChIPseq_CebpE_300_targets)
targets_3000 <- nrow(KoefflerLab_BM_ChIPseq_CebpE_3000_targets)

write.table(KoefflerLab_BM_ChIPseq_CebpE_300_targets,
            paste0(work_dir, 'Output/KoefflerLab_BM_ChIPseq_CebpE_300_targets.bed'),
            row.names=F, quote=F, sep='\t')
write.table(KoefflerLab_BM_ChIPseq_CebpE_3000_targets,
            paste0(work_dir, 'Output/KoefflerLab_BM_ChIPseq_CebpE_3000_targets.bed'),
            row.names=F, quote=F, sep='\t')
```

2 MACS versus jaHMM

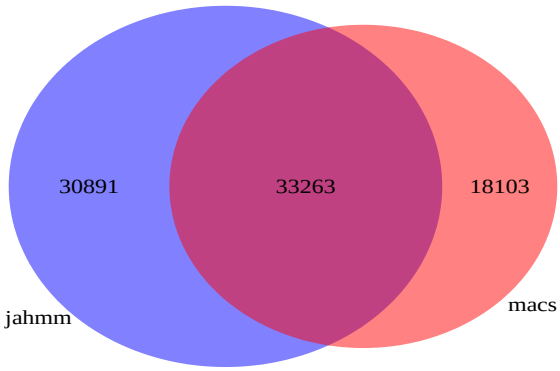
```
    # MACS peak called using the bash script ./macsPeakCalling

    # The peak summits were then binned into 300-bins
    ./binitBed.py -b 300 -l 'mm10' -F 'bed' -n 2 -od 'CebpE/Output/'
    CebpE/Output/KoefflerLab_BM_ChIPseq_CebpE_mm10_summits.bed
    # filter out the bins without peak summits
     awk '{if($4 != 0) print $0;}' 300bin-KoefflerLab_BM_ChIPseq_CebpE_mm10_summits.bed >
     300bin-KoefflerLab_BM_ChIPseq_CebpE_mm10_summits_filtered.bed
```

```
macs_targets <- read.table(paste0(work_dir,
                    'Output/300bin-KoefflerLab_BM_ChIPseq_CebpE_mm10_summits_filtered.bed'))
colnames(macs_targets) <- c('chr', 'start', 'end', 'peakScore')
# bam_scores for number of reads for target and input
bam_scores <- read.table(paste0(work_dir,
                    'Input/300bin-KoefflerLab_BM_ChIPseq_CebpE_Input.bed'), header=T)

# substitute peakScore with the bam score
```

```
macs_targets <- merge(macs_targets, bam_scores, by=c('chr', 'start', 'end') )
macs_targets <- macs_targets[, c(1,2,3,5,6)]


jahmm_targets <- KoefflerLab_BM_ChIPseq_CebpE_300_targets[,c(1,2,3,4,5)]
rownames(jahmm_targets) <- 1:nrow(jahmm_targets)
macs_targets <- data.table(macs_targets, key=c('chr', 'start', 'end'))
jahmm_targets <- data.table(jahmm_targets, key=c('chr', 'start', 'end'))

macs_targetsOnly <- macs_targets[!jahmm_targets]
jahmm_targetsOnly <- jahmm_targets[!macs_targets]
macs_jahmm <- merge(macs_targets, jahmm_targets,
                    by=c('chr', 'start', 'end'))

head(macs_jahmm)

##     chr   start     end Input.x CebpE.x Input.y CebpE.y
## 1: chr1 4774801 4775100      10      67      10      67
## 2: chr1 4775401 4775700       4     111       4     111
## 3: chr1 6215401 6215700      11      28      11      28
## 4: chr1 6406201 6406500      12      38      12      38
## 5: chr1 6467101 6467400       3      69       3      69
## 6: chr1 7088701 7089000       9      37       9      37

macs_jahmm_targets <- macs_jahmm[, 1:5, with=F]
head(macs_jahmm_targets)

##     chr   start     end Input.x CebpE.x
## 1: chr1 4774801 4775100      10      67
## 2: chr1 4775401 4775700       4     111
## 3: chr1 6215401 6215700      11      28
## 4: chr1 6406201 6406500      12      38
## 5: chr1 6467101 6467400       3      69
## 6: chr1 7088701 7089000       9      37

setnames(macs_jahmm_targets, c('chr', 'start', 'end', 'Input.x', 'CebpE.x'),
                             c('chr', 'start', 'end', 'Input', 'CebpE'))
```

```
plotTargetInput(macs_targetsOnly, 1:500, ylim= c(0,100),
                'Output/KoefflerLab_BM_ChIPseq_CebpE_macs_targets.pdf',
                title_f = 'macs Targets Only')

## Saving 7 x 7 in image

plotTargetInput(jahmm_targetsOnly, 1:500, ylim = c(0,100),
                'Output/KoefflerLab_BM_ChIPseq_CebpE_jahmm_targets.pdf',
                title_f = 'jahmm Targets Only')

## Saving 7 x 7 in image

plotTargetInput(macs_jahmm_targets, 1:500, ylim=c(0,100),
                'Output/KoefflerLab_BM_ChIPseq_CebpE_macs_jahmm_targets.pdf',
                title_f = 'macs and jahmm Targets')

## Saving 7 x 7 in image
```

```
# plot venn diagram
pdf(paste0(work_dir, 'Output/KoefflerLab_BM_ChIPseq_CebpE_macs_jahmm_venn.pdf'),
    pointsize=21)
v <- draw.pairwise.venn(nrow(macs_targets), nrow(jahmm_targets), nrow(macs_jahmm),
                        c('macs', 'jahmm'),
                        fill=c('red', 'blue'), lty='blank',
                        cex = 1, lwd = 1)
dev.off()

## pdf
##   2
```



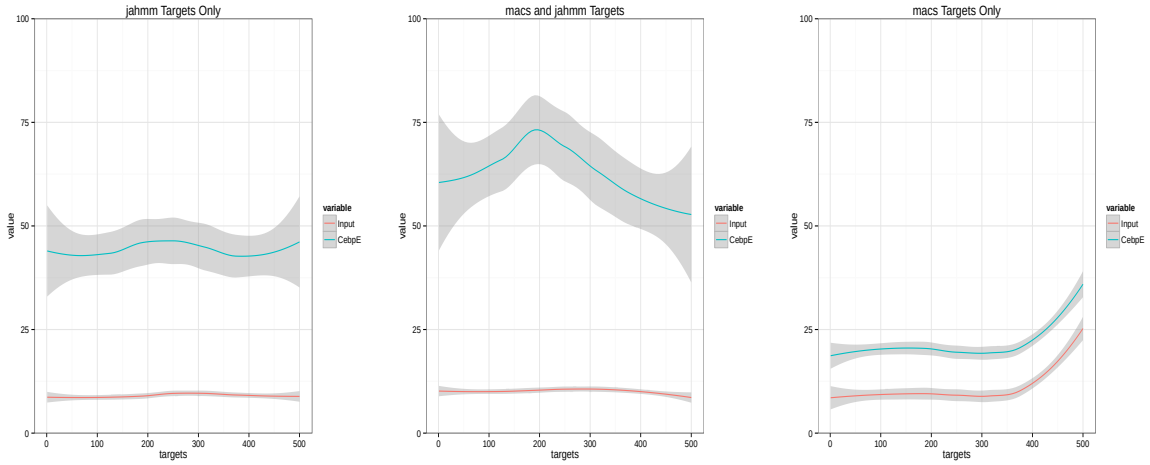


Figure 3: Comparison of Targets Identified by macs and jahmm

3 Results: jaHMM

Results from this PeakAnalysis pipeline on CebpE dataset using jahmm:

- 3 states were sufficient to discretize the ChIPseq peaks (300bp or 3000bp resolution) into targets and non-targets.
- In 300bp and 3000bp peak resolution, 64154and 16490 targets were identified, respectively.
- In comparison with macs peak caller, jahmm(64154) could identify more targets than macs(51366).
- Peaks identified by jahmm characterized by higher enrichment of target reads in comparison to the input.

```
  Filename: chromatinConformation.Rnw
  Working directory: /home/ricky/Rlim/ChromatinConformation/PeakCalls/CebpE/PeakAnalysis
```

4 Metainfo

```
sessionInfo()

## R version 3.2.0 (2015-04-16)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 14.04.1 LTS
##
## locale:
##  [1] LC_CTYPE=en_SG.UTF-8          LC_NUMERIC=C
##  [3] LC_TIME=en_SG.UTF-8           LC_COLLATE=en_SG.UTF-8
##  [5] LC_MONETARY=en_SG.UTF-8       LC_MESSAGES=en_SG.UTF-8
##  [7] LC_PAPER=en_SG.UTF-8          LC_NAME=en_SG.UTF-8
##  [9] LC_ADDRESS=en_SG.UTF-8        LC_TELEPHONE=en_SG.UTF-8
## [11] LC_MEASUREMENT=en_SG.UTF-8    LC_IDENTIFICATION=en_SG.UTF-8
##
## attached base packages:
##  [1] stats4    parallel  grid      stats     graphics  grDevices utils
##  [8] datasets  methods   base
##
## other attached packages:
##  [1] mgcv_1.8-6              nlme_3.1-120
##  [3] GenomicAlignments_1.4.1 Rsamtools_1.20.2
##  [5] Biostrings_2.36.1       XVector_0.8.0
##  [7] ggbio_1.16.0            ggplot2_1.0.1
##  [9] venneuler_1.1-0         rJava_0.9-6
## [11] VennDiagram_1.6.9       reshape2_1.4.1
## [13] data.table_1.9.4        GenomicRanges_1.20.3
## [15] GenomeInfoDb_1.4.0      IRanges_2.2.1
## [17] S4Vectors_0.6.0         BiocGenerics_0.14.0
## [19] gridExtra_0.9.1         xtable_1.7-4
## [21] Rjahmm_0.1-1            knitr_1.10.5
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.11.6              biovizBase_1.16.0
##  [3] lattice_0.20-31          digest_0.6.8
##  [5] plyr_1.8.2               chron_2.3-45
##  [7] futile.options_1.0.0     acepack_1.3-3.3
##  [9] RSQLite_1.0.0            evaluate_0.7
## [11] highr_0.5                zlibbioc_1.14.0
## [13] GenomicFeatures_1.20.1   Matrix_1.2-0
## [15] rpart_4.1-9              labeling_0.3
## [17] proto_0.3-10             splines_3.2.0
## [19] BiocParallel_1.2.2       stringr_1.0.0
## [21] foreign_0.8-63           RCurl_1.95-4.6
## [23] biomaRt_2.24.0           munsell_0.4.2
## [25] rtracklayer_1.28.3       nnet_7.3-9
## [27] codetools_0.2-11         Hmisc_3.16-0
## [29] XML_3.98-1.1             reshape_0.8.5
## [31] MASS_7.3-40              bitops_1.0-6
```

```
## [33] RBGL_1.44.0              GGally_0.5.0
## [35] gtable_0.1.2             DBI_0.3.1
## [37] magrittr_1.5             formatR_1.2
## [39] scales_0.2.4             graph_1.46.0
## [41] stringi_0.4-1            latticeExtra_0.6-26
## [43] futile.logger_1.4.1      Formula_1.2-1
## [45] lambda.r_1.1.7           RColorBrewer_1.1-2
## [47] tools_3.2.0              dichromat_2.0-0
## [49] OrganismDbi_1.10.0       BSgenome_1.36.0
## [51] Biobase_2.28.0           survival_2.38-1
## [53] AnnotationDbi_1.30.1     colorspace_1.2-6
## [55] cluster_2.0.1            VariantAnnotation_1.14.1
```