ChIPSeqSpike: ChIP-seq data scaling with spike-in control

Nicolas Descostes[*1]

[1]Howard Hughes Medical Institute - New York University

[*]nicolas.descostes@gmail.com

19 May 2021

Abstract

Vignette update - 2018-01-26

Package

ChIPSeqSpike 1.12.0

Contents

1 Introduction

Chromatin Immuno-Precipitation followed by Sequencing (ChIP-Seq) is used to determine the binding sites of any protein of interest, such as transcription factors or histones with or without a specific modification, at a genome scale (Barski et al. 2007; Park 2009). ChIP-Seq entails treating cells with a cross-linking reagent such as formaldehyde; isolating the chromatin and fragmenting it by sonication; immuno-precipitating with antibodies directed against the protein of interest; reversing crosslink; DNA purification and amplification before submission to sequencing. These many steps can introduce biases that make ChIP-Seq more qualitative than quantitative. Different efficiencies in nuclear extraction, DNA sonication, DNA amplification or antibody recognition make it challenging to distinguish between true differential binding events and technical variability.

This problem was addressed by using an external spike-in control to keep track of technical biases between conditions (Orlando et al. 2014; Bonhoure et al. 2014). Exogenous DNA from a different non-closely related species is inserted during the protocol to infer scaling factors. This modification was shown to be especially important for revealing global histone modification differences, that are not caught by traditional downstream data normalization techniques, such as Histone H3 lysine-27 trimethyl (H3K27me3) upon Ezh2 inhibitor treatment (Egan et al. 2016).

ChIPSeqSpike provides tools for ChIP-Seq spike-in normalization, assessment and analysis. Conversely to a majority of ChIP-Seq related tools, ChIPSeqSpike does not rely on peak detection. However, if one wants to focus on peaks, ChIPSeqSpike is flexible enough to do so. ChIPSeqSpike provides ready to use scaled bigwig files as output and scaling factors values. We believe that ChIPSeqSpike will be of great value to the genomics community.

2 Standard workflow

2.1 Note for Windows users

On Windows operating system, reading of BigWig files fail due to the Bioconductor package rtracklayer >= 1.37.6 that does not support this file format. Therefore, the boost mode, the input subtraction method, and scaling method do not work on this operating system.

2.2 Quick start

A case study reported Chromatin Immuno-Precipitation followed by Sequencing (ChIP-Seq) experiments that did not show differences upon inhibitor treatment with traditional normalization procedures (Orlando et al. 2014). These experiments looked at the effect of DOT1L inhibitor EPZ5676 (Daigle et al. 2013) treatment on the Histone H3 lysine-79 dimethyl (H3K79me2) modification in Jurkat cells. DOT1L is involved in the RNA Polymerase II pause release and licensing of transcriptional elongation. H3K79me2 ChIP-Seq were performed on cells treated with 0%, 50% and 100% EPZ5676 inhibitor (see next section for details on data).

The following code performs spike-in normalization with a wrapper function on data sub-samples. A 'test_chipseq' temporary results folder is also created.

```
library("ChIPSeqSpike")

## Preparing testing data
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file("extdata/bam_files", package="ChIPSeqSpike")
bigwig_path <- system.file("extdata/bigwig_files", package="ChIPSeqSpike")
gff_vec <- system.file("extdata/test_coord.gff", package="ChIPSeqSpike")
genome_name <- "hg19"
output_folder <- "test_chipseqspike"
bigwig_files <- system.file("extdata/bigwig_files",
                            c("H3K79me2_0-filtered.bw",
                              "H3K79me2_100-filtered.bw",
                              "H3K79me2_50-filtered.bw",
                              "input_0-filtered.bw",
                              "input_100-filtered.bw",
                              "input_50-filtered.bw"), package="ChIPSeqSpike")

## Copying example files
dir.create("./test_chipseqspike")
mock <- file.copy(bigwig_files, "test_chipseqspike")

## Performing spike-in normalization
if (.Platform$OS.type != "windows") {
    csds_test <- spikePipe(info_file_csv, bam_path, bigwig_path, gff_vec,
                           genome_name, outputFolder = output_folder)
}
```

2.3 Data

The data used in this documentation represent a gold-standard example of the importance of using spike-in controls with ChIP-Seq experiments. It uses chromatin from Drosophila Melanogaster as exogenous spike-in control to correct experimental biases. Without a spike-in control and using only RPM normalization, proper differences in the H3K79me2 histone modification in human Jurkat cells upon EPZ5676 inhibitor treatment were not observed (Orlando et al. 2014).

This dataset is made of bigwig and bam files of H3K79me2 ChIP-Seq data and corresponding input DNA controls (see input subtraction section). Bam files contain data aligned to the Human reference genome Hg19 or to the Drosophila reference genome dm3. The latest is used to compute external spike-in scaling factors. All above mentioned data are available at 0%, 50% and 100% EPZ5676 inhibitor treatment.

2.3.1 Testing data

For the sake of memory and computation time efficiency, bigwig files used in this vignette are limited to chromosome 1. Reads falling in the top 10% mostly bound genes (at 0% treatment) with length between 700-800 bp were kept in bam files. For efficient plotting functions testing, only binding values of the top 100 mostly bound genes are used and can be accessed with `data(result_extractBinding)`. Scores for factors and read counts were computed on the whole dataset and are available through `data(result_estimateScalingFactors)` (see below).

2.3.2 Complete data

The whole dataset is accessible at GSE60104. Specifically, the data used are H3K79me2 0% (GSM1465004), H3K79me2 50% (GSM1465006), H3K79me2 100% (GSM1465008), input DNA 0% (GSM1511465), input DNA 50% (GSM1511467) and input DNA 100% (GSM1511469).

The data were treated as follows: Quality of sequencing was assessed with FastQC v0.11.4. Reads having less than 80% of quality scores above 25 were removed with NGSQCToolkit v2.3.3 (Patel and Jain 2012). Homo Sapiens Hg19 and Drosophila Melanogaster dm3 from Illumina igenomes UCSC Collection were used. ChIP-Seq data were aligned with Bowtie2 v2.1.0 (Langmead and Salzberg 2012) with default parameters. Sam outputs were converted to Bam with Samtools v1.0.6 (Li et al. 2009) and sorted with Picard tools v1.88. Data were further processed with Pasha v0.99.21 (Fenouil et al. 2009). Fixed steps wiggle files were converted to bigwigs with wigToBigWig.

Results with the complete dataset are also provided in this documentation.

2.4 Detailed operations

The spike-in normalization procedure consists of 4 steps: RPM scaling, input DNA subtraction, RPM scaling reversal and exogenous spike-in DNA scaling. Below is detailed the different steps included in the above mentioned wrapper function 'spikePipe'.

2.4.1 Dataset generation

The different data necessary for proper spike-in scaling are provided in a csv or a tab separated txt file. The columns must contain proper names and are organized as follows: Experiment name (expName); bam file name of data aligned to the endogenous reference genome (endogenousBam); bam file name of data aligned to the exogenous reference genome (exogenousBam); the corresponding input DNA bam file aligned to the endogenous reference genome (inputBam); the fixed steps bigwig file name of data aligned to the endogenous reference genome (bigWigEndogenous) and the fixed steps bigwig file names of the corresponding input DNA experiment aligned to the endogenous reference genome (bigWigInput).

```
info_file <- read.csv(info_file_csv)
head(info_file)
##        expName                  endogenousBam                  exogenousBam
## 1   H3K79me2_0   H3K79me2_0_hg19-filtered.bam   H3K79me2_0_dm3-filtered.bam
## 2  H3K79me2_50  H3K79me2_50_hg19-filtered.bam  H3K79me2_50_dm3-filtered.bam
## 3 H3K79me2_100 H3K79me2_100_hg19-filtered.bam H3K79me2_100_dm3-filtered.bam
##                      inputBam         bigWigEndogenous           bigWigInput
## 1   input_0_hg19-filtered.bam   H3K79me2_0-filtered.bw   input_0-filtered.bw
## 2  input_50_hg19-filtered.bam  H3K79me2_50-filtered.bw  input_50-filtered.bw
## 3 input_100_hg19-filtered.bam H3K79me2_100-filtered.bw input_100-filtered.bw
```

From the info file, two kinds of objects can be generated: either a ChIPSeqSpikeDataset or a ChIPSeqSpikeDatasetList depending upon the number of input DNA experiments. A ChIPSeqSpikeDatasetList object is a list of ChIPSeqSpikeDataset object that is created if several input DNA experiments are used. In this latter case, ChIP-Seq experiments are

grouped by their corresponding input DNA. The function spikeDataset creates automatically the suitable object. The folder path to the bam and fixed steps bigwig files must be provided.

```
csds_test <- spikeDataset(info_file_csv, bam_path, bigwig_path)
is(csds_test)
## [1] "ChIPSeqSpikeDatasetList"
```

2.4.1.1 Boost mode

Reading and processing bigwig and bam files can be memory-greedy. By default, files are read, processed and written at each step of the normalization procedure to enable data treatment on a regular desktop computer. One could wish to reduce the time of computation especially when processing a lot of data. ChIPSeqSpike reduces such time by providing a boost mode.

```
if (.Platform$OS.type != "windows") {
    csds_testBoost <- spikeDataset(info_file_csv, bam_path, bigwig_path,
    boost = TRUE)
    is(csds_testBoost)
}
```

Binding scores for each experiment are stored in a GRanges object and are directly accessible by functions.

```
if (.Platform$OS.type != "windows") {
    getLoadedData(csds_testBoost[[1]])
}
```

Even if optimizing greatly the time of computing, one should know that loading binding scores of all experiments is greedy in memory and should be used with caution. The boost mode is ignored in the rest of the vignette, but all code provided in the following sections, with the exception of section 3.1 (plotTransform), can be run with csds_testBoost.

2.4.2 Summary and control

A ChIPSeqSpikeDataset object, at this point, is made of slots storing paths to files. In order to compute scaling factors, bam counts are first computed. A scaling factor is defined as 1000000/bam_count. The method estimateScalingFactors returns bam counts and endogenous/exogenous scaling factors for all experiments. In the following example, scores are computed using chromosome 1 only. Scores on the whole dataset are also indicated below.

```
csds_test <- estimateScalingFactors(csds_test, verbose = FALSE)
```

The different scores can be visualized:

```
## Scores on testing sub-samples
spikeSummary(csds_test)
##              endoScalFact exoScalFact endoCount exoCount
## H3K79me2_0       2439.024   1267.4271       410      789
## input            1602.564          NA       624       NA
```

```
## H3K79me2_50      2762.431    926.7841       362     1079
## input            3300.330          NA       303       NA
## H3K79me2_100     6578.947    443.2624       152     2256
## input            2747.253          NA       364       NA

##Scores on whole dataset
data(result_estimateScalingFactors)
spikeSummary(csds)
##              endoScalFact exoScalFact endoCount exoCount
## H3K79me2_0     0.04046008  0.15618313  24715719  6402740
## input          0.06852631          NA  14592936       NA
## H3K79me2_50    0.04779062  0.12081979  20924609  8276790
## input          0.16936140          NA   5904533       NA
## H3K79me2_100   0.10244954  0.05738227   9760903 17426985
## input          0.14037080          NA   7123989       NA
```

An important parameter to keep in mind when performing spike-in with ChIP-seq is the percentage of exogenous DNA relative to that of endogenous DNA. The amount of exogenous DNA should be between 2-25% of endogenous DNA. The method getRatio returns the percentage of exogenous DNA and throws a warning if this percentage is not within the 2-25% range. In theory, having more than 25% exogenous DNA should not affect the normalization, whereas having less than 2% is usually not sufficient to perform a reliable normalization.

```
getRatio(csds_test)
## Warning in (function (ratio, expname) : H3K79me2_0 contains more than 25% of
## endogenous DNA.
## Warning in (function (ratio, expname) : H3K79me2_50 contains more than 25% of
## endogenous DNA.
## Warning in (function (ratio, expname) : H3K79me2_100 contains more than 25% of
## endogenous DNA.
##              Percentage Exo
## H3K79me2_0            192.4
## H3K79me2_50           298.1
## H3K79me2_100         1484.2

## Result on the whole dataset
data(ratio)
ratio
##              Percentage Exo
## H3K79me2_0             20.0
## H3K79me2_50            25.6
## H3K79me2_100           54.6
```

2.4.3 RPM scaling

The first normalization applied to the data is the 'Reads Per Million' (RPM) mapped reads. The method 'scaling' is used to achieve such normalization using default parameters. It is also used to reverse the RPM normalization and apply exogenous scaling factors (see sections

2.3.5 and 2.3.6).

```
if (.Platform$OS.type != "windows") {
    csds_test <- scaling(csds_test, outputFolder = output_folder)
}
```

In the context of this vignette, output_folder is precised since the testing files were copied to a temporary 'test_chipseqspike' folder. The slots containing paths to files will be updated to this folder. If not precised, the RPM scaled bigwig files are written to the same folder containing bigwigs. This statement is also applicable for the next operations below.

2.4.4 Input Subtraction

When Immuno-Precipitating (IP) DNA bound by a given protein, a control is needed to distinguish background noise from true signal. This is typically achieved by performing a mock IP, omitting the use of antibody. After mock IP sequencing, one can notice peaks of signal above background. These peaks have to be removed from the experiment since they represent false positives. The inputSubtraction method simply subtracts scores of the input DNA experiment from the corresponding ones. If in boost mode, the input subtracted values are stored in the dataset object and no files are written. For this latter case, the method exportBigWigs can be used to output the transformed files.

```
if (.Platform$OS.type != "windows") {
    csds_test <- inputSubtraction(csds_test)
}
```

2.4.5 RPM scaling reversal

After RPM and input subtraction normalization, the RPM normalization is reversed in order for the data to be normalized by the exogenous scaling factors.

```
if (.Platform$OS.type != "windows") {
    csds_test <- scaling(csds_test, reverse = TRUE)
}
```

2.4.6 Exogenous scaling

Finally, exogenous scaling factors are applied to the data.

```
if (.Platform$OS.type != "windows") {
    csds_test <- scaling(csds_test, type = "exo")
}
```

2.4.7 Extract binding values

The last step of data processing is to extract and format binding scores in order to use plotting methods. The 'extractBinding' method extracts binding scores at different locations and stores these values in the form of PlotSetArray objects and matrices (see ?extractBinding for more details). The scores are retrieved on annotations provided in a gff file. If one wishes to focus on peaks, their coordinates should be submitted at this step. The genome name must also be provided. For details about installing the required BSgenome package corresponding to the endogenous organism, see the BSgenome package documentation.

```
if (.Platform$OS.type != "windows") {
    csds_test <- extractBinding(csds_test, gff_vec, genome_name)
}
```

3 Plotting data

ChIPSeqSpike offers several graphical methods for normalization diagnosis and data exploration. These choices enable one to visualize each step of the normalization through exploring inter-samples differences using profiles, heatmaps, boxplots and correlation plots.

In the following sections, the testing data are restricted to the 100 mostly bound genes. Results on the complete set of hg19 genes are also indicated.

3.1 Meta-profiles and transformations

The first step of spike-in normalized ChIP-Seq data analysis is an inter-sample comparison by meta-gene or meta-annotation profiling. The method 'plotProfile' automatically plots all experiments at the start, midpoint, end and composite locations of the annotations provided to the method extractBinding in gff format. Here is the result of profiling H3K79me2 on the 100 mostly bound genes at 0% inhibitor treatment (figure 1).

```
data(result_extractBinding)
plotProfile(csds, legend = TRUE)
```

The unspiked data (however RPM scaled and input subtracted) can be added to the plot (figure 2).

```
plotProfile(csds, legend = TRUE, notScaled = TRUE)
```

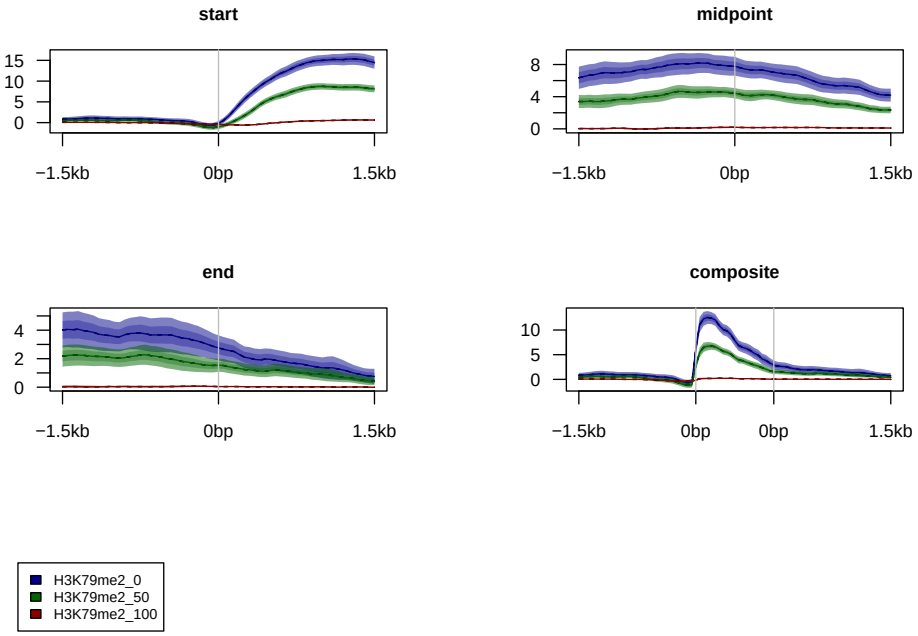

Figure 1: Spiked experiment upon different percentages concentrations of inhibitor treatment

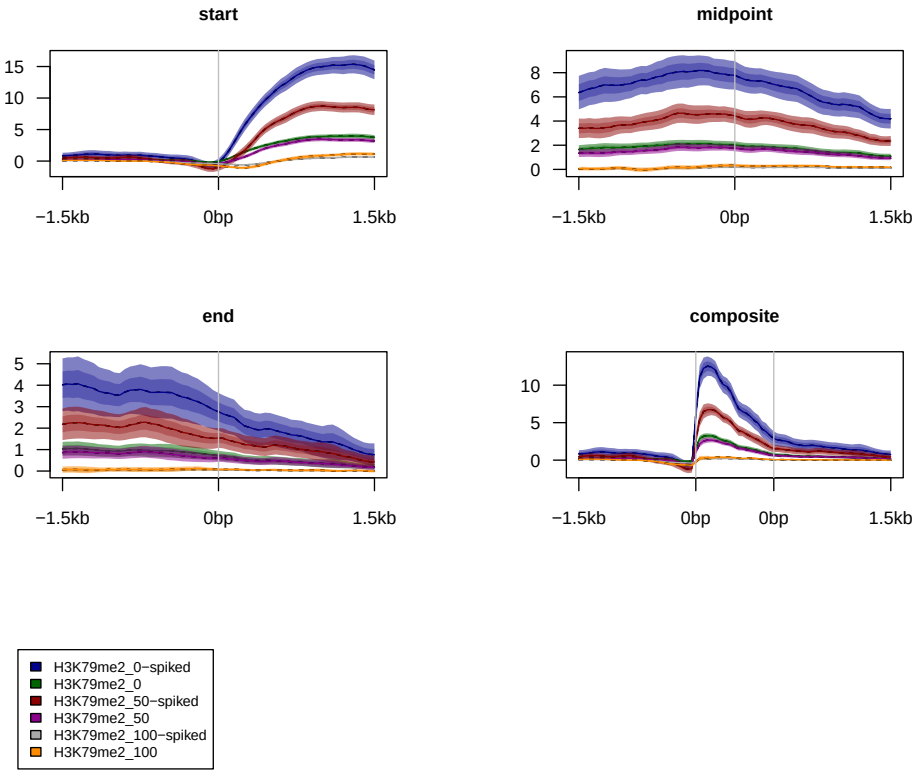

Figure 2: Same as figure 1 including unspiked data

The effect of the individual processing steps for each experiment can also be plotted.

```
plotTransform(csds, legend = TRUE, separateWindows = TRUE)
```

3.2 Heatmaps

plotHeatmaps is a versatile method based on the plotHeatmap method of the seqplots package (Stempor and Ahringer 2016). This method enables one to represent data at different locations (start, end, midpoint, composite) and at different stages of the normalization process. Different scaling (log, zscore, etc) and different clustering approaches (k-means, hierarchical, etc) can be used (see documentation for more details).

Figure 3 shows a k-means clustering of spiked data, each group being sub-sorted by decreasing values.

```
plotHeatmaps(csds, nb_of_groups = 2, clustering_method = "kmeans")
```

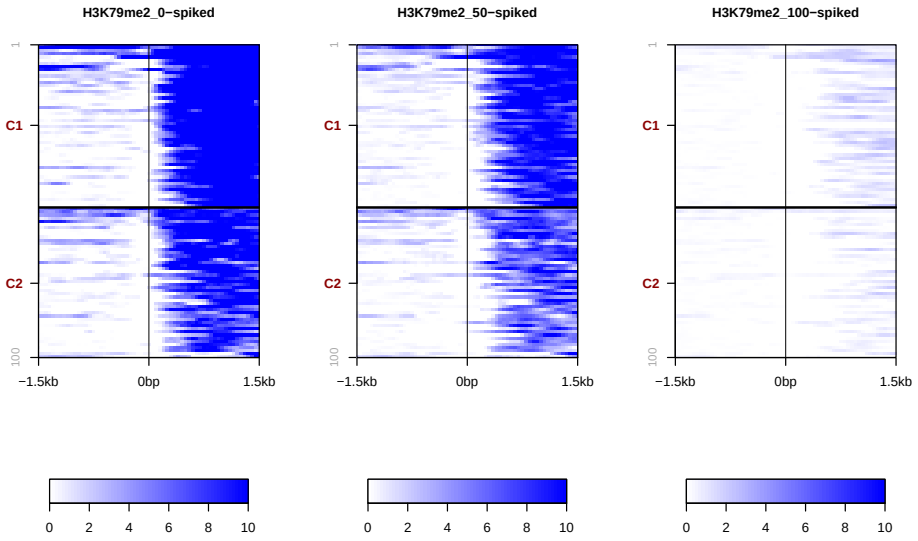

Figure 3: kmeans clustering of spiked data

Figure 4 illustrates a clustering by decreasing values on the whole dataset.

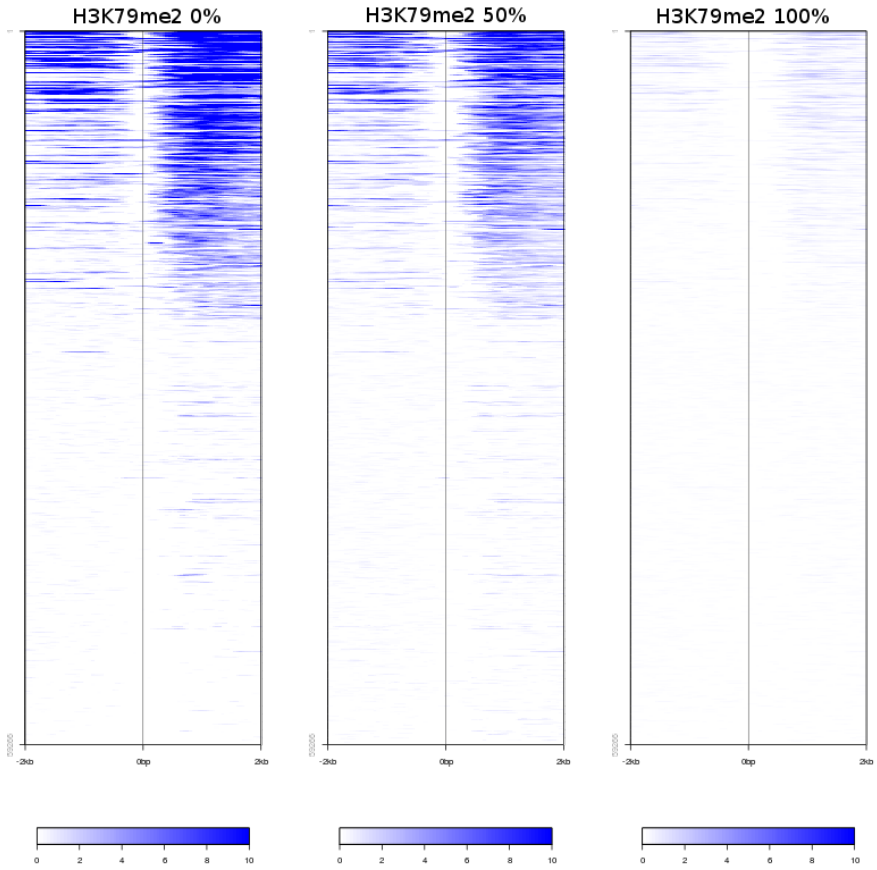

Figure 4: Spiked data organized by decreasing values at start position of all refseq Hg19 genes

3.3 Boxplots

boxplotSpike plots boxplots of the mean values of ChIP-seq experiments on the annotations given to the extractBinding method. It offers a wide range of graphical representations that includes violin plots (see documentation for details). Figure 5 illustrates all transformations of all dataset indicating confidence intervals.

```
par(cex.axis=0.5)
boxplotSpike(csds, rawFile = TRUE, rpmFile = TRUE, bgsubFile = TRUE,
revFile = TRUE, spiked = TRUE, outline = FALSE)
```

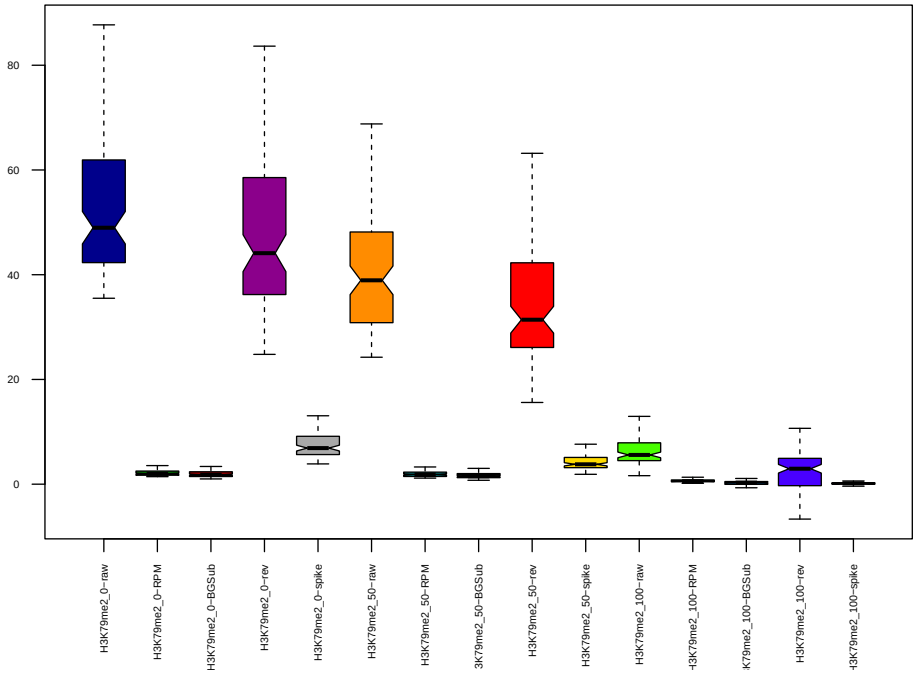

Figure 5: Complete representation of the whole procedure using boxplots (without outliers)

Figure 6 shows spiked experiments indicating each mean value, mean and standard deviation with a violin plot representation.

```
boxplotSpike(csds, outline = FALSE, violin=TRUE, mean_with_sd = TRUE,
 jitter = TRUE)
```

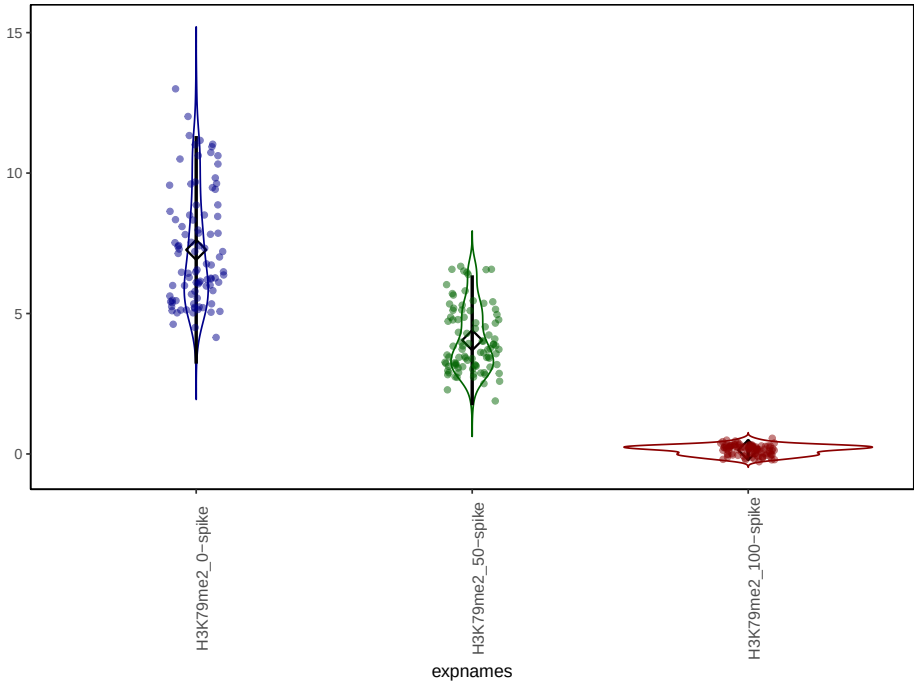

Figure 6: Spiked data with mean and standard deviation - Each point represents a mean binding value on a given gene

3.4 Correlation plots

The plotCor method plots the correlation between ChIP-seq experiments using heatscatter plot or, if heatscatterplot = FALSE, correlation tables. For heatscatter plots, ChIPSeqSpike makes use of the heatscatter function of the package LSD and the corrplot function of the package corrplot is used to generate correlation tables. This offers a wide range of graphical possibilities for assessing the correlation between experiments and transformation steps (see documentation for more details).

Figure 7 shows two correlation table representations between spiked experiments.

```
par(mfrow=c(1,2))
plotCor(csds, heatscatterplot = FALSE)
plotCor(csds, heatscatterplot = FALSE, method_corrplot = "number")
```

Figure 8 illustrates a heatscatter plot of spiked data after log transformation (only positive mean binding values are kept) and figure 9 is the result of running the same code on the whole refseq Hg19 gene set.

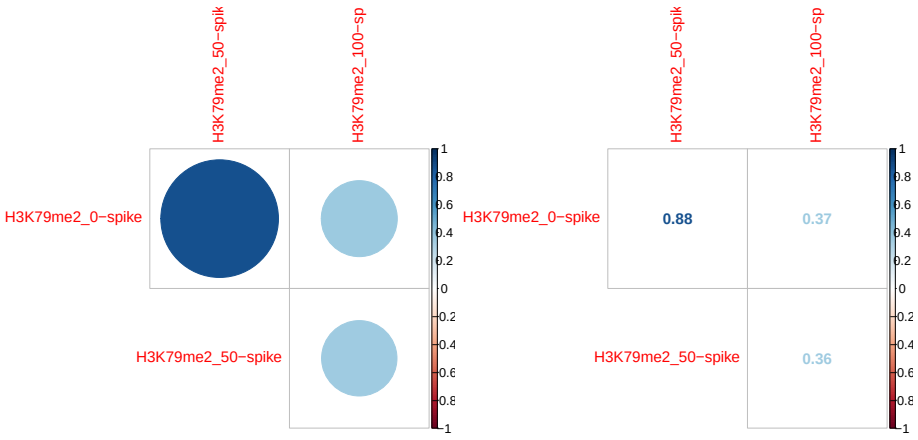

Figure 7: Correlation table of spiked data with circle (left) or numbers (right)

```
plotCor(csds, method_scale = "log")
```

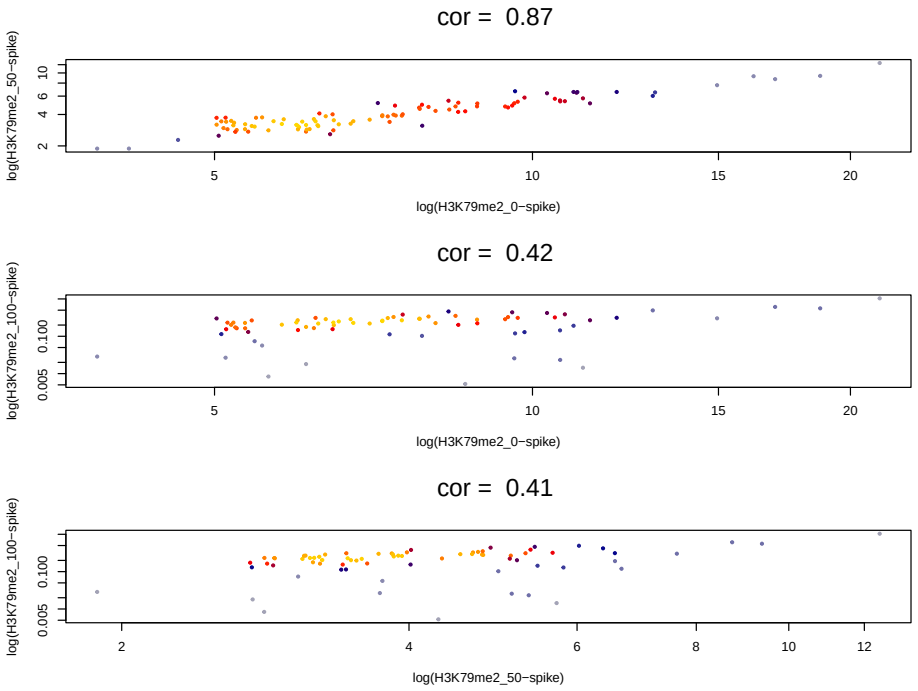

Figure 8: Heatscatter of spiked data after log transformation

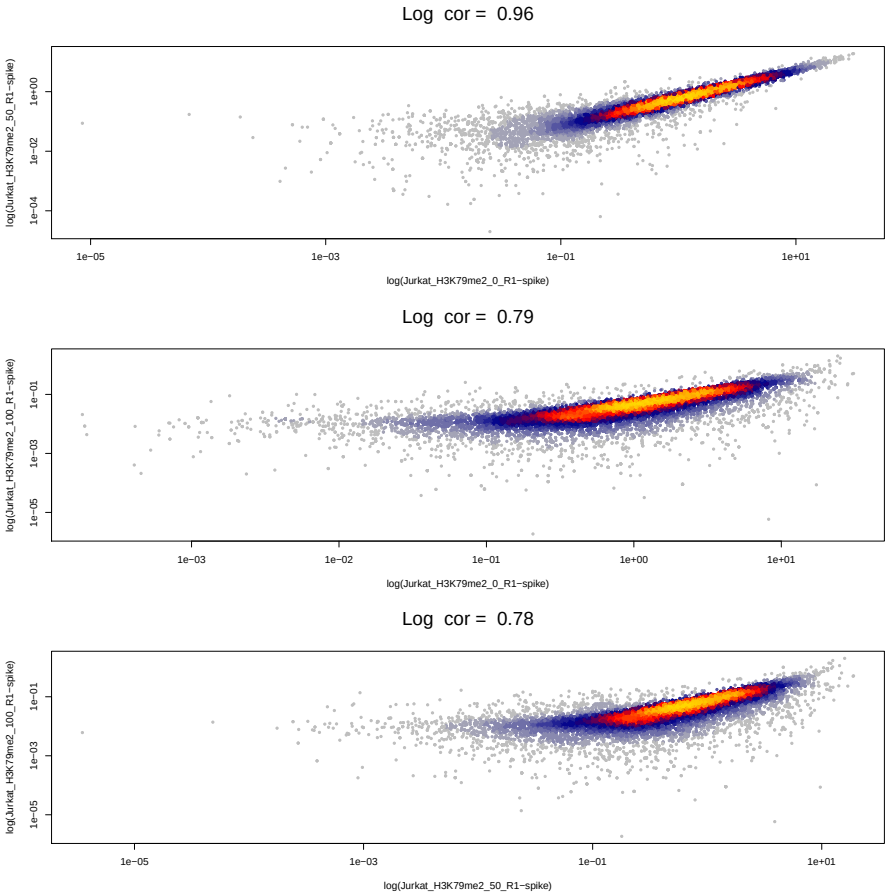

Figure 9: Heatscatter of spiked data after log transformation on all Hg19 refseq genes

4 Session info

```
sessionInfo(package="ChIPSeqSpike")
## R version 4.1.0 RC (2021-05-10 r80283)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows Server x64 (build 17763)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=C
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## character(0)
##
```

```
## other attached packages:
## [1] ChIPSeqSpike_1.12.0
##
## loaded via a namespace (and not attached):
##   [1] colorspace_2.0-1            rjson_0.2.20
##   [3] ellipsis_0.3.2              class_7.3-19
##   [5] htmlTable_2.2.1             XVector_0.32.0
##   [7] GenomicRanges_1.44.0        base64enc_0.1-3
##   [9] rstudioapi_0.13             farver_2.1.0
##  [11] stats_4.1.0                 DT_0.18
##  [13] bit64_4.0.5                 fansi_0.4.2
##  [15] splines_4.1.0               cachem_1.0.5
##  [17] knitr_1.33                  spam_2.6-0
##  [19] Formula_1.2-4               jsonlite_1.7.2
##  [21] Rsamtools_2.8.0             base_4.1.0
##  [23] cluster_2.1.2               png_0.1-7
##  [25] shiny_1.6.0                 BiocManager_1.30.15
##  [27] compiler_4.1.0              backports_1.2.1
##  [29] assertthat_0.2.1            Matrix_1.3-3
##  [31] fastmap_1.1.0               later_1.2.0
##  [33] htmltools_0.5.1.1           tools_4.1.0
##  [35] dotCall64_1.0-1             gtable_0.3.0
##  [37] glue_1.4.2                  GenomeInfoDbData_1.2.6
##  [39] reshape2_1.4.4              dplyr_1.0.6
##  [41] maps_3.3.0                  grDevices_4.1.0
##  [43] Rcpp_1.0.6                  Biobase_2.52.0
##  [45] vctrs_0.3.8                 Biostrings_2.60.0
##  [47] rtracklayer_1.52.0          xfun_0.23
##  [49] stringr_1.4.0               mime_0.10
##  [51] lifecycle_1.0.0             restfulr_0.0.13
##  [53] XML_3.99-0.6                zlibbioc_1.38.0
##  [55] scales_1.1.1                BiocStyle_2.20.0
##  [57] BSgenome_1.60.0             kohonen_3.0.10
##  [59] graphics_4.1.0              promises_1.2.0.1
##  [61] MatrixGenerics_1.4.0        parallel_4.1.0
##  [63] SummarizedExperiment_1.22.0 RColorBrewer_1.1-2
##  [65] fields_12.3                 utils_4.1.0
##  [67] yaml_2.2.1                  memoise_2.0.0
##  [69] gridExtra_2.3               ggplot2_3.3.3
##  [71] datasets_4.1.0              rpart_4.1-15
##  [73] latticeExtra_0.6-29         stringi_1.6.2
##  [75] RSQLite_2.2.7               S4Vectors_0.30.0
##  [77] BiocIO_1.2.0                corrplot_0.88
##  [79] plotrix_3.8-1               checkmate_2.0.0
##  [81] BiocGenerics_0.38.0         BiocParallel_1.26.0
##  [83] GenomeInfoDb_1.28.0         rlang_0.4.11
##  [85] pkgconfig_2.0.3             matrixStats_0.58.0
##  [87] bitops_1.0-7                evaluate_0.14
##  [89] lattice_0.20-44             purrr_0.3.4
##  [91] labeling_0.4.2              GenomicAlignments_1.28.0
##  [93] htmlwidgets_1.5.3           LSD_4.1-0
```

```
##  [95] bit_4.0.4                 tidyselect_1.1.1
##  [97] plyr_1.8.6                magrittr_2.0.1
##  [99] bookdown_0.22             R6_2.5.0
## [101] IRanges_2.26.0            generics_0.1.0
## [103] Hmisc_4.5-0               DelayedArray_0.18.0
## [105] DBI_1.1.1                 withr_2.4.2
## [107] pillar_1.6.1              foreign_0.8-81
## [109] survival_3.2-11           RCurl_1.98-1.3
## [111] nnet_7.3-16               tibble_3.1.2
## [113] crayon_1.4.1              seqplots_1.30.0
## [115] utf8_1.2.1                rmarkdown_2.8
## [117] viridis_0.6.1             jpeg_0.1-8.1
## [119] grid_4.1.0                data.table_1.14.0
## [121] blob_1.2.1                methods_4.1.0
## [123] digest_0.6.27             xtable_1.8-4
## [125] httpuv_1.6.1              stats4_4.1.0
## [127] munsell_0.5.0             viridisLite_0.4.0
```

References

Barski, Artem, Suresh Cuddapah, Kairong Cui, Tae Young Roh, Dustin E Schones, Zhibin Wang, Gang Wei, Iouri Chepelev, and Keji Zhao. 2007. "High-Resolution Profiling of Histone Methylations in the Human Genome." *Cell* 129 (4): 823–37. https://doi.org/10.1016/j.cell.2007.05.009.

Bonhoure, Nicolas, Gergana Bounova, David Bernasconi, Viviane Praz, Fabienne Lammers, Donatella Canella, Ian M Willis, et al. 2014. "Quantifying ChIP-Seq Data: A Spiking Method Providing an Internal Reference for Sample-to-Sample Normalization." *Genome Research* 24 (April): 1157–68. https://doi.org/10.1101/gr.168260.113.

Daigle, SR, EJ Olhava, CA Therkelsen, A Basavapathruni, L Jin, PA Boriack-Sjodin, CJ Allain, et al. 2013. "Potent Inhibition of DOT1L as Treatment of MLL-Fusion Leukemia." *Blood* 122 (6): 1017–25. https://doi.org/10.1182/blood-2013-04-497644.

Egan, Brian, Chih Chi Yuan, Madeleine Lisa Craske, Paul Labhart, Gulfem D Guler, David Arnott, Tobias M Maile, et al. 2016. "An Alternative Approach to ChIP-Seq Normalization Enables Detection of Genome-Wide Changes in Histone H3 Lysine 27 Trimethylation Upon EZH2 Inhibition." *PLoS ONE* 11 (11). https://doi.org/10.1371/journal.pone.0166438.

Fenouil, Romain, Nicolas Descostes, Lionel Spinelli, Frederic Koch, Muhammad A Maqbool, Touati Benoukraf, Pierre Cauchy, Charlene Innocenti, Pierre Ferrier, and Jean-Christophe Andrau. 2009. "Pasha a Versatile R Package for Piling Chromatin HTS Data." *Bioinformatics* 25 (16). https://doi.org/10.1093/bioinformatics/btp352.

Langmead, Ben, and Steven L Salzberg. 2012. "Fast Gapped-Read Alignment with Bowtie 2." *Nature Method* 9 (4). https://doi.org/10.1038/nmeth.1923.

Li, Heng, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. 2009. "The Sequence Alignment/Map Format and SAMtools." *Bioinformatics* 25 (16). https://doi.org/10.1093/bioinformatics/btp352.

Orlando, David A, Mei Wei Chen, Victoria E Brown, Snehakumari Solanki, Yoon J Choi, Eric R Olson, Christian C Fritz, James E Bradner, and Matthew G Guenther. 2014. "Quantitative ChIP-Seq Normalization Reveals Global Modulation of the Epigenome." *Cell Reports* 9 (3): 1163–70. https://doi.org/10.1016/j.celrep.2014.10.018.

Park, PJ. 2009. "ChIP–Seq: Advantages and Challenges of a Maturing Technology." *Nature Reviews Genetics* 10 (October): 669–80. https://doi.org/10.1038/nrg2641.

Patel, Ravi K, and Mukesh Jain. 2012. "NGS QC Toolkit: A Toolkit for Quality Control of Next Generation Sequencing Data." *PLoS ONE* 7 (2). https://doi.org/10.1371/journal.pone.0030619.

Stempor, Przemyslaw, and Julie Ahringer. 2016. "SeqPlots - Interactive Software for Exploratory Data Analyses, Pattern Discovery and Visualization in Genomics." *Wellcome Open Research* 1 (November). https://doi.org/10.12688/wellcomeopenres.10004.1.