Package 'CoverageView'

November 11, 2026

Type Package

Title Coverage visualization package for R

Version 1.50.0

Date 2017-06-08

Author Ernesto Lowy

Maintainer Ernesto Lowy <ernestolowy@gmail.com>

Description This package provides a framework for the visualization of genome coverage profiles. It can be used for ChIP-seq experiments, but it can be also used for genome-wide nucleosome positioning experiments or other experiment types where it is important to have a framework in order to inspect how the coverage distributed across the genome

biocViews ImmunoOncology, Visualization,RNASeq,ChIPSeq,Sequencing,Technology,Software

License Artistic-2.0

Depends R (>= 2.10), methods, Rsamtools (>= 1.19.17), rtracklayer

Imports S4Vectors (>= 0.7.21), IRanges(>= 2.3.23), GenomicRanges, GenomicAlignments, parallel, tools

NeedsCompilation no

git_url https://git.bioconductor.org/packages/CoverageView

git_branch RELEASE_3_23

git_last_commit e9dfb79

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

CoverageView-package 2
cov.interval 3

cov.matrix 5
CoverageBamFile-class 9
CoverageBigWigFile-class 10
CoverageView-internal 11
DF_H3K36me3 11
DF_H3K36me3_control 12
DF_H3K4me3 13
DF_H3K4me3_ctl 14
DF_H3K4me3_nopeaks_ratios 15
draw.heatmap 16
draw.profile 17
export.wig 18
FoxA1_chr19 19
genome.covplot.cumdepth 20
genome.covplot.depth 21
write.profile 23

Index **25**

`CoverageView-package` *Coverage visualization package for R*

Description

Package for the generation of coverage profiles using data originated in different types of NGS experiments

Details

This package provides a framework for the visualization of genome coverage profiles.

It can be used for ChIP-seq experiments, but it can be also used for nucleosome or RNA-seq experiments or whatever type of experiment where one can produce an aligment in the BAM or WIG/BigWIG formats

Author(s)

Ernesto Lowy

Maintainer: Ernesto Lowy <ernestolowy@gmail.com>

References

BAM format specification: http://samtools.sourceforge.net/SAMv1.pdf

WIG format specification: https://genome.ucsc.edu/FAQ/FAQformat.html

cov.interval *arithmetic operation on an interval*

Description

This method computes a numeric vector with the result of a certain arithmetic operation with the coverage for two particular BAM or BigWIG files within a specific genomic interval defined by the user

Usage

```
  ## S4 method for signature 'CoverageBamFile,CoverageBamFile'
cov.interval(tr,ctl,normalization,chr,start,end,bin_width,do)
  ## S4 method for signature 'CoverageBigWigFile,CoverageBigWigFile'
cov.interval(tr,ctl,normalization,chr,start,end,bin_width,do)
```

Arguments

tr	An instance of a CoverageBamFile or a CoverageBigWigFile object used to compute an arithmetic operation with the coverage values for the genomic interval defined by the chr, start and end parameters
ctl	An instance of a CoverageBamFile or a CoverageBigWigFile object used to compute a certain arithmetic operation with the coverage values for the genomic interval defined by the chr, start and end parameters
normalization	Normalize the coverage values in the resulting numeric vector. Possible normalization types are rpm (Read per million). Default=NULL
chr	chromosome of the genomic interval
start	start coordinate of the genomic interval. If the start and end arguments are not set, then the entire chromosome defined with the chr argument will be used for the calculations
end	end coordinate of the genomic interval
bin_width	calculate the average coverage within bins of size defined by this argument. Default is 1 nucleotide
do	specify the arithmetic operation to perform on the genomic interval. Possible values are 'log2ratio','ratio' or 'substraction'.Default=log2ratio

Details

The cov.interval function receives 2 CoverageBamFiles or CoverageBigWigFiles objects and returns a numeric vector with the result of doing a certain arithmetic operation with the two files using the coverage values for a certain position or genomic bin that is included in the genomic interval defined by the user using the chr, start and end arguments. The size of the bins is controlled by the bin_width argument and for each bin the average coverage value is computed and used for the different arithmetic operations.The arithmetic operation to perform is set with the do argument.

If the normalization parameter is set to 'rpm' then the number of reads aligned into the reference is required to perform the normalization. This number can be provided using the reads_mapped argument of the CoverageBamFile or CoverageBigWigFile objects, or it is automatically calculated if the function is used with a CoverageBamFile object. NOTE: If the reads_mapped argument is set then the automatic calculation will NOT be performed and the provided number will be taken as correct for all downstream calculations.

Author(s)

Ernesto Lowy <ernestolowy@gmail.com>

References

BAM format specification: http://samtools.sourceforge.net/SAMv1.pdf

WIG format specification: https://genome.ucsc.edu/FAQ/FAQformat.html

Examples

```
##BAM files

#get treatment and control test files
treat1file<-system.file("extdata","treat.bam",package="CoverageView")
control1file<-system.file("extdata","ctrl.bam",package="CoverageView")

#create two CoverageBamFile objects representing single-end alignments
trm1<-CoverageBamFile(treat1file,run_type="single")
ctl1<-CoverageBamFile(control1file,run_type="single")

#calculate the ratio of the coverages for the defined genomic interval using a bin_width equal to 10 nts
cov1=cov.interval(trm1,ctl1,chr="chrI",start=1,end=100,bin_width=10,do="ratio")

#create a WIG file with the obtained vector with the ratios
outfolder=system.file("extdata",package="CoverageView")
an_outfile1=paste(outfolder,"out.wig",sep="/")
export.wig(cov1,outfile=an_outfile1)

##BigWIG files

#get a treatment and control test files
treat2file<-system.file("extdata","treat.bw",package="CoverageView")
control2file<-system.file("extdata","ctrl.bw",package="CoverageView")

#create the 'treatment' and 'control' CoverageBigWigFile objects
trm2<-CoverageBigWigFile(path=treat2file,reads_mapped=28564510)
ctl2<-CoverageBigWigFile(path=control2file,reads_mapped=26713667)

#calculate the ratio of the coverages for the defined genomic interval
cov2=cov.interval(trm2,ctl2,bin_width=1,chr="chrI",start=1,end=1000,do='ratio')
```

cov.matrix *Compute a coverage matrix*

Description

This method generates a coverage matrix for which each column is a different genomic feature (i.e. the coordinate of the transcription start site) or genomic interval (defined by start/end coordinates) and each row can be either a position or a bin related to this feature or interval. Each element in the matrix is the result of computing the coverage value for this position/bin that is at a certain distance upstream or dowstream the genomic feature (i.e. the TSS) or it is in the genomic interval

Usage

```
## S4 method for signature 'ANY,missing'
cov.matrix(tr,coordfile,normalization,bin_width,extend,no_windows,offset,num_cores)
## S4 method for signature 'ANY,ANY'
cov.matrix(tr,ctl,coordfile,normalization,bin_width,extend,no_windows,offset,do,num_cores)
```

Arguments

tr	An instance of a CoverageBamFile or CoverageBigWigFile object
ctl	An instance of a CoverageBamFile or CoverageBigWigFile object. This argument is optional and if provided then this object will be used in the arithmetic operations with the coverages specified with the do argument. Default=NULL
coordfile	File in the BED or GFF format containing the coordinates used to generate the coverage matrix
normalization	Normalize the coverage values in the matrix. Possible normalization types are rpm (Read per million). Default=NULL
bin_width	Calculate the average coverage for bins of length defined by this argument. Default is 1 nucleotide
extend	Controls the number of nucleotides to extend relative to the anchor coordinates (i.e. TSS) as specified in the coordfile argument
no_windows	Number of windows used to divide the interval defined by the start and end coordinates as specified in the coordfile argument
offset	If the no_windows argument is set, this controls the number of windows to extend the interval defined by the start/end coordinates. Default=0
do	Used to set the arithmetic operation to perform in a certain position or bin. Possible values are: log2ratio, ratio, substraction. This argument only makes sense when either a CoverageBamFile or CoverageBigWigFile object is provided with the ctl argument. Default= log2ratio
num_cores	Number of cores to use. Default= The detectCores() function from the library parallel will be used to check the number of cores available in the system and this number will be used as the default value

Details

The cov.matrix function receives one or two CoverageBamFile or CoverageBigWigFile objects and returns a coverage matrix in which every column can be either a single genomic feature (i.e. the coordinates of the transcription start site for a gene) or a certain genomic interval defined by the start and end coordinates specified in the file used with the coordfile argument, and each row represents a different position/bin relative to the feature or a certain window within a genomic interval. This behaviour is controlled by the extend and no_windows parameters. For example, if one set the extend argument, then each row represents a single genomic position (or bin with a a size defined by the bin_width argument) relative to the anchor coordinate. Each matrix element will be the result of calculating the depth of coverage for that particular position/bin relative to the anchor coordinate defined in coordfile. If the bin_width parameter is >1 nt, then the average coverate for each bin is used for all downstream calculations. On the other hand, if the no_windows parameter is set, then each row in the matrix will represent a certain genomic window within an interval defined by the start and end coordinates as specified in the coordfile file. In this case, each element in the matrix will be the average coverage value for that particular genomic window. The number of windows in which the genomic interval is divided is controlled by the no_windows argument. Finally, if only one CoverageBamFile or CoverageBigWigFile is provided using the tr argument, then the matrix elements are the result of computing the coverage value or the average coverage value for a certain position or a certain genomic bin/window respectively. If, on the other hand, two files are provided using the tr and the ctl arguments, then the matrix elements will be the result of performing a certain arithmetic operation (specified with the do argument) with the coverages.

If the normalization parameter is set to 'rpm' then the number of reads aligned into the reference is required to perform the normalization. This number can be provided using the reads_mapped argument of the CoverageBamFile or CoverageBigWigFile objects, or it is automatically calculated if the function is used with a CoverageBamFile object. NOTE: If the reads_mapped argument is set then the automatic calculation will NOT be performed and the provided number will be taken as correct for all downstream calculations

Value

Returns a matrix

Author(s)

Ernesto Lowy <ernestolowy@gmail.com>

References

BAM format specification: http://samtools.sourceforge.net/SAMv1.pdf

WIG format specification: https://genome.ucsc.edu/FAQ/FAQformat.html

BED format specification: https://genome.ucsc.edu/FAQ/FAQformat.html

See Also

draw.profile draw.heatmap

Examples

```
###########
##BAM files
###########

## Generating a coverage matrix with a single BAM file

#get a BAM test file
aBAMfile<-system.file("extdata","treat.bam",package="CoverageView")

#get a BED file with the TSS (transcription start site) coordinates of some genes
bedfile<-system.file("extdata","testTSS.bed",package="CoverageView")

#create the CoverageBamFile object
trm<-CoverageBamFile(aBAMfile,reads_mapped=28564510)

#generate the coverage matrix extending 100 nts on each side of the provided TSS
#in the bedfile
DF1=cov.matrix(tr=trm,coordfile=bedfile,extend=100,num_cores=2)

#generate the coverage matrix extending 100 nts on each side of the TSS provided
#in the bedfile and normalize the resulting coverages
DF2=cov.matrix(tr=trm,coordfile=bedfile,extend=100,normalization="rpm",num_cores=2)

#generate the coverage matrix extending 100 nts on each side of the TSS provided
#in the bedfile and normalize the resulting coverages. This time we calculate the average
#coverage in bins of 2 nucleotides
DF3=cov.matrix(tr=trm,coordfile=bedfile,extend=100,normalization="rpm",bin_width=2,num_cores=2)

## Generating a coverage matrix with 2 BAM files

#get 2 BAM test files
treatBAMfile<-system.file("extdata","treat.bam",package="CoverageView")
ctrlBAMfile<-system.file("extdata","ctrl.bam",package="CoverageView")

#get a BED file with the TSS (transcription start site) coordinates of some genes
bedfile<-system.file("extdata","testTSS.bed",package="CoverageView")

#create 2 CoverageBamFile objects
trm<-CoverageBamFile(treatBAMfile,reads_mapped=28564510)
ctl<-CoverageBamFile(ctrlBAMfile,reads_mapped=26713667)

#generate the coverage matrix extending 100 nts on each side of the TSS provided
#in the bedfile and normalize the resulting coverages.The matrix elements are obtained
#by computing the ratio of the coverages of the trm against the ctl objects and then
#calculating the log2 of the ratios
DF4=cov.matrix(tr=trm,ctl=ctl,coordfile=bedfile,extend=100,normalization="rpm",do="log2ratio",num_cores=2)

######################

#get a treatment BAM test file
treatBAMfile<-system.file("extdata","treat.bam",package="CoverageView")
```

```
#get a GFF file with the chr,start and end coordinates of different genomic
#features (i.e. CDS)
gffile<-system.file("extdata","test.gff",package="CoverageView")

#create the 'treatment' CoverageBamFile object
trm<-CoverageBamFile(treatBAMfile,reads_mapped=28564510)

#generate the coverage matrix dividing each genomic interval defined by the start and
#end coordinates in the gff file into 10 windows and calculating the average coverage
#for each window
DF1=cov.matrix(trm,coordfile=gffile,no_windows=10,num_cores=2)

#generate the coverage matrix dividing each genomic interval defined by the start and
#end coordinates in the gff file into 10 windows and calculating the average coverage
#for each window, this time we extend the genomic interval by 1 window before the start
#coordinate and 1 window after the end coordinate (offset argument is set to 1)
DF1=cov.matrix(trm,coordfile=gffile,no_windows=10,offset=1)

############
##BigWIG files
############

## Generating a coverage matrix with a single WIG file

#get a bigWIG test file
abigWIGfile<-system.file("extdata","treat.bw",package="CoverageView")

#get a BED file with the TSS (transcription start site) coordinates of some genes
bedfile<-system.file("extdata","testTSS.bed",package="CoverageView")

#create the CoverageBigWigFile object
trm<-CoverageBigWigFile(abigWIGfile)

#generate the coverage matrix extending 100 nts on each side of the TSS provided
DF1=cov.matrix(trm,coordfile=bedfile,extend=100,bin_width=10,num_cores=2)

## Generating a coverage matrix with 2 BigWIG files

#get 2 BigWIG test files
treatBigWIGfile<-system.file("extdata","treat.bw",package="CoverageView")
ctrlBigWIGfile<-system.file("extdata","ctrl.bw",package="CoverageView")

#create 2 CoverageBigWigFile objects
trm<-CoverageBigWigFile(treatBigWIGfile)
ctl<-CoverageBigWigFile(ctrlBigWIGfile)

#generate the coverage matrix extending 100 nts on each side of the TSS provided
#in the bedfile .The matrix elements are obtained by computing the ratio of the
#coverages of the trm against the ctl objects and then calculating the log2 of the ratios
DF2=cov.matrix(tr=trm,ctl=ctl,coordfile=bedfile,extend=100,do="log2ratio",num_cores=2)
```

CoverageBamFile-class *Class* "CoverageBamFile"

Description

The CoverageBamFile class contains information on a BAM file and inherits fields from the BamFile class in the Rsamtools package.

Arguments

path	A character string that details the path to the BAM file
reads_mapped	A number representing the number of reads having an alignment. Default = 0
run_type	A character string describing the type of sequencing run. Possible values are 'single' and 'paired'. Default='single'
...	Additional arguments

Objects from the Class

Use CoverageBamFile() to create a reference to a BAM file.

This class represents an alignment in the BAM format that can be processed by the different methods of the CoverageView package.

The reads_mapped argument is optional and will be automatically calculated (if it is not provided) when the normalization argument is set in the cov.matrix and cov.interval functions

Author(s)

Ernesto Lowy <ernestolowy@gmail.com>

References

BAM format specification: http://samtools.sourceforge.net/SAMv1.pdf

See Also

- BamFile
- CoverageBigWigFile

Examples

```
#get a Bam test file
inputfile<-system.file("extdata","treat.bam",package="CoverageView")

#create a CoverageBamFile object
trm<-CoverageBamFile(inputfile)

#create a CoverageBamFile object including the information of the number
#of reads aligned and the sequencing run type
trm1<-CoverageBamFile(inputfile,reads_mapped=28654321,run_type="single")
```

CoverageBigWigFile-class
Class "CoverageBigWigFile"

Description

The CoverageBigWigFile class contains information on a BigWIG file and inherits fields from the BigWigFile class in the rtracklayer package.

Objects from the Class

Objects can be created by calls of the form CoverageBigWigFile(). This class represents an alignment in the BigWIG format for which each genomic position has a depth of coverage value associated.

The reads_mapped argument is optional but it is required if the normalization parameter is set in the cov.matrix and cov.interval functions

Slots

resource A character string that details the path to the BigWIG file

reads_mapped A number representing the number of reads aligned in the file. Default = 0

... Additional arguments

Author(s)

Ernesto Lowy <ernestolowy@gmail.com>

References

WIG format specification: https://genome.ucsc.edu/FAQ/FAQformat.html

See Also

- BigWigFile
- CoverageBamFile

Examples

```
##get BigWIG test file
inputfile<-system.file("extdata","treat.bw",package="CoverageView")

#create a CoverageBigWigFile object
trm<-CoverageBigWigFile(inputfile)

#create a CoverageBigWigFile object including the information of the number of reads aligned
trm1<-CoverageBigWigFile(inputfile,reads_mapped=28654321)
```

```
##get BigWIG test file
inputfile<-system.file("extdata","treat.bw",package="CoverageView")

#create a CoverageBigWigFile object
trm<-CoverageBigWigFile(inputfile)

#create a CoverageBigWigFile object including the information of the number of reads aligned
trm1<-CoverageBigWigFile(inputfile,reads_mapped=28654321)
```

`CoverageView-internal` *Internal Methods*

Description

Internal CoverageView functions

Details

These are not to be called by the user

`DF_H3K36me3` *Example of a coverage matrix using the ChIP-seq data for the H3K36me3 histone modification experiment*

Description

A coverage matrix calculated using the H3K36me3 ChIP-seq data from the GM12878 cell line. This matrix was computed using only the reads aligned into the Human chromosome 19

Usage

```
data(DF_H3K36me3)
```

Format

`matrix` with the normalized (read per million) coverages for the genomic interval corresponding to the gene body of a certain set of genes

Details

This matrix was obtained by running the `CoverageView` function named `cov.matrix` and setting the `no_windows` parameter to 100, the `normalization` parameter to "rpm" and the offset paremeter to 10. The BED format coordinate file was passed to the function through the `coordfile` argument and contained the genomic coordinates of a set of genes for which the region starting 2500 nucleotides before the transcription start site (TSS) and ending in the transcription end site (TES) contains at least one enrichment peak identified by MACS using the ChIP-seq data. All peaks, regardless its false discovery rate, were considered. For illustrative purposes only the genes located in the chr19 were analyzed.

Source

H3K36me3 ChIP-seq data set for the GM12878 cell line from the Broad Institute that was used as a test case in the Nature protocol named: Identifying ChIP-seq enrichment using MACS. Feng J, Liu T, Qin B, Zhang Y, Liu XS. Nat Protoc. 2012 Sep;7(9):1728-40.

References

Zhang et al. Model-based Analysis of ChIP-Seq (MACS). Genome Biol (2008) vol. 9 (9) pp. R137".

DF_H3K36me3_control *Example of a coverage matrix using the control replicate for the ChIP-seq data for the H3K36me3 histone modification experiment*

Description

A coverage matrix calculated using the control replicate for the H3K36me3 ChIP-seq data from the GM12878 cell line. This matrix was computed using only the reads aligned into the Human chromosome 19

Usage

```
data(DF_H3K36me3_control)
```

Format

`matrix` with the normalized (read per million) coverages for the genomic interval corresponding to the gene body of a certain set of genes

Details

This matrix was obtained by running the `CoverageView` function named `cov.matrix` and setting the `no_windows` parameter to 100, the `normalization` parameter to "rpm" and the offset paremeter to 10. The BED format coordinate file was passed to the function through the `coordfile` argument and contained the genomic coordinates of a set of genes for which the region starting 2500 nucleotides before the transcription start site (TSS) and ending in the transcription end site (TES) contains at least one enrichment peak identified by MACS using the ChIP-seq data that was used for generating the other coverage matrix example used in `CoverageView` (see `DF_H3K36me3`). For illustrative purposes only the genes located in the chr19 were analyzed.

Source

Control replicate for the H3K36me3 ChIP-seq data set for the GM12878 cell line from the Broad Institute that was used as a test case in the Nature protocol named: Identifying ChIP-seq enrichment using MACS. Feng J, Liu T, Qin B, Zhang Y, Liu XS. Nat Protoc. 2012 Sep;7(9):1728-40.

References

Zhang et al. Model-based Analysis of ChIP-Seq (MACS). Genome Biol (2008) vol. 9 (9) pp. R137".

`DF_H3K4me3` *Example of a coverage matrix using the ChIP-seq data for the H3K4me3 histone modification experiment*

Description

A coverage matrix calculated using the H3K4me3 ChIP-seq data from the K562 cell line. This matrix was computed using only the reads aligned into the Human chromosome 19

Usage

```
data(DF_H3K4me3)
```

Format

`matrix` with the normalized (read per million) coverages for the genomic interval corresponding to the gene body of a certain set of genes

Details

This matrix was obtained by running the `CoverageView` function named `cov.matrix` and setting the `no_windows` parameter to 100, the `normalization` parameter to "rpm" and the offset parameter to 10. The BED format coordinate file was passed to the function through the `coordfile` argument and contained the genomic coordinates of a set of genes for which the region starting 2500 nucleotides before the transcription start site (TSS) and ending in the transcription end site (TES) contains at least one enrichment peak identified by MACS using the ChIP-seq data. All peaks, regardless its false discovery rate, were considered. For illustrative purposes only the genes located in the chr19 were analyzed.

Source

H3K4me3 ChIP-seq data set for the K562 cell line from the HudsonAlpha Institute that was used as a test case in the Nature protocol named: Identifying ChIP-seq enrichment using MACS. Feng J, Liu T, Qin B, Zhang Y, Liu XS. Nat Protoc. 2012 Sep;7(9):1728-40.

References

Zhang et al. Model-based Analysis of ChIP-Seq (MACS). Genome Biol (2008) vol. 9 (9) pp. R137".

DF_H3K4me3_ctl *Example of a coverage matrix using the control replicate for the ChIP-seq data for the H3K4me3 histone modification experiment*

Description

A coverage matrix calculated using the control replicate for the H3K4me3 ChIP-seq data from the K562 cell line. This matrix was computed using only the reads aligned into the Human chromosome 19

Usage

```
data(DF_H3K4me3_ctl)
```

Format

`matrix` with the normalized (read per million) coverages for the genomic interval corresponding to the gene body of a certain set of genes

Details

This matrix was obtained by running the `CoverageView` function named `cov.matrix` and setting the `no_windows` parameter to 100, the `normalization` parameter to "rpm" and the offset parameter to 10. The BED format coordinate file was passed to the function through the `coordfile` argument and contained the genomic coordinates of a set of genes for which the region starting 2500 nucleotides before the transcription start site (TSS) and ending in the transcription end site (TES) contains at least one enrichment peak identified by MACS using the ChIP-seq data that was used for generating the other coverage matrix example used in `CoverageView` (see `DF_H3K4me3`). For illustrative purposes only the genes located in the chr19 were analyzed.

Source

Control replicate for the H3K4me3 ChIP-seq data set for the cell line from the HudsonAlpha Institute that was used as a test case in the Nature protocol named: Identifying ChIP-seq enrichment using MACS. Feng J, Liu T, Qin B, Zhang Y, Liu XS. Nat Protoc. 2012 Sep;7(9):1728-40.

References

Zhang et al. Model-based Analysis of ChIP-Seq (MACS). Genome Biol (2008) vol. 9 (9) pp. R137".

DF_H3K4me3_nopeaks_ratios *Example of a matrix with the ratio of the coverages using the ChIP-seq data for the H3K4me3 histone modification experiment*

Description

A matrix with the ratio of the coverages calculated using the H3K4me3 ChIP-seq data from the K562 cell line and its respective control replicate. This matrix was computed using only the reads aligned into the Human chromosome 19

Usage

```
data(DF_H3K4me3_nopeaks_ratios)
```

Format

`matrix` with the normalized (read per million) coverages for the genomic interval corresponding to the gene body of a certain set of genes

Details

This matrix was obtained by running the `CoverageView` function named `cov.matrix` with two CoverageBamFile objects (one sample and one control respectively) and setting the `no_windows` parameter to 100, the `normalization` parameter to "rpm" and the offset parameter to 10. The `do` argument was set to `ratio`, so each element in the matrix will be the result of calculating the ratio of the coverages between the sample and the control. The BED format coordinate file was passed to the function through the `coordfile` argument and contained the genomic coordinates of a set of genes for which the region starting 2500 nucleotides before the transcription start site (TSS) and ending in the transcription end site (TES) did not contain any enrichment peak identified by MACS using the ChIP-seq data. For illustrative purposes only the genes located in the chr19 were analyzed.

Source

H3K4me3 ChIP-seq data set for the K562 cell line from the HudsonAlpha Institute that was used as a test case in the Nature protocol named: Identifying ChIP-seq enrichment using MACS. Feng J, Liu T, Qin B, Zhang Y, Liu XS. Nat Protoc. 2012 Sep;7(9):1728-40.

References

Zhang et al. Model-based Analysis of ChIP-Seq (MACS). Genome Biol (2008) vol. 9 (9) pp. R137".

`draw.heatmap` *Draw a coverage heatmap*

Description

This method draws a heatmap of the coverage values using the matrix that is passed as an argument to the method

Usage

```
  ## S4 method for signature 'matrix'
draw.heatmap(data,outfile, color,...)
  ## S4 method for signature 'list'
draw.heatmap(data,outfile, color,...)
```

Arguments

`data`	Matrix or list of matrices generated by the function `cov.matrix`
`outfile`	URL of the `.png` file where the plot will be created
`color`	Graphical argument to set the color of the plot
`...`	Additional parameters

Details

This function is used to create a coverage heatmap of the matrix/list of matrices generated using the `cov.matrix` function

Author(s)

Ernesto Lowy <ernestolowy@gmail.com>

See Also

`cov.matrix`

Examples

```
## draw the heatmap for a coverate matrix that was previously calculated
# using the cov.matrix function for a BAM file containing ChIP-seq data
# from a H3K36me3 histone modification experiment
data(DF_H3K36me3)
draw.heatmap(DF_H3K36me3,outfile="testHeatmap.png")

## Now, draw two heatmaps for 2 different coverage matrices previously
# obtained for the same H3K36me3 histone modification experiment and its
# respective 'control' file
data(DF_H3K36me3_control)
```

```
# create a list with the two matrices
input_l=list(DF_H3K36me3,DF_H3K36me3_control)
draw.heatmap(input_l,outfile="testHeatmap.png")
```

`draw.profile` *Draw a coverage profile*

Description

This method draws a coverage profile for the coverage matrix that is passed as an argument to the method

Usage

```
## S4 method for signature 'matrix'
draw.profile(data,outfile=NULL,...)
## S4 method for signature 'list'
draw.profile(data,outfile=NULL,...)
```

Arguments

`data`	Matrix or list of matrices generated by the function `cov.matrix`
`outfile`	URL of the `.png` file where the plot will be created
`...`	Additional parameters

Details

Function to generate a plot in the `.png` format displaying the profile resulting from calculating the average of each row in the matrix generated by using the `cov.matrix` function. See the `cov.matrix` documentation for more details on the matrix used to draw the profile

Author(s)

Ernesto Lowy <ernestolowy@gmail.com>

See Also

`cov.matrix`

Examples

```
## draw the coverage profile for a coverate matrix that was previously calculated
# using the cov.matrix function for a BAM file containing ChIP-seq data
# from a H3K36me3 histone modification experiment
data(DF_H3K36me3)
draw.profile(DF_H3K36me3,ylab="coverage",outfile="testProfile.png",main="testProfile",col="red")

## Now, draw two profiles for 2 different coverage matrices previously
```

```
# obtained for the same H3K36me3 histone modification experiment and its
# respective 'control' file
data(DF_H3K36me3_control)

#create a list with the two matrices
input_l=list(DF_H3K36me3,DF_H3K36me3_control)
draw.profile(input_l,ylab="coverage",outfile="testProfile.png",main="testProfile")
```

`export.wig` *Generates a WIG file containing the coverage values for a certain genomic interval*

Description

Write the coverage values calculated using the `cov.interval` function into a WIG format file

Usage

```
## S4 method for signature 'numeric'
export.wig(cov,outfile=NULL)
```

Arguments

`cov`	Numeric vector containing the coverage values generated by the function `cov.interval`
`outfile`	Where to write the `.WIG` format file that will contain the coverages

Details

This method is used to write the coverage values calculated by the `cov.interval` function into a WIG format file.

Author(s)

Ernesto Lowy <ernestolowy@gmail.com>

References

WIG format specification: https://genome.ucsc.edu/FAQ/FAQformat.html

See Also

`cov.interval`

Examples

```
#get treatment and control test files
treat1file<-system.file("extdata","treat.bam",package="CoverageView")
control1file<-system.file("extdata","ctrl.bam",package="CoverageView")

#create two CoverageBamFile objects representing single-end alignments
trm1<-CoverageBamFile(treat1file,run_type="single")
ctl1<-CoverageBamFile(control1file,run_type="single")

#calculate the ratio of the coverages for the defined genomic interval using a bin_width equal to 10 nts
cov1=cov.interval(trm1,ctl1,chr="chrI",start=1,end=100,bin_width=10,do="ratio")

#create a WIG file with the obtained vector with the ratios
outfolder=system.file("extdata",package="CoverageView")
an_outfile1=paste(outfolder,"out.wig",sep="/")
export.wig(cov1,outfile=an_outfile1)
```

FoxA1_chr19 *Example of a coverage matrix using the ChIP-seq data for the FoxA1 transcription factor experiment*

Description

A coverage matrix calculated using the FoxA1 ChIP-seq data from the T-47D cell line. This matrix was computed using only the reads aligned into the Human chromosome 19

Usage

```
data(FoxA1_chr19)
```

Format

`matrix` with the coverages for the genomic interval corresponding to the gene body of a certain set of genes

Details

This matrix was obtained by running the `CoverageView` function named `cov.matrix` and setting the `extend` parameter to 1000 and the `bin\_width` paremeter to 10. The BED format coordinate file was passed to the function through the `coordfile` argument and contained the transcription start sites (TSS) of the genes that had a valid MACS peak in the region spanning 2500 nucleotides on each side of the TSS. All peaks, regardless its false discovery rate, were considered. For illustrative purposes only the genes located in the chr19 were analyzed.

Source

FoxA1 ChIP-seq data set for the T-47D cell line from the HudsonAlpha Institute that was used as a test case in the Nature protocol named: Identifying ChIP-seq enrichment using MACS. Feng J, Liu T, Qin B, Zhang Y, Liu XS. Nat Protoc. 2012 Sep;7(9):1728-40.

References

Zhang et al. Model-based Analysis of ChIP-Seq (MACS). Genome Biol (2008) vol. 9 (9) pp. R137".

`genome.covplot.cumdepth`

Generates a cumulative genome coverage plot

Description

This method generates a plot showing the percentage of the genome covered at different read depths

Usage

```
  ## S4 method for signature 'CoverageBamFile'
genome.covplot.cumdepth(data,outfile,max_depth)
  ## S4 method for signature 'list'
genome.covplot.cumdepth(data,outfile,max_depth)
```

Arguments

`data`	Either an instance of `CoverageBamFile` or a list of `CoverageBamFile` objects
`outfile`	URL of the `.png` file where the plot will be created
`max_depth`	Maximum read depth to be displayed in the X-axis

Details

This method receives either a single `CoverageBamFile` object or a list of `CoverageBamFile` objects and generates a plot for which the X-axis is a range of cumulative read depths and the Y-axis is the percentage of the genome covered at a certain read depth. If a list of `CoverageBamFile` objects is passed to the function then it will generate a different coloured line for each of the passed objects

Author(s)

Ernesto Lowy <ernestolowy@gmail.com>

See Also

`genome.covplot.depth`

Examples

```
##draw a cumulative coverage plot for a test case BAM file

#get a BAM test file
treatBAMfile<-system.file("extdata","treat.bam",package="CoverageView")

#create a CoverageBamFile object
trm<-CoverageBamFile(treatBAMfile)

#draw the plot
genome.covplot.cumdepth(trm,outfile="test.png")

#draw the plot setting the max_depth parameter (30X in this case)
genome.covplot.cumdepth(trm,outfile="test.png",max_depth=30)

##draw two overlapping cumulative coverage plots for two different BAM files

#get the first BAM file
treatBAMfile<-system.file("extdata","treat.bam",package="CoverageView")
#create the CoverageBamFile object
trm<-CoverageBamFile(treatBAMfile)

#get the second BAM test file
ctrlBAMfile<-system.file("extdata","ctrl.bam",package="CoverageView")
#create the CoverageBamFile object
ctl<-CoverageBamFile(ctrlBAMfile)

#create a list with the two files
input_d=list(trm,ctl)

#draw the plot
genome.covplot.cumdepth(input_d,outfile="test.png")
```

---

genome.covplot.depth *Generates a genome coverage plot*

Description

This method generates a plot showing the number of genomic positions reaching a certain read depth

Usage

```
  ## S4 method for signature 'CoverageBamFile'
genome.covplot.depth(data,outfile,max_depth)
  ## S4 method for signature 'list'
genome.covplot.depth(data,outfile,max_depth)
```

Arguments

data	Either an instance of a CoverageBamFile object or a list of CoverageBamFile objects
outfile	URL of the .png file where the plot will be created
max_depth	Maximum read depth to be displayed in the X-axis

Details

This method receives either a single CoverageBamFile object or a list of CoverageBamFile objects and generates a plot for which the X-axis represents a range of coverage read depths and the Y-axis corresponds to the number of megabases having a specific read coverage value. If a list of CoverageBamFile objects is passed to the function then it will generate a different coloured line for each of the objects

Author(s)

Ernesto Lowy <ernestolowy@gmail.com>

See Also

genome.covplot.cumdepth

Examples

```
##draw a coverage plot for a test case BAM file

#get a BAM test file
treatBAMfile<-system.file("extdata","treat.bam",package="CoverageView")

#create the CoverageBamFile object
trm<-CoverageBamFile(treatBAMfile)

#draw the plot
genome.covplot.depth(trm,outfile="test.png")

#draw the plot setting the max_depth parameter (70X in this case)
genome.covplot.depth(trm,outfile="test.png",max_depth=70)

##draw two overlapping coverage plots for two different test BAM files

#get a first BAM test file
treatBAMfile<-system.file("extdata","treat.bam",package="CoverageView")
#create the CoverageBamFile object
trm<-CoverageBamFile(treatBAMfile)

#get a second BAM test file
ctrlBAMfile<-system.file("extdata","ctrl.bam",package="CoverageView")
#create the CoverageBamFile object
ctl<-CoverageBamFile(ctrlBAMfile)
```

```
#create a list with the two files
input_d=list(trm,ctl)

#draw the plot
genome.covplot.depth(input_d,outfile="test.png")
```

`write.profile` *Write coverage values into a file*

Description

Write the coverage values calculated using the `cov.matrix` function into a file

Usage

```
## S4 method for signature 'matrix'
write.profile(DF,outfile=NULL)
```

Arguments

`DF`	Coverage matrix generated by the function `cov.matrix`
`outfile`	Where to write the `.txt` file that will contain the matrix

Details

This method is used to write the matrix containing the coverage values calculated by `cov.matrix` into a file.

Author(s)

Ernesto Lowy <ernestolowy@gmail.com>

See Also

`cov.matrix`

Examples

```
#get a BAM test file
treatBAMfile<-system.file("extdata","treat.bam",package="CoverageView")

#get a BED file with the TSS (transcription start site) coordinates of some genes
bedfile<-system.file("extdata","testTSS.bed",package="CoverageView")

#create the CoverageBamFile object
trm<-CoverageBamFile(treatBAMfile)

#generate the coverage matrix extending 100 nts on each side of the TSS provided
#in the BED file
```

```
DF1=cov.matrix(trm,coordfile=bedfile,extend=100)

#write the coverage matrix into the provided file
write.profile(DF1,outfile="DF1.txt")
```

Index

∗ **classes**
CoverageBamFile-class, 9
∗ **datasets**
DF_H3K36me3, 11
DF_H3K36me3_control, 12
DF_H3K4me3, 13
DF_H3K4me3_ctl, 14
DF_H3K4me3_nopeaks_ratios, 15
FoxA1_chr19, 19
∗ **internal**
CoverageView-internal, 11
.bed.to.GRanges *(CoverageView-internal)*, 11
.cov.bin *(CoverageView-internal)*, 11
.do.normalization *(CoverageView-internal)*, 11
.gff.to.GRanges *(CoverageView-internal)*, 11

BamFile, *9*
BigWigFile, *10*

cov.bin *(CoverageView-internal)*, 11
cov.bin,ANY,missing,numeric,GRanges,missing-method *(CoverageView-internal)*, 11
cov.bin,ANY,missing,numeric,missing,CoverageBamFile,missing-method *(CoverageView-internal)*, 11
cov.bin,ANY,missing,numeric,missing,CoverageBigWigFile,missing-method *(CoverageView-internal)*, 11
cov.bin,ANY,missing,numeric,missing,missing-method *(CoverageView-internal)*, 11
cov.bin,ANY,numeric,missing,GRanges,missing-method *(CoverageView-internal)*, 11
cov.bin,ANY,numeric,missing,missing,CoverageBamFile,missing-method *(CoverageView-internal)*, 11
cov.bin,ANY,numeric,missing,missing,CoverageBigWigFile,missing-method *(CoverageView-internal)*, 11
cov.bin,ANY,numeric,missing,missing,missing-method *(CoverageView-internal)*, 11
cov.bin,missing,missing,missing,GRanges,character-method *(CoverageView-internal)*, 11
cov.bin,missing,missing,missing,GRanges,missing,character-method *(CoverageView-internal)*, 11
cov.bin,missing,missing,missing,missing,character-method *(CoverageView-internal)*, 11
cov.bin,missing,missing,missing,missing,CoverageBamFile,character-method *(CoverageView-internal)*, 11
cov.bin-methods *(CoverageView-internal)*, 11
cov.interval, 3, *18*
cov.interval,CoverageBamFile,CoverageBamFile-method *(cov.interval)*, 3
cov.interval,CoverageBamFile-method *(cov.interval)*, 3
cov.interval,CoverageBigWigFile,CoverageBigWigFile-method *(cov.interval)*, 3
cov.interval,CoverageBigWigFile-method *(cov.interval)*, 3
cov.matrix, 5, *16*, *17*, *23*
cov.matrix,ANY,ANY-method *(cov.matrix)*, 5
cov.matrix,ANY,missing-method *(cov.matrix)*, 5
CoverageBamFile, *3*, *5*, *6*, *10*, *20*, *22*
CoverageBamFile *(CoverageBamFile-class)*, 9
CoverageBamFile-class, 9
CoverageBigWigFile, *3*, *5*, *6*, *9*
CoverageBigWigFile *(CoverageBigWigFile-class)*, 10
CoverageBigWigFile-class, 10
CoverageView *(CoverageView-package)*, 2
CoverageView-internal, 11
CoverageView-package, 2

DF_H3K36me3, 11
DF_H3K36me3_control, 12
DF_H3K4me3, 13
DF_H3K4me3_ctl, 14

DF_H3K4me3_nopeaks_ratios, 15
draw.heatmap, *6*, 16
draw.heatmap,list-method
(draw.heatmap), 16
draw.heatmap,matrix-method
(draw.heatmap), 16
draw.profile, *6*, 17
draw.profile,list-method
(draw.profile), 17
draw.profile,matrix-method
(draw.profile), 17

export.wig, 18
export.wig,numeric-method (export.wig), 18

FoxA1_chr19, 19

genome.covplot.cumdepth, 20, *22*
genome.covplot.cumdepth,CoverageBamFile-method
(genome.covplot.cumdepth), 20
genome.covplot.cumdepth,list-method
(genome.covplot.cumdepth), 20
genome.covplot.depth, *20*, 21
genome.covplot.depth,CoverageBamFile-method
(genome.covplot.depth), 21
genome.covplot.depth,list-method
(genome.covplot.depth), 21

imageScale (CoverageView-internal), 11

path,CoverageBamFile-method
(CoverageBamFile-class), 9

reads_mapped,CoverageBamFile-method
(CoverageBamFile-class), 9
run_type,CoverageBamFile-method
(CoverageBamFile-class), 9

show,CoverageBamFile-method
(CoverageBamFile-class), 9
show,CoverageBigWigFile-method
(CoverageBigWigFile-class), 10

write.profile, 23
write.profile,matrix-method
(write.profile), 23