Package 'ATACseqQC'

April 11, 2018

Type Package

Title ATAC-seq Quality Control

Version 1.2.10

Author Jianhong Ou, Jun Yu, Haibo Liu, Michelle Kelliher, Lucio Castilla, Nathan Lawson, Lihua Julie Zhu

Maintainer Jianhong Ou <jianhong.ou@duke.com>

Description ATAC-seq, an assay for Transposase-Accessible Chromatin using sequencing, is a rapid and sensitive method for chromatin accessibility analysis. It was developed as an alternative method to MNase-seq, FAIRE-seq and DNAse-seq. Comparing to the other methods, ATAC-seq requires less amount of the biological samples and time to process. In the process of analyzing several ATAC-seq dataset produced in our labs, we learned some of the unique aspects of the quality assessment for ATAC-seq data.To help users to quickly assess whether their ATAC-seq experiment is successful, we developed ATACseqQC package partially following the guideline published in Nature Method 2013 (Greenleaf et al.), including diagnostic plot of fragment size distribution, proportion of mitochondria reads, nucleosome positioning pattern, and CTCF or other Transcript Factor footprints.

Depends R (>= 3.4), BiocGenerics, S4Vectors

Imports BSgenome, Biostrings, ChIPpeakAnno, IRanges, GenomicRanges, GenomicAlignments, GenomeInfoDb, GenomicScores, graphics, grid, limma, Rsamtools, randomForest, rtracklayer, stats, motifStack

Suggests utils, BiocStyle, knitr, BSgenome.Hsapiens.UCSC.hg19, TxDb.Hsapiens.UCSC.hg19.knownGene, phastCons100way.UCSC.hg19, MotifDb, trackViewer, testthat

License GPL (>= 2)

LazyData TRUE

VignetteBuilder knitr

RoxygenNote 6.0.1

biocViews Sequencing, DNASeq, ATACSeq, GeneRegulation, QualityControl, Coverage, NucleosomePositioning

NeedsCompilation no

R topics documented:

`ATACseqQC-package` *ATAC-seq Quality Control*

Description

ATAC-seq, an assay for Transposase-Accessible Chromatin using sequencing, is a rapid and sensitive method for chromatin accessibility analysis. It was developed as an alternative method to MNase-seq, FAIRE-seq and DNAse-seq. Comparing to the other methods, ATAC-seq requires less amount of the biological samples and time to process. In the process of analyzing several ATAC-seq dataset produced in our labs, we learned some of the unique aspects of the quality assessment for ATAC-seq data.To help users to quickly assess whether their ATAC-seq experiment is successful, we developed ATACseqQC package partially following the guideline published in Nature Method 2013 (Greenleaf et al.), including diagnostic plot of fragment size distribution, proportion of mitochondria reads, nucleosome positioning pattern, and CTCF or other Transcript Factor footprints.

`bamQC` *Mapping quality control*

Description

Check the mapping rate, PCR duplication rate, and mitochondria reads contamination.

Usage

```
bamQC(bamfile, index = bamfile, mitochondria = "chrM",
  outPath = sub(".bam", ".clean.bam", basename(bamfile)))
```

Arguments

`bamfile`	character(1). File name of bam.
`index`	character(1). File name of index file.
`mitochondria`	character(1). Sequence name of mitochondria.
`outPath`	character(1). File name of cleaned bam.

Value

A list of quality summary.

Author(s)

Jianhong Ou

Examples

```
bamfile <- system.file("extdata", "GL1.bam", package="ATACseqQC")
bamQC(bamfile, outPath=NULL)
```

`enrichedFragments` *enrichment for nucleosome-free fragments and nucleosome signals*

Description

Get the enrichment signals for nucleosome-free fagments and nucleosomes.

Usage

```
enrichedFragments(bamfiles, index = bamfiles, gal, TSS, librarySize,
  upstream = 1010L, downstream = 1010L, n.tile = 101L,
  normal.method = "quantile", adjustFragmentLength = 80L,
  TSS.filter = 0.5, seqlev = paste0("chr", c(1:22, "X", "Y")))
```

Arguments

`bamfiles`	A vector of characters indicates the file names of bams.
`index`	The names of the index file of the 'BAM' file being processed; This is given without the '.bai' extension.
`gal`	A GAlignmentsList object or a list of GAlignmentPairs. If bamfiles is missing, gal is required.
`TSS`	an object of GRanges indicates the transcript start sites. All the width of TSS should equal to 1. Otherwise, TSS will be reset to the center of input TSS.
`librarySize`	A vector of numeric indicates the library size. Output of estLibSize
`upstream, downstream`	numeric(1) or integer(1). Upstream and downstream size from each TSS.
`n.tile`	numeric(1) or integer(1). The number of tiles to generate for each element of TSS.
`normal.method`	character(1). Normalization methods, could be "none" or "quantile". See normalizeBetweenArrays.
`adjustFragmentLength`	numeric(1) or integer(1). The size of fragment to be adjusted to. Default is set to half of the nucleosome size (80)
`TSS.filter`	numeric(1). The filter for signal strength of each TSS. Default 0.5 indicates the average signal strength for the TSS from upstream to downstream bins should be greater than 0.5.
`seqlev`	A vector of character indicates the sequence names to be considered.

Value

A list of matrixes. In each matrix, each row record the signals for corresponding feature.

Author(s)

Jianhong Ou

Examples

```
bamfiles <- system.file("extdata", "splited",
                        c("NucleosomeFree.bam",
                          "mononucleosome.bam",
                          "dinucleosome.bam",
                          "trinucleosome.bam"), package="ATACseqQC")
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txs <- transcripts(TxDb.Hsapiens.UCSC.hg19.knownGene)
TSS <- promoters(txs, upstream=0, downstream=1)
library(ChIPpeakAnno)
librarySize <- estLibSize(bamfiles)
sigs <- enrichedFragments(bamfiles, TSS=TSS, librarySize=librarySize,
                          seqlev="chr1", TSS.filter=0)
sigs.log2 <- lapply(sigs, function(.ele) log2(.ele+1))
featureAlignedHeatmap(sigs.log2, reCenterPeaks(TSS, width=2020),
                      zeroAt=.5, n.tile=101, upper.extreme=2)
featureAlignedDistribution(sigs, reCenterPeaks(TSS, width=2020),
                           zeroAt=.5, n.tile=101, type="l")
```

---

factorFootprints *plot ATAC-seq footprints infer factor occupancy genome wide*

Description

Aggregate ATAC-seq footprint for a given motif generated over binding sites within the genome.

Usage

```
factorFootprints(bamfiles, index = bamfiles, pfm, genome,
  min.score = "95%", bindingSites, seqlev = paste0("chr", c(1:22, "X",
  "Y")), upstream = 100, downstream = 100)
```

Arguments

bamfiles	A vector of characters indicates the file names of bams.
index	The names of the index file of the 'BAM' file being processed; This is given without the '.bai' extension.
pfm	A Position frequency Matrix represented as a numeric matrix with row names A, C, G and T.
genome	An object of BSgenome.
min.score	The minimum score for counting a match. Can be given as a character string containing a percentage (e.g. "95 score or as a single number. See matchPWM.

`bindingSites` A object of GRanges indicates candidate binding sites (eg. the output of fimo).

`seqlev` A vector of characters indicates the sequence levels.

`upstream, downstream`
numeric(1) or integer(1). Upstream and downstream of the binding region for aggregate ATAC-seq footprint.

Value

an invisible list of matrixes with the signals for plot. It includes: - signal mean values of coverage for positive strand and negative strand in feature regions - spearman.correlation spearman correlations of cleavage counts in the highest 10-nucleotide-window and binding prediction score. - bindingSites predicted binding sites.

Author(s)

Jianhong Ou, Julie Zhu

References

Chen, K., Xi, Y., Pan, X., Li, Z., Kaestner, K., Tyler, J., Dent, S., He, X. and Li, W., 2013. DANPOS: dynamic analysis of nucleosome position and occupancy by sequencing. Genome research, 23(2), pp.341-351.

Examples

```
bamfile <- system.file("extdata", "GL1.bam",
                       package="ATACseqQC")
library(MotifDb)
CTCF <- query(MotifDb, c("CTCF"))
CTCF <- as.list(CTCF)
library(BSgenome.Hsapiens.UCSC.hg19)
factorFootprints(bamfile, pfm=CTCF[[1]],
                 genome=Hsapiens,
                 min.score="95%", seqlev="chr1",
                 upstream=100, downstream=100)
```

---

fragSizeDist *fragment size distribution*

Description

estimate the fragment size of bams

Usage

```
fragSizeDist(bamFiles, bamFiles.labels, ylim = NULL, logYlim = NULL)
```

Arguments

bamFiles	A vector of characters indicates the file names of bams.
bamFiles.labels	labels of the bam files, used for pdf file naming.
ylim	numeric(2). ylim of the histogram.
logYlim	numeric(2). ylim of log-transformed histogram for the insert.

Value

Invisible fragment length distribution list.

Author(s)

Jianhong Ou

Examples

```
bamFiles <- system.file("extdata", "GL1.bam", package="ATACseqQC")
bamFiles.labels <- "GL1"
fragSizeDist(bamFiles, bamFiles.labels,
             ylim=c(0, 1e4), logYlim=log10(c(5e-3, 2)))
```

---

plotFootprints *Plots a footprint estimated by Centipede*

Description

Visualizing the footprint profile

Usage

```
plotFootprints(Profile, Mlen = 0, xlab = "Dist. to motif (bp)",
  ylab = "Cut-site probability", legTitle, newpage = TRUE, motif)
```

Arguments

Profile	A vector with the profile estimated by CENTIPEDE
Mlen	Length of the motif for drawing vertical lines delimiting it
xlab	Label of the x axis
ylab	Label for the y axis
legTitle	Title for one of the plot corners
newpage	Plot the figure in a new page?
motif	a pfm object.

Value

Null.

Author(s)

Jianhong Ou

Examples

```
library(MotifDb)
CTCF <- query(MotifDb, c("CTCF"))
CTCF <- as.list(CTCF)
motif <- new("pfm", mat=CTCF[[1]], name="CTCF")
ATACseqQC:::plotFootprints(Profile=sample.int(500),
                        Mlen=ncol(CTCF[[1]]), motif=motif)
```

---

pwmscores *max PWM scores for sequences*

Description

calculate the maximal PWM scores for each given sequences

Usage

```
pwmscores(pwm, subject)
```

Arguments

pwm	A Position Weight Matrix represented as a numeric matrix with row names A, C, G and T.
subject	Typically a DNAString object. A Views objecton a DNAString subject, a MaskedDNAString object, or a single character string, are also supported. IUPAC ambiguity letters in subject are ignored (i.e. assigned weight 0) with a warning.

Value

a numeric vector

Author(s)

Jianhong

`readBamFile` *read in bam files*

Description

wraper for readGAlignments/readGAlignmentsList to read in bam files.

Usage

```
readBamFile(bamFile, which, tag = character(0), what = c("qname", "flag",
  "mapq", "isize", "seq", "qual", "mrnm"),
  flag = scanBamFlag(isSecondaryAlignment = FALSE, isUnmappedQuery = FALSE,
  isNotPassingQualityControls = FALSE), asMates = FALSE, ...)
```

Arguments

`bamFile`	character(1). Bam file name.
`which`	A GRanges, RangesList, or any object that can be coerced to a RangesList, or missing object, from which a IRangesList instance will be constructed. See ScanBamParam.
`tag`	A vector of characters indicates the tag names to be read. See ScanBamParam.
`what`	A character vector naming the fields to return. Fields are described on the Rsamtools[scanBam] help page.
`flag`	An integer(2) vector used to filter reads based on their 'flag' entry. This is most easily created with the Rsamtools[scanBamFlag] helper function.
`asMates`	logical(1). Paired ends or not
`...`	parameters used by readGAlignmentsList or readGAlignments

Value

A GAlignmentsList object when asMats=TRUE, otherwise A GAlignments object.

Author(s)

Jianhong Ou

Examples

```
library(BSgenome.Hsapiens.UCSC.hg19)
which <- as(seqinfo(Hsapiens)["chr1"], "GRanges")
bamfile <- system.file("extdata", "GL1.bam",
                       package="ATACseqQC", mustWork=TRUE)
readBamFile(bamfile, which=which, asMates=TRUE)
```

shiftGAlignmentsList *shift 5' ends*

Description

shift the GAlignmentsLists by 5' ends. All reads aligning to the positive strand will be offset by +4bp, and all reads aligning to the negative strand will be offset -5bp by default.

Usage

```
shiftGAlignmentsList(gal, positive = 4L, negative = 5L)
```

Arguments

gal	An object of GAlignmentsList.
positive	integer(1). the size to be shift for positive strand
negative	integer(1). the size to be shift for negative strand

Value

An object of GAlignments with 5' end shifted reads.

Author(s)

Jianhong Ou

Examples

```
bamfile <- system.file("extdata", "GL1.bam", package="ATACseqQC")
tags <- c("AS", "XN", "XM", "XO", "XG", "NM", "MD", "YS", "YT")
library(BSgenome.Hsapiens.UCSC.hg19)
which <- as(seqinfo(Hsapiens)["chr1"], "GRanges")
gal <- readBamFile(bamfile, tag=tags, which=which, asMates=TRUE)
objs <- shiftGAlignmentsList(gal)
export(objs, "shift.bam")
```

shiftReads *shift read for 5'end*

Description

shift reads for 5'ends

Usage

```
shiftReads(x, positive = 4L, negative = 5L)
```

Arguments

x	an object of GAlignments
positive	integer(1). the size to be shift for positive strand
negative	integer(1). the size to be shift for negative strand

Value

an object of GAlignments

Author(s)

Jianhong Ou

---

splitBam *prepare bam files for downstream analysis*

Description

shift the bam files by 5'ends and split the bam files.

Usage

```
splitBam(bamfile, tags, outPath = NULL, txs, genome, conservation,
  positive = 4L, negative = 5L, breaks = c(0, 100, 180, 247, 315, 473,
  558, 615, Inf), labels = c("NucleosomeFree", "inter1", "mononucleosome",
  "inter2", "dinucleosome", "inter3", "trinucleosome", "others"),
  seqlev = paste0("chr", c(1:22, "X", "Y")), cutoff = 0.8)
```

Arguments

bamfile	character(1). File name of bam.
tags	A vector of characters indicates the tags in bam file.
outPath	Output file path.
txs	GRanges of transcripts.
genome	An object of BSgenome
conservation	An object of GScores.
positive	integer(1). the size to be shift for positive strand
negative	integer(1). the size to be shift for negative strand
breaks	A numeric vector for fragment size of nucleosome freee, mononucleosome, dinucleosome and trinucleosome
labels	A vector of characters indicates the labels for the levels of the resulting category. The length of labels = length of breaks - 1
seqlev	A vector of characters indicates the sequence levels.
cutoff	numeric(1). Cutoff value for prediction by randomForest.

Value

an invisible list of GAlignments

Author(s)

Jianhong Ou

See Also

shiftGAlignmentsList, splitGAlignmentsByCut, and writeListOfGAlignments

Examples

```
bamfile <- system.file("extdata", "GL1.bam", package="ATACseqQC")
tags <- c("AS", "XN", "XM", "XO", "XG", "NM", "MD", "YS", "YT")
library(BSgenome.Hsapiens.UCSC.hg19)
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txs <- transcripts(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(phastCons100way.UCSC.hg19)
objs <- splitBam(bamfile, tags,
                 txs=txs, genome=Hsapiens,
                 conservation=phastCons100way.UCSC.hg19,
                 seqlev="chr1")
```

---

splitGAlignmentsByCut *split bams into nucleosome free, mononucleosome, dinucleosome and trinucleosome*

Description

use random forest to split the reads into nucleosome free, mononucleosome, dinucleosome and trinucleosome. The features used in random forest including fragment length, GC content, and UCSC phastCons conservation scores.

Usage

```
splitGAlignmentsByCut(obj, txs, genome, conservation, breaks = c(0, 100, 180,
  247, 315, 473, 558, 615, Inf), labels = c("NucleosomeFree", "inter1",
  "mononucleosome", "inter2", "dinucleosome", "inter3", "trinucleosome",
  "others"), labelsOfNucleosomeFree = "NucleosomeFree",
  labelsOfMononucleosome = "mononucleosome", trainningSetPercentage = 0.15,
  cutoff = 0.8, halfSizeOfNucleosome = 80L)
```

Arguments

obj	an object of GAlignments
txs	GRanges of transcripts
genome	an object of BSgenome
conservation	an object of GScores.
breaks	a numeric vector for fragment size of nucleosome freee, mononucleosome, dinucleosome and trinucleosome. The breaks pre-defined here is following the description of Greenleaf's paper (see reference).
labels	a character vector for labels of the levels of the resulting category.

`labelsOfNucleosomeFree, labelsOfMononucleosome`
: character(1). The label for nucleosome free and mononucleosome.

`trainningSetPercentage`
: numeric(1) between 0 and 1. Percentage of trainning set from top coverage.

`cutoff`
: numeric(1) between 0 and 1. cutoff value for prediction.

`halfSizeOfNucleosome`
: numeric(1) or integer(1). Thre read length will be adjusted to half of the nucleosome size to enhance the signal-to-noise ratio.

Value

a list of GAlignments

Author(s)

Jianhong Ou

References

Buenrostro, J.D., Giresi, P.G., Zaba, L.C., Chang, H.Y. and Greenleaf, W.J., 2013. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. Nature methods, 10(12), pp.1213-1218.

Chen, K., Xi, Y., Pan, X., Li, Z., Kaestner, K., Tyler, J., Dent, S., He, X. and Li, W., 2013. DANPOS: dynamic analysis of nucleosome position and occupancy by sequencing. Genome research, 23(2), pp.341-351.

Examples

```
library(GenomicRanges)
bamfile <- system.file("extdata", "GL1.bam",
                       package="ATACseqQC", mustWork=TRUE)
tags <- c("AS", "XN", "XM", "XO", "XG", "NM", "MD", "YS", "YT")
gal1 <- readBamFile(bamFile=bamfile, tag=tags,
                    which=GRanges("chr1", IRanges(1, 1e6)),
                    asMates=FALSE)
names(gal1) <- mcols(gal1)$qname
library(BSgenome.Hsapiens.UCSC.hg19)
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txs <- transcripts(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(phastCons100way.UCSC.hg19)
splitGAlignmentsByCut(gal1, txs=txs, genome=Hsapiens,
                      conservation=phastCons100way.UCSC.hg19)
```

---

writeListOfGAlignments

export list of GAlignments into bam files

Description

wraper for export to export list of GAlignment into bam files.

Usage

```
writeListOfGAlignments(objs, outPath = ".")
```

Arguments

objs	A list of GAlignments.
outPath	character(1). Output file path.

Value

status of export.

Author(s)

Jianhong Ou

Examples

```
library(GenomicAlignments)
gal1 <- GAlignments(seqnames=Rle("chr1"), pos=1L, cigar="10M",
                    strand=Rle(strand(c("+"))), names="a", score=1)
galist <- GAlignmentsList(a=gal1)
writeListOfGAlignments(galist)
```

Index