Package ‘gcapc’

April 27, 2025

Title GC Aware Peak Caller

Version 1.32.0

Author Mingxiang Teng and Rafael A. Irizarry

Maintainer Mingxiang Teng <tengmx@gmail.com>

Description Peak calling for ChIP-seq data with consideration of potential GC bias in sequencing reads. GC bias is first estimated with generalized linear mixture models using effective GC strategy, then applied into peak significance estimation.

Depends R (>= 3.4)

Imports BiocGenerics, GenomeInfoDb, S4Vectors, IRanges, Biostrings, BSgenome, GenomicRanges, Rsamtools, GenomicAlignments, matrixStats, MASS, splines, grDevices, graphics, stats, methods

VignetteBuilder knitr

Suggests BiocStyle, knitr, rmarkdown, BSgenome.Hsapiens.UCSC.hg19, BSgenome.Mmusculus.UCSC.mm10

URL https://github.com/tengmx/gcapc

License GPL-3

LazyData true

biocViews Sequencing, ChIPSeq, BatchEffect, PeakDetection

RoxygenNote 6.0.1

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

git_branch RELEASE_3_21

git_last_commit bd66b5c

git_last_commit_date 2025-04-15

Repository Bioconductor 3.21

Date/Publication 2025-04-27

Contents

bindWidth 2
gcapcPeaks 3
gcEffects 4
peaksCAT 7
read5endCoverage 8
refinePeaks 9
refineSites 10

Index **13**

`bindWidth` *ChIP-seq Binding Width And Peak Window Size Estimation*

Description

ChIP-seq experiments usually use crosslinking strategy to capture sequencing fragments. The fragment location is affected by at least but not limited to two factors, protein real binding and crosslinking operation. This function estimate size of binding part in crosslinked DNA-protein complexes, and denoted that as ChIP-seq binding width.Also, the peak detection window half size is estimated based on binding width.

Usage

```
bindWidth(coverage, range = c(50L, 500L), step = 50L, odd = TRUE)
```

Arguments

`coverage`	A list object returned by function `read5endCoverage`.
`range`	A non-nagative integer vector with length 2. This vector set the range within which binding width and peak window size are estimated. Default c(50,500) represents most ChIP-seq experiments.
`step`	A non-negative integer to set the resolution of binding width estimation within `range`. This value will be tuned if `auto` is TRUE. Default 50 is based on default value of `range`.
`odd`	A logical vector which, when TRUE, only allows return odd number of binding width, which is preferred by the effective GC content estimation. Default: TRUE.

Value

A numeric vector with 2 elements: Estimated binding width and half size of peak detection window.

Examples

```
bam <- system.file("extdata", "chipseq.bam", package="gcapc")
cov <- read5endCoverage(bam)
bindWidth(cov)
```

gcapcPeaks *GC Effects Aware Peak Calling*

Description

This function calls ChIP-seq peaks using potential GC effects information. Enrichment scores are calculated on sliding windows of prefiltered large regions, with GC effects considered. Permutation analysis is used to determine significant binding peaks.

Usage

```
gcapcPeaks(coverage, gcbias, bdwidth, flank = NULL, prefilter = 4L,
  permute = 5L, pv = 0.05, plot = FALSE, genome = "hg19",
  gctype = c("ladder", "tricube"))
```

Arguments

`coverage`	A list object returned by function `read5endCoverage`.
`gcbias`	A list object returned by function `gcEffects`.
`bdwidth`	A non-negative integer vector with two elements specifying ChIP-seq binding width and peak detection half window size. Usually generated by function `bindWidth`. A bad estimation of bdwidth results no meaning of downstream analysis. The values need to be the same as it is when calculating `gcbias`.
`flank`	A non-negative integer specifying the flanking width of ChIP-seq binding. This parameter provides the flexibility that reads appear in flankings by decreased probabilities as increased distance from binding region. This paramter helps to define effective GC content calculation. Default is NULL, which means this paramater will be calculated from `bdwidth`. However, if customized numbers provided, there won't be recalucation for this parameter; instead, the 2nd elements of `bdwidth` will be recalculated based on `flank`. The value needs to be the same as it is when calculating `gcbias`.
`prefilter`	A non-negative integer specifying the minimum of reads to qualify a potential binding region. Regions with total of reads from forward and reverse strands larger or equivalent to `prefilter` are selected for downstream analysis. Default is 4.
`permute`	A non-negative integer specifying times of permutation to be performed. Default is 5. When whole large genome is used, such as human genome, 5 times of permutation could be enough.
`pv`	A numeric specifying p-value cutoff for significant binding peaks. Default is 0.05.
`plot`	A logical vector which, when TRUE (default), returns density plots of real and permutation enrichment scores.

genome — A BSgenome object containing the sequences of the reference genome that was used to align the reads, or the name of this reference genome specified in a way that is accepted by the getBSgenome function defined in the **BSgenome** software package. In that case the corresponding BSgenome data package needs to be already installed (see ?getBSgenome in the **BSgenome** package for the details). The value needs to be the same as it is when calculating gcbias.

gctype — A character vector specifying choice of method to calculate effective GC content. Default ladder is based on uniformed fragment distribution. A more smoother method based on tricube assumption is also allowed. However, tricube should be not used if estimated peak half size is 3 times or more larger than estimated bind width. The value needs to be the same as it is when calculating gcbias.

Value

A GRanges of peaks with meta columns:

es — Estimated enrichment score.

pv — p-value.

Examples

```
bam <- system.file("extdata", "chipseq.bam", package="gcapc")
cov <- read5endCoverage(bam)
bdw <- bindWidth(cov)
gcb <- gcEffects(cov, bdw, sampling = c(0.15,1))
gcapcPeaks(cov, gcb, bdw)
```

---

gcEffects — *ChIP-seq GC Effects Estimation*

Description

GC effects are estimated based on effective GC content and reads count on genome-wide windows, using generalized linear mixture models. Genome wide windows are randomly or supervised sampled with given proportions. GC effects of background and foreground are estimated separately.

Usage

```
gcEffects(coverage, bdwidth, flank = NULL, plot = TRUE, sampling = c(0.05,
  1), supervise = GRanges(), gcrange = c(0.3, 0.8), emtrace = TRUE,
  model = c("nbinom", "poisson"), mu0 = 1, mu1 = 50, theta0 = mu0,
  theta1 = mu1, p = 0.02, converge = 0.001, genome = "hg19",
  gctype = c("ladder", "tricube"))
```

Arguments

`coverage` A list object returned by function `read5endCoverage`.

`bdwidth` A non-negative integer vector with two elements specifying ChIP-seq binding width and peak detection half window size. Usually generated by function `bindWidth`. A bad estimation of bdwidth results no meaning of downstream analysis.

`flank` A non-negative integer specifying the flanking width of ChIP-seq binding. This parameter provides the flexibility that reads appear in flankings by decreased probabilities as increased distance from binding region. This paramter helps to define effective GC content calculation. Default is NULL, which means this paramater will be calculated from `bdwidth`. However, if customized numbers provided, there won't be recalucation for this parameter; instead, the 2nd elements of `bdwidth` will be recalculated based on `flank`.

`plot` A logical vector which, when TRUE (default), returns plots of intermediate results.

`sampling` A numeric vector with length 2. The first number specifies the proportion of regions to be sampled for GC effects estimation. The second number specifies the repeat times for sampling. Default c(0.05,1) gives pretty robust estimation for human genome. However, smaller genomes might need both higher proportion and more repeat times for robust estimation.

`supervise` A GRanges object specifying peak regions in the studied data, such as peaks called by peak callers, e.g. MACS & SPP. These peak regions provide supervised window sampling for both mixtures in the generalized linear model. Default no supervising. Or, if provided peak regions have too few covered windows, supervised sampling will be replaced by random sampling automatically.

`gcrange` A non-negative numeric vector with length 2. This vector sets the range of GC content to filter regions for GC effect estimation. For human, most regions have GC content between 0.3 and 0.8, which is set as the default. Other regions with GC content beyond this range will be ignored. This range is critical when very few foreground regions are selected for mixture model fitting, since outliers could drive the regression lines. Thus, if possible, first make a scatter plot between counts and GC content to decide this parameter. Alternatively, select a narrower range, e.g. c(0.35,0.7), to aviod outlier effects from both high and low GC-content regions.

`emtrace` A logical vector which, when TRUE (default), allows to print the trace of log likelihood changes in EM iterations.

`model` A character specifying the distribution model to be used in generalized linear model fitting. The default is negative binomial(`nbinom`), while `poisson` is also supported currently. Based on our tests of multiple datasets, mostly poisson is a very good approximation of negative binomial, and provides much faster model fitting.

`mu0` A non-negative numeric initiating read count signals for background regions. This is treated as the starting value of background mean for poisson/nbinom fitting. Default is 1.

mu1	A non-negative numeric initiating read count signals for foreground regions. This is treated as the starting value of foreground mean for poisson/nbinom fitting, Default is 50.
theta0	A non-negative numeric initiating the shape parameter of negative binomial model for background regions. For more detail, see theta in glm.nb function.
theta1	A non-negative numeric initiating the shape parameter of negative binomial model for foreground regions. For more detail, see theta in glm.nb function.
p	A non-negative numeric specifying the proportion of foreground regions in all estimated regions. This is treated as a starting value for EM algorithm. Default is 0.02.
converge	A non-negative numeric specifying the condition of EM algorithm termination. EM algorithm stops when the ratio of log likelihood increment to whole log likelihood is less or equivalent to converge.
genome	A BSgenome object containing the sequences of the reference genome that was used to align the reads, or the name of this reference genome specified in a way that is accepted by the getBSgenome function defined in the **BSgenome** software package. In that case the corresponding BSgenome data package needs to be already installed (see ?getBSgenome in the **BSgenome** package for the details).
gctype	A character vector specifying choice of method to calculate effective GC content. Default ladder is based on uniformed fragment distribution. A more smoother method based on tricube assumption is also allowed. However, tricube should be not used if estimated peak half size is 3 times or more larger than estimated bind width.

Value

A list of objects

gc	The GC contents at which GC effects are estimated.
mu0	Predicted background signals at GC content gc.
mu1	Predicted foreground signals at GC content gc .
mu0med0	Median of predicted background signals.
mu1med1	Median of predicted foreground signals.
mu0med1	Median of predicted background signals at GC content of foreground windows.
mu1med0	Median of predicted foreground signals at GC content of background windows.

Examples

```
bam <- system.file("extdata", "chipseq.bam", package="gcapc")
cov <- read5endCoverage(bam)
bdw <- bindWidth(cov)
gcb <- gcEffects(cov, bdw, sampling = c(0.15,1))
```

peaksCAT *CATplot of Peaks*

Description

Plot the consistancy between two peak lists by their significance.

Usage

```
peaksCAT(x, y, ranks = seq(200, min(length(x), length(y), 20000), 50),
  exclude = GRanges(), seqinfo = NULL, esx = 1, esy = 1, add = FALSE,
  ...)
```

Arguments

x	A GRanges of identified peaks from one method or one replicate. At least one meta column should be included to allow for significance ranking of peaks.
y	A GRanges of identified peaks from compared method or anoter replicate. At least one meta column should be included to allow for significance ranking of peaks.
ranks	A non-negative integer vector specifying the ranks to be used for CAT plot.
exclude	A GRanges object specifying regions to be excluded for CAT plot, such as the blacklist regions proposed by ENCODE Consortium.
seqinfo	A vector of chromosome names to limit the CAT plot to selected chromosomes. Chromosome names here must be in the same format as `seqnames` in `x` and `y`. This parameter also helps exclude uncommon chromosomes, e.g. using value `paste0('chr',c(1:22,'X','Y'))` for human genome. Default: NULL means no limit to chromosomes.
esx	A non-negative integer specifying which meta column of `x` to be used to rank peak significance. Larger values in this column should indicate higher significance.
esy	A non-negative integer specifying which meta column of `y` to be used to rank peak significance. Larger values in this column should indicate higher significance.
add	A logical vector which, when TRUE, adds the current plotting line to existing plots. FALSE will generate a new plot.
...	Other parameters passed to `plot` or `lines`.

Value

A CAT plot.

Examples

```
bam <- system.file("extdata", "chipseq.bam", package="gcapc")
cov <- read5endCoverage(bam)
bdw <- bindWidth(cov)
gcb1 <- gcEffects(cov, bdw, sampling=c(0.15,1), plot=FALSE)
peaks1 <- gcapcPeaks(cov, gcb1, bdw)
gcb2 <- gcEffects(cov, bdw, sampling=c(0.2,1), plot=FALSE)
peaks2 <- gcapcPeaks(cov, gcb2, bdw)
peaksCAT(peaks1, peaks2, ranks=seq(100,200,5), ylim=c(0,1))
```

---

read5endCoverage *Reads Coverage Using 5-end Base*

Description

Reads coverage in single base pair resolution using only 5-prime end of BAM file records. Coverages are reported for forward and reverse strands separately. Options for customized filtering of BAM records are provided.

Usage

```
read5endCoverage(bam, chroms = NULL, mapq = 30L, duplicate = FALSE,
  flag = scanBamFlag(isUnmappedQuery = FALSE, isSecondaryAlignment = FALSE,
  isNotPassingQualityControls = FALSE))
```

Arguments

`bam`	The path to a BAM file, which is sorted and indexed.
`chroms`	NULL or a vector of chromosome names that compatible with the provided BAM file. Reads coverage will be generated for these chromosomes. Default (NULL) will use all chromosomes in BAM file.
`mapq`	A non-negative integer specifying the minimum mapping quality to include. BAM records with mapping qualities less than `mapq` are discarded.
`duplicate`	A logical vector which, when FALSE (Default), returns maximum coverage of 1 for every base pair. Reads that start at the same position but on different strands are not treated as duplicates.
`flag`	A returned object by `Rsamtools::scanBamFlag`. Additional options for BAM records filtering.

Value

A list of two objects by `GenomicRanges::coverage`

`fwd`	Coverage object for forward strand.
`rev`	Coverage object for reverse strand.

Examples

```
bam <- system.file("extdata", "chipseq.bam", package="gcapc")
read5endCoverage(bam)
```

`refinePeaks` *Refine Peaks with GC Effects*

Description

This function refines the ranks (i.e. significance/pvalue) of pre-determined peaks by potential GC effects. These peaks can be obtained from other peak callers, e.g. MACS or SPP.

Usage

```
refinePeaks(coverage, gcbias, bdwidth, peaks, flank = NULL, permute = 5L,
  genome = "hg19", gctype = c("ladder", "tricube"))
```

Arguments

`coverage`	A list object returned by function `read5endCoverage`.
`gcbias`	A list object returned by function `gcEffects`.
`bdwidth`	A non-negative integer vector with two elements specifying ChIP-seq binding width and peak detection half window size. Usually generated by function `bindWidth`. A bad estimation of bdwidth results no meaning of downstream analysis. The values need to be the same as it is when calculating `gcbias`.
`peaks`	A GRanges object specifying the peaks to be refined. A flexible set of peaks are preferred to reduce potential false negative, meaning both significant (e.g. p<=0.05) and non-significant (e.g. p>0.05) peaks are preferred to be included. If the total number of peaks is not too big, a reasonable set of peaks include all those with p-value/FDR less than 0.99 by other peak callers.
`flank`	A non-negative integer specifying the flanking width of ChIP-seq binding. This parameter provides the flexibility that reads appear in flankings by decreased probabilities as increased distance from binding region. This paramter helps to define effective GC content calculation. Default is NULL, which means this paramater will be calculated from `bdwidth`. However, if customized numbers provided, there won't be recalucation for this parameter; instead, the 2nd elements of `bdwidth` will be recalculated based on `flank`. The value needs to be the same as it is when calculating `gcbias`.
`permute`	A non-negative integer specifying times of permutation to be performed. Default is 5. When whole large genome is used, such as human genome, 5 times of permutation could be enough.
`genome`	A BSgenome object containing the sequences of the reference genome that was used to align the reads, or the name of this reference genome specified in a way that is accepted by the `getBSgenome` function defined in the **BSgenome** software package. In that case the corresponding BSgenome data package needs to be already installed (see ?`getBSgenome` in the **BSgenome** package for the details). The value needs to be the same as it is when calculating `gcbias`.

gctype — A character vector specifying choice of method to calculate effective GC content. Default `ladder` is based on uniformed fragment distribution. A more smoother method based on tricube assumption is also allowed. However, tricube should be not used if estimated peak half size is 3 times or more larger than estimated bind width. The value needs to be the same as it is when calculating `gcbias`.

Value

A GRanges object the same as `peaks` with two additional meta columns:

newes — Refined enrichment scores.

newpv — Refined pvalues.

Examples

```
bam <- system.file("extdata", "chipseq.bam", package="gcapc")
cov <- read5endCoverage(bam)
bdw <- bindWidth(cov)
gcb <- gcEffects(cov, bdw, sampling = c(0.15,1))
peaks <- gcapcPeaks(cov, gcb, bdw)
refinePeaks(cov, gcb, bdw, peaks)
```

---

refineSites — *Adjust ChIP-seq Read Count Table*

Description

For a given set of sites with the same/comparable width, their read count table from multiple samples are adjusted based on potential GC effects. For each sample separately, GC effects are estimated based on their effective GC content and reads count using generalized linear mixture models. Then, count table is adjusted based on estimated GC effects. It it important that the given sites includes both foreground and background regions, see `sites` below.

Usage

```
refineSites(counts, sites, flank = 250L, outputidx = rep(TRUE,
  nrow(counts)), gcrange = c(0.3, 0.8), emtrace = TRUE, plot = TRUE,
  model = c("nbinom", "poisson"), mu0 = 1, mu1 = 50, theta0 = mu0,
  theta1 = mu1, p = 0.2, converge = 1e-04, genome = "hg19",
  gctype = c("ladder", "tricube"))
```

Arguments

counts — A count matrix with each row corresponding to each element in `sites` and each column corresponding to one sample. Every value in the matrix indicates the read counts for one site in one sample. It is noted that since effective GC content is used in this function, it is important to extend either original reads or

original `sites` to consider reads that 5' starting in `flank` regions, when counting sequencing reads.

`sites` A GRanges object with length equivalent to number of rows in `counts` matrix. It is preferable that every GRange have the same width; otherwise, the mixture model is modeling different things with wider GRanges certainly have more reads. However, it is OK if only a minority of GRanges have different width, since the model is pretty robust to outliers. Also, it is important that `sites` including both foreground and background regions in each sample, otherwise the mixture model will fail to fit two components. Fortunately, if you are inputing a large collection of samples, foreground sites in one sample may play the role as background in other samples. In this case, manually selecting real background is not necessary.

`flank` A non-negative integer specifying the flanking width of ChIP-seq binding. This parameter provides the flexibility that reads appear in flankings by decreased probabilities as increased distance from binding region. This paramter helps to define effective GC content calculation.

`outputidx` A logical vector with the length equivalent to number of rows in `counts`. This provides which subset of adjusted count matrix should be outputed. This would be extremely useful if you have manually collected background sites and want to only export the sites you care about.

`gcrange` A non-negative numeric vector with length 2. This vector sets the range of GC content to filter regions for GC effect estimation. For human, most regions have GC content between 0.3 and 0.8, which is set as the default. Other regions with GC content beyond this range will be ignored. This range is critical when very few foreground regions are selected for mixture model fitting, since outliers could drive the regression lines. Thus, if possible, first make a scatter plot between counts and GC content to decide this parameter. Alternatively, select a narrower range, e.g. c(0.35,0.7), to aviod outlier effects from both high and low GC-content regions.

`emtrace` A logical vector which, when TRUE (default), allows to print the trace of log likelihood changes in EM iterations.

`plot` A logical vector which, when TRUE (default), returns miture fitting plot.

`model` A character specifying the distribution model to be used in generalized linear model fitting. The default is negative binomial(`nbinom`), while `poisson` is also supported currently. More details see `gcEffects`.

`mu0` A non-negative numeric initiating read count signals for background sites. This is treated as the starting value of background mean for poisson/nbinom fitting.

`mu1` A non-negative numeric initiating read count signals for foreground sites. This is treated as the starting value of foreground mean for poisson/nbinom fitting.

`theta0` A non-negative numeric initiating the shape parameter of negative binomial model for background sites. For more detail, see theta in `glm.nb` function.

`theta1` A non-negative numeric initiating the shape parameter of negative binomial model for foreground sites. For more detail, see theta in `glm.nb` function.

`p` A non-negative numeric specifying the proportion of foreground sites in all estimated sites. This is treated as a starting value for EM algorithm.

`converge` A non-negative numeric specifying the condition of EM algorithm termination. EM algorithm stops when the ratio of log likelihood increment to whole log likelihood is less or equivalent to `converge`.

`genome` A BSgenome object containing the sequences of the reference genome that was used to align the reads, or the name of this reference genome specified in a way that is accepted by the `getBSgenome` function defined in the **BSgenome** software package. In that case the corresponding BSgenome data package needs to be already installed (see ?`getBSgenome` in the **BSgenome** package for the details).

`gctype` A character vector specifying choice of method to calculate effective GC content. Default `ladder` is based on uniformed fragment distribution. A more smoother method based on tricube assumption is also allowed. However, tricube should be not used if `flank` is too large.

Value

The count matrix after GC adjustment. The matrix values are not integer any more.

Index

bindWidth, 2
BSgenome, *4*, *6*, *9*, *12*

gcapcPeaks, 3
gcEffects, 4
getBSgenome, *4*, *6*, *9*, *12*
glm.nb, *6*, *11*

peaksCAT, 7

read5endCoverage, 8
refinePeaks, 9
refineSites, 10