Package: betterChromVAR (via r-universe)

October 22, 2026

Title Improved ChromVAR (Chromatin Variation Across Regions)

Description A much faster analytical implementation of chromVAR, with additional features, used to infer TF activity from (bulk or single-cell) ATAC-seq data and motif annotations (or binding probabilities). The package also includes the CVnorm normalization method based on the chromVAR logic.

Version 1.1.12

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

URL https://github.com/plger/betterChromVAR

BugReports https://github.com/plger/betterChromVAR/issues

biocViews Software, ATACSeq, Normalization, Epigenetics, Sequencing

Depends SummarizedExperiment

Imports BiocParallel, BiocNeighbors, Biostrings, DelayedMatrixStats, GenomicRanges, IRanges, Matrix, matrixStats, sparseMatrixStats, MatrixGenerics, methods, S4Vectors, stats

Suggests BiocStyle, knitr, rmarkdown, sessioninfo, testthat

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Config/pak/sysreqs zlib1g-dev

Repository https://bioc.r-universe.dev

Date/Publication 2026-10-19 09:23:34 UTC

RemoteUrl https://github.com/bioc/betterChromVAR

RemoteRef HEAD

RemoteSha d64e3b5d0f52fa449c248be5e59010c8055da9ec

Contents

addGCBias . . . 2
as.list,bcvBackground-method . . . 3
bcvBackground-class . . . 3
betterChromVAR . . . 4
computeBackgrounds . . . 5
computeDeviationsAnalytic . . . 7
computeDeviationsFromKNN . . . 8
computeDeviationsWeighted . . . 9
computeMotifReliability . . . 11
computeMotifVariability . . . 12
CVnorm . . . 13
estimateNoiseKappa . . . 15
getBackgroundBins . . . 17
getBackgroundKNN . . . 18
getDummyData . . . 19
getExpectation . . . 20
normalizeDevsForSize . . . 21
sampleBackgroundPeaks . . . 21
shrinkColumnProps . . . 22
smoothDeviationsKNN . . . 23

Index **26**

`addGCBias` *addGCBias*

Description

Add the `bias` column to the object's rowData, containing the regions' proportion of Gs and Cs.

Usage

```
addGCBias(object, genome)
```

Arguments

`object`	An object inheriting RangedSummarizedExperiment or GRanges.
`genome`	A BSgenome object or any other genome object supported by `getSeq`.

Value

`object` with the GC content in `mcols(object)$bias` (if GRanges) or `rowData(object)$bias`.

Examples

```
# not run:
# se <- addGCBias(se, genome)
```

as.list,bcvBackground-method *Coerce bcvBackground to a list*

Description

Coerce bcvBackground to a list

Show a bcvBackground object

Subsetting a bcvBackground

Usage

```
## S4 method for signature 'bcvBackground'
as.list(x)

## S4 method for signature 'bcvBackground'
show(object)

## S4 method for signature 'bcvBackground,ANY,ANY,ANY'
x[i, j, ..., drop = TRUE]
```

Arguments

x	A bcvBackground object.
object	A bcvBackground object.
i, j	Indices for subsetting (if j is provided, i is ignored).
...	Additional arguments.
drop	Logical, whether to drop dimensions.

Value

A list containing the slots of the object.

Nothing, prints an overview of the object.

An bcvBackground object.

bcvBackground-class *Bin and background data for betterChromVAR (for internal use)*

Description

Bin and background data for betterChromVAR (for internal use)

betterChromVAR *betterChromVAR*

Description

A fast, analytic implementation of chromVAR. This is a wrapper around the getBackgroundBins, computeBackgrounds, and computeDeviationsAnalytic steps. It additionally allows for multi-threading. For more control or optimization, see the individual steps.

Usage

```
betterChromVAR(
  object,
  annotations,
  grouping = NULL,
  nthreads = NULL,
  verbose = FALSE,
  fpc = TRUE,
  ...
)
```

Arguments

object	A SummarizedExperiment (or SingleCellExperiment) with an assay 'counts', and with a 'bias' column in rowData(object). Note that the regions should have similar widths.
annotations	Peak annotation (sparse) matrix, with motifs as columns, or a SummarizedExperiment containing this in the first assay. Values should be either logical or between 0 and 1.
grouping	An optional factor or vector coercible to a factor indicating the groupings of the columns of object. This is optionally used to compute the base expectation such that rare cell types are given as much weight as abundant ones. In single-cell data, the grouping can for instance be the interaction of samples and cell types. This should either be a vector coercible to factor of length equal to ncol(object), or a character of length 1 specifying a column of colData(object).
nthreads	Either an integer scalar indicating the number of threads to use, or a BiocParallelParam object.
verbose	Logical; whether to output progress messages (default FALSE).
fpc	Logical; whether to apply the finite-population correction to the null variance used for the z-scores (default TRUE, see computeDeviationsAnalytic).
...	Passed to getBackgroundBins.

Details

Contrarily to the original chromVAR, this function is entirely deterministic, and achieves higher precision and much higher efficiency through two changes:

1. working with expected background sampling mean and variances, rather than actual permutations, and 2) computing expectations and variance at the level of bias bins, instead of in the peak-space. The function additionally includes experimental bias shrinkage options, the possibility to handle annotations that are not binary (e.g. probability scores) and a third bias dimension (fragment length bias, which should be stored in `rowData(object)$flbias` see `getBackgroundBins` for details).

Value

A SummarizedExperiment containing the adjusted deviations and z-scores for each motif/sample. The rowData additionally contains the number of motif matches and their variability.

Author(s)

Pierre-Luc Germain

References

Schep A.N., Wu B., Buenrostro J.D., Greenleaf W.J. (2017) chromVAR: inferring transcription-factor-associated accessibility from single-cell epigenomic data, Nature Methods, doi: 10.1038/nmeth.4401

Examples

```
attach(getDummyData())
# if GC content not already in the object, use:
# counts <- addGCBias(counts, genome=YOUR_GENOME)
dev <- betterChromVAR(counts, motifMatches)
dev
# note that this is the exact equivalent of doing:
# bg <- getBackgroundBins(counts)
# bg <- computeBackgrounds(counts, bg)
# dev <- computeDeviationsAnalytic(counts, bg, motifMatches)
```

`computeBackgrounds` *computeBackgrounds*

Description

computeBackgrounds

Usage

```
computeBackgrounds(
  object,
  bins,
  grouping = NULL,
  expectation = NULL,
  shrinkage = c("none", "average", "smooth"),
  sigma = 1,
  verbose = FALSE
)
```

Arguments

object	A SummarizedExperiment (or SingleCellExperiment) with an assay 'counts', or a (sparse) matrix of counts.
bins	A bcvBackground object, as produced by getBackgroundBins.
grouping	An optional factor or vector coercible to a factor indicating the groupings of the columns of object. This is optionally used to 1) compute the base expectation such that rare cell types are given as much weight as abundant ones, and 2) apply shrinkage (if shrinkage!="none") on a per-grouping fashion. In single-cell data, the grouping can for instance be the interaction of samples and cell types. (The name of a colData column of object can also be provided.)
expectation	Optional vector of length equal to nrow(object) giving the expected counts. If NULL, defaults to mean counts (eventually grouped, see grouping and getExpectation).
shrinkage	The method to use to shrink background (i.e. bias) bin frequencies. Either "average" (shrinks towards the bin's average across cells/samples of the same group), "smooth" (per-sample 2D smoothing over the bin matrix, somewhat redundant with w), or "none" (default).
sigma	Sigma parameter for the 2D smoothing. Ignored unless shrinkage="smooth".
verbose	Logical; whether to output progress messages.

Value

A bcvBackground object with bins*samples slots filled, for use with computeDeviationsAnalytic.

Examples

```
attach(getDummyData())
# if GC content not already in the object, use:
# counts <- addGCBias(counts, genome=YOUR_GENOME)

# we fist get the background bins:
bg <- getBackgroundBins(counts)
# then we can compute the backgrounds for each sample:
bg <- computeBackgrounds(counts, bg)
# for use in computeDeviationsAnalytic...
```

computeDeviationsAnalytic
computeDeviationsAnalytic

Description

computeDeviationsAnalytic

Usage

```
computeDeviationsAnalytic(
  object,
  background,
  annotations,
  verbose = FALSE,
  retSE = TRUE,
  compute = c("deviations", "z", "variability"),
  denominator = c("global", "local", "none"),
  fpc = TRUE
)
```

Arguments

object	A SummarizedExperiment (or SingleCellExperiment) with an assay 'counts', and with a 'bias' column in `rowData(object)`. Note that the regions should have similar widths.
background	A bcvBackground object with bins*samples slots filled, as produced by `computeBackgrounds`.
annotations	Peak annotation (sparse) matrix, with motifs as columns, or a SummarizedExperiment containing this in the first assay. Values should be either logical or between 0 and 1.
verbose	Logical; whether to output progress messages.
retSE	Logical; whether to return a SummarizedExperiment object.
compute	What to compute. Defaults to everything: deviations, z and motif variability.
denominator	The type of denominator to use for the deviations. Either 'global' (default), i.e. the global expectation (same as the original chromVAR), 'local' (background expectation of the cell/sample), or 'none' (denominator of 1). 'global' (default) is recommended.
fpc	Logical, whether to apply finite population correction (see details below). Disable (`fpc=FALSE`) to reproduce the behavior of the original chromVAR.

Details

`fpc` corrects correct the null variance used (as denominator) for the z-scores for the fact that the expectation of each cell is estimated from the cell's own counts, including in peaks containing the

motif. This correlation between observed and expected counts reduces the variance of their difference in a way that depends on the proportion of peaks matching the motif. This is negligilbe for motifs matching few peaks, and especially affects prevalent motifs concentrated in a few background bins (e.g. GC-rich motifs). The correction typically has little effect on cell ranking within a motif.

Value

A SummarizedExperiment (or a list if `retSE=FALSE`).

Examples

```
attach(getDummyData())
# if GC content not already in the object, use:
# counts <- addGCBias(counts, genome=YOUR_GENOME)

# we fist get the background bins:
bg <- getBackgroundBins(counts)
# then we compute the backgrounds for each sample:
bg <- computeBackgrounds(counts, bg)
# then we can compute the deviations:
dev <- computeDeviationsAnalytic(counts, bg, motifMatches)
dev
```

---

computeDeviationsFromKNN

computeDeviationsFromKNNbg

Description

Computes analytical deviations using a nearest-neighbor background matrix while optionally excluding peaks containing the target motif from its own background pool.

Usage

```
computeDeviationsFromKNN(
  object,
  cBg,
  annotations,
  l = 1,
  chunkSize = 1000,
  fpc = TRUE,
  verbose = TRUE
)
```

Arguments

object	A SummarizedExperiment or sparse matrix of counts.
cBg	The peak-by-peak sparse kNN matrix, as produced by getBackgroundKNN.
annotations	Peak annotation (sparse) matrix, with motifs as columns, or a SummarizedExperiment containing this in the first assay. Values should be either logical or between 0 and 1.
l	Lambda parameter determining the weight by which background peaks containing the foreground motif are scaled in relative importance. Set to 1 to treat them normally (default), to 0 to exclude them entirely (potentially unstable, a small value such as 0.1 is instead recommended).
chunkSize	Number of cells to process simultaneously. Increasing this will increase speed, but also memory consumption.
fpc	Logical; whether to correct the null variance used for the z-scores for the fact that the expectation of each cell is estimated from the cell's own counts in the neighbors of the motif's peaks, some of which are themselves peaks of the motif (default TRUE, see Details and computeDeviationsAnalytic).
verbose	Logical; whether to print progress messages.

Details

This method combines the analytic strategy used by betterChromVAR with Ruochi Zhang's approach to use a continuous, multidimensional background space instead of background bins. If l<1 it downweighs (multiplying them by l) peaks harboring the tested motif from the corresponding motif's background.

Value

A SummarizedExperiment with 'deviations' and 'z' assays. If overall motif variability and their significance are additionally needed, see computeMotifVariability.

Examples

```
attach(getDummyData())
bg <- getBackgroundKNN(counts)
dev <- computeDeviationsFromKNN(object=counts, cBg=bg,
                                annotations=motifMatches)
dev
```

---

computeDeviationsWeighted

computeDeviationsWeighted

Description

A variant of computeDeviationsAnalytic enabling the computation of deviations from weighted foreground counts. Specifically, this functions handles the normalization of the difference in magnitude between the (weighted) foreground and background.

Usage

```
computeDeviationsWeighted(
  weightedMotifCounts,
  unweightedPeakCounts,
  annotations,
  bg = NULL,
  retSE = TRUE,
  fpc = TRUE,
  ...
)
```

Arguments

weightedMotifCounts	A matrix of weighted counts per motif (rows) and sample (columns), or a SummarizedExperiment containing this as first assay.
unweightedPeakCounts	A matrix of unweighted counts per peak (rows) and sample (columns), or a SummarizedExperiment containing this as first assay.
annotations	Peak annotation (sparse) matrix, with motifs as columns, or a SummarizedExperiment containing this in the first assay. Values should be either logical or between 0 and 1.
bg	Either a bcvBackground-class object as produced by computeBackgrounds, or a SummarizedExperiment of background peak counts (with bias data in rowData). If missing, will be created based on unweightedPeakCounts.
retSE	Logical; whether to return a SummarizedExperiment object.
fpc	Logical; whether to apply the finite-population correction to the null variance used for the z-scores (see computeDeviationsAnalytic).
...	Passed to getBackgroundBins (can for instance be used to pass bias info if not contained in the objects). Ignored if bg is a bcvBackground-class object.

Details

The background expectation of the (unweighted) peak counts is rescaled, for each motif, by the ratio between the mean weighted and mean unweighted counts of the motif, and its variance by the square of that ratio. This assumes that the weights are similar across the peaks of a motif: if the weights vary a lot between peaks (coefficient of variation CV), the variance of the weighted counts is larger, by a factor ~1+CV^2, than the variance used here, and the z-scores are overestimated accordingly. The finite-population correction (fpc) also assumes that the weighted counts are proportional to the unweighted counts of the same peaks, from which the expectation is estimated. Neither the z-scores nor the deviations depend on the overall magnitude of the weights.

Value

A SummarizedExperiment (or a list if retSE=FALSE). If overall motif variability and their significance are additionally needed, see computeMotifVariability.

See Also

computeDeviationsAnalytic()

Examples

```
attach(getDummyData())
# if GC content not already in the object, use:
# counts <- addGCBias(counts, genome=YOUR_GENOME)
# For the purpose of this example, we'll use standard (unweighted counts),
# although at this step we'd compute counts weighted in the desired fashion:
motifCounts <- Matrix::t(motifMatches) %*% assay(counts)
dev1 <- computeDeviationsWeighted(motifCounts, counts, motifMatches)
dev1
# in this case, the results are identical to :
dev2 <- betterChromVAR(counts, motifMatches)
stopifnot(identical(assays(dev1), assays(dev2)))
```

---

computeMotifReliability

computeMotifReliability

Description

Estimates the reliability of the single-cell deviations of each motif, i.e. the proportion of their variance across cells that is not noise, from the analytic noise variance of the deviations.

Usage

```
computeMotifReliability(
  dev,
  kappa = 1,
  cellWeights = TRUE,
  nBlocks = 20L,
  chunkSize = 250L
)
```

Arguments

dev	A SummarizedExperiment as produced by betterChromVAR or computeDeviationsAnalytic, with the assays 'deviations' and 'z'.
kappa	A scalar, a numeric vector (one value per motif), or the output of estimateNoiseKappa, by which the analytic noise variance of the deviations is multiplied, to calibrate (see estimateNoiseKappa). This should not be necessary anymore, and therefore defaults to 1 (no calibration).
cellWeights	Logical; whether to weigh the cells by the precision of their deviations (default TRUE), preventing noisy (e.g. low libsize) cells from dominating the estimates.

nBlocks	The number of blocks of cells used for the (jackknife) estimation of the standard error of the reliability.
chunkSize	The number of motifs processed at a time (this affects only memory usage).

Details

Motifs with a reliability close to 0 carry no cell-level information in the given dataset (and could be discarded for the purpose of analyses at the single-cell level). The p-value (reliability.pval) is that of a one-sided test of the null hypothesis that the deviations of the motif only vary across cells because of noise, i.e. that the reliability is zero. If this is the case, the deviations standardized by their noise standard deviation (and centered on their precision-weighted mean) have a variance of 1. We test whether the observed variance is larger, using the kurtosis-adjusted standard error of the log-variance of Bonett (2006), as in computeMotifVariability (the test does not depend on cellWeights). As it relies on the noise variance, it is sensitive to its calibration: with many cells, even a small underestimation of the noise variance leads to significant p-values for motifs with a negligible reliability, so that the p-value should be interpreted together with the reliability. Adjusted p-values use the Benjamini-Hochberg method.

The estimate is based on the accuracy of the analytic noise variance, which in our hands is excellent. Alternatively, the estimateNoiseKappa function estimates it empirically, but it is slower and based on a subset of the cells.

Value

A data.frame with one row per motif, and the columns reliability, reliability.se, reliability.pval and reliability.adjPval.

Examples

```
attach(getDummyData())
dev <- betterChromVAR(counts, motifMatches)
computeMotifReliability(dev, nBlocks=5)
```

---

computeMotifVariability

computeMotifVariability

Description

computeMotifVariability

Usage

```
computeMotifVariability(
  z,
  confInt = 0.95,
  n = 100,
  method = c("bonett", "normal", "bootstrap")
)
```

Arguments

z	A matrix of z-scores (with motifs as rows), or a SummaziedExperiment object with such a matrix in assay z (as produced e.g. by computeDeviationsAnalytic).
confInt	The the confidence interval (a numeric scalar between 0 and 1, default 0.95).
n	The number of bootstrap samples. Ignored unless method="bootstrap".
method	The method used to compute the confidence interval. 'normal' computes it analytically, assuming that the z-scores are normally distributed. 'bonett' (default) adjusts this analytic estimate for kurtosis (based on Bonett, Computational Statistics & Data Analysis, 2006). 'bootstrap' uses bootstrapping, which does not scale very well.

Value

A data.frame containing the variability and confidence interval around it, as well as significance, for each motif. If z is a SummarizedExperiment, the data.frame will be stored in rowData(z).

See Also

computeMotifReliability.

Examples

```
# we generate random z-scores:
z <- matrix(rnorm(mean=rnorm(10), sd=runif(10, max=2), 200), nrow=10)
var <- computeMotifVariability
```

CVnorm *CVnorm: chromVAR-inspired ATAC-seq normalization*

Description

Corrects ATAC peak counts by removing the effects of technical biases (GC/accessibility) using the chromVAR background binning approach and an optional variance-based bias shrinkage (inspired from the qsmooth package) to preserve group biological signal.

Usage

```
CVnorm(
  object,
  bias = NULL,
  grouping = NULL,
  smoothGrouping = grouping,
  shrinkMode = c("dampen", "qsmooth"),
  toAssay = "corrected",
  bs = NULL,
  w = 0.1,
```

```
  Z = FALSE,
  useWidthAdj = NULL,
  enforceZeros = TRUE
)
```

Arguments

object	A matrix of counts, or a SummarizedExperiment-like object with an assay named 'counts'.
bias	A vector of length equal to ncol(object) specifying the per-peak bias (i.e. GC content). If omitted, will try to get it from rowData(object)$bias.
grouping	Optional grouping for the baseline expectation (prevents bias toward more abundant groups). This should either be a vector coercible to factor of length equal to ncol(object), of a character of length 1 specifying a column of colData(object) (if object is a SummarizedExperiment).
smoothGrouping	Optional grouping to determine correction strength. If bias is consistent within these groups, correction is reduced. Accepts the same type of inputs as grouping, and by default takes the same values.
shrinkMode	The way to perform the group-based shrinkage. With shrinkMode="dampen" (default), no corrected is applied in bins when the bias is entirely explained by groups. shrinkMode="qsmooth" instead reproduces the logic of the qsmooth package: if variance in bias is chiefly explained by groups, between group bias will not be corrected, but within-group differences will be. Using this prior to differential analysis however leads to increase Type I error rate, and it should therefore not be used for downstream application.
toAssay	The name of the assay in which to store the corrected data (default 'corrected'). Ignored unless object is a SummarizedExperiment-like object.
bs	Number of bins per dimension (see getBackgroundBins).
w	Standard deviation of the Gaussian kernel for bin smoothing.
Z	Logical; whether to return standardized residuals (Z-scores) instead of the (default) corrected counts.
useWidthAdj	Whether to adjust for the different width of the regions. If omitted, will be TRUE if the average absolute difference to the median width is greater than 10% of the median width. If TRUE, will adjust for pmax(200L,width). If an integer scalar, will adjust for pmax(useWidthAdj,width).
enforceZeros	Logical; whether to enforce that zero counts should remain zeroes after correction (ignored if Z=TRUE).

Value

If object is a matrix, then a matrix of corrected counts of the same dimensions. If object is a SummarizedExperiment-like object, then the object is returned with an extra assay named based on toAssay.

Author(s)

Pierre-Luc Germain

References

- Schep A.N., Wu B., Buenrostro J.D., Greenleaf W.J. (2017) chromVAR: inferring transcription-factor-associated accessibility from single-cell epigenomic data, Nature Methods, doi: 10.1038/nmeth.4401
- Hicks SC, Okrah K, Paulson JN, Quackenbush J, Irizarry RA, Corrado Bravo H (2018). "Smooth quantile normalization." Biostatistics 19 (2), doi: 10.1093/biostatistics/kxx028

Examples

```
counts_se <- getDummyData()$counts
# if GC content not already in the object, use:
# counts_se <- addGCBias(counts_se, genome=YOUR_GENOME)
counts_se <- CVnorm(counts_se)
```

`estimateNoiseKappa` *estimateNoiseKappa*

Description

Empirically calibrates, for each motif, the analytic estimate of the noise variance of the single-cell deviations, and estimates the reliability of the single-cell deviations of each motif. This is mostly for development and testing: for practical purposes the analytic estimates are sufficient.

Usage

```
estimateNoiseKappa(
  object,
  annotations,
  expectation = NULL,
  background = NULL,
  grouping = NULL,
  ncells = 3000L,
  pairs = NULL,
  fpc = TRUE,
  nthreads = NULL,
  chunkSize = 500L,
  cellWeights = TRUE,
  nBlocks = 20L,
  verbose = FALSE,
  ...
)
```

Arguments

`object`	A SummarizedExperiment (or SingleCellExperiment) with an assay 'counts', and with a 'bias' column in `rowData(object)` (as for `betterChromVAR`), or a (sparse) matrix of counts, in which case `background` is required.

annotations	Peak annotation (sparse) matrix, with motifs as columns, or a SummarizedExperiment containing this in the first assay.
expectation	Optional vector of expected counts per peak (as used for the computation of the deviations). If NULL, taken from background if available, otherwise computed from object using getExpectation (with grouping).
background	Optional bcvBackground object as produced by getBackgroundBins. If NULL, created from the expectation and the bias in rowData(object).
grouping	Optional grouping of the cells, used to compute the expectation (see betterChromVAR); ignored if expectation is given.
ncells	The (maximum) number of randomly-selected cells to use (default 3000). More cells increase the precision of the estimates (and the running time).
pairs	Logical; whether counts are Tn5 insertions (e.g. ArchR peak matrices), in which case a fragment with both ends in a peak contributes 2 and the read-splitting needs to keep such pairs together. If NULL (default), this is guessed from the data (pairs are assumed if more than half of all counts are in entries with an even value).
fpc	Logical; whether the finite-population correction was applied to the deviations whose noise is to be calibrated (see computeDeviationsAnalytic). This should match the setting used for the deviations passed to smoothDeviationsKNN.
nthreads	Either an integer scalar indicating the number of threads to use, or a BiocParallelParam object (default NULL, i.e. no parallelization). The cells are processed in chunks of chunkSize.
chunkSize	Number of cells processed at a time (per worker). Lower values reduce memory usage.
cellWeights	Logical; whether to weigh the cells by the precision of their deviations (default TRUE) rather than equally. Precision weighting prevents the (many) shallow, very noisy cells from dominating the estimates.
nBlocks	The number of blocks of cells used for the (jackknife) estimation of the standard error of the reliability.
verbose	Logical; whether to output progress messages.
...	Passed to getBackgroundBins (when background is NULL).

Details

The variance of the deviations computed analytically overestimates the noise for some motifs. This function first computes the proportion of the variance that is due to noise in each motif, using the difference between the deviations computed from two independent halves of the reads/fragments. This provides two pieces of information: first, it allows estimating, for each motif, the reliability of its single-cell deviations, understood as the proportion of their variance across cells that is not noise. Second, it enables computing a factor, kappa, that calibrates the analytic variance computed from the deviations (a value close to 1 means that the analytic variance is accurate).

The selection of the cells and the splitting of their reads are random: use set.seed before calling the function for reproducible results (if you provide a BiocParallelParam as nthreads, set its RNGseed).

Value

A data.frame with one row per motif and the columns kappa (to be given to smoothDeviationsKNN), reliability and reliability.se (a jackknife estimate of the standard error of the variability).

See Also

smoothDeviationsKNN

Examples

```
attach(getDummyData())
res <- estimateNoiseKappa(counts, motifMatches, pairs=FALSE)
res
```

---

getBackgroundBins *getBackgroundBins*

Description

Computes chromVAR-like background (i.e. bias) bins, as well as bin-to-bin selection probabilities needed for betterChromVAR.

Usage

```
getBackgroundBins(
  x,
  bias = NULL,
  flbias = NULL,
  w = 0.1,
  bs = NULL,
  pseudo = 0,
  verbose = TRUE
)
```

Arguments

x	A SummarizedExperiment containing a 'counts' assay, or a matrix of counts, or a vector of expected (e.g. mean) counts.
bias	A vector of length equal to ncol(object) specifying the per-peak bias (i.e. GC content). If omitted, will try to get it from rowData(x)$bias.
flbias	A vector of length equal to ncol(object) specifying the per-peak fragment length bias (e.g. log10-transformed median length of fragments overlapping each region). If omitted, will try to get it from rowData(object)$flbias. If absent, will use 2-dimensional bias bins (which is already pretty good).
w	Standard deviation of the Gaussian kernel.

bs	Number of bins per dimension. This can be a single integer (total bins = bs^2), or an integer vector of length 2 (if flbias=NULL) or 3 (in which case there are prod(bs) total bins). The values specify the number of bins for, in order: enrichment, GC and fragment length. By default, bs=50 if flbias is not provided (mimicking chromVAR), and bs=c(30, 30, 6) if it is.
pseudo	Optional pseudocount to be added. This should not be needed with standard workflows.
verbose	Whether to print processing info.

Details

The procedure underlying this function is the same as in chromVAR::getBackgroundPeaks, with the following differences:

- Rather than producing a set of background peaks for each input peak, the function returns peak-to-bin mappings and bin-to-bin background selection probabilities, which enables an analytic background computation. It is, as such, entirely deterministic.
- The function supports the optional use of a third bias dimension, provided through the flbias argument, meant for fragment length bias. This is still an experimental feature.

Value

A bcvBackground object, to be used with computeBackgrounds.

References

Schep A.N., Wu B., Buenrostro J.D., Greenleaf W.J. (2017) chromVAR: inferring transcription-factor-associated accessibility from single-cell epigenomic data, Nature Methods, doi: 10.1038/nmeth.4401

Examples

```
counts_se <- getDummyData()$counts
background <- getBackgroundBins(counts_se)
```

---

getBackgroundKNN *getBackgroundKNN*

Description

Computes k-nearest neighbors for each peaks based on the continuous, multidimensional bias space. This is inpsired by Ruochi Zhang's approach in scPrinter.

Usage

```
getBackgroundKNN(
  se,
  expectation = NULL,
  bias = NULL,
  k = 50,
  weights = c("linear", "poly", "none"),
  pseudo = 0.1,
  ...
)
```

Arguments

se	A SummarizedExperiment containing a 'counts' assay, or a matrix of counts.
expectation	A vector of expectations. If NULL, will use the mean counts of se
bias	A data.frame of sources of bias (beside expectation) to consider (one per column), with the same nrow as se.
k	Number of nearest neighbors to use.
weights	How to weigh the different bias dimensions. If "none", they will not be re-weighted. If 'linear' (default), they are weighted by the absolute Pearson correlation with the over-dispersion. If "poly", by the R^2 of a 2nd degree polynomial fit of the over-dispersion.
pseudo	Pseudocount for log transformation.
...	Passed to findKNN.

Value

A sparse peak-by-peak kNN matrix.

Examples

```
SE <- getDummyData()$counts
bg <- getBackgroundKNN(SE)
```

---

getDummyData *Dummy data for testing purposes*

Description

Dummy data for testing purposes

Usage

```
getDummyData(nRegions = 500, nSamples = 10, nMotifs = 5)
```

Arguments

nRegions	Number of regions to generate
nSamples	Number of samples to generate
nMotifs	Number of motifs to generate

Value

A list with the slots counts (a peak counts SummarizedExperiment) and matches (a sparse matrix of binary motif matches per peaks)

Examples

```
out <- getDummyData()
(counts <- out$counts)
matches <- out$motifMatches
```

---

getExpectation *getExpectation*

Description

Computes expected counts (a glorified rowMeans)

Usage

```
getExpectation(counts, grouping = NULL, normalize = TRUE)
```

Arguments

counts	A count matrix, or object inheriting SummarizedExperiment with a 'counts' assay.
grouping	An optional vector of length equal to ncol(counts), indicating the grouping of the cells. If provided, cells will be averaged by group before averaging across groups.
normalize	Logical; whether to normalize data between averaging (but after grouping). Default TRUE and highly recommended if providing grouping.

Value

A vector of expectation for each row of counts

Examples

```
attach(getDummyData())
e <- getExpectation(counts)
```

normalizeDevsForSize *normalizeDevsForSize*

Description

Normalizes the z-scores assay of a deviations object to make the scores comparable across motifs with different number of matches.

Usage

```
normalizeDevsForSize(dev)
```

Arguments

dev	A SummarizedExperiment object as produced by betterChromVAR or computeDeviationsAnalytic.

Details

If a TF changes the accessibility of all its motif matches by the same relative amount, the z-score of its motif increases with the square root of the effective number of matches, Neff (see the rowData of the object). Neff is smaller than the number of matches N for motifs matching a large share of the peaks in the bins in which they are found, since the expectation of the cells is estimated from the same peaks. The z-scores are multiplied by sqrt(median(Neff)/Neff). For objects without Neff (e.g. produced by earlier versions) N is used instead.

Value

The dev object with an additional assay named 'norm'.

Examples

```
attach(getDummyData())
dev <- betterChromVAR(counts, motifMatches)
dev <- normalizeDevsForSize(dev)
dev
```

sampleBackgroundPeaks *sampleBackgroundPeaks*

Description

Given a background generated by getBackgroundBins, samples background peaks for each input peak.

Usage

```
sampleBackgroundPeaks(background, niterations = 50)
```

Arguments

background	A bcvBackground produced by getBackgroundBins.
niterations	Number of background peaks to sample for each target peak.

Details

This function is not used by betterChromVAR, which is deterministic, but for other applications requiring an outputs similar to that of the original getBackgroundPeaks.

Value

A peaks x niterations matrix of integers representing the indices of the sampled background peaks.

Examples

```
counts_se <- getDummyData()$counts
background <- getBackgroundBins(counts_se)
bg_peaks <- sampleBackgroundPeaks(background, niterations=20)
```

---

shrinkColumnProps *shrinkColumnProps*

Description

Empirical Bayes shrinkage of a matrix of counts towards a prior proportion (by default the mean across columns).

Usage

```
shrinkColumnProps(x, shrinkTo = NULL, var.theo = FALSE)
```

Arguments

x	A matrix of counts, with features as rows and samples as columns.
shrinkTo	A vector (of length equal to nrow(x)) of sampling probabilities to shrink towards, or a matrix (of the same dimensions as x) of such probabilities. If omitted, will be the weighted mean of the columns' relative frequencies.
var.theo	Logical; whether to use theoretical (i.e. binomial) variances of the proportions, rather than the observed (weighted) variance.

Value

A matrix of the same dimensions as x representing the shrunk column-wise proportions.

Examples

```
# generate a matrix of 5 sampling (with different total counts) of 20
# features based on the same base frequency :
baseFreq <- abs(rnorm(20))
baseFreq <- baseFreq/sum(baseFreq)
mat <- sapply(c(10,20,30,40,50), function(tot){
  rpois(length(baseFreq), baseFreq*tot)
})
# apply shrinkage and confirm that shrunk proportions are better correlated
shrunk_mat <- shrinkColumnProps(mat)
mean(cor(shrunk_mat))>mean(cor(mat))
```

`smoothDeviationsKNN` *smoothDeviationsKNN*

Description

Adaptive smoothing of single-cell motif deviations across nearest-neighbors.

Usage

```
smoothDeviationsKNN(
  dev,
  knn,
  k = NULL,
  kappa = 1,
  tauScale = 3,
  retWeights = FALSE,
  blockSize = 250L,
  verbose = FALSE
)
```

Arguments

dev	A SummarizedExperiment as produced by `betterChromVAR` or `computeDeviationsAnalytic`, with the assays 'deviations' and 'z'.
knn	The k-nearest neighbors (kNN) of the cells, in the format returned by `findKNN`, i.e. a list with an `index` element (an integer matrix with one row per cell, in the same order as the columns of `dev`, and one column per neighbor, ordered by distance). The index matrix can also be given directly. The cell itself should not be included in its neighbors (any self-neighbor will be ignored). The kNN should be computed from an embedding that is independent of the motif deviations (e.g. an embedding of the peak counts).
k	The number of neighbors to use. If NULL (default), `min(50, ncol(knn$index))` is used.

kappa	A scalar, a numeric vector (one value per motif), or the output of estimateNoiseKappa, by which the analytic noise variance of the deviations is multiplied, to calibrate (see estimateNoiseKappa). This should not be necessary anymore, and therefore defaults to 1 (no calibration).
tauScale	A multiplier applied to the estimated biological variance. Higher values mean less smoothing (default 3).
retWeights	Logical; whether to additionally return, as assay 'own.weight', the weight attributed to each cell's own value (0 to 1).
blockSize	The number of motifs processed at a time (this affects only memory usage).
verbose	Logical; whether to output progress messages.

Details

The deviation of each cell is shrunk towards the mean deviation of its k nearest neighbors, with the weight kept on the cell's own value being

$$\lambda = \frac{\tau^2 + v}{\tau^2 + v + \sigma^2}$$

where σ^2 is the noise variance of the cell's own deviation (obtained from deviations/z, and multiplied by kappa), v the sampling variance of the neighbors' mean (the mean being weighted by the precision of each neighbor's deviation, i.e. cells with more information count more), and τ^2 the biological variance, specific to each motif, of the cells' activity around the mean of their neighborhood, which is estimated across all cells (method of moments). Cells with more information (i.e. smaller noise variance, such as the cells with the most fragments) and motifs that show real cell-to-cell variation are therefore smoothed less. Motifs with no detectable variation between neighboring cells are smoothed strongly.

The shrinkage assumes that neighboring cells have a similar motif activity. Effects that are specific to individual cells, and small relative to the noise, cannot be distinguished from the noise and therefore risk being smoothed out; tauScale can be increased to reduce this at the expense of less noise reduction.

Value

dev with the additional assays 'smoothed' (the smoothed deviations), 'smoothed.z' (the smoothed deviations divided by their standard deviation under the null hypothesis that the deviation is zero in the cell and its neighbors, using the calibrated noise variance), and optionally 'own.weight'. The tau2 (variance of the motif's activity around the mean of neighboring cells, on the scale of the deviations) is added to the rowData.

See Also

estimateNoiseKappa

Examples

```
attach(getDummyData())
dev <- betterChromVAR(counts, motifMatches)
# The kNN should be based on an embedding that is independent of the
```

```
# deviations, for instance a PCA. For simplicity, for this example we use
# the PCA on raw counts:
emb <- prcomp(t(log1p(assay(counts))))$x[,1:3]
knn <- BiocNeighbors::findKNN(emb, k=3)
dev <- smoothDeviationsKNN(dev, knn)
dev
```

Index

[,bcvBackground,ANY,ANY,ANY-method
(as.list,bcvBackground-method), 3

addGCBias, 2
as.list,bcvBackground-method, 3

bcvBackground-class, 3
betterChromVAR, 4, *11*, *15–17*, *21–23*

computeBackgrounds, *4*, 5, *7*, *10*, *18*
computeDeviationsAnalytic, *4*, *6*, 7, *9–11*, *13*, *16*, *21*, *23*
computeDeviationsAnalytic(), *11*
computeDeviationsFromKNN, 8
computeDeviationsWeighted, 9
computeMotifReliability, 11, *13*
computeMotifVariability, *9*, *10*, 12, *12*
CVnorm, 13

estimateNoiseKappa, *11*, *12*, 15, *24*

findKNN, *19*, *23*

getBackgroundBins, *4–6*, *10*, *14*, *16*, 17, *21*, *22*
getBackgroundKNN, *9*, 18
getBackgroundPeaks, *22*
getDummyData, 19
getExpectation, *6*, *16*, 20
getSeq, *2*

normalizeDevsForSize, 21

sampleBackgroundPeaks, 21
show,bcvBackground-method
(as.list,bcvBackground-method), 3
shrinkColumnProps, 22
smoothDeviationsKNN, *16*, *17*, 23