Package 'scPloidy'

April 29, 2024

Type Package

Title Infer Ploidy of Single Cells

Version 0.6.2

Description Compute ploidy of single cells (or nuclei)
based on single-cell (or single-nucleus)
ATAC-seq (Assay for Transposase-Accessible Chromatin using sequencing)
data <https://github.com/fumi-github/scPloidy>.

BugReports https://github.com/fumi-github/scPloidy/issues

Depends R (>= 3.5.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Imports dplyr, GenomicRanges, magrittr, MASS, matrixStats, mixtools,
nimble, rlang, Rsamtools, tibble, tidyr, utils

Suggests gplots, IRanges, knitr, readr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Fumihiko Takeuchi [aut, cre] (<https://orcid.org/0000-0003-3185-5661>)

Maintainer Fumihiko Takeuchi <fumihiko@takeuchi.name>

Repository CRAN

Date/Publication 2024-04-28 22:20:02 UTC

R topics documented:

---

cnv *Infer Copy Number Variations (CNVs) in Cancer Cells from ATAC-seq Fragment Overlap*

Description

Infer Copy Number Variations (CNVs) in Cancer Cells from ATAC-seq Fragment Overlap

Usage

```
cnv(
  fragmentoverlap,
  windowcovariates,
  levels = c(2, 4),
  nfragspercellmin = 5000,
  nfragspercellmax = 10^5.5,
  deltaBICthreshold = 0
)
```

Arguments

`fragmentoverlap`
: Frequency of fragment overlap in each cell-window computed by the function `fragmentoverlapcount`. `barcode` should be named as `AAACGAAAGATTGACA-1.window_1`, which represents cell `AAACGAAAGATTGACA-1` and window `window_1`. The format is "cell barcode", ".window_" and integer.

`windowcovariates`
: Chromosomal windows for which copy number gain/loss are initially inferred. Required columns are chr, start, end, window (for example, `window_1`) and peaks. Peaks is a numeric column representing chromatin accessibility.

`levels`
: Possible values of ploidy. For example, `c(2, 4)` if the cells can be diploids or tetraploids. The values must be larger than one.

`nfragspercellmin`
: Minimum number of fragments for a cell-window to be eligible.

`nfragspercellmax`
: Maximum number of fragments for a cell-window to be eligible.

`deltaBICthreshold`
: Only the CNVs with deltaBIC smaller than this threshold are adopted.

Value

A list with two elements. `CNV` is a data frame of the CNVs identified in the dataset. `cellwindowCN` is a data frame indicating the ploidy for each cell and the inferred standardized copy number for each cell-window.

`fragmentoverlapcount` *Count Overlap of ATAC-seq Fragments*

Description

Count Overlap of ATAC-seq Fragments

Usage

```
fragmentoverlapcount(
  file,
  targetregions,
  excluderegions = NULL,
  targetbarcodes = NULL,
  Tn5offset = c(1, 0),
  barcodesuffix = NULL,
  dobptonext = FALSE
)
```

Arguments

`file`	Filename of the file for ATAC-seq fragments. The file must be block gzipped (using the `bgzip` command) and accompanied with the index file (made using the `tabix` command). The uncompressed file must be a tab delimited file, where each row represents one fragment. The first four columns are chromosome name, start position, end position, and barcode (i.e., name) of the cell including the fragment. The remaining columns are ignored. See vignette for details.
`targetregions`	GRanges object for the regions where overlaps are counted. Usually all of the autosomes. If there is memory problem, split a chromosome into smaller chunks, for example by 10 Mb. The function loads each element of `targetregions` sequentially, and smaller elements require less memory.
`excluderegions`	GRanges object for the regions to be excluded. Simple repeats in the genome should be listed here, because repeats can cause false overlaps. A fragment is discarded if its 5' or 3' end is located in `excluderegions`. If `NULL`, fragments are not excluded by this criterion.
`targetbarcodes`	Character vector for the barcodes of cells to be analyzed, such as those passing quality control. If `NULL`, all barcodes in the input file are analyzed.
`Tn5offset`	Numeric vector of length two. The enzyme for ATAC-seq is a homodimer of Tn5. The transposition sites of two Tn5 proteins are 9 bp apart, and the (representative) site of accessibility is in between. If the start and end position of your input file is taken from BAM file, set the paramater to `c(4, -5)` to adjust the offset. Alternatively, values such as `c(0, -9)` could generate similar results; what matters the most is the difference between the two numbers. The fragments.tsv.gz file generated by 10x Cell Ranger already adjusts the shift but is recorded as a BED file. In this case, use `c(1, 0)` (default value). If unsure, set to `"guess"`, in which case the program returns a guess.

barcodesuffix	Add suffix to barcodes per targetregions.
dobptonext	(experimental feature) Whether to compute smoothed distance to the next fragment (irrelevant to BC) as bptonext, which is the inverse of chromatin accessibility, and append as 9th to 14th columns.

Value

A tibble with each row corresponding to a cell. For each cell, its barcode, the total count of the fragments nfrag, and the count distinguished by overlap depth are given.

GSE129785_SU008_Tumor_Pre

Basal cell carcinoma sample SU008_Tumor_Pre

Description

The dataset includes 788 nuclei obtained from basal cell carcinoma sample SU008_Tumor_Pre. Overlapping of single-nucleus ATAC-seq fragments was computed with the fragmentoverlapcount function.

Usage

```
data(GSE129785_SU008_Tumor_Pre)

SU008_Tumor_Pre_windowcovariates

rescnv
```

Format

SU008_Tumor_Pre_fragmentoverlap is a dataframe of fragmentoverlap.

SU008_Tumor_Pre_windowcovariates is a dataframe of windows and peaks.

rescnv is a list containing the output of cnv function.

Source

GEO, GSE129785

References

Satpathy et al. (2019) Nature Biotechnology 37:925 doi:10.1038/s415870190206z

Examples

```
## Not run:
data(GSE129785_SU008_Tumor_Pre)
levels = c(2, 4)
result = cnv(SU008_Tumor_Pre_fragmentoverlap,
             SU008_Tumor_Pre_windowcovariates,
             levels = levels,
             deltaBICthreshold = -600)

## End(Not run)
```

`ploidy` *Infer Ploidy from ATAC-seq Fragment Overlap*

Description

Infer Ploidy from ATAC-seq Fragment Overlap

Usage

```
ploidy(
  fragmentoverlap,
  levels,
  s = 100,
  epsilon = 1e-08,
  subsamplesize = NULL,
  dobayes = FALSE,
  prop = 0.9
)
```

Arguments

`fragmentoverlap`	Frequency of fragment overlap in each cell computed by the function `fragmentoverlapcount`.
`levels`	Possible values of ploidy. For example, `c(2, 4)` if the cells can be diploids or tetraploids. The values must be larger than one.
`s`	Seed for random numbers used in EM algorithm.
`epsilon`	Convergence criterion for the EM algorithm.
`subsamplesize`	EM algorithm becomes difficult to converge when the number of cells is very large. By setting the parameter (e.g. to 1e4), we can run EM algorithm iteratively, first for `subsamplesize` randomly sampled cells, next for twice the number of cells in repetition. The inferred lambda/theta parameters are used as the initial value in the next repetition.
`dobayes`	(experimental feature) Whether to perform Bayesian inference, which takes long computation time.
`prop`	Proportion of peaks that can be fitted with binomal distribution in ploidy.bayes. The rest of peaks are allowed to have depth larger than the ploidy.

Value

A data.frame with each row corresponding to a cell. For each cell, its barcode, ploidy inferred by 1) moment method, 2) the same with additional K-means clustering, 3) EM algorithm of mixture, and, optionally, 4) Bayesian inference are given. I recommend using `ploidy.moment` or `ploidy.em`. When `fragmentoverlapcount` was computed with `dobptonext=TRUE`, we only use the chromosomal sites with chromatin accessibility in top 10 This requires longer computation time.

`SHR_m154211` *Liver Cells from a Rat*

Description

The dataset includes 3572 nuclei obtained from the liver of a 16 weeks old male rat, which was fed normal diet. Overlapping of single-nucleus ATAC-seq fragments was computed with the `fragmentoverlapcount` function and saved as `fragmentoverlap`. The cell type of the nuclei are saved in the data.frame `cells`. The data for rat SHR_m154211 was taken from the publication cited below.

Usage

```
data(SHR_m154211)
```

Format

An object of class `list` of length 2.

Source

Takeuchi et al. (2022) bioRxiv doi:10.1101/2022.07.12.499681

Examples

```
data(SHR_m154211)
fragmentoverlap = SHR_m154211$fragmentoverlap
p = ploidy(fragmentoverlap, c(2, 4, 8))
head(p)
cells = SHR_m154211$cells
table(cells$celltype, p$ploidy.moment[match(cells$barcode, p$barcode)])
```

Index