Package 'partCNV'

April 28, 2025

Type Package

Title Infer locally aneuploid cells using single cell RNA-seq data

Version 1.6.0

Description This package uses a statistical framework for rapid and accurate detection of aneuploid cells with local copy number deletion or amplification. Our method uses an EM algorithm with mixtures of Poisson distributions while incorporating cytogenetics information (e.g., regional deletion or amplification) to guide the classification (partCNV). When applicable, we further improve the accuracy by integrating a Hidden Markov Model for feature selection (partCNVH).

Imports stats, data.table, depmixS4, Seurat, SingleCellExperiment, AnnotationHub, magrittr, GenomicRanges, BiocStyle

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

Dependents R (>= 4.2.0)

VignetteBuilder knitr

License GPL-2

Encoding UTF-8

RoxygenNote 7.2.3

biocViews Software, CopyNumberVariation, HiddenMarkovModel, SingleCell, Classification

Config/testthat/edition 3

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

git_branch RELEASE_3_21

git_last_commit 6d45f26

git_last_commit_date 2025-04-15

Repository Bioconductor 3.21

Date/Publication 2025-04-27

Author Ziyi Li [aut, cre, ctb], Ruoxing Li [ctb]

Maintainer Ziyi Li <zli16@mdanderson.org>

Contents

GetCytoLocation 2
GetExprCountCyto 3
Hg38_gtf 4
NormalizeCounts 4
partCNV 5
partCNVH 6
SimData 7
SimDataSce 7

Index **8**

`GetCytoLocation` *Get exact location of the interested cytogenetics feature*

Description

This function helps you identify the location of the cytogenetics feature. For example, if the region of interest is chr20(q11.1-q13.1), this function greps the start and end location of this region. Additionally, you can just put in "chr20", and it provides you all the available cytogenetics locations on chromosome 20. It also report the number of genes within the region. If the number of genes is too few, we recommend to include neighboring regions to provide more stable results.

Usage

```
GetCytoLocation(cyto_feature = NULL, chr = NULL, start = NULL, end = NULL)
```

Arguments

`cyto_feature`	the cytogenetics location you are interested. It can be of two format: chr20(q11.1-q13.1) or chr20. For the first format, the start and end regions need to be separated by "-". If you are interested in one region for example, chr20(q11.1), put it as chr20(q11.1-11.1). For the second format, all the available regions will be printed for selection.
`chr`	chromosome location of the interested region. This is only used when cyto_feature is null.
`start`	starting location of the interested region. This is only used when cyto_feature is null.
`end`	ending location of the interested region. This is only used when cyto_feature is null.

Value

If the first format of cyto_feature is provided, the starting and ending location as well as the number of genes overlapped with be provided. If the second format of cyto_feature is provided, all the cytogenetics locations will be displayed for review. If the region location (chr, start, end) is provided, the number of genes overlapped will be the output.

Examples

```
### example 1
GetCytoLocation(cyto_feature = "chr20(q11.1-q13.1)")
### example 2
GetCytoLocation(cyto_feature = "chr20")
### example 3
GetCytoLocation(chr = "chr20", start = 25600000, end = 49800000)
```

`GetExprCountCyto` *Get normalized gene expression counts for selected genes*

Description

This function helps normalize the gene expression count matrix if needed and select the genes that are located in the interested region. This procedure happens after applying GetCytoLocation().

Usage

```
GetExprCountCyto(
  cytoloc_output,
  Counts = NULL,
  normalization = TRUE,
  qt_cutoff = 0.99
)
```

Arguments

`cytoloc_output`	The output from the function GetCytoLocation(). The function needs to be run with a complete cytogenetics feature input, e.g., chr20(q11.1-11.1), or providing the chr/start/end loction of the interested region.
`Counts`	The single cell expression matrix for the whole genome of the sample. Rows are genes and columns are cell IDs.
`normalization`	Specify whether the data need to be normalized. Default is TRUE.
`qt_cutoff`	A quantile cut-off to remove genes that are almost all zeros. If the cut-off is 0.99, then all the genes expressed in less than 0.01 percent of cells will be eliminated for further analysis.

Value

A list with normalized and ordered gene expression for the interested cytogenetics region.

Examples

```
res <- GetCytoLocation(cyto_feature = "chr20(q11.1-q13.1)")
data(SimData)
GetExprCountCyto(cytoloc_output = res, Counts = as.matrix(SimData), normalization = TRUE,
qt_cutoff = 0.99)
```

Hg38_gtf *GTF data for Hg38 genome*

Description

Gene location data for Hg38 genome

Usage

```
data(Hg38_gtf)
```

Format

An object of class "data frame"

Value

Gene location data for Hg38 genome

Source

Gencode Archive

References

Frankish, Adam, et al. Nucleic acids research 47.D1 (2019): D766-D773. (PubMed)

Examples

```
data(Hg38_gtf)
head(Hg38_gtf)
```

NormalizeCounts *Extract and normalize gene expression counts for a SingleCellExperiment object*

Description

This function helps normalize the gene expression count matrix for a SingleCellExperiment object.

Usage

```
NormalizeCounts(obj, scale_factor = 10000)
```

Arguments

obj	The SingleCellExperiment object.
scale_factor	Feature counts for each cell are divided by the total counts for that cell and multiplied by the scale.factor, and then natural-log transformed using log1p.

Value

A normalized gene expression counts matrix.

Examples

```
data(SimDataSce)
counts_mat <- NormalizeCounts(SimDataSce)
```

---

partCNV *Infer cells that are locally aneuploid using partCNV*

Description

This function uses EM algorithm to cluster the cells with a Poisson Mixture model. Cells will be grouped into two groups, locally aneuploid (status = 1) and diploid (status = 0).

Usage

```
partCNV(int_counts, cyto_type, cyto_p, tau = 0.1, maxniter = 1000)
```

Arguments

int_counts	Normalized gene expression counts for the genes in the interested region, e.g., the ProcessedCount variable from the output of GetExprCountCyto().
cyto_type	The type of the cytogenetics alteration. It can only be "del" or "amp"
cyto_p	The percentage of cells with the cytogenetics alteration, e.g., 0.2.
tau	The variance of the prior information. Default is 0.1. If you have less confidence, specify a larger tau, e.g., 10.
maxniter	The maximum number of iterations of the EM algorithm.

Value

A vector with the cell status inferred by the method, 1 is aneuploid and 0 is diploid.

Examples

```
### example 1
cytoloc <- GetCytoLocation(cyto_feature = "chr20(q11.1-q13.1)")
data(SimData)
exprout <- GetExprCountCyto(cytoloc_output = cytoloc, Counts = as.matrix(SimData), normalization = TRUE, qt_cutoff
status <- partCNV(int_counts = exprout$ProcessedCount, cyto_type = "del", cyto_p = 0.2)
```

---

partCNVH *Infer cells that are locally aneuploid using partCNVH*

Description

This function uses EM algorithm to cluster the cells with a Poisson Mixture model in the first step. With the results, it applies hidden markov model to improve feature selection. After that, another round of EM algorithm is applied to obtain the final cell status. Cells will be grouped into two groups, locally aneuploid (status = 1) and diploid (status = 0).

Usage

```
partCNVH(int_counts, cyto_type, cyto_p, tau = 0.1, maxniter = 1000, navg = 50)
```

Arguments

int_counts	Normalized gene expression counts for the genes in the interested region, e.g., the ProcessedCount variable from the output of GetExprCountCyto().
cyto_type	The type of the cytogenetics alteration. It can only be "del" or "amp"
cyto_p	The percentage of cells with the cytogenetics alteration, e.g., 0.2.
tau	The variance of the prior information. Default is 0.1. If you have less confidence, specify a larger tau, e.g., 10.
maxniter	The maximum number of iterations of the EM algorithm.
navg	Number of genes used for rolling average.

Value

A vector with the cell status inferred by the method, 1 is aneuploid and 0 is diploid.

Examples

```
cytoloc <- GetCytoLocation(cyto_feature = "chr20(q11.1-q13.1)")
data(SimData)
exprout <- GetExprCountCyto(cytoloc_output = cytoloc, Counts = as.matrix(SimData), normalization = TRUE, qt_cutoff
status <- partCNVH(int_counts = exprout$ProcessedCount, cyto_type = "del", cyto_p = 0.2, navg = 50)
```

SimData — *Simulation data to examplify the usage of the method*

Description

Simulation data to examplify the usage of the method

Usage

```
data(SimData)
```

Format

An object of class "data frame"

Value

Simulation data to examplify the usage of the method

Examples

```
data(SimData)
dim(SimData)
```

SimDataSce — *Simulation SingleCellExperiment object to examplify the usage of the method*

Description

Simulation SingleCellExperiment object to examplify the usage of the method

Usage

```
data(SimDataSce)
```

Format

A SingleCellExperiment object

Value

Simulation SingleCellExperiment object to examplify the usage of the method

Examples

```
data(SimDataSce)
```

Index

∗ **datasets**
Hg38_gtf, 4
SimData, 7
SimDataSce, 7

GetCytoLocation, 2
GetExprCountCyto, 3

Hg38_gtf, 4

NormalizeCounts, 4

partCNV, 5
partCNVH, 6

SimData, 7
SimDataSce, 7