Package ‘genomicInstability’

May 3, 2024

Version 1.10.0
Date 2020-02-21
Title Genomic Instability estimation for scRNA-Seq
Depends R (>= 4.1.0), checkmate
Suggests SingleCellExperiment, ExperimentHub, pROC
Imports mixtools, SummarizedExperiment
Description This package contain functions to run genomic instability analysis (GIA) from scRNA-Seq data.
GIA estimates the association between gene expression and genomic location of the coding genes.
It uses the aREA algorithm to quantify the enrichment of sets of contiguous genes (loci-blocks) on the gene expression profiles and estimates the Genomic Instability Score (GIS) for each analyzed cell.
License file LICENSE
biocViews SystemsBiology, GeneExpression, SingleCell
BiocType Software
BugReports https://github.com/DarwinHealth/genomicInstability
Encoding UTF-8
ARoxygenNote 7.1.1
RoxygenNote 7.1.1
git_url https://git.bioconductor.org/packages/genomicInstability
git_branch RELEASE_3_19
git_last_commit 5ab8fa0
git_last_commit_date 2024-04-30
Repository Bioconductor 3.19
Date/Publication 2024-05-03
Author Mariano Alvarez [aut, cre], Pasquale Laise [aut], DarwinHealth [cph]
Maintainer Mariano Alvarez <reef103@gmail.com>
Contents

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>genomicInstability-package</td>
<td>2</td>
</tr>
<tr>
<td>geneLength</td>
<td>3</td>
</tr>
<tr>
<td>genePosition</td>
<td>3</td>
</tr>
<tr>
<td>generateChromosomeGeneSet</td>
<td>4</td>
</tr>
<tr>
<td>genomicInstabilityScore</td>
<td>4</td>
</tr>
<tr>
<td>giDensityPlot</td>
<td>5</td>
</tr>
<tr>
<td>giLikelihood</td>
<td>6</td>
</tr>
<tr>
<td>inferCNV</td>
<td>7</td>
</tr>
<tr>
<td>plot.inferCNV</td>
<td>8</td>
</tr>
</tbody>
</table>

Index 10

---

**genomicInstability-package**

*genomicInstability: Genomic Instability estimation for scRNA-Seq*

**Description**

This package contains functions to run genomic instability analysis (GIA) from scRNA-Seq data. GIA estimates the association between gene expression and genomic location of the coding genes. It uses the aREA algorithm to quantify the enrichment of sets of contiguous genes (loci-blocks) on the gene expression profiles and estimates the Genomic Instability Score (GIS) for each analyzed cell.

**Details**

The basic functionality of this package can be performed by inferCNV(), to infer the enrichment of loci-blocks on gene expression; genomicInstabilityScore(), to estimate the genomic instability for each of the cells in the scRNASeq dataset; giLikelihood(), to estimate the relative likelihood for each cell to be normal (low genomic instability) or tumor (high genomic instability); plot() and giDensityPlot() to plot the scores per loci-block and the distribution of the genomic instability score, respectively.

**Author(s)**

**Maintainer:** Mariano Alvarez <reef103@gmail.com>

Authors:

- Pasquale Laise <plaise@darwinhealth.com>

Other contributors:

- DarwinHealth [copyright holder]
See Also

[inferCNV()] for estimating loci-block enrichment, [genomicInstabilityScore()] for estimating the
genomic instability of each cell in the dataset, [giLikelihood()] for estimating the relative likelihood
for the cells to be normal or neoplastic, [plot.inferCNV()] and [giDensityPlot()] to plot the results.

---

**geneLength**  
*Average length of human and mouse known genes*

**Description**  
A dataset containing the average length for known mouse and human genes

**Usage**  
geneLength

**Format**  
Vector of integers indicating the average length in bp for each gene, indicated with EntrezIDs as
name argument. To access this data use:

- `data(hg38)`  Human
- `data(mm10)`  Mouse

---

**genePosition**  
*Chromosomal coordinate of human and mouse known genes*

**Description**  
A dataset containing the chromosomal coordinate for known human and mouse genes

**Usage**  
genePosition

**Format**  
data.frame with 2 columns: Chromosome and Coordinate. To access this data use:

- `data(hg38)`  Human
- `data(mm10)`  Mouse
genomicInstabilityScore

Description
This function computes the genomic instability for an object of class inferCNV

Usage
genomicInstabilityScore(cnv, likelihood = FALSE)

Arguments
  cnv          Object of class inferCNV generated by inferCNV() function
  likelihood   Logical, whether the genomic instability likelihood should be estimated

generateChromosomeGeneSet

Topological gene sets

Description
This function generates a list of sets of k genes encoded by neighbor loci

Usage
generateChromosomeGeneSet(species = c("human", "mouse"), k = 100, skip = 25)

Arguments
  species     Character string indicating the species, either human or mouse
  k           Integer indicating the number of genes per set
  skip        Integer indicating the displacement of the window for selecting the k genes

Value
List of topologically-close gene sets

Examples
  chrom_set <- generateChromosomeGeneSet('human')
  length(chrom_set)
  chrom_set[seq_len(2)]
giDensityPlot

Value

Object of class inferCNV with updated slots for gis and gisnull

See Also

[inferCNV()] to infer the enrichment of loci-blocks in the gene expression data.

Examples

eh <- ExperimentHub::ExperimentHub()
dset <- eh[["EH5419"]]
tpm_matrix <- SummarizedExperiment::assays(dset)$TPM
set.seed(1)
tpm_matrix <- tpm_matrix[, sample(ncol(tpm_matrix), 500)]
cnv <- inferCNV(tpm_matrix)
cnv <- genomicInstabilityScore(cnv)
plot(density(cnv$gis))

---

**giDensityPlot**

*Genomic instability plot*

Description

This function plot the genomic instability distribution, gaussian fits and null distribution if available

Usage

`giDensityPlot(inferCNV, legend = c("topleft", "top", "topright", "none"), ...)`

Arguments

- `inferCNV`: Object of class inferCNV
- `legend`: Character string indicating the location of the legend. none to not include it
- `...`: Additional parameters for plot()

Value

None, a figure is created in the default output device

See Also

[giLikelihood()] to estimate the relative likelihood, [genomicInstabilityScore()] to estimate the genomic instability score for each cell in the dataset, and [inferCNV()] to infer the enrichment of loci-blocks in the gene expression data.
Examples

eh <- ExperimentHub::ExperimentHub()
dset <- eh[["EH5419"]]
 TPM <- SummarizedExperiment::assays(dset)$TPM
 set.seed(1)
 TPM <- TPM[, sample(ncol(TPM), 500)]
cnv <- inferCNV(TPM)
cnv <- genomicInstabilityScore(cnv)
cnv <- giLikelihood(cnv, distros=c(3, 3), tumor=2:3)
giDensityPlot(cnv)

---

### giLikelihood

**Genomic instability likelihood**

**Description**

This function computes the genomic instability likelihood

**Usage**

```r
giLikelihood(
inferCNV, 
recompute = TRUE, 
distros = c(1, 3), 
tumor = NULL, 
normal = NULL
)
```

**Arguments**

- `inferCNV` InerCNV-class object
- `recompute` Logical, whether the model fits should be re-computed
- `distros` Vector of 2 integers indicating the minimum and maximum number of Gaussian models to fit
- `tumor` Optional vector of integers indicating the Gaussians considered as tumors
- `normal` Optional vector of integers indicating the Gaussians considered as normal. This is only useful when no null model has been provided for the analysis

**Value**

Updated inferCNV-class object with gi_likelihood slot

**See Also**

- [genomicInstabilityScore()] to estimate the genomic instability score for each cell in the dataset, and [inferCNV()] to infer the enrichment of loci-blocks in the gene expression data.
Examples

```r
eh <- ExperimentHub::ExperimentHub()
dset <- eh[["EH5419"]]
tpm_matrix <- SummarizedExperiment::assays(dset)$TPM
set.seed(1)
tpm_matrix <- tpm_matrix[, sample(ncol(tpm_matrix), 500)]
cnv <- inferCNV(tpm_matrix)
cnv <- genomicInstabilityScore(cnv)
cnv <- giLikelihood(cnv, distros=c(3, 3), tumor=2:3)
print(cnv$gi_fit)
plot(density(cnv$gi_likelihood, from=0, to=1))
```

---

**inferCNV**

Inference of CNV from expression data

**Description**

This function estimates the CNV score based on expression data.

**Usage**

```r
inferCNV(
  expmat,
  nullmat = NULL,
  species = c("human", "mouse"),
  k = 100,
  skip = 25,
  min_geneset = 10,
  verbose = TRUE
)
```

**Arguments**

- **expmat**: Matrix of gene expression profiles or signatures with genes `(entrezID)` in rows and samples in columns.
- **nullmat**: Optional matrix with same number of rows as `expmat` to be used as null model.
- **species**: Character string indicating the species, either human or mouse.
- **k**: Integer indicating the number of genes per set.
- **skip**: Integer indicating the displacement of the window for selecting the `k` genes.
- **min_geneset**: Integer indicating the minimum size for the genesets.
- **verbose**: Logical, whether progress should be reported.

**Value**

Object of class `inferCNV`, which is a list containing matrix of nes, and parameters (param), including species, window (k) and skip.
Examples

```r
eh <- ExperimentHub::ExperimentHub()
dset <- eh["EH5419"]
tpm_matrix <- SummarizedExperiment::assays(dset)$TPM
set.seed(1)
tpm_matrix <- tpm_matrix[, sample(ncol(tpm_matrix), 500)]
cnv <- inferCNV(tpm_matrix)
class(cnv)
names(cnv)
cnv$nes[1:5, 1:3]
```

plot.inferCNV

Plot chromosome map

Description

This function generates a chromosomes map plot for the inferred CNVs

Usage

```r
## S3 method for class 'inferCNV'
plot(x, output = NULL, threshold = 0.2, gamma = 1.5, resolution = 150, ...)
```

Arguments

- `x`: Object of class inferCNV
- `output`: Optional output PDF file name (with extension)
- `threshold`: Likelihood threshold for identifying genomically instable cells/samples, 0 disables this filter
- `gamma`: Number indicating the gamma transformation for the colors
- `resolution`: Integer indicating the ppi for the png and jpg output files
- `...`: Additional parameters for plot

Value

Nothing, a plot is generated in the default output devise

See Also

- `giLikelihood()` to estimate the relative likelihood, `genomicInstabilityScore()` to estimate the genomic instability score for each cell in the dataset, and `inferCNV()` to infer the enrichment of loci-blocks in the gene expression data.
Examples

```r
eh <- ExperimentHub::ExperimentHub()
dset <- eh["EH5419"]
 TPM_matrix <- SummarizedExperiment::assays(dset)$TPM
set.seed(1)
 TPM_matrix <- TPM_matrix[, sample(ncol(TPM_matrix), 500)]
cnv <- inferCNV(TPM_matrix)
cnv <- genomicInstabilityScore(cnv)
cnv <- giLikelihood(cnv, distros=c(3, 3), tumor=2:3)
plot(cnv, output='test.png')
```
Index

* datasets
  geneLength, 3
  genePosition, 3
* internal
  genomicInstability-package, 2

geneLength, 3
genePosition, 3
generateChromosomeGeneSet, 4
genomicInstability
  (genomicInstability-package), 2
  genomicInstability-package, 2
genomicInstabilityScore, 4
giDensityPlot, 5
giLikelihood, 6

inferCNV, 7
plot.inferCNV, 8