Package 'oncomix'

October 14, 2021

Title Identifying Genes Overexpressed in Subsets of Tumors from Tumor-Normal mRNA Expression Data

Version 1.14.0

Author Daniel Pique, John Greally, Jessica Mar

Maintainer Daniel Pique <daniel.pique@med.einstein.yu.edu>

Description This package helps identify mRNAs that are overexpressed in subsets of tumors relative to normal tissue. Ideal inputs would be paired tumor-normal data from the same tissue from many patients (>15 pairs). This unsupervised approach relies on the observation that oncogenes are characteristically overexpressed in only a subset of tumors in the population, and may help identify oncogene candidates purely based on differences in mRNA expression between previously unknown subtypes.

Depends R (>= 3.4.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests knitr, rmarkdown, testthat, RMySQL

Imports ggplot2, ggrepel, RColorBrewer, mclust, stats, SummarizedExperiment

VignetteBuilder knitr

biocViews GeneExpression, Sequencing

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

git_branch RELEASE_3_13

git_last_commit 4c21cd9

git_last_commit_date 2021-05-19

Date/Publication 2021-10-14

R topics documented:

`exprNmlIsof` *Human Breast Cancer RNA-sequencing data from TCGA - Adj. Normal Tissue*

Description

These RNA-sequencing expression data were obtained from the Cancer Genome Atlas project (now the Genomic Data Commons (GDC)). The sequencing data were generated from adjacent normal breast tissue data from 113 patients with breast cancer. Quantification of RNA expression values was performed using standard GDC pipelines. The expression values are reported in transcripts per million reads. Out of an initial 73,599 RNA transcripts, 700 are included as part of this dataset. These 700 transcripts represent a random subset of the transcripts with at least 20 samples. Rows contain anonymized patient identifiers, while columns contain UCSC gene symbols.

Author(s)

Daniel Pique <daniel.pique@med.einstein.yu.edu>

References

https://gdc.cancer.gov/

`exprTumIsof` *Human Breast Cancer RNA-sequencing data from TCGA - Tumor Tissue*

Description

These RNA-sequencing expression data were obtained from the Cancer Genome Atlas project (now the Genomic Data Commons (GDC)). The sequencing data were generated from breast carcinoma samples from 113 patients. Quantification of RNA expression values was performed using standard GDC pipelines. The expression values are reported in transcripts per million reads. Out of an initial 73,599 RNA transcripts, 700 are included as part of this dataset. These 700 transcripts represent a random subset of the transcripts with at least 20 anonymized patient identifiers, while columns contain UCSC gene symbols.

Author(s)

Daniel Pique <daniel.pique@med.einstein.yu.edu>

References

https://gdc.cancer.gov/

`mixModelParams` *Generate the parameters for two 2-component Gaussian mixture models with equal variances*

Description

This function allows you to generate the parameters for two 2-component Gaussian mixture model with equal variances from 2 matrices of data with a priori labels (eg tumor vs normal.) This application was originally intended for matrices of gene expression data treated with 2 conditions.

Usage

```
mixModelParams(exprNml, exprTum)
```

Arguments

`exprNml`	A dataframe (S3 or S4), matrix, or SummarizedExperiment object containing normal data with patients as columns and genes as rows.
`exprTum`	A dataframe (S3 or S4), matrix, or SummarizedExperiment object containing tumor data with patients as columns and genes as rows.

Value

Returns a dataframe, each element of which contains the 12 mixture model parameters for each gene in an n x 12 matrix, where n is the number of genes.

Examples

```
exprNml <- as.data.frame(matrix(data=rgamma(n=150, shape=2, rate=2),
nrow=10, ncol=15))
colnames(exprNml) <- paste0("patientN", seq_len(ncol(exprNml)))
rownames(exprNml) <- paste0("gene", seq_len(nrow(exprNml)))

exprTum <- as.data.frame(matrix(data=rgamma(n=150, shape=4, rate=3),
nrow=10, ncol=15))
colnames(exprTum) <- paste0("patientT", seq_len(ncol(exprTum)))
rownames(exprTum) <- paste0("gene", seq_len(nrow(exprTum)))

mmParams <- mixModelParams(exprNml, exprTum)
```

---

oncoMixBimodal *Creating a schematic of a 2-component mixture model*

Description

This function allows you to generate a plot

Usage

```
oncoMixBimodal(means = c(3, 7))
```

Arguments

means	Set the values for the difference between parameter means

Value

Returns a ggplot object that shows a 2-component Gaussian mixture model

Examples

```
oncoMixBimodal(means=c(3,7))
oncoMixBimodal(means=c(3,10))
```

oncoMixIdeal *Creating an ideal oncomix gene*

Description

This function allows you to generate a plot

Usage

```
oncoMixIdeal(means = c(3, 7))
```

Arguments

means	Set the difference between parameter means for the overexpressed (oe) group. Defaults to c(3,7)

Value

Returns a ggplot object that shows the statistical model for an idealized/theoretical oncogene candidate mRNA that is overexpressed in a subset of tumors

Examples

```
oncoMixIdeal(means=c(3,10))
oncoMixIdeal(means=c(2,18.5))
```

oncoMixTraditionalDE *Creating a schematic of a traditional diff. expression experiment*

Description

This function allows you to generate a schematic of the assumptions of a traditional DE expermiment between two known groups.

Usage

```
oncoMixTraditionalDE(means = c(3, 7))
```

Arguments

means	Set the values for the difference between parameter means

Value

Returns a ggplot object that shows the traditional method (2 sample t-test) for mRNA differential expression.

Examples

```
oncoMixTraditionalDE(means=c(3,7))
oncoMixTraditionalDE(means=c(3,10))
```

`plotGeneHist` *Plot a histogram of gene expression values from tumor and adjacent normal tissue.*

Description

This function allows you to plot a histogram of gene expression values from tumor and adjacent normal tissue with the option of including the best fitting Gaussian curve.

Usage

```
plotGeneHist(mmParams, exprNml, exprTum, isof)
```

Arguments

`mmParams`	The output from the getMixModelParams function.
`exprNml`	A dataframe (S3 or S4), matrix, or SummarizedExperiment object containing normal data with patients as columns and genes as rows.
`exprTum`	A dataframe (S3 or S4), matrix, or SummarizedExperiment object containing tumor data with patients as columns and genes as rows.
`isof`	The gene isoform to visualize

Value

Returns a histogram of the gene expression values from the two groups.

See Also

`mixModelParams`

Examples

```
exprNml <- as.data.frame(matrix(data=rgamma(n=150, shape=2, rate=2),
nrow=10, ncol=15))
colnames(exprNml) <- paste0("patientN", seq_len(ncol(exprNml)))
rownames(exprNml) <- paste0("gene", seq_len(nrow(exprNml)))

exprTum <- as.data.frame(matrix(data=rgamma(n=150, shape=4, rate=3),
nrow=10, ncol=15))
colnames(exprTum) <- paste0("patientT", seq_len(ncol(exprTum)))
rownames(exprTum) <- paste0("gene", seq_len(nrow(exprTum)))
mmParams <- mixModelParams(exprNml, exprTum)
isof <- rownames(mmParams)[1]
plotGeneHist(mmParams, exprNml, exprTum, isof)
```

queryRes *Oncogene Database Mapping Gene Symbol to UCSC ID (kgID)*

Description

These data were downloaded in September 2017 from the url listed below and represent a mapping between gene symbols in ongene, an oncogene database curated from the scientific literature, and gene identifiers from the University of California, Santa Cruz's genomic database.

Author(s)

Min Zhao <mzhao@usc.edu.au>

References

http://ongene.bioinfo-minzhao.org/ongene_human.txt

scatterMixPlot *Generate a scatter plot with the output from mixModelParams*

Description

This function allows you to generate the parameters for two 2-component mixture models with equal variances

Usage

```
scatterMixPlot(mmParams, selIndThresh = 1, geneLabels = NULL)
```

Arguments

mmParams	The output from the mixModelParams function. Will utilize the deltaMu2 and deltaMu1 rows.
selIndThresh	This is the selectivity index threshold to use. All genes with SI values above this threshold will be highlighted in purple. Specify either selIndThresh or geneLabels (not both simultaneously).
geneLabels	A character vector of gene names used to label the genes with that name on the scatter plot. Specify either selIndThresh or geneLabels (not both simultaneously).

Value

Returns a ggplot scatter object that can be plotted

See Also

mixModelParams

Examples

```
exprNml <- as.data.frame(matrix(data=rgamma(n=150, shape=2, rate=2),
nrow=10, ncol=15))
colnames(exprNml) <- paste0("patientN", seq_len(ncol(exprNml)))
rownames(exprNml) <- paste0("gene", seq_len(nrow(exprNml)))

exprTum <- as.data.frame(matrix(data=rgamma(n=150, shape=4, rate=3),
nrow=10, ncol=15))
colnames(exprTum) <- paste0("patientT", seq_len(ncol(exprTum)))
rownames(exprTum) <- paste0("gene", seq_len(nrow(exprTum)))

mmParams <- mixModelParams(exprNml, exprTum)
scatterMixPlot(mmParams)
```

---

topGeneQuants *Identify genes that meet pre-specified quantiles*

Description

This function allows you to subset genes that are above pre-specified quantiles and that most closely resemble the distribution of oncogenes.

Usage

```
topGeneQuants(mmParams, deltMu2Thr = 90, deltMu1Thr = 10, siThr = 0.99)
```

Arguments

mmParams	The output from the mixModelParams function.
deltMu2Thr	The percentile threshold for the deltaMu2 statistic. All genes exceeding this percentile threshold will be selected.
deltMu1Thr	The percentile threshold for the deltaMu1 statistic. All genes exceeding this percentile threshold will be selected.
siThr	The threshold for the selectivity index statistic (between 0-1). All genes exceeding this threshold will be selected.

Value

Returns a dataframe containing all genes meeting the prespecified thresholds.

See Also

mixModelParams

Examples

```
exprNml <- as.data.frame(matrix(data=rgamma(n=150, shape=2, rate=2),
nrow=10, ncol=15))
colnames(exprNml) <- paste0("patientN", seq_len(ncol(exprNml)))
rownames(exprNml) <- paste0("gene", seq_len(nrow(exprNml)))

exprTum <- as.data.frame(matrix(data=rgamma(n=150, shape=4, rate=3),
nrow=10, ncol=15))
colnames(exprTum) <- paste0("patientT", seq_len(ncol(exprTum)))
rownames(exprTum) <- paste0("gene", seq_len(nrow(exprTum)))

mmParams <- mixModelParams(exprNml, exprTum)
topGeneQuants(mmParams)
```

Index