Package 'NetSciDataCompanion'

July 25, 2023

Title Tools for Analyzing TCGA and GTEx Data

Version 0.0.0.9000

Description What the package does (one paragraph).

License `use_mit_license()`

Depends data.table,
dplyr,
edgeR,
EpiSCORE,
GenomicDataCommons,
magrittr,
presto,
recount,
recount3,
stringr,
TCGAPurityFiltering,
tidyr

Remotes immunogenomics/presto,
aet21/EpiSCORE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

R topics documented:

convertBetaToM 2
CreateNetSciDataCompanionObject 2
extractSampleAndGeneInfo 3
filterBarcodesIntersection 4
filterChromosome 4
filterDuplicatesSeqDepth 5
filterDuplicatesSeqDepthOther 6
filterGenesByTPM 6
filterPurity 7
filterSampleType 8
filterTumorType 8
geneNameToENSG 9
getGeneInfo 10

logTPMNormalization 10
mapBarcodeToBarcode 11
mapProbesToGenes 12
NetSciDataCompanion 13

Index **14**

`convertBetaToM` *Convert methylation beta values to M-values.*

Description

This function uses the typical logit base 2 transformation to convert from methylation beta values (in the [0,1] range) to m-values (on the real line). The formula is m = log2(beta/(1-beta)).

Usage

```
convertBetaToM(methylation_betas)
```

Arguments

`methylation_betas`
A numeric vector of values in the range [0,1].

Value

A numeric vector of m-values corresponding to the converted values of `methylation_betas`.

Author(s)

Kate Hoff Shutta (kshutta@hsph.harvard.edu)

References

https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-11-587

`CreateNetSciDataCompanionObject` *Constructor for the* `NetSciDataCompanion` *object.*

Description

This function is used to construct a `NetSciDataCompanion` object. The member functions of this object are the functions of this package.

Usage

```
CreateNetSciDataCompanionObject(clinical_patient_file, project_name)
```

Arguments

`clinical_patient_file`
: Path to a comma-separated file containing clinical data for the samples of interest.

`project_name`
: A character string that identifies the project.

Value

A `NetSciDataCompanion` object.

Author(s)

Panagiotis Mandros (pmandros@hsph.harvard.edu)

`extractSampleAndGeneInfo` *Extracts experiment-specific information and metadata from ranged summarized experiment object.*

Usage

```
extractSampleAndGeneInfo(expression_rds_obj)
```

Arguments

`expression_rds_obj`
: A ranged summarized experiment object

Value

`rds_sample_info`
: metadata about the samples (columns)

`rds_gene_info`
: metadata about the genes (rows)

Author(s)

Jonas Fischer (jfischer@hsph.harvard.edu)

`filterBarcodesIntersection` *Convenience wrapper function for* `mapBarcodeToBarcode` *that applies the function directly to two data frames.*

Description

This function returns a list of the two argument data frames, intersected, and the second frame ordered to match the first. NOTE: Ordering is done based on columns, which are expected to be named by TCGA barcodes.

Usage

```
filterBarcodesIntersection(exp1, exp2)
```

Arguments

`exp1`	A matrix or dataframe with TCGA barcodes as the `colnames` attribute.
`exp2`	A matrix or dataframe with TCGA barcodes as the `colnames` attribute.

Details

No additional details at this time.

Value

`mappedExp1`	A data frame filtered to include only columns with TCGA barcodes that are in both `colnames(exp1)` and `colnames(exp2)`.
`mappedExp2`	A data frame filtered to include only columns with TCGA barcodes that are in both `colnames(exp1)` and `colnames(exp2)`. IMPORTANT: The columns of `mappedExp2` are ordered to match the column ordering of `mappedExp1`.

Author(s)

Jonas Fischer (jfischer@hsph.harvard.edu)

`filterChromosome` *Filter for genes in a particular chromosome or chromosomes.*

Description

This function filters for genes in a particular chromosome or chromosomes.

Usage

```
filterTumorType(rds_gene_info, chroms)
```

Arguments

`rds_gene_info`	A data frame extracted from a `RangedSummarizedExperiment` object containing expression data using the `extractSampleAndGeneInfo` function.
`chroms`	A character vector of chromosomes. Must exactly match chromosomes in the `seqname` attribute of `rds_gene_info`.

Value

Integer vector of the row indices (genes) in `rds_gene_info` to keep.

Author(s)

Jonas Fischer (jfischer@hsph.harvard.edu)

`filterDuplicatesSeqDepth` *A function to filter duplicates based on RNA sequencing depth.*

Description

This function filters out duplicates based on RNA-seq, keeping the samples with maximum read depth. Returns indices of samples to KEEP.

Usage

```
filterDuplicatesSeqDepth(expression_count_matrix)
```

Arguments

`expression_count_matrix`	Matrix of count data from RNA-seq experiment, with genes in rows and samples in columns.

Value

Integer vector of indices to keep, corresponding to columns of `expression_count_matrix`.

Author(s)

Jonas Fischer(jfischer@hsph.harvard.edu)

`filterDuplicatesSeqDepthOther` *A version of* `filterDuplicatesSeqDepth` *to handle the case when sequencing depth is not available.*

Description

This function takes a random duplicate if no info is available on sequencing depth for all vials.

Usage

```
filterDuplicatesSeqDepthOther(expression_count_matrix, tcga_barcodes)
```

Arguments

`expression_count_matrix`
Matrix of count data from RNA-seq experiment, with genes in rows and samples in columns.

`tcga_barcodes` List of TCGA barcodes for filtering.

Value

Integer vector of indices to KEEP in `tcga_barcodes`.

Author(s)

Jonas Fischer (jfischer@hsph.harvard.edu)

`filterGenesByTPM` *Filter genes based on minimum expression level (TPM) across samples.*

Description

Filter all genes which have at least `tpm_threshold` TPM scores in at least `sample_fraction` of samples.

Usage

```
filterGenesByTPM(expression_tpm_matrix, tpm_threshold, sample_fraction)
```

Arguments

`expression_tpm_matrix`
A data frame extracted from a `RangedSummarizedExperiment` object containing expression data using the `extractSampleAndGeneInfo` function.

`tpm_threshold` Numeric > 0. Genes with TPM below this values in more than `sample_fraction` of the data will be excluded from the analysis.

`sample_fraction`
Numeric in [0,1]. Genes with TPM below `tpm_threshold` in more than this fraction of the data will be excluded from the analysis.

Value

Integer vector indexing the rows of `expression_tpm_matrix` that correspond to genes that should be kept.

Author(s)

Jonas Fischer (jfischer@hsph.harvard.edu)

`filterPurity` *Filter samples based on tumor purity.*

Description

This function filters a character vector of TCGA barcodes for tumor purity based on a particular method and threshold.

Usage

```
filterPurity(TCGA_barcodes, method="ESTIMATE", threshold=.6)
```

Arguments

`TCGA_barcodes`	Character vector of TCGA barcodes that the user wishes to filter based on tumor purity.
`method`	One of "ESTIMATE","ABSOLUTE", "LUMP", "IHC", or "CPE". Default is "ESTIMATE".
`threshold`	Threshold for purity-based filtering. Samples with a purity below `threshold` will be filtered out.

Details

Describe the method options.

Value

Integer vector of indices indicating which samples in `TCGA_barcodes` should be kept.

Author(s)

Panagiotis Mandros (pmandros@hsph.harvard.edu)

References

This code is based on the `TCGAPurityFiltering` package found at `https://github.com/pmandros/TCGAPurityFiltering`.

filterSampleType *Filter samples based on sample type.*

Description

This function filters samples based on sample types. Some examples are: "Primary Tumor", "Solid Tissue Normal", "Primary Blood Derived Cancer - Peripheral Blood".

Usage

```
filterSampleType(TCGA_barcodes, types_of_samples, rds_info)
```

Arguments

`TCGA_barcodes` A character vector of TCGA barcodes.

`types_of_samples` A character vector representing the types of samples to select.

`rds_info` A data frame extracted from a `RangedSummarizedExperiment` object containing TCGA metadata on samples.

Details

Candidate values for `types_of_samples`: "Primary Tumor", "Solid Tissue Normal", "Primary Blood Derived Cancer - Peripheral Blood". If unsure, call this function with an invalid argument. The error message will list available sample types in rds_info.

Value

Named list of containing "index", an integer vector of indices in `TCGA_barcodes` to keep, and "type", a character vector of sample type corresponding to each index.

Author(s)

Jonas Fischer (jfischer@hsph.harvard.edu)

filterTumorType *Filter samples based on tumor type.*

Description

This function filters samples based on tumor type. Some examples are: "Primary Tumor", "Solid Tissue Normal", "Primary Blood Derived Cancer - Peripheral Blood". This function is particularly useful for excluding normal samples from analyses.

Usage

```
filterTumorType(TCGA_barcodes, type_of_tumor, rds_info)
```

Arguments

`TCGA_barcodes`	A character vector of TCGA barcodes.
`type_of_tumor`	A string representing the type of tumor to select. Currently, only a single tumor type is supported.
`rds_info`	A data frame extracted from a `RangedSummarizedExperiment` object containing expression data using the `extractSampleAndGeneInfo` function.

Details

Candidate values for `type_of_tumor`: "Primary Tumor", "Solid Tissue Normal", "Primary Blood Derived Cancer - Peripheral Blood", etc. There are other options that show up in TCGA that are not listed here. Make sure it is an exact match - check spaces, case, etc.

Value

Integer vector of indices in `TCGA_barcodes` to keep.

Author(s)

Jonas Fischer (jfischer@hsph.harvard.edu)

`geneNameToENSG` *Convert from gene name to Ensembl stable id.*

Description

Given an input character vector of gene names, this function converts them to Ensembl stable IDs. Note from `https://useast.ensembl.org/Help/Faq?id=488`: "An Ensembl stable ID consists of five parts: ENS(species)(object type)(identifier).(version)."

Usage

```
geneNameToENSG(gene_names, version = FALSE)
```

Arguments

`gene_names`	Character vector of gene names.
`version`	Boolean; retrieve Ensembl version along with Ensembl identifier.

Value

Character vector of Ensembl stable IDs.

Author(s)

Panagiotis Mandros (pmandros@hsph.harvard.edu)

References

`https://useast.ensembl.org/index.html`

getGeneInfo *Retrieve a variety of gene information based on gene name or Ensemble stable ID.*

Description

This function uses the gene_mapping attribute of the NetSciDataCompanion object to provide information on seqid, source,start,end,strand, gene_id, gene_name, gene_type, and gene_id_no_ver.

Usage

```
getGeneInfo(gene_names_or_ids)
```

Arguments

gene_names_or_ids
A character vector of gene names or Ensembl stable IDs.

Details

This function will determine the input type based on the presence of the string "ENSG".

Value

A data frame with rows representing genes and columns representing gene attributes (e.g., source, start, end.)

Author(s)

Panagiotis Mandros (pmandros@hsph.harvard.edu)

logTPMNormalization *Function for within-array normalization of a RangedSummarizedExperiment object with log transcripts per million (TPM) normalization.*

Description

Returns a named list with raw counts (useful for duplicate filtering based on sequencing depth, see ?filterDuplicatesSeqDepth), TPM, (useful for TPM-based filtering, see ?filterGenesByTPM), and the actual log TPM. A pseudocount of 1 is added to each TPM value for this function, so returned "log TPM" values actually correspond to log(TPM + 1).

Usage

```
logTPMNormalization(expression_rds_obj)
```

Arguments

expression_rds_obj
A RangedSummarizedExperiment object.

Value

counts	A data frame of RNA sequencing counts matching the row and column ordering of expression_rds_obj.
TPM	A data frame of TPM matching the row and column ordering of expression_rds_obj.
logTPM	A data frame of log-transformed TPM with pseudocounts (i.e., log(TPM + 1)) matching the row and column ordering of expression_rds_obj.

Author(s)

Jonas Fischer (jfischer@hsph.harvard.edu)

References

RangedSummarizedExperiment documentation

mapBarcodeToBarcode *Helper function for mapping two sets of TCGA barcodes to each other.*

Description

There are 4 different pieces of information returned in a named list that are all useful depending on the context in which they are used.

is_inter1 is an indicator (boolean) vector of the same length as bc1 that indicates which elements of bc1 are present in bc2.

idcs1 indicates where to find each barcode of bc1 in bc2, returning NA if there is no match. That is, idcs1[i] != NA, then bc1[i] == bc2[idcs1[i]].

The same information is provided for bc2.

Usage

```
mapBarcodeToBarcode(bc1,bc2)
```

Arguments

bc1	Character vector of barcodes in the first set.
bc2	Character vector of barcodes in the second set.

Value

is_inter1	Boolean vector of the same length as bc1 that indicates which elements of bc1 are present in bc2.
idcs1	Integer vector of the same length as bc1 that indicates where to find each barcode of bc1 in bc2, returning NA if there is no match. That is, idcs1[i] != NA, then bc1[i] == bc2[idcs1[i]]
is_inter2	Boolean vector of the same length as bc2 that indicates which elements of bc2 are present in bc1.
idcs2	Integer vector of the same length as bc2 that indicates where to find each barcode of bc2 in bc1, returning NA if there is no match. That is, idcs2[i] != NA, then bc2[i] == bc1[idcs2[i]].

Note

For example, if you want to map experiment 1 onto experiment 2, keeping only the information for samples that are present in both, and reordering the first experiment to match the samples of the second, you can do:

`exp1[,is_inter1] # this will remove samples that are not in experiment 2)`

`exp2[,idcs1] # this will remove samples that are not in exp1 and reorder to match exp1`

Author(s)

Jonas Fischer (jfischer@hsph.harvard.edu)

`mapProbesToGenes` *Maps input probe IDs to gene TSS within a certain range upstream and downstream.*

Usage

```
mapProbesToGenes(probelist, rangeUp, rangeDown, localManifestPath=NA)
```

Arguments

`probelist` A character vector of Illumina array probes, e.g., c("cg03636183","cg19859270").

`rangeUp` The number of base pairs upstream to search for a TSS. Must be a non-negative number.

`rangeDown` The number of base pairs downstream to search for a TSS. Must be a non-negative number.

`localManifestPath`
If you wish to use a manifest file other than the Illumina manifest found at `https://zhouserver.research.chop.edu/InfiniumAnnotation/20210615/HM450/HM450.hg38.manifest.gencode.v36.tsv.gz`, you can pass a path to that file here. It should be formatted in the same way as the Illumina manifest.

Value

A matrix with four columns: probeID, geneName, ensemblID, distToTSS. When a probe maps to more than one TSS within the upstream and downstream parameters provided, the geneName, ensemblID, and distToTSS columns wil contain lists of genes separated by a semicolon (";"). Ordering of the lists matches between the three columns.

Author(s)

Kate Hoff Shutta (kshutta@hsph.harvard.edu)

References

https://zwdzwd.github.io/InfiniumAnnotation

`NetSciDataCompanion` *A package for easy and reproducible wrangling of TCGA and GTEx data.*

Description

Placeholder

Author(s)

Jonas Fischer (jfischer@hsph.harvard.edu)

Panagiotis Mandros (pmandros@hsph.harvard.edu)

Kate Hoff Shutta (kshutta@hsph.harvard.edu)

References

Placeholder

Index

convertBetaToM, 2
CreateNetSciDataCompanionObject, 2

extractSampleAndGeneInfo, 3

filterBarcodesIntersection, 4
filterChromosome, 4
filterDuplicatesSeqDepth, 5
filterDuplicatesSeqDepthOther, 6
filterGenesByTPM, 6
filterPurity, 7
filterSampleType, 8
filterTumorType, 8

geneNameToENSG, 9
getGeneInfo, 10

logTPMNormalization, 10

mapBarcodeToBarcode, 11
mapProbesToGenes, 12

NetSciDataCompanion, 13