Package: drugfindR (via r-universe)

June 17, 2024

Title Investigate iLINCS for candidate repurposable drugs

Version 0.99.657

Description This package provides a convenient way to access the LINCS Signatures available in the iLINCS database. These signatures include Consensus Gene Knockdown Signatures, Gene Overexpression signatures and Chemical Perturbagen Signatures. It also provides a way to enter your own transcriptomic signatures and identify concordant and discordant signatures in the LINCS database.

License GPL-3 + file LICENSE

Encoding UTF-8

URL https://github.com/CogDisResLab/drugfindR

BugReports https://github.com/CogDisResLab/drugfindR/issues

LazyData false

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

biocViews FunctionalPrediction, DifferentialExpression, GeneSetEnrichment

Imports httr, magrittr, tibble, rlang, dplyr, purrr, readr, stringr, stats, lifecycle, S4Vectors

Depends R (>= 4.4.0)

Suggests AnnotationDbi, BiocStyle, biocthis, codemetar, devtools, knitr, rmarkdown, testthat (>= 3.0.0), tidyverse, usethis, vcr (>= 0.6.0), webmockr

Config/testthat/edition 3

Config/testthat/parallel true

VignetteBuilder knitr

X-schema.org-applicationCategory Genomics

X-schema.org-keywords LINCS, iLINCS, drug repurposing, drug discovery, transcriptomics, gene expression, gene knockdown, gene overexpression, chemical perturbagen, drugfindR

X-schema.org-isPartOf https://bioconductor.org

Collate 'utilities.R' 'consensusConcordants.R' 'drugfindR-package.R' 'filterSignature.R' 'getConcordants.R' 'getSignature.R' 'prepareSignature.R' 'investigateSignature.R' 'investigateTarget.R'

Repository https://cogdisreslab.r-universe.dev

RemoteUrl https://github.com/CogDisResLab/drugfindR

RemoteRef HEAD

RemoteSha dbbb832f42dba500ecccb10fd108d197243a81f3

Contents

`.validateLibrary` *Check if the library is valid*

Description

Check if the library is valid

Usage

```
.validateLibrary(lib)
```

Arguments

`lib`	a string of libraries

Value

a boolean

consensusConcordants *Generate a Consensus list of Targets* **[Experimental]**

Description

This function takes a list of (optionally split) concordance dataframes and returns a ranked list of gene or drug targets that have been chose for their maximal similarity to the signature

Usage

```
consensusConcordants(..., paired = FALSE, cutoff = 0.321, cellLine = NULL)
```

Arguments

...	One or Two (see paired) Data Frames with the concordants
paired	Logical indicating whether you split the dataframes by up and down regulated in prior analysis
cutoff	A similarity cutoff value. Defaults to 0.321
cellLine	A character vector of Cell Lines you are interested in.

Value

A tibble with the filtered and deduplicated results

Examples

```
# Get the L1000 signature for LINCSKD_28
kdSignature <- getSignature("LINCSKD_28")

# Get concordant gene knockdown signatures
concordantSignatures <- getConcordants(kdSignature, ilincsLibrary = "KD")

# Get the consensus list of signatures with defaults
consensus <- consensusConcordants(concordantSignatures)

# Get the consensus list of signatures with a different cutoff
consensus <- consensusConcordants(concordantSignatures,
    cutoff = 0.5
)

# Get the consensus list of signatures with a specified cell lines
consensus <- consensusConcordants(concordantSignatures,
    cellLine = c("A549", "MCF7")
)

# Doing a paired analysis
filteredUp <- filterSignature(kdSignature,
    direction = "up", threshold = 0.5
```

```
)
filteredDown <- filterSignature(kdSignature,
    direction = "down", threshold = -0.5
)

concordants_up <- getConcordants(filteredUp, ilincsLibrary = "KD")
concordants_down <- getConcordants(filteredDown, ilincsLibrary = "KD")

consensus <- consensusConcordants(concordants_up,
    concordants_down,
    paired = TRUE
)
```

`filterSignature` *Filter the L1000 Signature* **[Experimental]**

Description

This function filters the L1000 Signature to a given threshold, identifying up-regulated or down-regulated or both up- and down-regulated genes

Usage

```
filterSignature(signature, direction = "any", threshold = NULL, prop = NULL)
```

Arguments

`signature`	A dataframe with the L1000 signature
`direction`	Direction to filter to. Must be one of "up", "down" or "any". Defaults to "any"
`threshold`	A Log Fold-Change Threshold to filter at. This can either be a single value or a vector of two values. If a single value is given, then it is assumed to be a symmetric threshold. If two values are given, then the first value is the down-regulated threshold and the second value is the up-regulated threshold. Cannot be specified with prop
`prop`	A proportion of genes to take from top and bottom. Cannot be specified with threshold

Value

a tibble with the filtered L1000 Signature

Examples

```
# Get the L1000 signature for LINCSKD_28
kdSignature <- getSignature("LINCSKD_28")

# Filter signature by a specific threshold
filteredSignature <- filterSignature(kdSignature, threshold = 0.5)

# Filter signature by a proportion

filteredSignature <- filterSignature(kdSignature, prop = 0.1)

# Filter Signature to up-regulated genes only by a threshold

filteredSignature <- filterSignature(kdSignature,
    direction = "up", threshold = 0.5
)

# Filter the signature using differing thresholds for up and
# down-regulated genes

filteredSignature <- filterSignature(kdSignature,
    threshold = c(-0.75, 0.5)
)
```

---

getConcordants *Get Concordant Signatures from iLINCS* **[Experimental]**

Description

This function takes a full or filtered signature and gets concordant signatures from any of the 3 LINCS databases in iLINCS. This can get Overexpression, Knockdown or Chemical Perturbagen signatures.

Usage

```
getConcordants(signature, ilincsLibrary = "CP")
```

Arguments

`signature`	A data frame with the names of genes, their expression value and optionally their p-value
`ilincsLibrary`	The Library you want to search. Must be one of "OE", "KD" or "CP" for Overexpression, Knockdown or Chemical Perturbagens

Value

A tibble with the list of concordant and discordant signatures

Examples

```
# Get the L1000 signature for LINCSKD_28
kdSignature <- getSignature("LINCSKD_28")

# Get concordant gene knockdown signatures

concordant_signatures <- getConcordants(kdSignature, ilincsLibrary = "KD")

head(concordant_signatures)
```

`getSignature` *Get the L1000 Signature from iLINCS* **[Experimental]**

Description

This function acts as the entrypoint to the iLINCS database. This takes in an ID and returns the signature after making a call to the iLINCS database. The default mode for `drugfindR` is to use L1000 signatures. However, if you are trying to retrieve a different transcriptomic signature, that is also supported by setting the `l1000` parameter to `FALSE`.

Usage

```
getSignature(sigId, l1000 = TRUE)
```

Arguments

`sigId`	character. The ilincs signature_id
`l1000`	boolean. If you have a known l1000 signature

Value

a tibble with the L1000 Signature

Examples

```
# Get the L1000 signature for LINCSKD_28
kdSignature <- getSignature("LINCSKD_28")
```

`investigateSignature` *Investigate a given DGE dataset* **[Experimental]**

Description

This function takes a DGE Data frame and then finds concordant signatures to that. This generates an L1000 signature from the DGE dataset and then uploads that signature to iLINCS to find the relevant concordant (or discordant) signatures

Usage

```
investigateSignature(
  expr,
  outputLib,
  filterThreshold = NULL,
  filterProp = NULL,
  similarityThreshold = 0.2,
  paired = TRUE,
  outputCellLines = NULL,
  geneColumn = "Symbol",
  logfcColumn = "logFC",
  pvalColumn = "PValue",
  sourceName = "Input",
  sourceCellLine = "NA",
  sourceTime = "NA",
  sourceConcentration = "NA"
)
```

Arguments

`expr`	A dataframe that has differential gene expression analysis
`outputLib`	The library to search
`filterThreshold`	The Filtering threshold.
`filterProp`	The Filtering proportion.
`similarityThreshold`	The Similarity Threshold
`paired`	Logical. Whether to query iLINCS separately for up and down regulated genes
`outputCellLines`	A character vector of cell lines to restrict the output search to.
`geneColumn`	The name of the column that has gene symbols
`logfcColumn`	The name of the column that has log_2 fold-change values
`pvalColumn`	The name of the column that has p-values
`sourceName`	(Optional) An annotation column to identify the signature by name

sourceCellLine (Optional) An annotation column to specify the cell line for the input data

sourceTime (Optional) An annotation column to specify the time for the input data

sourceConcentration
(Optional) An annotation column to specify the concentration for the input data

Value

A tibble with the the similarity scores and signature metadata

Examples

```
# Investigate a signature

# Load and prepare the signature
inputSignature <- read.table(system.file("extdata",
    "dCovid_diffexp.tsv",
    package = "drugfindR"
), header = TRUE)


# Investigate the signature

investigatedSignature <- investigateSignature(inputSignature,
    outputLib = "CP",
    filterThreshold = 0.5,
    geneColumn = "hgnc_symbol",
    logfcColumn = "logFC",
    pvalColumn = "PValue"
)
```

---

investigateTarget *Investigate a Given Gene or Drug* **[Experimental]**

Description

This function takes the name of a gene or a drug and a database to use to pull signatures from and then queries iLINCS to get concordant signatures

Usage

```
investigateTarget(
  target,
  inputLib,
  outputLib,
  filterThreshold = 0.85,
  similarityThreshold = 0.321,
  paired = TRUE,
  inputCellLines = NULL,
  outputCellLines = NULL
)
```

Arguments

target	The name of the gene or drug
inputLib	One of "OE", "KD" or "CP". Marks the database to use.
outputLib	One of "OE", "KD" or "CP". Marks the database to query.
filterThreshold	The Filtering threshold.
similarityThreshold	The Similarity Threshold
paired	Logical. Whether to query iLINCS separately for up and down regulated genes
inputCellLines	A character vector of cell lines to restrict our search for input signatures to.
outputCellLines	A character vetor of cell lines to restrict the output search to.

Value

A tibble with the the similarity scores and signature metadata

Examples

```
# Search the whole iLINCS database for top concordant signatures for an
# ABL2 knockdown signature

investigatedSignature <- investigateTarget("ABL2",
    inputLib = "KD",
    outputLib = "CP",
    filterThreshold = 0.5
)
```

---

loadMetadata *Load the correct metadata table*

Description

Load the correct metadata table

Usage

```
loadMetadata(lib)
```

Arguments

lib	a string. One of "OE", "KD" or "CP"

Value

a tibble

prepareSignature *Prepare an L1000 Signature froma given differential gene expression output* **[Experimental]**

Description

This function takes a differential gene expression output from any pipeline like edgeR or DeSeq2 or any that give you the gene symbol, log_2 fold-change and p-value and transforms that into an L1000 signature for later processing.

Usage

```
prepareSignature(
  dge,
  geneColumn = "Symbol",
  logfcColumn = "logFC",
  pvalColumn = "PValue"
)
```

Arguments

dge	A dataframe-like object that has the differential gene expression information
geneColumn	The name of the column that has gene symbols
logfcColumn	The name of the column that has log_2 fold-change values
pvalColumn	The name of the column that has p-values

Value

A tibble with the L1000 signature.

Examples

```
# Prepare an L1000 signature from a differential gene expression output

inputSignature <- read.table(system.file("extdata",
    "dCovid_diffexp.tsv",
    package = "drugfindR"
), header = TRUE)

signature <- prepareSignature(inputSignature,
    geneColumn = "hgnc_symbol",
    logfcColumn = "logFC", pvalColumn = "PValue"
)

head(signature)
```

stopIfInvalidLibraries *Stop if the libraries are invalid*

Description

Stop if the libraries are invalid

Usage

```
stopIfInvalidLibraries(libs)
```

Arguments

libs	a character vector of libraries

Value

a stop if the libraries are invalid

targetRename *Rename the Target-Related Columns*

Description

Rename the Target-Related Columns

Usage

```
targetRename(inputNames)
```

Arguments

inputNames	A character vector of input_names

Value

A character vector of new names

validateLibraries *Check if the libraries input are valid*

Description

Check if the libraries input are valid

Usage

```
validateLibraries(libs)
```

Arguments

libs	a character vector of libraries

Value

a boolean

Index