Package 'DepInfeR'

November 11, 2026

Type Package

Title Inferring tumor-specific cancer dependencies through integrating ex-vivo drug response assays and drug-protein profiling

Version 1.16.0

Description
DepInfeR integrates two experimentally accessible input data matrices: the drug sensitivity profiles of cancer cell lines or primary tumors ex-vivo (X), and the drug affinities of a set of proteins (Y), to infer a matrix of molecular protein dependencies of the cancers (ß). DepInfeR deconvolutes the protein inhibition effect on the viability phenotype by using regularized multivariate linear regression. It assigns a "dependence coefficient" to each protein and each sample, and therefore could be used to gain a causal and accurate understanding of functional consequences of genomic aberrations in a heterogeneous disease, as well as to guide the choice of pharmacological intervention for a specific cancer type, sub-type, or an individual patient. For more information, please read out preprint on bioRxiv: https://doi.org/10.1101/2022.01.11.475864.

License GPL-3

Encoding UTF-8

Depends R (>= 4.2.0)

Imports matrixStats, glmnet, stats, BiocParallel

Suggests testthat (>= 3.0.0), knitr, rmarkdown, dplyr, tidyr, tibble, ggplot2, missForest, pheatmap, RColorBrewer, ggrepel, BiocStyle, ggbeeswarm

VignetteBuilder knitr

RoxygenNote 7.1.2

biocViews Software, Regression, Pharmacogenetics, Pharmacogenomics, FunctionalGenomics

BugReports https://github.com/Huber-group-EMBL/DepInfeR/issues

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

git_branch RELEASE_3_23

git_last_commit 9012d9c

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Junyan Lu [aut, cre] (ORCID: <https://orcid.org/0000-0002-9211-0746>), Alina Batzilla [aut]

Maintainer Junyan Lu <jylu1118@gmail.com>

Contents

DepInfeR-package 2
drug_response_GDSC 3
mutation_GDSC 3
processTarget 4
responseInput 5
runLASSORegression 5
targetInput 7
targetMatrix 7
targetsGDSC 8

Index **9**

`DepInfeR-package` *DepInfeR for inferring sample-specific protein dependencies*

Description

DepInfeR integrates two experimentally accessible input data matrices: the drug sensitivity profiles of cancer cell lines or primary tumors ex-vivo (X), and the drug affinities of a set of proteins (Y), to infer a matrix of molecular protein dependencies of the cancers (ß). DepInfeR deconvolutes the protein inhibition effect on the viability phenotype by using regularized multivariate linear regression. It assigns a "dependence coefficient" to each protein and each sample, and therefore could be used to gain a causal and accurate understanding of functional consequences of genomic aberrations in a heterogeneous disease, as well as to guide the choice of pharmacological intervention for a specific cancer type, sub-type, or an individual patient. For more information, please read out preprint on bioRxiv: https://doi.org/10.1101/2022.01.11.475864.

Details

The main functions are:

- `runLASSORegression` - perform inference of target importance
- `processTarget` - pre-process drug-protein affinity dataset

For detailed information on usage, see the package vignette, by typing `vignette("DepInfeR")`.

All software-related questions should be posted to the Bioconductor Support Site:

https://support.bioconductor.org

The code can be viewed at the GitHub repository. https://github.com/Huber-group-EMBL/DepInfeR

Author(s)

Alina Batzilla, Junyan Lu

References

Batzilla, A. and Lu, J. et al. (2022) Inferring tumor-specific cancer dependencies through integrating ex-vivo drug response assays and drug-protein profiling. https://www.biorxiv.org/content/10.1101/2022.01.11.475864v1

`drug_response_GDSC` *drug_response_GDSC*

Description

This is the processed Genomics of Drug Sensitivity in Cancer (GDSC) drug sensitivity dataset. The raw dataset was downloaded from https://www.cancerrxgene.org/downloads/bulk_download. The post-processing steps can be found at: https://www.huber.embl.de/users/jlu/depInfeR/process_GDSC.html.

Usage

```
data(drug_response_GDSC)
```

Format

an object of "tbl_df" (tidy table)

Examples

```
data(drug_response_GDSC)
```

`mutation_GDSC` *mutation_GDSC*

Description

This cancer type and genomic background annotation for cancer cell lines, use for the analysis of the GDSC dataset in the package vignette. The raw dataset was downloaded from https://www.cancerrxgene.org/downloads/bulk_download. The post-processing steps can be found at: https://www.huber.embl.de/users/jlu/depInfeR/process_GDSC.html.

Usage

```
data(mutation_GDSC)
```

Format

an object of "tbl_df" (tidy table)

Examples

```
data(mutation_GDSC)
```

`processTarget` *Function for pre-processing drug-protein affinity dataset*

Description

This function is used to preprocess the drug-protein affinity dataset including the following steps: - log-transform kd values (KdAsInput = TRUE) - arctan-transform log(kd) values (KdAsInput = TRUE) - check target similarity and remove highly correlated proteins (removeCorrelated = TRUE) - specify targets that should be kept in the matrix (keepTargets = NULL)

Usage

```
processTarget(
  targetsMat,
  KdAsInput = TRUE,
  removeCorrelated = TRUE,
  keepTargets = NULL,
  cutoff = 0.8
)
```

Arguments

`targetsMat`	Drug-protein affinity matrix with kd values (or optionally other affinity measurement values at roughly normal distribution). Each row is a drug and each column is a sample (cell line or tumor sample).
`KdAsInput`	A boolean value indicating whether the drug-protein affinity matrix contains kd values which should be log- and arctan-transformed. The default value is TRUE.
`removeCorrelated`	A boolean value indicating whether highly correlated proteins should be summarized in target groups. The default value is TRUE.
`keepTargets`	A character variable that specifies important proteins that should be retained in the matrix.
`cutoff`	A Cosine similarity cutoff value for clustering proteins into one target group. The value should be between 0 and 1.

Details

All steps within this function are optional depending on input data. The transformation steps should be performed if the affinity matrix consists of kd values. If there are highly correlated features within the affinity matrix, they can be removed using the provided function.

Value

A list of two elements: 1)`targetMatrix` Pre-processed drug-protein affinity matrix; 2)`targetCluster`, a list that contains the targets show high correlations with each other.

Examples

```
data(targetMatrix)
processTarget(targetsMat = targetMatrix, KdAsInput = TRUE , removeCorrelated = TRUE)
```

`responseInput` *responseInput*

Description

A toy data set that contains processed drug response matrix for examples and test of runLASSORegression function. Rows contain drugs and columns contain samples.

Usage

```
data(responseInput)
```

Format

an object of matrix

Examples

```
data(responseInput)
```

`runLASSORegression` *Main function for running LASSO regression to calculate protein dependence coefficient*

Description

This function performs multivariate linear regression with LASSO penalty and cross-validation to infer per-sample protein dependence coefficients. Please refer to the package vignette for more detailed information about this function. For the mathematical model behind this function, please refer to our preprint on bioRxiv: https://doi.org/10.1101/2022.01.11.475864.

Usage

```
runLASSORegression(
  TargetMatrix,
  ResponseMatrix,
  repeats = 100,
  BPPARAM = bpparam()
)
```

Arguments

TargetMatrix	Pre-processed drug-protein affinity matrix. Each row is a drug and each column is a protein target.
ResponseMatrix	Pre-processed drug-response viability matrix. Each row is a drug and each column is a sample (cell line or tumor sample).
repeats	A integer variable specifying the number of regression repeats. The default value is 100. A higher number can result in better stability but also takes longer time.
BPPARAM	The BiocParallel back-end. If not specified, the default MulticoreParam will be used.

Value

A list of 6 elements:

coefMat	A matrix containing the inferred protein dependency coefficient matrix. Each row is a protein and each column is a sample.
freqMat	A matrix containing the selection frequency a protein (row) for each sample (column).
lambdaList	A vector containing the lambda values selected for all repeated runs.
varExplain.all	A vector containing the R2 (variance explained) values for all repeated runs.
inputX	A copy of the input drug-protein affinity matrix.
inputY	A copy of the input drug-repsonse viability matrix.

Examples

```
data(responseInput) #load drug response matrix
data(targetInput) #load drug-target affinity matrix
runLASSORegression(TargetMatrix = targetInput, ResponseMatrix = responseInput, repeats = 5)
```

targetInput *targetInput*

Description

A toy data set that contains processed drug-target affinity matrix for examples and test of runLASSOregression function. Rows contain drugs and columns contain targets.

Usage

```
data(targetInput)
```

Format

an object of matrix

Examples

```
data(targetInput)
```

targetMatrix *targetMatrix*

Description

A toy data set that contains drug-target affinity matrix for examples and test of processTarget function. Rows contain drugs and columns contain targets.

Usage

```
data(targetMatrix)
```

Format

an object of matrix

Examples

```
data(targetMatrix)
```

targetsGDSC *targetsGDSC*

Description

This drug-protein affinity profiling data for the analysis of the GDSC dataset - a subset of the data provided by Klaeger et al. 2017. The raw data can be found in the supplementary file of the paper (Table_S1 & Table_S2): https://science.sciencemag.org/content/358/6367/eaan4368/tab-figures-data. The post-processing steps can be found at: https://www.huber.embl.de/users/jlu/depInfeR/process_kinobeads.html.

Usage

```
data(targetsGDSC)
```

Format

an object of "tbl_df" (tidy table)

Examples

```
data(targetsGDSC)
```

Index

∗ **datasets**
drug_response_GDSC, 3
mutation_GDSC, 3
responseInput, 5
targetInput, 7
targetMatrix, 7
targetsGDSC, 8
∗ **package**
DepInfeR-package, 2

DepInfeR-package, 2
drug_response_GDSC, 3

mutation_GDSC, 3

processTarget, 2, 4

responseInput, 5
runLASSORegression, 2, 5

targetInput, 7
targetMatrix, 7
targetsGDSC, 8