Package ‘oncoPredict’

July 22, 2025

Type Package

Title Drug Response Modeling and Biomarker Discovery

Version 1.2

URL https://github.com/HuangLabUMN/oncoPredict

BugReports https://github.com/HuangLabUMN/oncoPredict/issues

Maintainer Robert Gruener <rgruener@umn.edu>

Description Allows for building drug response models using screening data between bulk RNA-Seq and a drug response metric and two additional tools for biomarker discovery that have been developed by the Huang Laboratory at University of Minnesota.
There are 3 main functions within this package.
(1) calcPhenotype is used to build drug response models on RNA-Seq data and impute them on any other RNA-Seq dataset given to the model.
(2) GLDS is used to calculate the general level of drug sensitivity, which can improve biomarker discovery.
(3) IDWAS can take the results from calcPhenotype and link the imputed response back to available genomic (mutation and CNV alterations) to identify biomarkers.
Each of these functions comes from a paper from the Huang research laboratory. Below gives the relevant paper for each function.
calcPhenotype - Geeleher et al, Clinical drug response can be predicted using baseline gene expression levels and in vitro drug sensitivity in cell lines.
GLDS - Geeleher et al, Cancer biomarker discovery is improved by accounting for variability in general levels of drug sensitivity in pre-clinical models.
IDWAS - Geeleher et al, Discovering novel pharmacogenomic biomarkers by imputing drug response in cancer patients from large genomics studies.

License GPL-2

Encoding UTF-8

RoxygenNote 7.3.1

Depends R (>= 4.1.0)

biocViews sva, preprocessCore, stringr, biomaRt, genefilter, org.Hs.eg.db, GenomicFeatures, TxDb.Hsapiens.UCSC.hg19.knownGene, genefilter, TCGAbiolinks, BiocGenerics, GenomicRanges, IRanges, S4Vectors

Imports parallel, ridge, car, glmnet, pls, sva, preprocessCore, GenomicFeatures, org.Hs.eg.db, TxDb.Hsapiens.UCSC.hg19.knownGene, tidyverse, TCGAbiolinks, BiocGenerics, GenomicRanges, IRanges, S4Vectors

Suggests knitr, rmarkdown, gdata, genefilter, maftools, readxl, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Danielle Maeser [aut] (ORCID: <https://orcid.org/0000-0002-3890-887X>), Robert Gruener [aut, cre]

Repository CRAN

Date/Publication 2024-04-05 07:53:00 UTC

Contents

calcPhenotype 2
completeMatrix 4
doVariableSelection 5
glds 5
homogenizeData 6
idwas 7
map_cnv 8
summarizeGenesByMean 8

Index 10

calcPhenotype *Generate predicted drug sensitivity scores*

Description

This function predicts a phenotype (drug sensitivity score) when provided with microarray or bulk RNAseq gene expression data of different platforms. The imputations are performed using ridge regression, training on a gene expression matrix where phenotype is already known. This function integrates training and testing datasets via a user-defined procedure, and power transforming the known phenotype.

Usage

```
calcPhenotype(
  trainingExprData,
  trainingPtype,
  testExprData,
  batchCorrect,
  powerTransformPhenotype = TRUE,
  removeLowVaryingGenes = 0.2,
  minNumSamples,
  selection = 1,
  printOutput,
  pcr = FALSE,
  removeLowVaringGenesFrom,
  report_pc = FALSE,
  cc = FALSE,
  percent = 80,
  rsq = FALSE,
  folder = TRUE
)
```

Arguments

`trainingExprData`
: The training data. A matrix of expression levels. rownames() are genes, colnames() are samples (cell line names or cosmic ides, etc.). rownames() must be specified and must contain the same type of gene ids as "testExprData"

`trainingPtype`
: The known phenotype for "trainingExprData". This data must be a matrix of drugs/rows x cell lines/columns or cosmic ids/columns. This matrix can contain NA values, that is ok (they are removed in the calcPhenotype() function).

`testExprData`
: The test data where the phenotype will be estimated. It is a matrix of expression levels, rows contain genes and columns contain samples, "rownames()" must be specified and must contain the same type of gene ids as "trainingExprData".

`batchCorrect`
: How should training and test data matrices be homogenized. Choices are "eb" (default) for ComBat, "qn" for quantiles normalization or "none" for no homogenization.

`powerTransformPhenotype`
: Should the phenotype be power transformed before we fit the regression model? Default to TRUE, set to FALSE if the phenotype is already known to be highly normal.

`removeLowVaryingGenes`
: What proportion of low varying genes should be removed? 20 percent be default

`minNumSamples`
: How many training and test samples are required. Print an error if below this threshold

`selection`
: How should duplicate gene ids be handled. Default is -1 which asks the user. 1 to summarize by their or 2 to disguard all duplicates.

`printOutput`
: Set to FALSE to supress output.

`pcr`
: Indicates whether or not you'd like to use pcr for feature (gene) reduction. Options are 'TRUE' and 'FALSE'. If you indicate 'report_pc=TRUE' you need to also indicate 'pcr=TRUE'

`removeLowVaringGenesFrom`
: Determine method to remove low varying genes. Options are 'homogenizeData' and 'rawData'.

`report_pc` Indicates whether you want to output the training principal components. Options are 'TRUE' and 'FALSE'. Folder must be set to TRUE.

`cc` Indicate if you want correlation coefficients for biomarker discovery. folder must be set to TRUE

`percent` Indicate percent variability (of the training data) you'd like principal components to reflect if pcr=TRUE. Default is 80 for 80% These are the correlations between a given gene of interest across all samples vs. a given drug response across samples. These correlations can be ranked to obtain a ranked correlation to determine highly correlated drug-gene associations.

`rsq` Indicate whether or not you want to output the R^2 values for the data you train on from true and predicted values. These values represent the percentage in which the optimal model accounts for the variance in the training data. Options are 'TRUE' and 'FALSE'. folder must be set to TRUE

`folder` Indicate whether the user wants to return a folder or simply assign the calcPhenotype output. If true, run the function without assignment as it will return a folder with the results. If false, assign <- calcphenotype to save results

Value

Depends on the folder parameter. If folder = True, .txt files will be saved into a folder in your working directory automatically. The folder will include the estimated drug response values as a .txt file. Depending on the rsq, cc, report_pc parameters specified, the .txt file outputs of this function will also include the R^2 data, and the correlation coefficients and principal components are stored as .RData files for each drug in your drug dataset. If folder = 'FALSE', then only the predicted drug response values will be returned as an object.

`completeMatrix` *This function performs an iterative matrix completion algorithm to predict drug response for pre-clinical data when there are missing ('NA') values.*

Description

This function performs an iterative matrix completion algorithm to predict drug response for preclinical data when there are missing ('NA') values.

Usage

```
completeMatrix(senMat, nPerms = 50)
```

Arguments

`senMat` A matrix of drug sensitivity data with missing ('NA') values. rownames() are samples (e.g. cell lines), and colnames() are drugs.

`nPerms` The number of iterations that the EM-algorithm (expectation maximization approach) run. The default is 50, as previous findings recommend 50 iterations (https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-1050-9)

Value

A matrix of drug sensitivity scores without missing values. rownames() are samples, and colnames are drugs.

`doVariableSelection` *Remove genes with low variation.*

Description

This function performs variable selection by removing genes with the lowest variance in the datasets.

Usage

```
doVariableSelection(exprMat, removeLowVaryingGenes = 0.2)
```

Arguments

`exprMat` A matrix of gene expression levels. rownames() are genes, and colnames() are samples.

`removeLowVaryingGenes` The proportion of low varying genes to be removed.The default is .2

Value

A vector of row/genes to keep.

`glds` *This function determines drug-gene associations for pre-clinical data.*

Description

This function determines drug-gene associations for pre-clinical data.

Usage

```
glds(
  drugMat,
  drugRelatedness,
  markerMat,
  minMuts = 5,
  additionalCovariateMatrix = NULL,
  expression = NULL,
  threshold = 0.7
)
```

Arguments

`drugMat` A matrix of drug sensitivity data. rownames() are pre-clinical samples, and colnames() are drug names.

`drugRelatedness`
A matrix in which column 1 contains a list of compounds, and column 2 contains a list of their corresponding target pathways. Given the subjective nature of drug classification, please ensure these pathways are as specific as possible for accurate results.

`markerMat` A matrix containing the data for which you are looking for an association with drug sensitivity (e.g. a matrix of somatic mutation data). rownames() are marker names (e.g. gene names), and colnames() are samples.

`minMuts` The minimum number of non-zero entries required so that a p-value can be calculated (e.g. how many somatic mutations must be present). The default is 5.

`additionalCovariateMatrix`
A matrix containing covariates to be fit in the drug biomarker association models. This could be, for example, tissue of origin or cancer type. Rows are sample names. The default is NULL.

`expression` A matrix of expression data. rownames() are genes, and colnames() are the same pre-clinical samples as those in the drugMat (also in the same order). The default is NULL. If expression data is provided, a gene signature will be obtained.

`threshold` Determine the correlation coefficient. Drugs with a correlation coefficient greater than or equal to this number with the drug under scrutiny will be removed from the negative control group. The default is 0.7

`homogenizeData` *Homogenizes two expression matrices*

Description

This function takes two gene expression matrices (like trainExprMat and testExprMat) and returns homogenized versions of the matrices by employing the homogenization method specified. By default, the Combat method from the sva library is used. In both matrices, genes are row names and samples are column names. It will deal with duplicated gene names, as it subsets and orders the matrices correctly.

Usage

```
homogenizeData(
  testExprMat,
  trainExprMat,
  batchCorrect = "eb",
  selection = -1,
  printOutput = TRUE
)
```

Arguments

`testExprMat` A gene expression matrix for samples on which we wish to predict a phenotype.Genes are rows, samples are columns.

`trainExprMat` A gene expression matrix for samples for which the phenotype is already known.Genes are rows, samples are columns.

`batchCorrect` The type of batch correction to be used. Options are 'eb' for Combat, 'none', or 'qn' for quantile normalization. #The default is 'eb'.

`selection` This parameter can be used to specify how duplicates are handled. The default value of -1 means to ask the user. #Other options include '1' to summarize duplicates by their mean, and '2'to discard all duplicated genes.

`printOutput` To suppress output, set to false. Default is TRUE.

Value

A list containing two entries $train and $test, which are the homogenized input matrices.

idwas *This function will test every drug against every CNV or somatic mutation for your cancer type.*

Description

This function will test every drug against every CNV or somatic mutation for your cancer type.

Usage

```
idwas(drug_prediction, data, n = 10, cnv)
```

Arguments

`drug_prediction`
The drug prediction data. Must be a data frame. rownames are samples, colnames are drugs. Make sure sample names are of the same form as the sample names in your cnv or mutation data. e.g. if the rownames() are TCGA barcodes of the form TCGA-##-####-###, make sure your cnv/mutation data also uses samples in the form TCGA-##-####-###

`data` The cnv or mutation data. Must be a data frame. If you wish to use cnv data, use the output from map_cnv(), transpose it so that colnames() are samples. Or use data of similar form. If you wish to use mutation data, use the method for downloading mutation data outlined in the vignette, and make sure the TCGA barcodes use '-' instead of '.'; if you use another dataset (and don't download data from TCGA), make sure your data file includes the following columns: 'Variant_Classification', 'Hugo_Symbol', 'Tumor_Sample_Barcode'.

`n` The minimum number of samples you want CNVs or mutations to be amplified in. The default is 10 (arbitrarily chosen).

`cnv` TRUE or FALSE. Indicate whether or not you would like to test cnv data. If TRUE, you will test cnv data. If FALSE, you will test mutation data.

Value

Raw p-value and beta-values for cnv and somatic mutations.

map_cnv *This function maps cnv data to genes. The output of this function is a .RData file called map.RData; this file contains theCnvQuantVecList_mat (rows are genes, and columns are samples) and tumorSamps (indicates which samples are primary tumor samples, 01A).*

Description

This function maps cnv data to genes. The output of this function is a .RData file called map.RData; this file contains theCnvQuantVecList_mat (rows are genes, and columns are samples) and tumorSamps (indicates which samples are primary tumor samples, 01A).

Usage

```
map_cnv(Cnvs)
```

Arguments

`Cnvs`	The cnv data. A table with the following colnames: Sample (named using the TCGA patient barcode), Chromosome, Start, End, Num_Probes, and Segment_Mean.

Value

A .RData file called, map.RData, which stores two objects: theCnvQuantVecList_mat (rows are genes, columns are samples), tumorSamps (indicates which samples are primary tumor/01A). This output will serve as the input for test().

summarizeGenesByMean *Average over duplicate gene values*

Description

This function takes a gene expression matrix and if duplicate genes are measured, summarizes them by their means.

Usage

```
summarizeGenesByMean(exprMat)
```

Arguments

`exprMat`	A gene expression matrix with genes as rownames() and samples as colnames().

Value

A gene expression matrix that does not contain duplicate genes.

Index

∗ **CNV**
    idwas, 7
    map_cnv, 8
∗ **Drug**
    completeMatrix, 4
∗ **Homogenize**
    homogenizeData, 6
∗ **Map**
    map_cnv, 8
∗ **Summarize**
    summarizeGenesByMean, 8
∗ **Test**
    idwas, 7
∗ **and**
    calcPhenotype, 2
∗ **by**
    summarizeGenesByMean, 8
∗ **data.**
    homogenizeData, 6
∗ **data**
    idwas, 7
    map_cnv, 8
∗ **drug**
    calcPhenotype, 2
∗ **duplicate**
    summarizeGenesByMean, 8
∗ **expression**
    homogenizeData, 6
∗ **genes.**
    idwas, 7
∗ **genes**
    map_cnv, 8
    summarizeGenesByMean, 8
∗ **gene**
    homogenizeData, 6
∗ **mean.**
    summarizeGenesByMean, 8
∗ **mutation**
    idwas, 7
∗ **or**
    idwas, 7
∗ **phenotype**
    calcPhenotype, 2
∗ **prediction.**
    completeMatrix, 4
∗ **predict**
    calcPhenotype, 2
∗ **response**
    completeMatrix, 4
∗ **sensitivity**
    calcPhenotype, 2
∗ **their**
    summarizeGenesByMean, 8
∗ **to**
    idwas, 7
    map_cnv, 8

calcPhenotype, 2
completeMatrix, 4

doVariableSelection, 5

glds, 5

homogenizeData, 6

idwas, 7

map_cnv, 8

summarizeGenesByMean, 8