Package 'pathMED'

May 26, 2026

Title Scoring Personalized Molecular Portraits

Version 1.5.0

Description PathMED is a collection of tools to facilitate precision medicine studies with omics data (e.g. transcriptomics). Among its funcionalities, genesets scores for individual samples may be calculated with several methods. These scores may be used to train machine learning models and to predict clinical features on new data. For this, several machine learning methods are evaluated in order to select the best method based on internal validation and to tune the hyperparameters. Performance metrics and a ready-to-use model to predict the outcomes for new patients are returned.

Depends R (>= 4.5.0)

Suggests ada, AUCell, Biobase, BiocGenerics, BiocStyle, fgsea (>= 1.15.4), gam, GSEABase, import, kernlab, klaR, knitr, mboost, MLeval, randomForest, ranger, rmarkdown, RUnit, SummarizedExperiment, utils, xgboost

Imports BiocParallel, caret, caretEnsemble, decoupleR, ggplot2, GSVA, factoextra, FactoMineR, magrittr, matrixStats, methods, metrica, pbapply, reshape2, singscore, stats, stringi, dplyr,

VignetteBuilder knitr

biocViews Pathways, Classification, FeatureExtraction, Transcriptomics

BugReports https://github.com/jordimartorell/pathMED/issues

URL https://github.com/jordimartorell/pathMED

License GPL-2

Encoding UTF-8

LazyData false

RoxygenNote 7.3.2

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

git_branch devel

git_last_commit 23eb51c

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-05-25

Author Jordi Martorell-Marugán [cre, aut] (ORCID:
<https://orcid.org/0000-0002-5186-0735>),
Daniel Toro-Domínguez [aut] (ORCID:
<https://orcid.org/0000-0001-8440-312X>),
Raúl López-Domínguez [aut] (ORCID:
<https://orcid.org/0000-0001-8634-117X>),
Iván Ellson [aut] (ORCID: <https://orcid.org/0000-0001-6307-3141>)

Maintainer Jordi Martorell-Marugán <jmartorellm@gmail.com>

Contents

`ann2term` *Annotate the pathways from a scores matrix*

Description

Annotate the pathways from a scores matrix

Usage

```
ann2term(scoresMatrix)
```

Arguments

`scoresMatrix`	Matrix with pathways IDs as row names

Value

A data frame with the input IDs and their corresponding terms

Author(s)

Raúl López-Domínguez, <raul.lopez@genyo.es>

Jordi Martorell-Marugán, <jordi.martorell@genyo.es>

Daniel Toro-Dominguez, <danieltorodominguez@gmail.com>

References

Toro-Domínguez, D. et al (2022). *Scoring personalized molecular portraits identify Systemic Lupus Erythematosus subtypes and predict individualized drug responses, symptomatology and disease progression* . Briefings in Bioinformatics. 23(5)

See Also

getScores

Examples

```
data(pathMEDExampleData)
scoresExample <- getScores(pathMEDExampleData, geneSets = "tmod",
                           method = "GSVA")
annotatedTerms <- ann2term(scoresExample)
```

`buildRefObject` *Create a reference data object for input to the pathMED functions*

Description

Create a reference data object for input to the pathMED functions

Usage

```
buildRefObject(data, metadata = NULL, groupVar, controlGroup, use.assay = 1)
```

Arguments

`data`	A list of matrices, data frames, ExpressionSets or SummarizedExperiments with samples in columns and features in rows. A single matrix, dataframe, ExpressionSet or SummarizedExperiment may be also used.
`metadata`	A list of data frames or a single data frame with information for each sample. Samples in rows and variables in columns. If a list of ExpressionSets or SummarizedExperiments are used as @data, it is not necessary to provide @metadata.
`groupVar`	Character or list of characters indicating the column name of @metadata classifying the samples in controls and cases. If several metadata objects are provided a @groupVar can be specified for each metadata.
`controlGroup`	Character or list of characters indicating which @groupVar level corresponds to the control group, usually healthy samples. All other samples will be considered as cases, usually disease samples. If several @groupVar are provided a @controlGroup can be specified for each @groupVar
`use.assay`	If SummarizedExperiments are used, the number of the assay to extract the data.

Value

A refObject that serves as input for mScores_createReference and dissectDB functions.

Author(s)

Iván Ellson, <ivan.ellson@gmail.com >

Jordi Martorell-Marugán, <jordi.martorell@genyo.es>

Daniel Toro-Dominguez, <danieltorodominguez@gmail.com>

References

Toro-Domínguez, D. et al (2022). *Scoring personalized molecular portraits identify Systemic Lupus Erythematosus subtypes and predict individualized drug responses, symptomatology and disease progression* . Briefings in Bioinformatics. 23(5)

See Also

mScores_createReference, dissectDB

Examples

```
data(refData)

refObject <- buildRefObject(
    data = list(
        refData$dataset1, refData$dataset2,
        refData$dataset3, refData$dataset4
    ),
    metadata = list(
        refData$metadata1, refData$metadata2,
        refData$metadata3, refData$metadata4
    ),
    groupVar = "group",
    controlGroup = "Healthy_sample"
)

## Also works with a metadata for all datasets
metadata <- rbind(
    refData$metadata1, refData$metadata2,
    refData$metadata3, refData$metadata4
)
refObject <- buildRefObject(
    data = list(
        refData$dataset1, refData$dataset2,
        refData$dataset3, refData$dataset4
    ),
    metadata = metadata,
    groupVar = "group",
    controlGroup = "Healthy_sample"
)
```

dissectDB *Split pathways into coexpressed subpathways*

Description

Split pathways into coexpressed subpathways

Usage

```
dissectDB(
  refObject,
  geneSets,
  minPathSize = 10,
  minSplitSize = 3,
  maxSplits = NULL,
  explainedVariance = 60,
  percSharedGenes = 90,
  use.assay = 1
)
```

Arguments

refObject	A refObject object structure: a list of lists, each one with a cases omic matrix and controls omic matrix (named as Disease and Healthy). It can be constructed with the buildRefObject function. A list with one or more expression matrices, ExpressionSets or SummarizedExperiments without controls, can also be used. Data should be normalized and log2-transformed. Feature names must match the gene sets nomenclature. To use preloaded databases, they must be gene symbols.
geneSets	A named list with each gene set, or the name of one preloaded database (go_bp, go_cc, go_mf, kegg, reactome, pharmgkb, lincs, ctd, disgenet, hpo, wikipathways, tmod) or a GeneSetCollection.
minPathSize	numeric, minimum number of genes in a pathway to consider splitting it.
minSplitSize	numeric, minimum number of genes in a subpathway. Smaller splits will be merged with the closest coexpressed subpathway.
maxSplits	numeric, maximum number of subpathways for a pathway. If NULL (default), there is not limit.
explainedVariance	numeric, percentage of cumulative variance explained within a pathway. This parameter is used to select the number of subdivisions of a pathway that manage to explain at least the percentage of variance defined by explainedVariance.
percSharedGenes	numeric, minimum percentage of common genes across datasets to merge them before clustering. If NULL or this percentage is not reached, clustering is performed for each dataset independently and consensus subpathways are obtained from co-occurrence across datasets.
use.assay	If SummarizedExperiments are used, the number of the assay to extract the data.

Value

A list with the subpathways.

Author(s)

Jordi Martorell-Marugán, <jordi.martorell@genyo.es>

Daniel Toro-Dominguez, <danieltorodominguez@gmail.com>

References

Toro-Domínguez, D. et al (2022). *Scoring personalized molecular portraits identify Systemic Lupus Erythematosus subtypes and predict individualized drug responses, symptomatology and disease progression* . Briefings in Bioinformatics. 23(5)

See Also

buildRefObject, mScores_createReference, getScores

Examples

```
data(refData)

refObject <- buildRefObject(
    data = list(
        refData$dataset1, refData$dataset2,
        refData$dataset3, refData$dataset4
    ),
    metadata = list(
        refData$metadata1, refData$metadata2,
        refData$metadata3, refData$metadata4
    ),
    groupVar = "group",
    controlGroup = "Healthy_sample"
)

set.seed(123)
custom.tmod <- dissectDB(refObject, geneSets = "tmod")
```

---

genesetsData *Preloaded gene sets*

Description

genesetsData was constructed from the GeneCodis database (https://genecodis.genyo.es/)

Usage

```
data(genesetsData)
```

Format

An object of class `"list"` with one list per database. Each database consists on a list of gene sets, containing the gene symbols associated to it.

getScores *Calculate pathways scores for a dataset*

Description

Calculate pathways scores for a dataset

Usage

```
getScores(
  inputData,
  geneSets,
  method = "GSVA",
  labels = NULL,
  cores = 1,
  use.assay = 1,
  ...
)
```

Arguments

inputData	Matrix, data frame, ExpressionSet or SummarizedExperiment with omics data. Feature names must match the gene sets nomenclature. To use preloaded databases, they must be gene symbols.
geneSets	A named list with each gene set, or the name of one preloaded database (go_bp, go_cc, go_mf, kegg, reactome, pharmgkb, lincs, ctd, disgenet, hpo, wikipathways, tmod) or a GeneSetCollection. For using network methods, a data frame including columns: "source","target","weight" and "mor" (optional).
method	Scoring method: M-Scores, GSVA, ssGSEA, singscore, Plage, Z-score, AUCell, MDT, MLM, ORA, UDT, ULM, FGSEA, norm_FGSEA, WMEAN, norm_WMEAN, corr_WMEAN, WSUM, norm_WSUM or corr_WSUM.
labels	(Only for M-Scores) Vector with the samples class labels (0 or "Healthy" for control samples). Optional.
cores	Number of cores to be used.
use.assay	If SummarizedExperiments are used, the number of the assay to extract the data.
...	Additional parameters for the scoring functions.

Value

A list with the results of each of the analyzed regions. For each region type, a data frame with the results and a list with the probes associated to each region are generated. In addition, this list also contains the input methData, pheno and platform objects

Author(s)

Jordi Martorell-Marugán, <jordi.martorell@genyo.es>

Daniel Toro-Dominguez, <danieltorodominguez@gmail.com>

References

Toro-Domínguez, D. et al (2022). *Scoring personalized molecular portraits identify Systemic Lupus Erythematosus subtypes and predict individualized drug responses, symptomatology and disease progression* . Briefings in Bioinformatics. 23(5)

See Also

trainModel

Examples

```
data(pathMEDExampleData)
scoresExample <- getScores(pathMEDExampleData, geneSets = "tmod",
                           method = "GSVA")
```

`methodsML` *Prepare the models parameter for the trainModel function*

Description

Prepare the models parameter for the trainModel function

Usage

```
methodsML(algorithms = c("rf", "knn", "nb"), outcomeClass, tuneLength = 20)
```

Arguments

`algorithms`	Vector with one or more of these methods: 'glm', 'lm', 'lda', 'xgbTree', 'rf', 'knn', 'svmLinear', 'nnet', 'svmRadial', 'nb', 'lars','rpart', 'gamboost', 'ada', 'brnn', 'enet', or 'all' to use all algorithms
`outcomeClass`	Predicted variable type ('character' or 'numeric')
`tuneLength`	maximum number of tuning parameter combinations

Value

A list with the selected models ready to use as the 'models' parameter in the trainModel function

Author(s)

Jordi Martorell-Marugán, <jordi.martorell@genyo.es>

Daniel Toro-Dominguez, <danieltorodominguez@gmail.com>

References

Toro-Domínguez, D. et al (2022). *Scoring personalized molecular portraits identify Systemic Lupus Erythematosus subtypes and predict individualized drug responses, symptomatology and disease progression* . Briefings in Bioinformatics. 23(5)

See Also

trainModel

Examples

```
models <- methodsML(c("rf", "knn"), tuneLength = 20,
                    outcomeClass = "character")
```

`mScores_createReference` *Create a reference dataset based on M-scores*

Description

Create a reference dataset based on M-scores

Usage

```
mScores_createReference(refObject, geneSets, cores = 1)
```

Arguments

`refObject`	A refObject object structure: a list of lists, each one with a cases omic matrix and controls omic matrix (named as Disease and Healthy). It can be constructed with the buildRefObject function. Feature names must match the gene sets nomenclature. To use preloaded databases, they must be gene symbols.
`geneSets`	A named list with each gene set, or the name of one preloaded database (go_bp, go_cc, go_mf, kegg, reactome, pharmgkb, lincs, ctd, disgenet, hpo, wikipathways, tmod) or a GeneSetCollection.
`cores`	Number of cores to be used.

Value

A list with three elements. The first one is a list with the M-scores for each dataset. The second one is the geneSet used for the analysis and the third one is the input data.

Author(s)

Jordi Martorell-Marugán, <jordi.martorell@genyo.es>

Daniel Toro-Dominguez, <danieltorodominguez@gmail.com>

References

Toro-Domínguez, D. et al (2022). *Scoring personalized molecular portraits identify Systemic Lupus Erythematosus subtypes and predict individualized drug responses, symptomatology and disease progression* . Briefings in Bioinformatics. 23(5)

See Also

`mScores_imputeFromReference`, `dissectDB`, `mScores_filterPaths`, `trainModel`

Examples

```
data(refData)

refObject <- buildRefObject(
    data = list(
        refData$dataset1, refData$dataset2,
        refData$dataset3, refData$dataset4
    ),
    metadata = list(
        refData$metadata1, refData$metadata2,
        refData$metadata3, refData$metadata4
    ),
    groupVar = "group",
    controlGroup = "Healthy_sample"
)

refMscore <- mScores_createReference(refObject, geneSets = "tmod")
```

---

mScores_filterPaths *Filter pathways from the reference M-scores dataset*

Description

Filter pathways from the reference M-scores dataset

Usage

```
mScores_filterPaths(
  MRef,
  min_datasets = round(length(MRef[[1]]) * 0.34),
  perc_samples = 10,
  Pcutoff = 0.05,
  plotMetrics = TRUE
)
```

Arguments

MRef	output from the mScores_createReference function
min_datasets	number of datasets that each pathway must meet the perc_samples threshold
perc_samples	minimun percentage of samples in a dataset in which a pathway must be significant
Pcutoff	P-value cutoff for significance
plotMetrics	Plot number of significant pathways selected based on the different combination of perc_samples and min_datasets parameters

Value

A list with the selected pathways

Author(s)

Jordi Martorell-Marugán, <jordi.martorell@genyo.es>

Daniel Toro-Dominguez, <danieltorodominguez@gmail.com>

References

Toro-Domínguez, D. et al (2022). *Scoring personalized molecular portraits identify Systemic Lupus Erythematosus subtypes and predict individualized drug responses, symptomatology and disease progression* . Briefings in Bioinformatics. 23(5)

See Also

mScores_createReference

Examples

```
data(refData)

refObject <- buildRefObject(
    data = list(
        refData$dataset1, refData$dataset2,
        refData$dataset3, refData$dataset4
    ),
    metadata = list(
        refData$metadata1, refData$metadata2,
        refData$metadata3, refData$metadata4
    ),
    groupVar = "group",
    controlGroup = "Healthy_sample"
)

exampleRefMScore <- mScores_createReference(refObject, geneSets = "tmod")
relevantPaths <- mScores_filterPaths(exampleRefMScore, min_datasets = 3)
```

---

mScores_imputeFromReference

Estimate M-scores for a dataset without healthy controls

Description

Estimate M-scores for a dataset without healthy controls

Usage

```
mScores_imputeFromReference(
  inputData,
  geneSets,
  externalReference,
  nk = 5,
  distance.threshold = 30,
  cores = 1,
  use.assay = 1
)
```

Arguments

inputData	Data matrix, data frame ExpressionSet or SummarizedExperiment. Feature names must match the gene sets nomenclature. To use preloaded databases, they must be gene symbols.
geneSets	A named list with each gene set, or the name of one preloaded database (go_bp, go_cc, go_mf, kegg, reactome, pharmgkb, lincs, ctd, disgenet, hpo, wikipathways, tmod) or a GeneSetCollection.
externalReference	External reference created with the mScores_createReference function.
nk	Number of most similar samples from the external reference to impute M-scores.
distance.threshold	Only samples that do not surpass the mean Euclidean distance of distance.threshold (by default = 30) with the external reference are imputed. If NULL,impute all samples.
cores	Number of cores to be used.
use.assay	If SummarizedExperiments are used, the number of the assay to extract the data.

Value

A list with the results of each of the analyzed regions. For each region type, a data frame with the results and a list with the probes associated to each region are generated. In addition, this list also contains the input methData, pheno and platform objects

Author(s)

Jordi Martorell-Marugán, <jordi.martorell@genyo.es>

Daniel Toro-Dominguez, <danieltorodominguez@gmail.com>

References

Toro-Domínguez, D. et al (2022). *Scoring personalized molecular portraits identify Systemic Lupus Erythematosus subtypes and predict individualized drug responses, symptomatology and disease progression* . Briefings in Bioinformatics. 23(5)

See Also

mScores_filterPaths, trainModel

Examples

```
data(refData, pathMEDExampleData)

refObject <- buildRefObject(
    data = list(
        refData$dataset1, refData$dataset2,
        refData$dataset3, refData$dataset4
    ),
    metadata = list(
        refData$metadata1, refData$metadata2,
        refData$metadata3, refData$metadata4
    ),
    groupVar = "group",
```

```
    controlGroup = "Healthy_sample"
)

refMScores <- mScores_createReference(refObject,
    geneSets = "tmod", cores = 1
)

exampleMScores <- mScores_imputeFromReference(pathMEDExampleData,
    geneSets = "tmod",
    externalReference = refMScores,
    distance.threshold = 50
)
```

`pathMEDExampleData` *Example of test gene expression data*

Description

pathMEDExampleData was obtained from a dataset downloaded from NCBI GEO (GSE224705), that contains lupus patients treated with Micophenolate mofetil. The same preprocessing was done as for the datasets used to create refData. 40 patients were randomly selected, 20 samples from responding patients and 20 from non-responders.

Usage

```
data(pathMEDExampleData)
```

Format

An object of class `"data.frame"` with genes in rows and samples in columns.

`pathMEDExampleMetadata`
Metadata of test gene expression data

Description

Metadata from the dataset GSE224705. Response column conteins the information about the response and non-response to the drug for each sample.

Usage

```
data(pathMEDExampleMetadata)
```

Format

An object of class `"data.frame"` with samples in rows and variables in columns.

predictExternal *Predict conditions in external datasets*

Description

Predict conditions in external datasets

Usage

```
predictExternal(
  testData,
  model,
  realValues = NULL,
  positiveClass = NULL,
  use.assay = 1
)
```

Arguments

testData	Numerical matrix or data frame with the same features used for the model construction in rows, and the samples (new observations) in columns. An ExpressionSet may or SummarizedExperiment may also be used.
model	trainModel output or a caret-like model object
realValues	Optional, named vector (for numerical variables) or named factor (for categorical variables) with real values for each sample
positiveClass	Optional, positive class to get confusion matrix. Only needed when realValues = TRUE and for categorical variables
use.assay	If SummarizedExperiments are used, the number of the assay to extract the data.

Value

A dataframe with predictions (if realValues is not provided) or a list with the dataframe with predictions and a dataframe with the performance metrics (if realValues is provided)

Author(s)

Iván Ellson, <ivan.ellson.l@gmail.com >

Jordi Martorell-Marugán, <jordi.martorell@genyo.es>

Daniel Toro-Dominguez, <danieltorodominguez@gmail.com>

References

Toro-Domínguez, D. et al (2022). *Scoring personalized molecular portraits identify Systemic Lupus Erythematosus subtypes and predict individualized drug responses, symptomatology and disease progression* . Briefings in Bioinformatics. 23(5)

Examples

```
data(refData)

commonGenes <- intersect(rownames(refData$dataset1),
                         rownames(refData$dataset2))
dataset1 <- refData$dataset1[commonGenes, ]
dataset2 <- refData$dataset2[commonGenes, ]

scoresExample <- getScores(dataset1, geneSets = "tmod", method = "Z-score")

set.seed(123)
trainedModel <- trainModel(
    inputData = scoresExample,
    metadata = refData$metadata1,
    var2predict = "group",
    models = methodsML("svmLinear",
        outcomeClass = "character"
    ),
    Koutter = 2,
    Kinner = 2,
    repeatsCV = 1
)

externalScores <- getScores(dataset2, geneSets = "tmod", method = "Z-score")
realValues <- refData$metadata2$group
names(realValues) <- rownames(refData$metadata2)
predictions <- predictExternal(externalScores, trainedModel,
    realValues = realValues
)

print(predictions)
```

---

refData *Example of reference gene expression datasets*

Description

refData contains processed gene expression data from four datasets, including Systemic Lupus Erythematosus patients and healthy controls. Raw data for each dataset were downloaded from NCBI GEO (GSE65391, GSE45291, GSE61635, and GSE72509, respectively). Platform-dependent preprocessing was performed following established guidelines (Martorell-Marugán et al., 2021). Gene expression data were log2-transformed, and probe sets were annotated to gene symbols. To reduce computational cost in examples, 20 patient and 10 control samples were randomly selected from each dataset.

Usage

```
data(refData)
```

Format

An object of class "list" containing eight objects (dataset1-4 and metadata1-4). Each dataset is a matrix of normalized gene expression values (genes in rows, samples in columns). Each metadata is a dataframe with two columns: samples and group.

`trainModel` *Train ML models and perform internal validation*

Description

Train ML models and perform internal validation

Usage

```
trainModel(
  inputData,
  metadata = NULL,
  models = methodsML(outcomeClass = "character"),
  var2predict,
  positiveClass = NULL,
  pairingColumn = NULL,
  Koutter = 5,
  Kinner = 4,
  repeatsCV = 5,
  filterFeatures = NULL,
  filterSizes = seq(2, 100, by = 2),
  rerank = FALSE,
  continue_on_fail = TRUE,
  saveLogFile = NULL,
  use.assay = 1
)
```

Arguments

`inputData`	Numerical matrix or data frame with samples in columns and features in rows. An ExpressionSet or SummarizedExperiment may also be used.
`metadata`	Data frame with information for each sample. Samples in rows and variables in columns. If @inputData is an ExpressionSet or SummarizedExperiment, the metadata will be extracted from it.
`models`	Named list with the ML models generated with caret::caretModelSpec function. methodsML function may be used to prepare this list.
`var2predict`	Character with the column name of the @metadata to predict
`positiveClass`	Value that must be considered as positive class (only for categoric variables). If NULL, the last class by alphabetical order is considered as the positive class.
`pairingColumn`	Optional. Character with the column name of the @metadata with pairing information (e.g. technical replicates). Paired samples will always be assigned to the same set (training/test) to avoid data leakage.

Koutter	Number of outter cross-validation folds. A list of integer with elements for each resampling iteration is admitted. Each list element is a vector of integers corresponding to the rows used for training on that iteration.
Kinner	Number of innter cross-validation folds (for parameter tuning).
repeatsCV	Number of repetitions of the parameter tuning process.
filterFeatures	"rfe" (Recursive Feature Elimination), "sbf" (Selection By Filtering) or NULL (no feature selection).
filterSizes	Only for filterFeatures = "rfe". A numeric vector of integers corresponding to the number of features that should be retained.
rerank	Only for filterFeatures = "rfe". A boolean indicating if the variable importance must be re-calculated each time features are removed.
continue_on_fail	Whether or not to continue training the models if any of them fail.
saveLogFile	Path to a .txt file in which to save error and warning messages.
use.assay	If SummarizedExperiments are used, the number of the assay to extract the data.

Value

A list with four elements. The first one is the model. The second one is a table with different metrics obtained. The third one is a list with the best parameters selected in tuning process. The last element contains data for AUC plots

Author(s)

Jordi Martorell-Marugán, <jordi.martorell@genyo.es>

Daniel Toro-Dominguez, <danieltorodominguez@gmail.com>

References

Toro-Domínguez, D. et al (2022). *Scoring personalized molecular portraits identify Systemic Lupus Erythematosus subtypes and predict individualized drug responses, symptomatology and disease progression* . Briefings in Bioinformatics. 23(5)

Examples

```
data(pathMEDExampleData, pathMEDExampleMetadata)

scoresExample <- getScores(pathMEDExampleData, geneSets = "tmod",
                           method = "GSVA")

modelsList <- methodsML("svmLinear", outcomeClass = "character")

set.seed(123)
trainedModel <- trainModel(
    inputData = scoresExample,
    metadata = pathMEDExampleMetadata,
    var2predict = "Response",
    models = modelsList,
    Koutter = 2,
    Kinner = 2,
    repeatsCV = 1
)
```

Index