Package 'FORESEE'

May 17, 2018

Type Package

Title A Tool for the Systematic Comparison of Translational Drug Response Modeling Pipelines

Version 1.0.0

Author Lisa-Katrin Turnhoff <turnhoff@combine.rwth-aachen.de> and Ali Hadizadeh Esfahani <hadizadeh@combine.rwth-aachen.de>

Maintainer Lisa-Katrin Turnhoff <turnhoff@combine.rwth-aachen.de>

Description The uniFied translatiOnal dRug rESponsE prEdiction platform FORESEE is designed to act as a scaffold in developing and benchmarking translational drug sensitivity models. The package is generally geared to utilize drug sensitivity knowledge gained in cancer cell line modeling to predict clinical therapy outcome in patients. For this purpose, FORESEE includes different public cell line and patient data sets in a standardized format on the one hand and incorporates state-of-the-art preprocessing methods, model training algorithms and different validation techniques on the other hand. The modular implementation of these elements offers the training and testing of diverse combinatorial models, which can be used to re-evaluate and improve already existing modeling pipelines, but also to develop new ones.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

R topics documented:

`BlackBoxFilter` *Train a Black Box Model for Drug Efficacy Prediction*

Description

The BlackBoxFilter applies a machine learning algorithm to the feature matrix that was created from molecular data characterizing the samples of the TrainObject to create a model that is predictive of the drug response.

Usage

```
BlackBoxFilter(TrainObject, BlackBox = "ridge", nfoldCrossvalidation = 1,
  ...)
```

Arguments

`TrainObject`	Object that contains all data needed to train a model, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
`BlackBox`	Modeling algorithm for training: The function 'linear' fits a linear regression model to the training data, The function 'ridge' fits a linear ridge regression model by Cule et al. (2012) to the training data, The function 'lasso' fits a lasso regression model from the glmnet package by Friedman et al. (2008) to the training data, The function 'elasticnet' fits an elastic net regression model from the glmnet package by Friedman et al. (2008) to the training data, The function 'svm' fits a support vector regression model from the e1071 package by Meyer and Chih-Chung (2017) to the training data, The function 'rf' fits

a random forest regression model by Breiman (2001) to the training data The function 'rf_ranger' fits a fast random forest regression model by Marvin N. Wright (2018) to the training data The function 'tandem' fits a two-stage regression model by Nanne Aben (2017) to the training data. The function 'listInputOptions("BlackBoxFilter")' returns a list of the possible options. Instead of choosing one of the implemented options, a user-defined function can be used as an input.

`nfoldCrossvalidation`
: # folds to use for crossvalidation while training the model. If put to one, the complete data of the TrainObject is used for training.

Value

`ForeseeModel`
: A black box model trained on the TrainObject features that can be applied to new test data.

`TrainObject`
: The TrainObject that was used to train the model.

`CCLE` *Broad Institute Cancer Cell Line Encyclopedia or CCLE*

Description

Broad Institute Cancer Cell Line Encyclopedia, or CCLE for short, is a cell line dataset included as a ForeseeCell instance. All relevant files for the CCLE object were downloaded on May 2018 from https://portals.broadinstitute.org/ccle/data. You can check the data vignette for more information (browseVignettes(package = "FORESEE")).

Usage

```
CCLE
```

Format

An object of class `ForeseeCell` of length 14.

Source

https://portals.broadinstitute.org/ccle/data

CellorPatient *Test if the Object is a Cell or a Patient Based on the Drug Response Type*

Description

The CellorPatient Function tests if the object is a cell or a patient based on the type of the drug response annotation.

Usage

```
CellorPatient(Object)
```

Arguments

Object	FORESEE TrainObject or TestObject that should be analyzed.

Value

CellorPatient_status

The Status of the FORESEE Object that states, whether it is a cell (true) or a patient (false).

Examples

```
CellorPatient(GDSC)
CellorPatient(GSE33072_sorafenib)
```

CellResponseProcessor *Transform Drug Response Data*

Description

The CellResponseProcessor transforms the response data of the TrainObject for prediction.

Usage

```
CellResponseProcessor(TrainObject, DrugName, CellResponseType,
  CellResponseTransformation)
```

Arguments

TrainObject	Object that contains all data needed to train a model, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
DrugName	Name of the drug whose efficacy is supposed to be predicted with the model
CellResponseType	Format of the drug response data of the TrainObject, such as LN_IC50, AUC, GI50, etc., that is included in the TrainObject and to be used for prediction

CellResponseTransformation

Method that is to be used to transform the drug response data of the TrainObject: the function 'powertransform' power transforms the drug response data, the function 'logarithm' returns the natural logarithm of the drug response data, the function 'binarization_kmeans' returns a binarized drug response vector based on 2 kmeans clusters, the function 'binarization_cutoff' returns a binarized drug response vector based on a cutoff at the median, the function 'none' returns the unchanged drug response data. The function 'listInputOptions("CellResponseProcessor")' returns a list of the possible options. Instead of choosing one of the implemented options, a user-defined function can be used as an input.

Value

TrainObject The TrainObject with preprocessed drug response data.

Examples

```
CellResponseProcessor(GDSC, "Docetaxel", "LN_IC50", "binarization_cutoff")
```

CellResponseTypeAvailabilityCheck

Checking CellResponseType Availability

Description

Checking If CellResponseType is Available in Object

Usage

```
CellResponseTypeAvailabilityCheck(OBJ, RESP)
```

Arguments

OBJ	A ForeseeTrain object that you want to check the CellResponseType availability in.
RESP	CellResponseType to be checked.

Value

Returns an invisible TRUE if RESP is available in OBJ, if not available an error will be generat

Examples

```
CellResponseTypeAvailabilityCheck(DAEMEN,"GI50")
```

DAEMEN *DAEMEN Breast Cancer Cell Line Data Set*

Description

DAEMEN ForeseeCell contains the data used in Daemen et. al. 2013 publication. We downloaded the data used for modeling, as provided by the paper with the link https://www.synapse.org/#!Synapse:syn2179898. You can check the data vignette for more information (browseVignettes(package = "FORESEE")).

Usage

```
DAEMEN
```

Format

A ForeseeCell object is very similar to the list data type in R programming language; it is a data structure which includes different data types, and can be indexed using double brackets or dollar sign, for example, ForeseeCell\$variable1 or ForeseeCell[["variable1"]] or ForeseeCell[[1]].

Available components in DAEMEN are:

GeneExpression A matrix, with gene Entrez IDs in rows and cell lines in columns, containing RMA-normalized gene expression profiles measured by DNA array.

GeneExpressionRNAseq Matrix of counts based on RNA-seq technology. We transformed the seq values into logarithmic (base 2) scale for having semi-normal distributed values, which is necessary for linear modeling. As a prerequisite for transforming to log-scale, we replaced all values lower than 1 with 1.

Methylation and SNP6 Directly imported into matrices from downloaded files.

GI50 A matrix with cell lines in rows and drugs in columns. A description about GI50 can be found in ResponseTypes component.

TissueInfo A data frame with tissue-related information about cell lines in the dataset. Column 'Site' of this component is used to extract relevant samples that user set by assigning a value to 'TrainingTissue' input in ForeseeTrain(). Hence, this component is needed if user wants to use a specific tissue for 'TrainingTissue'.

InputTypes InputTypes is a data frame with two columns of 'Name' and 'Description', which provide the names of all components in the object that can be used as input data (in ForeseeTrain for example), and description for each input data.

ResponseTypes ResponseTypes is another two-column data frame with a 'Name' column, providing the names of all components in the object that are a measure of drug activity and can be used as response variable (called 'CellResponseType' in ForeseeTrain) and a 'Description' column for each response variable.

Source

https://www.ncbi.nlm.nih.gov/pubmed/24176112

DuplicationHandler *Remove Duplicated Gene Names from a FORESEE Object*

Description

DuplicationHandler finds duplicates in the gene names (features) from the FORESEE Object and summarizes or deletes them according to the user's preferences.

Usage

```
DuplicationHandler(Object, DuplicationHandling)
```

Arguments

Object	FORESEE Object (ForeseeCell or ForeseeTrain) that contains all data needed to train a model, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
DuplicationHandling	Method for handling duplicates of gene names. The function 'mean' calculates the mean of all rows that have the same gene name, The function 'first' chooses the first hit of duplicated genes and discards the rest of genes with the same name, The function 'none' removes all gene names that occur more than once. The function 'listInputOptions("DuplicationHandler")' returns a list of the possible options. Instead of chosing one of the implemented options, a user-defined function can be used as an input.

Value

Object	The object with unique gene names

Examples

```
DuplicationHandler(GDSC,"first")
```

EGEOD18864 *EGEOD18864 Patient Data Set*

Description

EGEOD18864 (identical to GSE18864 on GEO) is a gene expression response dataset of 24 patients to Cisplatin treatment. You can check the data vignette for more information (browseVignettes(package = "FORESEE")).

Usage

```
EGEOD18864
```

Format

A ForeseePatient is a data structure having components of different data types, that can be indexed using double brackets or dollar sign (for example ForeseePatient\$variable1 or ForeseePatient[["variable1"]] or ForeseePatient[[1]]), similar to a list data type in R programming language.

Available components in EGEOD18864 are:

GeneExpression is a matrix, with genes in rows and patients in columns. Entrez IDs are saved in 'rownames' of the matrix and patient identifiers in 'colnames'.

Raw CEL files were downloaded from Array Express, and normalized using RMA from affy package.

Annotation is a logical or numeric vector indicating the patient response to a drug. Extra information is provided in names(Annotation), e.g. when Annotation is a logical vector, names(Annotation) provides information about what True and False in Annotation mean in terms of patient response.

ExtraAnnotation is a data frame including all annotations that was contained in the original patient data set. This component is not used in the FORESEE pipeline, but is included for the user (e.g. to divide a patient data set into sub groups based on ExtraAnnotation for better modeling).

Source

https://www.ebi.ac.uk/arrayexpress/experiments/E-GEOD-18864/

`FeatureCombiner` *FeatureCombiner*

Description

The Feature Combiner combines all selected Input Data Types into one Feature Matrix

Usage

```
FeatureCombiner(TrainObject, TestObject, InputDataTypes)
```

Arguments

`TrainObject`	Object that contains all data needed to train a model, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
`TestObject`	Object that contains all data that the model is to be tested on, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
`InputDataTypes`	Data types of the TrainObject that are to be used to train the model, such as "GeneExpression", "Mutation", "CopyNumberVariation", "Methylation", "CancerType", etc.

Value

TrainObject	The TrainObject with a new Feature matrix combining all specified input data types and a Featuretype Vector indicating the molecular data type of each feature
TestObject	The TestObject with a new Feature matrix combining all specified input data types and a Featuretype Vector indicating the molecular data type of each feature

Examples

```
FeatureCombiner(GDSC,GSE6434,c("GeneExpression", "Mutation"))
```

FeaturePreprocessor *Preprocess the Gene Expression Inputs of both TrainObject and TestObject for Modeling Drug Response*

Description

The FeaturePreprocessor converts the original gene expression features into predictive features with a function defined by FeaturePreprocessing.

Usage

```
FeaturePreprocessor(TrainObject, TestObject, FeaturePreprocessing)
```

Arguments

TrainObject	Object that contains all data needed to train a model, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
TestObject	Object that contains all data that the model is to be tested on, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
FeaturePreprocessing	Method for preprocessing the inputs of the model: The function 'zscore_genewise' calculates the zscore normalizing each gene over all samples, The function 'zscore_samplewise' calculates the zscore normalizing each sample over all genes, The function 'pca' does principal component analysis, The function 'physio' does physiospace analysis with the samples using cell line gene expression of the gdsc data base as physiological references, The function 'none' keeps the gene expression values unchanged, The function 'listInputOptions("FeaturePreprocessor")' returns a list of the possible options. Instead of chosing one of the implemented options, a user-defined function can be used as an input.

Value

TrainObject	The TrainObject with preprocessed features.
TestObject	The TestObject with preprocessed features.

Examples

```
FeaturePreprocessor(GDSC,GSE6434,"zscore_genewise")
```

FeatureSelector *Select the Genes that are used as Model Features*

Description

The FeatureSelector selects a subset of features from all genes to be used for drug efficacy prediction.

Usage

```
FeatureSelector(TrainObject, TestObject, GeneFilter, DrugName)
```

Arguments

TrainObject	Object that contains all data needed to train a model, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
TestObject	Object that contains all data that the model is to be tested on, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
GeneFilter	Set of genes to be considered for training the model, such as all, a certain percantage based on variance or p-value, specific gene sets like landmark genes, gene ontologies or pathways, etc. The option 'variance' removes the 20 The option 'pvalue' removes the 20 The option 'landmarkgenes' uses the L1000 gene set downloaded from CLUE Command App The option 'ontology' uses a specific set of genes included in the ontology associated with the drug The option 'pathway' uses a specific set of genes included in the pathway associated with the drug The option 'all' keeps all genes as features The function 'listInputOptions("FeatureSelector")' returns a list of the possible options. Instead of choosing one of the implemented options, a user-defined function can be used as an input. If the user inserts a list as an input, the list is considered as chosen features.

Value

TrainObject	The TrainObject with the selected gene set as features.
TestObject	The TestObject with homogenized features according to the chosen TrainObject.

Examples

```
FeatureSelector(GDSC,GSE6434,"variance","Docetaxel")
FeatureSelector(GDSC,GSE6434,"pathway","Tamoxifen")
```

Foreseer — *Apply the trained ForeseeModel on a TestObject*

Description

The Foreseer applies the ForeseeModel that was trained on the features of a FORESEE TrainObject to the test data to gain a prediction of the TestObject's drug response.

Usage

```
Foreseer(TestObject, ForeseeModel, BlackBox)
```

Arguments

TestObject	Object that contains all data that the model is to be tested on, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
ForeseeModel	Model that has been trained on a TrainObject with the function ForeseeTrain().

Value

Foreseen	Predicted drug response of the samples listed in the TestObject obtained by applying the ForeseeModel to the features of the TestObject.

ForeseeTest — *Test a Drug Efficacy Prediction Model on a TestObject*

Description

ForeseeTest applies the machine learning based model ForeseeModel that has been trained on the features of a FORESEE TrainObject to a FORESEE TestObject to evaluate the Predictability of Drug Efficacy. First, the Foreseer applies the ForeseeModel to the test data to gain the predicted response 'Foreseen'. Second, the Validator evaluates the performance of the model by comparing the predicted response 'Foreseen' to the reported, true response.

Usage

```
ForeseeTest(TestObject, ForeseeModel, Evaluation = "rocauc",
  BlackBox = "ridge")
```

Arguments

TestObject	Object that contains all data that the model is to be tested on, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
ForeseeModel	Model that has been trained on a TrainObject with the function ForeseeTrain.

`Evaluation` Measure for evaluating the model performance. The option 'fpvalue' calculates the p value of an F test on a linear model between predictions and the actual annotations, The option 'mse' calculates the mean square error of a linear model between predictions and the actual annotations, The option 'pearson' calculates the pearson correlation between predictions and the actual annotations, The option 'prauc' calculates the AUC of the precision-recall curve The option 'rocauc' calculates the AUC of the ROC curve The option 'rocpvalue' calculates the t.test p value of the ROC curve AUC versus 10000 permutated annotation values, The option 'rsquared' calculates the fraction of variance explained by a linear model between predictions and actual annotations, The option 'rsquared_adjusted' calculates the fraction of variance explained by a linear model between predictions and actual annotations, corrected the p-value of F-test, The option 'spearman'. calculates the spearman correlation between predictions and the actual annotations. The function 'listInputOptions("Validator")' returns a list of the possible options. Instead of choosing one of the implemented options, a user-defined function can be used as an input.

`BlackBox` BlackBox used for training ForeseeModel.

Value

`Performance` Evaluation Measure of the Predictability of the ForeseeModel trained on the TrainObject and tested on the TestObject.

`Foreseen` Predicted drug response of the TestObject obtained by applying the ForeseeModel to the molecular data of the TestObject.

`ForeseeTrain` *Train a Drug Efficacy Prediction Model*

Description

ForeseeTrain uses the data of the TrainObject to train a black box model that can later be applied to new data in order to predict drug efficacy. The CellResponseProcessor prepares the response data of the TrainObject for prediction. Duplicates in the gene names are removed using the Foresee DuplicationHandler. The Homogenizer function reduces batch effects between train and test data. The FeatureSelector restricts the input features of the TrainObject to a specific set tht is to be used for the model. The FeaturePreprocessor converts the original features into predictive features. The Sample Selector restricts the training samples to those of a specific tissue. The FeatureCombiner combines features of all specified data types into one feature matrix. The BlackBox applies a machine learning algorithm to the preprocessed data to train a model that is predictive of drug response.

Usage

```
ForeseeTrain(TrainObject, TestObject, DrugName, CellResponseType,
  CellResponseTransformation = "powertransform",
  InputDataTypes = "GeneExpression", TrainingTissue = "all",
  TestingTissue = "all", DuplicationHandling = "first",
  HomogenizationMethod = "ComBat", GeneFilter = "all",
  FeaturePreprocessing = "none", BlackBox = "ridge",
  nfoldCrossvalidation = 1, ...)
```

Arguments

TrainObject	Object that contains all data needed to train a model, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
TestObject	Object that contains all data that the model is to be tested on, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
DrugName	Name of the drug whose efficacy is supposed to be predicted with the model. You can get all possible values with listDrugs(OBJ) or listInputOptions("DrugName", OBJ), where OBJ is the object you want to use as TrainObject.
CellResponseType	Format of the drug response data of the TrainObject, such as IC50, AUC, GI50, etc. You can get all possible values with listInputOptions("CellResponseType", OBJ), where OBJ is the object you want to use as TrainObject.
CellResponseTransformation	Method that is to be used to transform the drug response data of the TrainObject, such as power transform, logarithm, binarization, user defined functions, etc. Get all possible values with listInputOptions("CellResponseProcessor").
InputDataTypes	Data types of the TrainObject that are to be used to train the model, such as GeneExpression, Mutation, CopyNumberVariation, Methylation, Cancertype, etc. You can get all possible values with listInputOptions("InputDataTypes", OBJ), where OBJ is the object you want to use as TrainObject.
TrainingTissue	Tissue type that the cell lines of the TrainObject should be of, such as pancreas or lung. Default is "all" for pancancer analysis. You can get all possible values with listInputOptions("TrainingTissue", OBJ), where OBJ is the object you want to use as TrainObject.
TestingTissue	Tissue type that the cell lines or samples of the TestObject should be of, such as pancreas or lung. Default is "all" for analysis of all samples. You can get all possible values with listInputOptions("TestingTissue", OBJ), where OBJ is the object you want to use as TestObject.
DuplicationHandling	Method for handling duplicates of gene names, such as considering none, the mean, the first hit, etc. Get all possible values with listInputOptions("DuplicationHandler").
HomogenizationMethod	Method for homogenizing data of the TrainObject and TestObject, such as ComBat, quantile normalization, limma, RUV, etc. Get all possible values with listInputOptions("Homogenizer").
GeneFilter	Set of genes to be considered for training the model, such as all, a certain percantage based on variance or p-value, specific gene sets like landmark genes, gene ontologies or pathways, etc. Get all possible values with listInputOptions("FeatureSelector").
FeaturePreprocessing	Method for preprocessing the inputs of the model, such as z-score, principal component analysis, PhysioSpace similarity, etc. Get all possible values with listInputOptions("FeaturePreprocessor").
BlackBox	Modeling algorithm for training, such as linear regression, elastic net, lasso regression, ridge regression, tandem, support vector machines, random forests, user defined functions, etc. Get all possible values with listInputOptions("BlackBoxFilter").

`nfoldCrossvalidation`
: # folds to use for crossvalidation while training the model. If put to one, the complete data of the TrainObject is used for training.

Value

`ForeseeModel`
: A black box model trained on the TrainObject data that can be applied to new test data.

`TrainObject`
: The TrainObject with preprocessed and filtered features.

`TestObject`
: The TestObject with preprocessed and filtered features.

GAO *GAO Xenograft Data Set*

Description

GAO is one of the two xenograft data sets included in FORESEE. The data is downloaded as supplement files of Gao et. al. 2015 (https://www.ncbi.nlm.nih.gov/pubmed/26479923), which are freely available and can be downloaded via https://media.nature.com/original/nature-assets/nm/journal/v21/n11/extref/nm.3954 S2.xlsx. You can check the data vignette for more information (browseVignettes(package = "FORESEE")).

Usage

```
GAO
```

Format

A ForeseeCell object is very similar to the list data type in R programming language; it is a data structure which includes different data types, and can be indexed using double brackets or dollar sign, for example, ForeseeCell\$variable1 or ForeseeCell[["variable1"]] or ForeseeCell[[1]].

Available components in GAO are:

GeneExpression A matrix of RNA-seq values in FPKM (Fragments Per Kilobase of transcript per Million). The matrix contains genes as rows and samples as columns, with Entrez IDs in rownames. FPKM values were transformed into logarithmic scale (base 2) for having semi-normal distributed values, which is necessary for linear modeling.

SNP6 Copy number data measured by SNP array (Affymetrix genome-wide human SNP Array 6.0 chip)

Mutation, CNVGain and CNVLoss Binary matrices, pointing toward mutations, gaining copy number variations and losing copy number variations respectively.

BestResponse, BestResponseCombo, TimeToDouble, ... This data set includes 14 different response matrices, all of which have samples in rows and drugs in columns. List and description of these response matrices can be found in ResponseTypes component.

TissueInfo A data frame with tissue-related information about cell lines in the dataset. Column 'Site' of this component is used to extract relevant samples that user set by assigning a value to 'TrainingTissue' input in ForeseeTrain(). Hence, this component is needed if user wants to use a specific tissue for 'TrainingTissue'.

DrugInfo A data frame with extra information about the included drugs in the dataset. Columns 'DRUG_NAME' and 'TARGET' from this component are used in FeatureSelector.ontology() and FeatureSelector.pathway() and are needed if the user wants to use any of the mentioned FeatureSelector methods, where the pipeline uses only the gene names for training the model that are contained in the ontology or pathway associated with the chosen drug.

InputTypes InputTypes is a data frame with two columns of 'Name' and 'Description', which provide the names of all components in the object that can be used as input data (in ForeseeTrain for example), and description for each input data.

ResponseTypes ResponseTypes is another two-column data frame with a 'Name' column, providing the names of all components in the object that are a measure of drug activity and can be used as response variable (called 'CellResponseType' in ForeseeTrain) and a 'Description' column for each response variable.

Source

https://www.ncbi.nlm.nih.gov/pubmed/26479923

GDSC *Genomics of Drug Sensitivity in Cancer or GDSC*

Description

Genomics of Drug Sensitivity in Cancer, or GDSC for short, is one of the ForeseeCell datasets available in the FORESEE package. All files related to the GDSC dataset were downloaded on 25.4.2018 from https://www.cancerrxgene.org/downloads. You can check the data vignette for more information (browseVignettes(package = "FORESEE")).

Usage

```
GDSC
```

Format

A ForeseeCell object is very similar to the list data type in R programming language; it is a data structure which includes different data types, and can be indexed using double brackets or dollar sign, for example, ForeseeCell\$variable1 or ForeseeCell[["variable1"]] or ForeseeCell[[1]].

Available components in GDSC are:

GeneExpression RMA normalized DNA array values were converted into an R matrix, with genes in rows and cell lines in columns. Column names that were originally COSMIC IDs were converted to cell line names, and row names were converted from Ensembl gene IDs to Entrez IDs using biomaRt.

CNVGain, CNVLoss, Mutation and Methylation Four different binary matrices, with genes in rows and cell lines in columns, all extracted from supplement Table S3B of the paper Iorio et. al. 2016. Gene names were converted from symbols to Entrez IDs.

LN_IC50, AUC ,RMSE ,Z_SCORE ,MAX_CONC_MICROMOLAR , MIN_CONC_MICROMOLAR Response data were all in one data frame. We rearranged drug responses into different matrices, with cell lines as rows and drugs as columns.

DrugInfo A data frame with extra information about the included drugs in the dataset. Columns 'DRUG_NAME' and 'TARGET' from this component are used in FeatureSelector.ontology() and FeatureSelector.pathway() and are needed if the user wants to use any of the mentioned FeatureSelector methods, where the pipeline uses only the gene names for training the model that are contained in the ontology or pathway associated with the chosen drug.

TissueInfo A data frame with tissue-related information about cell lines in the dataset. Column 'Site' of this component is used to extract relevant samples that user set by assigning a value to 'TrainingTissue' input in ForeseeTrain(). Hence, this component is needed if user wants to use a specific tissue for 'TrainingTissue'.

CelllineInfo A data frame with extra information about cell lines in the dataset.

InputTypes InputTypes is a data frame with two columns of 'Name' and 'Description', which provide the names of all components in the object that can be used as input data (in ForeseeTrain for example), and description for each input data.

ResponseTypes ResponseTypes is another two-column data frame with a 'Name' column, providing the names of all components in the object that are a measure of drug activity and can be used as response variable (called 'CellResponseType' in ForeseeTrain) and a 'Description' column for each response variable.

Source

https://www.cancerrxgene.org/downloads

GetCellResponseData *Get Cell Line Drug Response from FORESEE Object*

Description

The GetCellResponseData Function extracts the entries of the drug response data that are relevant to predict the drug response with regard to the user's choice of CellResponseType and DrugName. It returns the given input object with a new element "DrugResponse".

Usage

```
GetCellResponseData(TrainObject, DrugName, CellResponseType)
```

Arguments

TrainObject	Object that contains all data needed to train a model, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
DrugName	Name of the drug whose efficacy is supposed to be predicted with the model
CellResponseType	Format of the drug response data of the TrainObject, such as LN_IC50, AUC, GI50, etc., that is used for prediction

Value

TrainObject	The TrainObject with extracted drug response data.

Examples

```
GetCellResponseData(GDSC,"Gemcitabine","AUC")
```

GSE33072_erlotinib *GSE33072_erlotinib Patient Data Set*

Description

GSE33072_erlotinib (subset of GSE33072 on GEO) is a gene expression response dataset of 25 patients to Erlotinib treatment. You can check the data vignette for more information (browseVignettes(package = "FORESEE")).

Usage

```
GSE33072_erlotinib
```

Format

A ForeseePatient is a data structure having components of different data types, that can be indexed using double brackets or dollar sign (for example ForeseePatient\$variable1 or ForeseePatient[["variable1"]] or ForeseePatient[[1]]), similar to a list data type in R programming language.

Available components in GSE33072_erlotinib are:

GeneExpression is a matrix, with genes in rows and patients in columns. Entrez IDs are saved in 'rownames' of the matrix and patient identifiers in 'colnames'.

Raw CEL files were downloaded from GEO, and normalized using RMA from affy package.

Annotation is a logical or numeric vector indicating the patient response to a drug. Extra information is provided in names(Annotation), e.g. when Annotation is a logical vector, names(Annotation) provides information about what True and False in Annotation mean in terms of patient response.

ExtraAnnotation is a data frame including all annotations that was contained in the original patient data set. This component is not used in the FORESEE pipeline, but is included for the user (e.g. to divide a patient data set into sub groups based on ExtraAnnotation for better modeling).

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE33072

GSE33072_sorafenib *GSE33072_sorafenib Patient Data Set*

Description

GSE33072_sorafenib (subset of GSE33072 on GEO) is a gene expression response dataset of 39 patients to Sorafenib treatment. You can check the data vignette for more information (browseVignettes(package = "FORESEE")).

Usage

```
GSE33072_sorafenib
```

Format

A ForeseePatient is a data structure having components of different data types, that can be indexed using double brackets or dollar sign (for example ForeseePatient\$variable1 or ForeseePatient[["variable1"]] or ForeseePatient[[1]]), similar to a list data type in R programming language.

Available components in GSE33072_sorafenib are:

GeneExpression is a matrix, with genes in rows and patients in columns. Entrez IDs are saved in 'rownames' of the matrix and patient identifiers in 'colnames'.
Raw CEL files were downloaded from GEO, and normalized using RMA from affy package.

Annotation is a logical or numeric vector indicating the patient response to a drug. Extra information is provided in names(Annotation), e.g. when Annotation is a logical vector, names(Annotation) provides information about what True and False in Annotation mean in terms of patient response.

ExtraAnnotation is a data frame including all annotations that was contained in the original patient data set. This component is not used in the FORESEE pipeline, but is included for the user (e.g. to divide a patient data set into sub groups based on ExtraAnnotation for better modeling).

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE33072

GSE6434 *GSE6434 Patient Data Set*

Description

GSE6434 is a gene expression response dataset of 24 patients to Docetaxel treatment. You can check the data vignette for more information (browseVignettes(package = "FORESEE")).

Usage

```
GSE6434
```

Format

A ForeseePatient is a data structure having components of different data types, that can be indexed using double brackets or dollar sign (for example ForeseePatient\$variable1 or ForeseePatient[["variable1"]] or ForeseePatient[[1]]), similar to a list data type in R programming language.

Available components in GSE6434 are:

GeneExpression is a matrix, with genes in rows and patients in columns. Entrez IDs are saved in 'rownames' of the matrix and patient identifiers in 'colnames'.
Raw CEL files were downloaded from GEO, and normalized using RMA from affy package.

Annotation is a logical or numeric vector indicating the patient response to a drug. Extra information is provided in names(Annotation), e.g. when Annotation is a logical vector, names(Annotation) provides information about what True and False in Annotation mean in terms of patient response.

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE6434

GSE9782_GPL96_bortezomib

GSE9782_GPL96_bortezomib Patient Data Set

Description

GSE9782_GPL96_bortezomib (subset of GSE9782 on GEO, measured by Affymetrix Human Genome U133A Array) is a gene expression response dataset of 169 patients to Bortezomib treatment. You can check the data vignette for more information (browseVignettes(package = "FORESEE")).

Usage

```
GSE9782_GPL96_bortezomib
```

Format

A ForeseePatient is a data structure having components of different data types, that can be indexed using double brackets or dollar sign (for example ForeseePatient\$variable1 or ForeseePatient[["variable1"]] or ForeseePatient[[1]]), similar to a list data type in R programming language.

Available components in GSE9782_GPL96_bortezomib are:

GeneExpression is a matrix, with genes in rows and patients in columns. Entrez IDs are saved in 'rownames' of the matrix and patient identifiers in 'colnames'.

Raw CEL files were not available at GEO, so we downloaded the MAS5.0 normalized values from GEO and transformed the data to a logarithmic scale (base 2).

Annotation is a logical or numeric vector indicating the patient response to a drug. Extra information is provided in names(Annotation), e.g. when Annotation is a logical vector, names(Annotation) provides information about what True and False in Annotation mean in terms of patient response.

ExtraAnnotation is a data frame including all annotations that was contained in the original patient data set. This component is not used in the FORESEE pipeline, but is included for the user (e.g. to divide a patient data set into sub groups based on ExtraAnnotation for better modeling).

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE9782

GSE9782_GPL96_dexamethasone

GSE9782_GPL96_dexamethasone Patient Data Set

Description

GSE9782_GPL96_dexamethasone (subset of GSE9782 on GEO, measured by Affymetrix Human Genome U133A Array) is a gene expression response dataset of 70 patients to Dexamethasone treatment. You can check the data vignette for more information (browseVignettes(package = "FORESEE")).

Usage

```
GSE9782_GPL96_dexamethasone
```

Format

A ForeseePatient is a data structure having components of different data types, that can be indexed using double brackets or dollar sign (for example ForeseePatient\$variable1 or ForeseePatient[["variable1"]] or ForeseePatient[[1]]), similar to a list data type in R programming language.

Available components in GSE9782_GPL96_dexamethasone are:

GeneExpression is a matrix, with genes in rows and patients in columns. Entrez IDs are saved in 'rownames' of the matrix and patient identifiers in 'colnames'.

Raw CEL files were not available at GEO, so we downloaded the MAS5.0 normalized values from GEO and transformed the data to a logarithmic scale (base 2).

Annotation is a logical or numeric vector indicating the patient response to a drug. Extra information is provided in names(Annotation), e.g. when Annotation is a logical vector, names(Annotation) provides information about what True and False in Annotation mean in terms of patient response.

ExtraAnnotation is a data frame including all annotations that was contained in the original patient data set. This component is not used in the FORESEE pipeline, but is included for the user (e.g. to divide a patient data set into sub groups based on ExtraAnnotation for better modeling).

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE9782

GSE9782_GPL97_bortezomib

GSE9782_GPL97_bortezomib Patient Data Set

Description

GSE9782_GPL97_bortezomib (subset of GSE9782 on GEO, measured by Affymetrix Human Genome U133B Array) is a gene expression response dataset of 169 patients to Bortezomib treatment. You can check the data vignette for more information (browseVignettes(package = "FORESEE")).

Usage

```
GSE9782_GPL97_bortezomib
```

Format

A ForeseePatient is a data structure having components of different data types, that can be indexed using double brackets or dollar sign (for example ForeseePatient\$variable1 or ForeseePatient[["variable1"]] or ForeseePatient[[1]]), similar to a list data type in R programming language.

Available components in GSE9782_GPL97_bortezomib are:

GeneExpression is a matrix, with genes in rows and patients in columns. Entrez IDs are saved in 'rownames' of the matrix and patient identifiers in 'colnames'.

Raw CEL files were not available at GEO, so we downloaded the MAS5.0 normalized values from GEO and transformed the data to a logarithmic scale (base 2).

Annotation is a logical or numeric vector indicating the patient response to a drug. Extra information is provided in names(Annotation), e.g. when Annotation is a logical vector, names(Annotation) provides information about what True and False in Annotation mean in terms of patient response.

ExtraAnnotation is a data frame including all annotations that was contained in the original patient data set. This component is not used in the FORESEE pipeline, but is included for the user (e.g. to divide a patient data set into sub groups based on ExtraAnnotation for better modeling).

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE9782

`GSE9782_GPL97_dexamethasone`

GSE9782_GPL97_dexamethasone Patient Data Set

Description

GSE9782_GPL97_dexamethasone (subset of GSE9782 on GEO, measured by Affymetrix Human Genome U133B Array) is a gene expression response dataset of 70 patients to Dexamethasone treatment. You can check the data vignette for more information (browseVignettes(package = "FORESEE")).

Usage

```
GSE9782_GPL97_dexamethasone
```

Format

A ForeseePatient is a data structure having components of different data types, that can be indexed using double brackets or dollar sign (for example ForeseePatient\$variable1 or ForeseePatient[["variable1"]] or ForeseePatient[[1]]), similar to a list data type in R programming language.

Available components in GSE9782_GPL97_dexamethasone are:

GeneExpression is a matrix, with genes in rows and patients in columns. Entrez IDs are saved in 'rownames' of the matrix and patient identifiers in 'colnames'.

Raw CEL files were not available at GEO, so we downloaded the MAS5.0 normalized values from GEO and transformed the data to a logarithmic scale (base 2).

Annotation is a logical or numeric vector indicating the patient response to a drug. Extra information is provided in names(Annotation), e.g. when Annotation is a logical vector, names(Annotation) provides information about what True and False in Annotation mean in terms of patient response.

ExtraAnnotation is a data frame including all annotations that was contained in the original patient data set. This component is not used in the FORESEE pipeline, but is included for the user (e.g. to divide a patient data set into sub groups based on ExtraAnnotation for better modeling).

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE9782

`Homogenizer` *Homogenize a Pair of two FORESEE Objects*

Description

The Homogenizer function reduces batch effects and homogenizes the data of a given TrainObject and TestObject.

Usage

```
Homogenizer(TrainObject, TestObject, HomogenizationMethod)
```

Arguments

`TrainObject`	Object that contains all data needed to train a model, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
`TestObject`	Object that contains all data that the model is to be tested on, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
`HomogenizationMethod`	Method for homogenizing data of the TrainObject and TestObject. The function 'ComBat' uses the batch effect removal ComBat of the sva package, The function 'quantile' uses the quantile normalization of the preprocessCore package, The function 'limma' uses the removeBatchEffect function of the limma package, The function 'YuGene' uses the function of the YuGene package by Le Cao, The function 'RUV' regresses out unwanted variation using the 10 principal components of negative control genes (here: list of human housekeeping by Eisenberg and Levanon (2013)) The function 'RUV4' regresses out unwanted variation using the ruv package, The function 'none' does not do any batch effect correction. The function 'listInputOptions("Homogenizer")' returns a list of the possible options. Instead of choosing one of the implemented options, a user-defined function can be used as an input.

Value

`TrainObject`	The TrainObject with homogenized features according to the chosen TestObject.
`TestObject`	The TestObject with homogenized features according to the chosen TrainObject.

Examples

```
Homogenizer(GDSC,GSE6434,"quantile")
```

listCellLines *List All Cell lines Inside a ForeseeTrain Object*

Description

listCellLines returns all cell lines (or sample names in case of a xenograft data set) available in a ForeseeTrain Object (all cell lines that are included in gene expression matrix of ForeseeTrain Object).

Usage

```
listCellLines(OBJ)
```

Arguments

OBJ	A ForeseeTrain object of which you want to extract its containing cell lines.

Value

Character vector of all cell line names in OBJ.

Examples

```
listCellLines(GDSC)
listCellLines(WITKIEWICZ)
```

listDrugs *List All Drugs Inside a ForeseeTrain Object*

Description

listDrugs returns all possible drugs that were tested and have reponses available in a ForeseeTrain Object.

Usage

```
listDrugs(OBJ)
```

Arguments

OBJ	A ForeseeTrain object that you want to extract all possible drugs it has response information on.

Value

Character vector of all drug names in OBJ.

Examples

```
listDrugs(GDSC)
listDrugs(GAO)
```

`listInputOptions` *List Input Options for FORESEE Methods*

Description

listInputOptions returns possible input arguments in ForeseeTrain (for DrugName, CellResponseType, InputDataTypes, TrainingTissue, TestingTissue, CellResponseTransformation, DuplicationHandling, HomogenizationMethod, GeneFilter, FeaturePreprocessing and BlackBox arguments) and in ForeseeTest (for the Evaluation argument).

Usage

```
listInputOptions(FunArgument, OBJ)
```

Arguments

`FunArgument`	Character string of the input argument (for DrugName, CellResponseType, InputDataTypes, TrainingTissue and TestingTissue) or the function corresponding to the input of ForeseeTrain (CellResponseTransformation, DuplicationHandling, HomogenizationMethod, GeneFilter, FeaturePreprocessing and BlackBox) or ForeseeTest (Evaluation). Check the help of ForeseeTrain and ForeseeTest for more information.
`OBJ`	A ForeseeTrain or ForeseeTest object that you want to extract options for. Only necessary for when FunArgument is DrugName, CellResponseType, InputDataTypes, TrainingTissue or TestingTissue.

Value

`Character vector of all possible inputs`

Examples

```
listInputOptions("CellResponseProcessor")
listInputOptions("DuplicationHandler")
listInputOptions("Homogenizer")
listInputOptions("Validator")
listInputOptions("DrugName", CCLE)[1:10]
listInputOptions("InputDataTypes", GAO)
listInputOptions("CellResponseType", GDSC)
listInputOptions("TrainingTissue", CCLE)
listInputOptions("TestingTissue", GSE33072_erlotinib)
```

requireForesee *Loading/Attaching (and Installing) a Package*

Description

requireForesee is the same as 'require' from the base package, except in case of a missing package, it tries to install it via 'biocLite' from Bioconductor. For installation to work, R needs to have access to the internet (more precisely "https://bioconductor.org/biocLite.R" should be accessable to R).

Usage

```
requireForesee(package)
```

Arguments

package	the name of a package to be loaded and attached (and installed).

Value

requireForesee returns (invisibly) a logical indicating whether the required package was available (before installation attempts).

Examples

```
requireForesee(ranger)
```

SampleSelector *SampleSelector*

Description

The Sample Selector returns features of only those samples that have a drug response for the drug specified by the user's input for DrugName and are available for all chosen input data types. If a specific tissue is selected, the samples are reduced to cell lines of that tissue only.

Usage

```
SampleSelector(TrainObject, TrainingTissue, InputDataTypes)
```

Arguments

TrainObject	Object that contains all data needed to train a model, including molecular data (such as gene expression, mutation, copy number variation, methylation, cancer type, etc.) and drug response data
TrainingTissue	Tissue type that the cell lines of the TrainObject should be of, such as "lung". Default should be "all" for pancancer analysis.
InputDataTypes	Data types of the TrainObject that are to be used to train the model, such as "GeneExpression", "Mutation", "CopyNumberVariation", "Methylation", "CancerType", etc.

Value

`TrainObject` The TrainObject with only those samples that have the specified drug response and are available for all chosen input data types.

Examples

```
SampleSelector(GDSC, "pancreas", "GeneExpression")
SampleSelector(GDSC, "all", c("GeneExpression","Mutation"))
```

Validator *Validate the Performance of a ForeseeModel on a new TestObject*

Description

The Validator evaluates the performance of the model by comparing the predicted response Foreseen to the reported, true response from the TestObject's Annotation.

Usage

```
Validator(Foreseen, TestObject, Evaluation)
```

Arguments

`Foreseen`	Predicted drug response of the TestObject obtained by applying the ForeseeModel.
`TestObject`	Object that contains all data that the model is to be tested on, especially the true, measured drug response.
`Evaluation`	Measure for evaluating the model performance, such as ROC-Curve, AUC or p-value of ROC-Curve, Rsquared, MSE, Correlation, F-Test, etc. The option 'fp-value' calculates the p value of an F test on a linear model between predictions and the actual annotations, The option 'mse' calculates the mean square error of a linear model between predictions and the actual annotations, The option 'pearson' calculates the pearson correlation between predictions and the actual annotations, The option 'prauc' calculates the AUC of the precision-recall curve The option 'rocauc' calculates the AUC of the ROC curve The option 'rocp-value' calculates the t.test p value of the ROC curve AUC versus 10000 permutated annotation values, The option 'rsquared' calculates the fraction of variance explained by a linear model between predictions and actual annotations, The option 'rsquared_adjusted' calculates the fraction of variance explained by a linear model between predictions and actual annotations, corrected the p-value of F-test, The option 'spearman'. calculates the spearman correlation between predictions and the actual annotations. The function 'listInputOptions("Validator")' returns a list of the possible options. Instead of choosing one of the implemented options, a user-defined function can be used as an input.

Value

`Performance` Evaluation Measure of the Predictability of the ForeseeModel trained on the TrainObject and tested on the TestObject.

Examples

```
Validator(rep(1,length(GSE6434$Annotation)),GSE6434,"rocauc")
```

WITKIEWICZ *WITKIEWICZ Xenograft Data Set*

Description

WITKIEWICZ is the other xenograft data set included in FORESEE. This data set is from a study by Witkiewcz et al. 2016, studying Pancreatic ductal adenocarcinoma (PDAC) drug response. Data used in building the WITKIEWICZ ForeseeCell are two excel files, which are included as supplements in the original paper and the GEO data set GSE84023, which includes the RNA-seq gene expression relevant to the paper.

Usage

```
WITKIEWICZ
```

Format

A ForeseeCell object is very similar to the list data type in R programming language; it is a data structure which includes different data types, and can be indexed using double brackets or dollar sign, for example, ForeseeCell\$variable1 or ForeseeCell[["variable1"]] or ForeseeCell[[1]].

Available components in WITKIEWICZ are:

GeneExpression Matrix of gene expressions measured by RNA-seq. We used the already processed data available on GEO. Based on GSE84023 page on GEO, this is the processing pipeline they used: "Illumina Casava1.7 software used for basecalling. Sequenced reads were trimmed for adaptor sequence, mapped to hg19 genome using bowtie TopHat. Counts per gene was obtained using HTseq counts and normalized using edgeR package in R. Genome_build: hg19. files_format_andy_content: tab-delimited text file include matrix of normalized log counts per million for each sample." We averaged over all samples from the same patient.

AUC and AUCCombo Response data were imported from supplement excel files, and then formatted as a matrix with samples in rows and drugs in columns. More information about these two response matrices can be found in ResponseTypes component. In AUC, samples from the same patient are averaged.

DrugInfo A data frame with extra information about the included drugs in the dataset. Columns 'DRUG_NAME' and 'TARGET' from this component are used in FeatureSelector.ontology() and FeatureSelector.pathway() and are needed if the user wants to use any of the mentioned FeatureSelector methods, where the pipeline uses only the gene names for training the model that are contained in the ontology or pathway associated with the chosen drug.

InputTypes InputTypes is a data frame with two columns of 'Name' and 'Description', which provide the names of all components in the object that can be used as input data (in ForeseeTrain for example), and description for each input data.

ResponseTypes ResponseTypes is another two-column data frame with a 'Name' column, providing the names of all components in the object that are a measure of drug activity and can be used as response variable (called 'CellResponseType' in ForeseeTrain) and a 'Description' column for each response variable.

Details

REMEMBER listDrugs(OBJ = WITKIEWICZ) or listInputOptions(FunArgument = "DrugName", OBJ = WITKIEWICZ)) only return drugs of single treatment (acceptable as DrugName in ForeseeTrain when CellResponseType="AUC"), if you want to have CellResponseType="AUCCombo" in ForeseeTrain, list of possible "DrugName" values can be listed by colnames(WITKIEWICZ$AUCCombo).

You can check the data vignette for more information (browseVignettes(package = "FORESEE")).

Source

https://www.ncbi.nlm.nih.gov/pubmed/27498862

Index