

Package ‘LundTax2023Classifier’

September 30, 2024

Title Predictor of Lund Taxonomy molecular subtypes from gene-expression data

Version 1.1.4

Description

The package implements a Random Forest rule-based single-sample predictor that classifies transcriptomic samples into the 5 (or 7, including subclasses) Lund Taxonomy molecular subtypes.

The final classifier is composed of two separate predictors applied sequentially - first a sample is classified as one of the 5 main classes (Uro, GU, BaSq, Mes or ScNE), and then samples classified as

Uro are subclassified into UroA, UroB or UroC by a second predictor.

Depends R (>= 2.10)

Imports circlize,
ComplexHeatmap,
dplyr,
ggplot2,
ggpubr,
grDevices,
multiclassPairs,
openxlsx,
tibble

License GPL (>= 2) + file LICENSE

Encoding UTF-8

LazyData true

LazyDataCompression xz

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

R topics documented:

LundTax2023Classifier-package	2
classifier_grade3	2
classifier_hg	3
classifier_lundtax_5c	4
classifier_lundtax_7c	4
gene_list	5
int_calc_immune_proportions	6

int_calc_score	6
int_check_ties	7
int_merge_suburo_matrix	8
int_predict_grade	8
int_ratio_score	9
lundtax_calc_sigscore	10
lundtax_predict_sub	11
lund_colors	13
plot_hm_scores	14
plot_hm_signatures	15
plot_ranked_score	17
signatures	19
sjodahl_2017	19
sjodahl_2017_meta	20

Index	21
--------------	-----------

LundTax2023Classifier-package
LundTax2023

Description

This packages implements a Random Forest rule-based single-sample predictor that classifies transcriptomic samples into the 5 (or 7, including subclasses) Lund Taxonomy molecular subtypes. The final classifier is composed of two separate predictors applied sequentially: first a sample is classified as one of the 5 main classes (Uro, GU, BaSq, Mes or ScNE), and then samples classified as Uro are subclassified into UroA, UroB or UroC by a second predictor

Author(s)

Maintainer: Elena Aramendía <elena.aramendia_cotillas@med.lu.se>

Authors:

- Pontus Eriksson <pontus.eriksson@med.lu.se>
- Adam Mattsson <adam.mattsson@med.lu.se>

classifier_grade3	<i>Classifier Grade 3.</i>
-------------------	----------------------------

Description

Classifier generated with `multiclassPairs::predict_RF()`, this dataset is needed for predicting the grade (1, 2, or 3) and calculating scores for incoming data.

Usage

```
data(classifier_grade3)
```

Format

A list of 6.

Details

A Large rule_based_RandomForest object. A list of 6.

- genes. Genes in ensembl format.
- rules. A set of rules for the classifier in ensembl format.
- TrainingMatrix. Binary matrix for the rules in the training data.
- boruta. Boruta results for the classifier.
- RF_classifier. Random forest classifier details.
- calls. Information on how the model was generated.

classifier_hg	<i>Classifier High Grade.</i>
---------------	-------------------------------

Description

Classifier generated with `multiclassPairs::predict_RF()`, this dataset is needed for predicting the grade (high grade/lowgrade) and calculating scores for incoming data.

Usage

```
data(classifier_hg)
```

Format

A list of 6.

Details

A Large rule_based_RandomForest object. A list of 6.

- genes. Genes in ensembl format.
- rules. A set of rules for the classifier in ensembl format.
- TrainingMatrix. Binary matrix for the rules in the training data.
- boruta. Boruta results for the classifier.
- RF_classifier. Random forest classifier details.
- calls. Information on how the model was generated.

classifier_lundtax_5c *Classifier LundTax 5c.*

Description

Classifier as a 'rule_based_RandomForest' object. Predicts samples as one of the 5 main Lund Taxonomy molecular subtypes, Uro, GU, BaSq, Mes, or ScNE. Object includes the final RF classifier, the used genes and rules in the final model, the Boruta results, and the training matrix. The training matrix is a binary matrix containing the rule values for the training data and it is used for imputation purposes during the prediction if values are missing in the sample. This object was generated using the `multiclassPairs::predict_RF()` function.

Usage

```
data(classifier_lundtax_5c)
```

Format

A list of 6.

Details

A Large rule_based_RandomForest object. A list of 6.

- genes. Genes in hgnc format.
- rules. A set of rules for the classifier in hgnc format.
- TrainingMatrix. Binary matrix for the rules in the training data.
- boruta. Boruta results for the classifier.
- RF_classifier. Random forest classifier details.
- calls. Information on how the model was generated.

classifier_lundtax_7c *Classifier LundTax 7c.*

Description

Classifier as a rule_based_RandomForest object. Predicts samples as one of the 3 Uro subclasses, UroA, UroB, or UroC. Object includes the final RF classifier, the used genes and rules in the final model, the Boruta results, and the training matrix. The training matrix is a binary matrix containing the rule values for the training data and it is used for imputation purposes during the prediction if values are missing in the sample. This object was generated using the `multiclassPairs::predict_RF()` function.

Usage

```
data(classifier_lundtax_7c)
```

Format

A list of 6.

Details

A Large rule_based_RandomForest object. A list of 6.

- genes. Genes in hgnc format.
- rules. A set of rules for the classifier in hgnc format.
- TrainingMatrix. Binary matrix for the rules in the training data.
- boruta. Boruta results for the classifier.
- RF_classifier. Random forest classifier details.
- calls. Information on how the model was generated.

gene_list	<i>Gene List.</i>
-----------	-------------------

Description

Gene annotations in hgnc and ensembl format. Features, for the classifier relevant genes. For convenience, both formats are available for seamless conversion of IDs, regardless of the format of the incoming data.

Usage

```
data(gene_list)
```

Format

A data frame with 1900 rows (genes) and 2 columns (hgnc_symbol and ensembl_gene_id).

Details

A data frame with gene information in different formats.

- ensembl_gene_id. Gene annotation in ensembl format.
- hgnc_symbol. Gene annotation in HUGO format.

int_calc_immune_proportions

Calculate Immune Proportions.

Description

Calculate immune score proportions.

Usage

```
int_calc_immune_proportions(immune_results)
```

Arguments

`immune_results` Required parameter. Data frame with immune scores from [int_calc_score\(\)](#).

Details

Internal function called by [int_calc_score\(\)](#), if variable is set to immune. Not meant for out-of-package use.

Value

A data frame with scores as percentages.

int_calc_score

Calculate Scores.

Description

Calculate immune and infiltration scores.

Usage

```
int_calc_score(
  this_data = NULL,
  variable = NULL,
  log_transform = TRUE,
  gene_id = "hgnc_symbol",
  adjust = TRUE,
  adj_factor = 5.1431,
  verbose = TRUE
)
```

Arguments

this_data	Required parameter. Data frame or matrix with expression values.
variable	Required parameter. Input should be one of the following; immune, score141up, proliferation, or progression.
log_transform	Boolean parameter. If TRUE (default), the function log transforms the incoming expression values.
gene_id	Specify the type of gene identifier used in this_data. Accepted values are; hgnc_symbol (default) or ensembl_gene_id.
adjust	Boolean parameter. If TRUE, the function will proceed with adjusting the scores based on stable genes. If FALSE (default), no adjustment will be made and the original score values will be retained.
adj_factor	Only applicable if adjust is set to TRUE. Allows users to apply a proportional adjustment to the normalized scores, enabling finer control over the final output values. After dividing each score by the mean expression of stable genes, the result is multiplied by this factor. Default is 5.1431
verbose	A logical value indicating whether processing messages will be printed or not. Default is TRUE.

Details

Internal function called by `lundtax_calc_sigscore()`. Not meant for out of package use. Takes a data frame of matrix with expression values and calculates scores based on gene expression.

Value

A data frame with scores for the selected variable.

int_check_ties	<i>Check Ties.</i>
----------------	--------------------

Description

Check score ties between predictions scores.

Usage

```
int_check_ties(first, last)
```

Arguments

first	Predictions when setting ties.method = "first"
last	Predictions when setting ties.method = "last"

Details

Internal function called by `lundtax_calc_sigscore()`. Not meant for out of package use. Checks score ties in the prediction scores and prints a message indicating the sample where the tie occurred, the two subtypes with the tied scores and the subtype that is reported in the output object.

Value

Nothing.

int_merge_suburo_matrix

Merge subUro Matrix.

Description

Merge prediction score matrices from two classifiers.

Internal function called by [lundtax_calc_sigscore\(\)](#). Not meant for out of package use. This function merges the prediction score matrices from the 5-class and 7-class (UroA,UroB,UroC) classifiers into 1 unique score matrix.

Usage

```
int_merge_suburo_matrix(
  score_matrix1,
  score_matrix2,
  row.names = list(NULL, NULL)
)
```

Arguments

score_matrix1 Prediction score matrix from the 7 class classifier.
 score_matrix2 Prediction score matrix from the 5 class classifier.
 row.names Rownames (sample names) for both matrices.

Value

Merged matrix including scores for 8 classes (Uro, UroA, UroB, UroC, GU, BaSq, Mes, ScNE)

int_predict_grade

Predict Grade

Description

Function for predicting grade using random forest and the bundled classifiers.

Usage

```
int_predict_grade(
  this_data = NULL,
  grade_predictor = NULL,
  gene_id = "hgnc_symbol",
  impute = FALSE,
  impute_reject = 0.67,
  impute_kNN = 5,
  verbose = TRUE
)
```


Arguments

this_data	Required parameter. Data frame or matrix with expression values.
grade_predictor	Required parameter, the predictor needed for grading. Should be one of the following bundled classifiers (classifier_GRADE3 or classifier_HG).
gene_id	Specify the type of gene identifier used in this_data. Accepted values are; hgnc_symbol (default) or ensembl_gene_id.
impute	From <code>multiclassPairs::predict_RF()</code> . Boolean. To determine if missed genes and NA values should be imputed or not. The non missed rules will be used to determine the closest samples in the training binary matrix (i.e. which is stored in the classifier object). For each sample, the mode value for nearest samples in the training data will be assigned to the missed rules. Default is FALSE.
impute_reject	From <code>multiclassPairs::predict_RF()</code> . A number between 0 and 1 indicating the threshold of the missed rules in the sample. Based on this threshold the sample will be rejected (i.e. skipped if higher than the impute_reject threshold) and the missed rules will not be imputed in this sample. Default is 0.67. NOTE, The results object will not have any results for this sample.
impute_kNN	From <code>multiclassPairs::predict_RF()</code> . Integer determines the number of the nearest samples in the training data to be used in the imputation. Default is 5. It is not recommended to use large number (i.e. >10).
verbose	A logical value indicating whether processing messages will be printed or not. Default is TRUE.

Details

Internal function called by `lundtax_calc_sigscore()`. Not meant for out of package use. This function is internally calling `multiclassPairs::predict_RF()` to run a random forest prediction using one of the two bundled classifiers for grade prediction.

Value

A data frame with grade results.

int_ratio_score	<i>Calculate Scores.</i>
-----------------	--------------------------

Description

Calculate proliferation and progression scores.

Usage

```
int_ratio_score(
  this_data = NULL,
  variable = NULL,
  gene_id = "hgnc_symbol",
  verbose = TRUE
)
```

Arguments

this_data	Required parameter. Data frame or matrix with expression values.
variable	Required parameter. Input should be one of the following; proliferation, or progression.
gene_id	Specify the type of gene identifier used in this_data. Accepted values are; hgnc_symbol (default) or ensembl_gene_id.
verbose	A logical value indicating whether processing messages will be printed or not. Default is TRUE.

Details

Internal function called by `lundtax_calc_sigscore()`. Not meant for out of package use. Takes a data frame of matrix with expression values and calculates scores based on gene expression.

Value

A data frame with scores for the selected variable.

lundtax_calc_sigscore *Calculate Lund Taxonomy Signature Scores*

Description

Wrapper function for calculating Lund Taxonomy scores.

Usage

```
lundtax_calc_sigscore(
  this_data = NULL,
  gene_id = "hgnc_symbol",
  threshold_progression = 0.58,
  log_transform = TRUE,
  adjust = FALSE,
  adj_factor = 5.1431,
  impute = FALSE,
  impute_reject = 0.67,
  impute_kNN = 5,
  verbose = TRUE
)
```

Arguments

this_data	Required parameter. Data frame or matrix with expression values.
gene_id	Specify the type of gene identifier used in this_data. Accepted values are; hgnc_symbol (default) or ensembl_gene_id.
threshold_progression	Threshold to flag a sample as high risk of progression, default is 0.58.
log_transform	Boolean parameter. If TRUE (default), the function log transforms the incoming expression values.

adjust	Boolean parameter. If TRUE, the function will proceed with adjusting the scores based on stable genes. If FALSE (default), no adjustment will be made and the original score values will be retained.
adj_factor	Only applicable if adjust is set to TRUE. Allows users to apply a proportional adjustment to the normalized scores, enabling finer control over the final output values. After dividing each score by the mean expression of stable genes, the result is multiplied by this factor.
impute	From <code>multiclassPairs::predict_RF()</code> . Boolean. To determine if missed genes and NA values should be imputed or not. The non missed rules will be used to determine the closest samples in the training binary matrix (i.e. which is stored in the classifier object). For each sample, the mode value for nearest samples in the training data will be assigned to the missed rules. Default is FALSE.
impute_reject	From <code>multiclassPairs::predict_RF()</code> . A number between 0 and 1 indicating the threshold of the missed rules in the sample. Based on this threshold the sample will be rejected (i.e. skipped if higher than the impute_reject threshold) and the missed rules will not be imputed in this sample. Default is 0.67. NOTE, The results object will not have any results for this sample.
impute_kNN	From <code>multiclassPairs::predict_RF()</code> . Integer determines the number of the nearest samples in the training data to be used in the imputation. Default is 5. It is not recommended to use large number (i.e. >10).
verbose	A logical value indicating whether processing messages will be printed or not. Default is TRUE.

Details

This function internally calls a set of non-exported scoring functions starting with the `int_` prefix. See individual function documentation for more information on the individual function calls.

Value

A data frame with scores for the selected variable.

Examples

```
#load packages
library(dplyr, multiclassPairs)

#get scores for bundled data
my_scores = lundtax_calc_sigscore(this_data = sjodahl_2017,
                                impute = TRUE)
```

lundtax_predict_sub	<i>Lund Taxonomy Predict Subtype</i>
---------------------	--------------------------------------

Description

Predict Lund Taxonomy subtypes based on rule-based Random Forest classifiers.

Usage

```

lundtax_predict_sub(
  this_data = NULL,
  gene_id = "hgnc_symbol",
  threshold_progression = 0.58,
  log_transform = TRUE,
  adjust = FALSE,
  adj_factor = 5.1431,
  impute = FALSE,
  impute_reject = 0.67,
  impute_kNN = 5,
  subtype_only = FALSE,
  include_data = FALSE,
  include_pred_scores = TRUE,
  verbose = TRUE
)

```

Arguments

<code>this_data</code>	Required parameter. Data frame or matrix with expression values.
<code>gene_id</code>	Specify the type of gene identifier used in <code>this_data</code> . Accepted values are; <code>hgnc_symbol</code> (default) or <code>ensembl_gene_id</code> .
<code>threshold_progression</code>	Threshold to flag a sample as high risk of progression, default is 0.58.
<code>log_transform</code>	Boolean parameter. If TRUE, the function log transforms the incoming expression values. Default is FALSE.
<code>adjust</code>	Boolean parameter. If TRUE, the function will proceed with adjusting the scores based on stable genes. If FALSE (default), no adjustment will be made and the original score values will be retained.
<code>adj_factor</code>	Only applicable if <code>adjust</code> is set to TRUE. Allows users to apply a proportional adjustment to the normalized scores, enabling finer control over the final output values. After dividing each score by the mean expression of stable genes, the result is multiplied by this factor.
<code>impute</code>	From <code>multiclassPairs::predict_RF()</code> . Boolean. To determine if missed genes and NA values should be imputed or not. The non missed rules will be used to determine the closest samples in the training binary matrix (i.e. which is stored in the classifier object). For each sample, the mode value for nearest samples in the training data will be assigned to the missed rules. Default is FALSE.
<code>impute_reject</code>	From <code>multiclassPairs::predict_RF()</code> . A number between 0 and 1 indicating the threshold of the missed rules in the sample. Based on this threshold the sample will be rejected (i.e. skipped if higher than the <code>impute_reject</code> threshold) and the missed rules will not be imputed in this sample. Default is 0.67. NOTE, The results object will not have any results for this sample.
<code>impute_kNN</code>	From <code>multiclassPairs::predict_RF()</code> . Integer determines the number of the nearest samples in the training data to be used in the imputation. Default is 5. It is not recommended to use large number (i.e. >10).
<code>subtype_only</code>	Boolean parameter. Set to TRUE to return subtypes and nothing else. Default is FALSE.

`include_data` Boolean parameter. Set to TRUE to include data in output, FALSE is default.

`include_pred_scores` Boolean parameter. Set to TRUE (default) to include prediction scores for each sample and class in output.

`verbose` A logical value indicating whether processing messages will be printed or not. Default is TRUE.

Details

This function uses 2 classifiers to classify the samples: 5-class classifier first classifies samples into Uro, GU, BaSq, Mes or ScNE. Samples classified as Uro receive a second classification as UroA, B or C by the second classifier. This function internally calls [lundtax_calc_sigscore\(\)](#) for retrieving signature scores.

Value

Returns a list object including: Data (optional, not included by default), Prediction scores for all classes (optional, included by default), Predicted LundTax class for 7-class system, Predicted Lund-Tax class for 5-class system

Examples

```
sjodahl_predicted = lundtax_predict_sub(this_data = sjodahl_2017,
                                         impute = TRUE)
```

lund_colors	<i>Lund Colors.</i>
-------------	---------------------

Description

Standardized colors used for Lund Taxonomy subtypes and datasets.

Usage

```
data(lund_colors)
```

Format

A list of 4.

Details

A list of 4 with color palettes frequently used in this package.

- `lund_colors`. Color palette for the LundTax subtypes.
- `lund_colors_transp`. Same as `lund_colors` but transparent.
- `stage_colors`. Color palette for stages.
- `dataset_colors`. Color palette for the LundTax datasets.

plot_hm_scores	<i>Plot Signature Scores.</i>
----------------	-------------------------------

Description

Build a heatmap with scores retrieved with the `lundtax_predict_sub()` function.

Usage

```
plot_hm_scores(
  these_predictions = NULL,
  out_path = NULL,
  out_format = "png",
  return_scores = FALSE,
  to_xlsx = FALSE,
  title = NULL,
  hm_split = NULL,
  subtype_annotation = "5_class",
  hm_cluster = FALSE,
  plot_anno_legend = NULL,
  plot_hm_legend = FALSE,
  plot_width = 14,
  plot_height = 6,
  plot_font_size = 12,
  verbose = TRUE
)
```

Arguments

<code>these_predictions</code>	A list with a data frame object called scores. Returned with <code>lundtax_predict_sub()</code> .
<code>out_path</code>	Optional, set path to export plot. If not provided, tidy version of incoming scores in data frame format will be returned (<code>return_scores</code> will be auto-defaulted to TRUE).
<code>out_format</code>	Required parameter if <code>out_path</code> is specified. Can be "png" (default) or "pdf". if pdf, the returned pdf will be in A4 (horizontal) format, if png is specified the user can control the dimensions, see <code>plot_width</code> and <code>plot_height</code> .
<code>return_scores</code>	Set to TRUE to return prediction scores in a tidy format. Default is FALSE.
<code>to_xlsx</code>	Boolean parameter, set to TRUE to export score data frame in xlsx format. Default is FALSE. If set to TRUE, the spreadsheet will be saved to the same path as the heatmap.
<code>title</code>	Required parameter. Heatmap title, will also be pasted to the exported file(s) as well as a new column in the scores data frame under cohort.
<code>hm_split</code>	Optional parameter for controlling how the data is split into different groups. If not provided, the function will split on what is specified within <code>subtype_annotation</code> .
<code>subtype_annotation</code>	Can be one of the following; "5_class" (default) or "7_class" annotation.
<code>hm_cluster</code>	Boolean parameter, set to TRUE to cluster the rows (default is FALSE).

plot_anno_legend	Expects a vector with TRUE/FALSE (7 in total), this decides what legends will be on the final heatmap. Default is to only show the legend for the subtypes.
plot_hm_legend	Boolean parameter. Set to TRUE to show heatmap legend. Default is FALSE.
plot_width	This parameter controls the width in inches. Default is 14(4200 pixels at 300 PPI).
plot_height	This parameter controls the height in inches. Default is 6(1800 pixels at 300 PPI)
plot_font_size	Optional parameter to control the size of the font in the generated heatmap. Note, the title of the plot will always be twice that of the set value here (default = 12).
verbose	Set to TRUE for debugging purposes. Default is FALSE.

Details

Construct and export (pdf or png) a highly customizable heatmap visualizing prediction scores for each sample and class, predicted with `lundtax_predict_sub()`. This function depends on Complexheatmap. It is also possible to return a data frame with prediction scores in a tidy format. To do so, set `return_scores = TRUE`. For a greater explanation on how to use the function, see parameter descriptions and examples.

Value

Data frame with prediction score for each sample and class, if `return_scores = TRUE`. Otherwise, nothing.

Examples

```
## Not run:
my_predictions = lundtax_predict_sub(these_predictions = sjodahl_2017,
                                   gene_id = "hgnc_symbol",
                                   impute = TRUE,
                                   adjust = TRUE)

plot_hm_scores(these_predictions = my_predictions,
               out_path = "../",
               out_format = "pdf",
               title = "Lund2017")

## End(Not run)
```

plot_hm_signatures *Plot Heatmap Signatures.*

Description

Plot heatmap for classification results.

Usage

```

plot_hm_signatures(
  these_predictions = NULL,
  this_data = NULL,
  gene_id = "hgnc_symbols",
  subtype_annotation = "5_class",
  norm = TRUE,
  plot_scores = TRUE,
  show_ann_legend = FALSE,
  show_hm_legend = FALSE,
  ann_height = 8,
  title = "My Plot",
  plot_width = 14,
  plot_height = 10,
  plot_font_size = 10,
  plot_font_row_size = 8,
  out_path = NULL,
  out_format = "png"
)

```

Arguments

these_predictions	Required parameter, should be the output from lundtax_predict_sub() .
this_data	Expression data used for predictions. Required if the output from lundtax_predict_sub() is run with include_data = FALSE (default).
gene_id	Specify the type of gene identifier used in this_data. Accepted values are; hgnc_symbol (default) or ensembl_gene_id.
subtype_annotation	Can be one of the following; "5_class" (default) or "7_class" annotation.
norm	Boolean parameter. Set to TRUE (default) to normalize the data into Z-scaled values.
plot_scores	Boolean parameter. Set to TRUE (default) to plot prediction scores for each class.
show_ann_legend	Boolean parameter, set to TRUE to show annotation legend (Lund classes). Default is FALSE.
show_hm_legend	Boolean parameter, set to TRUE to show heatmap legend, default is FALSE.
ann_height	Plotting parameter, optional. Annotation height in cm. Default = 8.
title	Plotting parameter. The title for the generated heatmap. Default is "My Plot".
plot_width	This parameter controls the width in inches. Default is 14 (4200 pixels at 300 PPI).
plot_height	This parameter controls the height in pixels. Default is 10 (3000 pixels at 300 PPI)
plot_font_size	Optional parameter to control the size of the font in the generated heatmap. Note, the title of the plot will always be twice that of the set value here (default = 10).

plot_font_row_size	Optional parameter to control the size of the font in the generated heatmap. Note, the title of the plot will always be twice that of the set value here (default = 8).
out_path	Optional, set path to export plot.
out_format	Required parameter if out_path is specified. Can be "png" (default) or "pdf". The user can control the dimensions with plot_width and plot_height.

Details

This function plots a heatmap including genes and signatures of interest, with prediction results and scores on top.

Value

Draws heatmap and silently returns the sample order.

Examples

```
## Not run:
#example 1 including data in results object
#run predictor on the bundled expression data
sjodahl_predicted = lundtax_predict_sub(this_data = sjodahl_2017,
                                       include_data = TRUE,
                                       impute = TRUE)

#example 2 - 5 class annotation and Without prediction scores
plot_hm_signatures(these_predictions = sjodahl_predicted,
                   subtype_annotation = "5_class",
                   ann_height = 0.5,
                   plot_scores = FALSE)

## End(Not run)
```

plot_ranked_score	<i>Plot Ranked Score</i>
-------------------	--------------------------

Description

Return a point plot with ranked scores for a set variable.

Usage

```
plot_ranked_score(
  these_predictions = NULL,
  this_data = NULL,
  this_score = NULL,
  this_subtype = NULL,
  subtype_class = NULL,
  plot_title = NULL,
```

```

    return_plot_data = FALSE
)

```

Arguments

<code>these_predictions</code>	Required parameter, if no data provided with <code>this_data</code> . Predictions returned with <code>lundtax_predict_sub()</code> .
<code>this_data</code>	Optional parameter, makes it possible for the user to give the plotting function their own data, preferably retrieved with this function with <code>return_plot_data = TRUE</code> . If provided, the plotting function will disregard any other parameters and only draw the plot using the data provided.
<code>this_score</code>	Required parameter, should be one of the numeric columns in the scores data frame from the list returned with <code>lundtax_predict_sub()</code> .
<code>this_subtype</code>	Optional parameter, lets the user subset the return to a specific subtype. If nopt provided, all subtypes within the selected class will be returned.
<code>subtype_class</code>	Required, one of the following; <code>5_class</code> or <code>7_class</code> . Needed for coloring the points based on subtype classification.
<code>plot_title</code>	Optional parameter, the name of the returned plot. If NULL (default), no title will be printed to the generated figure.
<code>return_plot_data</code>	Boolean parameter, set to TRUE and return the formatted data used for plotting. Default is FALSE

Details

This function takes predictions returned with `lundtax_predict_sub()`, together with a score variable, subtype class and returns a point plot (grob) with ranked scores along the x axis and scores along the y axis. The returned plot will also color fill the points based on subtype classification.

Value

Nothing.

Examples

```

sjodahl_2017_predicted = lundtax_predict_sub(this_data = sjodahl_2017,
                                             impute = TRUE)

plot_ranked_score(these_predictions = sjodahl_2017_predicted,
                  this_score = "proliferation_score",
                  subtype_class = "7_class",
                  plot_title = "Sjodahl 2017")

```

signatures	<i>Signatures</i>
------------	-------------------

Description

Bundles a set of signatures in different molecular processes and the genes associated. This dataset is needed for calculating the signature scores.

Usage

```
data(signatures)
```

Format

A list of 6.

Details

A list of 6 with different signatures and the associated genes.

- signatures_plot. Genes in hgnc and ensembl format associated with specific signatures.
- proliferation. Genes in hgnc and ensembl format associated with proliferation.
- progression. Genes in hgnc and ensembl format associated with progression.
- prostate. Genes in hgnc and ensembl format associated with prostate cancer.
- immune. Genes in hgnc and ensembl format associated with immuno process.
- stable_genes. Genes in hgnc and ensembl format associated with stable genes.

sjodahl_2017	<i>Sjodahl 2017.</i>
--------------	----------------------

Description

Gene expression data derived from the Sjödahl et. al. (2017) cohort. A matrix of RMA normalized and ComBat adjusted gene expression values for 15697 genes (hgnc format) for 267 samples

Usage

```
data(sjodahl_2017)
```

Format

A data frame with 15697 rows (gene expressions) and 267 columns (samples).

Details

A data frame with gene information in different formats.

sjodahl_2017_meta	<i>Sjodahl 2017 Meta.</i>
-------------------	---------------------------

Description

metadata associated with the bundled expresison data.

Usage

```
data(sjodahl_2017_meta)
```

Format

A data frame with 15697 rows (gene expressions) and 267 columns (samples).

Details

A data frame with meta data for the bundled samples.

- sample_id. Unique sample ID.
- cluster_order. Cluster order.
- age. Age of patient, years.
- gender. Patient gender.
- region_cx. Geographical region of CX.
- turb_grade. Pathological grading of TURB-specimen. 0 = WHO1999 Grade 2, 1 = WHO1999 Grade 3.
- turb_stage. Pathological staging of TURB-specimen. 0 = Non muscle-invasive, 1 = Muscle invasive.
- turb_skiv. Presence of keratinization or squamous differentiation in TURB-specimen. 0 = No presence, 1 = presence.
- turb_cis. Presence of cis in the TURB-specimen. 0 = No presence, 1 = presence.
- turb_lvi. Presence of lymphovascular invasion in the TURB-specimen. 0 = No presence, 1 = presence.
- turb_necros. Presence of necrosis in TURB-specimen. 0 = no, 1 = present, 2 = extensive.
- turb_apoptos. Presence of apoptosis in the TURB-specimen. 0 = No presence, 1 = presence.
- variant_histology. Variant histology present 1 = yes, 0 = no.
- cT. Clinical tumor state
- cN. Clinical node state
- hydroneph. Presence of hydronephrose. 0 = No presence, 1 = presence.
- pTCx. Pathological tumor state at cystectomy.
- pN. Patological node state at cystectomy.
- adj_chemo. Adjuvant chemotherapy. 0 = No presence, 1 = presence.
- surv_os_event. Survival data, 1 = event, 0 = non event.
- surv_os_time. Survival data, in months.
- surv_css_event. Cancer specific survival, 1 = event, 0 = non event.
- surv_css_time. Cancer specific survival, in months.
- surv_pfs_event. Progression-free survival, 1 = event, 0 = non event.
- surv_pfs_time. Progression-free survival, in months.

Index

* datasets

- classifier_grade3, [2](#)
- classifier_hg, [3](#)
- classifier_lundtax_5c, [4](#)
- classifier_lundtax_7c, [4](#)
- gene_list, [5](#)
- lund_colors, [13](#)
- signatures, [19](#)
- sjodahl_2017, [19](#)
- sjodahl_2017_meta, [20](#)

- classifier_grade3, [2](#)
- classifier_hg, [3](#)
- classifier_lundtax_5c, [4](#)
- classifier_lundtax_7c, [4](#)

- gene_list, [5](#)

- int_calc_immune_proportions, [6](#)
- int_calc_score, [6](#)
- int_calc_score(), [6](#)
- int_check_ties, [7](#)
- int_merge_suburo_matrix, [8](#)
- int_predict_grade, [8](#)
- int_ratio_score, [9](#)

- lund_colors, [13](#)

- LundTax2023Classifier
(LundTax2023Classifier-package),
[2](#)

- LundTax2023Classifier-package, [2](#)
- lundtax_calc_sigscore, [10](#)
- lundtax_calc_sigscore(), [7–10](#), [13](#)
- lundtax_predict_sub, [11](#)
- lundtax_predict_sub(), [14–16](#), [18](#)

- multiclassPairs::predict_RF(), [2–4](#), [9](#),
[11](#), [12](#)

- plot_hm_scores, [14](#)
- plot_hm_signatures, [15](#)
- plot_ranked_score, [17](#)

- signatures, [19](#)
- sjodahl_2017, [19](#)
- sjodahl_2017_meta, [20](#)