

Package: ODT (via r-universe)

October 21, 2024

Type Package

Title Optimal Decision Trees Algorithm

Version 1.0.1

Maintainer Katyna Sada Del Real <ksada@unav.es>

Description Implements a tree-based method specifically designed for personalized medicine applications. By using genomic and mutational data, 'ODT' efficiently identifies optimal drug recommendations tailored to individual patient profiles. The 'ODT' algorithm constructs decision trees that bifurcate at each node, selecting the most relevant markers (discrete or continuous) and corresponding treatments, thus ensuring that recommendations are both personalized and statistically robust. This iterative approach enhances therapeutic decision-making by refining treatment suggestions until a predefined group size is achieved. Moreover, the simplicity and interpretability of the resulting trees make the method accessible to healthcare professionals. Includes functions for training the decision tree, making predictions on new samples or patients, and visualizing the resulting tree. For detailed insights into the methodology, please refer to Gimeno et al. (2023) <[doi:10.1093/bib/bbad200](https://doi.org/10.1093/bib/bbad200)>.

Depends R (>= 4.0), matrixStats, partykit, data.tree, stats

Imports magick, DiagrammeRsvg, grDevices, DiagrammeR, rsvg

Suggests RUnit, Matrix, rmarkdown, robustbase, knitr

License Artistic-2.0

LazyData true

LazyDataCompression xz

RoxygenNote 7.3.2

Encoding UTF-8

VignetteBuilder knitr

NeedsCompilation no

Repository <https://katynasada.r-universe.dev>

RemoteUrl <https://github.com/katynasada/odt>

RemoteRef HEAD

RemoteSha 3ce5bf2b0d0327f305e26fedd0b84ac5a74ca1f1

Contents

drug_response_w12	2
drug_response_w34	3
expression_w12	3
expression_w34	4
mutations_w12	4
mutations_w34	5
niceTree	5
predictTree	6
trainTree	8
Index	10

drug_response_w12	<i>drug_response_w12 data</i>
-------------------	-------------------------------

Description

A matrix containing drug response values (IC50 values) obtained from patients in Waves 1 and 2 of the BeatAML2 cohort. This dataset serves as a toy example for demonstrating the functionality of the `trainTree`, `predictTree`, and `niceTree` functions.

Usage

```
data("drug_response_w12")
```

Format

The format is: num [1:247, 1:119] 2.710983 2.8755433 3.4390103 2.6527257...

Examples

```
data(drug_response_w12)
```

drug_response_w34 *drug_response_w34 data*

Description

A matrix containing drug response values (IC50 values) obtained from patients in Waves 3 and 4 of the BeatAML2 cohort. This dataset serves as a toy example for demonstrating the functionality of the `trainTree`, `predictTree`, and `niceTree` functions.

Usage

```
data("drug_response_w34")
```

Format

The format is: num [1:142, 1:119] 3.4156359 3.2345985 3.1836058 3.7874252...

Examples

```
data(drug_response_w34)
```

expression_w12 *expression_w12 Data Set*

Description

A dataframe containing gene expression values obtained from patients in Waves 1 and 2 of the BeatAML cohort. This dataset serves as a toy example for demonstrating the functionality of the `trainTree`, `predictTree`, and `niceTree` functions.

Usage

```
data("expression_w12")
```

Format

A dataframe consisting of 247 rows and 1000 columns, where each row represents a different patient and each column corresponds to the expression levels of a specific gene. The entries in the data frame are floating-point values, indicating the gene expression levels measured for each patient.

Examples

```
data(expression_w12)
```

expression_w34 *expression_w34 Data Set*

Description

A dataframe containing gene expression values obtained from patients in Waves 3 and 4 of the BeatAML cohort. This dataset serves as a toy example for demonstrating the functionality of the `trainTree`, `predictTree`, and `niceTree` functions.

Usage

```
data("expression_w34")
```

Format

A dataframe consisting of 142 rows and 1000 columns, where each row represents a different patient and each column corresponds to the expression levels of a specific gene. The entries in the data frame are floating-point values, indicating the gene expression levels measured for each patient.

Examples

```
data(expression_w34)
```

mutations_w12 *mutations_w12 Data Set*

Description

A binary matrix representing mutation status for patients from Waves 1 and 2 of the BeatAML cohort, indicating whether specific mutations are present (1) or absent (0) in each patient. This dataset serves as a toy example for demonstrating the functionality of the `trainTree`, `predictTree`, and `niceTree` functions.

Usage

```
data("mutations_w12")
```

Format

A binary matrix consisting of 247 rows and 70 columns, where each row represents a different patient and each column corresponds to a specific mutation.

The format is as follows: `num [1:247, 1:70] 0 0 0 0 1 ...`

Examples

```
data(mutations_w12)
```

`mutations_w34`*mutations_w34 Data Set*

Description

A binary matrix representing mutation status for patients from Waves 3 and 4 of the BeatAML cohort, indicating whether specific mutations are present (1) or absent (0) in each patient. This dataset serves as a toy example for demonstrating the functionality of the `trainTree`, `predictTree`, and `niceTree` functions.

Usage

```
data("mutations_w34")
```

Format

A binary matrix consisting of 142 rows and 70 columns, where each row represents a different patient and each column corresponds to a specific mutation.

The format is as follows: num [1:247, 1:70] 0 0 0 0 1 ...

Examples

```
data(mutations_w34)
```

`niceTree`*niceTree function*

Description

A graphical display of the tree. It can also be saved as an image in the selected directory.

Usage

```
niceTree(  
  tree,  
  folder = NULL,  
  colors = c("", "#367592", "#39A7AE", "#96D6B6", "#FDE5B0", "#F3908B", "#E36192",  
    "#8E4884", "#A83333"),  
  fontname = "Roboto",  
  fontstyle = "plain",  
  shape = "diamond",  
  output_format = "png"  
)
```

Arguments

tree	A party of the trained tree with the treatments assigned to each node.
folder	Directory to save the image (default is the current working directory).
colors	A vector of colors for the boxes. Can include hex color codes (e.g., "#FFFFFF").
fontname	The name of the font to use for the text labels (default is "Roboto").
fontstyle	The style of the font (e.g., "plain", "italic", "bold").
shape	The format of the boxes for the different genes (e.g., "diamond", "box").
output_format	The image format for saving (e.g., "png", "jpg", "svg", "pdf").

Details

- The user has already defined a style for the plot; the parameters are set if not modified when calling niceTree.

Value

(Invisibly) returns a list. The representation of the tree in the command window and the plot of the tree.

Examples

```
# Basic example of how to perform niceTree:
data("mutations_w12")
data("drug_response_w12")
ODTmut <- trainTree(PatientData = mutations_w12,
                    PatientSensitivity = drug_response_w12, minbucket = 10)
niceTree(ODTmut)

# Example for plotting the tree trained for gene expressions:
data("expression_w34")
data("drug_response_w34")
ODTExp <- trainTree(PatientData = expression_w34,
                    PatientSensitivity = drug_response_w34, minbucket = 20)
niceTree(ODTExp)
```

predictTree

Predict Treatment Outcomes with a Trained Decision Tree

Description

This function utilizes a trained decision tree model (ODT) to predict treatment outcomes for test data based on patient sensitivity data and features, such as mutations or gene expression profiles.

`trainTree`*trainTree Function*

Description

This function trains a decision tree model based on patient data, which can either be gene expression levels or a binary matrix indicating mutations.

Usage

```
trainTree(PatientData, PatientSensitivity, minbucket = 20)
```

Arguments

- PatientData** A matrix representing patient features, where rows correspond to patients/samples and columns correspond to genes/features. This matrix can contain:
- Binary mutation data (e.g., presence/absence of mutations).
 - Continuous data from gene expression profiles (e.g., expression levels).
- PatientSensitivity** A matrix representing drug response values, where rows correspond to patients in the same order as in 'PatientData', and columns correspond to drugs. Higher values indicate greater drug resistance and, consequently, lower sensitivity to treatment. This matrix can represent various measures of drug response, such as IC50 values or area under the drug response curve (AUC). Depending on the interpretation of these values, users may need to adjust the sign of this data.
- minbucket** An integer specifying the minimum number of patients required in a node to allow for a split.

Value

An object of class 'party' representing the trained decision tree, with the assigned treatments for each node.

Examples

```
# Basic example of using the trainTree function with mutational data
data("drug_response_w12")
data("mutations_w12")
ODTmut <- trainTree(PatientData = mutations_w12,
                   PatientSensitivity = drug_response_w12,
                   minbucket = 10)

plot(ODTmut)

# Example using gene expression data instead
data("drug_response_w34")
data("expression_w34")
ODTExp <- trainTree(PatientData = expression_w34,
```


trainTree

9

```
plot(ODTExp, PatientSensitivity = drug_response_w34,  
minbucket = 20)
```

Index

* datasets

drug_response_w12, 2

drug_response_w34, 3

expression_w12, 3

expression_w34, 4

mutations_w12, 4

mutations_w34, 5

drug_response_w12, 2

drug_response_w34, 3

expression_w12, 3

expression_w34, 4

mutations_w12, 4

mutations_w34, 5

niceTree, 5

predictTree, 6

trainTree, 8