Package 'rSCA'

March 10, 2020

Type Package

Title An R Package for Stepwise Cluster Analysis

Version 3.1

Date 2020-03-03

Author Xiuquan (Xander) Wang

Maintainer Xiuquan (Xander) Wang <xiuquan.wang@gmail.com>

Description A statistical tool for multivariate modeling and clustering using stepwise cluster analysis. The modeling output of rSCA is constructed as a cluster tree to represent the complicated relationships between multiple dependent and independent variables. A free tool (named rSCA Tree Generator) for visualizing the cluster tree from rSCA is also released and it can be downloaded at <https://rscatree.weebly.com/>.

License GPL (>= 2)

Depends R (>= 2.10.0)

LazyLoad no

NeedsCompilation no

Repository CRAN

Date/Publication 2020-03-10 15:30:02 UTC

R topics documented:

`rSCA.correlation` *Correlation Analysis for Modeling Data Sets*

Description

This function aims to analyze the correlations between dependent variables and independent ones. It prints out a table consisting of the correlation coefficient for each pair of dependent and independent variables.

Usage

```
rSCA.correlation(xfile, yfile, x.row.names = FALSE, x.col.names = FALSE,
        y.row.names = FALSE, y.col.names = FALSE, x.missing.flag = "NA",
        y.missing.flag = "NA", x.type = ".txt", y.type = ".txt")
```

Arguments

`xfile`	a string to specify the full filename of the independent (x) data file, only supports files in *.txt or *.csv.
`yfile`	a string to specify the full filename of the dependent (y) data file, only supports files in *.txt or *.csv.
`x.row.names`	a logical value to specify if the independent (x) data file contains row names or not. Default value is FALSE.
`x.col.names`	a logical value to specify if the independent (x) data file contains column names or not. Default value is FALSE.
`y.row.names`	a logical value to specify if the dependent (y) data file contains row names or not. Default value is FALSE.
`y.col.names`	a logical value to specify if the dependent (y) data file contains column names or not. Default value is FALSE.
`x.missing.flag`	a string to specify the missing flag used in the independent (x) data file. Default value is "NA".
`y.missing.flag`	a string to specify the missing flag used in the dependent (y) data file. Default value is "NA".
`x.type`	a string to specify the type of independent (x) data file. Default value is ".txt".
`y.type`	a string to specify the type of dependent (y) data file. Default value is ".txt".

Author(s)

Xiuquan Wang <xiuquan.wang@gmail.com>

Examples

```
## Load rSCA package
library(rSCA)

## X data file
xdata <- c("A B C D\r", "0.095 0.044 39.9 27\r",
           "0.810 0.058 9.1 8\r", "0.101 0.077 11.4 14\r",
           "0.006 0.141 20.5 29\r", "0.070 0.281 27.3 26\r",
           "0.481 0.514 30.2 48\r", "0.120 0.286 36.4 39\r",
           "0.480 0.199 40.9 27\r", "0.112 0.101 29.9 18\r",
           "0.026 0.203 48.1 28\r", "0.128 1.235 48.2 61\r",
           "2.681 0.439 51.1 98\r", "1.601 0.333 56.1 99\r",
           "1.398 0.455 19.3 103\r", "1.256 0.314 14.9 17\r",
           "2.618 0.609 9.1 19\r", "1.217 0.880 17.2 73\r",
           "1.411 2.115 19.6 203\r", "0.245 6.839 49.2 296\r",
           "0.724 3.060 17.1 192\r", "0.019 2.252 29.1 123\r",
           "1.321 5.730 41.1 288\r", "0.903 3.078 39.0 97\r",
           "0.714 1.013 16.7 5\r", "0.581 1.398 11.7 57\r",
           "0.080 1.734 10.2 52\r", "0.120 1.848 6.6 132\r",
           "0.089 1.357 10.3 148\r", "0.112 0.585 19.3 79\r",
           "0.192 0.675 6.9 39\r", "0.301 1.937 11.9 6\r")
xdatafile <- tempfile()
writeLines(xdata, xdatafile)

## Y data file
ydata <- c("Y1 Y2 Y3\r", "0.020 0.034 10.01\r",
           "0.011 0.011 6.92\r", "0.016 0.018 9.53\r",
           "0.022 0.018 5.04\r", "0.031 0.029 8.90\r",
           "0.057 0.036 9.98\r", "0.040 0.048 12.96\r",
           "0.061 0.050 9.84\r", "0.023 0.031 8.84\r",
           "0.025 0.020 4.66\r", "0.041 0.042 9.02\r",
           "0.070 0.029 11.37\r", "0.077 0.022 11.88\r",
           "0.105 0.038 11.06\r", "0.038 0.027 11.64\r",
           "0.058 0.019 8.25\r", "0.051 0.050 10.01\r",
           "0.073 0.038 9.20\r", "0.123 0.080 9.91\r",
           "0.089 0.046 9.37\r", "0.073 0.039 7.99\r",
           "0.139 0.069 13.28\r", "0.095 0.048 9.80\r",
           "0.034 0.040 8.50\r", "0.055 0.034 9.21\r",
           "0.020 0.050 8.67\r", "0.070 0.036 8.03\r",
           "0.058 0.039 8.01\r", "0.057 0.031 6.30\r",
           "0.050 0.014 7.92\r", "0.039 0.040 8.08\r")
ydatafile <- tempfile()
writeLines(ydata, ydatafile)

## Analyze correlations between y and x
rSCA.correlation(xfile = xdatafile, x.col.names = TRUE,
    yfile = ydatafile, y.col.names = TRUE)

## Remove temporary data files
unlink(xdatafile)
unlink(ydatafile)
```

`rSCA.inference` *Inference for Stepwise Cluster Analysis*

Description

This function is used for statistical inference or prediction based on an existing stepwise cluster analysis (SCA) model. The results are saved into a text file (file name: rsl_modelname.txt) under the model's output folder. In some cases, the training process of SCA with large samples will be very slow, but users may want to use the trained model to do multiple predictions. If this is the case, users can construct a SCA model based on previously-generated tree and map files (that means you only need to run rSCA.modeling function once). Such model can be assigned to the last parameter (i.e., "model") of this function to perform multiple predictions. Please refer to the following example codes for more details.

Usage

```
rSCA.inference(xfile, x.row.names = FALSE, x.col.names = FALSE,
        x.missing.flag = "NA", x.type = ".txt", model)
```

Arguments

`xfile`	a string to specify the full filename of the independent (x) data file, only supports files in *.txt or *.csv.
`x.row.names`	a logical value to specify if the independent (x) data file contains row names or not. Default value is FALSE.
`x.col.names`	a logical value to specify if the independent (x) data file contains column names or not. Default value is FALSE.
`x.missing.flag`	a string to specify the missing flag used in the independent (x) data file. Default value is "NA".
`x.type`	a string to specify the type of independent (x) data file. Default value is ".txt".
`model`	a SCA model object to be used for statistical inference or prediction.

Author(s)

Xiuquan Wang <xiuquan.wang@gmail.com>

References

Wang, X., G. Huang, S. Zhao, and G. Guo (2015), An open-source software package for multivariate modeling and clustering: applications to air quality management. Environmental Science and Pollution Research, 22(18), 14220-14233.

Wang, X., G. Huang, Q. Lin, X. Nie, G. Cheng, Y. Fan, Z. Li, Y. Yao, and M. Suo (2013), A stepwise cluster analysis approach for downscaled climate projection - A Canadian case study. Environmental Modelling & Software, 49, 141-151.

Examples

```
## Load rSCA package
library(rSCA)

## X data file
xdata <- c("A B C D\r", "0.095 0.044 39.9 27\r",
           "0.810 0.058 9.1 8\r", "0.101 0.077 11.4 14\r",
           "0.006 0.141 20.5 29\r", "0.070 0.281 27.3 26\r",
           "0.481 0.514 30.2 48\r", "0.120 0.286 36.4 39\r",
           "0.480 0.199 40.9 27\r", "0.112 0.101 29.9 18\r",
           "0.026 0.203 48.1 28\r", "0.128 1.235 48.2 61\r",
           "2.681 0.439 51.1 98\r", "1.601 0.333 56.1 99\r",
           "1.398 0.455 19.3 103\r", "1.256 0.314 14.9 17\r",
           "2.618 0.609 9.1 19\r", "1.217 0.880 17.2 73\r",
           "1.411 2.115 19.6 203\r", "0.245 6.839 49.2 296\r",
           "0.724 3.060 17.1 192\r", "0.019 2.252 29.1 123\r",
           "1.321 5.730 41.1 288\r", "0.903 3.078 39.0 97\r",
           "0.714 1.013 16.7 5\r", "0.581 1.398 11.7 57\r",
           "0.080 1.734 10.2 52\r", "0.120 1.848 6.6 132\r",
           "0.089 1.357 10.3 148\r", "0.112 0.585 19.3 79\r",
           "0.192 0.675 6.9 39\r", "0.301 1.937 11.9 6\r")
xdatafile <- tempfile()
writeLines(xdata, xdatafile)

## Y data file
ydata <- c("Y1 Y2 Y3\r", "0.020 0.034 10.01\r",
           "0.011 0.011 6.92\r", "0.016 0.018 9.53\r",
           "0.022 0.018 5.04\r", "0.031 0.029 8.90\r",
           "0.057 0.036 9.98\r", "0.040 0.048 12.96\r",
           "0.061 0.050 9.84\r", "0.023 0.031 8.84\r",
           "0.025 0.020 4.66\r", "0.041 0.042 9.02\r",
           "0.070 0.029 11.37\r", "0.077 0.022 11.88\r",
           "0.105 0.038 11.06\r", "0.038 0.027 11.64\r",
           "0.058 0.019 8.25\r", "0.051 0.050 10.01\r",
           "0.073 0.038 9.20\r", "0.123 0.080 9.91\r",
           "0.089 0.046 9.37\r", "0.073 0.039 7.99\r",
           "0.139 0.069 13.28\r", "0.095 0.048 9.80\r",
           "0.034 0.040 8.50\r", "0.055 0.034 9.21\r",
           "0.020 0.050 8.67\r", "0.070 0.036 8.03\r",
           "0.058 0.039 8.01\r", "0.057 0.031 6.30\r",
           "0.050 0.014 7.92\r", "0.039 0.040 8.08\r")
ydatafile <- tempfile()
writeLines(ydata, ydatafile)

## New X data
xnewdata <- c("A B C D\r", "0.085 0.054 35.9 29\r",
           "0.820 0.068 9.2 7\r", "0.121 0.067 12.4 13\r",
           "0.016 0.151 21.5 24\r", "0.075 0.283 25.3 16\r",
           "0.581 0.524 31.2 38\r", "0.130 0.486 33.4 36\r")
xnewdatafile <- tempfile()
writeLines(xnewdata, xnewdatafile)
```

```
## Modeling the relationship between Y and X with SCA
myModel = rSCA.modeling(xfile = xdatafile, yfile = ydatafile,
              x.col.names = TRUE, y.col.names = TRUE)

## Predict Y with the SCA model
rSCA.inference(xfile = xnewdatafile, x.col.names = TRUE, model = myModel)

## Perform multiple predictions based on a previously-trained model
## Step 1. Construct a model based on previous tree and map files:
## >> preModel = list(treefile = path_to_tree_file, mapfile = path_to_map_file, type = "mean")
## Note that the value for the parameter of "type" should be the same as what you
## used in the previous traning
## Step 3. Perform multiple predictions with different xfiles
## >> rSCA.inference(xfile = xnewdatafile_1, x.col.names = TRUE, model = preModel)
## >> rSCA.inference(xfile = xnewdatafile_2, x.col.names = TRUE, model = preModel)
```

`rSCA.missing` *Checking Missing Values for Modeling Data Sets*

Description

This function helps check if there are missing values in the modeling data sets. It prints out general statistics for missing values and their specific locations in the data matrix. It returns TRUE/FALSE to indicate whether the data file passes the missing check.

Usage

```
rSCA.missing(file, row.names = FALSE, col.names = FALSE,
        missing.flag = "NA", type = ".txt")
```

Arguments

`file`	a string to specify the full filename of the data file, only supports files in *.txt or *.csv.
`row.names`	a logical value to specify if the data file contains row names or not. Default value is FALSE.
`col.names`	a logical value to specify if the data file contains column names or not. Default value is FALSE.
`missing.flag`	a string to specify the missing flag used in the data file. Default value is "NA".
`type`	a string to specify the type of data file. Default value is ".txt".

Value

Return a logical value to indicate whether the data file contains missing value(s).

Author(s)

Xiuquan Wang <xiuquan.wang@gmail.com>

Examples

```
## Load rSCA package
library(rSCA)

## Case 1: without column name and row name, using NA as missing flag
data <- c("5 7 9 10\r",
          "12 3 4 5\r",
          "1 NA NA 3\r",
          "2 NA 13 23\r",
          "0 12 1 8\r",
          "NA 9 0 1\r")
datafile <- tempfile()
writeLines(data, datafile)
bPass = rSCA.missing(file = datafile)

## Remove temporary data files
unlink(datafile)


## Case 2: with column name and row name, using -99 as missing flag
data <- c("Year V1 V2 V3 V4\r",
          "2010 5 7 9 10\r",
          "2012 12 3 4 5\r",
          "2013 1 -99 -99 3\r",
          "2014 2 -99 13 23\r",
          "2015 0 12 1 8\r",
          "2016 -99 9 0 1\r")
datafile <- tempfile()
writeLines(data, datafile)
bPass = rSCA.missing(file = datafile, col.names = TRUE,
            row.names = TRUE, missing.flag = "-99")

## Remove temporary data files
unlink(datafile)

## Case 3: with column name and row name, using NA as missing flag,
##         stored in comma-separated value (CSV) format.
data <- c("Year,V1,V2,V3,V4\r",
          "2010,5,7,9,10\r",
          "2012,12,3,4,5\r",
          "2013,1,NA,NA,3\r",
          "2014,2,NA,13,23\r",
          "2015,0,12,1,8\r",
          "2016,NA,9,0,1\r")
datafile <- tempfile()
writeLines(data, datafile)
bPass = rSCA.missing(file = datafile, col.names = TRUE,
            row.names = TRUE, missing.flag = "NA", type = ".csv")
```

```
## Remove temporary data files
unlink(datafile)
```

---

rSCA.modeling *Multivariate Modeling with Stepwise Cluster Analysis*

Description

This function models the relationship between multivariate dependent variable and independent ones. It represents the relationship as a clustered tree. The information for the clustered tree is saved into two text files: tree file (file name: tree_modelname.txt) and map file (file name: map_modelname.txt). A tree file stores in the structure of the clustered tree, and a map file contains the detailed information of leaf nodes. There two files are generated under the output folder. If the debug mode is enabled, a log file (file name: log_modelname.txt) will also be generated under the output folder. A free tool (named rSCA Tree Generator) for visualizing the cluster tree from rSCA is also released along with this new version and it can be downloaded at: https://rscatree.weebly.com/.

Usage

```
rSCA.modeling(modelname = "rSCA", alpha = 0.05, xfile, yfile,
        x.row.names = FALSE, x.col.names = FALSE,
y.row.names = FALSE, y.col.names = FALSE,
        x.missing.flag = "NA", y.missing.flag = "NA",
x.type = ".txt", y.type = ".txt",
mapvalue = "mean", GSS = FALSE, debug = FALSE,
outfolder = tempdir())
```

Arguments

modelname	specify a unique model name, default model name is rSCA.
alpha	significance level for clustering, usually in 0.001 - 0.1, default value is 0.05.
xfile	a string to specify the full filename of the independent (x) data file, only supports files in *.txt or *.csv.
yfile	a string to specify the full filename of the dependent (y) data file, only supports files in *.txt or *.csv.
x.row.names	a logical value to specify if the independent (x) data file contains row names or not. Default value is FALSE.
x.col.names	a logical value to specify if the independent (x) data file contains column names or not. Default value is FALSE.
y.row.names	a logical value to specify if the dependent (y) data file contains row names or not. Default value is FALSE.
y.col.names	a logical value to specify if the dependent (y) data file contains column names or not. Default value is FALSE.

`x.missing.flag` a string to specify the missing flag used in the independent (x) data file. Default value is "NA".

`y.missing.flag` a string to specify the missing flag used in the dependent (y) data file. Default value is "NA".

`x.type` a string to specify the type of independent (x) data file. Default value is ".txt".

`y.type` a string to specify the type of dependent (y) data file. Default value is ".txt".

`mapvalue` a predefined string to specify how the information of leaf nodes will be stored in the map file. A full list of options for mapvalue is: mean, max, min, median, interval, radius, variation, random. Default value is "mean". Use "interval" to get an interval bounded by the minimum and maximum values, (i.e., [min, max]). The option of "radius" will output the mean and the radius [i.e., (max - min) / 2] at the same time. Use "variation" to output the mean and the standard deviation. The option of "random" will output the information as an interval like the "interval" option, but it will use a random value within the interval as the predicted value.

`GSS` a logical value to specify if the Golden Section Search method will be used for seeking the best cutting point. Default value is FALSE.

`debug` a logical value to specify if the debug mode is enabled or not. Default value is FALSE. A log file will be created under the output folder if debug model is enabled.

`outfolder` full path to the output folder, the default output folder is tempdir().

Value

`treefile` a string shows the name of the tree file, tree_modelname.txt. The tree file will be located under the output folder.

`mapfile` a string shows the name of the map file, map_modelname.txt. The tree file will be located under the output folder.

`logfile` a string shows the name of the log file, log_modelname.txt. If the debug mode is disabled, the value of logfile will be NA. The tree file will be located under the output folder.

`type` a string indicates how the information of leaf nodes will be stored. Generally, the "type" keeps the same value as specified by the input parameter – "mapvalue".

`totalNodes` a number indicates how many nodes are included in the clustered tree.

`leafNodes` a number indicates how many leaf nodes are generated the clustered tree.

`cuttingActions` a number indicates how many cutting actions are executed during the modeling process.

`mergingActions` a number indicates how many merging actions are executed during the modeling process.

Author(s)

Xiuquan Wang <xiuquan.wang@gmail.com>

References

Wang, X., G. Huang, S. Zhao, and G. Guo (2015), An open-source software package for multivariate modeling and clustering: applications to air quality management. Environmental Science and Pollution Research, 22(18), 14220-14233.

Wang, X., G. Huang, Q. Lin, X. Nie, G. Cheng, Y. Fan, Z. Li, Y. Yao, and M. Suo (2013), A stepwise cluster analysis approach for downscaled climate projection - A Canadian case study. Environmental Modelling & Software, 49, 141-151.

Huang, G. (1992). A stepwise cluster analysis method for predicting air quality in an urban environment. Atmospheric Environment (Part B. Urban Atmosphere), 26(3): 349-357.

Liu, Y. Y. and Y. L. Wang (1979). Application of stepwise cluster analysis in medical research. Scientia Sinica, 22(9): 1082-1094.

Examples

```
## Load rSCA package
library(rSCA)

## X data file
xdata <- c("A B C D\r", "0.095 0.044 39.9 27\r",
           "0.810 0.058 9.1 8\r", "0.101 0.077 11.4 14\r",
           "0.006 0.141 20.5 29\r", "0.070 0.281 27.3 26\r",
           "0.481 0.514 30.2 48\r", "0.120 0.286 36.4 39\r",
           "0.480 0.199 40.9 27\r", "0.112 0.101 29.9 18\r",
           "0.026 0.203 48.1 28\r", "0.128 1.235 48.2 61\r",
           "2.681 0.439 51.1 98\r", "1.601 0.333 56.1 99\r",
           "1.398 0.455 19.3 103\r", "1.256 0.314 14.9 17\r",
           "2.618 0.609 9.1 19\r", "1.217 0.880 17.2 73\r",
           "1.411 2.115 19.6 203\r", "0.245 6.839 49.2 296\r",
           "0.724 3.060 17.1 192\r", "0.019 2.252 29.1 123\r",
           "1.321 5.730 41.1 288\r", "0.903 3.078 39.0 97\r",
           "0.714 1.013 16.7 5\r", "0.581 1.398 11.7 57\r",
           "0.080 1.734 10.2 52\r", "0.120 1.848 6.6 132\r",
           "0.089 1.357 10.3 148\r", "0.112 0.585 19.3 79\r",
           "0.192 0.675 6.9 39\r", "0.301 1.937 11.9 6\r")
xdatafile <- tempfile()
writeLines(xdata, xdatafile)

## Y data file
ydata <- c("Y1 Y2 Y3\r", "0.020 0.034 10.01\r",
           "0.011 0.011 6.92\r", "0.016 0.018 9.53\r",
           "0.022 0.018 5.04\r", "0.031 0.029 8.90\r",
           "0.057 0.036 9.98\r", "0.040 0.048 12.96\r",
           "0.061 0.050 9.84\r", "0.023 0.031 8.84\r",
           "0.025 0.020 4.66\r", "0.041 0.042 9.02\r",
           "0.070 0.029 11.37\r", "0.077 0.022 11.88\r",
           "0.105 0.038 11.06\r", "0.038 0.027 11.64\r",
           "0.058 0.019 8.25\r", "0.051 0.050 10.01\r",
           "0.073 0.038 9.20\r", "0.123 0.080 9.91\r",
           "0.089 0.046 9.37\r", "0.073 0.039 7.99\r",
           "0.139 0.069 13.28\r", "0.095 0.048 9.80\r",
```

```
            "0.034 0.040 8.50\r", "0.055 0.034 9.21\r",
            "0.020 0.050 8.67\r", "0.070 0.036 8.03\r",
            "0.058 0.039 8.01\r", "0.057 0.031 6.30\r",
            "0.050 0.014 7.92\r", "0.039 0.040 8.08\r")
ydatafile <- tempfile()
writeLines(ydata, ydatafile)

## Modeling with SCA: default parameters
myModel = rSCA.modeling(xfile = xdatafile, yfile = ydatafile,
              x.col.names = TRUE, y.col.names = TRUE)

## Modeling with SCA: alpha = 0.1, with debug mode enabled
myModel = rSCA.modeling(alpha = 0.1, xfile = xdatafile, yfile = ydatafile,
              x.col.names = TRUE, y.col.names = TRUE, debug = TRUE)

## Modeling with SCA: alpha = 0.05, use interval for leaf nodes
myModel = rSCA.modeling(xfile = xdatafile, yfile = ydatafile,
              x.col.names = TRUE, y.col.names = TRUE, mapvalue = "interval")
```

Index