

Package ‘didec’

February 2, 2026

Type Package

Title Directed Dependence Coefficient

Version 1.1.0

Maintainer Yuping Wang <yuping.wang@plus.ac.at>

Description Directed Dependence Coefficient (didec) is a measure of functional dependence. Multivariate Feature Ordering by Conditional Independence (MFOCI) is a variable selection algorithm based on didec. Hierarchical Variable Clustering (VarClustPartition) is a variable clustering method based on didec. For more information, see the paper by Ansari and Fuchs (2025, <[doi:10.48550/arXiv.2212.01621](https://doi.org/10.48550/arXiv.2212.01621)>), and the paper by Fuchs and Wang (2024, <[doi:10.1016/j.ijar.2024.109185](https://doi.org/10.1016/j.ijar.2024.109185)>).

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports copBasic (>= 2.2.3), cowplot (>= 1.1.2), dendextend (>= 1.17.1), FOCI (>= 0.1.3), ggplot2 (>= 3.4.4), graphics (>= 4.3.0), grDevices (>= 0.5-1), gtools (>= 3.9.5), pcaPP (>= 2.0-5), phylogram (>= 2.1.0), RANN (>= 2.6.1), rlang (>= 1.1.4), stats (>= 4.3.0)

Depends R (>= 3.5)

NeedsCompilation no

Author Yuping Wang [aut, cre],
Sebastian Fuchs [aut],
Jonathan Ansari [aut]

Repository CRAN

Date/Publication 2026-02-02 08:30:02 UTC

Contents

bioclimatic	2
didec	3
mfoci	4
VarClustPartition	6
Index	10

bioclimatic	<i>Bioclimatic variables</i>
-------------	------------------------------

Description

A data set of bioclimatic variables for $n = 1,862$ locations homogeneously distributed over the global landmass from CHELSA ("Climatologies at high resolution for the earth's land surface areas").

Usage

`bioclimatic`

Format

An object of class `data.frame` with 1862 rows and 19 columns.

References

D.N. Karger, O. Conrad, J. Böhner, T. Kawohl, H. Kreft, R.W. Soria-Auza, N.E. Zimmermann, H.P. Linder, M. Kessler, Climatologies at high resolution for the Earth's land surface areas, *Sci. Data* 4(1), 2017.

Examples

```
data(bioclimatic)
head(bioclimatic)
```

didec	<i>Computes the directed dependence coefficient.</i>
-------	------------------------------------------------------

Description

The directed dependence coefficient (didec) estimates the degree of functional dependence of a random vector Y on a random vector X , based on an i.i.d. sample of (X, Y) .

Usage

```
didec(
  X,
  Y,
  trans = FALSE,
  trans.method = c("standardization"),
  estim.method = c("copula"),
  perm = FALSE,
  perm.method = c("decreasing")
)
```

Arguments

<code>X</code>	A numeric matrix or data.frame/data.table. Contains the predictor vector X .
<code>Y</code>	A numeric matrix or data.frame/data.table. Contains the response vector Y .
<code>trans</code>	A logical. If TRUE the inputs of X are standardized (transformed) before didec is computed.
<code>trans.method</code>	An optional character string specifying the data standardization method. This must be one of the strings "standardization" (default), "rank" or "rescaling". "standardization" centers and scales each predictor to zero mean and unit variance (z-score). "rank" uses the rank of values instead of the values themselves. "rescaling" rescales each predictor to $[0, 1]$ (min-max normalization).
<code>estim.method</code>	An optional character string specifying a method for estimating the directed dependence coefficient. This must be one of the strings "codec" or "copula" (default).
<code>perm</code>	A logical. If TRUE a version of didec is computed that takes into account the permutations (specified by perm.method) of the response variables.
<code>perm.method</code>	An optional character string specifying a method for permuting the response variables. This must be one of the strings "sample", "increasing", "decreasing" (default) or "full". The version "full" is invariant under permutations of the response variables.

Details

The directed dependence coefficient (didec) is an extension of Azadkia & Chatterjee's measure of functional dependence (Azadkia & Chatterjee, 2021) to a vector of response variables introduced

in (Ansari & Fuchs, 2025). `estim.method` specifies two methods for estimating the directed dependence coefficient. `"codec"` uses the function `codec` which estimates Azadkia & Chatterjee's measure of functional dependence and is provided in the R package FOCI. `"copula"` estimates the directed dependence coefficient based on a dimension reduction principle; see (Fuchs 2024). The value returned by `didec` may be positive or negative. In the asymptotic limit, however, it is guaranteed to lie between 0 and 1.

By definition, `didec` is invariant under permutations of the variables within the predictor vector X . Invariance under permutations within the q -dimensional response vector Y is achieved by computing the arithmetic mean over all possible permutations. In addition to the option `"full"` of running all $q!$ permutations of $(1, \dots, q)$, less computationally intensive options are also available: a random selection of q permutations `"sample"`, cyclic permutations such as $(1, 2, \dots, q)$, $(2, \dots, q, 1)$ either `"increasing"` or `"decreasing"`. Note that when the number of variables q is large, choosing `"full"` may result in long computation times.

Value

The degree of functional dependence of the random vector Y on the random vector X .

Author(s)

Yuping Wang, Sebastian Fuchs, Jonathan Ansari

References

- J. Ansari, S. Fuchs, A direct extension of Azadkia & Chatterjee's rank correlation to multi-response vectors, Available at <https://arxiv.org/abs/2212.01621>, 2025.
- M. Azadkia, S. Chatterjee, A simple measure of conditional dependence, *Ann. Stat.* 49 (6), 2021.
- S. Fuchs, Quantifying directed dependence via dimension reduction, *J. Multivariate Anal.* 201, Article ID 105266, 2024.

mfoci

Multivariate feature ordering by conditional independence.

Description

A variable selection algorithm based on the directed dependence coefficient (`didec`).

Usage

```
mfoci(
  X,
  Y,
  trans = FALSE,
  trans.method = c("standardization"),
  estim.method = c("copula"),
  perm = FALSE,
  perm.method = c("decreasing"),
```

```

pre.selected = NULL,
select.method = c("forward"),
autostop = TRUE,
max.num = NULL
)

```

Arguments

<code>X</code>	A numeric matrix or data.frame/data.table. Contains the predictor vector X.
<code>Y</code>	A numeric matrix or data.frame/data.table. Contains the response vector Y.
<code>trans</code>	A logical. If TRUE the inputs of X are standardized (transformed) before the variable selection.
<code>trans.method</code>	An optional character string specifying a method for data standardization. This must be one of the strings "standardization" (default), "rank" or "rescaling".
<code>estim.method</code>	An optional character string specifying a method for estimating the directed dependence coefficient <code>didec</code> . This must be one of the strings "codec" or "copula" (default).
<code>perm</code>	A logical. If TRUE a version of <code>didec</code> that takes into account the permutations of the response variables is used in the variable selection algorithm.
<code>perm.method</code>	An optional character string specifying a method for permuting the response variables. This must be one of the strings "sample", "increasing", "decreasing" (default) or "full".
<code>pre.selected</code>	An integer vector for indexing pre-selected components from predictor X.
<code>select.method</code>	An optional character string specifying a feature selection method. This must be one of the strings "forward" (default) or "subset".
<code>autostop</code>	A logical. If True (default) the forward feature selection algorithm stops at the first non-increasing value of <code>didec</code> .
<code>max.num</code>	An integer for limiting the maximal number of selected variables if <code>select.method == "subset"</code> .

Details

`mfoci` involves a forward feature selection algorithm for multiple-outcome data that employs the directed dependence coefficient (`didec`) at each step.

If `autostop == TRUE` the algorithm stops at the first non-increasing value of `didec`, thereby selecting a subset of variables. Otherwise, all predictor variables are ranked according to their predictive strength measured by `didec`.

In addition to the forward feature selection algorithm, this function also provides a best subset selection, which can be accomplished by `select.method == "subset"`. This method selects features by calculating the directed dependence coefficient of all possible feature combinations. Note that the features selected by this method are not ordered.

Value

A list containing:

features A vector listing all features in X;

pre.selected.features A vector listing the pre.selected features in X if `pre.selected != NULL`;

selected.features A data.frame listing the selected and ranked variables and the corresponding values of the directed dependence coefficient if `select.method == "forward"`; A vector listing the selected features if `select.method == "subset"`;

valueT The values of the directed dependence coefficient if `select.method == "subset"`.

Author(s)

Sebastian Fuchs, Jonathan Ansari, Yuping Wang

References

J. Ansari, S. Fuchs, A direct extension of Azadkia & Chatterjee's rank correlation to multi-response vectors, Available at <https://arxiv.org/abs/2212.01621>, 2025.

Examples

```
library(didec)
df <- as.data.frame(bioclimatic)
X <- df[, c(9:12)]
Y <- df[, c(1,8)]
mfoci(X, Y, pre.selected = c(1, 3))
```

VarClustPartition

Hierarchical variable clustering and partition.

Description

VarClustPartition is a hierarchical variable clustering algorithm based on the directed dependence coefficient ([didec](#)) or a concordance measure (Kendall tau τ or Spearman's footrule) according to a pre-selected number of clusters or an optimality criterion (Adiam&Msplite or Silhouette coefficient).

Usage

```
VarClustPartition(
  X,
  trans = FALSE,
  trans.method = c("standardization"),
  dist.method = c("PD"),
  estim.method = c("copula"),
  linkage = FALSE,
  link.method = c("complete"),
```

```

part.method = c("optimal"),
part.criterion = c("Adiam&Msplitt"),
num.cluster = NULL,
plot = FALSE
)

```

Arguments

<code>X</code>	A numeric matrix or data.frame/data.table. Contains the variables to be clustered.
<code>trans</code>	A logical. If TRUE the inputs are standardized (transformed) before clustering.
<code>trans.method</code>	An optional character string specifying a method for data standardization. This must be one of the strings "standardization" (default), "rank" or "rescaling".
<code>dist.method</code>	An optional character string computing a distance function for clustering. This must be one of the strings "PD" (default), "MPD", "kendall" or "footrule".
<code>estim.method</code>	An optional character string specifying a method for estimating the directed dependence coefficient if <code>dist.method == "PD"</code> or <code>dist.method == "MPD"</code> . This must be one of the strings "codec" or "copula" (default).
<code>linkage</code>	A logical. If TRUE a linkage method is used.
<code>link.method</code>	An optional character string selecting a linkage method. This must be one of the strings "complete" (default), "average" or "single".
<code>part.method</code>	An optional character string selecting a partitioning method. This must be one of the strings "optimal" (default) or "selected".
<code>part.criterion</code>	An optional character string selecting a criterion for the optimal partition if <code>part.method = "optimal"</code> . This must be one of the strings "Adiam&Msplitt" (default) or "Silhouette".
<code>num.cluster</code>	An integer value for the pre-selected number of clusters if <code>part.method = "selected"</code> .
<code>plot</code>	A logical. If TRUE a dendrogram is plotted with colored branches according to the corresponding partitioning method.

Details

VarClustPartition performs a hierarchical variable clustering based on the directed dependence coefficient (didec) and provides a partition of the set of variables.

If `dist.method == "PD"` (perfect dependence) or `dist.method == "MPD"` (mutual perfect dependence) the clustering is performed using didec either as a directed ("PD") or as a symmetric ("MPD") dependence coefficient. If `dist.method == "kendall"` or `dist.method == "footrule"`, clustering is performed using either multivariate Kendall's tau ("kendall") or multivariate Spearman's footrule ("footrule"). "kendall" uses the function `cor.fk` which is provided in the R package `pcaPP` to calculate bivariate Kendall's tau.

Instead of using one of the above-mentioned four multivariate measures for the clustering, the option `linkage == TRUE` enables the use of bivariate linkage methods, including complete linkage (`link.method == "complete"`), average linkage (`link.method == "average"`) and single linkage (`link.method == "single"`). Note that the multivariate distance methods are computationally demanding because higher-dimensional dependencies are included in the calculation, in contrast to linkage methods which only incorporate pairwise dependencies.

A pre-selected number of clusters `num.cluster` can be realized with the option `part.method == "selected"`. Otherwise (`part.method == "optimal"`), the number of clusters is determined by maximizing the intra-cluster similarity (similarity within the same cluster) and minimizing the inter-cluster similarity (similarity among the clusters). Two optimality criteria (Fuchs & Wang 2024) are available:

"*Adiam*&*Mspl*it": *Adiam* measures the intra-cluster similarity and *Mspl*it measures the inter-cluster similarity.

"*Silhouette*": A mixed coefficient incorporating the intra-cluster similarity and the inter-cluster similarity. The optimal number of clusters corresponds to the maximum Silhouette coefficient.

Value

A list containing:

dendrogram A dendrogram without colored branches;

num.cluster An integer value determining the number of clusters after partitioning;

clusters A list containing the clusters after partitioning.

Author(s)

Yuping Wang, Sebastian Fuchs

References

S. Fuchs, Y. Wang, Hierarchical variable clustering based on the predictive strength between random vectors, *Int. J. Approx. Reason.* 170, Article ID 109185, 2024.

P. Hansen, B. Jaumard, Cluster analysis and mathematical programming, *Math. Program.* 79 (1) 191–215, 1997.

L. Kaufman, *Finding Groups in Data*, John Wiley & Sons, 1990.

Examples

```
library(didec)
n <- 50
X1 <- rnorm(n,0,1)
X2 <- X1
X3 <- rnorm(n,0,1)
X4 <- X3 + X2
X <- data.frame(X1=X1,X2=X2,X3=X3,X4=X4)
vcp <- VarClustPartition(X,
                        dist.method = c("PD"),
                        part.method = c("optimal"),
                        part.criterion = c("Silhouette"),
                        plot          = TRUE)

vcp$clusters

data("bioclimatic")
X <- bioclimatic[c(2:4,9)]
vcp1 <- VarClustPartition(X,
```



```
linkage      = TRUE,
link.method = c("complete"),
dist.method = "PD",
part.method = "optimal",
part.criterion = "Silhouette",
plot        = TRUE)

vcp1$clusters
vcp2 <- VarClustPartition(X,
linkage      = TRUE,
link.method = c("complete"),
dist.method = "footrule",
part.method = "optimal",
part.criterion = "Adiam&Msplitt",
plot        = TRUE)

vcp2$clusters
```

Index

* **datasets**

bioclimatic, [2](#)

bioclimatic, [2](#)

didec, [3](#), [4](#), [6](#)

mfoci, [4](#)

VarClustPartition, [6](#)