

# Package ‘clusterCrit’

June 4, 2015

**Type** Package

**Title** Clustering Indices

**Version** 1.2.5

**Date** 2015-06-04

**Author** Bernard Desgraupes (University of Paris Ouest - Lab Modal'X)

**Maintainer** Bernard Desgraupes <bernard.desgraupes@u-paris10.fr>

**Description** Compute clustering validation indices.

**License** GPL (>= 2)

**URL** <http://www.r-project.org>

**Collate** main.R zzz.R

**Encoding** latin1

**Suggests** RUnit, rbenchmark

## R topics documented:

bestCriterion . . . . .	1
clusterCrit . . . . .	2
concordance . . . . .	3
extCriteria . . . . .	5
getCriteriaNames . . . . .	6
intCriteria . . . . .	7

<b>Index</b>	<b>10</b>
--------------	-----------

---

bestCriterion	<i>Best clustering index</i>
---------------	------------------------------

---

## Description

bestCriterion returns the best index value according to a specified criterion.

## Usage

```
bestCriterion(x, crit)
```

## Arguments

<code>x</code>	[matrix] : a numeric vector of quality index values.
<code>crit</code>	[character] : a string specifying the name of the criterion which was used to compute the quality indices.

## Details

Given a vector of several clustering quality index values computed with a given criterion, the function `bestCriterion` returns the index of the *"best"* one in the sense of the specified criterion. Typically, a set of data has been clusterized several times (using different algorithms or specifying a different number of clusters) and a clustering index has been calculated each time : the `bestCriterion` function tells which value is considered the best according to the given clustering index. For instance, if one uses the Calinski\_Harabasz index, the best value is the largest one.

A list of all the supported criteria can be obtained with the `getCriteriaNames` function. The criterion name (`crit` argument) is case insensitive and can be abbreviated.

## Value

The index in vector `x` of the best value according to the criterion specified by the `crit` argument.

## Author

Bernard Desgraupes  
 <bernard.desgraupes@u-paris10.fr>  
 University of Paris Ouest - Nanterre  
 Lab Modal'X (EA 3454)

## See Also

`getCriteriaNames`, `intCriteria`.

## Examples

```
# Create some spheric data around three distinct centers
x <- rbind(matrix(rnorm(100, mean = 0, sd = 0.5), ncol = 2),
            matrix(rnorm(100, mean = 2, sd = 0.5), ncol = 2),
            matrix(rnorm(100, mean = 4, sd = 0.5), ncol = 2))
vals <- vector()
for (k in 2:6) {
  # Perform the kmeans algorithm
  cl <- kmeans(x, k)
  # Compute the Calinski_Harabasz index
  vals <- c(vals, as.numeric(intCriteria(x, cl$cluster, "Calinski_Harabasz")))
}
idx <- bestCriterion(vals, "Calinski_Harabasz")
cat("Best index value is", vals[idx], "\n")
```

## Description

Package: clusterCrit  
Type: Package  
Version: 1.2.5  
Date: 2015-06-04  
License: GPL (>= 2)

## Details

clusterCrit computes various clustering validation or quality criteria and partition comparison indices. Type

```
library(help="clusterCrit")
```

for more info about the available functions.

## Author

Bernard Desgraupes  
<bernard.desgraupes@u-paris10.fr>  
University of Paris Ouest - Nanterre  
Lab Modal'X (EA 3454)

## References

For more information about the algebraic background of clustering indices and their definition, see the vignette accompanying this package. To display the vignette, type the following instruction in the R console :

```
> vignette("clusterCrit")
```

## See Also

[extCriteria](#), [getCriteriaNames](#), [intCriteria](#), [bestCriterion](#), [concordance](#).

---

concordance

*Compute Concordance Matrix*

---

## Description

concordance calculates the concordance matrix between two partitions of the same data.

## Usage

```
concordance(part1, part2)
```

**Arguments**

part1            [vector] : the first partition vector.  
 part2            [vector] : the second partition vector.

**Details**

Given two partitions, the function `concordance` calculates the number of pairs classified as belonging or not belonging to the same cluster with respect to partitions `part1` or `part2`.

**Value**

A 2x2 matrix of the form :

		P1		P2	
-----					
P1		Nyy		Nyn	
P2		Nny		Nnn	
-----					

where

- Nyy is the number of points belonging to the same cluster both in `part1` and `part2`
- Nyn is the number of points belonging to the same cluster in `part1` but not in `part2`
- Nny is the number of points belonging to the same cluster in `part2` but not in `part1`
- Nnn is the number of points *not* belonging to the same cluster both in `part1` and `part2`

**Author**

Bernard Desgraupes  
 <bernard.desgraupes@u-paris10.fr>  
 University of Paris Ouest - Nanterre  
 Lab Modal'X (EA 3454)

**See Also**

[extCriteria](#), [intCriteria](#).

**Examples**

```
# Generate two artificial partitions
part1<-sample(1:3,150,replace=TRUE)
part2<-sample(1:5,150,replace=TRUE)

# Compute the table of concordances and discordances
concordance(part1,part2)
```

---

extCriteria	<i>Compute external clustering criteria</i>
-------------	---

---

## Description

extCriteria calculates various external clustering comparison indices.

## Usage

```
extCriteria(part1, part2, crit)
```

## Arguments

part1	[vector] : the first partition vector.
part2	[vector] : the second partition vector.
crit	[vector] : a vector containing the names of the indices to compute.

## Details

The function extCriteria calculates external clustering indices in order to compare two partitions. The list of all the supported criteria can be obtained with the [getCriteriaNames](#) function.

The currently available indices are :

- "Czekanowski\_Dice"
- "Folkes\_Mallows"
- "Hubert"
- "Jaccard"
- "Kulczynski"
- "McNemar"
- "Phi"
- "Precision"
- "Rand"
- "Recall"
- "Rogers\_Tanimoto"
- "Russel\_Rao"
- "Sokal\_Sneath1"
- "Sokal\_Sneath2"

All the names are case insensitive and can be abbreviated. The keyword "all" can also be used as a shortcut to calculate all the external indices.

The partition vectors should not have empty subsets. No attempt is made to verify this.

## Value

A list containing the computed criteria, in the same order as in the crit argument.

**Author**

Bernard Desgraupes  
 <bernard.desgraupes@u-paris10.fr>  
 University of Paris Ouest - Nanterre  
 Lab Modal'X (EA 3454)

**References**

See the bibliography at the end of the vignette.

**See Also**

[getCriteriaNames](#), [intCriteria](#), [bestCriterion](#), [concordance](#).

**Examples**

```
# Generate two artificial partitions
part1<-sample(1:3,150,replace=TRUE)
part2<-sample(1:5,150,replace=TRUE)

# Compute all the external indices
extCriteria(part1,part2,"all")
# Compute some of them
extCriteria(part1,part2,c("Rand","Folkes"))
# The names are case insensitive and can be abbreviated
extCriteria(part1,part2,c("ra","fo"))
```

---

getCriteriaNames	<i>Get clustering criteria names</i>
------------------	--------------------------------------

---

**Description**

getCriteriaNames returns the available clustering criteria names.

**Usage**

```
getCriteriaNames(isInternal)
```

**Arguments**

isInternal      [logical] : get internal indices if TRUE, external indices otherwise.

**Details**

getCriteriaNames returns a list of the available internal or external clustering indices depending on the isInternal logical argument.

The internal indices can be used in the crit argument of the [intCriteria](#) function and the external indices similarly in the [extCriteria](#) function.

**Value**

A character vector containing the supported criteria names.

**Author**

Bernard Desgraupes  
<bernard.desgraupes@u-paris10.fr>  
University of Paris Ouest - Nanterre  
Lab Modal'X (EA 3454)

**References**

See the bibliography at the end of the vignette.

**See Also**

[intCriteria](#), [extCriteria](#), [bestCriterion](#).

**Examples**

```
getCriteriaNames(TRUE)  
getCriteriaNames(FALSE)
```

---

intCriteria	<i>Compute internal clustering criteria</i>
-------------	---

---

**Description**

intCriteria calculates various internal clustering validation or quality criteria.

**Usage**

```
intCriteria(traj, part, crit)
```

**Arguments**

traj	[matrix] : the matrix of observations (trajectories).
part	[vector] : the partition vector.
crit	[vector] : a vector containing the names of the indices to compute.

**Details**

The function intCriteria calculates internal clustering indices. The list of all the supported criteria can be obtained with the [getCriteriaNames](#) function.

The currently available indices are :

- "Ball\_Hall"
- "Banfeld\_Raftery"
- "C\_index"
- "Calinski\_Harabasz"
- "Davies\_Bouldin"
- "Det\_Ratio"
- "Dunn"

- "Gamma"
- "G\_plus"
- "GDI11"
- "GDI12"
- "GDI13"
- "GDI21"
- "GDI22"
- "GDI23"
- "GDI31"
- "GDI32"
- "GDI33"
- "GDI41"
- "GDI42"
- "GDI43"
- "GDI51"
- "GDI52"
- "GDI53"
- "Ksq\_DetW"
- "Log\_Det\_Ratio"
- "Log\_SS\_Ratio"
- "McClain\_Rao"
- "PBM"
- "Point\_Biserial"
- "Ray\_Turi"
- "Ratkowsky\_Lance"
- "Scott\_Symons"
- "SD\_Scat"
- "SD\_Dis"
- "S\_Dbw"
- "Silhouette"
- "Tau"
- "Trace\_W"
- "Trace\_WiB"
- "Wemmert\_Gancarski"
- "Xie\_Beni"

All the names are case insensitive and can be abbreviated. The keyword "all" can also be used as a shortcut to calculate all the internal indices.

The GDI (*Generalized Dunn Indices*) are designated by the following convention:  $GDI_{mn}$ , where the integers  $m$  ( $1 \leq m \leq 5$ ) and  $n$  ( $1 \leq n \leq 3$ ) correspond to the between-group and within-group distances respectively. See the vignette for a comprehensive definition of the various distances. GDI alone is synonym of GDI11 and is the genuine Dunn's index.



**Value**

A list containing the computed criteria, in the same order as in the `crit` argument.

**Author**

Bernard Desgraupes  
<bernard.desgraupes@u-paris10.fr>  
University of Paris Ouest - Nanterre  
Lab Modal'X (EA 3454)

**References**

See the bibliography at the end of the vignette.

**See Also**

[getCriteriaNames](#), [extCriteria](#), [bestCriterion](#).

**Examples**

```
# Create some data
x <- rbind(matrix(rnorm(100, mean = 0, sd = 0.5), ncol = 2),
           matrix(rnorm(100, mean = 1, sd = 0.5), ncol = 2),
           matrix(rnorm(100, mean = 2, sd = 0.5), ncol = 2))
# Perform the kmeans algorithm
cl <- kmeans(x, 3)
# Compute all the internal indices
intCriteria(x, cl$cluster, "all")
# Compute some of them
intCriteria(x, cl$cluster, c("C_index", "Calinski_Harabasz", "Dunn"))
# The names are case insensitive and can be abbreviated
intCriteria(x, cl$cluster, c("det", "cal", "dav"))
```

# Index

- \*Topic **clusters**
  - clusterCrit, [2](#)
- \*Topic **indices**
  - clusterCrit, [2](#)
- \*Topic **package**
  - clusterCrit, [2](#)
- \*Topic **partition**
  - clusterCrit, [2](#)

bestCriterion, [1](#), [3](#), [6](#), [7](#), [9](#)

clusterCrit, [2](#)

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

concordance, [3](#), [3](#), [6](#)

extCriteria, [3](#), [4](#), [5](#), [6](#), [7](#), [9](#)

getCriteriaNames, [2](#), [3](#), [5](#), [6](#), [6](#), [7](#), [9](#)

intCriteria, [2-4](#), [6](#), [7](#), [7](#)