

# Package ‘DBCVindex’

December 13, 2024

**Encoding** UTF-8

**Title** Calculates the Density-Based Clustering Validation Index (DBCV) Index

**Version** 1.1

**Description** A metric called 'Density-Based Clustering Validation index' (DBCV) index to evaluate clustering results, following the <<https://github.com/FelSiq/DBCV>> 'Python' implementation by Felipe Alves Siqueira. Original 'DBCV' index article: Moulavi, D., Jaskowiak, P. A., Campello, R. J., Zimek, A., & Sander, J. (2014, April). ``Density-based clustering validation", Proceedings of SDM 2014 -- the 2014 SIAM International Conference on Data Mining (pp. 839-847), <[doi:10.1137/1.9781611973440.96](https://doi.org/10.1137/1.9781611973440.96)>.

**Depends** R (>= 4.0.0)

**License** GPL-3

**URL** <https://github.com/davidechicco/DBCVindex>

**Imports** pacman, Matrix, stats

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Davide Chicco [aut, cre] (<<https://orcid.org/0000-0001-9655-7142>>)

**Maintainer** Davide Chicco <[davidechicco@davidechicco.it](mailto:davidechicco@davidechicco.it)>

**Repository** CRAN

**Date/Publication** 2024-12-13 09:30:02 UTC

## Contents

compute_pair_to_pair_dists . . . . .	2
dbcv . . . . .	2
remove_duplicates . . . . .	3
<b>Index</b>	<b>5</b>

---

`compute_pair_to_pair_dists`*Function to compute pairwise distances and ensure matrix format*

---

**Description**

Function to compute pairwise distances and ensure matrix format

**Usage**

```
compute_pair_to_pair_dists(data, metric = "euclidean")
```

**Arguments**

<code>data</code>	input clustering results
<code>metric</code>	metric of the distance, Euclidean by default

**Value**

a pairwise distances' matrix

**Examples**

```
n = 300; noise = 0.05; seed = 1782;
theta <- seq(0, pi, length.out = n / 2)
x1 <- cos(theta) + rnorm(n / 2, sd = noise)
y1 <- sin(theta) + rnorm(n / 2, sd = noise)
x2 <- cos(theta + pi) + rnorm(n / 2, sd = noise)
y2 <- sin(theta + pi) + rnorm(n / 2, sd = noise)
X <- rbind(cbind(x1, y1), cbind(x2, y2))

dist_matrix <- compute_pair_to_pair_dists(X)
```

---

`dbcv`*Function that calculates the Density-Based Clustering Validation index (DBCv) of clustering results*

---

**Description**

Function that calculates the Density-Based Clustering Validation index (DBCv) of clustering results

**Usage**

```
dbcv(data, labels, metric = "euclidean", noise_id = -1)
```

**Arguments**

data	input clustering results
labels	labels of the clustering
metric	metric of the distance, Euclidean by default
noise_id	the code of the noise cluster points, -1 by default

**Value**

a real value containing the Saturn coefficient

**Examples**

```
n = 300; noise = 0.05; seed = 1782;
theta <- seq(0, pi, length.out = n / 2)
x1 <- cos(theta) + rnorm(n / 2, sd = noise)
y1 <- sin(theta) + rnorm(n / 2, sd = noise)
x2 <- cos(theta + pi) + rnorm(n / 2, sd = noise)
y2 <- sin(theta + pi) + rnorm(n / 2, sd = noise)
X <- rbind(cbind(x1, y1), cbind(x2, y2))
y <- c(rep(0, n / 2), rep(1, n / 2))

cat("dbcv(X, y) = ", dbcv(X, y), "\n", sep="")
```

---

remove\_duplicates      *Function to remove duplicate samples from the input data*

---

**Description**

Function to remove duplicate samples from the input data

**Usage**

```
remove_duplicates(data, labels)
```

**Arguments**

data	input clustering results
labels	labels of the clustering

**Value**

a list of data and labels without duplicates

**Examples**

```
n = 300; noise = 0.05; seed = 1782;
theta <- seq(0, pi, length.out = n / 2)
x1 <- cos(theta) + rnorm(n / 2, sd = noise)
y1 <- sin(theta) + rnorm(n / 2, sd = noise)
x2 <- cos(theta + pi) + rnorm(n / 2, sd = noise)
y2 <- sin(theta + pi) + rnorm(n / 2, sd = noise)
X <- rbind(cbind(x1, y1), cbind(x2, y2))
y <- c(rep(0, n / 2), rep(1, n / 2))

cat("remove_duplicates(X, y) = ")
print(remove_duplicates(X, y))
```

# Index

`compute_pair_to_pair_dists`, [2](#)

`dbcv`, [2](#)

`remove_duplicates`, [3](#)