

# Package ‘dccc’

September 27, 2023

**Type** Package

**Title** Fast Computation of Distance Correlations

**Version** 0.1.0

**Date** 2023-09-27

**Description** Fast computation of the distance covariance 'dcov' and distance correlation 'dcor'. The computation cost is only  $O(n \log(n))$  for the distance correlation (see Chaudhuri, Hu (2019) <[arXiv:1810.11332](https://arxiv.org/abs/1810.11332)> <[doi:10.1016/j.csda.2019.01.016](https://doi.org/10.1016/j.csda.2019.01.016)>). The functions are written entirely in C++ to speed up the computation.

**License** GPL (>= 3)

**URL** <https://dccc.berrisch.biz/>, <https://github.com/BerriJ/dccc>

**BugReports** <https://github.com/BerriJ/dccc/issues>

**Encoding** UTF-8

**Imports** Rcpp (>= 1.0.8)

**LinkingTo** Rcpp, RcppArmadillo

**RoxygenNote** 7.2.3

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** yes

**Author** Jonathan Berrisch [aut, cre] (<<https://orcid.org/0000-0002-4944-9074>>)

**Maintainer** Jonathan Berrisch <[Jonathan@Berrisch.biz](mailto:Jonathan@Berrisch.biz)>

**Repository** CRAN

**Date/Publication** 2023-09-27 08:30:07 UTC

## R topics documented:

dcor	2
dcov	2

<b>Index</b>	<b>4</b>
--------------	----------

---

dcor *Distance Correlation*

---

**Description**

Distance Correlation

**Usage**

dcor(x,y)

**Arguments**

x                numeric vector  
y                numeric vector

**Value**

Returns a numeric value: the distance correlation between x and y.

**Examples**

```
## Not run:  
  
set.seed(1)  
x <- rnorm(1000)  
y <- -x ^ 2  
  
dcor(x, y) # dcor shows dependence between x and y  
cor(x, y) # cor does not detect any dependence due to nonlinearity  
  
## End(Not run)
```

---

dcov *Distance Covariance*

---

**Description**

Distance Covariance

**Usage**

dcov(x,y)

**Arguments**

x	numeric vector
y	numeric vector

**Details**

Implements the algorithm described in Chaudhuri, Hu (2019) [doi:10.1016/j.csda.2019.01.016](https://doi.org/10.1016/j.csda.2019.01.016) which only has  $O(n \log(n))$  complexity.

**Value**

Returns a numeric value: the distance covariance between x and y.

**Examples**

```
## Not run:  
  
set.seed(1)  
x <- rnorm(1000)  
y <- -x ^ 2  
  
dcov(x, y)  
dvov(x, x)  
dvov(y, y)  
  
## End(Not run)
```

# Index

dcor, 2  
dcov, 2