

# Package ‘distops’

January 23, 2024

**Title** Usual Operations for Distance Matrices in R

**Version** 0.1.0

**Description** It provides the subset operator for dist objects and a function to compute medoid(s) that are fully parallelized leveraging the 'RcppParallel' package. It also provides functions for package developers to easily implement their own parallelized dist() function using a custom 'C++'-based distance function.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.0

**URL** <https://github.com/lmj1-alea/distops>,  
<https://lmjl-alea.github.io/distops/>

**BugReports** <https://github.com/lmj1-alea/distops/issues>

**SystemRequirements** GNU make

**Imports** cli, desc, fs, glue, Rcpp, RcppParallel, usethis

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**LinkingTo** Rcpp, RcppParallel

**NeedsCompilation** yes

**Author** Aymeric Stamm [aut, cre] (<<https://orcid.org/0000-0002-8725-3654>>)

**Maintainer** Aymeric Stamm <aymeric.stamm@cnrs.fr>

**Repository** CRAN

**Date/Publication** 2024-01-23 13:53:26 UTC

## R topics documented:

distops-package . . . . .	2
find_medoids . . . . .	2
use_distance . . . . .	3
use_distops . . . . .	3
[.dist . . . . .	4

---

distops-package	<i>distops: Usual Operations for Distance Matrices in R</i>
-----------------	---

---

### Description

It provides the subset operator for dist objects and a function to compute medoid(s) that are fully parallelized leveraging the 'RcppParallel' package. It also provides functions for package developers to easily implement their own parallelized dist() function using a custom 'C++'-based distance function.

### Author(s)

**Maintainer:** Aymeric Stamm <aymeric.stamm@cnrs.fr> ([ORCID](#))

### See Also

Useful links:

- <https://github.com/lmjl-alea/distops>
- <https://lmjl-alea.github.io/distops/>
- Report bugs at <https://github.com/lmjl-alea/distops/issues>

---

find_medoids	<i>Finds the medoids from a distance matrix</i>
--------------	---

---

### Description

This function finds the medoids from a distance matrix. The medoid is the object that minimizes the sum of distances to all other objects. This function takes advantage of the **RcppParallel** package to compute the medoids in parallel.

### Usage

```
find_medoids(D, memberships = NULL)
```

### Arguments

D	An object of class <code>stats::dist</code> .
memberships	A factor specifying the cluster memberships of the objects.

### Value

A named integer vector specifying the indices of the medoids.

**Examples**

```
D <- stats::dist(iris[, 1:4])
find_medoids(D)
memberships <- as.factor(rep(1:3, each = 50L))
find_medoids(D, memberships)
```

---

use_distance	<i>Adds a distance function to the package</i>
--------------	--

---

**Description**

This function adds a distance function to the package. It first creates the `R/{distance_name}Distance.R` file with the R wrapper function for the distance function. It then creates the `src/{distance_name}Distance.cpp` file with the C++ implementation of the distance function. It finally opens the latter file in the default editor. The user will be able to implement the desired distance function in a way compatible with the **RcppParallel** workflow.

**Usage**

```
use_distance(distance_name)
```

**Arguments**

`distance_name` A character string specifying the name of the distance that the user aims at implementing.

**Value**

Nothing.

**Examples**

```
use_distance("euclidean")
```

---

use_distops	<i>Setups package to use the <b>distops</b> package</i>
-------------	---

---

**Description**

This function setups the package to use the **distops** package. It first creates the DESCRIPTION file adding the **Rcpp** and **RcppParallel** packages to both the Imports: and LinkingTo: fields and the **distops** package to the LinkingTo: field. It also adds the SystemRequirements: GNU make field. It then creates the NAMESPACE file adding the importFrom() directives for the **Rcpp** and **RcppParallel** packages and the useDynLib() directive for packages with compiled code. It finally creates the src/Makevars and src/Makevars.win files with the appropriate compilation flags.

**Usage**

```
use_distops()
```

**Value**

Nothing.

**Examples**

```
use_distops()
```

---

```
[.dist
```

```
Distance Matrix Subset Operator
```

---

**Description**

Subset operator for the distance matrix stored as an object of class `stats::dist`.

**Usage**

```
## S3 method for class 'dist'
x[i, j, drop = TRUE, ...]
```

**Arguments**

<code>x</code>	An object of class <code>stats::dist</code> .
<code>i</code>	An integer vector of row indices. Values must be either all positive in which case they indicate the rows to select, or all negative in which case they indicate the rows to omit.
<code>j</code>	An integer vector of column indices. Values must be either all positive in which case they indicate the columns to select, or all negative in which case they indicate the columns to omit.
<code>drop</code>	A logical value indicating whether the result should be coerced to a vector or matrix if possible.
<code>...</code>	Additional arguments passed to <code>[.dist</code> .

**Value**

A numeric matrix storing the pairwise distances between the requested observations.

**Examples**

```
D <- stats::dist(iris[, 1:4])
D[2:3, 7:12]
```

# Index

`[.dist`, 4

`distops (distops-package)`, 2

`distops-package`, 2

`find_medoids`, 2

`stats::dist`, 2, 4

`use_distance`, 3

`use_distops`, 3