

# Package ‘pdist’

July 23, 2025

**Type** Package

**Title** Partitioned Distance Function

**Version** 1.2.1

**Date** 2013-01-31

**Author** Jeffrey Wong

**Maintainer** Jeffrey Wong <jeff.ct.wong@gmail.com>

**URL** <https://github.com/jeffwong/pdist>

**Description** Computes the euclidean distance between rows of a matrix X and rows of another matrix Y. Previously, this could be done by binding the two matrices together and calling 'dist', but this creates unnecessary computation by computing the distances between a row of X and another row of X, and likewise for Y. pdist strictly computes distances across the two matrices, not within the same matrix, making computations significantly faster for certain use cases.

**License** GPL

**LazyLoad** yes

**Depends** methods

**Collate** 'pdist.R' 'setup.R'

**Repository** CRAN

**Date/Publication** 2022-05-02 13:01:57 UTC

**NeedsCompilation** yes

## Contents

pdist-package . . . . .	2
pdist . . . . .	2
[ . . . . .	3

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

---

pdist-package

*Distances between Observations for a Partitioned Matrix*

---

### Description

Computes the euclidean distance between rows of a matrix X and rows of another matrix Y. Previously, this could be done by binding the two matrices together and calling 'dist', but this creates unnecessary computation by computing the distances between a row of X and another row of X, and likewise for Y. pdist strictly computes distances across the two matrices, not within the same matrix, making computations significantly faster for certain use cases.

### Details

Package: pdist  
 Type: Package  
 Version: 1.2  
 Date: 2013-01-31  
 License: GPL  
 LazyLoad: yes

### Author(s)

Jeffrey Wong <jeff.ct.wong@gmail.com>

---

pdist

*Partitioned Distances*

---

### Description

Compute distance matrix between two matrices of observations, or two subsets of one matrix

### Usage

```
pdist(X, Y = NULL, indices.A = NULL, indices.B = NULL)
```

### Arguments

X a matrix of n observations where columns represent features of the observations  
 Y optional. A second matrix of p observations like X. Y must have the same number of columns as X  
 indices.A optional. A vector of integers representing row indices from X. This should only be used when Y is not provided.  
 indices.B optional. A vector of integers representing row indices from X. This should only be used when Y is not provided.

### Details

`pdist` computes a  $n$  by  $p$  distance matrix using two separate matrices. `pdist` allows the user to factor out observations into separate matrices to improve computations. The function `dist` computes the distances between all possible pair wise elements, `pdist` only computes the distance between observations in  $X$  with observations in  $Y$ ; distances between observations in  $X$  and other observations in  $X$  are not computed, and likewise for  $Y$ .

If a second matrix  $Y$  is not provided, `indices.A` and `indices.B` can be provided together to specify subsets of  $X$  to be computed. A new matrix  $X$  is created by taking  $X[\text{indices.A},]$  and  $Y$  is created using  $X[\text{indices.B},]$ .

The return value of `pdist` is a distance vector, much like the default return value for `dist`. However, it can be accessed like a full distance matrix. If `mydist = pdist(X,Y)`, `mydist[i,j]` is the distance between  $X[i,]$  and  $Y[j,]$ . Similarly, `mydist[i,]` is a vector of distances between  $X[i,]$  and all observations in  $Y$ .

### Examples

```
x = matrix(rnorm(100),10,10)
x.pdist = pdist(x, indices.A = 1:3, indices.B = 8:10)
message("Find the distance between observation 1 and 10 of x")
x.pdist[1,3]
message("Converting a pdist object into a traditional distance matrix")
as.matrix(x.pdist)
```

---

[

*extract parts of pdist*

---

### Description

extract parts of `pdist`

# Index

## \* **package**

    pdist-package, 2

    [], 3

    [], pdist-method ([]), 3

pdist, 2

pdist-package, 2