Package 'BASIX'

February 19, 2015

Type Package

Title BASIX: An efficient C/C++ toolset for R.

Version 1.1

Date 2013-08-4

Author Bastian Pfeifer

Maintainer Bastian Pfeifer <Bastian.Pfeifer@uni-duesseldorf.de>

Depends R (>= 2.14.2), methods

Description BASIX provides some efficient C/C++ implementations to speed up calculations in R.

License GPL-2

LazyLoad yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2013-10-24 15:02:32

R topics documented:

BASIX.combnapply *Combine-Apply*

Description

This function applies a specific calculation, defined by the user, to all pairs of entries of a vector.

Usage

```
BASIX.combnapply(vec, mode='*')
```

Arguments

vec	vector
mode	*,-,/,+ or ==

Details

BASIX.combnapply returns values for each pair-combination.

Author(s)

Bastian Pfeifer

Examples

```
vec  <- c(1,3,5,7,9,10)
BASIX.combnapply(vec)
```

BASIX.equal *Test if two vectors are equal*

Description

This function checks if two vectors are equal, by comparing each cell and quits when the first mismatch occurs.

Usage

```
BASIX.equal(a,b)
```

Arguments

a	first vector
b	second vector

Details

Native R functions compare every entry of the entired vectors,

The R solution would be:

```
all(a==b)
```

The function can be applied to numeric as well as character vectors

Author(s)

Bastian Pfeifer

Examples

```
a <- 1:10
b <- a
b[2] <- 9
BASIX.equal(a,b)
```

BASIX.find.interval *Positions of elements in a defined region*

Description

This function returns the positions of elements which are in a defined region.

Usage

```
BASIX.find.interval(vec,from,to,start=1)
```

Arguments

vec	base vector
from	left side
to	right side
start	the start position

Details

The function returns the position of the elements in vector `vec`,
which are in a specific interval, defined by `from` and `to`,
starting from position `start`.

The native R solution would be:

```
 X  <- (vec >= from) & (vec <= to)
ids <- which(X)
```

or the function `findInterval()`, which is just a little bit slower than `BASIX.interval`

Author(s)

Bastian Pfeifer

Examples

```
vec  <- c(1,3,5,7,9,10)
from <- 5
to   <- 8
BASIX.find.interval(vec,from,to,1)
```

BASIX.match *Value Matching for sorted vectors.*

Description

The function returns a vector of the positions of (first) matches of its first argument in its second.

Usage

```
BASIX.match(elements,vec)
```

Arguments

`elements`	values to be matched
`vec`	the values to be matched against

Details

`BASIX.match` stops at the first match and continues with the next element of `elements` starting from the match position in `vec`.

The native R solution would be:

```
match(elements,vec)
```

Author(s)

Bastian Pfeifer

Examples

```
elements <- c(1,2,10)
vec      <- c(2,3,4,10,11)

BASIX.match(elements,vec)
```

BASIX.table *Count Tabulation of Matrices*

Description

`BASIX.table` computes the counts of unique rows of a matrix.

Usage

```
BASIX.table(matrix)
```

Arguments

`matrix`	a matrix

Details

This function can be applied to numeric as well as character matrices.

Author(s)

Bastian Pfeifer

Examples

```
mat <- matrix(0,3,3)
mat[1,1] <- 1

BASIX.table(mat)
```

BASIX.unique *Extract unique rows of a matrix*

Description

BASIX.unique returns a matrix with duplicate rows removed. The unique rows ids will be saved as rownames.

Usage

```
BASIX.unique(matrix)
```

Arguments

matrix	a matrix

Details

BASIX.unique is a version of unique specialized on matrices.

The native R solution would be:
unique(matrix)

This function can be applied to numeric as well as character vectors.

Author(s)

Bastian Pfeifer

Examples

```
mat <- matrix(0,3,3)
mat[1,1] <- 1

BASIX.unique(mat)
```

Index