

The WilcoxCV Package

February 16, 2008

Version 1.0-1

Date 2007-04-11

Title Wilcoxon-based variable selection in cross-validation

Author Anne-Laure Boulesteix <boulesteix@slcmsr.org>.

Maintainer Anne-Laure Boulesteix <boulesteix@slcmsr.org>

Depends R (>= 2.0.0)

Suggests

Description This package provides functions to perform fast variable selection based on the Wilcoxon rank sum test in the cross-validation or Monte-Carlo cross-validation settings, for use in microarray-based binary classification.

License GPL version 2 or newer

URL <http://cran.r-project.org/src/contrib/Descriptions/WilcoxCV.html>

R topics documented:

WilcoxCV-internal	2
generate.cv	2
generate.split	3
wilcox.selection.split	4
wilcox.split	6

Index	8
--------------	----------

WilcoxCV-internal *Internal WilcoxCV Functions*

Description

Internal WilcoxCV functions.

Note

These are not to be called by the user (or in some cases are just waiting for proper documentation to be written).

generate.cv *Generating groups for cross-validation*

Description

The function `generate.cv` generates randomly m groups for m -fold cross-validation.

Usage

```
generate.cv(n, m)
```

Arguments

n	The total number of observations in the data set.
m	The desired number of groups.

Details

Leave-one-out cross-validation is a special case of cross-validation, with $m=n$.

Value

A $m \times \text{ceiling}(n/m)$ matrix giving the indices of the observations included in each group. The i -th row gives the indices of observations included in the i -th group. If the m groups are not perfectly equally sized, the last column includes one or several zero(s).

Author(s)

Anne-Laure Boulesteix (<http://www.slcmr.net/boulesteix>)

References

A. L. Boulesteix (2007). WilcoxCV: an R package for fast variable selection in cross-validation.

See Also

[generate.split](#), [wilcox.split](#), [wilcox.selection.split](#)

Examples

```
# load WilcoxCV library
library(WilcoxCV)

# Generate 10 groups for a data set of size 95.
my.cv<-generate.cv(n=95,m=10)
```

generate.split *Generating random splittings into learning and test data sets*

Description

The function `generate.split` generates `niter` random splittings into learning and test data sets for use in Monte-Carlo cross-validation (MCCV).

Usage

```
generate.split(niter, n, ntest)
```

Arguments

<code>niter</code>	The number of iterations (number of splits into learning and split sets).
<code>n</code>	The total number of observations in the data set.
<code>ntest</code>	The number of observations in the test sets.

Details

This function is meant for use in Monte-Carlo cross-validation (MCCV).

Value

A `niter` x `ntest` matrix giving the indices of the observations included in the test sets. The *i*-th row gives the indices of the `ntest` observations included in the test set for the *i*-th MCCV iteration.

Author(s)

Anne-Laure Boulesteix (<http://www.slcmsr.net/boulesteix>)

References

A. L. Boulesteix (2007). WilcoxCV: an R package for fast variable selection in cross-validation.

See Also

[generate.cv](#), [wilcox.split](#), [wilcox.selection.split](#)

Examples

```
# load WilcoxCV library
library(WilcoxCV)

# Generate 50 splits with ratio 2:1 for a data set including 90 observations
my.split<-generate.split(niter=50,n=90,ntest=30)
```

```
wilcox.selection.split
```

Wilcoxon-based variable selection in cross-validation (CV) and Monte-Carlo cross-validation (MCCV)

Description

The function `wilcox.selection.split` performs variable ordering based on the Wilcoxon rank sum test for all `niter` CV or MCCV iterations.

Usage

```
wilcox.selection.split(x,y,split,algo="new",pvalue=FALSE)
```

Arguments

<code>x</code>	a matrix or a data frame of size $n \times p$ giving the expression levels of the p variables (genes) for the n observations (arrays). Variables correspond to columns, observations to rows.
<code>y</code>	a vector of length n giving the class membership for the n observations (arrays). <code>y</code> can be either a factor or a numeric and must be coded as 0,1.
<code>split</code>	A $niter \times ntest$ matrix giving the indices of the $ntest$ observations included in each of the $niter$ test sets, as generated by the functions generate.split or generate.cv . The i -th row of <code>split</code> gives the indices of the observations included in the test data set for the i -th random splitting iteration.
<code>algo</code>	either "new" or "naive". If <code>type="new"</code> , the new fast method described in Boulesteix (2007) is used. If <code>type="naive"</code> , results are obtained by running the function <code>wilcox.test</code> $niter$ times.
<code>pvalue</code>	Logical. Should p-values be returned?

Details

The Wilcoxon rank sum statistic is defined as the sum of the X -ranks of the observations with $y=0$. The Wilcoxon rank sum test is equivalent to the Mann-Whitney test. It is implemented in the function `wilcox.test`.

In the context of cross-validation (CV) or Monte-Carlo cross-validation (MCCV), `wilcox.selection.split` computes the Wilcoxon rank sum statistic for each iteration, for each variable. At each iteration, a subset of the n observations is excluded from the data set and considered as test data set. The indices of the observations considered as test set for each of the `niter` iterations are given in the `niter x ntest` matrix `split`.

Value

A list with the following components:

`ordering.split`

A `niter x p` matrix giving the indices of the genes ordered by `pvalue`. For example, the first column of `ordering.split` gives the index of the variable with lowest `pvalue` in each of the `niter` random splitting iterations, the second column of `ordering.split` gives the index of the variable with the second lowest `pvalue` in each of the `niter` random splitting iterations. For the i -th iteration, the indices of the 50 best variables are given in the 50 first columns of row i .

`pvalue.split` Returned only if `pvalue=TRUE`. A `niter x p` matrix of `pvalues`. The element in the i -th row and j -th column is the `pvalue` of variable j in the i -th iteration.

Author(s)

Anne-Laure Boulesteix (<http://www.slcmsr.net/boulesteix>)

References

A. L. Boulesteix (2007). WilcoxCV: an R package for fast variable selection in cross-validation.

See Also

[wilcox.test](#), [generate.split](#), [generate.cv](#), [wilcox.split](#)

Examples

```
# load WilcoxCV library
library(WilcoxCV)

# Generate data
x<-matrix(rnorm(1000),100,10)
y<-sample(c(0,1),100,replace=TRUE)

# Generate 50 MCCV splits with ratio 2:1 for a data set including 90 observations
my.split<-generate.split(niter=50,n=90,ntest=30)

# Compute the Wilcoxon rank sum statistic for the 50 iterations.
```

```
wilcox.selection.split(x=x,y=y,split=my.split,algo="new",pvalue=TRUE)
```

wilcox.split	<i>Wilcoxon rank sum statistic in cross-validation (CV) and Monte-Carlo cross-validation (MCCV)</i>
--------------	---

Description

The function `wilcox.split` computes the Wilcoxon rank sum statistic for all `niter` CV or MCCV iterations defined by the matrix `split`.

Usage

```
wilcox.split(x,y,split,algo="new")
```

Arguments

<code>x</code>	a numeric vector of length <code>n</code> giving the expression levels of a gene for the <code>n</code> arrays.
<code>y</code>	a vector of length <code>n</code> giving the class membership for the <code>n</code> arrays. <code>y</code> can be either a factor or a numeric and must be coded as 0,1.
<code>split</code>	A <code>niter x ntest</code> matrix giving the indices of the <code>ntest</code> observations included in each of the <code>niter</code> test sets, as generated by the functions <code>generate.split</code> or <code>generate.cv</code> . The <code>i</code> -th row of <code>split</code> gives the indices of the observations included in the test data set for the <code>i</code> -th iteration.
<code>algo</code>	either "new" or "naive". If <code>algo="new"</code> , the new fast method described in Boulesteix (2007) is used to compute the Wilcoxon rank statistic. If <code>algo="naive"</code> , the Wilcoxon rank sum statistics are obtained by running the function <code>wilcox.test</code> <code>niter</code> times.

Details

The Wilcoxon rank sum statistic is defined as the sum of the X-ranks of the observations with `y=0`. The Wilcoxon rank sum test is equivalent to the Mann-Whitney test. It is implemented in the function `wilcox.test`.

In the context of cross-validation (CV) or Monte-Carlo cross-validation (MCCV), `wilcox.selection.split` computes the Wilcoxon rank sum statistic for each iteration. At each iteration, a subset of the `n` observations is excluded from the data set and considered as test data set. The indices of the observations considered as test set for each of the `niter` iterations are given in the `niter x ntest` matrix `split`.

Value

A list with the following components:

<code>wilcox.split</code>	a numeric vector of length <code>niter</code> whose <code>i</code> -th component gives the Wilcoxon rank sum statistic obtained in the <code>i</code> -th iteration.
---------------------------	--

Author(s)

Anne-Laure Boulesteix (<http://www.slcmsr.net/boulesteix>)

References

A. L. Boulesteix (2007). WilcoxCV: an R package for fast variable selection in cross-validation.

See Also

[wilcox.test](#), [generate.split](#), [generate.cv](#), [wilcox.selection.split](#)

Examples

```
# load WilcoxCV library
library(WilcoxCV)

# Generate data
x<-rnorm(100)
y<-sample(c(0,1),100,replace=TRUE)

# Generate 50 MCCV splits with ratio 2:1 for a data set including 90 observations
my.split<-generate.split(niter=50,n=90,ntest=30)

# Compute the Wilcoxon rank sum statistic for the 50 iterations.
wilcox.split(x=x,y=y,split=my.split,algo="new")
```

Index

*Topic **hstest**

- generate.cv, 2
- generate.split, 3
- wilcox.selection.split, 4
- wilcox.split, 5

*Topic **internal**

- WilcoxCV-internal, 1

generate.cv, 2, 3–6

generate.split, 2, 3, 4–6

wilcox.selection.split, 2, 3, 4, 6

wilcox.split, 2, 3, 5, 5

wilcox.split.internal

(*WilcoxCV-internal*), 1

wilcox.test, 5, 6

wilcox.test2(*WilcoxCV-internal*),

1

WilcoxCV-internal, 1