

# Package ‘roccv’

May 10, 2019

**Type** Package

**Title** ROC for Cross Validation Results

**Version** 1.2

**Date** 2019-05-10

**Author** Ben Sherwood [aut, cre]

**Depends** R (>= 3.0.0), glmnet, parallel, pROC

**Maintainer** Ben Sherwood <ben.sherwood@ku.edu>

**Description** Cross validate large genetic data while specifying clinical variables that should always be in the model using the function `cv()`. An ROC plot from the cross validation data with AUC can be obtained using `rocplot()`, which also can be used to compare different models. Framework was built to handle genetic data, but works for any data.

**ByteCompile** TRUE

**License** MIT + file LICENSE

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-05-10 16:20:03 UTC

## R topics documented:

<code>cv</code> . . . . .	2
<code>fit_pred_fold</code> . . . . .	3
<code>randomly_assign</code> . . . . .	4
<code>roccv</code> . . . . .	4
<code>rocplot</code> . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

cv

*Cross validation results for a model***Description**

Cross validation results for a model

**Usage**

```
cv(clinical_x = NULL, genomic_x = NULL, y = NULL, data = NULL,
   clinical_formula = NULL, family = "binomial", folds = NULL, k = 10,
   fit_method = "glm", method_name = NULL, n.cores = 1, ...)
```

**Arguments**

<code>clinical_x</code>	clinical variables that will always be included in the model
<code>genomic_x</code>	genomic variables that will be penalized if a penalized model is used
<code>y</code>	response variables
<code>data</code>	dataframe if clinical formula is used
<code>clinical_formula</code>	formula for clinical variables
<code>family</code>	gaussian, binomial or poisson
<code>folds</code>	predefined partitions for cross validation
<code>k</code>	number of cross validation folds. A value of $k=n$ is leave one out cross validation.
<code>fit_method</code>	glm or glmnet used to fit the model
<code>method_name</code>	tracking variable to include in return dataframe
<code>n.cores</code>	Number of cores to be used
<code>...</code>	additional commands to glm or cv.glmnet

**Value**

returns a dataframe of predicted values and observed values. In addition, `method_name` is recorded if that variable is defined.

**Author(s)**

Ben Sherwood <ben.sherwood@ku.edu>

**Examples**

```
x <- matrix(rnorm(800),ncol=8)
y <- runif(100) < exp(1 + x[,1] + x[,5])/(1+exp(1 + x[,1] + x[,5]))
cv_results <- cv(x,y=y,method_name="without_formula")
combined_data <- data.frame(y=y,x1=x[,1],x5=x[,5])
gx <- x[,c(2,3,4,6,7,8)]
cvf <- cv(genomic_x=gx,clinical_formula=y~x1+x5,data=combined_data,method_name="with_form")
```

---

fit\_pred\_fold                      *Cross validation on fold i*

---

## Description

Cross validation on fold i

## Usage

```
fit_pred_fold(i, x, y, folds, fit_method, family, non_pen_vars = NULL, ...)
```

## Arguments

i	target partition
x	matrix of predictors
y	vector of responses
folds	defines how data is seperated into folds for cross validation
fit_method	model being used to fit the data
family	family used to fit the data
non_pen_vars	index of variables that will not be penalized if glmnet is used
...	additional commmands to glm or cv.glmnet

## Value

returns predictions for partition i

## Author(s)

Ben Sherwood <ben.sherwood@ku.edu>

## Examples

```
folds_10 <- randomly_assign(100,10)
x <- matrix(rnorm(800),ncol=8)
y <- runif(100) < exp(1 + x[,1] + x[,5])/(1+exp(1 + x[,1] + x[,5]))
fold_1_results <- fit_pred_fold(1,x,y,folds_10,"glm","binomial")
fold_2_results <- fit_pred_fold(2,x,y,folds_10,"glm","binomial")
```

---

randomly\_assign      *Assigns n samples into k groups*

---

**Description**

Assigns n samples into k groups

**Usage**

```
randomly_assign(n, k)
```

**Arguments**

n	sample size
k	number of groups

**Value**

returns a vector of length n with a random assignment of entries from 1 to k

**Author(s)**

Ben Sherwood <ben.sherwood@ku.edu>

**Examples**

```
n <- 100
folds_10 <- randomly_assign(n,10)
folds_5 <- randomly_assign(n,5)
```

---

roccv      *roccv: A package for creating ROC plots on cross validated data*

---

**Description**

roccv: A package for creating ROC plots on cross validated data

---

`rocplot`*Create ROC plot from cross validation results*

---

**Description**

Create ROC plot from cross validation results

**Usage**

```
rocplot(plot_data, ...)
```

**Arguments**

<code>plot_data</code>	dataframe with columns: response, prediction and method
<code>...</code>	additional commmands plot.roc such as main

**Value**

returns ROC plot

**Author(s)**

Ben Sherwood <ben.sherwood@ku.edu>

**Examples**

```
x <- matrix(rnorm(800),ncol=8)
y <- runif(100) < exp(1 + x[,1] + x[,5])/(1+exp(1 + x[,1] + x[,5]))
cv_results <- cv(x,y=y,method_name="without_formula")
combined_data <- data.frame(y=y,x1=x[,1],x5=x[,5])
gx <- x[,c(2,3,4,6,7,8)]
cvf <- cv(genomic_x=gx,clinical_formula=y~x1+x5,
data=combined_data,method_name="with_form")
total_results <- rbind(cv_results,cvf)
rocplot(total_results,main="rocplot test")
```

# Index

`cv`, [2](#)

`fit_pred_fold`, [3](#)

`randomly_assign`, [4](#)

`roccv`, [4](#)

`roccv-package (roccv)`, [4](#)

`rocplot`, [5](#)