

Package ‘CNVassoc’

March 26, 2012

Type Package

Title Association analysis of CNV data

Version 1.2

Date 2012-03-26

Depends R (>= 2.10.0), mixdist, mclust, survival

Suggests CGHcall, CGHregions, snow, CNVtools, xtable, MASS

Author Juan R González, Isaac Subirana

Maintainer Juan R González <jrgonzalez@creal.cat>

Description This package carries out association analysis of common copy number variants in population-based studies. This package includes functions for analysing association under a series of study designs (case-control, cohort, etc), using several dependent variables (class status, censored data, counts) as response, adjusting for covariates and considering various inheritance models. It also includes functions for inferring copy number (CNV genotype calling). Various classes and generic functions (print, summary, plot, anova, ...) have been created to facilitate the analysis.

License GPL (>= 2)

LazyLoad yes

URL <http://www.creal.cat/jrgonzalez/software.html>

Encoding latin1

BuildVignettes False

Repository CRAN

Date/Publication 2012-03-26 13:28:41

R topics documented:

A112	2
cnv	3
CNVassoc	5
CNVtest	7
dataMLPA	8
getProbs	8
getProbsRegions	9
getPvalBH	10
getQualityScore	10
multiCNVassoc	12
NeveCalled	12
NeveData	13
NeveRegions	14
plotSignal	14
simCNVdataBinary	16
simCNVdataCaseCon	17
simCNVdataNorm	18
simCNVdataPois	20
simCNVdataWeibull	21
SNPTEST	22
Index	23

A112

Copy Number Variant intensity data (from CNVtools)

Description

This data set has been obtained from CNVtools package in order to illustrate how CNVassoc and CNVtools compare

Usage

```
data(A112)
```

Source

Obtained from CNVtools package (Wellcome Trust Case Control Consortium)

References

<http://www.wtccc.org.uk/> "Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls." Wellcome Trust Case Control Consortium Nature. 2007;447;661-78.

cnv	<i>CNV object</i>
-----	-------------------

Description

cnv creates a 'cnv' object is returns TRUE if x is of class 'cnv' print gives a summary for an object of class 'cnv' including ... plot plots an object of class 'cnv' ...

Usage

```

cnv(x, batches, ...)
cnvDefault(x, num.copies, num.class, cnv.tol = 0.001, mix.method = "mixdist", check.probs = TRUE, thr
cnvBatches(intensities, batches, threshold.0, threshold.k, common.pi = TRUE, ...)
is.cnv(obj)
## S3 method for class 'cnv'
plot(x, ...)
## S3 method for class 'cnv'
print(x, digits = 4, ...)

```

Arguments

x	a vector of CNV intensity signal for each individual, or a matrix with CNV calling probabilities per row
num.copies	vector with copy number status values, i.e, number of copies or a vector of characters indicating loss ('l'), normal ('n') or gain ('g') for example
num.class	integer indicating how many classes CNV contains
cnv.tol	error tolerance when x is a probability matrix and row sums are not identical to one
mix.method	normal mixture fitting method when x is a vector of univariate CNV signal intensities. Current methods are "mixdist" that uses the function mix from the package mixdist, "mclust" that uses de function Mclust from the package mclust and "EMmixt" that uses an internal function EMmixt from the CNVassoc package. The last two are based on Expectation-Maximization procedure and the first one is based on quasi-Newton-Raphson procedure
check.probs	logical. If TRUE it checks weather row sums are equal to one +/- cnv.tol when x is a probability matrix
threshold.0	assigns zero copies (or first copy number status) to all individuals whose CNV signal intensity is lower than threshold.0
threshold.k	assigns k copies (or last copy number status) to all individuals whose CNV signal intensity is bigger than threshold.k
mu.ini	an opcional vector to specify the initial values of means when fitting a normal mixture to CNV intensity signal data
sigma.ini	an opcional vector to specify the initial values of standard deviations when fitting a normal mixture to CNV intensity signal data

<code>pi.ini</code>	an optional vector to specify the initial values of copy number status probabilities when fitting a normal mixture to CNV intensity signal data
<code>cutoffs</code>	a vector indicating the cut-off points to assign the copy number status assign individuals to the individuals according to the categories defined by these cut-off points on CNV intensity signal data
<code>check.alpha</code>	significance level to goodness-of-fit test indicating weather the normal mixture model to CNV intensity data has been fitted appropriately
<code>check.cnv</code>	logical. If TRUE, <code>cnv</code> functions returns and error when normal mixture model does not fit well to the univariate CNV intensity signal data
<code>var.equal</code>	logical. If TRUE, standard deviation are supposed to be the same for all copy number status when fitting univariate CNV intensity signal data
<code>intensities</code>	a vector with the univariate CNV intensity signal data
<code>batches</code>	a vector indicating the batch (leave it missing if no batch effect is present)
<code>common.pi</code>	logical. If TRUE, copy number status probabilities for each individual are computed estimating specific means and standard deviations separately for every batch, but the same population copy number status probabilities for all batches. It is suggested to leave it as TRUE
<code>obj</code>	an object of any class
<code>digits</code>	number of digits when printing a <code>cnv</code> object
<code>...</code>	other arguments passed to <code>cnvDefault</code> , <code>print.default</code> or <code>plot.cnv</code> . The arguments passed to <code>plot.cnv</code> are the same as the ones for the <code>plotSignal</code> function

Details

When argument `batches` is not specified, then `cnvDefault` is used, otherwise `cnvBatch` is called. If univariate CNV intensity signal data is used to create the `cnv` class object, then one can introduce the batch effect if it necessary. But, if other algorithms have been used previously and the `cnv` class object is created directly from the CNV calling probabilities matrix, then it is not possible to specify the batch argument. The batch effect is important when cases and controls have been genotyped in different platforms for example. In this situations, the platform should be introduced in the batch argument as a vector indicating which platform every CNV intensity signal data comes from. Generic plot function applied on a 'cnv' class object performs two types of plots whether 'cnv' class object has been created from univariate CNV intensity signal data or whether it has been created directly from a probability matrix provided by any CNV calling algorithm. The first type is a plot similar to the one created by `plotSignal` function, and the second type is a barplot.

Value

`cnv` return an object of class 'cnv' with generic function such as `print` or `plot` implemented for this kind of objects. `is.cnv` is a function that returns TRUE of FALSE weather `obj` is of class 'cnv' or not.

References

Gonzalez JR, Subirana I, Escaramis G, Peraza S, Caceres A, Estivill X and Armengol L. Accounting for uncertainty when assessing association between copy number and disease: a latent class model. *BMC Bioinformatics*, 2009;10:172.

See Also

[CNVassoc](#), [plotSignal](#)

Examples

```
data(dataMLPA)
CNV <- cnv(x = dataMLPA$Gene2, threshold.0 = 0.01, mix.method = "mixdist")
CNV
plot(CNV)
```

CNVassoc

Association analysis between a CNV and phenotype

Description

This function performs an association analysis between a CNV and a dependent variable (phenotype) using a latent class model that incorporates the uncertainty arising from calling procedure. The phenotype may be quantitative or categorical. In the second case (e.g. case-control studies) this variable must be coded as 1 (for cases) and 0 (for controls). The association can be adjusted for other covariates (e.g. clinical covariates, stratification, ...)

Usage

```
CNVassoc(formula, data, subset, na.action, model = "multiplicative", family = "binomial", tol = 1e-06,
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. Right side of ~ should have an object of class 'cnv'.
data	an optional data frame, list or environment (or object coercible by 'as.data.frame' to a data frame) containing the variables in the model. If not found in 'data', the variables are taken from 'environment(formula)'.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
na.action	a function which indicates what should happen when the data contain 'NA's. The default is set by the 'na.action' setting of 'options', and is 'na.fail' if that is unset. The 'factory-fresh' default is 'na.omit'. Another possible value is 'NULL', no action. Value 'na.exclude' can be useful.

model	Genetic model to be tested. Possible values are "multiplicative" (model free, e.g. co-dominant) or "additive", partial matching allowed. Default value is "multiplicative".
family	a description of the error distribution and link function to be used in the model. This must be a character string naming a family function. Possible values are "binomial", "gaussian", "poisson" or "weibull". Default value is "binomial"
tol	Tolerance for convergence in fitting model. Default value is 1e-06.
max.iter	Maximum number of iterations in fitting model. Default value is 30.
emsteps	Number of iterations using Expectation Maximization (EM) algorithm to set initial values before using Newton-Rapson (NR) in fitting model. Default value is zero, that means that EM step is not performed
verbose	logical. If TRUE parameter values for each iteration are shown in the console. Default value is FALSE
coef.start	initial values for coefficients in NR procedure
sigma.start	initial values for scale parameter (only for "gaussian") in NR procedure
alpha.start	initial values for shape parameter (only for "weibull") in NR procedure

Value

An object of class 'CNVassoc'.

'print' returns model parameter estimates

'summary' returns a summary table similar to summary.glm

'anova' performs a Likelihood Ratio Test comparing two nested models fitted using CNVassoc

'logLik' returns the log-likelihood of a model fitted using CNVassoc

See examples for further illustration about all previous issues.

References

Gonzalez JR, Subirana I, Escaramis G, Peraza S, Caceres A, Estivill X and Armengol L. Accounting for uncertainty when assessing association between copy number and disease: a latent class model. *BMC Bioinformatics*, 2009;10:172.

See Also

[cnv](#), [CNVtest](#)

Examples

```
data(dataMLPA)
CNV <- cnv(x = dataMLPA$Gene2, threshold.0 = 0.01, mix.method = "mixdist")
modmul <- CNVassoc(casco ~ CNV, data = dataMLPA, model = "mul")
modmul
summary(modmul)
anova(modmul, update(modmul, model="add"))
logLik(modmul)
```

`CNVtest`*Testing association between a CNV and phenotype*

Description

This function perform a Wald test or a Likelihood Ratio Test (LRT) to determine whether a CNV is associated with the phenotype.

Usage

```
CNVtest(x, type = "Wald")
## S3 method for class 'CNVtest'
print(x, ...)
```

Arguments

<code>x</code>	An object of class 'CNVassoc'
<code>type</code>	The statistical test used. Possible values are "Wald" for Wald test or "LRT" for Likelihood Ratio Test
<code>...</code>	Further arguments passed to or from other methods

Value

An object of class 'CNVtest' with methods for the generic function `print`, returning the Chi-squared value, its degrees of freedom and the corresponding p-value of CNV significance in the associated test fitted by `CNVassoc` function.

References

Gonzalez JR, Subirana I, Escaramis G, Peraza S, Caceres A, Estivill X and Armengol L. Accounting for uncertainty when assessing association between copy number and disease: a latent class model. *BMC Bioinformatics*, 2009;10:172.

See Also

[CNVassoc](#), [cnv](#)

Examples

```
data(dataMLPA)
CNV<- cnv(x = dataMLPA$Gene2, threshold.0 = 0.01, mix.method = "mixdist")
mod<-CNVassoc(formula = casco ~ CNV, data = dataMLPA, model = "mul")
CNVtest(mod, type = "LRT")
CNVtest(mod, type = "Wald")
```

dataMLPA	<i>MLPA data</i>
----------	------------------

Description

This data set contains data from a MLPA assay for a case-control study. The data has intensities for two genes that can be used to infer copy number status. It also contains the TRUE copy number status obtained by using PCR technology.

Usage

```
data(dataMLPA)
```

Format

dataMLPA is a data.frame with the following columns:

id	The unique identifiers of individuals
casco	Case-control status 0:control 1:case
Gene1	Intensities for Gene1
Gene2	Intensities for Gene2
PCR.Gene1	True copy number status for Gene1
PCR.Gene2	True copy number status for Gene2
quanti	Simulated continuous variable. It was generated to illustrate how to perform association analysis between a CNV
cov	Simulated continuous variable. It was generated to illustrate how to perform association adjusting for covariates

Source

This data set was generated and kindly provided by Estivill's lab. The data is still unpublished and it has been made available in a blinded format for reproducing the findings presented in the paper

References

Gonzalez JR, Subirana I, Escaramis G, Peraza S, Caceres A, Estivill X and Armengol L. Accounting for uncertainty when assessing association between copy number and disease: a latent class model. *BMC Bioinformatics*, 2009;10:172.

getProbs	<i>Get posterior probabilities from an object of class 'cnv' or 'CGHcall'</i>
----------	---

Description

This function creates a list where each component correspond to a given probe. Any component contains the posterior probabilities obtained from CGHcall algorithm.

Usage

```
getProbs(x)
## S3 method for class 'cghCall'
getProbs(x)
## S3 method for class 'cnv'
getProbs(x)
```

Arguments

x an object of class 'CGHcall' or 'cnv'

Value

A list where each component correspond to a given probe. Any component contains the posterior probabilities obtained from CGHcall algorithm

Note

See vignette for an example.

Author(s)

This function was created using a script kindly provided by Mark van de Wiel

getProbsRegions *Get posterior probabilities for blocks/regions*

Description

See vignette for further details.

Usage

```
getProbsRegions(probs, regions, intensities)
```

Arguments

probs probabilities from CGH calling algorithms
regions probs that define a segment
intensities mean probe intensities for each region

Note

See vignette for an example.

`getPvalBH`*Corrected p values using Benjamini & Hochberg approach*

Description

This functions corrects the association p-values using the Benjamini & Hochberg approachby for multiple testing.

Usage

```
getPvalBH(x)
```

Arguments

x a list containing p values

Details

This function calls 'p.adjust' to compute 'BH' correction

Value

A data frame with the blocks and corrected p-values

See Also

[p.adjust](#)

`getQualityScore`*Computes a quality score for a CNV fit*

Description

This function provides different types of measurements of uncertainty after CNV calling

Usage

```
getQualityScore(x, ...)  
## Default S3 method:  
getQualityScore(x, sds, w, type, iter = 10000, threshold = 0.1, ...)  
## S3 method for class 'cnv'  
getQualityScore(x, type = "class", iter = 10000, threshold = 0.1, ...)
```

Arguments

x	and object of class <code>cnv</code> or means vector of intensity signal for each copy number status
...	further arguments passed to or from <code>getQualityScore</code> methods
type	the type of quality score measurement computed. Possible values are "class", "CNVtools" or "CANARY" (see Details)
iter	number of iterations when <code>type="class"</code> or <code>type="CANARY"</code> is specified
threshold	a value to compute the proportion of sample individuals with confidence score bigger than it (see Details)
sds	standard deviations vector of intensity signal for each copy number status
w	copy number status proportions vector

Details

The quality scores measures how well the clusters are separated. It compares the locations of the means with the standard error for each pair of adjacent cluster. Obviously, except for probability of good classification (`type="class"`), the lower quality score the highest uncertainty. There are 3 possible types of quality score measurements: "class": probability of good classification), "CNVtools": the score defined in 'CNVtools' package) and "CANARY": proportions of sample individuals with confidence score bigger than `threshold`. The confidence score is defined as the ratio between the second biggest copy number call probability divided by the biggest one.

Value

An object of class `getQualityScore` with a single number of quality score.

Note

For `cnv` objects created directly from probabilities and not from fitting a univariate intensity signal, only "class" quality score type can be calculated.

Examples

```
data(dataMLPA)
CNV<-cnv(x = dataMLPA$Gene2, threshold.0 = 0.01, mix.method = "mixdist")
getQualityScore(CNV,type="class")
getQualityScore(CNV,type="CNVtools")
getQualityScore(CNV,type="CANARY")
```

multiCNVassoc	<i>Association between several CNVs and disease</i>
---------------	---

Description

This function repeatedly calls CNVassoc function

Usage

```
multiCNVassoc(x, formula, ...)
```

Arguments

x	a list of calling probabilities matrix for each CNV
formula	see 'formula' argument of CNVassoc function.
...	other arguments passed through 'CNVassoc' function.

Details

See vignette for an example

Value

A list of p-values for each CNV

References

Gonzalez JR, Subirana I, Escaramis G, Peraza S, Caceres A, Estivill X and Armengol L. Accounting for uncertainty when assessing association between copy number and disease: a latent class model. *BMC Bioinformatics*, 2009;10:172.

See Also

[CNVassoc](#)

NeveCalled	<i>Breast Cancer aCGH data with CGHcall</i>
------------	---

Description

Breast cancer aCGH data called with CGHcall with default settings, containing 2621 features and 50 samples

Usage

```
data(NeveData)
```

Format

An object of class CGHcall

Source

Neve RM, Chin K, Fridlyand J et al. A collection of breast cancer cell lines for the study of functionally distinct cancer subtypes. *Cancer Cell*, 10:515-527, 2006.

Gonzalez JR, Subirana I, Escaramis G, Peraza S, Caceres A, Estivill X and Armengol L. Accounting for uncertainty when assessing association between copy number and disease: a latent class model. *BMC Bioinformatics*, 2009;10:172.

 NeveData

Breast Cancer aCGH data

Description

This data set contains breast cancer data studied by Neve et al. (2006). The data consists on CGH arrays of 1MB resolution. The authors chose the 50 samples that could be matched to the name tokens of caArrayDB data (June 9th 2007).

This object is a list with two components. The first component corresponds to a data.frame containing 2621 rows and 54 columns with aCGH data (4 columns for the annotation and 50 log2ratio intensities). The second component contains information about strogen receptor positivity (dichotomous variable; 0: negative, 1: positive)

Usage

```
data(NeveData)
```

Format

The first component is a data.frame with the following columns:

Clone	The unique identifiers of array elements
Chrom	Chromosome number of each array element
kb	Chromosomal position in bp of each array element
kb.1	Chromosomal position in bp of each array element
X600MPE	Raw log2 ratios for breast cancer sample X600MPE
AU565	Raw log2 ratios for breast cancer sample AU565
...	...
ZR75B	Raw log2 ratios for breast cancer sample ZR75B

The second component is a vector with information about strogen receptor positivity

Source

Data are freely available from the bioconductor website (<http://www.bioconductor.org/>) through the package Neve2006

NeveRegions

Breast Cancer aCGH data with CGHcall

Description

Breast cancer aCGH data called with CGHcall with default settings, containing 2621 features and 50 samples

Usage

```
data(NeveRegions)
```

Format

An object of class CGHregions

Source

Neve RM, Chin K, Fridlyand J et al. A collection of breast cancer cell lines for the study of functionally distinct cancer subtypes. *Cancer Cell*, 10:515-527, 2006.

Gonzalez JR, Subirana I, Escaramis G, Peraza S, Caceres A, Estivill X and Armengol L. Accounting for uncertainty when assessing association between copy number and disease: a latent class model. *BMC Bioinformatics*, 2009;10:172.

plotSignal

plots the intensities of a CNV univariate signal data

Description

This function creates a plot with probe intensity

Usage

```
plotSignal(x, my.colors = c("black", "red", "blue"), ylab = "Peak Intensity", xlab = c("individuals", "P
```

Arguments

x	A vector with probe intensities
my.colors	Colours for each copy number status.
ylab	Label of y-axis
xlab	Label of x-axis
case.control	Vector indicating case-control status
cex.legend	Size of legend
dens.bw	Adjustment for intensity signal density curve. See argument 'bw' of density function for more details
dens.adjust	Adjustment for intensity signal density curve. See argument 'adjust' of density function for more details
n	integer indicating the number of points to be placed on the plot interactively (using locator)) to define the thresholds that separate the different copy number status, colouring the points differently according to the assigned copy number status. If it zero or negative it makes no possible to place the threshold points.
...	Other arguments passed to plot.default

Details

See vignette for further description

References

Gonzalez JR, Subirana I, Escaramis G, Peraza S, Caceres A, Estivill X and Armengol L. Accounting for uncertainty when assessing association between copy number and disease: a latent class model. *BMC Bioinformatics*, 2009;10:172.

See Also

[cnv](#)

Examples

```
data(dataMLPA)
plotSignal(dataMLPA$Gene2)
```

simCNVdataBinary *Simulation of CNV and discrete traits*

Description

This function simulates intensity for a CNV and a binary trait response for different scenarios

Usage

```
simCNVdataBinary(n, mu.surrog, sd.surrog, w, p0, or, cnv.random = FALSE)
```

Arguments

n	number of simulated individuals
mu.surrog	a vector of intensity signal means for every copy number status
sd.surrog	a vector of intensity signal standard deviations for every copy number status
w	a vector of copy number status proportions
p0	prevalence of disease (trait) for populations with zero copies (reference category)
or	a vector of odds ratio for one, two,... copies respect to zero copies
cnv.random	A logical value. TRUE means that copy number status is drawn under a multinomial distribution with proportions indicated by 'w'. FALSE means that the real simulated frequency is always the same and is rounded to the most similar integer to the frequencies indicated by 'w'. Default value is FALSE

Details

This function is useful to calculate the power of association models with binary traits under different scenarios, e.g. setting different degrees of association (odds ratios), considering different degrees of uncertainty controlled by the distribution of intensity signal data, i.e. mean `mu.surrog`, standard deviation `sd.surrog` and proportion `w`, etc.

Value

Data frame with individual simulated data per row and with the following variables:

resp	Trait (response) variable following a Bernoulli distribution given the CNV status
surrog	Signal intensity following a mixture of normals with means, standard deviations and proportions specified by <code>mu.surrog</code> , <code>sd.surrog</code> and <code>w</code> respectively.
cnv	True copy number status

See Also

[simCNVdataCaseCon](#), [simCNVdataNorm](#), [simCNVdataPois](#), [simCNVdataWeibull](#), [cnv](#), [CNVassoc](#)

Examples

```

maf<-0.3
set.seed(123)
simData<-simCNVdataBinary(n=1000, mu.surrog=c(0,0.5,1), sd.surrog=rep(0.15,3),
  w=c((1-maf)^2,2*maf*(1-maf),maf^2), p0=0.1, or=c(1.3,1.3^2), cnv.random = FALSE)
CNV<-cnv(simData$surrog,mix.method="EMmixt")
getQualityScore(CNV,type="CNVtools")
mod<-CNVassoc(resp~CNV,data=simData,family="binomial")
CNVtest(mod)
summary(mod)

```

simCNVdataCaseCon *Simulation of CNV in a case-control study design*

Description

This function simulates intensity for a CNV within cases and control groups for different scenarios

Usage

```
simCNVdataCaseCon(n0, n1, w0, or, mu.surrog0, sd.surrog0, mu.surrog1 = mu.surrog0, sd.surrog1 = sd.surrog0, random = TRUE)
```

Arguments

n0	number of controls simulated
n1	number of cases simulated
w0	vector of proportions of copy number status in controls
or	a vector of odds ratio for one, two,... copies respect to zero copies
mu.surrog0	vector of means of CNV intensity signal, per copy number status, in control group
sd.surrog0	vector of standard deviations of CNV intensity signal, per copy number status, in control group
mu.surrog1	vector of means of CNV intensity signal, per copy number status, in control group
sd.surrog1	vector of standard deviations of CNV intensity signal, per copy number status, in control group
random	A logical value. TRUE means that individuals (rows) are randomly permuted, and FALSE means that simulated 'data.frame' contains controls first and then cases. Default value is TRUE

Details

This function is useful to calculate the power of association models in a case control study design under different scenarios ,e.g. setting different degrees of association (odds ratios), considering different degrees of uncertainty controlled by the distribution of intensity signal data, i.e. mean mu.surrog, standard deviation sd.surrog and proportion w, etc.

Value

Data frame with individual simulated data per row and with the following variables:

resp	Trait (response) variable with 0 or 1 if the individual is a control or a case respectively
surrog	Signal intensity following a mixture of normals with means, standard deviations and proportions specified by mu.surrog, sd.surrog and w respectively, within cases and controls
cnv	True copy number status

See Also

[simCNVdataBinary](#), [simCNVdataNorm](#), [simCNVdataPois](#), [simCNVdataWeibull](#), [cnv](#), [CNVassoc](#)

Examples

```
maf<-0.3
set.seed(123)
simData<-simCNVdataCaseCon(n0=1000, n1=1000, mu.surrog0=c(0,0.5,1), sd.surrog0=rep(0.15,3),
  mu.surrog1=c(0,0.5,1), sd.surrog1=rep(0.15,3),
  w0=c((1-maf)^2,2*maf*(1-maf), maf^2), or=c(1.3,1.3^2),
  random = FALSE)
CNV<-cnv(simData$surrog,mix.method="EMmixt")
getQualityScore(CNV,type="CNVtools")
mod<-CNVassoc(resp~CNV,data=simData,family="binomial")
CNVtest(mod)
summary(mod)
```

simCNVdataNorm

Simulation of CNV and quantitative traits

Description

This function simulates intensity for a CNV and a quantitative trait response for different scenarios

Usage

```
simCNVdataNorm(n, mu.surrog, sd.surrog, w, mu.y, sd.y, cnv.random = FALSE)
```

Arguments

n	An integer indicating the desired number of individuals to be simulated
mu.surrog	A vector containing the signal (surrogate variable) means for every copy number status (latent classes). Its length must be equal to the number of latent classes
sd.surrog	A vector containing the signal standard deviation for every copy number status. Its length must be equal to mu.surrog.

w	A vector containing the frequencies for every copy number status. Its length must be equal to mu.surrog and its components must sum up one.
mu.y	A vector containing the means of the response variable for every copy number status. Its length must be equal to mu.surrog.
sd.y	A single number indicating the residual standard deviation
cnv.random	A logical value. TRUE means that copy number status is drawn under a multinomial distribution with proportions indicated by 'w'. FALSE means that the real simulated frequency is always the same and is rounded to the most similar integer to the frequencies indicated by 'w'. Default value is FALSE

Details

This function is useful to calculate the power of association models for a continuous (normal-distributed) trait under different scenarios ,e.g. setting different degrees of association (effects), considering different degrees of uncertainty controlled by the distribution of intensity signal data, i.e. mean mu.surrog, standard deviation sd.surrog and proportion w, etc.

Value

Data frame with individual simulated data per row and with the following variables:

resp	Continous trait variable (response)
surrog	Signal intensity following a mixture of normals with means, standard deviations and proportions specified by mu.surrog, sd.surrog and w respectively
cnv	True copy number status

See Also

[simCNVdataBinary](#), [simCNVdataCaseCon](#), [simCNVdataPois](#), [simCNVdataWeibull](#), [cnv](#), [CNVassoc](#)

Examples

```
set.seed(123)
maf<-0.3
effect<-3
simData<-simCNVdataNorm(n=1000, mu.surrog=c(0,0.5,1), sd.surrog=rep(0.15,3),
  w=c((1-maf)^2,2*maf*(1-maf), maf^2), mu.y=100+c(0,effect,2*effect), sd.y=rep(20,3), cnv.random = FALSE)
CNV<-cnv(simData$surrog,mix.method="EMmixt")
getQualityScore(CNV,type="CNVtools")
mod<-CNVassoc(resp~CNV,data=simData,family="gaussian",emsteps=10)
CNVtest(mod)
summary(mod)
```

simCNVdataPois *Simulate Poisson data*

Description

This function simulates intensity for a CNV and a discrete counting trait response for different scenarios

Usage

```
simCNVdataPois(n, mu.surrog, sd.surrog, w, lambda, cnv.random = FALSE)
```

Arguments

n	An integer indicating the desired number of individuals to be simulated
mu.surrog	A vector containing the signal (surrogate variable) means for every copy number status (latent classes). Its length must be equal to the number of latent classes
sd.surrog	A vector containing the signal standard deviation for every copy number status. Its length must be equal to mu.surrog.
w	A vector containing the frequencies for every copy number status. Its length must be equal to mu.surrog and its components must sum up one.
lambda	A vector containing the means of the response variable for every copy number status. Its length must be equal to mu.surrog.
cnv.random	A logical value. TRUE means that copy number status is drawn under a multinomial distribution with proportions indicated by 'w'. FALSE means that the real simulated frequency is always the same and is rounded to the most similar integer to the frequencies indicated by 'w'. Default value is FALSE

Details

This function is useful to calculate the power of association models for discrete counting trait under different scenarios ,e.g. setting different degrees of association (risk ratios), considering different degrees of uncertainty controlled by the distribution of intensity signal data, i.e. mean mu.surrog, standard deviation sd.surrog and proportion w, etc.

Value

Data frame with individual simulated data per row and with the following variables:

resp	Discrete variable with simulated counts (response)
surrog	Signal intensity following a mixture of normals with means, standard deviations and proportions specified by mu.surrog, sd.surrog and w respectively
cnv	True copy number status

See Also

[simCNVdataBinary](#), [simCNVdataCaseCon](#), [simCNVdataNorm](#), [simCNVdataWeibull](#), [cnv](#), [CNVassoc](#)

Examples

```

set.seed(123)
rr<-1.5
maf<-0.3
simData<-simCNVdataPois(n=1000, mu.surrog=c(0,0.5,1), sd.surrog=rep(0.15,3),
  w=c((1-maf)^2,2*maf*(1-maf), maf^2), lambda=3*c(1,rr,rr^2), cnv.random = FALSE)
CNV<-cnv(simData$surrog,mix.method="EMmixt")
getQualityScore(CNV,type="CNVtools")
mod<-CNVassoc(resp~CNV,data=simData,family="poisson",emsteps=10)
CNVtest(mod)
summary(mod)

```

simCNVdataWeibull	<i>Simulate of CNV and a right censored Weibull distributed trait</i>
-------------------	---

Description

This function simulates intensity for a CNV and a time to event response (followed-up cohort study design) for different scenarios

Usage

```
simCNVdataWeibull(n, mu.surrog, sd.surrog, w, lambda, shape, time.cens = Inf, cnv.random = FALSE)
```

Arguments

n	An integer indicating the desired number of individuals to be simulated
mu.surrog	A vector containing the signal (surrogate variable) means for every copy number status (latent classes). Its length must be equal to the number of latent classes
sd.surrog	A vector containing the signal standard deviation for every copy number status. Its length must be equal to mu.surrog.
w	A vector containing the frequencies for every copy number status. Its length must be equal to mu.surrog and its components must sum up one.
lambda	A vector containing the means of the response variable for every copy number status
shape	A vector containing the shape of the response variable for every copy number status
time.cens	Censoring time, e.g. end of follow-up
cnv.random	A logical value. TRUE means that copy number status is drawn under a multinomial distribution with proportions indicated by 'w'. FALSE means that the real simulated frequency is always the same and is rounded to the most similar integer to the frequencies indicated by 'w'. Default value is FALSE

Value

Data frame with individual simulated data per row and with the following variables:

resp	Time to event or censoring variable (response)
cens	Censoring indicator
surrog	Signal intensity following a mixture of normals with means, standard deviations and proportions specified by <code>mu.surrog</code> , <code>sd.surrog</code> and <code>w</code> respectively
cnv	True copy number status

See Also

[simCNVdataBinary](#), [simCNVdataCaseCon](#), [simCNVdataPois](#), [simCNVdataNorm](#), [cnv](#), [CNVassoc](#)

Examples

```
library(survival)
maf<-0.3
hr<-1.5
set.seed(123)
simData<-simCNVdataWeibull(n=4000, mu.surrog=c(0,0.5,1), sd.surrog=rep(0.15,3),
  w=c((1-maf)^2,2*maf*(1-maf), maf^2), lambda=0.05*c(1,hr,hr^2), shape=rep(1,3), time.cens=1.5, cnv.random =
CNV<-cnv(simData$surrog,mix.method="EMmixt")
getQualityScore(CNV,type="CNVtools")
mod<-CNVassoc(Surv(resp, cens)~CNV,data=simData,family="weibull")
CNVtest(mod)
summary(mod)
```

SNPTEST

Case-control data with SNPTEST format

Description

Case-control data for 200 individuals and 500 SNPs using SNPTEST format (e.g. probabilities of AA, AB and BB genotypes)

Usage

```
data(SNPTEST)
```

Format

Two objects: cases and controls. Each object is a matrix with 100 rows and 1505 columns. Column 1-5 correspond to annotation. Files 6-1505 correspond to the probabilities of being AA, AB or BB for 500 SNPs

Source

<http://www.stats.ox.ac.uk/~marchini/software/gwas/snptest.html>

Index

*Topic **datasets**

A112, [2](#)
dataMLPA, [8](#)
NeveCalled, [12](#)
NeveData, [13](#)
NeveRegions, [14](#)
SNPTEST, [22](#)

*Topic **misc**

cnv, [3](#)
CNVassoc, [5](#)
CNVtest, [7](#)
getProbsRegions, [9](#)
getPvalBH, [10](#)
getQualityScore, [10](#)
multiCNVassoc, [12](#)
simCNVdataNorm, [18](#)

*Topic **utilities**

getProbs, [8](#)
plotSignal, [14](#)
simCNVdataBinary, [16](#)
simCNVdataCaseCon, [17](#)
simCNVdataPois, [20](#)
simCNVdataWeibull, [21](#)

A112, [2](#)

anova.CNVassoc (CNVassoc), [5](#)

cases (SNPTEST), [22](#)

cnv, [3](#), [6](#), [7](#), [15](#), [16](#), [18–20](#), [22](#)

CNVassoc, [5](#), [5](#), [7](#), [12](#), [16](#), [18–20](#), [22](#)

cnvBatches (cnv), [3](#)

cnvDefault (cnv), [3](#)

CNVtest, [6](#), [7](#)

controls (SNPTEST), [22](#)

dataMLPA, [8](#)

density, [15](#)

getProbs, [8](#)

getProbsRegions, [9](#)

getPvalBH, [10](#)

getQualityScore, [10](#)

is.cnv (cnv), [3](#)

logLik.CNVassoc (CNVassoc), [5](#)

multiCNVassoc, [12](#)

NeveCalled, [12](#)

NeveData, [13](#)

NeveRegions, [14](#)

p.adjust, [10](#)

plot.cnv (cnv), [3](#)

plotSignal, [4](#), [5](#), [14](#)

print.anova.CNVassoc (CNVassoc), [5](#)

print.cnv (cnv), [3](#)

print.CNVassoc (CNVassoc), [5](#)

print.CNVtest (CNVtest), [7](#)

print.QualityScore (getQualityScore), [10](#)

print.summary.CNVassoc (CNVassoc), [5](#)

simCNVdataBinary, [16](#), [18–20](#), [22](#)

simCNVdataCaseCon, [16](#), [17](#), [19](#), [20](#), [22](#)

simCNVdataNorm, [16](#), [18](#), [18](#), [20](#), [22](#)

simCNVdataPois, [16](#), [18](#), [19](#), [20](#), [22](#)

simCNVdataWeibull, [16](#), [18–20](#), [21](#)

SNPTEST, [22](#)

summary.CNVassoc (CNVassoc), [5](#)