Package 'waffect'

February 20, 2015

Type Package

Title A package to simulate constrained phenotypes under a disease model H1

Version 1.2

Date 2012-03-30

Author Gregory Nuel and Vittorio Perduca

Maintainer G. Nuel <gregory.nuel@parisdescartes.fr>

Description waffect (pronounced 'double-u affect' for 'weighted affectation') is a package to simulate phenotypic (case or control) datasets under a disease model H1 such that the total number of cases is constant across all the simulations (the constrain in the title). The package also makes it possible to generate phenotypes in the case of more than two classes, so that the number of phenotypes belonging to each class is constant across all the simulations. waffect is used to assess empirically the statistical power of Genome Wide Association studies.

License GPL (>= 3)

LazyLoad yes

Depends Rcpp (>= 0.9.5)

Suggests pROC

LinkingTo Rcpp

Repository CRAN

Date/Publication 2012-04-11 13:32:12

NeedsCompilation yes

R topics documented:

`waffect-package` *A package to simulate constrained phenotypes under a disease model H1*

Description

waffect (pronounced 'double-u affect' for 'weighted affectation') is a package to simulate phenotypic (case or control) datasets under a disease model H1 such that the total number of cases is constant across all the simulations (the constrain in the title). The package also makes it possible to generate phenotypes in the case of more than two classes, so that the number of phenotypes belonging to each class is constant across all the simulations. **waffect** is used to assess empirically the statistical power of Genome Wide Association studies without generating genotypes for each simulation. The vignette includes a tutorial for performing such power studies.

Details

waffect implements three alternative methods to simulate phenotypes with a fixed number of cases and controls and under a given disease model: i) an exact and efficient backward sampling algorithm; ii) a numerical Markov Chain Monte-Carlo (MCMC) approach; iii) a simple rejection algorithm. The backward algorithm is the default method. The rejection algorithm is deprecated. More details can be found in the companion article [1].

Installing and Using

To install this package, make sure you are connected to the Internet and issue the following command in the R prompt: `install.packages("waffect")`

To load the package in R: `library(waffect)`

To open the .pdf file of the vignette: `vignette("waffect-tutorial")`

Citation

If you use **waffect** in published research, please cite the companion article:

Perduca V, Sinoquet C, Mourad R, Nuel G: Alternative Methods for H1 Simulations in Genome-Wide Association Studies. Hum Hered 2012;73:95-104.

Type `citation("waffect")` for a BibTeX entry.

The authors would be glad to hear how **waffect** is employed. You are kindly encouraged to notify Gregory Nuel <gregory.nuel@parisdescartes.fr> and Vittorio Perduca <vittorio.perduca@parisdescartes.fr> about any work you publish that makes use of **waffect**.

Functions

`waffect` high level function for simulating phenotypes in the binary (case/control) and mulitclass cases

`waffectbin` low level function for simulating phenotypes in the binary case (not documented)

Datasets

This package comes with two datasets to be used for the fictive GWAs power analysis outlined in the vignette:

`ped` genotypic dataset in `.ped` format

`map` file in `.map` format describing the SNPs in the ped file

The signals of association to be used in the fictive GWAs power analysis are also included:

`p1_H0` Signal of association of the statistic S1 under H0

`p1_H1` Signal of association of the statistic S1 under H1

`p2_H0` Signal of association of the statistic S2 under H0

`p2_H1` Signal of association of the statistic S2 under H1

References

[1] Perduca V, Sinoquet C, Mourad R, Nuel G: Alternative Methods for H1 Simulations in Genome-Wide Association Studies. Hum Hered 2012;73:95-104

See Also

Documentation for the function `waffect`. Useful information can be also found in the vignette: `vignette("waffect-tutorial")`.

Examples

```
## Not run: Typical usage to simulate case/control phenotypes under H1 (in this example: 5 individuals, 2 cases, 3
waffect(prob =  c(0.5, 0.2, 0.9, 0.7, 0.1), count = c(2,3), label = c(1,0))

## Not run: We can just specify the number of cases:
waffect(prob =  c(0.5, 0.2, 0.9, 0.7, 0.1), count = 2, label = c(1,0))

## Not run: We can change the labels:
waffect(prob =  c(0.5, 0.2, 0.9, 0.7, 0.1), count = 2, label = c("case", "control"))

## Not run: We can also simulate under H0 just entering the number of cases and controls:
waffect(count=c(2,3), label = c(1,0))

## Not run: Simulation of phenotypes in the multiclass case (in this example: 4 individuals and  3 classes, 1 indi
pi = cbind(c(0.5,0.3,0.2), c(0.2,0.2,0.6), c(0.1,0.7,0.2), c(0.4,0.3,0.3))
waffect(prob =  pi, count = c(1,2,1), label = 1:3)
```

map *SNP description in MAP format*

Description

This dataset describes the SNPs to be considered in the tutorial for assessing the statistical power of GWAs with waffect (see the vignette). There are 1000 rows, one for each marker, and 4 columns:

- chr: the chromosome, here all the SNPs are on X
- SNP_id: the SNP identifier, here an integer from 1 to 1000
- dist: the genetic distance, here a dummy variable set to 0 for all the SNPs
- bp_pos: the base-pair position, here an integer from 1 to 1000.

Before using this dataset with PLINK it is necessary to save it into a file with extension .map (see the vignette).

Usage

```
data(map)
```

Format

A table with 1000 rows and 4 columns.

p1_H0 *Signal of association of the statistic S1 under H0*

Description

This dataset contains the signal of association of the statistic S1 under the null hypothesis H0. It consists of 200 values, one for each simulations under H0. See the vignette for the definition of S1 and more details: `vignette("waffect-tutorial")`.

Usage

```
data(p1_H0)
```

Format

A table with one column and 200 rows

p1_H1 *Signal of association of the statistic S1 under H1*

Description

This dataset contains the signal of association of the statistic S1 under the alternative hypothesis H1. It consists of 200 values, one for each simulations under H1. See the vignette for the definition of S1 and more details: `vignette("waffect-tutorial")`.

Usage

```
data(p1_H1)
```

Format

A table with one column and 200 rows

p2_H0 *Signal of association of the statistic S2 under H0*

Description

This dataset contains the signal of association of the statistic S2 under the null hypothesis H0. It consists of 200 values, one for each simulations under H0. See the vignette for the definition of S2 and more details: `vignette("waffect-tutorial")`.

Usage

```
data(p2_H0)
```

Format

A table with one column and 200 rows

p2_H1 *Signal of association of the statistic S2 under H1*

Description

This dataset contains the signal of association of the statistic S2 under the alternative hypothesis H1. It consists of 200 values, one for each simulations under H1. See the vignette for the definition of S2 and more details: `vignette("waffect-tutorial")`.

Usage

```
data(p2_H1)
```

Format

A table with one column and 200 rows

ped *Genotypic dataset in PED format*

Description

This dataset contains the genotypes of 2000 SNPs to be used with the tutorial for assessing the statistical power of GWAs with waffect (see the vignette). It has 100 rows (one for each individual) and 2007 columns:

- fam_id: family ID, here an integer from 1 to 100
- ind_id: individual ID, an integer from 1 to 100
- pat_id: paternal ID, here a dummy variable set to 0 for all individuals
- mat_id: maternal ID, dummy variable
- sex: all females (coded by 2)
- pheno: observed phenotypes; codes: case = 2, control = 1.

Genotypes (column 7 onwards) are biallelic, one allele for each column. For instance, columns 7 and 8 contain the two alleles for SNP1 and columns 9 and 10 contain the alleles for SNP2. Before using this dataset with PLINK it is necessary to save it into a file with extension .ped (see the vignette). The genotypes were obtained from public data released by the *1000 Genomes Projects*.

Usage

```
data(ped)
```

Format

A table with 100 rows and 2007 columns.

Source

1000 Genomes, A Deep Catalogue of Human Genetic Variation; http://www.1000genomes.org

---

`waffect` *Simulation of phenotypes in the binary (case/control) and multiclass cases.*

Description

This is the main function of the **waffect** package. Given a vector (matrix) of probabilities and the desired total number of cases and controls (resp.: individuals in each class) `waffect` outputs a simulated phenotypic dataset.

Usage

```
waffect(prob, count, label, method, burnin)
```

Arguments

`prob`	a vector of probabilities corresponding to the disease model H1: the i-th entry is the probability that the i-th individual is a case. Alternatively, a matrix with k rows and n columns where K = number of classes and n = total number of individuals. In this case, the entry in the k-th row and j-th column is the probability that the phenotype of the j-th individual is in the k-th class. If `prob` is missing and `count` is a vector of length 2, then the constant vector of probabilities `rep(0.1, sum(count))` is assumed, thus resulting in simulating phenotypes under the null hypothesis H0. If `prob` is missing and `count` is a vector with length greater or equal than 3, then for each individual the probability to be in the first class is 0.1 and the probability to be in each of the other classes is 0.9/(K-1).
`count`	either an integer (the total number of cases), or a vector of length two (number of cases and number of controls), or, in the multiclass case, a vector of length greater or equal than 3 (number of individuals in each class).
`label`	a list with either the labels for cases and controls or, in the multiclass case, the codes for each class. In the binary case the first entry must be the label for cases. By default `label = c(1,0)` in the binary case and `label = 1:K`, where `K` is the total number of classes.
`method`	the method to be implemented for the simulation. Three methods are available: `"backward"`, `"mcmc"`, `"reject"`. The default method is `"backward"`; `"reject"` is deprecated.
`burnin`	the burn-in step if method is `"reject"`; by default `burnin = 1e+05 * n`, where `n` is the total number of individuals.

Value

A list of phenotypes coded by the entries in `label`.

See Also

Package documentation: waffect-package. Useful information can be also found in the vignette: vignette("waffect-tutorial").

Examples

```
## Not run: Typical usage to simulate case/control phenotypes under H1 (in this example: 12 individuals, 7 cases,
waffect(prob =  c(0.2, 0.4, 0.9, 0.6, 0.9, 0.1, 0.4, 0.6, 0.6, 0.3, 0.8, 0.1), count=c(7,5), label=c(1,0))

## Not run: By rerunning the command we obtain another simulation:
waffect(prob =  c(0.2, 0.4, 0.9, 0.6, 0.9, 0.1, 0.4, 0.6, 0.6, 0.3, 0.8, 0.1), count=c(7,5), label=c(1,0))

## Not run: We can just specify the number of cases:
waffect(prob =  c(0.2, 0.4, 0.9, 0.6, 0.9, 0.1, 0.4, 0.6, 0.6, 0.3, 0.8, 0.1), count=7, label=c(1,0))

## Not run: It is possible to change the default code for cases and controls:
pi <- runif(100)
waffect(prob = pi, c(50,50), label = c("case","control"))

## Not run: If prob is not specified then a constant vector of probabilities is assumed by default. This is equiva
waffect(count = c(20,30), label=c(1,0))

## Not run: Example with 6 individuals and 3 classes:
pi1 = c(0.3,0.4,0.3)
pi2 = c(0.3,0.5,0.2)
pi3 = c(0.1,0.2,0.7)
pi4 = c(0.1,0.6,0.3)
pi5 = c(0.1,0.7,0.2)
pi6 = c(0.4,0.1,0.5)
pi = cbind(pi1,pi2,pi3,pi4,pi5,pi6)
waffect(prob = pi, count = c(1,2,3))
```

---

waffectbin *Low level function for simulating binary phenotypes.*

Description

Usage of waffectbin is deprecated, please use waffect.

See Also

Documentation for waffect and waffect-package. Useful information can be also found in the vignette: vignette("waffect-tutorial").

Index