The GenABEL Package

September 4, 2007

Type  Package

Title  genome-wide SNP association analysis

Version  1.2-8

Date  2007-08-20

Author  Yurii Aulchenko, with contributions from Stephan Ripke and Toby Johnson

Maintainer  Yurii Aulchenko <i.aoultchenko@erasmusmc.nl>

Depends  R (>= 2.4.0), methods, genetics, haplo.stats, qvalue, MASS

Description  a package for genome-wide association analysis between quantitative or binary traits and SNPs

License  GNU GPL v. 2.0 or later

R topics documented:

GenABEL-package ................................................................. 3
HWE.show ................................................................. 5
Xfix ................................................................. 6
add.plot ................................................................. 7
as.character.gwaa.data ........................................................ 8
as.character.snp.coding ......................................................... 9
as.character.snp.data ......................................................... 10
as.character.snp.strand ......................................................... 11
as.data.frame.gwaa.data ......................................................... 12
as.double.gwaa.data ............................................................ 13
as.double.snp.data ............................................................. 14
as.genotype ................................................................. 15
as.genotype.gwaa.data .......................................................... 16
as.genotype.snp.data .......................................................... 17
as.hsgeno ................................................................. 18
as.hsgeno.gwaa.data ............................................................ 19
as.hsgeno.snp.data ............................................................ 20
topics documented:
catable .......................................................... 21
ccfast ............................................................. 22
check.marker-class ........................................... 23
check.marker .................................................... 24
check.trait ....................................................... 27
convert.snp.illumina ........................................... 28
convert.snp.mach ............................................... 29
convert.snp_ped ................................................. 30
convert.snp.text ............................................... 32
convert.snp.tped ............................................... 34
crnames ........................................................ 35
descriptives.marker .......................................... 36
descriptives.scan ............................................. 37
descriptives.trait .............................................. 38
dprfast ............................................................ 39
egscore ........................................................... 40
emp.ccfast ....................................................... 42
emp.qtscore ..................................................... 43
estlambda ......................................................... 45
export.merlin ................................................. 46
ge03d2 ............................................................ 47
grammar .......................................................... 48
gwaa.data-class ................................................ 50
hom ............................................................... 51
ibs ................................................................. 52
load.gwaa.data .................................................. 54
mmscore .......................................................... 55
perid.summary .................................................. 56
plot.check.marker ............................................. 57
plot.scan.gwaa.2D ............................................ 58
plot.scan.gwaa .................................................. 59
polygenic .......................................................... 60
qtscore ............................................................ 63
qvaluebh95 ...................................................... 64
0.1 Names ......................................................... 65
r2fast ............................................................. 66
redundant ........................................................ 67
refresh.gwaa.data ............................................. 68
rhofast ............................................................ 69
save.gwaa.data .................................................. 70
scan.glm.2D ..................................................... 71
scan.glm .......................................................... 72
scan.gwaa ........................................................ 73
scan.gwaa.2D-class ........................................... 74
scan.haplo.2D ................................................... 76
scan.haplo ........................................................ 77
show.ncbi ........................................................ 79
snp.coding-class ............................................... 80
Description

Genome-wide association (GWA) analysis is a tool of choice for identification of genes for complex traits. Effective storage, handling and analysis of GWA data represent a challenge to modern computational genetics. GWA studies generate large amount of data: hundreds of thousands of single nucleotide polymorphisms (SNPs) are genotyped in hundreds or thousands of patients and controls. Data on each SNP undergoes several types of analysis: characterization of frequency distribution, testing of Hardy-Weinberg equilibrium, analysis of association between single SNPs and haplotypes and different traits, and so on. Because SNP genotypes in dense marker sets are correlated, significance testing in GWA analysis is preferably performed using computationally intensive permutation test procedures, further increasing the computational burden.

To make GWA analysis possible on standard desktop computers we developed GenABEL library which addresses the following objectives:

(1) Minimisation of the amount of rapid access memory (RAM) used and the time required for data transactions. For this, we developed an effective data storage and manipulation model.

(2) Maximisation of the throughput of GWA analysis. For this, we designed optimal fast procedures for specific genetic tests.

Imbedding GenABEL into R environment allows for easy data characterisation, exploration and presentation of the results and gives access to a wide range of standard and special statistical analysis functions available in base R and specific R packages, such as "haplo.stats", "genetics", etc.

Details

<table>
<thead>
<tr>
<th>Package</th>
<th>GenABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>1.2-8</td>
</tr>
<tr>
<td>Date</td>
<td>2007-08-20</td>
</tr>
<tr>
<td>License</td>
<td>GNU GPL v. 2.0 or later</td>
</tr>
</tbody>
</table>
To see (more or less complete) functionality of GenABEL, try running demo(ge03d2).

Other demo of interest could be run with demo(srdta). Depending on your user privileges in Windows, it may well not run. In this case, try demo(srdtawin).

The most important functions and classes are:

For converting data from other formats, see convert.snp.text (conversion from human-readable GenABEL format), convert.snp.ped (Linkage, Merlin, Mach, and similar files), convert.snp.mach (Mach-format), convert.snp.tped (from PLINK TPED format), convert.snp.illumina (Illumina/Affymetrix-like format).

To load the data, see load.gwaa.data.

Also check companion programs, affy2mega.pl, affy2gwaa.pl, and WTCCC2GWAA.

For data management and manipulations see gwaa.data-class, snp.data-class, snp.names, snp.subset.

For quality control, see check.trait, check.marker, HWE.show, summary.snp.data, perid.summary, ibs.hom.

For fast analysis function, see scan.gwaa-class, ccfast, qtscore, mmscore, egscore, ibs, r2fast, dprfast, rhofast.

For specific tools facilitating analysis of the data with stratification (population stratification or (possibly unknown) pedigree structure), see qtscore (implements basic Genomic Control), ibs (computations of IBS / genomic IBD), egscore (stratification adjustment following Price et al.), polygenic (heritability analysis), mmscore (score test of Chen and Abecasis), grammar (grammar test of Aulchenko et al.).

For functions facilitating construction of tables for your manuscript, see descriptives.marker, descriptives.trait, descriptives.scan.

For link to WEB databases, see show.ncbi.

For interfaces to other packages and standard R functions, also for 2D scans, see scan.glm, scan.glm.2D, scan.haplo, scan.haplo.2D, scan.gwaa-class, scan.gwaa.2D-class.

For graphical facilities, see plot.scan.gwaa, plot.check.marker.

Author(s)

Yurii Aulchenko

Maintainer: Yurii Aulchenko <i.aoultchenko@erasmusmc.nl>

References

If you use the package in your analysis, please cite the following work:


If you used polygenic residuals from "polygenic" for qtscore, used GRAMMAR and/or GRAM-MAS analysis, please cite

If you used mmscore, please cite


For exact HWE, please cite:


For haplo.stats (scan.haplo, scan.haplo.2D), please cite:


For fast LD computations (function dprfast, r2fast), please cite:


See Also

Packages genetics, haplo.stats, qvalue,

Examples

# to see more or less complete functionality, run
#   demo(ge03d2)
# also try
#   demo(srdta)
# it will take a while!
# if demo(srdta) does not work (for Windows) try
#   demo(srdtawin)
#

HWE.show(data, idsubset = c(1:data@gtdata@nids),
          snpssubset = c(1:data@gtdata@nsnps))
Arguments

- **data**: object of class `gwaa.data-class` or `snp.data-class`
- **snpssubset**: Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from `data` are used for analysis.
- **idssubset**: Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from `data/cc` are used for analysis.

Value

- Only screen output

Author(s)

Yurii Aulchenko

See Also

- `check.marker`

Examples

```r
data(srdta)
mc <- check.marker(srdta,p.lev=0.01,ibs.mrk=0)
mc$nohwe
HWE.show(data=srdta,snps=mc$nohwe)
```

---

**Xfix**  \hspace{1cm} *function to set heterozygous male X-linked genotypes to NA*

Description

Sets heterozygous male X-linked genotypes to NA

Usage

```
Xfix(data)
```

Arguments

- **data**: Object of `gwaa.data-class`

Details

Value

- The same object of `gwaa.data-class`, with fixed genotypes
add.plot

function to plot additional GWAA results

Description

Add plot of results of GWA analysis

Usage

add.plot(x, ..., df = 1)

Arguments

x  object of type scan.gwaa-class, as returned by scan.glm, qtscore, ccfast, emp.ccfast, emp.qtscore, or scan.haplo; or of type scan.gwaa.2D-class, as returned by scan.haplo.2D or scan.glm.2D.

...  additional arguments to be passed to plot

df  P-value at which df to add (1, 2 or "Pc1df")

Note

Author(s)

Yurii Aulchenko

References

See Also

check.marker

Examples

data(ge03d2c)
# many errors
mc0 <- check.marker(ge03d2c)
# take only people and markers passing QC
fixed0 <- ge03d2c[mc0$idok,mc0$snpok]
# major errors fixed, still few males are heterozygous for X-chromosome markers
mc1 <- check.marker(fixed0)
# fix minor X-chromosome problems
fixed1 <- Xfix(fixed0)
# no errors
mc2 <- check.marker(fixed1)
summary(mc2)
as.character.gwaa.data

Value
No value returned.

Author(s)
Yuri Aulchenko

See Also
plot, snp.subset, scan.glm, qtscore, ccfast, emp.qtscore, emp.ccfast, scan.haplo, scan.haplo.2D, scan glm.2D

Examples

data(srdta)
a <- ccfast("bt", srdta, snps=c(1:100))
plot(a)
a1 <- qtscore(bt, srdta, snps=c(1:100))
add.plot(a1, col="red", type="l")

Description
A function to convert @gtdata slot of an object of gwaa.data-class to "character"

Usage
as.character.gwaa.data(x, ...)

Arguments
x An object of gwaa.data-class
...

Details

Value
A matrix containing genotypes in character format

Note
as.character.snp.coding

Author(s)

Yurii Aulchenko

References

See Also

as.character.snp.data, as.double.gwaa.data, as.double.snp.data, as.hsgeno, as.genotype.gwaa.data, as.genotype.snp.data

Examples

data(srdta)
as.character(srdta[1:5,1:10])

Description

A function to convert an object of \texttt{snp.coding-class} to "character"

Usage

\texttt{as.character.snp.coding(x, ...)}

Arguments

\begin{itemize}
  \item \texttt{x} \hspace{1cm} An object of \texttt{snp.coding-class}
  \item \texttt{...} \hspace{1cm} ...
\end{itemize}

Details

Value

A vector containing actual (nucleotide) coding, for corresponding SNPs, in character format

Note
as.character.snp.data

Description

A function to convert an object of \texttt{snp.data-class} to "character"

Usage

\begin{verbatim}
as.character.snp.data(x, ...)
\end{verbatim}

Arguments

\begin{itemize}
  \item \texttt{x} \hspace{1cm} An object of \texttt{snp.data-class}
  \item \texttt{...} \hspace{1cm} ...
\end{itemize}

Details

Value

A matrix containing genotypes in character format

Note

Examples

\begin{verbatim}
data(srdta)
as.character(srdta@gtdata@coding[1:5])
\end{verbatim}
as.character.snp.strand

Author(s)
Yurii Aulchenko

References

See Also
as.double.snp.data, as.hsgeno, as.genotype.snp.data

Examples

```r
data(srdta)
as.character(srdta@gtdata[1:5,1:10])
```

Description
A function to convert an object of \texttt{snp.strand-class} to "character"

Usage

```r
as.character.snp.strand(x, ...)
```

Arguments

<table>
<thead>
<tr>
<th>x</th>
<th>An object of \texttt{snp.strand-class}</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Details

Value
A vector containing strand ("+", "-" or "u"), for corresponding SNPs, in character format

Note

Author(s)
Yurii Aulchenko
as.data.frame.gwaa.data

References

See Also

as.character.snp.coding, as.character.snp.data, as.double.snp.data, as.hsgeno, as.genotype.snp.data

Examples

data(srdta)
as.character(srdta@gtdata@strand[1:5])

as.data.frame.gwaa.data

Attempts to convert snp.data to "hsgeno"

Description

A function taking @phdata part (data.frame) of the object of gwaa.data-class

Usage

as.data.frame.gwaa.data(x, ...)

Arguments

x An object of data.frame-class

Details

Use is mainly internal

Value

A data-frame containing phenotypic data

Note

Author(s)

Yuri Aulchenko
as.double.gwaa.data

References

See Also

as.character.snp.data, as.double.snp.data, as.genotype.snp.data

Examples

data(srdta)
as.hsgeno(srdta[1:5,1:10])

---

as.double.gwaa.data

Attempts to convert gwaa.data to double

---

Description

A function to convert an object of gwaa.data-class to "double"

Usage

as.double.gwaa.data(x, ...)

Arguments

x An object of gwaa.data-class

... ...

Details

Value

A matrix containing genotypes in double (numeric) format

Note

Author(s)

Yurii Aulchenko

References
See Also

as.character.gwaa.data, as.character.snp.data, as.double.gwaa.data, as.double.snp.data,
as.hsgeno, as.genotype.gwaa.data, as.genotype.snp.data

Examples

data(srdta)
as.double(srdta[1,5,1:10])

---

as.double.snp.data Attempts to convert snp.data to double

Description

A function to convert an object of snp.data-class to "double"

Usage

as.double.snp.data(x, ...)

Arguments

x An object of snp.data-class
... ...

Details

Value

A matrix containing genotypes in double (numeric) format

Note

Author(s)

Yurii Aulchenko

References

See Also

as.character.snp.data, as.hsgeno, as.genotype.snp.data
as.genotype

Examples

```r
data(srdta)
as.double(srdta@gtdata[1:5,1:10])
```

---

**as.genotype**  
Attempts to convert object to "genotype"

Description

A function to convert an object to "genotype" data frame

Usage

```r
as.genotype(x, ...)
```

Arguments

- `x`  
  An object of `snp.data-class`  
- `...`  
  ... 

Details

Value

A data-frame containing "genotype" data class, consumable by "genetics" library

Note

Author(s)

Yurii Aulchenko

References

See Also

- `as.character.gwaa.data`, `as.character.snp.data`, `as.double.gwaa.data`, `as.double.snp.data`, `as.hsgeno`, `as.genotype.gwaa.data`, `as.genotype.snp.data`

Examples

```r
data(srdta)
as.genotype(srdta@gtdata[1:5,1:10])
```
as.genotype.gwaa.data

Attempts to convert gwaa.data to "genotype"

Description
A function to convert @gtdata slot of an object of \texttt{gwaa.data-class} to "genotype" data frame

Usage
\begin{verbatim}
as.genotype.gwaa.data(x, ...)\end{verbatim}

Arguments
\begin{itemize}
\item \textbf{x} \texttt{An object of gwaa.data-class}
\item \textbf{...} \texttt{...}
\end{itemize}

Details

Value
A data-frame containing genotypes consumable by "genetics" library

Note

Author(s)
Yurii Aulchenko

References

See Also
\begin{verbatim}
as.character.gwaa.data,as.character.snp.data,as.double.gwaa.data,as.double.snp.data,as.hsgeno,as.genotype.gwaa.data,as.genotype.snp.data\end{verbatim}

Examples
\begin{verbatim}
data(srdta)
as.genotype(srdta[1:5,1:10])\end{verbatim}
as.genotype.snp.data

Attempts to convert snp.data to "genotype"

Description

A function to convert an object of snp.data-class to "genotype" data frame

Usage

as.genotype.snp.data(x, ...)

Arguments

x

An object of snp.data-class

Details

Value

A data-frame containing genotypes consumable by "genetics" library

Note

Author(s)

Yurii Aulchenko

References

See Also

as.character.snp.data, as.double.snp.data, as.hsgeno

Examples

data(srdta)
as.genotype(srdta@gtdata[1:5,1:10])
Description

A function to convert an object to "hsgeno" data frame

Usage

\[
\text{as.hsgeno}(x, \ldots)
\]

Arguments

- \textit{x} An object of \texttt{snp.data-class}
- \textit{\ldots} ...

Details

Value

A data-frame containing alleles, consumable by "haplo.stats" library

Note

Author(s)

Yurii Aulchenko

References

See Also

\texttt{as.character.snp.data, as.double.snp.data, as.genotype.snp.data}

Examples

data(srdta)
as.hsgeno(srdta@gtdata[1:5,1:10])
as.hsgeno.gwaa.data

Attempts to convert gwaa.data to "hsgeno"

Description

A function to convert @gtdata slot of an object of gwaa.data-class to "hsgeno" data frame

Usage

as.hsgeno.gwaa.data(x, ...)

Arguments

x
An object of gwaa.data-class

...

Details

Value

A data-frame containing alleles, consumable by "haplo.stats" library

Note

Author(s)

Yuriy Aulchenko

References

See Also

as.character.gwaa.data, as.character.snp.data, as.double.gwaa.data, as.double.snp.data, as.hsgeno, as.genotype.gwaa.data, as.genotype.snp.data

Examples

data(srdta)
as.hsgeno(srdta[1:5,1:10])
as.hsgeno.snp.data  Attempts to convert snp.data to "hsgeno"

Description
A function to convert an object of `snp.data-class` to "hsgeno" data frame

Usage
```r
as.hsgeno.snp.data(x, ...)
```

Arguments
- `x` An object of `snp.data-class`
- `...` ...

Details

Value
A data-frame containing alleles, consumable by "haplo.stats" library

Note

Author(s)
Yurii Aulchenko

References

See Also
- `as.character.snp.data`, `as.double.snp.data`, `as.genotype.snp.data`

Examples
- `data(srdta)`
- `as.hsgeno(srdta@gtdata[1:5,1:10])`
catable

function to generate summary table for quantitative data

description

This function makes a table with number of observations which fall between user-defined categories

usage

catable(data, categories = c(quantile(data,c(0.01,0.1,0.5,0.9,0.99),na.rm=TRUE)), cumulative = FALSE, na.rm = TRUE, digits = 3)

arguments

data

A vector of numerics
categories

vector containing desired cut-off levels
cumulative

whether cumulative distribution should be shown
na.rm

how to treat NAs
digits

number of digits to be saved in rounding

details

value

table with number and proportion of observations falling between categories

note

author(s)

Yurii Aulchenko

references

see also

summary.snp.data, perid.summary
Examples

data(srdta)
callr <- summary(srdta@gtdata)[,"CallRate"]
catable(callr,c(0.93, 0.95, 0.99))
catable(callr)
catable(callr,cum=TRUE)

ccfast

---

fast case-control analysis

---

Description

Fast case-control analysis by computing chi-square test from 2x2 (allelic) or 2x3 (genotypic) tables

Usage

ccfast(y, data, snpsubset, idsubset, times=1, quiet=FALSE, bcast=10, clambda=TRUE, propPs=1.0)

Arguments

y character name of the vector of case-control status. Cases are denoted as 1 and controls as 0.
data An object of gwaa.data-class
snpsubset Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from data are used for analysis.
idsubset Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from data are used for analysis.
times If more then one, the number of replicas to be used in derivation of empirical genome-wide significance. See emp.qtscore, which calls qtscore with times>1 for details
quiet do not print warning messages
bcast If the argument times > 1, progress is reported once in bcast replicas
clambda If inflation facot Lambda is estimated as lower then one, this parameter controls if the original P1df (clambda=TRUE) to be reported in Pc1df, or the original 1df statistics is to be multiplied onto this "deflation" factor (clambda=FALSE). If a numeric value is provided, it is used as a correction factor.
propPs proportion of non-corrected P-values used to estimate the inflation factor Lambda, passed directly to the estlambda

Value

Object of class scan.gwaa-class

Author(s)

Yuri Aulchenko
See Also

emp.ccfast, plot.scan.gwaa, scan.gwaa-class

Examples

data(srdta)
a <- ccfast("bt",data=srdta,snps=c(1:10),ids=c(1:100))
a
a <- ccfast("bt",data=srdta)
plot(a)

check.marker-class  Class "check.marker"

Description

This class contains results of genotypic quality control. This is an list object, usually generated by check.marker.

Names

snpok  Markers which passed all criteria
idok  People which passed all criteria
nohwe  Markers which did not pass HWE check
Pex.nohwe  Exact HWE P-values for markers which did not pass HWE check
nocall  Markers with call rate < specified callrate
nofreq  Markers with MAF < specified maf
Xmrkfail  X-linked markers with too many heterozygous male genotypes
redundant  Redundant markers
details.redundancy  List with details on redundant markers (reference-marker <-> redundant-markers)
idnocall  People with too low SNP call rate across al SNPs
hetfail  People having too high heterozygosity
ibsfail  People having too high IBS with other people
Xidfail  Men with too many heterozygous X-linked markers
call  List with details on call: call, name (of marker), map, chromosome
Methods

**summary** signature(object = "check.marker"): gives a cross table summarising how many markers did not pass because of this or that criteria

**plot** signature(object = "check.marker"): Plots summary of genotypic data QC

Author(s)

Yurii Aulchenko

See Also

check.marker, summary.check.marker, redundant.plot.check.marker

Examples

data(srdta)
mc <- check.marker(data=srdta@gtdata[,1:100], redundant="all", maf=0.01, minconcordance=0.9, fdr=.1, ibs.mrk=0)
class(mc)
names(mc)
names(mc$call)
mc$nohwe
mc$Pex.nohwe
summary(mc)
plot(mc)

---

check.marker function to do genotypic quality control

Description

This function helps selecting the marker which should enter into GWA analysis based on call rate, minor allele frequency, value of the chi-square test for Hardy-Weinberg equilibrium, and redundancy, defined as concordance between the distributions of the genotypes (including missing values).

Usage

check.marker(data, snpsubset, idsubset, callrate = 0.95, perid.call=0.95, extr.call = 0.1, extr.perid.call = 0.1, het.fdr = 0.01, ibs.threshold = 0.95, ibs.mrk = 2000, ibs.exclude="lower", maf, p.level = -1, fdrate = 0.2, odds = 1000, hweidsubset, redundant = "no", minconcordance = 2.0, qoption = "bh95")
Arguments

data  gwaa.data or snp.data object
snps subset a subset of SNPs to check (names, indexes, logical), default is all from data
idsubset a subset of people to check (names, indexes, logical), default is all from data
callrate cut-off SNP call rate
perid.call cut-off individual call rate (maximum percent of missing genotypes in a person)
extr.call SNPs with this low call rate are dropped prior to main analysis
extr.perid.call people with this low call rate are dropped prior to main analysis
het.fdr FDR rate for unacceptably high individual heterozygosity
ibs.threshold threshold value for acceptable IBS
ibs.mrk How many random markers should be used to estimate IBS. When ibs.mrk < 1, IBS checks are turned off. When "all" all markers are used.
ibs.exclude "both" or "lower" – whether both samples with IBS>ibs.threshold should be excluded, or the one with lower call rate.
maf cut-off Minor Allele Frequency. If not specified, the default value is 5 chromosomes (5/data@nsnpa)
p.level cut-off p-value in check for Hardy-Weinberg Equilibrium. If negative, FDR is applied
fdrate cut-off FDR level in check for Hardy-Weinberg Equilibrium
odds cut-off odds to decide whether marker/person should be excluded based on sex/X-linked marker data inconsistency
hweidsubset a subset of people to check (names, indexes, logical) to use for HWE check
redundant if "bychrom", redundancy is checked within chromosomes; "all" – all pairs of markers; "no" – no redundancy checks
minconcordance a parameter passed to "redundant" function. If "minconcordance" is > 1.0 only pairs of markers which are exactly the same, including NA pattern, are considered as redundant; if 0 < "minconcordance" < 1, then pairs of markers with concordance > "minconcordance" are considered redundant. See redundant for details. Note that if "minconcordance" < 1 the program will take much longer time to run
qoption if "bh95", BH95 FDR used; if "storey", qvalue package is used

Details

In this procedure, sex errors are identified initially and then possible residual errors are removed iteratively. At the first step, of the iterative procedure, per-marker (minor allele frequency, call rate, exact P-value for Hardy-Weinberg equilibrium) and between-marker statistics are generated and controlled for, mostly using the internal call to the function summary.snp.data.

At the second step of the iterative procedure, per-person statistics, such call rate within a person, heterozygosity and and between-person statistics (identity by state across a random sample of markers)
are generated, using `perid.summary` and `ibs` functions. Heterozygosity and IBS are estimated using only autosomal data. If IBS is over `ibs.threshold` for a pair, one person from the pair is added to the `ibsfail` list and excluded from the `idok` list. At the second step, only the markers passing the first step are used.

The procedure is applied recursively till no further markers and people are eliminated.

**Value**

Object of class `check.marker-class`

**Author(s)**

Yurii Aulchenko

**See Also**

`check.trait`, `ibs`, `summary.snp.data`, `perid.summary`, `plot.check.marker`, `summary.check.marker`, `redundant`, `HWE.show`, `check.marker-class`

**Examples**

```r
# usual way
data(ge03d2c)
# many errors
mc0 <- check.marker(ge03d2c)
# take only people and markers passing QC
fixed0 <- ge03d2c[mc0$idok,mc0$snpok]
# major errors fixed, still few males are heterozygous for X-chromosome markers
mc1 <- check.marker(fixed0)
# fix minor X-chromosome problems
fixed1 <- Xfix(fixed0)
# no errors
mc2 <- check.marker(fixed1)
summary(mc2)
# ready to use fixed1 for analysis

# let us look into redundancy
data(srdta)
mc <- check.marker(data=srdta,ids=c(1:300),call=.92,perid.call=.92)
names(mc)
mc$nohwe
mc <- check.marker(data=srdta@gtdata[,1:100],call=0.95,perid.call=0.9,maf=0.02,minconcordance=0.9,fdr=0.1,redundant="all",ibs.mrk=0)
summary(mc)
HWE.show(data=srdta,snps=mc$nohwe)
plot(mc)
```
check.trait: function to do primitive trait quality control

Description

This function checks for outliers (using FDR framework) and plots the raw data.

Usage

check.trait(trait, data, fdrate = 0.05, graph = TRUE, binshow = FALSE, qoption = "bh95")

Arguments

- **trait**: name (or list of names) of trait(s) to be checked
- **data**: gwaa.data object or data frame containing the trait
- **fdrate**: false discovery rate to apply for QC
- **graph**: if graphical output should be produced
- **binshow**: if binary traits should be plotted
- **qoption**: how to compute q-values (not implemented, currently using only BH95)

Details

The P-value that a particular measurement is an outlier is computed as follows. Consider trait vector \( Y \) with particular \( i^{th} \) measurement denoted as \( y_i \). Let \( Y(-i) \) is vector, which is the same as \( Y \), except that \( i^{th} \) measurement is dropped. Then Chi-square for measurement \( i \) is computed as

\[
Chi_i = \frac{(\text{mean}(Y(-i)) - y_i)^2}{\text{var}(Y(-i))}
\]

P-value is computed using 1 d.f., and the vector of P-values enters FDR computation procedure (BH95 by default).

Value

No value returned, output is made to the screen and graphical device.

Author(s)

Yuri Aulchenko

See Also

check.marker
convert.snp.illumina

function to convert genotypic data from Illumina/Affymetrix to internal format

Description

Converts genotypic data from Illumina/Affymetrix-like format to internal genotypic data formatted file

Usage

convert.snp.illumina(infile, outfile, strand = "+", bcast = 1000000)

Arguments

infile Pre-makeped linkage genotypic data file name
outfile Output data file
strand Specification of strand, one of "u" (unknown), "+", "-" or "file". In the latter case, extra column specifying the strand (again, one of "u", "+", or "-") should be included on the infile.
bcast Reports progress after reading bcast portion of SNP genotypes

Details

Input file is the one which could be typically obtained from Illumina BeadStudio software. For example:

Name Chr Pos id1 id2 id3
rs1001 2 12897 AC AA AA
rs2401 3 12357 AG GG AG
rs123 3 5327 TC TT CC

Here, every row corresponds to a SNP, and each column, starting with the 4th, corresponds to a person.

When strand information is available (option strand="file"), the file should look like

Accepted allele codes: 1/2, A/B, A/T, A/G, A/C, T/G, T/C, G/C, A/-, T/-, G/-, C/-. Here, "-" stands of a deletion. Missing data can be coded as "-" or "00". Make sure that the coding for missing is "00" if you use one of the codings A/-, T/-, G/-, C-!

Name Chr Pos Strand id1 id2 id3
rs1001 2 12897 + AC AA AA
rs2401 3 12357 - AG GG AG
rs123 3 5327 + TC TT CC

Accepted strand coding: +, -, u (unknown)

Value
Does not return any value, but writes file with GenABEL raw data

Note
The function does not check if "outfile" already exists, thus it is always over-written

Author(s)
Yurii Aulchenko

See Also
load.gwaa.data, convert.snp.text, convert.snp.mach, convert.snp.tped

Examples
#
# convert.snp.illumina(infile="pedin.18", out="genos.raw", strand="+")
#

convert.snp.mach function to convert genotypic data from MACH format to internal data format

Description
Converts genotypic data from MACH format to internal genotypic data formatted file

Usage
convert.snp.mach(pedfile, mapfile, infofile, outfile, quality = 0.9, ...)

Arguments
pedfile File with genotypic data from MACH (geno or mlgeno)
mapfile Name of the map file
infofile Name MACH info-file
outfile Output data file
quality Drop the SNPs with quality (as specified in 6th column of info-file) lower than this threshold.
... Other arguments passed to convert.snp.ped
Details

This is a simple script converting the MACH data with `convert.snp.ped`, re-loading data, and filtering the `snp.data` object based on quality as specified in MACH info-file

Value

Does not return any value, but writes file with GenABEL raw data

Note

The function does not check if "outfile" already exists, thus it is always over-written

Author(s)

Yurii Aulchenko

See Also

`load.gwaa.data, convert.snp.illumina, convert.snp.text, convert.snp.ped, convert.snp.tped`

Examples

```r
# convert.snp.mach(ped="pedin.18",map="map.18",out="genos.raw")
#
```

```
convert.snp.ped function to convert genotypic data in pedigree fromat (+map) to internal data format
```

Description

Converts genotypic data in a variety of pedigree fromats (+map) to internal genotypic data formatted file

Usage

```r
convert.snp.ped(pedfile, mapfile, outfile, format = "premakeped", traits = 1, strand = "u", bcast = 10000000)
```
**Arguments**

- **pedfile**: Pre-makeped linkage genotypic data file name
- **mapfile**: Name of the map file
- **outfile**: Output data file
- **format**: Input data format, either "premakeped" (default, also works with Merlin files), or "mach"
- **traits**: How many traits are specified in the pedigree file (usually 1 – affection – or 2 – affection and liability). Has no effect when format = "mach".
- **strand**: Specification of strand, one of "u" (unknown), "+", "-" or "file". In the latter case, map-file should contain an extended map (the one including strand and coding)
- **bcast**: Reports progress after reading bcast portion of SNPs

**Details**

Pedfile must be standard pre-makeped/Merlin linkage file, or a Mach file. In pre-makeped linkage file, columns are:

`ped id fa mo sex snp1.allele1 snp1.allele2 snp2.allele1 snp2.allele2 ...`

For example:

```
1 1 0 0 1 2 A A G T ...
1 2 0 0 1 0 A G T T ...
1 3 0 0 2 1 A A T T ...
```

Would imply that persons 1, 2 and 3 are "founders" (which would be typical for a case-control study), 1 and 2 are males and 3 is female. Person 1 is homozygous for allele 1 at locus 1 and heterozygous at locus 2. Person 2 is heterozygous at both loci. Person 3 is homozygous for allele 2 at locus 1 and allele 1 at locus 2.

Only the second and the marker columns are used, thus make sure the IDs are unique!


The map file is standard Merlin map. For example:

```
chrom name position
18 rs679153 2859916
18 rs9965482 2860891
```

Says that locus 1 is named rs679153 and located at chromosome 18 position 2859916. Locus 2 (rs9965482) is located at chromosome 18, position 2860891.

In extended map format, there should be 4th column specifying the strand and 5th column specifying coding.

```
chrom name position strand coding
18 rs679153 2859916 - AG
18 rs9965482 2860891 + GT
```

Accepted strand coding: +, -, u (unknown)
convert.snp.text

Value

Does not return any value, but writes file with GenABEL raw data

Note

The function does not check if "outfile" already exists, thus it is always over-written

Author(s)

Yurii Aulchenko

See Also

load.gwaa.data, convert.snp.illumina, convert.snp.mach, convert.snp.text, convert.snp.tped

Examples

```
# convert.snp.ped(ped="pedin.18",map="map.18",out="genos.raw")
#
```

```
convert.snp.text  function to convert integer genotypic data file to raw internal data formatted file

Description

Converts integer genotypic data file to raw internal data formatted file

Usage

convert.snp.text(infile, outfile, bcast = 10000)

Arguments

(infile) Input data file name

(outfile) Output data file

(bcast) Reports progress after reading bcast portion of SNPs
Details

Input genotypic data file contains all kind of genetic information. The first line of this file contains IDs of all study subjects. The second line gives names of all SNPs in the study. The third line list the chromosomes the SNPs belong to. Sequential numbers are used for autosomes and "X" (capital!) is used for the sex-chromosome. The forth line lists genomic position of the SNPs, in order which is the same as order in the line 2. The genomic position can be chromosome-specific (each chromosome starts with "0") or, better, a true genomic position (chromosome 1 starts with 0 and chromosome 2 continues at the point chromosome 1 ends).

Starting with the line five, genetic data are presented. The 5th line contains the data for SNP, which is listed first on the second line. The first column of this line specifies the genotype for the person, who is listed first on the line 1; the second column gives the genotype for the second person, so on. The genotypes are coded as 0 (missing), 1 (for AA), 2 (for AB) and 3 (for BB). Here is a small example:

289982 325286 357273 872422 1005389
SNP-1886933 SNP-2264565 SNP-2305014
1 1 1
825852 2137143 2585920
3 3 3 2
3 2 3 3
2 2 1 1
In this example, we can see that SNP-2305014 (number 3 in the second line) is located on chromosome 1 at the position 2585920. If we would like to know what is genotype of person with ID 325286 (second in the first line), we need to take second column and the third line of the genotypic data. This cell contains 1, thus, person 325286 has genotype "AA" at SNP-2305014.

In the event that you do not want to use a map for some reason (such as prior ordering of the polymorphisms in the genotype file), make a dummy map-line, which contains order information.

The above described genotypic data file is (more or less) human-readable; actually, to achieve the aim of effective data storage GWAA package uses internal format. In this format, four genotypes are stored in single byte; "raw" data format of R is used.

Value

Does not return any value

Note

The function does not check if "outfile" already exists, thus it is always over-written

Author(s)

Yurii Aulchenko

See Also

load.gwaa.data, convert.snp.illumina, convert.snp.ped, convert.snp.mach, convert.snp.tped
Examples

```r
# convert.snp.text("genos.dat","genos.raw")
```

```
convert.snp.tped
```


function to convert genotypic data in transposed-ped format (.tped and .tfam) to internal genotypic data formatted file

**Description**

Converts genotypic data in transposed-ped format (.tped and .tfam) to internal genotypic data formatted file

**Usage**

```r
convert.snp.tped(tpedfile, tfamfile, outfile, bcast = 10000)
```

**Arguments**

- `tpedfile`: Name of transposed-ped format (.tped) file to read
- `tfamfile`: Name of individual data (.tfam) file to read
- `outfile`: Name for output data file
- `bcast`: Reports progress every time this number of SNPs have been read

**Details**

The transposed-ped file format may be preferred when extremely large numbers of markers have been genotyped. This file format is supported by plink! See http://pngu.mgh.harvard.edu/ purcell/plink/ for details.

The conversion is performed by C++ code that is both fast and memory efficient.

The genotype data are stored in the main transposed-ped format file, usually with a .tped file extension. If there are NSNP markers genotyped in NIND individuals, this file has NSNP rows and 4+NIND*2 columns. There is one row per marker, and no header. The first four columns are:

- **Chromosome**
- **Marker name (e.g. rs number)**
- **Genetic position (in Morgans)**
- **Physical position (in bp)**

These are followed by two columns per individual, which contain the genotype, coded as two characters. The ‘0’ character is used for missing data. For example, a file containing data for six individuals genotyped at two SNPs would look like:

1 rs1234 0 5000650 A A 0 0 C A C C C C C
1 rs5678 0 5000830 G T G T G T G T T T
In this example, the second individual is missing data for SNP rs1234, etc. The alleles can be coded by any two distinct characters, e.g. 'C' and 'G', or '1' and '2'. The '0' character is reserved for missing data, and each individual genotype must be either complete, or completely missing. In the current implementation, only the physical positions of the SNPs are read, and the genetic positions are ignored.

The indices for the columns are stored in a separate file, usually with a .tfam file extension. Traditionally, this file has six columns, and no header. In the current implementation, only the second column is used. This column must contain the individual id. Other columns are ignored.

Value

Does not return any value

Note

The function does not check if "outfile" already exists, thus it is always over-written

Author(s)

Toby Johnson <toby.johnson@unil.ch>

See Also

convert.snp.ped, convert.snp.illumina, convert.snp.text, convert.snp.mach, load.gwaa.data

Examples

```r
# convert.snp.tped("c21.tped",map="c21.tfam",out="c21.raw")
```

---

**crnames**

Return column and row names

Description

Given a dimnames, returns column and row names for index cells

Usage

```r
crnames(dnames,idx)
```

Arguments

- **dnames**: object *dimnames*
- **idx**: index (or logical condition on the original object)
**Details**

**Source**

**References**

**Examples**

```r
data(ge03d2ex)
a <- as.numeric(ge03d2ex[1:20,1:3])
crnames(dimnames(a),a==1)
```

---

**descriptives.marker**

*Function to generate descriptive summary tables for genotypic data*

**Description**

Function to generate descriptive summary tables for genotypic data

**Usage**

```r
descriptives.marker(data, snpsubset, idsubset, file, mafc, hwec, snpc, idcc, digits = 3)
```

**Arguments**

- `data` an object of `snp.data-class` or `gwaa.data-class`
- `snpsubset` Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from `data` are used for analysis.
- `idsubset` Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from `data` are used for analysis.
- `file` A string specifying the name of a file to write the tables to (default is missing).
- `mafc` vector containing desired cut-off levels for minor allele frequency
- `hwec` vector containing desired cut-off levels for exact HWE P-values
- `snpc` vector containing desired cut-off levels for SNP call rate
- `idcc` vector containing desired cut-off levels for individual SNP call rate
- `digits` number of digits to be printed

**Details**
Function to describe "top" hits in GWA scan

descriptives.scan  Function to describe "top" hits in GWA scan

Description
   Describes "top" hits in GWA scan

Usage
   descriptives.scan(data, file, top=10, sortby="P1df", digits = 6)

Arguments
   data  an object of snp.data-class or gwaa.data-class
   file  A string specifying the name of a file to write the tables to (default is no file output).
   top   How many "top" hits to describe
   sortby How to pick up "top" hits ("P1df","P2df","Pgw1df","Pgw2df")
   digits number of digits to be printed

Details
Function to generate descriptive summary tables for phenotypic data

Usage

```r
descriptives.trait(data, subset, file, by.var=NULL, digits = 3)
```

Arguments

- **data**: an object of `snp.data-class` or `gwaa.data-class`
- **subset**: Subset of people to run analysis on. If missing, all people from `data` are used for analysis.
- **file**: A string specifying the name of a file to write the tables to (default is no file output).
- **by.var**: a binary trait; statistics will be produced separately for the groups and compared
- **digits**: number of digits to be printed

Description

Function to generate descriptive summary tables for phenotypic data
**dprfast**

**Details**

**Value**

A table with descriptive statistics. Ptt: t-test; Pkw: kruskal.test; Pex: Fisher exact test (for factors with <5 levels)

**Note**

**Author(s)**

Yurii Aulchenko

**References**

**See Also**

**Examples**

```r
data(srdta)
descriptives.trait(srdta)
descriptives.trait(srdta, by.var=srdta@phdata$sex)
```

---

**dprfast**

*Estimates D’ between multiple markers*

**Description**

Given a set of SNPs, computes a matrix of D’

**Usage**

```r
dprfast(data, snpsubset, idsubset)
```

**Arguments**

- `data` : object of `snp.data-class`
- `snpsubset` : Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from data are used for analysis.
- `idsubset` : Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from data are used for analysis.
Details

The function is based on slightly modified code of Hao et al.

Value

A (Nsnps X Nsnps) matrix giving D’ values between a pairs of SNPs above the diagonal and number of SNP genotype measured for both SNPs below the diagonal

Author(s)

Yuri Aulchenko

References


See Also

rhofast

Examples

```r
data(ge03d2)
# D's using D'fast
a <- dprfast(ge03d2,snps=c(1:10))
# D's using package genetics
b <- LD(as.genotype(ge03d2[,1:10]))$"D'"
# see that the D's are not exactly the same
cor(a[upper.tri(a)],b[upper.tri(b)])
plot(a[upper.tri(a)],b[upper.tri(b)])
```

---

table

| egscore | Fast score test for association, corrected with PC |

Description

Fast score test for association between a trait and genetic polymorphism, adjusted for possible stratification by principal components.

Usage

```r
egscore(formula,data,snpsubset,idsubset,kinship.matrix,naxes=3,strata,times=1,quiet
```
egscore

Arguments

**formula**
Formula describing fixed effects to be used in analysis, e.g. \( y = a + b \) means that outcome (y) depends on two covariates, a and b. If no covariates used in analysis, skip the right-hand side of the equation.

**data**
An object of `{gwaa.data-class}`

**snpsubset**
Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from `data` are used for analysis.

**idsubset**
Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from `data/cc` are used for analysis.

**kinship.matrix**
Kinship matrix, as returned by `{ibs}` (use weight="freq"!)

**naxes**
Number of axes of variation to be used in adjustment (should be much smaller than number of subjects)

**strata**
Stratification variable. If provided, scores are computed within strata and then added up.

**times**
If more than one, the number of replicas to be used in derivation of empirical genome-wide significance.

**quiet**
do not print warning messages

**bcast**
If the argument times > 1, progress is reported once in bcast replicas

**clambda**
If inflation factor Lambda is estimated as lower than one, this parameter controls if the original P1df (clambda=TRUE) to be reported in Pc1df, or the original 1df statistics is to be multiplied onto this "deflation" factor (clambda=FALSE). If a numeric value is provided, it is used as a correction factor.

**propPs**
Proportion of non-corrected P-values used to estimate the inflation factor Lambda, passed directly to the `{estlambda}`

Details

The idea of this test is to use genomic kinship matrix to first, derive axes of genetic variation (principal components), and, second, adjust both trait and genotypes onto these axes.

The traits is first analysed using LM and with covariates as specified with formula and also with axes of variation as predictors. Corrected genotypes are defined as residuals from regression of genotypes onto axes (which are orthogonal). Correlation between corrected genotypes and phenotype is computed, and test statistics is defined as square of this correlation times \((N - K - 1)\), where \(N\) is number of genotyped subjects and \(K\) is the number of axes.

This test is defined only for 1 d.f.

Value

Object of class `{scan.gwaa-class}`

Author(s)

Yuriy Aulchenko
emp.ccfast

References


See Also

qtscore, mmscore, plot.scan.gwaa, scan.gwaa-class

Examples

data(ge03d2ex)
#egscore with stratification
a <- egscore(dm2~sex+age, data=ge03d2ex, kin=ibs(ge03d2ex, w="freq"))
plot(a)

emp.ccfast

Genome-wide significance for a case-control GWA scan

Description

Genome-wide significance for a case-control GWA scan. Analysis function is ccfast.

Usage

data(ge03d2ex)
#egscore with stratification
a <- egscore(dm2~sex+age, data=ge03d2ex, kin=ibs(ge03d2ex, w="freq"))
plot(a)

emp.ccfast (y, data, snpsubset, idsubset, times = 100, quiet=FALSE, bcast = 10)

Arguments

All arguments are the same as in and passed intact to the ccfast. See help for this function.

y character name of the vector of case-control status. Cases are denoted as 1 and controls as 0.
data An object of gwaa.data-class
snpsubset Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from data are used for analysis.
idsubset Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from data are used for analysis.
times If more then one, the number of replicas to be used in derivation of empirical genome-wide significance. See emp.qtscore, which calls qtscore with times>1 for details
quiet do not print warning messages
bcast If the argument times > 1, progress is reported once in bcast replicas
emp.qtscore

Details

In the analysis of empirical significance, first time the function \texttt{ccfast} is called and result object is saved. Later, the function \texttt{ccfast} is called \texttt{times} times with \texttt{replace=FALSE} in order to generate the distribution under the null. Each call, minimal P-value is extracted and compared with original P-values. For a particular SNP, empirical P-value is obtained as a proportion of times minimal Ps from resampled data was less then the original P.

The list elements effB, effAB and effBB are the ones obtained from the analysis of the original (not permuted) data set.

Value

Object of class \texttt{scan.gwaa-class}

Note

Author(s)

Yurii Aulchenko

See Also

\texttt{ccfast.emp.qtscore,scan.gwaa-class}

Examples

\begin{r}
data(srdta)
a<-ccfast("bt",data=srdta,snps=c(500:800))
plot(a)
# this does not make sense, as the whole experiment must be analysed, not a small region!
b<-emp.ccfast("bt",data=srdta,snps=c(500:800),bcast=10)
plot(b)
# compare qvalues and empirical P
qv<-qvaluebh95(a$P1df)$qval
qv
b$P1df
plot(qv,b$P1df,xlim=c(0,1),ylim=c(0,1))
abline(a=0,b=1)
\end{r}
Usage

```r
data emp.qtscore(formula, data, snpsubset, idsubset, strata, trait.type="gaussian", times = 100, quiet=FALSE, bcast = 10)
```

Arguments

All arguments are the same as in and passed intact to the `qtscore`. See help for this function.

- **formula**: Formula describing fixed effects to be used in analysis, e.g. \( y = a + b \) means that outcome \( y \) depends on two covariates, \( a \) and \( b \). If no covariates used in analysis, skip the right-hand side of the equation.
- **data**: An object of `gwaa.data-class`
- **snpsubset**: Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from `data` are used for analysis.
- **idsubset**: Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from `data/cc` are used for analysis.
- **strata**: Stratification variable. If provided, scores are computed within strata and then added up.
- **trait.type**: "gaussian" or "binomial". If not specified, the procedure guesses the type
- **times**: If more than one, the number of replicas to be used in derivation of empirical genome-wide significance. See `emp.qtscore`, which calls `qtscore` with \( \text{times} > 1 \) for details
- **quiet**: do not print warning messages
- **bcast**: If the argument `times` > 1, progress is reported once in `bcast` replicas

Details

In the analysis of empirical significance, first time the function `qtscore` is called and result object is saved. Later, the function `qtscore` is called `times` times with `replace=FALSE` in order to generate distribution under the null. Each call, minimal P-value is extracted and compared with original P-values. For a particular SNP, empirical P-value is obtained as a proportion of times minimal Ps from resampled data was less than original P.

The list elements `effB`, `effAB` and `effBB` are the ones obtained from the analysis of the original (not permuted) data set.

The function does not yet implement correct analysis for X-linked data.

Value

Object of class `scan.gwaa-class`

Note

Author(s)

Yuriii Aulchenko
estlambda

References

See Also

qtscore, emp.ccfast, scan.gwaa-class

Examples

data(srdta)
a<-qtscore(qt3~age+sex, data=srdta, snps=c(1:200))
plot(a)
# this does not make sense, as the whole experiment must be analysed, not a small region!
b<-emp.qtscore(qt3~age+sex, data=srdta, snps=c(1:200))
plot(b)

---

estlambda

Estimate the inflation factor for a distribution of P-values

Description

Estimate the inflation factor for a distribution of P-values or 1df chi-square test. The major use of this procedure is the Genomic Control, but can also be used to visualise the distribution of P-values coming from other tests.

Usage

estlambda(data, plot = TRUE, proportion = 1.0)

Arguments

data A vector of reals. If all are <=1, it is assumed that this is a vector of P-values, else it is treated as a vector of chi-squares with 1 d.f.
plot Wether the prot should be presented
proportion The proportion of lowest P (Chi2) to be used when estimating the inflation factor Lambda

Value

A list with elements

estimate Estimate of Lambda
se Standard error of the estimate

Author(s)

Yuri Aulchenko
See Also

ccfast, qtscore

Examples

data(srdta)
pex <- summary(srdta@gtdata)[,"Pexact"]
estlambda(pex)
a <- ccfast("bt",srdta)
a$lambda

export.merlin function to export GenABEL data in merlin format

Description

Usage

export.merlin(data, pedfile = "merlin.ped", datafile = "merlin.dat", mapfile = "merlin.map", format = "merlin", fixstrand = "no", extendedmap = TRUE, traits = 1)

Arguments

data gwaa.data object
pedfile Output pedigree data file name
datafile Output data (information) file name
mapfile Output map file name
format Output format: reserved for future use, currently only "merlin"
fixstrand "no" – the strand information and coding comes from the data; "+" – change all coding to "+" strand, "-" – change all coding to "-" strand
extendedmap if TRUE extended map (+ strand, + coding) is saved with the name "mapfile.ext", where "mapfile" is the parameter supplied by user
traits How many fake traits to insert before first column of marker data

Details

The use is straightforward, with only the "fixstrand" option requiring some explanation. Consider a SNP on "-" strand with alleles G and A. If this SNP is accessed on "+" strand, the corresponding alleles would be C and T. While for example Affymetrix reports SNPs on bot "+" and "-" strands, HapMap reports coding on "+" strand only. To make data compatible, and/or to run imputations, one will need to convert all SNP codes to "+" strand. This can be achieved by running `export.merlin()` with fixstrand="+" parameter.
Value

No value returned

Author(s)

Yurii Aulchenko

See Also

To load the data to GenABEL again, use `convert.snp.ped`, `load.gwaa.data`.

---

**ge03d2**  
**GWA-type data on few small region**

Description

`ge03d2` A small data set (approximately 1,000 people and 8,000 SNPs) containing data on 3 autosomes and X chromosome. Is a good set for demonstration of the QC procedures (different genotyping errors are introduced) and GWA analysis. Run demo(ge03d2) to see a demo. This data set was developed for the "Advances in population-based studies" (Ge03) course of the Nihes.

`ge03d2c` A small data set (approximately 200 people and 8,000 SNPs) containing data on 3 autosomes and X chromosome. This data set is complementary to `ge03d2`.

`ge03d2ex` A small data set (approximately 150 people and 4,000 SNPs) containing data on 3 autosomes and X chromosome. Is a good set for demonstration of the QC procedures (different genotyping errors are introduced) and GWA analysis. This data set was developed for the "Advances in population-based studies" (Ge03) course of the Nihes. See vignette "GenABEL-tutorial.pdf" for details.

Usage

```
data(ge03d2)
```

Format

Details

Source

References
**grammar**

**Approximate score test for association in related people**

**Description**

Fast approximate score test for association between a trait and genetic polymorphism, in samples of related individuals. When used with argument "times=1", it is equivalent to running `qtscore` on polygenic residuals from `polygenic`. However, it does not produce correct results with permutations, because the raw trait values, which are not exchangeable, are permuted. Use `qtscore` on polygenic residuals when you want to have empirical GW significance with GRAMMAR method.

**Usage**

```r
grammar(h2object, data, snpsubset, idsubset, strata, times=1, quiet=FALSE, bcast=10, clambda=FALSE, propPs=1.0)
```

**Arguments**

- **h2object**: An object returned by `polygenic` polygenic mixed model analysis routine. The sub-objects used are measuredIDs, residualY, h2an$estimates (last element, total variance, only), and InvSigma. One can supply grammar with a fake h2object, containing these list elements.

- **data**: An object of `gwaa.data-class`

- **snpsubset**: Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from `data` are used for analysis.

- **idsubset**: Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from `data/cc` are used for analysis.

- **strata**: Stratification variable. If provided, scores are computed within strata and then added up.

- **times**: If more than one, the number of replicas to be used in derivation of empirical genome-wide significance. NOTE: do not use times > 1 unless you are really sure you understand what you are doing!

- **quiet**: do not print warning messages

- **bcast**: If the argument times > 1, progress is reported once in bcast replicas
If inflation facor Lambda is estimated as lower than one, this parameter controls if the original P1df (clambda=TRUE) to be reported in Pc1df, or the original 1df statistics is to be multiplied onto this "deflation" factor (clambda=FALSE). With GRAMMAR, Lambda is expected ot be less than 1. If a numeric value is provided, it is used as a correction factor.

propPs  proportion of non-corrected P-values used to estimate the inflation factor Lambda, passed directly to the estlambda

Details

Approximate score test is performed using the formula

\[
\frac{\sigma^4 ((G - E[G])V^{-1}_{residualY})^2}{(G - E[G]) (G - E[G])}
\]

where \(\sigma^4\) is the square of the residual variance, \(G\) is the vector of genotypes (coded 0, 1, 2) and \(E[G]\) is a vector of (strata-specific) mean genotypic values; \(V^{-1}\) is the InvSigma and \(residualY\) are residuals from the trait analysis with polygenic procedure.

Compared to score test implemented in mmscore, grammar test is faster and computation time grows only linearly with the number of subjects (with mmscore this relation is quadratic). While raw P1df from grammar are not quite correct, the GC p-values correspond very closely to these from the mmscore.

Value

Object of class scan.gwaa-class; only 1 d.f. test is implemented currently.

Author(s)

Yurii Aulchenko

References


See Also

grammar, qtscore, plot.scan.gwaa, scan.gwaa-class

Examples
Description

This class contains objects holding all GWAA data – phenotypes, genotypes and other relevant information

Slots

- **phdata**: dataframe with phenotypic data used in GWAA
- **gtdata**: object of class `snp.data-class` used to store genotypic data, map, etc.

Extends

Methods

- **[** signature(x = "gwaa.data", i = "ANY", j = "ANY", drop = "ANY")**: sub-set operations. `x[i,j]` will select people listed in `i` and SNPs listed in `j`.
- **show** signature(object = "gwaa.data")**: shows both parts of the object. Take care that the objects are usually very large!
- **summary** signature(object = "gwaa.data")**: Calls standard summary to describe phenotypic part and calls `summary.snp.data` to `snp.data-class`

Author(s)

Yurii Aulchenko

See Also

- `snp.data-class`, `load.gwaa.data`, `snp.mx-class`

Examples

data(srdta)
srdta@phdata[1:10,]
srdta@gtdata[1:10,1:12]
srdta[1:10,1:12]
as.numeric(srdta@gtdata[1:12,1:10])
# very long output:
summary(srdta)
hom

function to compute average homozygosity within a person

Description

This function computes average homozygosity (inbreeding) for a set of people, across multiple markers. Can be used for Quality Control (e.g. contamination checks)

Usage

\[
\text{hom}(\text{data}, \text{snpsubset}, \text{idsubset}, \text{weight}="\text{no}")
\]

Arguments

- **data**: Object of \texttt{gwaa.data-class} or \texttt{snp.data-class}
- **snpsubset**: Subset of SNPs to be used
- **idsubset**: People for whom average homozygosity is to be computed
- **weight**: When "no", homozygosity is computed as a proportion of homozygous genotypes. When "freq", an estimate of inbreeding coefficient is computed (see details).

Details

With "freq" option, for person \(i\) inbreeding is estimated with

\[
f_i = \frac{O_i - E_i}{(L_i - E_i)}
\]

where \(O_i\) is observed homozygosity, \(L_i\) is the number of SNPs measured in individual \(i\) and

\[
E_i = \sum_{j=1}^{L_i} (1 - 2p_j(1 - p_j) \frac{T_{A_j}}{T_{A_j} - 1})
\]

where \(T_{A_j}\) is the number of measured genotypes at locus \(j\).

Only polymorphic loci with number of measured genotypes >1 are used with this option.

This measure is the same as used by PLINK (see reference).

You should use as many people and markers as possible when estimating inbreeding from marker data.

Value

With option weight="no": A matrix with rows corresponding to the ID names and columns showing the number of genotypes measured (NoMeasured) and homozygosity (Hom).

With option weight="freq": the same as above + expected homozygosity (E(Hom)) and the estimate of inbreeding, \(F\).
Note

Author(s)

Yurii Aulchenko

References


See Also

ibs, gwaa.data-class, snp.data-class

Examples

data(ge03d2)
h <- hom(ge03d2[,c(1:100)])
homsem <- h[,"Hom"]*(1-h[,"Hom"])/h[,"NoMeasured"]
plot(h[,"Hom"],homsem)
# wrong analysis: one should use all people (for right frequency) and markers (for right F)
h <- hom(ge03d2[,c(1:10)],weight="freq")
h

ibs

Computes (average) Identity-by-State for a set of people and markers

Description

Given a set of SNPs, computes a matrix of average IBS for a group of people

Usage

ibs(data, snpsubset, idsubset, weight="no")

Arguments

data object of snp.data-class
snpsubset Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from data are used for analysis.
idssubset Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from data are used for analysis.
weight "no" for direct IBS computations, "freq" to weight by allelic frequency
Details

This function facilitates quality control of genomic data. E.g. people with extremely high (close to 1) IBS may indicate duplicated samples (or twins), simply high values of IBS may indicate relatives.

When weight "freq" is used, IBS for a pair of people i and j is computed as

$$ f_{i,j} = \sum_k \frac{(x_{i,k} - p_k) \cdot (x_{j,k} - p_k)}{(p_k \cdot (1 - p_k))} $$

where k changes from 1 to N = number of SNPs GW, $x_{i,k}$ is a genotype of ith person at the kth SNP, coded as 0, 1/2, 1 and $p_k$ is the frequency of the "+" allele. This apparently provides an unbiased estimate of the kinship coefficient.

Only with "freq" option monomorphic SNPs are regarded as non-informative.

ibs() operation may be very lengthy for a large number of people.

Value

A (Npeople X Npeople) matrix giving average IBS (kinship) values between a pair below the diagonal and number of SNP genotype measured for both members of the pair above the diagonal.

On the diagonal, homozygosity (0.5+inbreeding) is provided.

Author(s)

Yurii Aulchenko

See Also

check.marker, summary.snp.data, snp.data-class

Examples

data(ge03d2c)
a <- ibs(data=ge03d2c,ids=c(1:10),snps=c(1:1000))
a

# compute IBS based on a random sample of 1000 autosomal marker
a <- ibs(ge03d2c,snps=sample(ge03d2c@gtdata@snpnames[ge03d2c@gtdata@chromosome!="X"],1000,replace=FALSE),weight="freq")
mds <- cmdscale(as.dist(1-a))
plot(mds)
# identify smaller cluster of outliers
km <- kmeans(mds,centers=2,nstart=1000)
c11 <- names(which(km$cluster==1))
c12 <- names(which(km$cluster==2))
if (length(c11) > length(c12)) c11 <- c12;
c11

# PAINT THE OUTLIERS IN RED
points(mds[c11,],pch=19,col="red")
load.gwaa.data  

function to load GWAA data

Description

Load data (genotypes and phenotypes) from files to gwaa.data object

Usage

load.gwaa.data(phenofile = "pheno.dat", genofile = "geno.raw",  
force = FALSE, makemap = FALSE, sort = TRUE)

Arguments

phenofile  
data table with phenotypes

 genofile  
internally formatted genotypic data file (see convert.snp.text to convert data)

force  
Force loading the data if heterozygous X-chromosome genotypes are found in male

makemap  
Make a consequent map in case if map is provided chromosome-specifically

sort  
Should SNPs be sorted in ascending order according to chromosome and position?

Details

The genofile must be the one resulting from convert.snp.text, convert.snp.ped, convert.snp.tped,  
or convert.snp.illumina (see documentation for these functions for the file formats).

The phenotype file relates study subjects with their covariate and outcome values. In the phenotypic data file, the first line gives a description of the data contained in a particular column; the names should be unique, otherwise R will change them. The first column of the phenotype file MUST contain the subjects’ unique ID, named "id"; there should also be a column named "sex" and giving sex information (0 = female, 1 = male). Other columns in the file should contain phenotypic information. Missing values should be coded with "NA"; binary traits should have values 0 or 1. An example of few first lines of a phenotype file is as follows:

id sex age bt1 qt qt1
"289982" 0 30.33 NA NA 3.93
"325286" 0 36.514 1 0.49 3.61
"357273" 1 37.811 0 1.65 5.30
"872422" 1 20.393 0 1.95 4.07
"1005389" 1 28.21 1 0.35 3.90

This file tells us that, for example, person 325286 is female (0 in second column), and she has "1" (usually this means a "case") value for the trait "bt1", so on. Person 289982 has measurements only for sex, age and qt1, while other measurements are missing (NA, Not Available).

IDs are better kept in quotation (this would keep away the problem of e.g., leading zeros).
mmscore

Value
Object of class gwaa.data

Author(s)
Yurii Aulchenko

See Also
save.gwaa.data, convert.snp.text, convert.snp.ped, convert.snp.tped, convert.snp.illumina

mmscore                  Score test for association in related people

Description
Score test for association between a trait and genetic polymorphism, in samples of related individuals

Usage
mmscore(h2object, data, snpsubset, idsubset, strata, times=1, quiet=FALSE, bcast=10, clambda=TRUE, propPs=1.0)

Arguments

h2object     An object returned by polygenic polygenic mixed model analysis routine. The sub-objects used are measuredIDs, residualY, and InvSigma. One can supply mmscore with a fake h2object, containing these list elements.
data         An object of gwaa.data-class
snpsubset    Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from data are used for analysis.
idsubset    Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from data/cc are used for analysis.
strata       Stratification variable. If provided, scores are computed within strata and then added up.
times        If more then one, the number of replicas to be used in derivation of empirical genome-wide significance. NOTE: The structure of the data is not exchangeable, therefore do not use times > 1 unless you are really sure you understand what you are doing!
quiet        do not print warning messages
bcast        If the argument times > 1, progress is reported once in bcast replicas
clambda      If inflation facot Lambda is estimated as lower then one, this parameter controls if the original P1df (clambda=TRUE) to be reported in Pc1df, or the original 1df statistics is to be multiplied onto this "deflation" factor (clambda=FALSE). If a numeric value is provided, it is used as a correction factor.
propPs       proportion of non-corrected P-values used to estimate the inflation factor Lambda, passed directly to the estlambda
Details

Score test is performed using the formula

\[
\frac{((G - E[G])V^{-1}residualY)^2}{(G - E[G])V^{-1}(G - E[G])}
\]

where \(G\) is the vector of genotypes (coded 0, 1, 2) and \(E[G]\) is a vector of (strata-specific) mean genotypic values; \(V^{-1}\) is the InvSigma and \(residualY\) are residuals from the trait analysis with polygenic procedure.

This test is similar to that implemented by Abecasis et al. (see reference).

Value

Object of class \texttt{scan.gwaa-class}; only 1 d.f. test is implemented currently.

Author(s)

Yuri Aulchenko

References


See Also

\texttt{grammar, qtscore, egscore, plot.scan.gwaa, scan.gwaa-class}

Examples

```r
perid.summary(data, snpsubset, idsubset)
```

Description

Produces call rate and heterozygosity per person

Usage

```r
perid.summary(data, snpsubset, idsubset)
```
Arguments

- **data**: object of `snp.data-class`
- **snps subset**: Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from `data` are used for analysis.
- **idsubset**: Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from `data` are used for analysis.

Details

This function facilitates quality control of genomic data. E.g. extreme outliers for heterozygosity indicate possibly contaminated DNA samples, while low call rate of a person may indicate poor DNA quality.

Value

A matrix, giving per person (row) its’ average heterozygosity ("Het" column) and call rate ("CallPP"), over all SNPs.

Author(s)

Yurii Aulchenko

See Also

`check.marker`, `summary.snp.data`, `snp.data-class`

Examples

```r
data(ge03d2c)
a <- perid.summary(data=ge03d2c,snps=c(1:100),ids=c(1:10))
a <- perid.summary(data=ge03d2c)
hist(a[,"CallPP"])
hist(a[,"Het"])
```

Description

Plots "check.marker" object, as returned by `check.marker`

Usage

```r
plot.check.marker(x, y, ...)
```
Arguments

x Object of class "check.marker", as returned by check.marker or snp.subset

y this argument is not used

... other arguments to be passed to plot

Details

In this plot, along the X axes, you can see colour representation of markers which did not pass (pass – black) the QC. The diagonal shows redundant markers. If for some marker there exist markers, which show exactly the same (or some minimum concordance) genotypic distribution, such markers are depicted as crosses an solid line is dropped on the X axes from it. Other solid line connects the original SNP with the redundant ones (depicted as circles). From each redundant SNP, a dashed line is dropped on X. Normally, one expects that redundant markers are positioned very closely and redundancy appears because of linkage disequilibrium.

Value

No value returned. Explanatory note is shown on the screen.

Author(s)

Yurii Aulchenko

See Also

check.marker, snp.subset

Examples

data(srdta)
mc <- check.marker(data=srdta@gtdata[,1:100], redundant="all", maf=0.01, minconcordance=0.9, fdr=.1,ibs.mrk=0)
mc <- check.marker(data=srdta@gtdata[,1:100], maf=0.01, fdr=.1,ibs.mrk=0)
plot(mc)
mcl <- snp.subset(mc, snps=srdta@gtdata@snpnames[20:40])
plot(mcl)

plot.scan.gwaa.2D function to plot 2D scan results

Description

Plots results of 2D analysis produced by scan.glm.2D or scan.haplo.2D

Usage

plot.scan.gwaa.2D(x, y, ..., df=1)
plot.scan.gwaa

**Arguments**

- `x`: object of type `scan.gwaa.2D-class`, as returned by `scan.glm.2D` or `scan.haplo.2D`
- `y`: this argument is not used
- `...`: additional arguments to be passed to `plot`
- `df`: Whether 1, 2, or "all" d.f.s should be plotted. Note that for `scan.haplo.2D` 1 and 2 d.f. list the same values.

**Details**

Now plots only "allelic" results. This is fine for `scan.haplo.2D` as only allelic tests are produced; however, `scan.glm.2D` also produces "genotypic" results.

**Value**

No value returned.

**Author(s)**

Yurii Aulchenko

**See Also**

`scan.gwaa.2D-class`, `scan.glm.2D`, `scan.haplo.2D`

**Examples**

```r
data(srdta)
a <- scan.glm.2D("qt3~CRSNP", data=srdta, snps=c(1:10))
# "allelic" results
plot(a)
# to plot "genotypic" results:
filled.contour(x=a$map, y=a$map, z=-log10(a$P2df))
```

---

**Description**

Plots results of GWA analysis

**Usage**

`plot.scan.gwaa(x, y, ..., df=1)`
polygenic

Estimation of polygenic model

Description

Estimates linear mixed (polygenic) model based on trait and covariates data and kinship matrix

Usage

polygenic(formula, kinship.matrix, data, fixh2, starth2=0.3, trait.type="gaussian", opt.method="nlm", scaleh2=1000, quiet=FALSE, ...)

Arguments

x  object of type scan.gwaa-class, as returned by scan.glm, qtscore, ccfast, emp.ccfast, emp.qtscore, or scan.haplo

y  this argument is not used

... additional arguments to be passed to plot

df  Plot results of 1 or 2-df test (1, 2). Could be also "Pc1df" (for GC corrected P-values) or "all" (to plot all three)

Value

No value returned.

Author(s)

Yurii Aulchenko

See Also

scan.gwaa-class, add.plot, snp.subset, scan.glm, qtscore, ccfast, emp.qtscore, emp.ccfast, scan.haplo

Examples

data(srdta)
a <- ccfast("bt",srdta,snps=c(1:250))
plot(a)
plot(a,df="all")
a1 <- snp.subset(a,snps=c(20:100))
plot(a1,df="all")
### Arguments

- **formula**: Formula describing fixed effects to be used in analysis, e.g., \( y = a + b \) means that outcome \((y)\) depends on two covariates, \(a\) and \(b\). If no covariates used in analysis, skip the right-hand side of the equation.

- **kinship.matrix**: Kinship matrix, as provided by e.g. `ibs(weight="freq")`, or estimated outside of GenABEL from pedigree data.

- **data**: An (optional) object of `gwaa.data-class` or a data frame with outcome and covariates

- **fixh2**: Optional value of heritability to be used, instead of maximisation. The uses of this option are two-fold: (a) testing significance of heritability and (b) using a priori known heritability to derive the rest of MLEs and var.-cov. matrix.

- **starth2**: Starting value for h2 estimate

- **trait.type**: "gaussian" or "binomial"

- **opt.method**: "nlm" or "optim". These two use different optimisation functions. `optim` is slower than `nlm`, but may give better results.

- **scaleh2**: Only relevant when "nlm" optimisation function is used. "scaleh2" is the heritability scaling parameter, regulating how "big" parameter changes in h2 with the respect to changes in other parameters. As other parameters are estimated from previous regression, these are expected to change little from the initial estimate. The default value of 1000 proved to work rather well under a range of conditions.

- **quiet**: If FALSE (default), details of optimisation process are reported.

- **...**: Optional arguments to be passed to `nlm` (`optim`) minimisation function.

### Details

This function maximises the likelihood of the data under polygenic model with covariates an reports the maximum likelihood estimates and the inverse of variance-covariance matrix at the point of ML.

One of the major use of this function is to estimate residuals of the trait and the inverse of the variance-covariance matrix for further use in analysis with `mmscore` and `grammar`.

Also, it can be used for a variant of GRAMMAR analysis, which allows for permutations for GW significance by use of polygenic residuals as an analysis trait with `qtscore`.

"Polygenic residuals" (not to be mistaken with just "residuals") are the residual where both the effect of covariates AND the estimated polygenic effect (breeding values) are factored out. This thus provides an estimate of the trait value contributed by environment (or, turning this other way around, the part of trait not explained by covariates and by the polygene). Polygenic residuals are estimated as

\[
\sigma^2 V^{-1}(Y - (\hat{\mu} + \hat{\beta}C_1 + ...))
\]

where \(\sigma^2\) is the residual variance, \(V^{-1}\) is the InvSigma (inverse of the var-cov matrix at the maximum of polygenic model) and \((Y - (\hat{\mu} + \hat{\beta}C_1 + ...))\) is the trait values adjusted for covariates (also at at the maximum of polygenic model likelihood).
It can also be used for heritability analysis. If you want to test significance of heritability, estimate the model and write down the function minimum reported at "h2an" element of the output (this is \(-2\times\text{MaxLikelihood}\)). Then do next round of estimation, but set fixh2=0. The difference between you function minima gives you one-sided test distributed as chi-squared with 1 d.f.

The way to compute the likelihood is partly based on the paper of Thompson (see refs), namely instead of taking inverse of var-cov matrix every time, eigenvectors of the inverse of G (taken only once) are used.

Value

A list with values

- **h2an**: A list supplied by the nlm minimisation routine. Of particular interest are elements "estimate" containing parameter maximal likelihood estimates (MLEs) (order: mean, betas for covariates, heritability, (polygenic + residual variance))
- **residualY**: Residuals from analysis, based on covariate effects only; NOTE: these are NOT grammar polygenic residuals!
- **esth2**: Estimate (or fixed value) of heritability
- **pgresidualY**: Polygenic residuals from analysis, based on covariate effects and predicted breeding value.
- **InvSigma**: Inverse of the variance-covariance matrix, computed at the MLEs – these are used in mmscore and grammar functions.
- **call**: The details of call
- **measuredIDs**: Logical values for IDs who were used in analysis (traits and all covariates measured) == TRUE

Author(s)

Yuri Aulchenko

References


See Also

mmscore, grammar

Examples
Fast score test for association between a trait and genetic polymorphism

Usage

```r
qtscore(formula, data, snpsubset, idsubset, strata, trait.type="gaussian", times=1, quiet=FALSE, bcast=10, clambda=TRUE, propPs=1.0)
```

Arguments

- **formula**: Formula describing fixed effects to be used in analysis, e.g. `y ~ a + b` means that outcome (y) depends on two covariates, a and b. If no covariates used in analysis, skip the right-hand side of the equation.
- **data**: An object of `gwaa.data-class`
- **snpsubset**: Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from `data` are used for analysis.
- **idsubset**: Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from `data/cc` are used for analysis.
- **strata**: Stratification variable. If provided, scores are computed within strata and then added up.
- **trait.type**: "gaussian" or "binomial"
- **times**: If more then one, the number of replicas to be used in derivation of empirical genome-wide significance. See `emp.qtscore`, which calls `qtscore` with `times>1` for details
- **quiet**: do not print warning messages
- **bcast**: If the argument `times > 1`, progress is reported once in `bcast` replicas
- **clambda**: If inflation facot Lambda is estimated as lower then one, this parameter controls if the original P1df (clambda=TRUE) to be reported in Pc1df, or the original 1df statistics is to be multiplied onto this "deflation" factor (clambda=FALSE). If a numeric value is provided, it is used as a correction factor.
- **propPs**: proportion of non-corrected P-values used to estimate the inflation factor Lambda, passed directly to the `estlambda`

Details

When formula contains covariates, the traits is analysed using GLM and later residuals used when score test is computed for each of the SNPs in analysis. For binary traits, residuals from GLM are transformed using `exp(x)/(1+exp(x))`.

With no adjustment for binary traits, 1 d.f., the test is equivalent to the Armitage test.

This is a valid function to analyse GWA data, including X chromosome. For X chromosome, stratified analysis is performed (strata=sex).
qvaluebh95

Computes Benjamini-Hochberg (95) q-value

Description

Computes Benjamini-Hochberg (95) q-value

Usage

qvaluebh95(p, fdrate=0.1)

Arguments

p vector containing p-values
fdrate desired FDR
0.1. NAMES

Details

Value
List

0.1 Names

normal-bracket17bracket-normal

pass Is true if this P-value passed specified FDR
qvalue qvalue

normal-bracket17bracket-normal

Names

pass Is true if this P-value passed specified FDR
qvalue qvalue

Author(s)
Yuri Aulchenko

See Also

Examples

data(srdta)
a<-qtscore(qt2, data=srdta)
qv <- qvaluebh95(a$P1df)
plot(a$map, -log10(qv$qvalue))
r2fast

Estimates r2 between multiple markers

Description

Given a set of SNPs, computes a matrix of r2

Usage

r2fast(data, snpsubset, idsubset)

Arguments

data object of snp.data-class
snpsubset Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from data are used for analysis.
idsubset Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from data are used for analysis.

Details

The function is based on slightly modified code of Hao et al.

Value

A (Nsnps X Nsnps) matrix giving r2 values between a pairs of SNPs above the diagonal and number of SNP genotype measured for both SNPs below the diagonal

Author(s)

Yurii Aulchenko

References


See Also

rhofast
Examples

data(ge03d2)
# r2s using r2fast
a <- r2fast(ge03d2, snps=c(1:10))
# r2s using package genetics
b <- LD(as.genotype(ge03d2[,1:10]))$"R^2"
# see that the r2s are not exactly the same
cor(a[upper.tri(a)], b[upper.tri(b)])
pplot(a[upper.tri(a)], b[upper.tri(b)])

redundant  function to do redundancy check

Description

Checks marker redundancy, understood as concordance between genotypic distributions (including missing values)

Usage

redundant(data, pairs = "bychrom", minconcordance = 2.0)

Arguments

data  gwaa.data or snp.data object
pairs  "bychrom" or "all" to check pairs within chromosome only or genome-wide
minconcordance  find "redundant" pairs of markers with concordance => "minconcordance". If "minconcordance" is more then 1.0, only pairs of markers which are exactly the same (independent of coding), including NA pattern, are considered as redundant. If "minconcordance" is <= 1, the concordance rate is computed as percent of genotypes which are the same, including the genotypes with NA. I.e. if both genotypes are NA, this is counted as a match, if one is NA and other is measured, this is counted as mismatch. Note that option with "minconcordance" <= 1 takes much longer time to run.

Value

A list containing reference SNP as a name and all SNPs which has "the same" genotypic distribution as values:

"refSNP1"   SNP11, SNP12, ...
"refSNP2"   SNP21, SNP22, ...
...  
"refSNPlast"   SNPlast1, SNPlast2, ...
"all"   list of all redundant SNPs, which can be dropped from consideration
Author(s)

Yurii Aulchenko

See Also

check.marker

Examples

data(srdta)
redundant(srdta@gtdata)
redundant(srdta@gtdata[,1:50],minconcordance=0.8)

refresh.gwaa.data  Updates an object from old to new GenABEL format

Description

Attempts to update an object of gwaa.data-class from old to new format

Usage

refresh.gwaa.data(data)

Arguments

data  An object of gwaa.data-class in pre-1.2-6 (data version 0) format.

Details

Takes old-style gwaa.data object and sets @coding and @strand attributes to SNPs. All coding is set to 1/2 and strand is set to "u" (unknown).

Value

Object of gwaa.data-class in new (GenABEL v > 1.2-6, raw data format version 0.1) format.

Author(s)

Yurii Aulchenko

See Also

load.gwaa.data
rhofast

Estimates rho between multiple markers

Description

Given a set of SNPs, computes a matrix of rho

Usage

rhofast(data, snpsubset, idsubset)

Arguments

- **data**: object of `snp.data-class`
- **snpsubset**: Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from `data` are used for analysis.
- **idsubset**: Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from `data` are used for analysis.

Details

Rho is the measure of association described by N. Morton and A. Collins (see reference). The function is based on slightly modified code of Hao et al.

Value

A (Nsnps X Nsnps) matrix giving rho values between a pairs of SNPs above the diagonal and Kij below the diagonal.

Author(s)

Yurii Aulchenko

References


See Also

`r2fast`
Examples

data(ge03d2)
# rhos using rhofast
a <- rhofast(ge03d2, snps = c(1:10))
# rhos using package genetics
b <- LD(as.genotype(ge03d2[, 1:10]))$"R^2"
# see that the rhos are not exactly the same
cor(a[upper.tri(a)], b[upper.tri(b)])
plot(a[upper.tri(a)], b[upper.tri(b)])

Description

Usage

save.gwaa.data(data, phenofile = "pheno.dat", genofile = "geno.raw",
human = FALSE)

Arguments

data gwaa.data object
phenofile name of file where the phenotypes will be saved to
genofile name of file where the genotypes will be saved to
human if human=TRUE, saves in human-readable format (to be converted to internal
format later)

Details

When running with human=TRUE, a lot of memory (and time to complete the operation) is required. Probably, this option would not work because of memory limitations in a GWA scan with more than a few hundreds of people. This is possible to fix; drop me a message if you need that.

Value

No value returned

Author(s)

Yurii Aulchenko

See Also

load.gwaa.data
scan.glm.2D

Scans regional data allowing for gene-gene interaction using glm

Description

Scans regional data allowing for gene-gene interaction using glm

Usage

scan.glm.2D(formula, family = gaussian(), data, snpsubset, idsubset, bcast = 50)

Arguments

- **formula**: character string containing formula to be used in glm. You should put CRSNP argument in the formula, to arrange how the SNP from the list would be treated. This allows to put in an interaction term.
- **family**: family to be passed to glm
- **snpsubset**: Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from data are used for analysis.
- **idsubset**: Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from data/cc are used for analysis.
- **data**: object of class "gwaa.data"
- **bcast**: show progress every bcast SNPs

Details

Value

Object of class scan.gwaa.2D-class

Author(s)

Yurii Aulchenko

See Also

scan.gwaa.2D-class, scan.haplo.2D

Examples

data(srdta)
a <- scan.glm.2D("bt~sex+age+CRSNP",family=binomial(),data=srdta,snps=(1:10),bcast=2)
plot(a)
scan.glm

Scan GWA data using glm

Description

Scan GWA data using glm

Usage

scan.glm(formula, family = gaussian(), data, snpsubset, idsubset, bcast = 50)

Arguments

formula character string containing formula to be used in glm. You should put CRSNP argument in the formula, to arrange how the SNP from the list would be treated. This allows to put in an interaction term.

family family to be passed to glm

snpsubset Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from data are used for analysis.

idsubset Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from data/cc are used for analysis.

data object of class "gwaa.data"

bcast show progress every bcast SNPs

Details

Value

Object of class scan.gwaa-class

Author(s)

Yurii Aulchenko

See Also

ccfast, qtscore, scan.gwaa-class
Examples

data(srdta)
a <- scan.glm("bt~sex+age+CRSNP", family=binomial(), data=srdta, snps=(1:10), bcast=2)
plot(a)

osnp <- "rs4934"
maposnp <- srdta@gtdata@map[osnp]
maposnp
reg <- snp.names(srdta, begin=maposnp-100000, end=maposnp+100000, chrom="1")
a <- scan.glm("qt3~sex+age+CRSNP", data=srdta, snps=reg)
plot(a)
plot(a, df="all")

# interaction with sex
a <- scan.glm("qt3~age+sex+CRSNP", data=srdta, snps=reg)
plot(a, df="all")
# you can do interaction with a selected polymorphisms in the same way

---

scan.gwaa-class Class "scan.gwaa"

Description

This class contains results of GWA analysis. This is an list object, generated by scan.glm, scan.haplo, ccfast, qtscore, emp.ccfast, or emp.qtscore.

Names

snpnames list of names of SNPs tested
P1df corresponding list of P-values of 1-d.f. (additive or allelic) test for association between SNP and trait
P2df corresponding list of P-values of 2-d.f. (genotypic) test for association between SNP and trait
Pc1df P-values from the 1-d.f. test for association between SNP and trait; the statistics is corrected for possible inflation
effB Effect of the B allele in allelic test (OR for ccfast, difference from the mean for qtscore and beta from the scan.glm)
effAB Effect of the AB genotype in genotypic test
effBB Effect of the BB genotype in genotypic test
map list of map positions of the SNPs
chromosome list of chromosomes the SNPs belong to
idnames  list of people used in analysis

lambda  list with elements "estimate" (inflation factor estimate, as computed using lower 90 percents of the distribution) and "se" (standard error of the estimate)

formula  which formula/function call was used to compute P-values

family  family of the link function / nature of the test

Methods

plot  signature(object = "scan.gwaa"): Plots summary of GWAA

Author(s)

Yurii Aulchenko

See Also

ccfast, qtscore, scan.glm, scan.haplo.emp.ccfast, emp.qtscore, estlambda, plot.scan.gwaa

Examples

data(srdta)
sc <- scan.glm("qt3~CRSNP", data=srdta, snps=c(1:10))
class(sc)
sc$P1df
sc$P2df
sc
plot(sc)

scan.gwaa.2D-class  Class "scan.gwaa.2D"

Description

This class contains results of 2D analysis. This is a list object, generated by scan.glm.2D or scan.haplo.2D.

Names

snpnames  list of names of SNPs tested

P1df  corresponding list of P-values of allelic test for association between SNP and trait.

Pint1df  corresponding list of P-values of significance of the interactions between SNPs, for the allelic model
scan.gwaa.2D-class

P2df  corresponding list of P-values of genotypic test for association between SNP and trait For link(scan.haplo) and link(scan.haplo.2D) this is equal to P1df and has nothing to do with the actual degrees of freedom of the test

Pint1df  corresponding list of P-values of significance of the interactions between SNPs for the genotypic test

medChi1df  Median Chi-square for allelic test

medChi2df  Median Chi-square on genotypic test

map  list of map positions of the SNPs

chromosome  list of chromosomes the SNPs belong to

formula  which formula/function call was used to compute P-values

family  family of the link function / nature of the test

idnames  list of people used in analysis

Methods

plot signature(object = "scan.gwaa.2D"): Plots summary of 2D scan, using list element P1df

Author(s)

Yurii Aulchenko

See Also

scan.gwaa.2D-class, scan.glm.2D, scan.haplo.2D, plot.scan.gwaa.2D

Examples

data(srdta)
sc <- scan.glm.2D("qt3~CRSNP", data=srdta, snps=c(1:10))
class(sc)
sc$P1df
sc$P2df
sc
plot(sc)
scan.haplo.2D runs haplo.score.slide with all pairs of markers in a region

Description

Runs haplo.score.slide from the package haplo.stats on all pairs of markers in a region and presents output as scan.gwaa.2D-class object

Usage

scan.haplo.2D(formula, data, snpsubset, idsubset, bcast = 10, simulate=FALSE, trait.type, ...)

Arguments

- **formula**: Formula to be used in analysis. It should be a character string following standard notation. On the left-hand side, there should be outcome. On the right-hand side, covariates are list, with "+" separating the covariates (additive action). The left- and right-hand sides are separated by " ". You should put CRSNP argument in the formula. For example "qt3 CRSNP" would analyse association between SNPs and trait "qt3", without any adjustment. To adjust for age and sex, use "qt3 age+sex+CRSNP". Currently, only additive effects (+) are allowed.

- **data**: object of class gwaa.data-class

- **snpsubset**: Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from data are used for analysis.

- **idsubset**: Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from data/cc are used for analysis.

- **bcast**: show progress every bcast percents of progress

- **simulate**: if simulated P-values should be generated

- **trait.type**: Character string defining type of trait, with values of "gaussian", "binomial", "poisson", "ordinal" (see help for haplo.score.slide for details). If not specified, the routine picks up "gaussian" or "binomial" (two levels of trait).

- **...**: other arguments to be passed to haplo.score.slide

Details

List element P2df is set equal to P1df, as only allelic results are returned. This has nothing to do with actual degrees of freedom of the test.

Value

Object of class scan.gwaa.2D-class

Author(s)

Yuriii Aulchenko
References

For haplo.stats (scan.haplo, scan.haplo.2D), please cite:


See Also

scan.gwaa.2D-class, scan.haplo, scan(glm.2D, haplo.score.slide

Examples

```r
data(srdta)
c <- scan.haplo.2D("bt~sex+age+CRSNP", data=srdta, snps=(717:733),
   ids=(srdta@phdata$age<40))
plot(c)
```

Description

Runs `haplo.score.slide` from the package `haplo.stats` and represents output as `scan.gwaa-class` data object

Usage

```r
scan.haplo(formula, data, snpsubset, idsubset, n.slide = 2, bcast = 10, simulate=FALSE, trait.type, ...)
```

Arguments

- **formula**: Formula to be used in analysis. It should be a character string following standard notation. On the left-hand side, there should be outcome. On the right-hand side, covariates are listed, with "+" separating the covariates (additive action). The left- and right-hand sides are separated by " ". You should put CRSNP argument in the formula. For example "qt3 CRSNP" would analyse association between SNPs and trait "qt3", without any adjustment. To adjust for age and sex, use "qt3 age+sex+CRSNP". Currently, only additive effects ("+") are allowed.
- **data**: object of class `gwaa.data-class`
- **snpsubset**: Index, character or logical vector with subset of SNPs to run analysis on. If missing, all SNPs from `data` are used for analysis.
- **idsubset**: Index, character or logical vector with subset of IDs to run analysis on. If missing, all people from `data/cc` are used for analysis.
- **n.slide**: Default = 2. Number of loci in each contiguous subset. The first subset is the ordered loci numbered 1 to `n.slide`, the second subset is 2 through `n.slide+1` and so on. If the total number of loci in `geno` is `n.loci`, then there are `n.loci - n.slide + 1` total subsets.
bcast show progress every bcast SNPs
simulate if simulated P-values should be generated
trait.type Character string defining type of trait, with values of "gaussian", "binomial", "poisson", "ordinal" (see help for \texttt{haplo.score.slide} for details). If not specified, the routine picks up "gaussian" or "binomial" (two levels of trait).
...
other arguments to be passed to \texttt{haplo.score.slide}

Details

List element P2df is set equal to P1df, as only allelic results are returned. This has nothing to do with degrees of freedom.

Value

Object of class \texttt{scan.gwaa-class}

Author(s)

Yurii Aulchenko

References

For \texttt{haplo.stats} (\texttt{scan.haplo, scan.haplo.2D}), please cite:


See Also

\texttt{scan.gwaa-class, haplo.score.slide}

Examples

data(srdta)
a <- ccfast("bt", srdta, snps=(717:733), ids=(srdta@phdata$age<40))
b <- scan.haplo("bt~sex+CRSNP", srdta, snps=(717:733),
ids=(srdta@phdata$age<40))
c <- scan.haplo("bt~sex+CRSNP", srdta, snps=(717:733),
ids=(srdta@phdata$age<40), n.slide=3)
plot(a)
add.plot(b, col="red", type="l")
add.plot(c, col="darkgreen", type="l")
Description

This function calls web browser and direct it to NCBI MapViewer, to show the region of interest.

Usage

\texttt{show.ncbi(region)}

Arguments

\texttt{region} \hspace{1cm} a vector containing regional landmarks

Details

The elements of input vector could be SNP rs-names

Value

Note

Author(s)

Yurii Aulchenko

References

See Also

Examples

\texttt{show.ncbi(c("rs7926624","rs11564708"))}
snp.coding-class  

Class "snp.coding"

Description

This class contains the actual nucleotide codes for the typed SNPs

Slots

.Data: nucleotide coding data

Methods

[ signature(x = "snp.coding", i = "ANY", j = "missing", drop = "missing"): subset operations. x[i] will show coding for SNPs selected in i.

coerce signature(from = "snp.coding", to = "character"): converts SNP coding from internal (raw) to human-readable character.

show signature(object = "snp.coding"): shows the object. Take care that this is internal representation

Author(s)

Yurii Aulchenko

See Also

snp.strand-class, gwaa.data-class, snp.data-class

Examples

data(srdta)
srdta@gtdata@coding[1:10]
as.character(srdta@gtdata@coding[1:10])

snp.data-class  

Class "snp.data"

Description

This class contains objects holding large arrays of single nucleotide polymorphism (SNP) genotypes
snp.data-class

Slots

**nbytes**: number of bytes used to store data on a SNP

**nids**: number of people

**male**: male code

**idnames**: ID names

**nsnps**: number of SNPs

**nsnpnames**: list of SNP names

**chromosome**: list chromosomes corresponding to SNPs

**coding**: list of nucleotide coding for the SNPs

**strand**: strands of the SNPs

**map**: list SNPs’ positions

**gtps**: `snp.mx-class` object used to store genotypes

Methods

```
[ signature(x = "snp.data", i = "ANY", j = "ANY", drop = "ANY")]: subset operations. x[i,j] will select people listed in i and SNPs listed in j.

coerce signature(from = "snp.data", to = "numeric")]: map to codes 0, 1, 2, or NA

coerce signature(from = "snp.data", to = "character")]: map to actual nucleotide codes, e.g. "A/A", "A/G", "G/G", ""

coerce signature(from = "snp.data", to = "genotype")]: map to data frame with genotype-class data, for later use with package genetics

coerce signature(from = "snp.data", to = "hsgeno")]: map to data frame with allelic data frame, for later use with package haplo.stats

show signature(object = "snp.data")]: shows the object. Take care that the objects are usually very large!

summary signature(object = "snp.data")]: calculate allele frequencies, genotype frequencies, and chi-square tests for Hardy-Weinberg equilibrium. Results are returned as a dataframe
```

Author(s)

Yuriii Aulchenko

See Also

`gwaa.data-class, snp.data, snp.mx-class`
Examples

```r
data(srdta)
class(srdta)
x <- srdta@gtdata
class(x)
x@nids
x@nsnps
x@idnames[1:12]
x@male[1:12]
x@male[c("p1","p2","p3","p4")]
x@snpnames[1:4]
x@chromosome[1:4]
x@map[1:4]
n4 <- c("rs18","rs655")
n4
x@map[n4]
n4 <- c("rs18","rs65")
n4
x@map[n4]
x@chromosome[n4]
x[1:12,1:4]
summary(x[,1:10])
as.numeric(x[,1:12,1:4])
as.numeric(x[,c("p1","p3","p4"),c("rs18","rs65")])
as.character(x[,c("p1","p3","p4"),c("rs18","rs65")])
as.genotype(x[,c("p1","p3","p4"),c("rs18","rs65")])
as.hsgeno(x[,c("p1","p3","p4"),c("rs18","rs65")])
```

snp.data creates an snp.data object

Description

Creates object of class **snp.data-class**

Usage

```r
snp.data(nids, rawdata, idnames = as.character(c(1:nids)),
  snpnames = as.character(c(1:(length(rawdata)/ceiling(nids/4)))),
  chromosome = as.factor(rep(1,(length(rawdata)/ceiling(nids/4)))),
  map = as.double(seq(1,(length(rawdata)/ceiling(nids/4))))),
  coding=as.raw(rep(1,length(rawdata)/ceiling(nids/4))),
  strand=as.raw(rep(0,length(rawdata)/ceiling(nids/4))),
  male = rep(0, nids))
```

Arguments

- **nids**: number of people
- **idnames**: list of IDs
snp.mx-class

Slots

- **male**: male indicator for IDs
- **snpnames**: list of SNP names
- **chromosome**: list of chromosomes SNPs belong to
- **coding**: list of nucleotide coding for the SNPs
- **strand**: strands of the SNPs
- **map**: map position of SNPs
- **rawdata**: genotypes presented in raw data format

Value

Object of class `snp.data-class`

Author(s)

Yuri Aulchenko

See Also

- `snp.data-class`

---

Description

This low-level class contains objects holding large arrays of single nucleotide polymorphism (SNP) genotypes

Methods

- **[** signature(x = "snp.mx", i = "ANY", j = "ANY", drop = "ANY")**: subset operations. x[i,j] will select people listed in i and SNPs listed in j.
- **coerce** signature(from = "raw", to = "snp.mx"): makes an snp.mx object out of raw data
- **show** signature(object = "snp.mx"): shows the object. Take care that (a) this is internal representation and (b) the objects are usually very large!

Note

User is not supposed to work with this class. Use `snp.data-class`. 
snp.names

Author(s)
Yurii Aulchenko

See Also
gwaa.data-class, snp.data-class

snp.names extracts names of SNPs in a region

Description
Based on boundary conditions specified and (or) chromosome selects SNP names in the region

Usage
snp.names(data, begin, end, chromosome)

Arguments
data object of class gwaa.data-class, snp.data-class, scan.gwaa-class or check.marker-class
begin Start position (or name of the first SNP)
end End-position or name of last SNP
chromosome Chromosome code

Details
Any of the arguments, except the data can be missing

Value
A vector of names of SNPs located in the region

Author(s)
Yurii Aulchenko

References

See Also
snp.data-class
Examples

data(srdta)
snp.names(srdta, begin = 50000, end = 100000)
snp.names(srdta, begin = 50000, end = 100000, chromosome = "1")

# does not make sense with these data:
snp.names(srdta, begin = 50000, end = 100000, chromosome = "X")

# again makes sense:
snp.names(srdta, end = 100000)
snp.names(srdta, begin = 2200000)

# show summary for SNPs in region between 50,000 and 100,000
a <- snp.names(srdta, begin = 50000, end = 100000)
snp.strand(srdta@gtdata[,a])

---

snp.strand-class Class "snp.strand"

Description

This class contains the strands of the typed SNPs

Slots

.Data: nucleotide strand data

Methods

[ signature(x = "snp.strand", i = "ANY", j = "missing", drop = "missing"): subset operations. x[i] will show strand for SNPs selected in i.

coerce signature(from = "snp.strand", to = "character"): converts SNP strand from internal (raw) to human-readable character.

show signature(object = "snp.strand"): shows the object. Take care that this is internal representation

Author(s)

Yuri Aulchenko

See Also

snp.coding-class, gwaa.data-class, snp.data-class

Examples

data(srdta)
srdta@gtdata@strand[1:10]
as.character(srdta@gtdata@strand[1:10])
snp.subset function to subset objects of class scan.gwaa and check.marker

Description
Computing objects of class scan.gwaa may take long, especially when haplotypic analysis is performed. Therefore this function helps substracting results on some region (indicated by list of SNPs)

Usage
snp.subset(data, snpsubset)

Arguments
data object of class scan.gwaa-class or check.marker-class
snpsubset character vector of snps to select

Value
Object of class scan.gwaa-class or check.marker-class

Author(s)
Yurii Aulchenko

See Also
scan.gwaa-class, check.marker-class

Examples
data(srdta)
# processing check.marker object
mc <- check.marker(data=srdta@gtdata[,1:100], redundant="all", maf=0.01, minconcordance=0.9, fdr=0.1, ibs.mrk=0)
m1 <- check.marker(data=srdta@gtdata[,1:100], maf=0.01, fdr=0.1, ibs.mrk=0)
snpsubset mc
# processing scan.gwaa object
a <- scan.glm("qt3~sex+age+CRSNP", data=srdta, snps=(1:30))
plot(a)
a1 <- snp.subset(a, snps=srdta@gtdata@snpnames[10:20])
plot(a1)
**snps.cell-class**  
*Class “snps.cell”*

**Description**

This is a lowest-level class based on which **snp.mx-class** is build.

**Note**

User is not supposed to work with this class. Use **snp.data-class**.

**Author(s)**

Yurii Aulchenko

**See Also**

**snp.mx-class, gwaa.data-class, snp.data-class**

---

**srdta**  
*GWA-type data on small region*

**Description**

**srdta** contains gwaa.data object with results on a small region of about 2.5 Mb. 833 SNPs are typed on 2500 people. NA rate is 95%. Sex, age, two quantitative (qt1 and qt2) and one binary (bt) traits are available for analysis. Run demo(srdta) and check tut-srdta.pdf to see examples of work with this data set. Original data files used for this set are located at YOUR_R_LIB_LOCATION/exdata/srphenos.dat (pheinotypes), srgenos.dat (human-readable genotypes) and srgenos.raw (genotypes in internal format).

**Usage**

```
data(srdta)
```

**Format**

Standard object of class **gwaa.data-class**

**Details**

**Source**
Examples

#main example: use this to see full functionality
# demo(srdta)

# load and work with srdta
data(srdta)
mc <- check.marker(data=srdta@gtdata[,1:100],redundant="all",maf=0.01,minconcordance=0.9,fdr=.1,ibs.mrk=0)
plot(mc)
check.trait(names(srdta@phdata),srdta)

---

sset  

Internal use function for class snp.mx-class

Description

Interface to C function sset subsetting genotypes from snp.mx-class

Usage

sset(data, nsnps, nids, list)

Arguments

data  
genotypic data in internal format
nsnps  
no. snps
nids  
no. people
list  
something internal...

Details

Value

Sub-set from snp.mx-class object

Note

Author(s)

Yuri Aulchenko
**Summary**

Provides cross-tabulation summarising number of marker which did not pass this or that criteria

**Usage**

```r
summary.check.marker(object, ...)
```

**Arguments**

- `object` object of class `check.marker-class`
- `...` additional arguments (not used)

**Value**

A list containing 2 tables: per-marker and per-person inconsistencies

**Author(s)**

Yuriii Aulchenko

**See Also**

- `check.marker`
- `check.marker-class`

**Examples**

```r
data(srdta)
mc <- check.marker(srdta, ids=c(1:500))
summary(mc)
```
**summary.gwaa.data**  
function to summarise GWAA data

**Description**

Summary of phenotypic and genotypic parts of GWAA data

**Usage**

```r
summary.gwaa.data(object, ...)
```

**Arguments**

- **object**: object of class `gwaa.data-class`
- **...**: additional arguments (not used)

**Value**

Returns list with two elements:

- **pheno**: Summary for phenotypic part of gwaa.data object
- **geno**: Summary for genotypic part of gwaa.data object

**Author(s)**

Yurii Aulchenko

**See Also**

`summary.snp.data`

**Examples**

```r
data(srdta)
# be prepared : long output!
summary(srdta)
```
**summary.snp.data**

**function to summary GWAA data**

**Description**

Provides summary of an object of class `snp.data-class`. Number of observed genotypes, allelic frequency, genotypic distribution, P-value of the exact test for HWE and chromosome are listed.

**Usage**

```r
summary.snp.data(object, ...)
```

**Arguments**

- `object` snp.data object
- `...` additional arguments (not used)

**Value**

Summary for snp.data object

**Note**

The P-values reported for X-chromosome are based on analysis of female data, but other statistics (frequencies, calls, ...) are based on all data.

**Author(s)**

Yurii Aulchenko

**References**


**See Also**

- `summary.gwaa.data`
- `snp.data-class`

**Examples**

```r
data(srdta)
summary(srdta@gtdata[,1:20])
```
Index

*Topic IO
convert.snp.illumina, 27
convert.snp.mach, 28
convert.snp.ped, 29
convert.snp.text, 31
convert.snp.tped, 33
export.merlin, 45
load.gwaa.data, 53
refresh.gwaa.data, 67
save.gwaa.data, 69

*Topic classes
check.marker-class, 22
gwaa.data-class, 49
scan.gwaa-class, 72
scan.gwaa.2D-class, 73
snp.coding-class, 79
snp.data, 81
snp.data-class, 79
snp.mx-class, 82
snp.strand-class, 84
snps.cell-class, 86

*Topic datasets
ge03d2, 46
srdta, 86

*Topic distribution
catable, 20
descriptives.marker, 35
descriptives.scan, 36
descriptives.trait, 37

*Topic hplot
add.plot, 6
plot.check.marker, 56
plot.scan.gwaa, 58
plot.scan.gwaa.2D, 57

*Topic htest
ccfast, 21
dprfast, 38
egscore, 39
emp.ccfast, 41
emp.qtscore, 42
estlambda, 44
grammar, 47
hom, 50
ibs, 51
mmscore, 54
perid.summary, 55
polygenic, 59
qtscore, 62
qvaluebh95, 63
r2fast, 65
redundant, 66
rhofast, 68
scan.glm, 71
scan.glm.2D, 70
scan.haplo, 76
scan.haplo.2D, 75
summary.check.marker, 88
summary.gwaa.data, 89
summary.snp.data, 90

*Topic misc
as.character.gwaa.data, 7
as.character.snp.coding, 8
as.character.snp.data, 9
as.character.snp.strand, 10
as.data.frame.gwaa.data, 11
as.double.gwaa.data, 12
as.double.snp.data, 13
as.genotype, 14
as.genotype.gwaa.data, 15
as.genotype.snp.data, 16
as.hsgeno, 17
as.hsgeno.gwaa.data, 18
as.hsgeno.snp.data, 19
check.marker, 23
check.trait, 26
crnames, 34
HWE.show, 4
show.ncbi, 78
snp.names, 83
snp.subset, 85
sset, 87
Xfix, 3

*Topic package
GenABEL-package, 1
[, gwaa.data, ANY, ANY, ANY-method
  (gwaa.data-class), 49
[,.snp.coding, ANY, missing, missing-method
  (snp.coding-class), 79
[,.snp.data, ANY, ANY, ANY-method
  (snp.data-class), 79
[,.snp.mx, ANY, ANY, ANY-method
  (snp.mx-class), 82
[,snp.strand, ANY, missing, missing-method
  (snp.strand-class), 84

add.plot, 6, 59
as.character.gwaa.data, 7, 12, 14,
  15, 18
as.character.snp.coding, 8, 10
as.character.snp.data, 7, 8, 9, 10–19
as.character.snp.strand, 8, 10
as.data.frame.gwaa.data, 11
as.double.gwaa.data, 7, 12, 12, 14, 15,
  18
as.double.snp.data, 7–12, 13, 14–19
as.genotype, 14
as.genotype.gwaa.data, 7, 12, 14, 15,
  18
as.genotype.snp.data, 7–15, 16,
  17–19
as.hsgeno, 7–10, 12–16, 17, 18
as.hsgeno.gwaa.data, 18
as.hsgeno.snp.data, 19

catable, 20
ccfast, 2, 6, 21, 41, 42, 45, 59, 71–73
check.marker, 2, 4, 5, 22, 23, 23, 26, 52,
  56, 57, 67, 88
check.marker-class, 25, 83, 85, 88
check.marker-class, 22
check.trait, 2, 25, 26
coerce,.snp.coding, character-method
  (snp.coding-class), 79
coerce,.snp.data, character-method
  (snp.data-class), 79
coerce, SNP.data, genotype-method
  (snp.data-class), 79
c coerce, SNP.data, hsgeno-method
  (snp.data-class), 79
c coerce, SNP.data, numeric-method
  (snp.data-class), 79
c coerce, SNP.mx, character-method
  (snp.mx-class), 82
c coerce, SNP.mx, numeric-method
  (snp.mx-class), 82

c convert.snp.illumina, 2, 27, 29, 31,
  32, 34, 53, 54
c convert.snp.mach, 2, 28, 28, 31, 32, 34
convert.snp.ped, 2, 28, 29, 29, 32, 34,
  46, 53, 54
convert.snp.text, 2, 28, 29, 31, 31, 34,
  53, 54
c crnames, 34
data.frame-class, 11
descriptives.marker, 2, 35
descriptives.scan, 2, 36
descriptives.trait, 2, 37
dimnames, 34
dprfast, 2, 38
egscore, 2, 39, 55, 63
emp.ccfast, 6, 22, 41, 44, 59, 72, 73
emp.qtscore, 6, 21, 41, 42, 42, 43, 59, 62,
  63, 72, 73
estlambda, 21, 40, 44, 48, 54, 62, 73
export.merlin, 45

g03d2, 46, 46
g03d2c, 46, 46
g03d2ex, 46, 46
GenABEL (GenABEL-package), 1
GenABEL-package, 1
genotype, 80
glm, 70, 71
grammar, 2, 47, 48, 55, 60, 61
gwaa.data-class, 2, 4, 7, 11, 12, 15, 18,
  21, 35–37, 40, 41, 43, 47, 51, 54, 60,
  62, 67, 75, 76, 79, 80, 83, 84, 86, 89
gwaa.data-class, 5, 49, 50

ehaplo.score.slide, 75–77
hom, 2, 50
HWE.show, 2, 4, 25
ibs, 2, 25, 40, 51, 51
load.gwaa.data, 2, 28, 29, 31, 32, 34, 46, 49, 53, 67, 69
mmscore, 2, 41, 48, 54, 60, 61, 63
n1m, 60, 61
optim, 60
perid.summary, 2, 20, 25, 55
plot, 6
plot, check.marker-method (check.marker-class), 22
plot, scan.gwaa-method (scan.gwaa-class), 72
plot, scan.gwaa.2D-method (scan.gwaa.2D-class), 73
plot.check.marker, 3, 23, 25, 56
plot.scan.gwaa, 3, 22, 41, 48, 55, 58, 63, 73
plot.scan.gwaa.2D, 57, 74
polygenic, 2, 47, 48, 54, 55, 59
pop2.ids (ge03d2), 46
qc.nostr (ge03d2), 46
qc.str (ge03d2), 46
qt.score, 2, 6, 41–45, 47, 48, 55, 59, 60, 62, 71–73
qvaluebh95, 63
r2fast, 2, 65, 68
redundant, 23–25, 66
refresh.gwaa.data, 67
rho.fast, 2, 39, 65, 68
save.gwaa.data, 54, 69
scan.glm, 3, 6, 59, 71, 72, 73
scan.glm.2D, 3, 6, 57, 58, 70, 73, 74, 76
scan.gwaa-class, 2, 3, 21, 22, 40–44, 48, 55, 59, 63, 71, 76, 77, 83, 85
scan.gwaa-class, 72
scan.gwaa.2D-class, 3, 58, 70, 74–76
scan.gwaa.2D-class, 73
scan.haplo, 3, 6, 59, 72, 73, 76, 76
scan.haplo.2D, 3, 6, 57, 58, 70, 73, 74, 75
show, gwaa.data-method (gwaa.data-class), 49
show, snp.coding-method (snp.coding-class), 79
show, snp.data-method (snp.data-class), 79
show, snp.mx-method (snp.mx-class), 82
show, snp.strand-method (snp.strand-class), 84
show.ncbi, 2, 78
snp.coding-class, 8, 84
snp.data, 80, 81
snp.data-class, 2, 4, 9, 13, 14, 16, 17, 19, 35–38, 49, 51, 52, 56, 65, 68, 79, 81–84, 86, 90
snp.data-class, 49, 50, 79
snp.mx-class, 49, 80, 86–88
snp.mx-class, 80, 82
snp.names, 2, 83
snp.strand-class, 10, 79
snp.strand-class, 84
snpsubset, 2, 6, 57, 59, 85
snps.cell-class, 86
srdata, 86
sset, 87
summary, check.marker-method (check.marker-class), 22
summary, gwaa.data-method (gwaa.data-class), 49
summary, snp.data-method (snp.data-class), 79
summary, snp.mx-method (snp.mx-class), 82
summary.check.marker, 23, 25, 88
summary.gwaa.data, 89, 90
summary.snp.data, 2, 20, 24, 25, 49, 52, 56, 89, 90
Xfix, 5