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 or 2)

Value

No value returned.

Author(s)

Yurii 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")
```

as.character.gwaa.data *Attempts to convert genotypic part of gwaa.data to character*

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

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])
```

as.character.snp.data *Attempts to convert snp.data to character*

Description

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

Usage

```
as.character.snp.data(x, ...)
```

Arguments

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

Details

Value

A matrix containing genotypes in character format

Note

Author(s)

Yurii Aulchenko

References

See Also

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

Examples

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

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)

Yurii Aulchenko

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

Examples

```
data(srdta)
as.double(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 gwaa.data-class to "genotype" data frame

Usage

```
as.genotype.gwaa.data(x, ...)
```

Arguments

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

Details

Value

A data-frame containing genotypes 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

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

`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])
```

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

Description

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

Usage

```
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

```
data(srdta)
as.genotype(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)

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.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

```
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])
```

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

Description

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

Usage

```
as.hsgeno(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 =
```

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)
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)

Yurii 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 summrising 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 redudndance, 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, het.fdr = 0.01, ibs.threshold = 0.95, ibs.mrk = 2000, maf, p.l
                odds = 1000, hweidsubset, redundant = "no",
                minconcordance = 2.0, qoption = "bh95")
```

Arguments

`data`	gwaa.data or snp.data object
`snpsubset`	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)
`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.
`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 initally 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

```
# 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-chromsome 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$nohwe
mc <- check.marker(data=srdta@gtdata[,1:100],call=0.95,perid.call=0.9,maf=0.02,minconcordance=0.9,fdr=0.1
summary(mc)
HWE.show(data=srdta,snps=mc$nohwe)
plot(mc)
```

`check.trait` *function to do primitive trait quality control*

Description

This function check 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 particulat measurment is an outlier is compted as folowing. Consider trait vector Y with particulat i^{th} measurment denodet as y_i. Let Y(-i) is vector, which is the same as Y, except that i^{th} measurment is dropped. Then Chi-square for measurment i is computed as

$$Chi_i = (mean(Y(-i)) - y_i)^2 / var(Y(-i))$$

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)

Yurii Aulchenko

See Also

`check.marker`

Examples

```
data(srdta)
check.trait("qt3",data=srdta)
n <- names(srdta@phdata)
check.trait(n,data=srdta)
```

`convert.snp.ped` *function to convert genotypic data in pre-makeped linkage fromat (+map) to internal genotypic data formated file*

Description

Converts genotypic data in pre-makeped linkage fromat (+map) to internal genotypic data formated file

Usage

```
convert.snp.ped(pedfile, mapfile, outfile, bcast = 10000)
```

Arguments

`pedfile`	Pre-makeped linkage genotypic data file name
`mapfile`	Mega2 map file
`outfile`	Output data file
`bcast`	Reports progress after reading bcast portion of SNPs

Details

Pedfile must be standard pre-makeped linkage file. In this file, columns are

ped id fa mo sex affection

Sex is coded as 1=male and 2=female. Affection status is not used. For example

1 1 0 0 1 2 1 1 1 2

1 2 0 0 1 0 1 2 1 2

1 3 0 0 2 1 2 2 1 1

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.

The map file is standard Mega2 map. For example:

chrom kosambi name

18 2859916 rs679153

18 2860891 rs9965482

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

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.text

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 formated file*

Description

Converts integer genotypic data file to raw internal data formated 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 3 2

3 2 3 3 3

2 2 1 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

Examples

```
#
# 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

```
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 C A C C C C C

1 rs5678 0 5000830 G T G T G G T 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.text` `load.gwaa.data`,

Examples

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

`descriptives.marker` *Function to generate descriptive summary tables for genotypic data*

Description

Function to generate descriptive summary tables for genotypic data

Usage

```
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

Value

A list containing descriptive tables and statistics

Note

Author(s)

Yurii Aulchenko

References

See Also

Examples

```
data(srdta)
descriptives.marker(srdta)
```

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

Value

A descritive table

Note

Author(s)

Yurii Aulchenko

References

See Also

Examples

```
data(srdta)
x <- qtscore(qt2,srdta)
descriptives.scan(x)
```

`descriptives.trait` *Function to generate descriptive summary tables for phenotypic data*

Description

Function to generate descriptive summary tables for phenotypic data

Usage

```
descriptives.trait(data,subset,file,by=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 otput).
`by`	a binary trait; statistics will be produced seprately for the groups and compared
`digits`	number of digits to be printed

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

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

`dprfast` *Estimates D' between multiple markers*

Description

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

Usage

```
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)

Yurii Aulchenko

References

Hao K, Di X, Cawley S. (2006) LdCompare: rapid computation of single- and multiple-marker D' and genetic coverage. Bioinformatics, 23: 252-254.

See Also

rhofast

Examples

```
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)])
```

`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

```
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

Details

In the analysis of empirical significance, first time the function ccfast is called and result object is saved. Later, the function ccfast is called `times` times with `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 scan.gwaa-class

Note

Author(s)

Yurii Aulchenko

See Also

ccfast, emp.qtscore, scan.gwaa-class

Examples

```
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)
```

---

emp.qtscore *Genome-wide significance for a GWA scan*

Description

Genome-wide significance for a GWA scan. Analysis function is qtscore.

Usage

```
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 provieded, scores are computed within strata and then added up.
trait.type	"gaussian" or "binomial". If not specified, the procedure quesses the type
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

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 then 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)

Yurii Aulchenko

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)

Yurii Aulchenko

See Also

ccfast, qtscore

Examples

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

---

ge03d2 *GWA-type data on small region*

Description

ge03d2 A small data set (approximately 1,000 people and 8,000 SNPs) containing data on 3 autosomes and X chromsome. Is a good set for demonatration 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.

Usage

```
data(ge03d2)
```

Format

Details

Source

References

Examples

```
#main example: use this to see full functionality
# demo(ge03d2)

# load and work with ge03d2
data(ge03d2)
a <- qtscore(dm2,ge03d2)
plot(a)
```

---

ge03d2c *GWA-type data on small region*

Description

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

Usage

```
data(ge03d2c)
```

Format

Details

Source

References

Examples

```
#main example: use this to see full functionality
# demo(ge03d2)

# load and work with ge03d2c
data(ge03d2c)
mc <- check.marker(ge03d2c)
summary(mc)
```

ge03d2ex *GWA-type data on small region*

Description

ge03d2ex A small data set (approximately 150 people and 4,000 SNPs) containing data on 3 autosomes and X chromsome. Is a good set for demonatration 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(ge03d2ex)
```

Format

Details

Source

References

Examples

```
#main example: use this to see full functionality
# demo(ge03d2ex)

# load and work with ge03d2ex
data(ge03d2exex)
```

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 computa-tional 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 pref-erably performed using computationally intensive permutation test procedures, further increasing the com-putational 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

Package:	GenABEL
Type:	Package
Version:	1.2-4
Date:	2007-06-14
License:	GNU GPL v. 2.0 or later

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 priveleges in Windows, it may well not run. In this case, try demo(srdtawin).

The most important functions and classes are:

For loading the data, see `convert.snp.text`, `convert.snp.ped`, `load.gwaa.data`.

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

For data managment, 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, emp.ccfast, emp.qtscore, ibs, r2fast, dprfast, rhofast

For specific tools facilitating analysis of the data with stratification (population stratification or (possibly unknown) pedigree structure), see ibs, polygenic, mmscore, grammar.

Also, principle components analysis (cmdscale may be useful function in this context.

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:

Aulchenko Y.S., Ripke S., Isaacs A., van Duijn C.M. (2007) GenABEL: an R package for genome-wide association analysis. Bioinformatics.

For exact HWE, please cite:

Wigginton G.E., Cutler D.J., Abecasis G.R. (2005) A note on exact tests of Hardy-Weinberg equilibrium. Am J Hum Genet, 76: 887-893.

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

Schaid DJ, Rowland CM, Tines DE, Jacobson RM, Poland GA. (2002) Score tests for association between traits and haplotypes when linkage phase is ambiguous. Am J Hum Genet, 70: 425-434.

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

Hao K, Di X, Cawley S. (2006) LdCompare: rapid computation of single- and multiple-marker r2 and genetic coverage. Bioinformatics, 23: 252-254.

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)
#
```

---

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 exchangable, are permuted. Use `qtscore` on polygenic residuals when you wnat to have empirical GW significance with GRAMMAR method.

Usage

```
grammar(h2object,data,snpsubset,idsubset,strata,times=1,quiet=FALSE,bcast=10,clambda=FALSE,prop
```

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 provieded, 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: 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)
`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

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

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

Chen WM, Abecasis GR (2007) Family-based assoication tests for genome-wide association scans.

See Also

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

Examples

gwaa.data-class *Class "gwaa.data"*

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"): subset operations. [x,y] will select people listed in x and SNPs listed in y.

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

```
hom(data, snpsubset, idsubset, weight="no")
```

Arguments

`data`	Object of gwaa.data-class or 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 coefficint 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

$$E_i = \Sigma_{j=1}^{L_i}(1 - 2p_j(1 - p_j)\frac{T_{Aj}}{T_{Aj} - 1})$$

where T_{Aj} is the numer of measured genotypes at locus j.

Only polymorphic loci with number of measured henotypes >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 colums 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

Purcell S. et al, (2007) PLINK: a toolset for whole genome association and population-based linkage analyses. Am. J. Hum. Genet.

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) available!
h <- hom(ge03d2[,c(1:10)],weight="freq")
h
```

HWE.show *show HWE tables*

Description

This function displays HWE tables and shows Chi2 and exact HWE P-values

Usage

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

Arguments

data	object of class gwaa.data-class or 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/cc are used for analysis.

Value

Only screen output

Author(s)

Yurii Aulchenko

See Also

check.marker

Examples

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

---

ibs — *Computes (average) Idenity-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.
idsubset	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 exteremly 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

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

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),
mds <- cmdscale(as.dist(1-a))
plot(mds)
# identify smaller cluster of outliers
km <- kmeans(mds,centers=2,nstart=1000)
cl1 <- names(which(km$cluster==1))
cl2 <- names(which(km$cluster==2))
if (length(cl1) > length(cl2)) cl1 <- cl2;
cl1
# PAINT THE OUTLIERS IN RED
points(mds[cl1,],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)
```

Arguments

`phenofile`	data table with pehnotypes
`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 consequtive map in case if map is provided chromosome-specifically

Details

The genofile must be the one resulting from `convert.snp.text`, see documentation for this function for the file format.

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).

Value

Object of class gwaa.data

Author(s)

Yurii Aulchenko

See Also

save.gwaa.data, convert.snp.text

---

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,propP
```

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 provieded, 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 exchangable, 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)

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

Author(s)

Yurii Aulchenko

References

Chen WM, Abecasis GR (2007) Family-based assoication tests for genome-wide association scans.

See Also

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

Examples

`perid.summary` *Summary of marker data per person*

Description

Produces call rate and heterozygosity per person

Usage

```
perid.summary(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

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

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

`plot.check.marker` *plots "check.marker" object*

Description

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

Usage

```
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
plot(mc)
mc1 <- snp.subset(mc,snps=srdta@gtdata@snpnames[20:40])
plot(mc1)
```

---

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)
```

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 "allelelic" results. This is fine for scan.haplo.2Das only alleic 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

```
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))
```

---

plot.scan.gwaa *function to plot GWAA results*

Description

Plots results of GWA analysis

Usage

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

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. Could be also "all" (to plot both)

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")
```

---

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",
```

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 dirrerent 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" are 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*MaxLikleihood). Then do next round of estimation, but set fixh2=0. The difference between you function minima gives you one-sided test distribued as chi-squared with 1 d.f.

The way to compute the likleihood 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)

Yurii Aulchenko

References

Thompson EA, Shaw RG (1990) Pedigree analysis for quantitative traits: variance components without matrix inversion. Biometrics 46, 399-413.

See Also

`mmscore`, `grammar`

Examples

qtscore *Fast score test for association*

Description

Fast score test for association between a trait and genetic polymorphism

Usage

```
qtscore(formula,data,snpsubset,idsubset,strata,trait.type="gaussian",times=1,quiet=FALSE,bcast=
```

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 provieded, 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)
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).

Value

Object of class scan.gwaa-class

Author(s)

Yurii Aulchenko

See Also

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

Examples

```
data(srdta)
#qtscore with stratification
a <- qtscore(qt3~sex,data=srdta)
plot(a)
b <- qtscore(qt3,strata=srdta@phdata$sex,data=srdta)
add.plot(b,col="green",cex=2)
# qtscore with extra adjustment
a <- qtscore(qt3~sex+age,data=srdta)
a
plot(a)
# compare results of score and chi-square test for binary trait
a1 <- ccfast("bt",data=srdta,snps=c(1:100))
a2 <- qtscore(bt,data=srdta,snps=c(1:100),trait.type="binomial")
plot(a1,ylim=c(0,2))
add.plot(a2,col="red",cex=1.5)
# the good thing about score test is that we can do adjustment...
a2 <- qtscore(bt~age+sex,data=srdta,snps=c(1:100),trait.type="binomial")
points(a2$map,-log10(a2$P1df),col="green")
```

---

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 FRD

Details

Value

List

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)

Yurii 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

Hao K, Di X, Cawley S. (2006) LdCompare: rapid computation of single- and multiple-marker r2 and genetic coverage. Bioinformatics, 23: 252-254.

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)])
plot(a[upper.tri(a)],b[upper.tri(b)])
```

`redundant` *function to do redundancy check*

Description

Checks marker redundancy, understood as comcordance 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 dismatch. 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)
```

---

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

Collins A, Morton NE. (1998) Mapping a disease locus by allelic association. PNAS, 17:1741-1745.

Hao K, Di X, Cawley S. (2006) LdCompare: rapid computation of single- and multiple-marker rho and genetic coverage. Bioinformatics, 23: 252-254.

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)])
```

---

save.gwaa.data *function to save gwaa.data object*

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 iwth more then 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 bestween SNP and trait

P2df corresponding list of P-values of 2-d.f. (genotypic) test for association bestween SNP and trait

Pc1df P-values from the 1-d.f. test for association bestween 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 comput 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 an 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 bestween SNP and trait.

Pint1df corresponding list of P-values of signifcance of the interactions between SNPs, for the allelic model

P2df corresponding list of P-values of genotypic test for association bestween 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 fo freedom of the test

Pint1df corresponding list of P-values of signifcance 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 comput P-values

family family of the link function / nature if the test

idnames list of people used in analysis

Methods

plot `signature(object = "scan.gwaa.2D")`: Plots summary of 2D scan, using lsit 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-had side, there should be outcome. On the right-hand side, covariates are liste, 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 asociation 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 calss 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)

Yurii Aulchenko

References

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

Schaid DJ, Rowland CM, Tines DE, Jacobson RM, Poland GA. (2002) Score tests for association between traits and haplotypes when linkage phase is ambiguous. Am J Hum Genet, 70: 425-434.

See Also

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

Examples

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

---

scan.haplo *scan.haplo*

Description

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

Usage

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

Arguments

`formula`	Formula to be used in analysis. It should be a character string following standard notation. On the left-had side, there should be outcome. On the right-hand side, covariates are liste, 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 asociation 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 calss `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 `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 degrees of freedom.

Value

Object of class `scan.gwaa-class`

Author(s)

Yurii Aulchenko

References

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

Schaid DJ, Rowland CM, Tines DE, Jacobson RM, Poland GA. (2002) Score tests for association between traits and haplotypes when linkage phase is ambiguous. Am J Hum Genet, 70: 425-434.

See Also

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")
```

---

show.ncbi *Shows the region on NCBI map*

Description

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

Usage

```
show.ncbi(region)
```

Arguments

region	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

```
show.ncbi(c("rs7926624","rs11564708"))
```

`snp.data-class` *Class "snp.data"*

Description

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

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

`map`: list SNPs' positions

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

Methods

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

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

coerce `signature(from = "snp.data", to = "character")`: map to codes "A/A", "A/B", "B/B", ""

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)

Yurii Aulchenko

See Also

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

Examples

```
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

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

Arguments

nids	number of people
idnames	list of IDs
male	male indicator for IDs
snpnames	list of SNP names
chromosome	list of chromosomes SNPs belong to
map	map position of SNPs
rawdata	genotypes presented in raw data format

Value

Object of class snp.data-class

Author(s)

Yurii Aulchenko

See Also

snp.data-class

---

snp.mx-class *Class "snp.mx"*

Description

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

Slots

.Data: object used to store genotypes

Methods

[`signature(x = "snp.mx")`: subset operations. [x,y] will select people listed in x and SNPs listed in y.

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`.

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)
summary(srdta@gtdata[,a])
```

---

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=.1,ibs.mrk=0
plot(mc)
mc1 <- snp.subset(mc,snps=srdta@gtdata@snpnames[20:50])
plot(mc1)
# 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 (pehnotypes), 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

References

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)

Yurii Aulchenko

References

See Also

snp.mx-class

Examples

summary.check.marker *Summary of check.marker object*

Description

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

Usage

```
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)

Yurii Aulchenko

See Also

check.marker, check.marker-class

Examples

```
data(srdta)
mc <- check.marker(srdta)
summary(mc)
```

summary.gwaa.data *function to summarise GWAA data*

Description

Summary of phenotypic and genotypic parts of GWAA data

Usage

```
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

```
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 distrbution, P-value of the exact test for HWE and chromosome are listed

Usage

```
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 (freqencies, calls, ...) are based on all data.

Author(s)

Yurii Aulchenko

References

Wigginton, JE, Cutler, DJ, and Abecasis, GR (2005) A Note on Exact Tests of Hardy-Weinberg Equilibrium. American Journal of Human Genetics. 76: 887-93.

See Also

summary.gwaa.data, snp.data-class

Examples

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

Xfix *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

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-chromsome markers
mc1 <- check.marker(fixed0)
# fix minor X-chromosome problems
fixed1 <- Xfix(fixed0)
# no errors
mc2 <- check.marker(fixed1)
summary(mc2)
```