Package 'rJPSGCS'

February 4, 2018

Type Package

Title R-Interface to Gene Drop Simulation from JPSGCS

Version 0.2-8

Maintainer Brad McNeney <mcneney@sfu.ca>

Depends R (>= 2.12.0), rJava (>= 0.8-4), chopsticks (>= 1.18.0)

Imports methods (>= 3.1.0), utils (>= 3.1.0)

SystemRequirements Java (>= 8), zlib headers and library.

Description R-interface to gene drop programs from Alun Thomas' Java Programs for Statistical Genetics and Computational Statistics (JPSGCS).

License GPL-3

LazyLoad yes

NeedsCompilation yes

Author Sigal Blay [aut],
Jinko Graham [aut],
Brad McNeney [aut, cre],
Annick Nembot-Simo [aut],
Alun Thomas [ctb],
Hin-Tak Leung [ctb]

Repository CRAN

Date/Publication 2018-02-04 16:47:00 UTC

R topics documented:

rJPSGCS-package *R wrappers for JPSGCS gene drop programs.*

Description

R wrappers for gene drop programs in Alun Thomas' Java Programs for Statistical Genetics and Computational Statistics (JPSGCS).

Details

R wrappers for programs from JPSGCS required to simulate data by gene drop (`GeneDrops`) based on a fitted LD model (`FitGMLD`). JPSGCS java source code was downloaded from Alun Thomas' website (no longer active) in March 2011. See the package NEWS file (`RShowDoc("NEWS", package="rJPSGCS")`) for a list of modifications needed to interface with R.

Author(s)

Sigal Blay, Jinko Graham, Brad McNeney and Annick Nembot-Simo

Maintainer: Brad McNeney <mcneney@sfu.ca>

exdat *example data for rJPSGCS*

Description

Example data, comprised of simulated genotypes for 10 SNPs on the same chromosome, for five unrelated individuals, along with a genetic map for the SNPs. The fifth subject has missing genotypes at SNPs 3 and 4.

Usage

```
data(exdat)
```

Format

A list with the following elements:

`$markers`	a numeric matrix of SNP genotypes coded as 0, 1 or 2 copies of an index allele
`$map`	a numeric vector containing the genetic map positions for the SNPs in centiMorgans

Examples

```
data(exdat)
```

`FitGMLD` *estimate a graphical model for linkage disequilibrium*

Description

This function is an R wrapper for the JPSGCS program to estimate a graphical model for linkage disequilibrium (LD) from a sample of genotypes. The genetic markers are assumed to be on the same chromosome.

Usage

```
FitGMLD(par = "input.par", fped = "flipped.ped", out.ld.par = "out.ld.par")
```

Arguments

`par`	Name of a LINKAGE parameter file (also called a DATAFILE: see http://linkage.rockefeller.edu/soft/linkage/) that contains marker and genetic map information.
`fped`	Name of a LINKAGE pedigree file "flipped" so that rows correspond to loci and columns correspond to individuals.
`out.ld.par`	Name of the file in the current working directory to contain the fitted LD model.

Details

The input pedigree file `fped` is not the standard format where rows correspond to individuals and columns to loci. Rather, it has been transposed so that rows correspond to loci. Transposed LINKAGE pedigree files may be written by the `write.pedfile` function with the option `transpose=TRUE`.

Further information on the graphical model is available from the article listed in the References.

Value

None. The result of the function call is the output file `out.ld.par`.

Note

This function can require large amounts of memory for large data sets (e.g. whole chromosomes of SNPs at the marker density of the HapMap project). The computation for fitting the LD model is done in java, using functions from Alun Thomas' suite of Java Programs for Statistical Genetics and Computational Statistics (JPSGCS). The JPSGCS java programs are accessed by R-wrappers provided by the **rJPSGCS** package, which initializes the java virtual machine (JVM) if not already done so by another package. To over-ride the default amount of memory java allocates for heap space, initialize the JVM *before* loading **rJPSGCS** as follows:

```
options(java.parameters="-Xmx2048m") #set max heap space to 2GB
library(rJava)
```

```
.jinit() #initialize the JVM
library(rJPSGCS) # now load rJPSGCS
```

Author(s)

Sigal Blay, Jinko Graham and Brad McNeney

References

Thomas A. Estimation of graphical models whose conditional independence graphs are interval graphs and its application to modeling linkage disequilibrium. Comput Stat Data Anal. 2009; 53:1818-1828.

See Also

write.parfile, write.pedfile

Examples

```
## Not run:
data(exdat)
sdat<-as(exdat$markers,"snp.matrix") #coerce to snp.matrix
# Write a LINKAGE parameter file
write.parfile(snp.data=sdat,map=exdat$map,file="test.par")
# Write a "flipped" pedfile for the SNP data on unrelated subjects.
write.pedfile(pedinf="unrelated",snp.data=sdat,file="ftest.ped",transpose=TRUE)
FitGMLD(par="test.par",fped="ftest.ped",out.ld.par="test.ld.par")
# Clean up
unlink(c("test.par","ftest.ped","test.ld.par"))

## End(Not run)
```

GeneDrops *simulate genotypes on a pedigree*

Description

This function is an R wrapper for the JPSGCS multi-locus gene-drop program.

Usage

```
GeneDrops(ld.par = "ld.par", ped = "in.ped", n, complete.data=FALSE,
gd.ped = "gd.ped", compress.pedfiles=FALSE)
```

Arguments

ld.par	Name of a LINKAGE parameter file or an LD model file output by FitGMLD.
ped	Name of the post-MAKEPED pedfile (see the explanation in the documentation for read.pedfile) giving the pedigree relationships and marker data availability to simulate.
n	Number of pedigrees to simulate.
complete.data	Should complete data be simulated; i.e., should we ignore the marker data availability in ped? Default is FALSE.
gd.ped	Name of the output file of simulated marker data in post-MAKEPED pedigree file format.
compress.pedfiles	Should the output pedigree file be compressed? Default is FALSE.

Value

No value. The output is in the file named by gd.ped.

Note

This function can require large amounts of memory for large data sets (e.g. whole chromosomes of SNPs at the marker density of the HapMap project). The computations are done in Java, using functions from Alun Thomas' suite of Java Programs for Statistical Genetics and Computational Statistics (JPSGCS). The JPSGCS Java programs are accessed by R-wrappers provided by the **rJPSGCS** package, which initializes the Java virtual machine (JVM) if not already done so by another package. To over-ride the default amount of memory Java allocates for heap space, initialize the JVM *before* loading **rJPSGCS** as follows:

```
options(java.parameters="-Xmx2048m") #set max heap space to 2GB
library(rJava)
.jinit() #initialize the JVM
library(rJPSGCS) # now load rJPSGCS
```

Author(s)

Sigal Blay, Jinko Graham and Brad McNeney

See Also

FitGMLD

Examples

```
# 1. Gene-drop assuming independent loci in the founders, using allele
# frequencies and map information from the example data set.
data(exdat)
# Print out the data to examine missing data pattern.
exdat$markers
```

```
#Coerce to snp.matrix
sdat<-as(exdat$markers,"snp.matrix")
# Need LINKAGE parameter and pedigree files as input.
# For independent loci in the founders use a standard LINKAGE parameter file.
write.parfile(snp.data=sdat, map=exdat$map, file="indep.par")
# To construct the pedigree file for unrelated individuals, use
# the pedinf="unrelated" option.
write.pedfile(pedinf="unrelated",snp.data=sdat,file="unrel.ped")
GeneDrops(ld.par="indep.par",ped="unrel.ped",n=10,gd.ped="gd.ped")
# Read the SNP data back into R.
simdat<-read.pedfile("gd.ped")$snp.data
# Look at the simulated data: need to coerce to numeric matrix to see
# values of individual genotypes, rather than the usual snp.matrix summary
as(simdat,"numeric")

# 2. Simulate data at 5 independent SNPs 0.1 cM apart on a parent-child trio.
map=(1:5)/10
# Marker data on each individual as follows:
mom=c(1,2,1,1,1) # complete data
dad=c(0,NA,0,1,0) # missing data at SNP 2
child=c(NA,1,1,2,NA) #missing data at SNPs 1 and 5
sdat<-rbind(mom,dad,child)
colnames(sdat)<-paste("SNP",1:5,sep="")
sdat<-as(sdat,"snp.matrix")
write.parfile(snp.data=sdat, map=map, file="trio.par")
# Block of code setting up pedigree information for mom, dad and child.
# See the Details section of the write.pedfile help file for
# information on the file format.
trioinf<-matrix(c(
    1, 1, 0, 0, 0, 0, 0, 2, 1, #mom: ID=1, no parents in pedigree
    1, 2, 0, 0, 0, 0, 0, 1, 1, #dad: ID=2, no parents in pedigree
    1, 3, 2, 1, 0, 0, 0, 2, 1),#child: ID=3, dadID=2, momID=1
    byrow=TRUE, ncol=9, nrow=3)
write.pedfile(pedinf=trioinf, snp.data=sdat,file="trio.ped")
GeneDrops(ld.par="trio.par",ped="trio.ped",n=10,gd.ped="gd.ped")
#Read the SNP data back into R.
simdat<-read.pedfile("gd.ped")$snp.data
# Look at the missing data patterns in simdat
as(simdat,"numeric")

## Not run:
# 3. Repeat gene drop for the trio, but this time use an LD model fit
# by FitGMLD. FitGMLD requires a LINKAGE parameter file and a transposed
# pedigree file as input. Can use the parameter file trio.par again.
# The transposed pedigree file can be generated as follows:
write.pedfile(pedinf=trioinf,snp.data=sdat,file="f.ped",transpose=TRUE)
FitGMLD("trio.par","f.ped","out.ld.par")
#Call to GeneDrops uses the output of FitGMLD as the parameter file and
# the regular (not transposed) pedigree file
GeneDrops(ld.par="out.ld.par",ped="trio.ped",n=10,gd.ped="gd.ped")
simdat<-read.pedfile("gd.ped")$snp.data
unlink(c("f.ped","out.ld.par"))
```

```
## End(Not run)
unlink(c("unrel.ped","trio.ped","indep.par","trio.par", "gd.ped"))
```

`read.pedfile` *read a LINKAGE pedigree file into a* `snp.matrix` *object*

Description

This function reads post-MAKEPED pedigree files (used by JPSGCS) into a `snp.matrix` object. The post-MAKEPED format includes nine columns of pedigree structure and subject characteristics (see **Details** below) before the marker data. By contrast, pre-MAKEPED pedigree file format has only six column of pedigree and subject data before the marker data. This function was adapted from the `read.snps.pedfile` function in the **chopsticks** package (formerly **snpMatrix**), which reads pre-MAKEPED pedigree files.

Usage

```
read.pedfile(file, snp.names=NULL, assign=NULL, missing=NULL,
X=FALSE, sep=".")
```

Arguments

`file`	The name of the pedigree file to be read.
`snp.names`	A character vector of SNP names. If `NULL` the function will look for SNP names in a corresponding `.map` or `.info` file with the same file name root as `file` (e.g., if `file="myfile.ped"` it will look for `myfile.map` or `myfile.info`).
`assign`	A list of named mappings from letters to alleles; not currently used.
`missing`	Missing data code in the pedigree file; not currently used.
`X`	Is the pedigree file comprised of X-chromosome SNPs? Default is `FALSE`.
`sep`	Character string to separate Family and Member IDs in the row.names of the output `snp.matrix` object; not used.

Details

JPSGCS pedigree files are in post-MAKEPED format (though much of this information is ignored):

Column 1:	Pedigree number
Column 2:	Individual ID number
Column 3:	ID of father; 0=no father in pedigree
Column 4:	ID of mother; 0=no mother in pedigree
Column 5:	First offspring ID; ignored by JPSGCS
Column 6:	Next paternal sibling ID; ignored by JPSGCS
Column 7:	Next maternal sibling ID; ignored by JPSGCS
Column 8:	Sex; 1=male, 2=female
Column 9:	Proband status (1=proband, 0=not); ignored by JPSGCS

The pre-MAKEPED format excludes columns 5 - 7. `read.pedfile` is essentially `read.snps.pedfile` for post-MAKEPED pedigree files, except that the option `low.mem` of `read.snps.pedfile` has not been implemented.

As in `read.snps.pedfile`, when reading in SNP genotype data the function looks for a corresponding `.map` or `.info` file with information on the SNPs such as physical map positions. For example, if `file="test.ped"`, the function looks for `test.map` or `test.info` to read SNP information. Any SNP information that is found is saved in the element `snp.support` of the output object (see **Value**).

Value

A list with three components:

`snp.data`	a `snp.matrix` object holding the genotypes,
`subject.support`	a data frame containing the first nine columns of the pedigree file, and
`snp.support`	a data frame of SNP information. NB: this is only meaningful if a `.map` or `.info` file was found.

Author(s)

Sigal Blay, Jinko Graham and Brad McNeney

See Also

`write.pedfile`, `read.snps.pedfile`

Examples

```
data(exdat)
sdat<-as(exdat$markers,"snp.matrix") #coerce to snp.matrix
#Write pedigree file for unrelated subjects
write.pedfile(pedinf="unrelated",snp.data=sdat,file="test.ped")
#Read it back into R
sdat2<-read.pedfile("test.ped")
#clean up
unlink("test.ped")
```

---

write.parfile *write a LINKAGE parameter file*

Description

Use a snp.matrix object and genetic map to create a LINKAGE parameter file.

Usage

```
write.parfile(snp.data, map, file="out.par")
```

Arguments

snp.data	a snp.matrix object of genotypes
map	genetic map in centiMorgans
file	file name for the output parameter file

Value

None. The output is the parameter file.

Author(s)

Jinko Graham, Brad McNeney and Annick Nembot-Simo

See Also

write.pedfile

Examples

```
data(exdat)
sdat<-as(exdat$markers,"snp.matrix") #coerce to snp.matrix
write.parfile(sdat,exdat$map,file="test.par")
unlink("test.par")
```

---

write.pedfile *write a* snp.matrix *object to a LINKAGE pedigree file.*

Description

Write a snp.matrix object to a LINKAGE pedigree file, with pedigree information followed by genotypes. This function can be used to write either post-MAKEPED or pre-MAKEPED pedigree file formats (see **Details**). In addition to standard pedigree file formats the function can optionally write transposed pedigree files, as required by FitGMLD.

Usage

```
write.pedfile(pedinf, snp.data, file, transpose=FALSE, sep=" ", eol="\n", na="0")
```

Arguments

pedinf	The pedigree information for the pedfile as a matrix or data frame. Altnerately, users may specify pedinf="unrelated" for unrelated subjects. See the **Examples**.
snp.data	A snp.matrix object containing the marker data.
file	File name for the output pedfile.

transpose	Should the pedigree file be transposed, as required by FitGMLD? Default is FALSE.
sep	Field separator (not currently used).
eol	The end-of-line character.
na	The missing data code in snp.data.

Details

JPSGCS pedigree files are in post-MAKEPED format (though much of this information is ignored):

Column 1:	Pedigree number
Column 2:	Individual ID number
Column 3:	ID of father; 0=no father in pedigree
Column 4:	ID of mother; 0=no mother in pedigree
Column 5:	First offspring ID; ignored by JPSGCS
Column 6:	Next paternal sibling ID; ignored by JPSGCS
Column 7:	Next maternal sibling ID; ignored by JPSGCS
Column 8:	Sex; 1=male, 2=female
Column 9:	Proband status (1=proband, 0=not); ignored by JPSGCS

The pre-MAKEPED format excludes columns 5 - 7.

Value

A numeric vector comprised of the number of subjects and SNPs written to the pedigree file.

Warning

The function makes **NO** attempt to check the pedigree information for correctness. It is up to the user to specify valid pedigree information.

Author(s)

Sigal Blay, Jinko Graham and Brad McNeney

See Also

read.pedfile, read.snps.pedfile

Examples

```
data(exdat)
sdat<-as(exdat$markers,"snp.matrix") #coerce to snp.matrix
#Write a post-MAKEPED pedigree file for unrelated subjects.
pedinf<-matrix(
 c( 1, 1, 0, 0, 0, 0, 0, 1, 1,
    2, 1, 0, 0, 0, 0, 0, 1, 1,
    3, 1, 0, 0, 0, 0, 0, 1, 1,
    4, 1, 0, 0, 0, 0, 0, 1, 1,
```

```
    5, 1, 0, 0, 0, 0, 0, 1, 1),
 byrow=TRUE, ncol=9, nrow=5)
write.pedfile(pedinf,sdat,file="test.ped")
#clean up
unlink("test.ped")
```

Index