

# Package ‘arp.gee’

April 6, 2005

**Version** 0.1.0

**Date** 2005-4

**Title** Affected Relative Pairs Linkage Mapping With Generalized Estimating Equations

**Author** Jason P. Sinnwell and Daniel J. Schaid

**Maintainer** Jason P. Sinnwell <sinnwell@mayo.edu>

**Description** Simultaneously estimate a trait-locus position and its genetic effects for affected relative pairs by one of two methods. Either allow a different trait-locus effect for each ARP type, or constrain the trait-locus effects according to the marginal effect of a single susceptibility locus. Includes a goodness of fit statistic for the constrained model.

**License** Copyright 2003 Mayo Foundation for Medical Education and Research. This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the GPL license, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. For other licensing arrangements, please contact Daniel J. Schaid. Daniel J. Schaid, Ph.D. Division of Biostatistics Harwick Building Room 775 Mayo Clinic 200 First St., SW Rochester, MN 55905. Phone: 507-284-0639 fax: 507-284-9542 email: schaid@mayo.edu.

**Depends** R (>= 2.0.1)

**URL** <http://mayoresearch.mayo.edu/mayo/research/biostat/schaid.cfm>

## R topics documented:

Ginv . . . . .	2
arp.fit . . . . .	3
arp.gof . . . . .	4
arp.ibd . . . . .	5
arp.ibd.fitter . . . . .	7
c.ls . . . . .	9
c2lambda . . . . .	10
derivC2lambda . . . . .	11
example.share . . . . .	12

finish.objects	12
finish.objects.default	13
finish.objects.genehunter.ibd	14
finish.objects.merlin	15
get.genehunter.ibd	16
get.object	17
ibd.df	18
ibd.df.merlin	19
ibd.peaks	20
ibd.share.genehunter	21
ibd.share.merlin	22
lambda.equal.arp	23
lambda2c	24
mergeIbdCovar	25
parse.genehunter.ibd.pl	26
plot.arp.ibd	26
plot.ibd.share	27
print.arp.ibd	28
printBanner	29
read.objects	30
reldpair.type	31
setupData	32
summary.arp.ibd	33
summary.ibd.share	34
<b>Index</b>	<b>35</b>

---

Ginv

---

*Compute Generalized Inverse of Input Matrix*


---

## Description

Singular value decomposition (svd) is used to compute a generalized inverse of input matrix.

## Usage

Ginv(x)

## Arguments

x                    A matrix.

## Details

The svd function uses the LAPACK library standard to compute the singular values of the input matrix, and the rank of the matrix is determined by the number of singular values that are at least as large as  $\max(\text{svd}) \cdot \text{eps}$ , where eps is a small value (currently  $\text{eps} = .000001$ ). For S-PLUS, the Matrix library is required.

**Value**

List with components:

Ginv	Generalized inverse of x.
rank	Rank of matrix x.

**Side Effects****References**

Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical recipes in C. The art of scientific computing. 2nd ed. Cambridge University Press, Cambridge.1992. page 61.

Anderson, E., et al. (1994). LAPACK User's Guide, 2nd edition, SIAM, Philadelphia.

**See Also**

svd

**Examples**

```
# for matrix x, extract the generalized inverse and
# rank of x as follows
#   > save <- Ginv(x)
#   > ginv.x <- save$Ginv
#   > rank.x <- save$rank
```

---

arp.fit

*Fitted values for an ARP type*

---

**Description**

Calculate fitted values for an Affected Relative Pair type, given parameters estimated from arp.ibd.

**Usage**

```
arp.fit(type, pos, tau, c.coef, eps)
```

**Arguments**

type	Integer code of ARP type. 1=FS (Full-Sibs), 2=HS (Half-Sibs), 3=FC (First-Cousins), 4=GP (GrandParent-Child), 5=AV (AVuncular pair i.e. uncle-nephew)
pos	Vector of chromosome positions, in centimorgans
tau	Estimated trait-locus position
c.coef	Locus effect coefficient for the ARP type
eps	Smoothing parameter used by Liang et al., as used in the Haldane mapping function.

**Details****Value**

A list with the following vectors

<code>s.fit</code>	Vector of fitted values of allele sharing, specific to ARP type
<code>pos</code>	Vector of chromosome positions, in centimorgans (same as input vector)

**Side Effects****References****See Also****Examples**

---

<code>arp.gof</code>	<i>Find 'r-squared' as the goodness-of-fit of an arp.ibd model</i>
----------------------	--------------------------------------------------------------------

---

**Description**

Measure goodness of fit of an arp.ibd model relative to a null model with only intercept (i.e. random sharing specific to the ARP type). Calculate an 'r-squared' measure as a simple residual sums of squares.

**Usage**

```
arp.gof(ibd.obj, model.fit)
```

**Arguments**

<code>ibd.obj</code>	An ibd.share object
<code>model.fit</code>	An arp.ibd object, as returned from the arp.ibd function.

**Details****Value**

An 'r-squared' goodness-of-fit measure for the arp.ibd model

**Side Effects****References****See Also**

[arp.ibd](#), [ibd.share.genehunter](#), [ibd.share.merlin](#), [arp.fit](#)

**Examples**


---

arp.ibd	<i>Estimate the trait locus position simultaneously with trait locus effects for Affected Relative Pairs</i>
---------	--------------------------------------------------------------------------------------------------------------

---

**Description**

Use the elements of an ibd.share object to estimate trait locus position and its effects, allowing different types of affected relative pairs (ARPs).

**Usage**

```
arp.ibd(ibd.obj, model="C", top.pct=.4, end.cut=10, near.rng=10, tau.init=NA,
        c.init=NA, lambda.init=NA, ci.prob=0.95, max.iter=50,
        tol=0.0001, eps=1, print.iter=FALSE)
```

**Arguments**

ibd.obj	An object of the 'ibd.share' class, a list which contains the following elements: \$smat - a matrix having allele sharing probabilities for affected relative pairs (rows) at each position (columns) of a chromosomal region; \$type - vector of affected relative pair types; \$ped - vector of pedigree id's; and \$pos - vector of chromosome positions (in centimorgans) matching the columns of \$smat.
model	Character string with either 'C' for C-coefficients or 'Lambda', which constrains all C's to be a function of a single lambda.
top.pct	In selecting tau starting values, only consider the positions whose mean allele sharing is in the top.pct*100 percent
end.cut	Do not select a tau starting value within end.cut centimorgans from the end
near.rng	The minimum range, in centimorgans, that a separate IBD sharing peak may be detected. All clusters of IBD sharing peaks consecutively within near.rng cM of each other, will provide one starting value for tau.
tau.init	Chromosome position initial value. If none is given, the candidate positions will be chosen based on ibd values
c.init	Vector of C-coefficients, one for each ARP type. If c.init=NA, each element is estimated from the ibd sharing data for the relevant ARP type by setting tau=tau.init and using least squares.

<code>lambda.init</code>	Initial value for lambda parameter under model='Lambda'. If not given, then the C's are initialized as described under <code>c.init</code> ; these are transformed into lambdas, and the weighted average of these is used as the initial value.
<code>ci.prob</code>	Probability level for constructing confidence intervals for the parameters.
<code>max.iter</code>	Maximum iterations allowed in Newton-Raphson steps
<code>tol</code>	Convergence tolerance for changes in scores and parameters from previous iteration
<code>eps</code>	Epsilon, smoothing adjustment constant from Liang et al.(2001), as used for adjustment within the Haldane mapping function.
<code>print.iter</code>	Logical, if TRUE, causes print of u-scores, gamma, and info-mat after each Newton-Raphson iteration.

### Details

Liang et al. [2001] proposed a multipoint method for simultaneously estimating trait-locus position and the genetic effect of a susceptibility locus when analyzing affected sib pairs. The methods in this package extend the estimation to allow for different types of affected relative pairs. The 'C' model allows unconstrained estimation of a separate C-coefficient for each ARP type. These coefficients represent the expected departure from random sharing at the trait-locus. Under the 'lambda' model, the C's are constrained to be a function of a single parameter, lambda. Lambda represents the ratio of risk for a relative who shares one trait allele IBD with an affected person to the risk for a relative who shares no alleles IBD.

### Value

An object with class `arp.ibd` containing the following components:

<code>var</code>	Variance matrix of the parameter estimates
<code>lambda</code>	If model='C', a table of the translated C's to lambdas
<code>tbl</code>	Table of estimated parameters with confidence intervals
<code>summary.df</code>	Results for each <code>tau.init</code> starting value, including final tau estimate, C/lambda estimates, and gof statistic
<code>type</code>	The unique ARP types in the dataset. 1=FS (Full-Sibs), 2=HS (Half-Sibs), 3=FC (First-Cousins), 4=GP (GrandParent-Child), 5=AV (AVuncular pair, i.e. uncle-nephew)
<code>n.type</code>	Vector of counts per ARP type
<code>n.pair</code>	Total number of relative pairs
<code>pos</code>	Vector of chromosome positions, in centimorgans
<code>iter</code>	Number of Newton-Raphson iterations completed
<code>u.scores</code>	Vector of final score values
<code>gamma</code>	Vector of final parameter estimates: tau is the first element, followed by either C's or lambda
<code>rank</code>	Rank of variance matrix
<code>converge</code>	Convergence status of the Newton-Raphson steps. 1 if converged; -2 if either C or lambda estimates are out of range; -1 if tau is out of range, 0 if <code>iter=max.iter</code> without convergence.

alias	Vector the same length as gamma, with elements of 1 if that element of gamma is aliased, 0 otherwise. Aliased parameters are not estimable because they are 'aliased' (i.e., correlation close to one) with another parameter.
info.robust	Robust information matrix
info	Last 'working' information matrix, as described in Schaid et al.(2004).

The following elements of the arp.ibd object are unchanged from input values: max.iter, eps, ci.prob, and model.

### Side Effects

### References

Liang K-Y, Chiu Y-F, Beaty, TH (2001). A robust identity-by-descent procedure using affected sib pairs: Multipoint mapping for complex diseases. Hum Hered 51:64-78.

Schaid DJ, Sinnwell JP, Thibodeau SN (2005). Robust Multipoint Identical-by-Descent Mapping for Affected Relative Pairs. Am J Hum Genet 76:128-138

### See Also

[print.arp.ibd](#), [summary.arp.ibd](#), [plot.arp.ibd](#), [arp.ibd.fitter](#), [c.ls](#)

### Examples

```
setupData(example.share)

# Fit the model-C, using defaults for optional parameters.
fit.c <- arp.ibd(example.share, model="C")
print(fit.c)

# Fit the model-Lambda, using defaults for optional parameters
fit.lambda <- arp.ibd(example.share, model="lambda")
print(fit.lambda)

# plot the fitted values from fit.c, compare against mean
# ibd sharing values from example.share
# plot(fit.c, example.share)

# plot the same for fit.lambda
# plot(fit.lambda, example.share)
```

---

arp.ibd.fitter      *For internal use within arp.ibd.*

---

### Description

For internal use within arp.ibd.

**Usage**

```
arp.ibd.fitter(ibd.obj, model="C", tau.init=NA, c.init=NA, lambda.init=NA,
ci.prob=0.95, max.iter=50, tol=0.0001, eps=1, print.iter=FALSE)
```

**Arguments**

<code>ibd.obj</code>	An object of the 'ibd.share' class, a list which contains the following elements: \$smat - a matrix having allele sharing probabilities for affected relative pairs (rows) at each position (columns) of a chromosomal region; \$type - vector of affected relative pair types; \$ped - vector of pedigree id's; and \$pos - vector of chromosome positions (in centimorgans) matching the columns of \$smat.
<code>model</code>	Character string with either 'C' for C-coefficients or 'Lambda', which constrains all C's to be a function of a single lambda.
<code>tau.init</code>	Chromosome position initial value. If none is given, the default is the position with the largest mean allele sharing.
<code>c.init</code>	Vector of C-coefficients, one for each ARP type. If c.init=NA, each element is estimated from the ibd sharing data for the relevant ARP type by setting tau=tau.init and using least squares.
<code>lambda.init</code>	Initial value for lambda parameter under model='Lambda'. If not given, then the C's are initialized as described under c.init; these are transformed into lambdas, and the weighted average of these is used as the initial value.
<code>ci.prob</code>	Probability level for constructing confidence intervals for the parameters.
<code>max.iter</code>	Maximum iterations allowed in Newton-Raphson steps
<code>tol</code>	Convergence tolerance for changes in scores and parameters from previous iteration
<code>eps</code>	Epsilon, smoothing adjustment constant from Liang et al.(2001), as used for adjustment within the Haldane mapping function.
<code>print.iter</code>	Logical, if TRUE, causes print of u-scores, gamma, and info-mat after each Newton-Raphson iteration.

**Details****Value**

A list to be made into an arp.ibd object by the arp.ibd function.

**Side Effects****References**

Liang K-Y, Chiu Y-F, Beaty, TH (2001). A robust identity-by-descent procedure using affected sib pairs: Multipoint mapping for complex diseases. Hum Hered 51:64-78.

Schaid DJ, Sinnwell JP, Thibodeau SN (2004). Robust Multipoint Identical-by-Descent Linkage Mapping with Affected Relative Pairs. Submitted.

**See Also**[arp.ibd](#)**Examples**

---

`c.ls`*Estimate C-coefficient for an ARP type by least squares*

---

**Description**

For a specified type of ARP and a fixed position tau, estimate the C-coefficient by least squares.

**Usage**

```
c.ls(type, smat, pos, tau, eps)
```

**Arguments**

type	Integer code of ARP type. 1=FS (Full-Sibs), 2=HS (Half-Sibs), 3=FC (First-Cousins), 4=GP (GrandParent-Child), 5=AV (AVuncular pair, i.e. uncle-nephew)
smat	A matrix with allele sharing probabilities for affected relative pairs (rows) at each position (columns) of a chromosomal region.
pos	Vector of chromosome positions (in centimorgans) matching the columns of smat
tau	Chromosome position for the trait-locus.
eps	Epsilon, smoothing adjustment constant from Liang et al.(2001), as used for adjustment within the Haldane mapping function.

**Details****Value**

Estimated C-coefficient

**Side Effects****References****See Also**[arp.ibd](#)**Examples**

c2lambda

*Convert C-coefficient to lambda for a specified type of ARP*

---

**Description**

Convert C-coefficient to lambda for a specified type of ARP

**Usage**

```
c2lambda(type, c)
```

**Arguments**

type	Integer code of ARP type. 1=FS (Full-Sibs), 2=HS (Half-Sibs), 3=FC (First-Cousins), 4=GP (GrandParent-Child), 5=AV (AVuncular pair i.e. uncle-nephew)
c	Coefficient for trait locus effect

**Details****Value**

lambda

**Side Effects****References****See Also****Examples**

---

derivC2lambda	<i>Derivative of lambda with respect to the C-coefficient</i>
---------------	---------------------------------------------------------------

---

**Description**

Derivative of lambda with respect to the C-coefficient

**Usage**

```
derivC2lambda(type, c)
```

**Arguments**

type	Integer code of ARP type. 1=FS (Full-Sibs), 2=HS (Half-Sibs), 3=FC (First-Cousins), 4=GP (GrandParent-Child), 5=AV (AVuncular pair i.e. uncle-nephew)
c	Coefficient of trait locus effect

**Details****Value**

Derivative of lambda with respect to C-coefficient

**Side Effects****References****See Also****Examples**

---

`example.share`*Example IBD sharing values for Affected Relative Pairs*

---

**Description**

Example IBD sharing values for Affected Relative Pairs. The data comes from a Prostate Cancer study with medium to large pedigrees. The IBD values are for individuals of 3 common ARP types on one particular chromosome.

**Usage**

```
data(example.share)
```

**Format**

`smat` Matrix of estimated IBD sharing values, arranged by ARP type (rows) and chromosome position (columns). Rows are sorted by ARP type and then pedigree.

`ped` Pedigree code for all ARPs, elements correspond to rows in `smat`. Many relative pairs from a pedigree may be included, but they don't appear together in `smat`. 82 pedigrees are in this data set.

`type` The ARP type, as a factor, of all ARPs in the order they are stored in `smat`. FS: 206; FC: 112; AV: 26.

`pos` 61 chromosome positions, measured in centimorgans from one end. They serve as column names of `smat`.

**References****Source**

Dataset kindly provided by S. N. Thibodeau.

---

`finish.objects`*Finish making objects that have been read by read.objects*

---

**Description**

Generic Method: Finish making objects that have been processed from data files by perl scripts and read by `read.objects`.

**Usage**

```
finish.objects(obj)
```

**Arguments**

obj            An object returned from read.objects. The class of the object determines which finish.object method gets invoked.

**Details**

**Value**

**Side Effects**

**References**

**See Also**

**Examples**

---

`finish.objects.default`

*finish.object is to finish making objects that have been read by read.objects*

---

**Description**

Generic Method: Finish making objects that have been processed from data files by perl scripts and read by read.objects. The default is to state no method was found for the object's class.

**Usage**

`finish.objects.default(obj)`

**Arguments**

obj            An object returned from read.objects. The class of the object determines which finish.object method gets invoked.

**Details**

**Value****Side Effects****References****See Also****Examples**

---

```
finish.objects.genehunter.ibd
```

*Method to finish an object of class genehunter.ibd, as read by read.objects*

---

**Description**

Method to finish an object of class genehunter.ibd, as read by read.objects

**Usage**

```
finish.objects.genehunter.ibd(obj)
```

**Arguments**

obj	An object returned from read.objects. The object contains Genehunter ibd sharing values, along with pedigree and relative pair information.
-----	---------------------------------------------------------------------------------------------------------------------------------------------

**Details****Value**

The returned object is of class ibd.df, which represents an "ibd data frame", where the ibd sharing values are stored in matrices.

**Side Effects****References**

**See Also**

[ibd.df](#)

**Examples**

---

```
finish.objects.merlin
```

*Method to finish an object of class merlin, as read by read.objects*

---

**Description**

Method to finish an object of class merlin, as read by read.objects

**Usage**

```
finish.objects.merlin(obj)
```

**Arguments**

`obj` An object returned from read.objects. The object contains Merlin ibd sharing values, along with pedigree and relative pair information.

**Details**

**Value**

**Side Effects**

**References**

**See Also**

[ibd.df](#)

**Examples**

---

get.genehunter.ibd *Compute IBD matrices by Genehunter and read into S*

---

### Description

Genehunter is used to compute the IBD sharing probability matrices, and the results are read into S, and converted to a class of type "ibd.df", an ibd data frame.

### Usage

```
get.genehunter.ibd(file.pre, file.par, mapfun="kosambi", file.ibd=NULL)
```

### Arguments

file.pre	A pedigree file in the LINKAGE format, prior to use of makeped (containing ped ID, person ID, father ID, mother ID, sex, affection status, liability class (optional), and pairs of columns for genetic markers).
file.par	A data file in the LINKAGE format (containing genetic map, penetrance, allele frequencies, etc.).
mapfun	Map function used by Genehunter. The default is "kosambi", and "haldane" is the other option. Because partial matching is used, the user need only specify mapfun to the minimum number of unique characters ("k" for kosambi and "h" for haldane).
file.ibd	The name of the file for saving Genehunter ibd probabilities. If missing as an argument, no file is saved.

### Details

The function removes any old files existing in the directory, dumps a file that is used to direct the steps of Genehunter analyses, runs Genehunter by batch, calls a Perl script that parses the output from Genehunter into a format that can be easily read into S-PLUS, and then calls get.object (a general purpose routine that reads in special tagged files, and uses a finishing function to complete the formatting of input).

### Value

An object of class "ibd.df". This is a data frame, with the number of rows the number of pairs of relatives. However, the matrices for IBD probabilities are inserted as matrices (via model.matrix), so that all columns of an IBD matrix stay together. The items in this data frame are:

ped	pedigree ID
per1	person ID for one person of the pair
per2	person ID for other person of the pair
pair.type	a factor describing the type of pair, according to the prior probabilities of linkage. The label for a pair type is "prior0, prior1, prior2", the three sharing probabilities. This factor can be used to subset according to type of pair (e.g., df[df\$pair.type=1,]), and table(pair.type) gives a count of the number of different types of relative pairs.
prior0, prior1, prior2	matrices of the prior probabilities that the pairs share 0, 1, or 2 alleles IBD.

```
post0, post1, post2
```

matrices of the posterior probabilities that the pairs share 0, 1, or 2 alleles IBD, given the genetic marker data.

### Side Effects

### References

### See Also

get.object

### Examples

```
# the general usage is as follows
# myIBD <- get.genehunter.ibd("myFile.pre", "myFile.par",mapfun="kos")
```

---

get.object	<i>Retrieve an object from a data file</i>
------------	--------------------------------------------

---

### Description

Parse a data file using PERL, read the data into into the session, then create a class for the object

### Usage

```
get.object(file, perlscript)
```

### Arguments

file	Character string giving the name and full path of the file from where data is read. The data file was created by an external software package, and is in a format not readable into this session without the perl script.
perlscript	The name of an object that contains the perl script to make a usable data set from the data file. The script became available as a vector of character strings within this session when this library was attached.

### Details

The function first exports the perl script into an external file, executes the external perl script to process the data file, then reads the resulting file using the read.objects function. This will only work for unix-like systems.

### Value

Depending upon the software that created the data file, an object with a given class, now with the retrieved data.

**Side Effects****References****See Also**

[read.objects](#)

**Examples**

---

<code>ibd.df</code>	<i>Create and return a dataframe of ped, per1, per2, and ibd information</i>
---------------------	------------------------------------------------------------------------------

---

**Description**

Create and return a dataframe of ped, per1, per2, and ibd information

**Usage**

```
ibd.df(ibd.dat)
```

**Arguments**

`ibd.dat` A data.frame as created within the `finish.object.genehunter.ibd` function

**Details****Value**

The returned object is of class `ibd.df`, which represents an "ibd data frame", where the ibd sharing values are stored in matrices.

**Side Effects****References****See Also**

[finish.objects.genehunter.ibd](#)

**Examples**

---

<code>ibd.df.merlin</code>	<i>Create and return a dataframe of ped, per1, per2, and ibd information from merlin</i>
----------------------------	------------------------------------------------------------------------------------------

---

## Description

Create and return a dataframe of ped, per1, per2, and ibd information from merlin

## Usage

```
ibd.df.merlin(ibd.dat)
```

## Arguments

`ibd.dat`      `data.frame`

## Details

## Value

An `ibd.df` object.

## Side Effects

## References

## See Also

## Examples

---

 ibd.peaks
 

---



---

*Find positions corresponding to ibd sharing peaks*


---

**Description**

Find positions corresponding to ibd sharing peaks

**Usage**

```
ibd.peaks(pos, ibd, top.pct=0.4, end.cut=10, near.rng=10)
```

**Arguments**

pos	Vector of chromosome positions, in centimorgans
ibd	Vector of mean ibd sharing values
top.pct	In selecting tau starting values, only consider the positions whose mean allele sharing is in the top.pct*100 percent
end.cut	Do not select a tau starting value within end.cut centimorgans from the end
near.rng	The minimum range, in centimorgans, that a separate IBD sharing peak may be detected. All clusters of IBD sharing peaks consecutively within near.rng cM of each other, will provide one starting value for tau.

**Details**

Evaluate the best candidate positions for ibd sharing peaks by three criteria. 1) Select candidates from points in the top top.pct\*100 percent of the ibd range. 2) Exclude positions within end.cut cM of the ends, as the estimation of a peak in those regions is unstable. 3) For all positions meeting the first two criteria, of the ibd sharing peaks approximated as having 2nd derivatives less than zero, find clusters of those which are consecutively within near.rng centimorgans of each other. Choose the point within the cluster with the highest ibd sharing peak.

**Value**

If peaks found, a vector of positions (pos) where those peaks occur. NA otherwise.

**Side Effects****References****See Also**

[arp.ibd](#)

**Examples**

---

`ibd.share.genehunter`*Make an ibd.share object from Genehunter data*

---

**Description**

Make an ibd.share object from Genehunter data

**Usage**

```
ibd.share.genehunter(ibd.file, pre.file, min.pairs=20)
```

**Arguments**

<code>ibd.file</code>	An IBD file from Genehunter
<code>pre.file</code>	File containing pedigree information and marker data (Linkage pre-made ped format)
<code>min.pairs</code>	The minimum number of pairs to be retained in the returned value. If the number of pairs of relatives of a particular type is less than <code>min.pairs</code> , all pairs of that type are excluded from the returned value. This is used to avoid fitting models to sparse data.

**Details****Value**

An ibd.share object

**Side Effects****References****See Also**

[ibd.share.merlin](#)

**Examples**

---

`ibd.share.merlin`    *Make an ibd.share object from Merlin data*

---

### Description

Make an ibd.share object from Merlin data

### Usage

```
ibd.share.merlin(ibd.file, pre.file, min.pairs=20)
```

### Arguments

<code>ibd.file</code>	A Merlin IBD output file
<code>pre.file</code>	File containing pedigree information and marker data (Linkage pre-made ped format)
<code>min.pairs</code>	The minimum number of pairs to be retained in the returned value. If the number of pairs of relatives of a particular type is less than <code>min.pairs</code> , all pairs of that type are excluded from the returned value. This is used to avoid fitting models to sparse data.

### Details

### Value

An ibd.share object

### Side Effects

### References

### See Also

[ibd.share.genehunter](#)

### Examples

---

lambda.equal.arp     *Score statistic to test equal lambda's for different type of ARP's*

---

### Description

Compute a robust score statistic to test whether the lambda's are equal across different types of ARP's

### Usage

```
lambda.equal.arp(ibd.obj, fit.lambda)
```

### Arguments

ibd.obj	An object of the 'ibd.share' class, a list which contains the following elements: \$smat - a matrix having allele sharing probabilities for affected relative pairs (rows) at each position (columns) of a chromosomal region; \$type - vector of affected relative pair types; \$ped - vector of pedigree id's; and \$pos - vector of chromosome positions (in centimorgans) matching the columns of \$smat.
fit.lambda	An object of class arp.ibd resulting from the function arp.ibd with model="Lambda" (e.g. the constrained model). constrained model).

### Details

Use both the input (ibd.obj) and output (fit.lambda) of the function arp.ibd, with the constrained model="Lambda", to compute the score statistic to test if the lambda's are equal across the different types of ARP's.

### Value

stat	stat = chi-square statistic
df	df = degrees-of-freedom
pval	pval = p-value

### Side Effects

### References

### See Also

[arp.ibd](#)

**Examples**

```
# load an ibd.share object, fit the lambda model, test it for equality
# C coeffs. lambda.equal.arp runs C model to compare against.
setupData(example.share)
fit.lambda <- arp.ibd(example.share, model="lambda")
stat.equal <- lambda.equal.arp(example.share, fit.lambda)
unlist(stat.equal)
```

---

`lambda2c`*Convert lambda to C-coefficient for a specified type of ARP*

---

**Description**

Convert lambda to C-coefficient for a specified type of ARP

**Usage**

```
lambda2c(type, lambda)
```

**Arguments**

<code>type</code>	Integer code of ARP type. 1=FS (Full-Sibs), 2=HS (Half-Sibs), 3=FC (First-Cousins), 4=GP (GrandParent-Child), 5=AV (AVuncular pair i.e. uncle-nephew)
<code>lambda</code>	Lambda for trait-locus effect

**Details****Value**

C-coefficient

**Side Effects****References****See Also****Examples**

---

mergeIbdCovar	<i>Merge ibd and pedigree information</i>
---------------	-------------------------------------------

---

**Description**

Merge ibd sharing values and pedigree information on pedigree and person id's

**Usage**

```
mergeIbdCovar(ibd.dat, covar.dat)
```

**Arguments**

<code>ibd.dat</code>	A data frame with pedigree id, person-1 id, person-2 id, and the estimated ibd sharing values at chromosome position for the two people.
<code>covar.dat</code>	Pedigree information from the .pre file

**Details****Value**

A data frame

**Side Effects****References****See Also****Examples**

---

```
parse.genehunter.ibd.pl
```

*Lines of a perl script used to process results from Genehunter or Merlin.*

---

### Description

Lines of a perl script printed to an external file, then used to process results from Genehunter or Merlin. The data is used to make a genehunter or merlin object with IBD data

### References

### Source

---

```
plot.arp.ibd
```

*Plot results of an arp.ibd object.*

---

### Description

Plot fitted IBD allele sharing values for affected relative pairs. If '...' includes an object of the ibd.share class, the observed average IBD sharing values will be plotted as dashed lines.

### Usage

```
plot.arp.ibd(x, ...)
```

### Arguments

x	An object with class arp.ibd
...	Extra plot parameters. This allows an object of ibd.share class to be passed to the function.

### Details

Plots for different relative pair types are stacked vertically in the plot area, which allows for easy comparison of different type of ARPs.

### Value

### Side Effects

## References

## See Also

[arp.ibd](#), [plot.ibd.share](#)

## Examples

```
setupData(example.share)

fit.c <- arp.ibd(example.share, model="C")

plot(fit.c)
plot(fit.c, example.share)
```

---

plot.ibd.share	<i>Plot mean ibd share values for an ibd.share object</i>
----------------	-----------------------------------------------------------

---

## Description

## Usage

```
plot.ibd.share(x, ...)
```

## Arguments

x	An ibd.share object
...	Optional plot method parameters

## Details

If an ibd.share object is passed to plot.arp.ibd, the ibd.share object will be plotted with the arp.ibd object, showing the fit of the model against the IBD sharing means.

## Value

Nothing is returned

## Side Effects

## References

## See Also

[ibd.share.genehunter](#), [ibd.share.merlin](#), [plot.arp.ibd](#)

**Examples**

```
# load ibd.share object, then call plot on it
setupData(example.share)
plot(example.share)
```

---

```
print.arp.ibd      Print the results of an arp.ibd object
```

---

**Description**

Print the estimates of the trait locus ( $\tau$ ) and locus effect coefficient(s) ( $c$  or  $\lambda$ ) with their confidence intervals. If `model=C`, also print  $c$ -coefficients translated to  $\lambda$ s.

**Usage**

```
print.arp.ibd(x, digits = max(options()$digits - 4, 5), ...)
```

**Arguments**

<code>x</code>	An object having <code>arp.ibd</code> class, computed by the <code>arp.ibd</code> function.
<code>digits</code>	Number of significant digits to print in non-integer results.
<code>...</code>	Additional print options.

**Details****Value**

Nothing is returned.

**Side Effects****References****See Also****Examples**

---

printBanner	<i>Print a nice banner</i>
-------------	----------------------------

---

## Description

## Usage

```
printBanner(str, banner.width=options()$width, char.perline=.75*banner.width, bo
```

## Arguments

str	character string - a title within the banner
banner.width	width of banner, the default is set to fit current options
char.perline	number of characters per line for the title, the default is 75% of the banner.width parameter
border	type of character for the border

## Details

This function prints a nice banner in both R and S-PLUS

## Value

## Side Effects

## References

## See Also

options

## Examples

```
printBanner("This is a pretty banner", banner.width=40, char.perline=30)

# the output looks like this:
# =====
#           This is a pretty banner
# =====
```

---

read.objects	<i>Read an s-tagged file and create an oldClass object</i>
--------------	------------------------------------------------------------

---

### Description

Read an external text file with S 'tags' to create a list, and then convert the list to the appropriate oldClass, determined by the tag "s.class".

### Usage

```
read.objects(file)
```

### Arguments

`file` Character string of an external file name.

### Details

The external text file can contain three types of tags: s.class, s.vector, and s.matrix. The syntax provides S-PLUS with information on how to create a list, with items that can be vectors and matrices, and then (if desired) convert this list to a user-defined oldClass. The syntax for these tags is provided below (note that in the text file, the colons ":" should not appear; they only appear in this help documentation to separate syntax from definitions).

### Value

If the tag 's.class' is in the external file, then the returned object is the oldClass defined by the 'name' following s.class (e.g., "s.class myclass" creates an object with class 'myclass'), otherwise the returned object is a list.

### Side Effects

#### **S.class name**

name is a character string that defines the class to be created within S-PLUS.

#### **S.vector name n v[1] v[2] ...**

name is a character string that names the vector; n is the length of the vector; v[1] v[2] ... are the items of the vector.

#### **S.matrix name nr nc col[1] col[2] ...**

name is a character string that names the matrix; nr the number of rows in the matrix; nc the number of columns in the matrix; col[1] col[2] ... the column labels. After this line of text follows nr rows of data that are used to fill the matrix (each row must have nc columns).

### See Also

finish.objects, finish.objects.default, finish.objects.genehunter

## Examples

---

relpair.type      *Determine relative pair types*

---

## Description

Based on pedigree and each person's parents, determine the relative pair type status.

## Usage

```
relpair.type(ped, p1, p2)
```

## Arguments

ped	Pedigree vector, contains parent ids of all pedigree members
p1	Relative pair, id for person 1
p2	Relative pair, id for person 2

## Details

Used in ibd.share.genehunter and ibd.share.merlin

## Value

Affected Relative Pair type for persons p1 and p2

## Side Effects

## References

## See Also

## Examples

---

`setupData`*Set up an example dataset provided within the library.*

---

**Description**

This function defines an alias function to run exactly as `data()` in R and does nothing in Splus. R keeps a data set within the working data frame, so we only want to load data it when calling an example. Splus keeps it in the background, so it is already loaded upon `library(mypkg)`.

**Usage**

```
setupData(...)
```

**Arguments**

... The name of a dataset provided within the Splus/R library.

**Details****Value****Side Effects****References****See Also****Examples**

```
## for a data set named my.data load it by
# setupData(my.data)

## check the names of my.data to see if it is loaded
# names(my.data)
```

---

summary.arp.ibd	<i>Summarize the results of an arp.ibd object, including details of the Newton-Raphson estimation process and parameter estimates.</i>
-----------------	----------------------------------------------------------------------------------------------------------------------------------------

---

### Description

Print a summary of an arp.ibd object, including details of the Newton-Raphson estimation process and parameter estimates.

### Usage

```
summary.arp.ibd(object, digits = max(options()$digits - 4, 5), ...)
```

### Arguments

object	An object of class arp.ibd
digits	Number of significant digits to include in numeric values
...	Extra parameters for the summary method.

### Details

### Value

No return value.

### Side Effects

### References

### See Also

[arp.ibd](#), [print.arp.ibd](#)

### Examples

---

summary.ibd.share *Summary of an ibd.share object*

---

**Description**

Print a summary of an ibd.share object

**Usage**

```
summary.ibd.share(object, ...)
```

**Arguments**

object	An ibd.share object
...	Optional parameters for the summary method

**Details****Value**

Nothing is returned

**Side Effects****References****See Also**

[ibd.share.genehunter](#), [ibd.share.merlin](#)

**Examples**

# Index

## \*Topic **datasets**

example.share, 12  
parse.genehunter.ibd.pl, 26

arp.fit, 3, 4  
arp.gof, 4  
arp.ibd, 4, 5, 8, 9, 20, 23, 27, 33  
arp.ibd.fitter, 7, 7

c.ls, 7, 9  
c2lambda, 10

derivC2lambda, 11

example.share, 12

finish.objects, 12  
finish.objects.default, 13  
finish.objects.genehunter.ibd,  
14, 18  
finish.objects.merlin, 15

get.genehunter.ibd, 16  
get.object, 17  
Ginv, 2

ibd.df, 15, 18  
ibd.df.merlin, 19  
ibd.peaks, 20  
ibd.share.genehunter, 4, 21, 22, 27,  
34  
ibd.share.merlin, 4, 21, 22, 27, 34

lambda.equal.arp, 23  
lambda2c, 24

mergeIbdCovar, 25

parse.genehunter.ibd.pl, 26  
parse.merlin.pl  
(parse.genehunter.ibd.pl),  
26

plot.arp.ibd, 7, 26, 27  
plot.ibd.share, 27, 27  
print.arp.ibd, 7, 28, 33

printBanner, 29

read.objects, 18, 30  
relpair.type, 31

setupData, 32  
summary.arp.ibd, 7, 33  
summary.ibd.share, 34