

Package ‘Haplin’

January 27, 2012

Type Package

Title Analyzing case-parent triad and/or case-control data with SNP haplotypes

Version 4.0

Date 2012-01-26

Author Hakon K. Gjessing

Maintainer Hakon K. Gjessing <hakon.gjessing@fhi.no>

Depends MASS, mgcv, filehash, snowfall, GenABEL, DatABEL, tools

Description Haplin performs a genetic association analysis of case-parent triad (trio) data with multiple markers. It can also incorporate complete or incomplete control triads, for instance independent control children. Estimation is based on haplotypes, for instance SNP haplotypes, even though phase is not known from the genetic data. Haplin estimates relative risk (RR + conf.int.) and p-value associated with each haplotype. It uses maximum likelihood estimation to make optimal use of data from triads with missing genotypic data, for instance if some SNPs has not been typed for some individuals. Haplin also allows estimation of effects of maternal haplotypes, particularly appropriate in perinatal epidemiology. Analyses of GWAS data with parallel processing are supported.

License GPL (>= 2)

URL <http://www.uib.no/smis/gjessing/genetics/software/haplin/>

Repository CRAN

Date/Publication 2012-01-27 11:17:42

R topics documented:

gwaaToHaplin	2
haplin	4
haplinSlide	9
haplinTDT	12
hactable	15
output	17
pedToHaplin	18
plot.haplin	20
plot.haplinTDT	22
pQQ	23
prepPed	24
print.haplin	26
print.summary.haplin	27
suest	28
summary.haplin	30
summary.haplinTDT	31
toDataFrame	32
Index	34

gwaaToHaplin	<i>Convert a loaded GenABEL gwaa.data object to a Haplin format character matrix</i>
--------------	--

Description

Mostly for internal use. Can be useful as a check that a loaded GenABEL data object converts correctly to a Haplin format; otherwise not needed by the user.

Usage

```
gwaaToHaplin(data, pedIndex, design = "triad")
```

Arguments

data	Same as data argument in haplin.
pedIndex	Same as pedIndex argument in haplin.
design	Same as design argument in haplin.

Details

To use Haplin on a large ped-format file, it should first be converted to a GenABEL raw file and loaded into R. See the documentation for haplin how this is done. Once a GenABEL object has been loaded into R, it can be fed directly to Haplin, and the internal conversion to Haplin format is done by gwaaToHaplin. The user might want to check the conversion and can apply gwaaToHaplin to the loaded data object to see the Haplin data matrix. Apart from that, the user has little need for gwaaToHaplin.

Value

A data matrix (character) with one row for each family. The column names should be more or less self-explanatory. "m", "f", and "c" are used to names columns belonging to the mother, the father, and the child, respectively. Alleles are in separate columns.

Warning

Do not try to convert a large GenABEL object directly. You should rather select a small subset of markers for inspection.

Author(s)

Hakon K. Gjessing
Professor of Biostatistics
Division of Epidemiology
Norwegian Institute of Public Health
<hakon.gjessing@fhi.no>

References

Gjessing HK and Lie RT. Case-parent triads: Estimating single- and double-dose effects of fetal and maternal disease gene haplotypes. *Annals of Human Genetics* (2006) 70, pp. 382-396.

Web Site: <http://www.uib.no/smis/gjessing/genetics/software/haplin/>

See Also

[haplin](#), [prepPed](#), [convert.snp.ped](#), [load.gwaa.data](#)

Examples

```
## Not run:

## Extract family and phenotype information:
prepPed(pedfile = "data/mygwas.ped", outdir = "data", create.map = T)

## Convert to raw file format:
convert.snp.ped(pedfile = "data/mygwas.ped",
  mapfile = "data/mygwas.map", outfile = "data/mygwas.raw")

## Load into R:
mygwas.data <- load.gwaa.data(phenofile = "data/mygwas.ph",
  genofile = "data/mygwas.raw")

## Convert subset of data:
temp <- gwaaToHaplin(mygwas.data[, 1:3],
  pedIndex = "data/mygwas.pedIndex")

## Inspect:
head(temp)
```

```
## End(Not run)
```

 haplin

Fitting log-linear models to case-parent triad and/or case-control data

Description

haplin fits a log-linear model to case-parent triads, case-control data, or combined (hybrid) case-parent control-parent triads or dyads. It estimates marker or haplotype frequencies, and uses the EM algorithm to reconstruct haplotypes and, if requested, impute missing genotypes. Produces an object of class haplin, which can be explored with summary, plot, and haptable.

Usage

```
haplin(filename, data, pedIndex,
  markers = "ALL", n.vars = 0, sep = " ", allele.sep = ";",
  na.strings = "NA", design = "triad", use.missing = FALSE,
  xchrom = FALSE, maternal = FALSE, test.maternal = FALSE,
  scoretest = "no", ccvar = NULL, covar = NULL, sex = NULL,
  comb.sex = "double",
  reference = "reciprocal", response = "free",
  threshold = 0.01, max.haplos = NULL, haplo.file = NULL,
  resampling = "no", max.EM.iter = 50, data.out = "no",
  verbose = TRUE, printout = TRUE)
```

Arguments

	Of the following arguments, either data or filename is required. The data argument is usually combined with the pedIndex argument. Use of the remaining arguments will depend on the type of analysis.
filename	A character string giving the name and path of the ASCII data file to be read. The file should be in the Haplin data format.
data	An R-object which is the result of using load.gwaa.data to load data into R. See the web page for a description of how to convert a ped file into a file that can be loaded. The conversion uses prepPed and convert.snp.ped.
pedIndex	A file of family indexes constructed by using prepPed on the original ped file. This file is used by Haplin to extract and store family information.
markers	Default is "ALL", which means HAPLIN uses all available markers in the data set in the analysis. For the current version of HAPLIN the number of markers used at a single run should probably not exceed 4 or 5 due to the computational burden. The markers argument can be used to select appropriate markers from the file without creating a new file for the selected markers. For instance, if

markers is set to `c(2,4)`, HAPLIN will only use the second and fourth markers supplied in the data set. When running HAPLIN, it may be a good idea to start exploring a few markers at a time, using this argument.

<code>n.vars</code>	Numeric. The number of variables (columns) in the data file before (to the left) of the genetic data.
<code>sep</code>	The character separator used in the data file to separate between "columns", where each column contains the two alleles of a single individual at a single marker.
<code>allele.sep</code>	The character separator used in the data file to separate the two alleles for a single individual in a single marker. The recommended (default) separator is ";", but for SNPs an empty "" is also common.
<code>na.strings</code>	The character string indicating missing data in the data file. Default is to use "NA" in place of, for instance, C;T for a SNP that hasn't been typed in that individual.
<code>design</code>	The value "triad" is used for the standard case triad design, without independent controls. The value "cc.triad" means a combination of case triads and control triads. This requires the argument <code>ccvar</code> to point to the data column containing the case-control variable. The value "cc" means a simple case-control design, where the parents have not been genotyped (there are no data columns for parental genes).
<code>use.missing</code>	A logical value used to determine whether triads with missing data should be included in the analysis. When set to TRUE, Haplin uses the EM algorithm to obtain risk estimates, also taking into account triads with missing data. The standard errors and p-values are adjusted to correct for this. The default, however, is FALSE. When FALSE, all triads having any sort of missing data are excluded before the analysis is run. Note that Haplin only looks at markers actually used in the analysis, so that if the markers argument (see below) is used to select a collection of markers for analysis, Haplin only excludes triads with missing data on the included markers.
<code>xchrom</code>	Logical, defaults to "FALSE". If set to "TRUE", haplin assumes the markers are on the x-chromosome. This option should be combined with specifying the sex argument, and setting (for the time being) <code>response = "mult"</code> , <code>reference = "ref.cat"</code> , <code>maternal = F</code> .
<code>maternal</code>	If TRUE, maternal effects are estimated as well as the standard fetal effects.
<code>test.maternal</code>	Not yet implemented.
<code>scoretest</code>	Special interest only. If "no", no score test is computed. If "yes", an overall score p-value is included in the output, and the individual score values are returned in the haplin object. If "only", haplin is only run under the null hypothesis, and a simple score object is returned instead of the full haplin object. Useful if only score testing is needed.
<code>ccvar</code>	Numeric. Should give the column number for the column containing the case-control indicator in the data file. Needed for the "cc" and "cc.triad" designs. The column should contain two numeric values, of which the largest one is always used to denote cases.
<code>covar</code>	Not yet implemented.

sex	To be used with <code>xchrom = TRUE</code> . A numeric value specifying which of the data columns that contains the sex variable. The variable should be coded 1 for males and 2 for females.
comb.sex	To be used with <code>xchrom = TRUE</code> . A character value that specifies how to handle gender differences on the X-chromosome. If set to "males" or "females", analyses are done either for just males or just females, respectively. If set to "single" or "double", males and females are used in a combined analysis. Specifically, when "single", the effect of a (single) allele in males is assumed to equal the effect of a single allele dose in females, and similarly, when "double", a single allele in males is assumed to have the same effect as a double allele dose in females. Default is "double", which corresponds to X-inactivation. See separate description for more details.
reference	Decides how HAPLIN chooses its reference category for the effect estimates. Default value is "reciprocal". With the reciprocal reference the effect of a single or double dose of each haplotype is measured relative to the remaining haplotypes. This means that a new reference category is used for each single haplotype. Other possible values are "population" (which is similar to reciprocal, but where the reference category is always the total population), and "ref.cat", where a single haplotype is used as reference for all the rest. For ref.cat, the default is to choose the most frequent haplotype as the reference haplotype. The reference haplotype can be set explicitly by giving a numeric value for the reference argument. Note that the numeric value refers to the haplotype's position among the haplotypes selected for analysis by HAPLIN. This means that one should run HAPLIN once first to see what haplotypes are used before giving a numeric value to reference.
response	The default value "free" means that both single- and double dose effects are estimated. Choosing "mult" instead specifies a multiplicative dose-response model.
threshold	Sets the (approximate) lower limit for the haplotype frequencies of those haplotypes that should be retained in the analysis. Haplotypes that are less frequent are removed, and information about this is given in the output.
max.haplos	Not yet implemented.
haplo.file	Not yet implemented.
resampling	Mostly for testing. Default is "no". When "no", the individual haplotypes reconstructed by the EM algorithm as assumed known when computing CIs and p-values. If set to "jackknife" a jackknife-based resampling procedure is used when computing confidence intervals and p-values for effect estimates. This takes more time, but corrects the CIs and p-values for the uncertainty contained in unphased data. Note: in all recent versions of Haplin, the resampling is no longer needed since the confidence intervals and p-values are already corrected in the standard computation.
max.EM.iter	The maximum number of iterations used by the EM algorithm. This value can be increased if necessary, which sometimes is the case with e.g. case-control data which a substantial amount of missing. However, for triad data with little missing information there is usually no need for many iterations.
data.out	Character. Accepts values "no", "prelim", "null" or "full", with "no" as default. For values other than default, haplin returns the data file prepared for analysis

rather than the usual haplin estimation results. The data file contains the haplotypes identified for each triad, and a vector of weights giving the probability distribution of different haplotype configurations within a triad. The probabilities are computed from preliminary haplotype frequency estimates, from the null model or from the full likelihood model. The "prelim" option will be much faster but somewhat less precise than the likelihood models.

verbose	Default is T (=TRUE). During the EM algorithm, HAPLIN prints the estimated parameters and deviance for each step. To avoid the output, set this argument to F (=FALSE).
printout	Logical. If TRUE (default), haplin prints a full summary of the results after finishing the estimation. If FALSE, no such printout is given, but the summary function can later be applied to a saved result to get the same summary.

Details

Input data can be supplied either as a Haplin format data file, using the `filename` argument, or as an R-object (of class `gwaa.data`) using the `data` argument. In the latter case, the data derives from a ped format file, which is converted using `prepPed` and `convert.snp.ped`. If the `data` argument is used, the arguments `filename`, `n.vars`, `sep`, `allele.sep`, `na.strings`, `ccvar`, and `sex` need not be specified.

The output can be examined by `print`, `summary`, `plot` and `haptable`.

Value

An object of class `haplin` is returned. (The only exception is when `data.out` is set different from "no", where haplin will produce a data file with haplotypes identified.)

Warning

Typically, some of the included haplotypes will be relatively rare, such as a frequency of 1% - 5%. For those haplotypes there may be too little data to estimate the double doses properly, so the estimates may be unreliable. This is seen from the extremely wide confidence intervals. The rare double dose estimates should be disregarded, but the remaining single and double dose estimates are valid. To avoid the problem one can also reduce the model to a purely multiplicative model by setting `response = "mult"` combined with `reference = "ref.cat"`.

Note

Further information is found on the web page.

Author(s)

Hakon K. Gjessing
Professor of Biostatistics
Division of Epidemiology
Norwegian Institute of Public Health
<hakon.gjessing@fhi.no>

References

Gjessing HK and Lie RT. Case-parent triads: Estimating single- and double-dose effects of fetal and maternal disease gene haplotypes. *Annals of Human Genetics* (2006) 70, pp. 382-396.

Web Site: <http://www.uib.no/smis/gjessing/genetics/software/haplin/>

See Also

[summary.haplin](#), [plot.haplin](#), [pedToHaplin](#), [haptable](#), [haplinSlide](#), [prepPed](#), [convert.snp.ped](#), [load.gwaa.data](#)

Examples

```
## Not run:

# Standard run:
haplin("data.dat")

# Specify path, estimate maternal effects:
haplin("C:/work/data.dat", maternal = T)

# Specify path, use haplotype no. 2 as reference:
haplin("C:/work/data.dat", reference = 2)

# Remove more haplotypes from estimation by increasing the threshold
# to 5%:
haplin("C:/work/data.dat", threshold = 0.05)

# Estimate maternal effects, using the most frequent haplotype as reference.
# Use all data, including triads with missing data. Select
# markers 3, 4 and 8 from the supplied data.
haplin("C:/work/data.dat", use.missing = T, maternal = T,
reference = "ref.cat", markers = c(3,4,8))
# Note: in this version of Haplin, the jackknife is
# no longer necessary since the standard errors are already corrected.

# Some examples showing how to save the Haplin result and later
# recall plot and summary results:

# Same analysis as above, saving the result in the object "result.1":
result.1 <- haplin("C:/work/data.dat", use.missing = T, maternal = T,
reference = "ref.cat", markers = c(3,4,8))

# Replot the saved result (fetal effects):
plot(result.1)

# Replot the saved result (maternal effects):
plot(result.1, plot.maternal = T)

# Print a very short summary of saved result:
result.1
```

```

# A full summary of saved result, with confidence intervals and
# p-values (the same as haplin prints when running):
summary(result.1)

# Some examples when the data file contains two covariates,
# the second is the case-control variable:

# The following standard triad run is INCORRECT since it disregards
# case status:
haplin("data.dat", use.missing = T, n.vars = 2, design = "triad")

# Combined run on "hybrid" design, correctly using both case-parent
# triads and control-parent triads:
haplin("data.dat", use.missing = T, n.vars = 2, ccvar = 2,
design = "cc.triad")

# If parent columns are not in the file, a plain case-control
# run can be used:
haplin("data.dat", use.missing = T, n.vars = 2, ccvar = 2,
design = "cc", response = "mult", reference = "ref.cat")

# An example of how to produce a data file with all possible haplotypes
# identified for each triad, together with their probability weights:
result.data <- haplin("C:/work/data.dat", use.missing = T,
markers = c(3,4,8), data.out = "prelim")
# result.data will then contain the data file, with a vector of
# probabilities (freq) computed from the preliminary haplotype
# frequencies.

## End(Not run)

```

haplinSlide

*Fitting a sliding window of log-linear models to case-parent triad
and/or case-control data*

Description

Produces a list, each element of which is an object of class `haplin`, which is the result of fitting the log-linear models to the data one "window" at a time.

Usage

```
haplinSlide(filename, data, pedIndex, markers = "ALL", winlength = 1, printout,
verbose, cpus = 1, table.output = FALSE, ...)
```

Arguments

filename	A character string giving the name and path of the ASCII data file to be read. The file should be in the Haplin data format.
data	An R-object which is the result of using <code>load.gwa.data</code> to load data into R. See the web page for a description of how to convert a ped file into a file that can be loaded. The conversion uses <code>prepPed</code> and <code>convert.snp.ped</code> .
pedIndex	A file of family indexes constructed by using <code>prepPed</code> on the original ped file. This file is used by Haplin to extract and store family information.
markers	Default is "ALL", which means haplinSlide uses all available markers in the data set in the analysis. Alternatively, the relevant markers can be specified by, for instance, <code>markers = c(1, 3:10)</code> , which would use the 10 first markers except marker 2. haplinSlide will then run haplin on a series of windows selected from the supplied markers. The <code>winlength</code> argument decides the length of the windows. See details.
winlength	Length of the sliding, overlapping windows to be run along the markers. See details.
printout	Default is FALSE. If TRUE, provides a full summary of each haplin result during the run of haplinSlide.
verbose	Same as for haplin, but defaults to FALSE to reduce output size.
cpus	haplinSlide allows parallel processing of its analyses. The <code>cpus</code> argument should preferably be set to the number of available cpu's. If set lower, it will save some capacity for other processes to run. Setting it too high should not cause any serious problems.
table.output	If TRUE, the <code>haptable</code> function will be applied to each result after estimation, greatly reducing the size of the output.
...	Remaining arguments to be used by <code>haplin</code> in each run.

Details

haplinSlide runs haplin on a series of overlapping windows of the chosen markers. Except for the `markers` and `winlength` arguments, all arguments are used exactly as in haplin itself. For instance, if `markers = c(1, 3, 4, 5, 7, 8)` and `winlength = 4`, haplinSlide will run haplin on first the markers `c(1, 3, 4, 5)`, then on `c(3, 4, 5, 7)` and finally on `c(4, 5, 7, 8)`. The results are returned in a list. The elements are named "1-3-4-5" etc., and can be extracted with, say, `summary(res[["1-3-4-5"]])` etc., where `res` is the saved result. Or the output can be examined by, for instance, using `lapply(res, summary)` and `lapply(res, plot)`.

Value

A list of objects of class haplin is returned.

Note

Further information is found on the web page.

Author(s)

Hakon K. Gjessing
Professor of Biostatistics
Division of Epidemiology
Norwegian Institute of Public Health
<hakon.gjessing@fhi.no>

References

Gjessing HK and Lie RT. Case-parent triads: Estimating single- and double-dose effects of fetal and maternal disease gene haplotypes. *Annals of Human Genetics* (2006) 70, pp. 382-396.

Web Site: <http://www.uib.no/smis/gjessing/genetics/software/haplin/>

See Also

[haplin](#), [summary.haplin](#), [plot.haplin](#), [haptable](#)

Examples

```
## Not run:  
# (Almost) all standard haplin runs can be done with haplinSlide.  
# Below is an illustration. See the haplin help page for more  
# examples.  
#  
# Analyzing the effect of fetal genes, including triads with missing data,  
# using a multiplicative response model. When winlength = 1, separate  
# markers are used. To make longer windows, winlength can be increased  
# correspondingly:  
result.1 <- haplinSlide("C:/work/data.dat", use.missing = T, response = "mult",  
reference = "ref.cat", winlength = 1)  
# Provide summary of separate results:  
lapply(result.1, summary)  
# Plot results:  
par(ask = T)  
lapply(result.1, plot)  
  
## End(Not run)
```

haplinTDT

*Transmission disequilibrium tests for case-parent triad data***Description**

Produces an object of class haplinTDT, which is the result of Transmission disequilibrium tests of data.

Usage

```
haplinTDT(filename, nsim.perm = 0, select.gender = NULL,
method = c("tdt", "hrr", "trimm"), names.marker = NULL,
use.haplotypes = FALSE, use.ambiguous = TRUE, design = "triad",
markers = "ALL", n.vars = 0, sep = " ", allele.sep = ";",
na.strings = "NA", use.missing = FALSE, xchrom = FALSE,
sex = NULL, threshold = 0.01, verbose = TRUE, printout = TRUE)
```

Arguments

Of the following arguments, only filename is required. Use of the remaining arguments will depend on the type of analysis.

filename	A character string giving the name and path of the ASCII data file to be read.
nsim.perm	Number of permutations. Default is 0, which means that haplinTDT does not do a permutation test.
select.gender	Do the analysis for a gender subset. Values: 1, 2, or NULL. 1: Male, 2: Female, NULL: All. Default is NULL.
method	A character vector containing the methods that are used in the analysis. Possible values are "tdt", "hrr" and "trimm". Default are all three.
names.marker	Marker names. Default is NULL which means that the markers are denoted 1, 2, ..., # markers.
use.haplotypes	A logical value, default is FALSE. If use.haplotypes=TRUE haplotypes corresponding to the individual markers are reconstructed by haplin using the EM algorithm. The haplotypes are then analysed as a single multiallelic marker. If use.haplotypes=FALSE, the markers are analysed individually.
use.ambiguous	A logical value, default is TRUE. If FALSE then we remove those family triads where it is ambiguous which allele is transferred to the child from a parent.
design	For the moment only the value "triad" is allowed. It is used for the standard case triad design, without independent controls.
markers	Default is "ALL", which means haplinTDT uses all available markers in the data set in the analysis. If use.haplotypes = TRUE then for the current version of haplin the number of markers used at a single run should probably not exceed 4 or 5 due to the computational burden. The markers argument can be used to select appropriate markers from the file without creating a new file for the selected markers. For instance, if markers is set to c(2,4), haplinTDT will

	only use the second and fourth markers supplied in the data set. When running haplinTDT, it may be a good idea to start exploring a few markers at a time, using this argument.
n.vars	Numeric. The number of variables (columns) in the data file before (to the left) of the genetic data.
sep	The character separator used in the data file to separate between "columns", where each column contains the two alleles of a single individual at a single marker.
allele.sep	The character separator used in the data file to separate the two alleles for a single individual in a single marker. The recommended (default) separator is ";", but for SNPs an empty "" is also common.
na.strings	The character string indicating missing data in the data file. Default is to use "NA" in place of, for instance, C:T for a SNP that hasn't been typed in that individual.
use.missing	A logical value used to determine whether triads with missing data should be included in the analysis. When set to TRUE, haplinTDT uses haplin to reconstruct the markers or haplotypes. The default, however, is FALSE. When FALSE, all triads having any sort of missing data are excluded before the analysis is run. Note that haplinTDT only looks at markers actually used in the analysis, so that if the markers argument (see below) is used to select a collection of markers for analysis, haplinTDT only excludes triads with missing data on the included markers.
xchrom	Logical, defaults to "FALSE". If set to "TRUE", haplinTDT assumes the markers are on the x-chromosome. This option should be combined with specifying the sex argument.
sex	A numeric value specifying which of the data columns that contains the sex variable. The variable should be coded 1 for males and 2 for females. To be used with xchrom = TRUE.
threshold	Sets the (approximate) lower limit for the haplotype frequencies of those haplotypes that should be retained in the analysis. Haplotypes that are less frequent are removed, and information about this is given in the output.
verbose	Default is T (=TRUE).
printout	Logical. If TRUE (default), haplinTDT prints a full summary of the results after finishing the estimation. If FALSE, no such printout is given, but the summary function can later be applied to a saved result to get the same summary.

Details

Three types of transmission disequilibrium tests (TDT) are provided:

Let t_{ij} be the number of parents transmitting allele i and not j to its child and let n be number of alleles. The standard TDT test is then defined as the sum of terms $(t_{ij} - t_{ji})^2 / (t_{ij} + t_{ji})$ for $1 \leq i < j \leq n$. This sum is asymptotically chi-squared with $n(n-1)$ degrees of freedom when the marker and disease loci are unlinked or not associated.

Let $t_{.i}$ and $t_{.j}$ be the marginal totals of t_{ij} . The Haplotype-based haplotype relative risk (HHRR) test is then defined as the sum of $(t_{.i} - t_{.j})^2 / (t_{.i} + t_{.j})$ for $i = 1, 2, \dots, n$. The HHRR test statistic is asymptotically chi-squared with $n-1$ degrees of freedom.

The Triad Multi-Marker test (TRIMM) test is only defined for diallelic markers.

If `use.ambiguous = FALSE`, then all ambiguous trios will be removed. Otherwise, the different contributions to TDT, HHRR and TRIMM are weighted with the probabilities of the different transmission configurations of alleles from parent to child. For example if the parents and the child are all heterozygous 1/2, then with probability 0.5 the mother (or father) will transfer allele 1 and not allele 2. The standard formulation of the TDT and HHRR tests correspond to having `use.ambiguous = TRUE`.

Value

An object of class `haplinTDT` is returned

Note

Further information is found on the web page.

Author(s)

Oivind Skare,
with Hakon K. Gjessing
Division of Epidemiology
Norwegian Institute of Public Health
<hakon.gjessing@fhi.no>

References

Haplotype-based haplotype relative risk (HHRR):
Terwilliger JD and Ott J. A haplotype-based 'haplotype relative risk' approach to detecting allelic associations. *Human Heredity* (1992) 42(6), pp. 337-46.

Transmission disequilibrium test (TDT):
Spielman RS, McGinnis RE and Ewens WJ. Transmission test for linkage disequilibrium: the insulin gene region and insulin-dependent diabetes mellitus (IDDM). *American Journal of Human Genetics* (1993) 52(3), pp. 506-16.

Triad multi-marker test (TRIMM):
Shi M, Umbach DM and Weinberg CR. Identification of risk-related haplotypes with the use of multiple SNPs from nuclear families. *The American Journal of Human Genetics* (2007) 81, pp. 53-66.

See Also

[summary.haplinTDT](#), [plot.haplinTDT](#)

Examples

```
## Not run:
```

```

# Standard run with permutation test:
res <- haplinTDT("data.dat", nsim.perm=1000)
# Plot the saved result:
plot(res)
# A full summary of saved result including p-values
summary(res)

# Include missing values:
res <- haplinTDT("data.dat", nsim.perm=1000, use.missing=TRUE)

## End(Not run)

```

haptable	<i>Create haplin table</i>
----------	----------------------------

Description

Create a comprehensive table of haplin output

Usage

```
haptable(object)
```

Arguments

object A haplin object, i.e. the result of running haplin.

Details

haptable extracts the most important information from a haplin object to produce a summary table. The table can then be saved with, for instance, `write.table`, making the results easily accessible to other applications. You can also use `output` to produce and save the same results.

Value

A dataframe is returned, with the following columns:

Original	Number of triads
After.rem.NA	Number of triads after removal of missing
After.rem.Mend..inc.	Number of triads after removal of Mendelian inconsistencies
After.rem.rare.haplos	Number of triads after removal of rare haplotypes
alleles	Alleles at each SNP
counts	Frequency counts for alleles
HWE.pv	P-value for deviation from HWE

haplo	Haplotypes found during estimation
pv.overall	Overall p-value
hfreq.est.	Estimated haplotype frequencies
hfreq.lower	Lower 95% CI for estimated haplotype frequencies
hfreq.upper	Upper 95% CI for estimated haplotype frequencies
reference	Reference method. If ref.cat is used, the reference category is labeled "ref"
RR.est.	Estimated single dose relative risk
RR.lower	Lower 95% CI for estimated single dose relative risk
RR.upper	Upper 95% CI for estimated single dose relative risk
RR.p.value	P-values for individual single dose estimates
RRdd.est.	Estimated double dose relative risk
RRdd.lower	Lower 95% CI for estimated double dose relative risk
RRdd.upper	Upper 95% CI for estimated double dose relative risk
RRdd.p.value	P-values for individual double dose estimates
RRm.est.	Estimated single dose relative risk for maternal haplotype
RRm.lower	Lower 95% CI for estimated single dose relative risk for maternal haplotype
RRm.upper	Upper 95% CI for estimated single dose relative risk for maternal haplotype
RRm.p.value	P-values for individual single dose estimates for maternal haplotype
RRmdd.est.	Estimated double dose relative risk for maternal haplotype
RRmdd.lower	Lower 95% CI for estimated double dose relative risk for maternal haplotype
RRmdd.upper	Upper 95% CI for estimated double dose relative risk for maternal haplotype
RRmdd.p.value	P-values for individual double dose estimates for maternal haplotype

Note that the results from the maternal haplotype risks will not be present unless maternal = T was used in the haplin call.

Note

Further information is found on the web page

Author(s)

Hakon K. Gjessing
 Professor of Biostatistics
 Division of Epidemiology
 Norwegian Institute of Public Health
 <hakon.gjessing@fhi.no>

References

Web Site: <http://www.uib.no/smis/gjessing/genetics/software/haplin/>

See Also

[haplin](#), [output](#)

Examples

```
## Not run:  
  
# Produce a table containing the most important output from haplin:  
res <- haplin("data.dat", use.missing = T, maternal = T)  
hactable(res)  
  
## End(Not run)
```

output

Save files with summary, table, and plot from a haplin object.

Description

Create summary tables and figure from a haplin object. Save results as separate files in a specified directory.

Usage

```
output(object, dirname, ask = T)
```

Arguments

object	A haplin object, i.e. the result of running haplin.
dirname	Text string, for instance "c:/work/haplinresults". Name of directory where results should be saved. Default is to save results in the current working directory.
ask	Logical. If TRUE (default), you will be asked before overwriting any files with the same name. If FALSE, output will overwrite without warning.

Details

After having run haplin and saved the result (in the R workspace), the output function will extract summary results, a summary table, and a plot of the results and save them to the specified directory. The filenames will be haplin_summary.txt, haplin_table.txt and haplin_plot.jpg, respectively. output simply uses the available functions summary, hactable, and plot to produce the files, but is a quick way of saving all the relevant results.

Note

Further information is found on the web page.

Author(s)

Hakon K. Gjessing
 Professor of Biostatistics
 Division of Epidemiology
 Norwegian Institute of Public Health
 <hakon.gjessing@fhi.no>

References

Gjessing HK and Lie RT. Case-parent triads: Estimating single- and double-dose effects of fetal and maternal disease gene haplotypes. *Annals of Human Genetics* (2006) 70, pp. 382-396.

Web Site: <http://www.uib.no/smis/gjessing/genetics/software/haplin/>

See Also

[haplin](#)

Examples

```
## Not run:

# Run haplin and save results in separate files
# in the c:\work\haplinresults directory:
res <- haplin("data.dat", use.missing = T, maternal = T)
output(res, dirname = "c:/work/haplinresults")

## End(Not run)
```

pedToHaplin

Convert from ped format data to Haplin format

Description

Converts an ASCII file from a standard ped format to the Haplin format

Usage

```
pedToHaplin(indata, outdata, merge = F, na.strings = "0", sep,
  colnames.out = F)
```

Arguments

indata	A character string giving the name and path of the ASCII data file to be converted.
outdata	A character string giving the name and path for saving the converted file.

merge	If the alleles of each genotype are in two separate columns in the <code>indata</code> file, they must be merged (with the ";" separator) in the <code>outdata</code> file. This is done by setting <code>merge = TRUE</code> . Otherwise, it must be set to <code>FALSE</code> .
na.strings	The symbol used to denote missing data in <code>indata</code> . It is passed directly to R's <code>read.table</code>
sep	Column separator in <code>indata</code> . If unspecified, any white space will be used, as in <code>read.table</code> .
colnames.out	Provided just for the purpose of checking data. If <code>TRUE</code> , adds <code>colnames</code> to the <code>outdata</code> file to make it more readable. NOTE: Haplin does currently not use <code>colnames</code> , so this should be set to <code>FALSE</code> when producing the file to run on.

Details

Important: The first 6 columns should always be family id, individual id, father's id, mother's id, sex and casetype, in that order, then followed by the genetic data columns. If the genetic data columns are separated into two individual alleles, one should use the option `merge = TRUE` to merge them in the output file. If they are already joined in single columns, for instance as CT or C;T, `merge` should be set to `FALSE` (default).

Additional covariates can be included in the input file. If so, they should be placed after the 6 leading columns but before the genetic data. In this case, one should make sure the genetic data columns are already merged, and that `merge = FALSE`. (The `merge = TRUE` option when covariates are present will hopefully be implemented at some point...)

Note that the output file usually has three columns before (to the left of) the columns containing genetic data. These columns are family id, sex, and casetype. When running haplin on the output file one should specify the argument `'n.vars = 3'` in haplin. If the data are from the x chromosome the haplin arguments should also include `'sex = 2'` and `'xchrom = T'`. Similarly, if the casetype variable is a case-control indicator one should use the argument `'ccvar = 3'`. If the intention is to only run haplin on the cases the case triads should be saved separately in a new file prior to running haplin on it.

Value

The `outdata` file is written to disk. `pedToHaplin` returns (invisibly) the converted data file.

Warning

Data files come in many shapes and formats, so you should always check the output from `pedToHaplin` before using it.

Note

Further information is found on the web page.

Author(s)

Hakon K. Gjessing
Professor of Biostatistics
Division of Epidemiology

Norwegian Institute of Public Health
<hakon.gjessing@fhi.no>

References

Gjessing HK and Lie RT. Case-parent triads: Estimating single- and double-dose effects of fetal and maternal disease gene haplotypes. *Annals of Human Genetics* (2006) 70, pp. 382-396.

Web Site: <http://www.uib.no/smis/gjessing/genetics/software/haplin/>

See Also

[haplin](#)

Examples

```
## Not run:

# Standard run on supplied test file:
pedToHaplin("test_pedToHaplin.ped", outdata = "test_pedToHaplin_result.txt",
colnames.out = F, merge = T)

## End(Not run)
```

plot.haplin

Plot a haplin object

Description

Plot a haplin object and (optionally) produce picture files

Usage

```
## S3 method for class 'haplin'
plot(x, reference, separate.plots = F, filename,
filetype = "png", use.dd, ...)
```

Arguments

	Of the following arguments, only x is required.
x	A haplin object, i.e. the result of running haplin.
reference	Same as reference argument in haplin. Note that when plotting, you can only choose "reciprocal", "population" or "ref.cat". You cannot use a numeric value to change the reference category, to do that haplin must be run over again. (See the reference argument of haplin.)

separate.plots	Logical. If you estimate effects of both fetal and maternal genes you can decide whether or not to plot them in the same plot. The default is the same plot (TRUE), the alternative (FALSE) means in separate plots. If you choose separate plots you may have to set the graphics window to "recording" to make sure you can scroll back to the first plot.
filename	If you want a file containing the plot to be produced, give a character string for the filename.
filetype	The default filetype is "png", alternatively you can choose "jpeg".
use.dd	Numeric vector indicating which double dose estimates should be plotted. For instance, if set to c(1,3) only the first and third haplotypes will be drawn with double dose estimates. This is useful if some haplotypes are rare and you want to exclude the uncertain estimates from the plot.
...	Further arguments to be passed on to the plot function

Note

Further information is found on the web page.

Author(s)

Hakon K. Gjessing
Professor of Biostatistics
Division of Epidemiology
Norwegian Institute of Public Health
<hakon.gjessing@fhi.no>

References

Gjessing HK and Lie RT. Case-parent triads: Estimating single- and double-dose effects of fetal and maternal disease gene haplotypes. *Annals of Human Genetics* (2006) 70, pp. 382-396.

Web Site: <http://www.uib.no/smis/gjessing/genetics/software/haplin/>

See Also

[haplin](#)

Examples

```
## Not run:  
  
# Produce separate plots for child and mother, dump plots to files:  
res <- haplin("data.dat", use.missing = T, maternal = T)  
plot(res, separate.plots = T, filename = "Haplinres.png")  
  
## End(Not run)
```

plot.haplinTDT *Plot a haplinTDT object*

Description

Plots a haplinTDT object.

Usage

```
## S3 method for class 'haplinTDT'  
plot(x, separate.plots = FALSE, filename, filetype = "png",  
     ask = TRUE, ...)
```

Arguments

Of the following arguments, only 'x' is required.

A [haplinTDT](#) object, i.e. the result of running [haplinTDT](#).

<code>separate.plots</code>	Logical. The default is several plots on one page (FALSE), the alternative (TRUE) is one plot per page.
<code>filename</code>	If you want a file containing the plot to be produced, give a character string for the filename.
<code>filetype</code>	The default filetype is "png", alternatively you can choose "jpeg" or "pdf".
<code>ask</code>	Logical; if 'TRUE', the user is asked before each new page to be plotted.
<code>...</code>	Further arguments to be passed on to the plot function.

Author(s)

Hakon K. Gjessing (Professor of Biostatistics) and Oivind Skare
Division of Epidemiology
Norwegian Institute of Public Health
<hakon.gjessing@fhi.no>

See Also

[haplinTDT](#), [summary.haplinTDT](#)

Examples

```
## Not run:  
  
## Standard run:  
res <- haplinTDT("data.dat", nsim.perm=1000)  
plot(res)  
  
## End(Not run)
```

pQQ

QQ-plot with confidence intervals for a vector of p-values

Description

Produces a QQ-plot of p-values. The x-axis is $-\log_{10}$ of the expected p-values (under a null hypothesis of no effects), the y-axis is $-\log_{10}$ values of the actual p-values. A (pointwise) confidence interval can be drawn, and names of snps/genes corresponding to the most significant ones can be added.

Usage

```
pQQ(pvals, nlabs = 6, conf = 0.95, lim, mark = 0.05, ...)
```

Arguments

pvals	A vector of p-values.
nlabs	The number of (most significant) p-values to be labeled using names(pvals).
conf	The confidence level of the pointwise confidence band. The default is 0.95. The confidence intervals are computed under the assumption of the p-values being drawn independently from a uniform [0,1] distribution. To leave out the confidence interval, set this to FALSE.
lim	A vector of length 2 giving the plot limits (on a \log_{10} scale, for instance c(0,4)). Plot limits are computed automatically. However, if other plot limits are desirable, they can be set using this argument.
mark	By default, the 0.05 significance level is marked by lines. Can be changed to a different value, or set to FALSE.
...	Other arguments passed on to the plotting function.

Details

The pvals argument should be a vector of p-values to be plotted. If the vector has names corresponding to marker (snp) names, the plot will label some of the most significant points with the marker names.

Value

No value is returned.

Author(s)

Hakon K. Gjessing
Professor of Biostatistics
Division of Epidemiology
Norwegian Institute of Public Health
<hakon.gjessing@fhi.no>

References

Gjessing HK and Lie RT. Case-parent triads: Estimating single- and double-dose effects of fetal and maternal disease gene haplotypes. *Annals of Human Genetics* (2006) 70, pp. 382-396.

Web Site: <http://www.uib.no/smis/gjessing/genetics/software/haplin/>

prepPed	<i>Extract family and phenotype information from a ped-format file, to prepare for use in Haplin</i>
---------	--

Description

Creates a pedIndex file containing family information, a phenotype file, and optionally an “dummy” map file. The files are used by GenABEL when loading data into R, and by Haplin when converting from a GenABEL file to a Haplin file.

Usage

```
prepPed(pedfile, outdir, create.map = F)
```

Arguments

pedfile	A character string giving the name and path of the ped-format file to be used.
outdir	The directory where the pedIndex file, phenotype file, and optionally the map file should be saved.
create.map	Logical. If "TRUE", prepPed creates a dummy map file which can be used by GenABEL when loading data into R. Can be used if no map file is available.

Details

To use Haplin on a large ped-format file, it should first be converted to a GenABEL raw file and loaded into R. Since GenABEL does not retain family information available in the ped file, prepPed should first be run on the file to extract the necessary family and phenotype information. prepPed stores family information in a .pedIndex file with the same name as the ped file, and saves it in the outdir directory. Similarly, it creates a phenotype file (.ph), which contains the individual ID, the sex variable, and the case-control status. Optionally, it can construct a simple .map file, which can be used in situations where no real map file (corresponding to the ped file) is available.

The format of the ped file should be something like this:

```
1104 1104-1 1104-2 1104-3 1 1 A B B B
1104 1104-2      0      0 1 1 B B A B
1104 1104-3      0      0 2 1 A B A B
1105 1105-1 1105-2 1105-3 2 2 B B A A
1105 1105-2      0      0 1 2 B B A A
1105 1105-3      0      0 2 2 0 0 A A
```

The column values are: Family id, Individual id, Father's id, Mother's id, Sex (1 = male, 2 = female), and Case-control status (0 = controls, 1 = cases).

Column 7 and onwards contain the genotype data, with alleles in separate columns, or joined, as AB BB, etc. A "0" is used to denote missing data.

Missing values in the sex and case-control columns are not accepted.

Value

There is no useful output; the task of prepPed is to save the extracted information in the outdir directory.

Note

More details on input format, output format etc. is found in the Haplin data description on the web page.

Author(s)

Hakon K. Gjessing
Professor of Biostatistics
Division of Epidemiology
Norwegian Institute of Public Health
<hakon.gjessing@fhi.no>

References

Gjessing HK and Lie RT. Case-parent triads: Estimating single- and double-dose effects of fetal and maternal disease gene haplotypes. *Annals of Human Genetics* (2006) 70, pp. 382-396.

Web Site: <http://www.uib.no/smis/gjessing/genetics/software/haplin/>

See Also

[convert.snp.ped](#), [load.gwaa.data](#)

Examples

```
## Not run:  
  
# Create the files mygwas.pedIndex, mygwas.ph and mygwas.map in the "data" directory  
prepPed(pedfile = "data/mygwas.ped", outdir = "data", create.map = T)  
  
## End(Not run)
```

print.haplin *Print a haplin object*

Description

Print basic information about a haplin object

Usage

```
## S3 method for class 'haplin'  
print(x, ...)
```

Arguments

x A haplin object, i.e. the result of running haplin.
... Other arguments, passed on to print.

Note

Further information is found on the web page

Author(s)

Hakon K. Gjessing
Professor of Biostatistics
Division of Epidemiology
Norwegian Institute of Public Health
<hakon.gjessing@fhi.no>

References

Gjessing HK and Lie RT. Case-parent triads: Estimating single- and double-dose effects of fetal and maternal disease gene haplotypes. *Annals of Human Genetics* (2006) 70, pp. 382-396.

Web Site: <http://www.uib.no/smis/gjessing/genetics/software/haplin/>

See Also

[haplin](#)

`print.summary.haplin` *Print the summary of a haplin object*

Description

Print the result of applying summary to a haplin object

Usage

```
## S3 method for class 'summary.haplin'  
print(x, digits, ...)
```

Arguments

<code>x</code>	A haplin object, i.e. the result of running haplin.
<code>digits</code>	The number of digits to be used in the printout. Defaults to 3.
<code>...</code>	Other arguments (ignored).

Note

Further information is found on the web page

Author(s)

Hakon K. Gjessing
Professor of Biostatistics
Division of Epidemiology
Norwegian Institute of Public Health
<hakon.gjessing@fhi.no>

References

Gjessing HK and Lie RT. Case-parent triads: Estimating single- and double-dose effects of fetal and maternal disease gene haplotypes. *Annals of Human Genetics* (2006) 70, pp. 382-396.

Web Site: <http://www.uib.no/smis/gjessing/genetics/software/haplin/>

See Also

[haplin](#)

Examples

```
## Not run:  
  
# Standard summary:  
res <- haplin("data.dat", use.missing = T, maternal = T)  
summary(res)  
  
# Increase number of digits in printout  
print(summary(res), digits = 8)  
  
## End(Not run)
```

suest

Compute a joint p-value for a list of haplin fits (usually from a sliding window approach), correcting for multiple testing.

Description

The first argument to `suest` should be a list of haplin estimation results (from the same data file), usually the output from `haplinSlide`. `suest` produces as a result a joint overall p-value based on aggregating the individual p-values and then correcting for multiple testing. The correction is achieved by using the principle of "seemingly unrelated" estimation, taking into account the correlation between the individual estimation results.

Usage

```
suest(reslist)
```

Arguments

`reslist` A list whose elements are different haplin runs on the same data file, typically the output of `haplinSlide`.

Details

`haplinSlide` runs `haplin` on a series of overlapping windows of markers from the same data file, typically within the same gene. Since each run produces a separate overall p-value, `suest` computes a joint overall p-value for the gene (or region) that has been scanned. It corrects the overall p-value for multiple testing, also taking into account the fact that the sequence of estimates produced by `haplinSlide` will be dependent, both because they are computed on the same data set and also since the windows are overlapping (if the window length is larger than 1). If the `suest` estimation fails (which doesn't happen very often), a standard Bonferroni correction is used instead.

Value

A list is returned, the most important elements of which are:

<code>pval.obs</code>	The overall score p-values from each haplin run
<code>pval.obs.corr</code>	The joint p-value, corrected for multiple testing
<code>bonferroni</code>	A logical, usually FALSE, which means the suest estimation went well. If TRUE, it means that the suest estimation failed for some reason, and a standard Bonferroni correction was used instead.

Note

Further information is found on the web page.

Author(s)

Hakon K. Gjessing
Professor of Biostatistics
Division of Epidemiology
Norwegian Institute of Public Health
<hakon.gjessing@fhi.no>

References

Gjessing HK and Lie RT. Case-parent triads: Estimating single- and double-dose effects of fetal and maternal disease gene haplotypes. *Annals of Human Genetics* (2006) 70, pp. 382-396.

Web Site: <http://www.uib.no/smis/gjessing/genetics/software/haplin/>

See Also

[haplin](#), [haplinSlide](#)

Examples

```
## Not run:
# (Almost) all standard haplin runs can be done with haplinSlide.
# Below is an illustration. See the haplin help page for more
# examples.
#
# Analyzing the effect of fetal genes, including triads with missing data,
# using a multiplicative response model. When winlength = 1, separate
# markers are used. To make longer windows, winlength can be increased
# correspondingly:
result.1 <- haplinSlide("C:/work/data.dat", use.missing = T, response = "mult",
reference = "ref.cat", winlength = 1)
# Provide summary of separate results:
lapply(result.1, summary)
# Plot results:
par(ask = T)
```

```
lapply(result.1, plot)
# Compute an overall p-value for the scan, corrected for multiple testing
# and dependencies between windows:
suest(result.1)

## End(Not run)
```

summary.haplin

Summary of a haplin object

Description

Provides detailed information about estimation results from a haplin object.

Usage

```
## S3 method for class 'haplin'
summary(object, reference, ...)
```

Arguments

object	A haplin object, i.e. the result of running haplin.
reference	Same as reference argument in haplin. Note that when producing the summary, you can only choose "reciprocal", "population" or "ref.cat". You cannot use a numeric value to change the reference category, to do that haplin must be run over again. (See the reference argument of haplin.)
...	Other arguments (ignored).

Note

Further information is found on the web page

Author(s)

Hakon K. Gjessing
Professor of Biostatistics
Division of Epidemiology
Norwegian Institute of Public Health
<hakon.gjessing@fhi.no>

References

Gjessing HK and Lie RT. Case-parent triads: Estimating single- and double-dose effects of fetal and maternal disease gene haplotypes. *Annals of Human Genetics* (2006) 70, pp. 382-396.

Web Site: <http://www.uib.no/smis/gjessing/genetics/software/haplin/>

See Also[haplin](#)**Examples**

```
## Not run:

# Produce separate plots for child and mother, dump plots to files:
res <- haplin("data.dat", use.missing = T, maternal = T)
summary(res)

## End(Not run)
```

summary.haplinTDT	<i>Summary of a haplinTDT object</i>
-------------------	--------------------------------------

Description

Provides detailed information about a haplinTDT object.

Usage

```
## S3 method for class 'haplinTDT'
summary(object, ...)
```

Arguments

object	A haplinTDT object, i.e. the result of running haplinTDT.
...	Further arguments to be passed on to the summary function.

Author(s)

Hakon K. Gjessing (Professor of Biostatistics) and Oivind Skare
Division of Epidemiology
Norwegian Institute of Public Health
<hakon.gjessing@fhi.no>

See Also[haplinTDT](#), [plot.haplinTDT](#)

Examples

```
## Not run:  
  
# Standard run with permutation test:  
res <- haplinTDT("data.dat", nsim.perm=1000)  
summary(res)  
  
## End(Not run)
```

toDataFrame

Stack dataframes from haplinSlide into a single dataframe

Description

When haplinSlide is run with the option `table.output = T`, the result is a list of haptables, i.e. tables with summary haplin results for each window haplinSlide is run on. `toDataFrame` stacks the separate dataframes into one large dataframe containing all results.

Usage

```
toDataFrame(x, reduce = F)
```

Arguments

x	The output from haplinSlide run with option <code>table.output = TRUE</code> .
reduce	Reduce output to one line per marker

Details

When haplinSlide is run with `winlength = 1` on SNP markers, each table in the output has only two rows, and can be condensed to a single row. By setting the argument `reduce` to `TRUE`, `toDataFrame` reduces each table to one line and returns a dataframe with one line for each SNP. In more general situations, with multi-allelic markers or, more commonly, `winlength` set to 2 or more, each output table will typically have more than two rows and cannot be reduced, so `reduce` should be set to `FALSE`.

Value

The output is a dataframe. First column contains the marker names. Second column are row numbers, counted within each output table. The remaining columns are identical to the individual output columns, which are described in more detail in the help file for [haptable](#).

Author(s)

Hakon K. Gjessing
Professor of Biostatistics
Division of Epidemiology
Norwegian Institute of Public Health
<hakon.gjessing@fhi.no>

References

Gjessing HK and Lie RT. Case-parent triads: Estimating single- and double-dose effects of fetal and maternal disease gene haplotypes. *Annals of Human Genetics* (2006) 70, pp. 382-396.

Web Site: <http://www.uib.no/smis/gjessing/genetics/software/haplin/>

Index

`convert.snp.ped`, 3, 8, 25

`gwaaToHaplin`, 2

`haplin`, 3, 4, 10–13, 17, 18, 20, 21, 26, 27, 29, 31

`haplinSlide`, 8, 9, 28, 29

`haplinTDT`, 12, 13, 22, 31

`hactable`, 8, 11, 15, 32

`load.gwaa.data`, 3, 8, 25

`output`, 17, 17

`pedToHaplin`, 8, 18

`plot.haplin`, 8, 11, 20

`plot.haplinTDT`, 14, 22, 31

`pQQ`, 23

`prepPed`, 3, 8, 24

`print.haplin`, 26

`print.summary.haplin`, 27

`suest`, 28

`summary.haplin`, 8, 11, 30

`summary.haplinTDT`, 14, 22, 31

`toDataFrame`, 32