Package 'related'

March 22, 2015

Type Package

Title related: an R package for analyzing pairwise relatedness data based on codominant molecular markers

Version 1.0

Date 2015-03-20

Author Jack Pew, Jinliang Wang, Paul Muir, and Tim Frasier

Maintainer Tim Frasier <timothy.frasier@smu.ca>

Description Provides functions for calculating relatedness from codominant genetic data using any or all of seven estimators, and includes options for considering inbreeding and genotyping errors, and can estimate 95% confidence intervals. Also includes simulation options for comparing the performance of different estimators on a given data set and estimating the expected resolution given a particular data set. Can also test hypotheses regarding relatedness patterns within pre-defined groups.

License GPL (>= 2)

Depends R (>= 2.15.0), tools, ggplot2

R topics documented:

`related-package` *related: an R package for analyzing pairwise relatedness based on codominant molecular markers.*

Description

Provides functions for calculating relatedness from codominant genetic data using any or all of seven estimators, and includes options for considering inbreeding and genotyping errors, and can estimate 95

Details

Package:	related
Type:	Package
Version:	1.0
Date:	2015-03-20
License:	GPL (>=2)

A complete walk-through of the package and available options are available as a tutorial vignette, accessible from the CRAN website.

Important functions are summarized below.

=== READING DATA INTO R ===

Genotype data can be read into R using the `readgenotypedata` function. This will create an R data frame of the genotype data, and an R object with all of the allele frequency data, in a format appropriate for subsequent analyses. The genotype file should have one column of individual identifiers and then 2 columns for each locus (one for each allele). No other columns are allowed. Missing data should be formatted as zeros ("0"). The file should NOT contain a header row.

=== ESTIMATING RELATEDNESS ===

Pairwise relatedness can be estimated, based on 7 different estimators, using the `coancestry` function. The input is a data frame of genotypes, as generated by the `readgenotypedata` function. Relatedness estimation can include point estimates, 95% confidence intervals, inbreeding, and genotyping errors. The output will include a data frame of pairwise estimates, and separate data frames with data pertaining to confidence intervals and inbreeding coefficients, if selected.

===CONDUCTING SIMULATIONS ===

Simulations can be conducted in two different ways. First, users can generate simulated individuals of known relatedness (parent-offspring, full-sib, half-sib, and unrelated), based on their allele frequency data, to assess how much resolution they can expect with the characteristics of their data. This can be done using the `familysim` function, which takes the allele frequency object generated by the `readgenotypedata` function, as well as a whole number indicating the number of pairs to simulate for each relatedness value, as arguments. Pairwise relatedness can then be estimated for these simulated individuals using the `coancestry` function. This will generate relatedness values for *all* pairs of individuals. However, we are only interested in the relatedness of *specific pairs* (i.e., those representing known relatedness values). Therefore, these results can be cleaned up using the `cleanuprvals` function. The results will be a data frame of relatedness values of all simulated pairs of known relatedness, which can be subsequently analyzed or visualized using other R functions.

Second, users may want to know what relatedness estimator will perform best, given the characteristics of their data. We have created a function, called `compareestimators`, that will: (1) create

simulated individuals of known relatedness from your allele frequency file; (2) calculate relatedness using the 4 most commonly used estimators; and (3) create a box plot using the `ggplot2` package so that the performance of each can be readily compared.

===Analyzing Within-Group Relatedness ===

Sometimes it is desirable to test if individuals within groups are more related than expected if these just represent random groups of individuals. `related` has a function for testing this, called `grouprel`. What this function will do is calculate the average relatedness within each of the specified groups. It will then generate a distribution of 'expected' relatedness values by randomly shuffling individuals between groups, while keeping each group size constant, and calculating the average relatedness within each group for each randomization step. It will then generate plots comparing the observed and expected values. It also writes a file called "`observed-r.csv`", and another one called "`expectedrel.csv`" to your working directory, containing the observed and expected values, respectively, so that you can conduct further analyses of these data.

Author(s)

Original Fortran code: Jinliang Wang\ R code and functions: Jack Pew and Tim Frasier

Maintainer: Tim Frasier <timothy.frasier@smu.ca>

References

Pew J, Muir P, Wang J, Frasier TR (*in press*) related: an R package for analyzing pairwise relatedness data based on codominant molecular markers.

Wang J (2011) COANCESTRY: a program for simulting, estimating and analysing relatedness and inbreeding coefficients. *Molecular Ecology Resources* 11: 141-145.

See Also

`cleanuprvals`, `coancestry`, `compareestimators`, `familysim`, `grouprel`, `readgenotypedata`

Examples

```
## Not run:
data(GenotypeData)

#---Read data into R---#
input <- readgenotypedata(GenotypeData)

#---Compare relatedness estimators---#
compareestimators(input, 100)

#---Estimate relatedness---#
rel <- coancestry(input$gdata, lynchli=1, lynchrd=1, quellergt=1, wang=1)

#---Create simulated individuals of known relatedness---#
sim <- familysim(input$freqs, 100)

    #----------------------------#
   #  Within group analysis     #
#----------------------------#

#--- Read in genotype file ---#
reldata <- readgenotypedata("simrel.txt")
```

```
#--- Assess relatedness within groups ---#
grouprel(genotypes = reldata$gdata, estimatorname = "wang", usedgroups = "all", iterations = 100)


## End(Not run)
```

`cleanuprvals` *Discard non-desired pairwise relatedness values.*

Description

This function removes non-informative pairwise relatedness values from simulated data sets.

Usage

```
cleanuprvals(simdata, ninds)
```

Arguments

`simdata`	A data frame containing pairwise relatedness values for a simulated data set (generated by the `familysim` function).
`ninds`	A single value representing the number of simulated pairs that were generated for each degree of relatedness. For example, this value would be 100 if you similated 100 parent-offspring pairs, 100 full-sib pairs, 100 half-sib pairs, and 100 unrelated pairs.

Details

The `familysim` function will generate the defined number of pairs of individuals of known relatedness. However, the `coancestry` function calculates relatedness for all pairwise comparisons. We are not interested in most of these. For example, for parent-offspring we are only interested in the relatedness of individuals 1 and 2, 3 and 4, 5 and 6, and so on. We don't care about the relatedness of 1 and 3, 1 and 4, etc. To deal with this, the `cleanuprvals` function will remove the unwanted relatedness values, resulting in a data frame containing only the relatedness values of interest (those for the simulated number of pairs for each relatedness value).

Value

A data frame containing pairwise relatedness values for simulated pairs of known relatedness.

Author(s)

Tim Frasier <timothy.frasier@smu.ca>

See Also

`familysim`, `coancestry`

Examples

```
## Not run:
#---Load Genotype Data---#
data(GenotypeData)
input <- readgenotypedata(GenotypeData)

#---Create a data set of simulated individuals based on observed allele frequencies---#
sim <- familysim(input$freqs, 100)

#---Calculate pairwise relatedness---#
output <- coancestry(sim, wang = 1)

#---Clean up results to contain only desired relatedness values---#
simrel <- cleanuprvals(output$relatedness, 100)

## End(Not run)
```

`coancestry` *Calculate relatedness and inbreeding coefficients*

Description

Implements Jinliang Wang's code for Coancestry, which allows relatedness to be estimated from codominant genetic data using any of seven estimators, and includes options for considering inbreeding and genotyping errors.

Usage

```
coancestry(genotype.data, error.rates = 0, allele.freqs = NULL, trioml = 0L,
wang = 0L, lynchli = 0L, lynchrd = 0L, ritland = 0L, quellergt = 0L, dyadml = 0L,
ci95.num.bootstrap = 100L, trioml.num.reference = 100L, allow.inbreeding = FALSE,
rng.seed = NULL, working.directory = tempdir(), output.file = FALSE)
```

Arguments

`genotype.data`	A data frame containing the genotype data, preferably generated using our `readgenotypedata` function.
`error.rates`	Optional. If one error rate across all loci is assummed, use that number. If each locus has a different error rate, create a vector containing the error rate values for each locus, and refer to that vector here.
`allele.freqs`	Optional. If data were read into R using our `readgenotypedata` function, then the allele frequency object will be detected automatically. If not, then the object you created should be referred to here.
`trioml`	Optional. The triadic likelihood relatedness estimate (Wang 2007). If point estimates using this estimator are desired, enter "1", if point estimates and 95% confidence intervals are desired, enter "2".
`wang`	Optional. The relatedness estimate described in Wang (2002). If point estimates using this estimator are desired, enter "1", if point estimates and 95% confidence intervals are desired, enter "2".

`lynchli` Optional. The relatedness estimate described in Li et al. (1993). If point estimates using this estimator are desired, enter "1", if point estimates and 95% confidence intervals are desired, enter "2".

`lynchrd` Optional. The relatedness estimate described in Lynch and Ritland (1999). If point estimates using this estimator are desired, enter "1", if point estimates and 95% confidence intervals are desired, enter "2".

`ritland` Optional. The relatedness estimate described in Ritland (1996). If point estimates using this estimator are desired, enter "1", if point estimates and 95% confidence intervals are desired, enter "2".

`quellergt` Optional. The relatedness estimate described in Queller and Goodnight (1989). If point estimates using this estimator are desired, enter "1", if point estimates and 95% confidence intervals are desired, enter "2".

`dyadml` Optional. The dyadic likelihood estimator, described in Milligan (2003). If point estimates using this estimator are desired, enter "1", if point estimates and 95% confidence intervals are desired, enter "2".

`ci95.num.bootstrap`
Optional. The number of bootstrap iterations to perform to calculate 95% confidence intervals (default = 100).

`trioml.num.reference`
Optional. The triadic likelihood estimator requires that you specify the number of reference individuals to use for estimating relatedness. Enter that number here. Default = 100.

`allow.inbreeding`
Optional. A logical where inbreeding should, or should not (default), be considered when estimating relatedness.

`rng.seed` Optional. Can manually set the see of the random number generator.

`working.directory`
Optional. Can indicate what directory files are in, if not in the current directory.

`output.file` Optional. Can specify name of the output file prefix (many files will be generated - see below), but can also do this by directing output into an object using standard R commands.

Value

`relatedness` A data frame containing all pairwise estimates of relatedness. This will always have 11 columns: (1) an integer for the pair number; (2) the ID for individual #1; (3) the ID for individual #2; (4) the group assignment (see section 3.5 of accompanying vignette); and (5 - 11) for the 7 relatedness estimators - contain values of 0 for estimators not chosen

`delta7` A data frame that contains the delta7 estimates for the relatedness estimators that use it (trioml, wang, lynchrd, dyadml). This data frame contains one row for each pairwise comparison, and 8 columns: (1) an integer for the pair number; (2) the ID for individual #1; (3) the ID for individual #2; (4) the group assignment (see section 3.5 of accompanying vignette); and (5 - 8) estimates of delta7 for the 4 relevant estimators, with values of 0 for estimators not chosen.

`delta8` A data frame that contains the delta8 estimates for the relatedness estimators that use it (trioml, wang, lynchrd, dyadml). This data frame contains one row for each pairwise comparison, and 8 columns: (1) an integer for the pair number; (2) the ID for individual #1; (3) the ID for individual #2; (4) the group assignment (see section 3.5 of accompanying vignette); and (5 - 8) estimates of delta8 for the 4 relevant estimators, with values of 0 for estimators not chosen.

`inbreeding` A data frame that contains the inbreeding estimates for each individual, as used in the relatedness estimates. Only four of the relatedness estimators can account for inbreeding: dyadml, lynchrd, ritland, trioml. This data frame contains one row for each individual, and 5 columns: (1) individual ID; (2-5) inbreeding estimates for the 4 relatedness estimators. Estimators not used will have a 0 in the corresponding column.

`relatedness.ci95`
If confidence intervals are calculated. A data frame containing the low and high cut-off values for the 95% confidence interval associated with each chosen estimator. This will always have 18 columns: (1) an integer for the pair number; (2) the ID for individual #1; (3) the ID for individual #2; (4) the group assignment (see section 3.5 of accompanying vignette); (5 - 18) for the high and low values associated with each of the 7 relatedness estimators—contain values of 0 for estimators not chosen.

`delta7.ci95` If confidence intervals are calculated. A data frame that contains the low and high cut-off values for the 95% confidence interval for the delta7 estimates associated with each chosen estimator that use it (trioml, wang, lynchrd, dyadml). This will always have 12 columns: (1) an integer for the pair number; (2) the ID for individual #1; (3) the ID for individual #2; (4) the group assignment (see section 3.5 of accompanying vignette); (5 - 12) for the high and low values associated with each of the 7 relatedness estimators—contain values of 0 for estimators not chosen.

`delta8.ci95` If confidence intervals are calculated. A data frame that contains the low and high cut-off values for the 95% confidence interval for the delta8 estimates associated with each chosen estimator that use it (trioml, wang, lynchrd, dyadml). This will always have 12 columns: (1) an integer for the pair number; (2) the ID for individual #1; (3) the ID for individual #2; (4) the group assignment (see section 3.5 of accompanying vignette); (5 - 12) for the high and low values associated with each of the 7 relatedness estimators—contain values of 0 for estimators not chosen.

`inbreeding.ci95`
If confidence intervals are calculated. A data frame that contains the low and high cut-off values for the 95% confidence interval for the inbreeding estimates for each individual, as used in the relatedness estimators. Only four of the relatedness estimators can account for inbreeding: dyadml, lynchrd, ritland, trioml. This data frame contains one row for each individual, and 9 columns: (1) individual ID; (2-9) inbreeding estimates for the four relatedness estimators. Estimators not used will have a zero (0) in the corresponding column.

References

Li CC, Weeks DE, Chakravarti A (1993) Similarity of DNA fingerprints due to chance and relatedness. *Human Heredity* 43: 45-52.

Lynch M, Ritland K (1999) Estimation of pairwise relatedness with molecular markers. *Genetics* 152: 1753-1766.

Milligan BG (2003) Maximum-likelihood estimation of relatedness. *Genetics* 163: 1153-1167.

Queller DC, Goodnight KF (1989) Estimating relatedness using molecular markers. *Evolution* 43: 258-275.

Ritland (1996) Estimators of pairwise relatedness and inbreeding coefficients. *Genetical Research* 67: 175-186.

Wang J (2002) An estimator of pairwise relatedness using molecular markers. *Genetics* 160: 1203-1215.

Wang J (2007) Triadic IBD coefficients and applications to estimating pairwise relatedness. *Genetical Research* 89: 135-153.

Wang J (2011) COANCESTRY: a program for simulating, estimating and analysing relatedness and inbreeding coefficients. *Molecular Ecology Resources* 11: 141-145.

See Also

readgenotypedata

Examples

```
## Not run:
#---Read data into R---#
data(GenotypeData)
input <- readgenotypedata(GenotypeData)

#---Calculate Relatedness---#
output <- coancestry(input$gdata, lynchrd=2, quellergt=2, wang=2)

#---View Point Estimates---#
output$relatedness

#---View 95
output$relatedness.ci95

## End(Not run)
```

`compareestimators` *Compare relatedness estimators.*

Description

Generates simulated individuals of known relatedness, based on observed allele frequencies, then estimates relatedness using 4 methods (lynchli, lynchrd, quellergt, wang) and creates a boxplot of results so that the performance of the different estimators can easily be compared.

Usage

```
compareestimators(filename, ninds)
```

Arguments

`filename`	An object generated by the `readgenotypedata` function, containing genotype and allele frequency data.
`ninds`	A whole number indicating the number of pairs of individuals of known relatedness you want to generate for each relatedness value (e.g., entering 100 would result in 100 parent-offspring pairs, 100 full-sib pairs, 100 half-sib pairs, and 100 unrelated pairs.)

Value

A figure containing box plots comparing the values obtained from each estimator for each known relatedness value.

Author(s)

Tim Frasier <timothy.frasier@smu.ca>

See Also

readgenotypedata, familysim, cleanuprvals

Examples

```
## Not run:
#---Load data set---#
data(GenotypeData)
input <- readgenotypedata(GenotypeData)

#---Compare relatedness estimators---#
compareestimators(input, 100)

## End(Not run)
```

---

familysim *Simulate related individuals*

Description

Uses observed allele frequencies to generated a user-defined number of pairs of individuals of known relatedness.

Usage

```
familysim(freqs, ninds = 100L)
```

Arguments

freqs	An object containing allele frequency data. Can be generated with the readgenotypedata function.
ninds	The number of pairs of individuals to simulate for each relatedness values. For example, entering 100 would generate 100 parent-offspring pairs, 100 full-sib pairs, 100 half-sib pairs, and 100 unrelated pairs.

Value

A data frame containing the genotypes of simulated individuals.

Author(s)

Tim Frasier <timothy.frasier@smu.ca>

See Also

readgenotypedata

Examples

```
## Not run:
#---Load data set---#
data(GenotypeData)
input <- readgenotypedata(GenotypeData)

#---Generate simulated individuals---#
sim <- familysim(input$freqs, 100)

## End(Not run)
```

GenotypeData *Sample genotype data.*

Description

Example genotype data containing genotypes of 389 individuals genotyped at 10 codominant loci. It is the same file that is distributed with the Coancenstry program.

Usage

```
data(GenotypeData)
```

Format

A space-delimited file, where the data for each individual is on a separate line. Contains one column of individual identifiers.

Examples

```
data(GenotypeData)
summary(GenotypeData)
```

---

grouprel *Calculate average within-group relatedness and compare to expected values.*

Description

This function calculates average within-group relatedness and then compares these to expected values for hypothesis testing. Expected values are generated by randomly shuffling individuals between groups, while keeping each group size constant, and calculating average within-group relatedness for these "artificial" groups. Group definition is based on the first two characters of the sample/individual identifiers.

Usage

```
grouprel(genotypes, estimatorname, usedgroups, iterations)
```

Arguments

genotypes	The genotype data to use for the analyses. Note that this is *just* the genotype data, not the entire set of files generated when the data are read into R. Therefore, we have to specify this using the "$gdata" part of our original file. Can be generated by the familysim function.
estimatorname	Tells R what estimator to use. Can be: "dyadml", "lynchli", "lynchrd", "quellergt", "ritland", "tioml", or "wang". Only one estimator can be used at a time.
usedgroups	Indicates what groups to consider in the analyses. This is based on the two-character code that you have used in your individual/sample names. The DEFAULT is "all". However, you can specify specific groups by creating a list of identifiers of the desired groups. For example, with the example file we could select to analyze just the full- and half-sib groups by typing <e2><80><9c>usedgroups = c("FS", "HS")<e2><80><9d> here. You can use a similar approach for analyzing only certain groups in your data.
iterations	A single value representing the number of times to shuffle individuals and then recalculate average within-group relatedness for hypothesis testing.

Details

The function will first estimate all pairwise relatedness values within each group, and take their average. It will print these averages to the screen (but they may pass by too quickly to be seen), and will also write them to a file in R's working directory called "observed-r.csv". This file contains three columns. The first is just a list of integers, indicating how many groups were analyzed, and giving each group an incremental number. The second contains a list of each group "name"", where the name is based on the two-character code from the sample/individual identifiers. These characters will be repeated. For example, the groups are designated "FS", "HS", and "UR" in the example file, so the list here will read "FSFS", "HSHS", and "URUR". This just indicates that the values represent relatedness values between individuals designated "FS" and "FS", "HS" and "HS", and "UR" and "UR".

Next, the function will randomly shuffle the order of individuals in the genotype file, but keep the rows representing each group constant (and the same as the observed data). Thus, individuals are shuffled between groups, while keeping each group size constant. Then relatedness is calculated within each group as described above. This process is repeated however many times is designated in the command.

The results are exported to a file called "expectedrel.csv". This file will contain one column for each group in the analyses, and one row for each iteration performed (the columns are not labeled, however). Thus, each column represents the distribution of expected relatedness values within that group if group membership is random with respect to relatedness. The order of groups (i.e., the order of columns) is the same order as they are encountered in the genotype file (and as printed in the "observed-r.csv" file). The last column is the average relatedness value across all groups for each iteration. This can be used to test global patterns across your data, rather than just for each group independently. By having this as a separate file, you can explore the data on your own.

Figures are automatically generated comparing observed and expected values, as described below.

Value

One figure is generated for each analyzed group. Each figure contains: (1) a histogram of the expected relatedness values within each group; (2) a red arrow indicating where the observed value lies, for easy comparison of observed and expected values; (3) a title indicating the group comparison; and (4) a p-value, indicating the percentage of randomized iterations where the expected values were greater than or equal to the observed value (see below for more explanation). A similar figure is generated for the "overall" data set, where the observed average within-group relatedness across all groups is compared to the expected values. Thus, this function simultaneously assess whether or not relatedness within each group is higher than expected, and whether or not relatedness within all groups is high than expected. P-values cannot be exact in this case, because they are based on simulation rather than direct calculation. Suppose that we conducted 500 simulations, and only had 1 of these simulations with an average within-group relatedness greater than or equal to our observed value. It would be tempting to say that our p-value is 0.002 (1 out of 500). However, we need to be conservative here, because our estimate is based on simulation. The appropriate way to do this is to say that we have observed fewer than 2 out of 500 values less than or equal to our observed value. Thus $p < 0.004$ (fewer than 2 out of 500). This is how p-values are reported in these figures.

Two files are also generated, and written to R's working directory. One contains the average relatedness within observed groups (called "observed-r.csv"). The other contains the relatedness values for each group for each simulations (described above), called "expectedrel.csv".

Author(s)

Tim Frasier <timothy.frasier@smu.ca>

See Also

readgenotypedata, coancestry

Examples

```
## Not run:
#---Load Example Data---#
reldata <- readgenotypedata("simrel.csv")

#---Conduct analyses---#
grouprel(genotypes = reldata$gdata, estimatorname = "wang", usedgroups = "all", iterations = 100)

## End(Not run)
```

---

readgenotypedata *Read a file of genotype data into a data.frame*

Description

This function will import a genotype file into a proper data frame in R, estimate allele frequencies and store them as an appropriate object for subsequent analyses.

Usage

```
readgenotypedata(genotype.data)
```

Arguments

genotype.data	The file containing the genotype data to be analyzed. The file will need to be in R's working directory, and have the following characteristics: (1) It should be a text file (not and Excel file); (2) It should be space- or tab-delimited; (3) Missing data must be represented as zeros (0); and (4) There should not be a header row containing column names. Column 1 should contain individual identifiers, columns 2 and 3 should contain alleles 1 and 2 for locus 1, columns 4 & 5 should contain alleles 1 and 2 for locus 2, and so on. Thus, the total number of columns should be 2 x the number of loci + 1.

Value

gdata	The data fram containing the genotype data. The first column is character data, and the remaining columns are all integers.
nloci	An integer containing the number of loci used
nalleles	A series of integers specifying the number of alleles at each locus
ninds	An integer containing the number of individuals in the genotype file
freqs	An object containing the allele frequency data for each locus, which is needed for subsequent analyses

Author(s)

Jack Pew

See Also

coancestry

Examples

```
data(GenotypeData)
input <- readgenotypedata(GenotypeData)
```

---

readrelatedoutput *Read Related output files into R data.frame objects*

Description

This function is not seen by the user. It is used for reading the output from the Fortran Coancestry code into appropriate R data frames.

Usage

```
readrelatedoutput(working.directory = tempdir(), file.prefix = "", any.ci95 = F,
allow.inbreeding = F)
```

Arguments

working.directory	The location of the working directory where Coancestry will place the files.
file.prefix	The prefix for the files generated by Coancestry.
any.ci95	A TRUE/FALSE statement of whether or not confidence intervals were estimated.
allow.inbreeding	A TRUE/FALSE statement of wheter or not inbreeding was allowed/estimated.

Author(s)

Jack Pew

See Also

coancestry

---

simrel *Sample genotype data for group relatedness analysis.*

Description

Example genotype data for group relatedness analysis. Contains genotypes of 30 individuals: 10 full-sibs, 10 half-sibs, and 10 unrelated individuals. The groups are designated by the characters "FS", "HS", and "UR", respectively.

Usage

```
data(simrel)
```

Format

A space-delimited file, where the data for each individual is on a separate line. Contains one column of individual identifiers.

Examples

```
data(simrel)
summary(simrel)
```

Index