# Package ‘pould’

October 16, 2020

**Title** Phased or Unphased Linkage Disequilibrium

**Version** 1.0.1

**Description** Computes the D', Wn, and conditional asymmetric linkage disequilibrium (ALD) measures for pairs of genetic loci. Performs these linkage disequilibrium (LD) calculations on phased genotype data recorded using Genotype List (GL) String or columnar formats. Alternatively, generates expectation-maximization (EM) estimated haplotypes from phased data, or performs LD calculations on EM estimated haplotypes. Performs sign tests comparing LD values for phased and unphased datasets, and generates heatmaps for each LD measure. Described by Osoegawa et al. (2019a) <doi:10.1016/j.humimm.2019.01.010>, and Osoegawa et. al. (2019b) <doi:10.1016/j.humimm.2019.05.018>.

**Depends** R (>= 3.5.0)

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Imports** haplo.stats,gap,stats,utils,ggplot2,reshape2,BIGDAWG,graphics

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Steven Mack [aut, cre]

**Maintainer** Steven Mack <steven.mack@ucsf.edu>

**Repository** CRAN

**Date/Publication** 2020-10-16 13:50:03 UTC

## R topics documented:

## cALD *Calculation of the $D'$, $Wn$, and conditional Asymmetric LD Measures*

### Description

Calculates $D'$, $Wn$ (Cramer's V) and Thomson and Single's conditional asymmetric LD ($ALD$) measures for pairs of loci.

### Usage

```
cALD(
  dataSet,
  inPhase = FALSE,
  verbose = TRUE,
  saveVector = FALSE,
  vectorName = "",
  vectorPrefix = "",
  vecDir = tempdir()
)
```

### Arguments

`dataSet` A data frame or tab delimited file consisting of four columns of genotype data named, e.g. locus1_1 locus1_2 locus2_1 locus2_2, with 1 row per sample. The columns must be organized in this exact order, but the column names should not have _1 or _2 appended; use the same locus name for each column of a given locus. For phased data, locus1_1 is in phase with locus2_1, and locus1_2 is in phase with locus_2_2. Because this funciton operates on locus pairs, any rows with missing data should be excluded from the input genotype data.

`inPhase` A boolean identifying the genotyping data as phased or unphased (TRUE = phased; FALSE = unphased); default is unphased.

`verbose` A boolean identifying if results should be printed to the console (verbose = TRUE), or returned in a vector of ($D'$, $Wn$, $WLocus2/Locus1$, $WLocus1/Locus2$, number of haplotypes) (verbose = FALSE)

`saveVector` A boolean identifying if the vector of all haplotypes should be exported as a text file (saveVector = TRUE), or not (saveVector = FALSE).

`vectorName` A name for the exported haplotype vector file; this name is not used if saveVector = FALSE. If a name is unspecified, then a filename including the locus-pair and a timestamp is generated.

`vectorPrefix` An optional prefix for the haplotpe vector to be used if saveVector = TRUE. This prefix will be appended, along with the phase status, before the locus name and timestamp. LDWrap() uses this parameter to identify the dataset and haplotype information passed to cALD().

`vecDir` The directory into which the haplotype vector should be written if saveVector = TRUE. The default is the directory specified by tempdir().

### Details

LD results can be directed to the console or to a data file or data frame object. This function can generate a haplotype vector file for each locus pair analyzed, and will return the LD results eiher in the console, or as a data frame object. The implementation of ALD applied here is calculated using individual $Dij$ LD values and allele frequencies.

### Value

A vector of D', Wn, WLocus2/Locus1, WLocus1/Locus2 values, and the number of haplotypes evaluated

### References

Thomson G. & Single R.M. GENETICS 2014;198(1):321-31. https://doi.org/10.1534/genetics.114.165266

### Examples

```
# Analyze the first 10 rows of the included drb1.dqb1.demo genotype dataset
# and report LD results to the console.
cALD(drb1.dqb1.demo[1:10,])
# Alternatively, return a vector of LD results.
LDvec <- cALD(drb1.dqb1.demo[1:10,],verbose=FALSE)
```

## `drb1.dqb1.demo` *Example HLA Genotype Data for DRB1 and DQB1*

### Description

A data frame of phased two-field HLA-DRB1 and HLA-DQB1 genotypes for 419 European American control subjects included in a study of Multiple Sclerosis. Genotype data for each locus is shown in two columns; the alleles in columns 1 and 2 are phased with the alleles in columns 3 and 4, respectively. These genotypes were extracted from the hla.hap.demo dataset.

### Usage

```
data(drb1.dqb1.demo)
```

### Format

A data frame with 419 rows and four columns

- DRB1: the first of two two-field HLA-DRB1 alleles
- DRB1: the second of two two-field HLA-DRB1 alleles
- DQB1: the first of two two-field HLA-DQB1 alleles
- DQB1: the second of two two-field HLA-DRB1 alleles

### Source

immport.org study SDY1045 doi:10.21430/M3QW34U2SG

### References

Mack et al. Genes Immun. 2019;20(4):308-326. doi: 10.1038/s41435-017-0006-8.

---

## `extractLoci` *Extract Locus Information from Supplied Haplotype Data*

---

### Description

This function extracts locus information from the haplotype data, and structures it for LDWrap().

### Usage

```
extractLoci(dataSet)
```

### Arguments

| | |
|---|---|
| `dataSet` | Data frame of two haplotypes extracted from the famData provided to LDWrap() |

### Value

List of two vector elements; the locus prefix (if any), e.g. "HLA-", and the interleaved unsuffixed and suffixed locus names (e.g., locus, locus_1)

### Note

This function is for internal POULD use only.

### Examples

```
#
```

## `hla.hap.demo` *Example Six-Locus HLA Haplotype Data in GL String Format*

### Description

A data frame of experimentally phased genotype data for the HLA-A, -C, -B, -DRB1, -DQB1 and DQB1 loci. These haplotypes are for 419 unrelated European American control subjects included in a study of Multiple Sclerosis. These haplotypes were experimentally phased using the EM algorithm; low-frequency haplotypes (with counts < 3) are assigned stochasically by the EM method, so the phase "quality" of these haplotypes should be considered lower than that for haplotypes determined via family segregation analysis.

### Usage

```
data(hla.hap.demo)
```

### Format

A data frame with 419 rows and two columns

- Relation: the genotyped individual's status in a family
- Gl String: GL String formatted multilocus phased HLA genotype

### Note

This data is formatted as example input for the LDWrap() function, but is not actual family data. Actual family data would include 'mother', 'father' and 'child' in the "Relation" field. Nevertheless, including 'Subject' in this field is sufficient for analysis using LDWrap().

### Source

immport.org study SDY1045 doi:10.21430/M3QW34U2SG

### References

- Published study: Mack et al. Genes Immun. 2019;20(4):308-326. doi: 10.1038/s41435-017-0006-8.
- GL String format: Milius et al. Tissue Antigens. 2013;82(2):106-12. doi: 10.1111/tan.12150.

## `LD.heat.map` *Generates heat-maps for four linkage disequilibrium (LD) values (D', Wn, WLoc1/Loc2 and WLoc2/Loc1) generated for all pairs of phased and unphased two-locus haplotypes by LDWrap().*

### Description

This function accepts *_LD_results.csv files generated by LDWrap() as input, and generates a PNG-formatted heat-map plot file for each LD measure.

### Usage

```
LD.heat.map(
  dataName = "",
  phasedData = "",
  unphasedData = "",
  phasedLabel = "Phased",
  unphasedLabel = "EM-estimated",
  color = TRUE,
  writePlot = FALSE,
  writeDir = tempdir()
)
```

### Arguments

`dataName` The "base" name of the _LD_result.csv files generated by LDWrap() without the "_Phased_LD_results.csv" or "_Unphased_LD_results.csv" suffixes. See Examples, below. If both corresponding "<dataName>_Phased_LD_results.csv" and "<dataName>_Unphased_LD_results.csv" files are not found, the funciton will halt with a notification. However, if only one of those files is found in the working director, half-matrix heat map plots will be generated.

`phasedData` The complete name of a file of phased LD results generated by LDWrap(). Provide this filename if no base name is provided for dataName and you want to generate heat-maps for a specific set of phased LD values.

`unphasedData` The complete name of a file of unphased LD results generated by LDWrap(). Provide this filename if no base name is provided for dataName and you want to generate heat-maps for a specific set of unphased LD values.

`phasedLabel` The label that should appear on the heat-map plots for the upper, phased half of the plot. The default option is 'Phased'.

`unphasedLabel` The label that should appear on the heat-map plots for the lower, unphased half of the plot. The default option is 'EM-estimated'.

`color` A logical parameter that identifies if the heat-maps should be plotted in color (TRUE) or greyscale (FALSE). The default option is TRUE.

`writePlot` A logical parameter that identifies if the heat-map plots should be automatically saved after they are generated. The default is 'writePlot=FALSE'.

`writeDir` The directory into which the heat-map plots should be saved when 'writePlot=TRUE'. The default is the directory specified by tempdir().

## References

Osoegawa et al. Hum Immunol. 2019;80(9):644 https://doi.org/10.1016/j.humimm.2019.05.018

## Examples

```
# Using the results of LDWrap() for the first 10 rows of the drb1.dqb1.demo dataset.
# Results are saved in the temporary directory as
# "hla-family-data_Phased_LD_results.csv" and
# "hla-family-data_Unphased_LD_results.csv", respectively.
LDWrap(drb1.dqb1.demo[1:10,])
LDWrap(drb1.dqb1.demo[1:10,],phased=FALSE)
exampleData <- paste(tempdir(),"hla-family-data",sep=.Platform$file.sep)
LD.heat.map(exampleData)
# Alternatively, these files can be sepcified individually to generate a half-matrix.
LD.heat.map(phasedData=paste(exampleData,"_Phased_LD_results.csv",sep=""),unphasedLabel="")
# Further, two different sets of results for the same loci can be plotted; e.g., using
# phasedData="my_Phased_LD_results.csv" and unphasedData="your_Phased_LD_results.csv".
```

# `LD.sign.test` *Perform the sign test on paired LD values for phased and unphased haplotypes*

## Description

Applies binom.test() to pairs of D', Wn, WLoc1/Loc2, WLoc2/Loc1 and number of haplotypes values in the *_LD_results.csv files generated by LDWrap().

## Usage

```
LD.sign.test(
  dataName,
  verbose = TRUE,
  returnFrame = TRUE,
  resultDir = tempdir()
)
```

## Arguments

`dataName` The "base" name of the _LD_result.csv files generated by LDWrap() without the "_Phased_LD_results.csv" or "_Unphased_LD_results.csv" suffixes. See Examples, below. If the corresponding "<dataName>_Phased_LD_results.csv" or "<dataName>_Unphased_LD_results.csv" files are not found, the funciton will halt with a notification.

`verbose` A boolean identifying if messages about function progress and results should be displayed in the console (verbose=TRUE) or not (verbose=FALSE). The default is verbose=TRUE.

`returnFrame` A boolean identifying if a data frame of results should be returned (returnFrame=TRUE). If 'returnFrame=FALSE', a CSV file of results named <dataName>_LD-sign-test_results.csv is written in the directory specified by the 'resultDir' parameter. The default is returnFrame=TRUE.

`resultDir` The directory into which the CSV file of results should be written when 'returnFrame=FALSE'. The default is the directory specified by tempdir().

### Details

This function returns p-values for the sign test comparing the phased and unphased values of each LD measure, as well as the number of haplotypes, for each locus pair in a dataset tested using LDWrap(). It also returns the number of locus pairs for which the value in quesiton is higher in unphased haplotypes than phased haplotypes, the number of locus pairs in which the values are the same, and the total number of locus pairs asssessed. This function writes a results file to the working directory for each dataset, and will optionally display those results in the console.

### Value

A data frame of five columns (D', Wn, WLoc1/Loc2, WLoc2/Loc1 and N_Haplotypes) and four rows (#unphased > phased, #unphased = phased, #locus pairs and p-values).

### Note

When verbose=TRUE, LD.sign.test() writes a table of results to the console with the column headers "Measure", "#U > P", "#U = P" and "p-value". Column "#U > P" identifies the number locus pairs for which that measure for unphased haplotypes (U) was greater than that measure for phased haplotypes (P). Similarly, column "#U = P" identiifes the number of locus pairs for each meausure where the value of that measure was the same in phased and unphased haplotypes.

Only the significance of the sign test is reported; when a significant trend is indicated, the directionality of the trend is not reported.

### References

Osoegawa et al. Hum Immunol. 2019;80(9):633 (https://doi.org/10.1016/j.humimm.2019.01.010)

### Examples

```
# Using LDWrap() to analyze the first 10 rows of the drb1.dqb1.demo dataset.
# LDWrap() results are saved in the temporary directory as
# "hla-family-data_Phased_LD_results.csv" and
# "hla-family-data_Unphased_LD_results.csv", respectively.
LDWrap(drb1.dqb1.demo[1:10,])
LDWrap(drb1.dqb1.demo[1:10,],phased=FALSE)
exampleData <- paste(tempdir(),"hla-family-data",sep=.Platform$file.sep)
LD.res <- LD.sign.test(exampleData)
# Return only a data frame of results.
LD.res <- LD.sign.test(exampleData,verbose=FALSE)
```

## LDWrap *Parser for CSV-formatted GL String Haplotype Data*

### Description

A wrapper for parsing phased haplotype data recorded in GL String format. Extracts all pairs of loci from GL String formatted haplotypes or column formatted genotypes, passes paired-genotype data to the cALD function, and generates files consumed by the LD.sign.test() and LD.heat.map() functions.

### Usage

```
LDWrap(
  famData,
  threshold = 10,
  phased = TRUE,
  frameName = "hla-family-data",
  trunc = 0,
  writeTo = tempdir()
)
```

### Arguments

`famData` A data frame or CSV formatted file (with a .csv filename suffix) that contains the two columns named "Gl String" and "Relation". Other columns can be included (in any order), but will not impact the analysis. The Relation column can contain any data; however anything other than "Relation=child" will be included in the LD analyses. The Gl String column should consist of two tilde (~) delimited haplotypes conneced by a plus (+) sign (GL String format). Allele names should be recorded using the LOCUS*VARIANT structure used for HLA and KIR alleles. A locus prefix (e.g., 'HLA-') is not required, but if a locus prefix is included, all allele names must include the same locus prefix. Alternatively, LDWrap() will consume genotype data in a data frame or headered tab-delimited text file (TXT or TSV), with two columns per locus. See the parseGenotype() documentation for additional requirements. The name of the file provided will serve as the basis for the name of the LD result files.

`threshold` An integer that specifies the minimnum number of subjects allowed for the analysis of a locus-pair. The default value is 10. If the number of subjects with haplotypes for a locus pair is less than the threshold, the *_LD_results.csv file will contain 'Not Calculated' 'Subject Threshold=##' 'Complete subjects=#' '.' in columns 2-5 for that locus pair, where ## is the set threshold and # is the number of subjects. Column 6 will be empty.

`phased` A boolean that determines if the LD calculations should be performed for phased data (TRUE) or unphased data (FALSE). If phased=FALSE, the EM algorithm is used to estimate haplotypes for the data in the Gl String column of family haplotype datasets.

`frameName` A descriptor for the data frame of family data provided. The default value is "hla-family-data". This value is not used if a CSV file is provided.

`trunc` An integer that specifies the number of fields to which colon-delimited allele names in famdData should be truncated. The default value of 0 indicates no truncation. A value higher than the number of fields in the supplied allele data will result in no truncation. When a positive value of trunc is provided, the names of the output files will include the specified truncation level.

`writeTo` The directory into which the LDWrap() output files should be written. The default is the directory specified by tempdir().

## Details

This function coerces cALD() to generate a haplotype vector file for each locus pair analyzed, and generates a single LD results file containing LD values for all locus pairs, along with the number of haplotypes tested, one locus pair per row. The LD results file will contain six columns ("Loc1~Loc2","D’","Wn","WLoc1/Loc2","WLoc2/Loc1","N_Haplotypes"), and will be named "<filename prefix>_<Phased/Unphased>_LD_results.csv".

## Note

When at least one locus in a locus pair is monomorphic, no LD calculations will be performed, and column 5 of the results for that locus pair will identify the monomorphic loci.

This function does not validate HLA allele names. Unusual allele names (e.g., ‘HLA-A*NULL‘, ‘HLA-DRB1*NoMatch‘, ‘HLA-DPB1*NT‘) and truncated versions of allele names (e.g., ‘HLA-A*01‘, ‘HLA-A*01:01‘, ‘HLA-A*01:01:01‘, etc.) will be analyzed as distinct alleles. Including unusual allele names or different truncated versions of the same allele name in a dataset will likely skew the analytic results. In the latter case, the trunc parameter can be used to specify analysis at a specific number of fields.

Column-formatted genotype data are generally unphased; unless genotype data have been structured so that all alleles in the first column for each locus are in one haplotype, and all of the alleles in the second column in each locus are in the other haplotype, phased should be set to FALSE for column-formatted genotype datasets.

## References

Osoegawa et al. Hum Immunol. 2019;80(9):633 (https://doi.org/10.1016/j.humimm.2019.01.010)

Osoegawa et al. Hum Immunol. 2019;80(9):644 (https://doi.org/10.1016/j.humimm.2019.05.018)

## Examples

```
# Analyze the first 10 rows of the drb1.dqb1.demo genotype dataset.
LDWrap(drb1.dqb1.demo[1:10,],frameName="DRDQDemo")
# Analyze the includeed example genotype data with all alleles truncated to one field.
LDWrap(drb1.dqb1.demo[1:10,],frameName="DRDQDemoTrunc",trunc=1)
```

## parseGenotypes *Reformat columnnar genotype data to GL String format*

### Description

This function accepts genotype data organized in locus-column pairs, and returns GL String-formatted data structured for LDWrap(). Of the resulting multilocus haplotype pair, the first haplotype is constructed from the first column for each locus, and the second haplotype is constructed from the second column.

### Usage

```
parseGenotypes(dataset)
```

### Arguments

| | |
|---|---|
| `dataset` | A tab-delimited text file (with a .txt or .tsv filename suffix) with a header row or a data frame. Each row corresponds to a subject, with two columns per locus. Allele names can include a locus name (e.g., locus*allele) or can can exclude the locus, but all allele names in the dataset must either include or exclude the locus. Missing (untyped) allele data can be identified with an empty cell or a set of four asterisks in files, and with NA values in data frames. Column names for each locus pair must be adjacent, but can be either identical (e.g., "locus" and "locus"), or suffixed (e.g., "locus_1" and "locus_2", where "locus_1" always precedes "locus_2"). A optional column of sample identifiers can be included, but must be named "SampleID". A column named "Disease" can be included, but will be ignored. No other non-locus columns are permitted. |

### Value

A data frame of two columns. The "Relation" column includes sample identifiers if provided, or numbers from 1 to the number of subjects. The "GL String" column contains the GL String formatted genotypes.

### Note

This function is for internal POULD use only.

### Examples

```
#
```

## trimAlleles *Truncate allele names in haplotypes to the specified number of fields.*

### Description

This function accepts a dataframe of tilde-delimited haplotypes and trims colon-delimited names to the number of fields specified by 'reso'.

### Usage

```
trimAlleles(haplotypes, reso)
```

### Arguments

| | |
|---|---|
| `haplotypes` | Data frame of tilde-delimited haplotypes extracted from the famData provided to LDWrap() |
| `reso` | An integer that specifies the number of fields to which colon-delimited allele names in famData should be truncated. The default value of 0 indicates no truncation. A value higher than the number of fields in the supplied allele data will result in no truncation. |

### Value

A data frame of two sets of tilde-delimited haplotypes.

### Note

This function is for internal POULD use only.

### Examples

```
#
```

## writeVector *Exporting Haplotype Vectors*

### Description

This function writes the haplotype vector and any accessory information to a text file.

### Usage

```
writeVector(nLoc1, Res, hapVec, numSamp, writeName, genPhase, Prefix, vDir)
```

## Arguments

`nLoc1` Number of alleles at locus 1

`Res` Data frame consisting of Locus Number, Locus Name, Allele Name, Allele Count, Allele Frequency and Frequency^2

`hapVec` Haplotype vector

`numSamp` The dataset's 2N value

`writeName` The specified name for the vector

`genPhase` Boolean describing phased (TRUE) or unphased (FALSE) analysis

`Prefix` The specified prefix for the written vector file

`vDir` The directory into which the vector file should be written.

## Note

This function is for internal POULD use only.

## Examples

```
#
```

# Index