

# Package ‘ldblock’

December 23, 2024

**Title** data structures for linkage disequilibrium measures in populations

**Version** 1.37.0

**Author** VJ Carey <stvjc@channing.harvard.edu>

**Description** Define data structures for linkage disequilibrium measures in populations.

**Suggests** RUnit, knitr, BiocStyle, gwascat, rmarkdown, snpStats, VariantAnnotation, GenomeInfoDb, ensemblDb, EnsDb.Hsapiens.v75, Rsamtools, GenomicFiles (>= 1.13.6)

**Imports** BiocGenerics (>= 0.25.1), httr, Matrix

**Depends** R (>= 3.5), methods, rlang

**Maintainer** VJ Carey <stvjc@channing.harvard.edu>

**License** Artistic-2.0

**LazyData** no

**BiocViews** genetics, SNP, GWAS, LinkageDisequilibrium

**VignetteBuilder** knitr

**RoxygenNote** 7.2.0

**Encoding** UTF-8

**git\_url** <https://git.bioconductor.org/packages/ldblock>

**git\_branch** devel

**git\_last\_commit** 6555373

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.21

**Date/Publication** 2024-12-23

## Contents

ldblock-package . . . . .	2
downloadPopByChr . . . . .	3
EUR_singletons . . . . .	4

expandSnpSet . . . . .	4
hml . . . . .	5
ldByGene . . . . .	6
ldmat . . . . .	7
ldmat,ldstruct-method . . . . .	8
ldstruct-class . . . . .	8
s3_1kg . . . . .	8
sampinf_1kg . . . . .	9
stack1kg . . . . .	10

<b>Index</b>	<b>11</b>
--------------	-----------

---

ldblock-package	<i>c("\Sexpr[results=rd,stage=build]tools:::Rd_package_title(\#1\)", "ldblock")data structures for linkage disequilibrium measures in populations</i>
-----------------	---

---

## Description

`c("\Sexpr[results=rd,stage=build]tools:::Rd_package_description(\#1\)", "ldblock")` Define data structures for linkage disequilibrium measures in populations.

## Details

The DESCRIPTION file: `c("\Sexpr[results=rd,stage=build]tools:::Rd_package_DESCRIPTION(\#1\)", "ldblock")` This package was not yet installed at build time.  
`c("\Sexpr[results=rd,stage=build]tools:::Rd_package_indices(\#1\)", "ldblock")` Index: This package was not yet installed at build time.

## Author(s)

`c("\Sexpr[results=rd,stage=build]tools:::Rd_package_author(\#1\)", "ldblock")` VJ Carey <stvjc@channing.harvard.edu>

Maintainer: `c("\Sexpr[results=rd,stage=build]tools:::Rd_package_maintainer(\#1\)", "ldblock")` VJ Carey <stvjc@channing.harvard.edu>

## Examples

# see vignette

---

downloadPopByChr      *download hapmap resource with LD estimates*

---

## Description

download hapmap resource with LD estimates

## Usage

```
downloadPopByChr(  
  chrname = "chr1",  
  popname = "CEU",  
  
  urlTemplate = "http://hapmap.ncbi.nlm.nih.gov/downloads/ld_data/2009-02_phaseIII_r2/ld_%%CHRN%%_%  
  targfolder = Sys.getenv("LDBLOCK_TXTGZ_DIR")  
)
```

## Arguments

chrname	UCSC format tag for chromosome
popname	hapmap three letter code for population, e.g. 'CEU'
urlTemplate	pattern for creating URL given chr and pop
targfolder	destination

## Details

delivers HapMap LD data to 'targfolder'

## Value

just run for side effect of download.file

## Examples

```
## Not run:  
  downloadPopByChr()  
  
## End(Not run)
```

---

EUR_singletons	<i>singletons from EUR</i>
----------------	----------------------------

---

**Description**

singletons from EUR

**Usage**

```
EUR_singletons
```

**Format**

character vector

**Source**

[ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/working/20130606\\_sample\\_info/20130606\\_sample\\_info.xlsx](ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/working/20130606_sample_info/20130606_sample_info.xlsx), to which superpopulation codes were added

---

expandSnpSet	<i>Given a set of SNP identifiers, use LD to expand the set to include linked loci</i>
--------------	--

---

**Description**

Given a set of SNP identifiers, use LD to expand the set to include linked loci

**Usage**

```
expandSnpSet(  
  rsl,  
  lb = 0.8,  
  ldstruct,  
  chrn = "chr17",  
  popn = "CEU",  
  txtgzfn = dir(system.file("hapmap", package = "ldblock"), full.names = TRUE)  
)
```

**Arguments**

rs1	input list – SNPs not found in the LD structure are simply returned along with those found, and the expansion list, all combined in a vector
lb	lower bound on statistic used to retrieve loci in LD
ldstruct	instance of <code>ldstruct-class</code>
chrn	chromosome identifier
popn	population identifier (one of 'CEU', 'MEX', ...)
txtgzfn	path to gzipped hapmap file with LD information

**Details**

direct use of elementwise arithmetic comparison

**Value**

character vector

**Note**

As of 2015, it appears that locus names are more informative than addresses for determining SNP identity across resources.

**Examples**

```
og = Sys.getenv("LDBLOCK_TXTGZ_DIR")
on.exit( Sys.setenv("LDBLOCK_TXTGZ_DIR" = og ) )
Sys.setenv("LDBLOCK_TXTGZ_DIR"=system.file("hapmap", package="ldblock"))
ld17 = hmlD(chr="chr17", pop="CEU")
ee = expandSnpSet( ld17@allrs[1:10], ldstruct = ld17 )
```

---

hmlD	<i>import hapmap LD data and create a structure for its management; generates a sparse matrix representation of pairwise LD statistics and binds metadata on variant name and position</i>
------	--

---

**Description**

import hapmap LD data and create a structure for its management; generates a sparse matrix representation of pairwise LD statistics and binds metadata on variant name and position

**Usage**

```
hmlD(hmgztxt, poptag, chrom, genome = "hg19", stat = "Dprime")
```

**Arguments**

hmgztxt	name of gzipped text file as distributed at <a href="http://hapmap.ncbi.nlm.nih.gov/downloads/ld_data/2009-02_phaseIII_r2/">hapmap.ncbi.nlm.nih.gov/downloads/ld_data/2009-02_phaseIII_r2/</a> . It will be processed by <a href="#">read.delim</a> .
poptag	heuristic tag identifying population
chrom	heuristic tag for chromosome name
genome	genome tag
stat	statistic to use, "Dprime", "R2", and "LOD" are options

**Value**

instance of ldstruct class

**Examples**

```
getClass("ldstruct")
# see vignette
```

---

ldByGene

---

*Obtain LD statistics in region specified by a gene model.*


---

**Description**

Obtain LD statistics in region specified by a gene model.

**Usage**

```
ldByGene(
  sym = "MMP24",
  vcf = system.file("vcf/c20exch.vcf.gz", package = "ldblock"),
  flank = 1000,
  vcfSLS = "NCBI",
  genomeSLS = "hg19",
  stats = "D.prime",
  depth = 10
)
```

**Arguments**

sym	A standard gene symbol for use with <code>genemodel</code>
vcf	Path to a tabix-indexed VCF file
flank	number of basepairs to flank gene model for search
vcfSLS	seqlevelsStyle (SLS) token for VCF; will be imposed on gene model
genomeSLS	character tag for genome, to be used with <code>readVcf</code>
stats	passed to <code>ld</code>
depth	passed to <code>ld</code>

**Value**

sparse matrix representation of selected LD statistic, as returned by `ld`

**Note**

Uses an internal function `genemod4ldbblock`, that relies on `EnsDb.Hsapiens.v75` to get gene model.

**Examples**

```
if (interactive()) { # there is a warning owing to non-SNV present
  ld1 = ldByGene(depth=150)
  image(ld1[1:200,1:200], col.reg=heat.colors(120), colorkey=TRUE,
        main="SNPs in MMP24 (chr20)")
}
```

---

`ldmat`*use LDmat API from NCI LDlink service*

---

**Description**

use LDmat API from NCI LDlink service

**Usage**

```
ldmat(rsvec, pop = "CEU", type = "d", token = Sys.getenv("LDLINK_TOKEN"))
```

**Arguments**

<code>rsvec</code>	character vector of SNP ids
<code>pop</code>	three letter code for HapMap population, defaults to CEU
<code>type</code>	'r2' or 'd', defaults to 'd' implying d-prime
<code>token</code>	the API token provided by NCI, defaults to value of environment variable <code>LDLINK_TOKEN</code>

**Value**

data.frame

**Examples**

```
if (interactive()) ldmat(c("rs77749396", "rs9303279", "rs9303280", "rs9303281"))
```

---

`ldmat, ldstruct-method` *accessor for matrix component*

---

### Description

accessor for matrix component

### Usage

```
## S4 method for signature 'ldstruct'
ldmat(x)
```

### Arguments

`x` instance of `ldstruct`

---

`ldstruct-class` *container for LD data*

---

### Description

Manage information about LD statistics as reported by HapMap.

### Objects from the Class

Objects can be created by calls of the form `new("ldstruct", ...)`.

### Examples

```
showClass("ldstruct")
```

---

`s3_1kg` *Create a URL referencing 1000 genomes content in AWS S3. `stack1kg` produces a `VcfStack` instance with references to VCF for 1000 genomes autosomal chrs. S3-resident VCF files with version "v5a.20130502" are used.*

---

### Description

Create a URL referencing 1000 genomes content in AWS S3. `stack1kg` produces a `VcfStack` instance with references to VCF for 1000 genomes autosomal chrs. S3-resident VCF files with version "v5a.20130502" are used.



**Usage**

```
s3_1kg(chrnum, tmpl, dropchr = TRUE)
```

**Arguments**

chrnum	a character string denoting a chromosome, such as '22'
tmpl	alternate template for full URL, useful if versions prior to 2010 are of interest
dropchr	if TRUE chrnum will have 'chr' removed if present

**Value**

by default, a TabixFile instance

**Note**

The "wrap" parameter has been removed. A TabixFile structure will be returned. The tag parameter has been removed. Supply a tmpl argument if you are not using 20130502 version.

**Examples**

```
requireNamespace("Rsamtools")  
s3_1kg("22") # try scanVcfHeader from VariantAnnotation
```

---

sampinf_1kg	<i>population and relationship information for 1000 genomes</i>
-------------	---

---

**Description**

population and relationship information for 1000 genomes

**Usage**

```
sampinf_1kg
```

**Format**

```
data.frame
```

**Source**

[ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/working/20130606\\_sample\\_info/20130606\\_sample\\_info.xlsx](ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/working/20130606_sample_info/20130606_sample_info.xlsx), to which superpopulation codes were added

stack1kg                      *couple together a group of VCFs*

---

**Description**

couple together a group of VCFs

**Usage**

```
stack1kg(chrs = as.character(1:22), index = FALSE, useEBI = FALSE)
```

**Arguments**

chrs	a vector of chromosome names for extraction from 1000 genomes VCF collection
index	logical telling whether VcfStack should attempt to create the local index; for 1000 genomes, the tbi are in the cloud and will be used by readVcf so FALSE is appropriate
useEBI	logical(1) defaults to FALSE ... if TRUE, use tabix-indexed vcf from EBI, but in July 2022 the EBI FTP site does not respond. If FALSE, the AWS Open Data access path is used

**Value**

VcfStack instance

**Note**

The seqinfo component of returned stack will have NA for genome. Please set it manually; for useEBI=TRUE this would be GRCh38; very likely so for useEBI=FALSE, but this should be checked.

**Examples**

```
if (interactive()) {  
  st1 = stack1kg()  
  st1  
}
```

# Index

- \* **classes**
  - ldstruct-class, 8
- \* **datasets**
  - EUR\_singletons, 4
  - sampinf\_1kg, 9
- \* **models**
  - downloadPopByChr, 3
  - expandSnpSet, 4
  - hml, 5
  - ldByGene, 6
  - s3\_1kg, 8
- \* **package**
  - ldblock-package, 2

downloadPopByChr, 3

EUR\_singletons, 4

expandSnpSet, 4

hml, 5

ld, 6, 7

ldblock (ldblock-package), 2

ldblock-package, 2

ldByGene, 6

ldmat, 7

ldmat, ldstruct-method, 8

ldstruct-class, 8

read.delim, 6

s3\_1kg, 8

sampinf\_1kg, 9

stack1kg, 10