Package 'plinkFile'

July 23, 2025

Title 'PLINK' (and 'GCTA') File Helpers

Version 0.2.1

Description Reads/write binary genotype file compatible with 'PLINK' <https://www.cog-genomics.org/plink/1.9/input#bed> into/from a R matrix; traverse genotype data one windows of variants at a time, like apply() or a for loop; reads/writes genotype relatedness/kinship matrices created by 'PLINK' <https://www.cog-genomics.org/plink/1.9/distance#make_rel> or 'GCTA' <https://cnsgenomics.com/software/gcta/#MakingaGRM> into/from a R square matrix. It is best used for bringing data produced by 'PLINK' and 'GCTA' into R workflow.

Depends R (>= 3.5.0)

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Author Xiaoran Tong [aut, cre]

Maintainer Xiaoran Tong <xiaoran.tong.cn@gmail.com>

Repository CRAN

Date/Publication 2023-11-24 12:00:04 UTC

Contents

`CHR` *A dictionary to map chromosome names to integers.*

Description

A dictionary to map chromosome names to integers.

`DBT` *A decoding byte table to map raw intergers to genotype dosages.*

Description

The table avoids bit shifting and may speed up the reading of plink BED.

Details

The decoding table approach is still experimental.

Actually, DBT == dbd(as.raw(seq(0x00, 0xFF)), 4L)

`readBED` *Read BED file*

Description

Read a PLINK BED file into a R matrix.

Usage

```
readBED(pfx, iid = 1, vid = 1, vfr = NULL, vto = NULL, quiet = TRUE)
```

Arguments

pfx	prefix of PLINK file set, or the fullname of a BED file.
iid	option to read N IID as row names. (def=1, see readIID())
vid	option to read P VID as col names. (def=1, see readVID())
vfr	variant-wise, from where to read? (number/proportion, def=1)
vto	varinat-wise, to where then stop? (number/proportion, def=P)
quiet	suppress screen printing? (def=TRUE)

Details

This is meant for genotype that can fit into system memory; the size of R matrix is 16 times the BED file. To traverse a huge BED several varants at time without loading it entirely into the memoty, see scanBED() and loopBED().

A PLINK1 binary fileset has three files,

pfx.fam: text table of N individuals.

pfx.bim: text table of P genomic variants (i.e., SNPs).

pfx.bed: N x P genotype matrix in condensed binary format.

The three files comprising a genotype data are typically referred by their common prefix, for example, the X chromosome genotype represented by chrX.bed, chrX.fam, and chrX.bim are jointly refered by chrX.

Value

genotype matrix with N row individuals and P column variants.

See Also

readBED

Examples

```
## read an entire small data
bed <- system.file("extdata", 'm20.bed', package="plinkFile")
gmx <- readBED(bed, quiet=FALSE)

## read part of a large data
bed <- system.file("extdata", '000.bed', package="plinkFile")
U <- readBED(bed, vfr=01, vto=10, quiet=FALSE)
V <- readBED(bed, vfr=11, vto=20, quiet=FALSE)
W <- cbind(U, V)
X <- readBED(bed, vfr=01, vto=20, quiet=FALSE)
all.equal(W, X)
```

`readBIM` *Read BIM file*

Description

Get variant meta-data form the *bim* file of a PLINK1 BED fileset.

Usage

```
readBIM(bim, vfr = NULL, vto = NULL)
```

Arguments

`bim`	prefix or name of a PLINK file.
`vfr`	variant-wise, from where to read? (index/proportion, def=1).
`vto`	varinat-wise, to where then stop? (index/proportion, def=P).

Details

There are six columns in a *bim* file

- chr: chromosme of the variant
- vid: variant id, such as an RS number;
- cmg: position by centimorgan;
- bps: position by basepairs;
- al1: allele 1, the one counted as dosage.
- al2: allele 2.

Value

data frame of variants, loaded from BIM.

Examples

```
bed <- file.path(system.file("extdata", package="plinkFile"), "000.bed")
bim <- readBIM(bed, 20, 30)
bim
```

readBSM *Read Binary Symmetric Matrix (BSM)*

Description

Read BSM represented by a pair of files suffixed by ".bin" and ".id", usually produced by PLINK and GCTA.

Usage

```
readBSM(pfx, dgv = 1, fid = NULL, id = NULL, bin = NULL)
```

Arguments

pfx	prefix of data files pfx.id and pfx.bin
dgv	diagonal value for matrix without a diagonal (def=1.0)
fid	separator between FID and IID (def=NULL, use IID only)
id	use id file instead of the default pfx.id
bin	use bin file instead of the default pfx.bin

Details

The ".bin" is a binary file storing the matrix entries, which can be

SQR the N x N symmetric matrix in full

LWD the lower triangle with diagonal

LND the lower triangle without diagonal

, saved as either single or double precision.

The ".id" is a text file of N family ID (FID) and individual ID (IID) in two columns. By default, IID is used as matix row and column names.

PLINK option --make-red bin, --distance bin, and GCTA option --make-grm all creats binary symmetric matrices, widely used in linear mixed model or kernel based models for genetics.

Value

symmetric matrix loaded, with sample ID as both row and column names.

Examples

```
pfx <- file.path(system.file("extdata", package="plinkFile"), "m20.rel")
(readBSM(pfx, fid=":"))
```

`readFAM` *Read FAM file*

Description

Read sample meta-data form the *fam* file of a PLINK1 BED fileset.

Usage

```
readFAM(fam)
```

Arguments

`fam`	prefix or name of a PLINK file.

Details

There are six columns in a *bim* file

- fid: family ID;
- iid: individual ID, default row name used by `[readBED]`;
- mom: maternal ID;
- dad: paternal ID;
- sex: individual sex.
- phe: phenotype, not often used;

The PLINK1 *bim* file has no header line, this is changed in PLINK2.

The columns "sex" and "phe" are mostly the legency of early GWAS, nowerdays it is common to provide sex, among other covariates, and multiple phenotypes in a separate file.

Value

data frame of individuals, loaded from FAM.

Examples

```
pfx <- file.path(system.file("extdata", package="plinkFile"), "m20")
fam <- readFAM(pfx)
fam
```

`readGRM` *Read Genetic Related Matrix (GRM) of GCTA*

Description

GRM is the core formt of GCTA, which is an binary symmetric matrix with an extra variant count matrix (VCM), this function reads the binary sysmmetric matrix.

Usage

```
readGRM(pfx, fid = NULL)
```

Arguments

`pfx`	prefix of GRM file set
`fid`	separator after family ID (def=NULL, use IID only)

Details

GCTA GRM is represented by a set of three files:

.grm.bin : GRM matrix in binary

.grm.id : sample FID and IID in text

.grm.N.bin : number of valid variants for each GRM entry

and it always uses single precision (4 bytes per entry).

To read the extra the extra VCM (grm.N.bin), use `readVCM`.

Value

matrix of relatedness with sample ID in row and column names.

Examples

```
pfx <- file.path(system.file("extdata", package="plinkFile"), "m20")
(readGRM(pfx))
```

readIBS *Read PLINK Binary IBS matrix*

Description

A PLINK IBS (Identity by State) matrix is represented by

.mibs.bin: IBS matrix in binary

.mibs.id : FID and IID in text

A binary IBS matrix is the result of PLINK `--distance ibs bin`

Usage

```
readIBS(pfx, fid = ".")
```

Arguments

`pfx`	prefix of the IBS file set.
`fid`	seperate after family ID (def=NULL, use IID only)

Value

IBS matrix with row and column names set to sample ID.

Examples

```
pfx <- file.path(system.file("extdata", package="plinkFile"), "m20")
(readIBS(pfx))
```

readIID *read individual ID*

Description

Generate individual ID automatically, or based on a *fam* file.

Usage

```
readIID(fam, opt = NULL)
```

Arguments

`fam`	prefix or name of a PLINK file, or data fram from a FAM file.
`opt`	option (def=1: the 2nd column in FAM file).

Details

The option (`opt`) can be:

- 1 = the *iid* column in FAM (default),
- 2 = formated as *fid*.*iid*,
- 0 = nothing
- -1 = numbering of individuals, decimal
- -2 = numbering of individuals, zero-padded fix-length decimal
- -3 = numbering of individuals, zero-padded fix-length hexedemical or, a vector of IDs to use.

Value

a vector of individual ID

Examples

```
pfx <- file.path(system.file("extdata", package="plinkFile"), "m20")
readIID(pfx,  1) # opt= 1: IID
readIID(pfx,  2) # opt= 2: FID.IID
readIID(pfx, -1) # opt=-1: number sequence
readIID(pfx, -2) # opt=-2: number sequence, fixed length, decimal
readIID(pfx, -3) # opt=-3: number sequence, fixed length, hexidemical
```

`readREL` *Read PLINK Binary REL matrix*

Description

A PLINK REL (Relatedness) matrix is represented by

.rel.bin: REL matrix in binary

.rel.id : FID and IID in text

A binary REL matrix is the result of PLINK `--make-rel bin`

Usage

```
readREL(pfx, fid = ".")
```

Arguments

`pfx`	prefix of the REL file set
`fid`	separate after family ID. (def=NULL, use IID only)

Value

relatedness matrix with row and column names set to sample ID.

Examples

```
pfx <- file.path(system.file("extdata", package="plinkFile"), "m20")
(readREL(pfx))
```

`readVCM` *Read Variant Count Matrix (VCM) accompanying a GCTA GRM*

Description

GRM (Genetic Relatedness Matrix) is the core formt of GCTA, which is a PLINK binary symmetric matrix with an extra variant count matrix (VCM), this function reads the VCM.

Usage

```
readVCM(pfx, fid = NULL)
```

Arguments

`pfx`	prefix of GRM file set
`fid`	seperate after family ID (def=NULL, use IID only)

Value

matrix of variant count with sample ID in row and column names.

Examples

```
pfx <- file.path(system.file("extdata", package="plinkFile"), "m20")
(readVCM(pfx))
```

`readVID` *read variant ID*

Description

Generate variant ID automatically, or based on a *bim* file.

Usage

```
readVID(bim, opt = NULL, vfr = NULL, vto = NULL)
```

Arguments

`bim`	prefix or name of a PLINK file, or data frame from a BIM file.
`opt`	option (def=1: the 2nd column in BIM file).
`vfr`	variant-wise, from where to read? (index/proportion, def=1).
`vto`	varinat-wise, to where then stop? (index/proportion, def=P).

Details

The option (`opt`) can be:

- 1 = the 2nd column in *pfx*.bim (default),
- 2 = formated as %CHR(02d):%BPS(09d),
- 3 = formated as %CHR(02d):%BPS(09d)_AL1(s)_AL2(s)
- 0 = nothing
- -1 = numbering of variants, decimal
- -2 = numbering of variants, zero-padded, fixed length decimal
- -3 = numbering of variants, zero-padded, fixed length hexedemical
- or, a vector of IDs to use.

Value

a vector of variant ID

Examples

```
# read variant ID
pfx <- file.path(system.file("extdata", package="plinkFile"), "m20")

# opt=1: 2nd column in the BED file (default)
vid <- readVID(pfx, 1); head(vid); tail(vid)

# opt=2: format by position
vid <- readVID(pfx, 2); head(vid); tail(vid)
```

```
# opt=3: format by position and alleles
vid <- readVID(pfx, 3); head(vid); tail(vid)

# opt=-1: number sequence
vid <- readVID(pfx, -1); head(vid); tail(vid)

# opt=-2: number sequence, fixed length, decimal
vid <- readVID(pfx, -2); head(vid); tail(vid)

# opt=-3: number sequence, fixed length, hexidemical
vid <- readVID(pfx, -3); head(vid); tail(vid)
```

---

saveBED *Save BED file*

Description

Save a R matrix into a PLINK BED file.

Usage

```
saveBED(pfx, bed, quiet = TRUE)
```

Arguments

pfx	prefix of the output file set, in PLINK1 BED format.
bed	N x P genotype matrix
quiet	do not report (def=TRUE)

Details

This is meant for genotype small enough to fit into system memory. The size of R matrix is 16 times the size of the BED file.

---

saveBSM *Save Symmetric Matrix to Binary*

Description

Save symmetric matrix to a binary core file (.bin), and a text file of IDs (.id), recognizable by PLINK.

Usage

```
saveBSM(pfx, x, ltr = TRUE, diag = TRUE, unit = 4L, fid = ".")
```

Arguments

pfx	prefix of output files
x	symmetric matrix to save
ltr	store the lower triangle only? (def=TRUE)
diag	save diagnal? (def=TRUE) ignored if ltr is FALSE.
unit	numerical unit, (def=4, single precision)
fid	separator between FID and IID (def=".").

Examples

```
pfx <- file.path(system.file("extdata", package="plinkFile"), "m20.rel")
rel <- readBSM(pfx)  # relatedness kernel matrix
re2 <- rel^2         # 2nd order polynomial kernel

tmp <- tempdir()
dir.create(tmp, FALSE)
out <- file.path(tmp, 'm20.re2')
saveBSM(out, re2)    # save the polynomial kernel
dir(tmp)             # show new files, then clean up
unlink(tmp, recursive=TRUE)
```

---

saveGRM *Save symmetic matrix to GCTA GRM format.*

Description

GRM (Genetic Relatedness Matrix) is the core formt of GCTA, this function saves a R symmetric matrix to a file set recgnizable by GCTA.

Usage

```
saveGRM(pfx, grm, vcm = NULL, fid = NULL)
```

Arguments

pfx	prefix of data files
grm	genome relatedness matrix to save
vcm	variant counts matrix to save (def=1).
fid	separator after family ID. (def=NULL)

Details

Three files will be saved:

.grm.bin : genetic relatedness matrix in binary

.grm.id : FID and IID for N individuals in text

.grm.N.bin : variant count matrix (VCM) in binary

FID and IID will be generated if the `grm` to be saved has no row names.

When save the `vcm`, if a single number is given, this number is used as the variant count for all entries in the GRM.

`saveGRM` is useful in exporting customized kinship matrices (such as a Gaussian or a Laplacian kernel) to a GRM acceptable by GCTA, which are not supported by GCTA's own GRM builder.

Examples

```
pfx <- file.path(system.file("extdata", package="plinkFile"), "m20")
gmx <- readBED(pfx)  # read genotype matrix from PLINK BED.
gmx <- scale(gmx)    # standardize
tmp <- tempdir()     # for example outputs
dir.create(tmp, FALSE)

# kinship matrix as Gaussian kernel, built from the first 10 variants
gmx.gau <- gmx[, +(1:10)]                # the first 10 variants
not.na.gau <- tcrossprod(!is.na(gmx.gau)) # variant count matrix
kin.gau <- exp(as.matrix(-dist(gmx.gau, "euc")) / not.na.gau)
print(kin.gau)                           # the Gaussian kernel
out.gau <- file.path(tmp, "m20.gau")
saveGRM(out.gau, kin.gau, not.na.gau)     # gau.grm.* should appear

# kinship matrix as Laplacian kernel, built without the first 10 variants
gmx.lap <- gmx[, -(1:10)]                # drop the first 10 variants
not.na.lap <- tcrossprod(!is.na(gmx.lap)) # variant count matrix
kin.lap <- exp(as.matrix(-dist(gmx.lap, "man")) / not.na.lap)
out.lap <- file.path(tmp, "m20.lap")
print(kin.lap)                           # the Laplacian kernel
saveGRM(out.lap, kin.lap, not.na.lap)     # lap.grm.* should appear

# merge kinship in R language for a radius based function kernel matrix
not.na.rbf <- not.na.gau + not.na.lap
kin.rbf <- (kin.gau * not.na.gau + kin.lap * not.na.lap) / not.na.rbf
print(kin.rbf)
out.rbf <- file.path(tmp, "m20.rbf")
saveGRM(out.rbf, kin.rbf, not.na.rbf)     # rbf.grm.* should appear

# show saved matrices, then clean up
dir(tmp, "(gau|lap|rbf)")
unlink(tmp, recursive=TRUE)
```

scanBED *travers variants in a PLINK1 BED fileset*

Description

Sequentially visits variants in a PLINK1 BED fileset with a stepping window matrix, and process each window matrix with user scripts either in function or expression form, meant for data to big to fit in the memory.

To read the entire BED into a R matrix, use `[readBED]()` instead.

Usage

```
scanBED(
  pfx,
  FUN,
  ...,
  win = 1,
  iid = 1,
  vid = 1,
  vfr = NULL,
  vto = NULL,
  buf = 2^24,
  simplify = TRUE
)

loopBED(
  pfx,
  EXP,
  GVR = "g",
  win = 1,
  iid = 1,
  vid = 1,
  vfr = NULL,
  vto = NULL,
  buf = 2^24,
  simplify = TRUE
)
```

Arguments

pfx	prefix of PLINK BED.
FUN	a function to process each window of variants;
...	additional argument for FUN when scanBED is used.
win	reading window size (def=100 variants per window)
iid	option to read N IID as row names (def=1, see readIID()).

vid	option to read P VID as col names (def=1, see readVID()).
vfr	variant-wise, from where to read (number/proportion, def=1)?
vto	varinat-wise, to where then stop (number/proportion, def=P)?
buf	buffer size in byptes (def=2^24, or 16 MB).
simplify	try simplifying the results into an array, or leave them in a list, or specify a function to simplify the said list.
EXP	a R expression to evaluate with each window of variants;
GVR	a R variable name to assign the window to (def="g").

Value

results of all windows processed by the user script.

Functions

- scanBED(): apply a function to variants in a PLINK1 BED fileset

  Travers P variants via a sliding window while calling a function on each window of variants without side effects on the calling environment, mimicking various R apply utilities.

- loopBED(): evaluate an expression on variants in a PLINK1 BED

  Travers P variants via a sliding window and evaluate an R expression given each window of variants, with side effects on the calling environment, mimicking the syntax of R for loop.

BED PLINK1 Binary Pedigree fileset

A popular format to store biallelic dosage genotype, with three files,

- *pfx*.fam: text table for N individuals, detailed in readFAM;
- *pfx*.bim: text table for P variants, detailed in readBIM;
- *pfx*.bed: transposed genotype matrix (P x N) in binary format.

The triplets are commonly referred by the shared prefix (pfx), e.g., the X chromosome represented by "chrX.bed", "chrX.fam", and "chrX.bim" are refered by "chrX".

The binary file "*pfx*.bed" represent each dosage value with two bits - just enough to encode all four possiblities: 0, 1, or 2 alleles, or missing.

The number of variants (P) and samples (N) equals to the number of lines in text file "*pfx*.bim" and "*pfx*.fam", respectively.

For the detailed specification of PLINK1 BED genotype format, see the lagecy PLINK v1.07 page at: \ https://zzz.bwh.harvard.edu/plink/binary.shtml. \ For the modern use and management of PLINK1 BED, see the PLINK v1.9 page: \ https://www.cog-genomics.org/plink/1.9/input#bed.

detailed arguments

- `win`: visiting window size.

  the number of variants per window, that is, the number of columns in each window matrix passed to the user script.

  For example, a size one window means the user script will be dealing with only one variant at a time, received from in a matrix of a single column – a manner similar to genome wide association analysis (GWAS). However, a larger, multi-variant window coupled with R language's vector and matrix syntax can significantly boost efficiency.

  The default size is 1000 variants / columns per window.

- `buf`: buffer size in bytes

  a large buffer reduces the frequency of hard disk visits when traversing a PLINK1 BED file, which in turn reduces non-computation overhead.

  The default size is `2^24` bytes, or 16 MB.

- `simplify`:

  when FALSE: resuts of user script processing each window of variants are returned in a list;

  when TRUE, use `simplify2array` to put the results into an array, if it fails, fallback and return a list.

  when a function is specified, it is then used to simplify the results, if an execption is thrown, fallback and return a list.

  e.g., the window script returns a data frame of estimate, standard error, t-statistic, and p-value for each variant, `simplify = rbind` to combine results of all windows into one data frame of `P` rows and four columns of statistics.

genotype context

context infomation such the number of variants and samples are updated in the window processing environment to ease user scripting, which includes:

- `.i`: indies of variants in the current visiting window;
- `.p`: number of variants in the current visiting window.
- `.P`: total number of variants;
- `.w`: index of the current window;
- `.W`: total number of windows to go through;
- `.N`: number of individuals.
- `.b`: index of the current buffer.
- `.B`: number of buffers to be swapped.

e.g. (1) print percentage progress with `print(.w / .W * 100);` \ e.g. (2) use `inf <- readBIM(pfx)` to read the table of variants before the window visits, later use `inf[.i, ]` to access meta-data for variants in each window.

See Also

`[readBED]`

Examples

```
## traverse genotype, apply R function without side effects
pfx <- file.path(system.file("extdata", package="plinkFile"), "000")
ret <- scanBED(pfx, function(g)
{
    .af <- colMeans(g, na.rm=TRUE) / 2
    maf <- pmin(.af, 1 - .af)
    mis <- colSums(is.na(g)) / .N
    pct <- round(.w / .W * 100, 2)
    cbind(buf=.b, wnd=.w, idx=.i, MAF=maf, MIS=mis, PCT=pct)
},
vfr=NULL, vto=NULL, win=13, simplify=rbind, buf=2^18)
head(ret)
tail(ret)

## traversing genotype, evaluate R expression with side effects
pfx <- file.path(system.file("extdata", package="plinkFile"), "000.bed")
ret <- list() # use side effect to keep the result of each window.
loopBED(pfx,
{
    af <- colMeans(gt, na.rm=TRUE) / 2
    sg <- af * (1 - af)
    ret[[.w]] <- cbind(wnd=.w, alf=af, var=sg)
},
win=13, GVR="gt", vid=3, buf=2^18)
head(ret)
tail(ret)
```

`testReadBED` *Test BED Reader*

Description

Read m20 (bed, bim, and fam) under "extdata" and compare with the content in text file "i10.txt" converted from m20 by PLINK.

Usage

```
testReadBED()
```

`testReadBSM` *Test Genetic Relatedness Matrix Reader*

Description

Compare the read from genetic relatedness matrix created from the same genome segment but stored in different shapes and types.

Usage

```
testReadBSM()
```

Index