Package: pgen2gds (via r-universe)

June 14, 2026

Type Package

Title Format Conversion from PLINK2 PGEN to GDS

Version 0.99.3

Date 2026-06-12

Depends methods, gdsfmt (>= 1.24.0)

Imports SeqArray (>= 1.49.6), pgenlibr

LinkingTo gdsfmt

Suggests parallel, digest, crayon, GenomicRanges, testthat (>= 3.0.0), knitr, rmarkdown, BiocStyle, BiocGenerics

Description Provides functions to convert files from the PLINK2 pgen format to SeqArray GDS.

License GPL-3 + file LICENSE

VignetteBuilder knitr

BugReports https://github.com/zhengxwen/pgen2gds/issues

URL https://github.com/zhengxwen/pgen2gds

biocViews Infrastructure, DataImport, Genetics

Config/pak/sysreqs zlib1g-dev

Repository https://bioc.r-universe.dev

Date/Publication 2026-06-12 05:33:13 UTC

RemoteUrl https://github.com/bioc/pgen2gds

RemoteRef HEAD

RemoteSha e52625a6d881236eee447f8dba5f4c95da6152b3

Contents

seqPGEN2GDS *Reformat PLINK2 PGEN files*

Description

Reformats PLINK2 pgen files to GDS format.

Usage

```
seqPGEN2GDS(pgen.fn, pvar.fn=NULL, psam.fn=NULL, out.gdsfn,
    compress.geno="LZMA_RA", compress.annot="LZMA_RA", variant.sel=NULL,
    sample.sel=NULL, start=1L, count=NA_integer_,
    ignore.chr.prefix=c("chr", "0"), reference=NULL, optimize=TRUE,
    digest=TRUE, parallel=FALSE, balancing=TRUE, verbose=TRUE)
```

Arguments

pgen.fn	a file name for the pgen file
pvar.fn	a file name for the pvar file, or NULL to use the default
psam.fn	a file name for the psam file, or NULL to use the default
out.gdsfn	the file name of output GDS file
compress.geno	the compression method for "genotype"; optional values are defined in the function add.gdsn
compress.annot	the compression method for the GDS variables, except "genotype"; optional values are defined in the function add.gdsn
variant.sel	NULL for no variant selection, a logical vector or a numeric vector to specify the variant selection
sample.sel	NULL for no sample selection, a logical vector or a numeric vector to specify the sample selection
start	the starting variant if importing part of the pgen file
count	the maximum count of variant if importing part of the pgen file, NA_integer_ or any non-positive value indicates importing to the end
ignore.chr.prefix	a vector of character, indicating the prefix of chromosome which should be ignored, e.g., "chr"; it is not case-sensitive
reference	genome reference, like "GRCh37", "GRCh38"; it is not specified if reference=NULL
optimize	if TRUE, optimize the access efficiency by calling cleanup.gds
digest	a logical value (TRUE/FALSE) or a character (e.g., "md5"); add hash codes to the GDS file if TRUE or a digest algorithm is specified
parallel	FALSE (serial processing), TRUE (parallel processing), a numeric value indicating the number of cores, or a cluster object for parallel processing; parallel is passed to the argument cl in seqParallel, see seqParallel for more details

balancing	whether to perform workload balancing or not, only applicable when multiple cores are used; if NA, use TRUE as a default until getOption("seqarray.balancing") is set and not TRUE
verbose	if TRUE, show information

Value

Return the file name of SeqArray GDS file with an absolute path.

Author(s)

Xiuwen Zheng

References

https://www.cog-genomics.org/plink/2.0/

See Also

seqReadPVAR

Examples

```
pgen_fn <- system.file("extdata", "plink2_gen.pgen", package="pgen2gds")

seqPGEN2GDS(pgen_fn, out.gdsfn="test.gds")

# delete the temporary file
unlink("test.gds", force=TRUE)
```

---

seqReadPVAR *Read PLINK2 pvar file*

Description

Read PLINK2 pvar file for variants

Usage

```
seqReadPVAR(pvar, sel=NULL)
```

Arguments

pvar	a file name of a pvar file (from NewPvar), or a pvar object, which can be queried for variant IDs and allele codes
sel	NULL, a logical vector or a numeric vector for specifying the variants; NULL for including all variants

Value

Return a data frame with the columns `chrom`, `pos`, `allele` and `rsid`.

Author(s)

Xiuwen Zheng

References

https://www.cog-genomics.org/plink/2.0/

See Also

seqPGEN2GDS

Examples

```
pvar_fn <- system.file("extdata", "plink2_gen.pvar", package="pgen2gds")

head(seqReadPVAR(pvar_fn))
```

Index