Package 'GDSAnnotator'

October 30, 2026

Type Package

Title Whole-genome variant annotation data manipulation using GDS files

Version 0.99.2

Depends R (>= 4.6.0), gdsfmt (>= 1.32.0), SeqArray (>= 1.53.2)

Imports methods, utils, IRanges, GenomicRanges, S4Vectors

Suggests VariantAnnotation, BiocStyle, knitr, rmarkdown, testthat (>= 3.0.0)

Description Fast and memory-efficient manipulation of whole-genome variant annotation data stored in Genomic Data Structure (GDS) files. The annotated GDS (aGDS) format introduced by FAVOR stores the FAVOR annotations of the variants together with the genotypes in one SeqArray GDS file. This package generalizes the format to annotation GDS files of any source, with or without genotypes. Annotation resources and annotator outputs, including FAVOR, gnomAD, Ensembl VEP and SnpEff, are converted into compact SeqArray-compatible GDS files with random access. Variants can be given as chr-pos-ref-alt strings, data frames, GRanges objects, VCF files or objects, or SeqArray GDS files. They are annotated by querying the annotation files, and the annotations can be written back into a genotype GDS file. Summary statistics of the variants and their annotations are computed in a single pass over a file and rendered as a VEP/SnpEff-like report. The package is intended for large-scale whole-genome sequencing studies.

License GPL-3

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

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

VignetteBuilder knitr

Encoding UTF-8

ByteCompile TRUE

biocViews Infrastructure, DataRepresentation, DataImport, Sequencing, WholeGenome, Genetics, Annotation, VariantAnnotation, GenomeAnnotation, GenomicVariation

git_url https://git.bioconductor.org/packages/GDSAnnotator

git_branch devel

git_last_commit 40f9a3d

git_last_commit_date 2026-10-18

Repository Bioconductor 3.24

Date/Publication 2026-10-29

Author Xiuwen Zheng [aut, cre] (ORCID:
<https://orcid.org/0000-0002-1390-0708>),
Yuantao (Raven) Chen [ctb],
Hufeng Zhou [ctb]

Maintainer Xiuwen Zheng <xiuwen.zheng@abbvie.com>

Contents

GDSAnnotator-package 2
GDSAnnotator-extdata 4
seqAnnotate-methods 5
seqAnnotGeneTable 9
seqAnnotList 10
seqAnnotReport 11
seqAnnotStat 12
seqANNOVAR2GDS 14
seqFAVOR2GDS 16
seqGnomAD2GDS 17
seqSnpEff2GDS 18
seqValueCounts 20
seqVEP2GDS 21

Index **23**

`GDSAnnotator-package` *Whole-genome variant annotation data manipulation using GDS files*

Description

Fast and memory-efficient manipulation of whole-genome variant annotation data stored in Genomic Data Structure (GDS) files.

Details

Package:	GDSAnnotator
Type:	Package
License:	GPL-3
Depends:	gdsfmt, SeqArray

The package stores per-variant annotations in SeqArray GDS files, one chromosome per file, under the node "annotation/info". The columnar, typed and block-compressed layout is compact and supports random access, so a small set of variants can be annotated without reading the whole database.

Storing variant annotations in a GDS file was introduced by FAVOR with the annotated GDS (aGDS) format, in which the FAVOR functional annotations of the variants of a study are added to the SeqArray GDS file holding the genotypes. This package generalizes the format to annotation GDS files of any source, with or without genotypes: a whole-genome annotation database is kept as one GDS file per chromosome and queried by position and alleles, and the annotations of the variants of a study are written into its genotype GDS file (with the FAVOR annotations, this produces an aGDS file).

- Import: `seqFAVOR2GDS`, `seqGnomAD2GDS`, `seqVEP2GDS`, `seqSnpEff2GDS` and `seqANNOVAR2GDS` convert the annotation resources and the outputs of the annotators to GDS files.
- Annotate: `seqAnnotate` annotates variants given as "chr-pos-ref-alt" strings, a `data.frame` or `DataFrame`, a `GRanges` object, a `VCF` object or file (`seqAnnotateVCF`) or a SeqArray GDS file (`seqAnnotateGDS`), and can write the annotations back into a GDS file block by block.
- Explore & summarize: `seqAnnotList` lists the annotations in a file, `seqValueCounts` tabulates their values, and `seqAnnotStat`, `seqAnnotReport` and `seqAnnotGeneTable` produce VEP/SnpEff-like summary statistics and reports in a single pass over the file.

Author(s)

Xiuwen Zheng <xiuwen.zheng@abbvie.com>

References

Zhou H, Arapoglou T, Li X, et al. FAVOR: functional annotation of variants online resource and annotator for variation across the human genome. Nucleic Acids Res. 2023;51(D1):D1300-D1311. DOI: 10.1093/nar/gkac966.

Zheng X, Gogarten SM, Lawrence M, et al. SeqArray: a storage-efficient high-performance data format for WGS variant calls. Bioinformatics. 2017;33(15):2251-2257. DOI: 10.1093/bioinformatics/btx145.

See Also

`seqOpen`, `seqGetData`, `seqVCF2GDS`

Examples

```
# FAVOR essential dataset of chromosome 22 (subset)
annot_gds <- system.file("extdata", "favor_chr22_sub.gds",
    package="GDSAnnotator")

# the annotations in the file
seqAnnotList(annot_gds)

# annotate two variants
seqAnnotate(c("22-10510007-T-C", "22-10510038-T-C"), annot_gds,
    varnm=c("rsid", "cadd_phred", "linsight"))
```

GDSAnnotator-extdata *Example data files*

Description

Small example files in the `extdata` folder of the package, used in the examples, the unit tests and the vignette. All of them are subsets of chromosome 22 (GRCh38).

Format

`favor_chr22_sub.csv.gz` 1,000 consecutive variants at the start of chromosome 22 (positions 10,510,001-10,510,333) from the FAVOR essential database, in the CSV format distributed by FAVOR: the columns "variant_vcf", "chromosome", "position", "ref_vcf" and "alt_vcf" identify the variants, and the other 30 columns are FAVOR annotation scores (aPC scores, CADD, LINSIGHT, GeneHancer, rDHS, ...).

`favor_chr22_sub.gds` the CSV file above converted with `seqFAVOR2GDS(root="")`.

`favor_csv_header.csv` the column names, types and descriptions of the FAVOR essential database CSV files, used by `seqFAVORTar2GDS`; compiled from the FAVOR documentation.

`gnomad.genomes.v4.chr22_sub.vcf.gz` **(and** `.csi`**)** 150 variants (positions 10,510,033-10,510,756) from the gnomAD v4.1 genomes sites VCF file of chromosome 22, with the header and the INFO fields unchanged.

`gnomad.genomes.v4.chr22_sub.gds` the VCF file above converted with `seqGnomAD2GDS`.

`example_wgs_sites_chr22_vep.vcf.gz` **(and** `.csi`**)** 250 variant sites annotated with Ensembl VEP v113 (cache GRCh38.p14, `--everything --flag_pick --nearest gene`, see the "##VEP" and "##VEP-command-line" header lines).

`example_wgs_sites_chr22_vep.gds` the VCF file above converted with `seqVEP2GDS`, the CSQ field being split into the sub-fields under "annotation/info/CSQ.list".

`example_wgs_sites_chr22_snpeff.vcf.gz` **(and** `.csi`**)** 200 variant sites annotated with SnpEff 5.2f (database GRCh38.p14, `-lof`, see the "##SnpEffVersion" and "##SnpEffCmd" header lines).

`example_wgs_sites_chr22_snpeff.gds` the VCF file above converted with `seqSnpEff2GDS`, the ANN, LOF and NMD fields being split into the sub-fields under "annotation/info/ANN.list", "LOF.list" and "NMD.list".

`example_wgs_sites_chr22_annovar.vcf.gz` 10 variant sites annotated with ANNOVAR (`table_annovar.pl -vcfinput`, with the refGeneWithVer, gnomAD exome and dbNSFP-based protocols, see the INFO header lines).

`vep_output_format.csv`, `snpeff_output_format.csv`, `annovar_output_format.csv` the sub-fields or INFO fields written by Ensembl VEP, SnpEff and ANNOVAR, with their descriptions and R types, compiled from the documentation of the three tools; used by the conversion functions.

`so_consequence.csv` the Sequence Ontology consequence terms ranked by severity, with the impact category, the functional class and the genomic region, compiled from the Ensembl variation documentation and the SnpEff documentation; used by `seqAnnotStat`.

Details

The VEP, SnpEff and ANNOVAR files are derived from the same sites-only VCF file of chromosome 22 (a set of imputed variant sites written by Minimac4 v1.0.2, with no sample and no genotype): the variant sites were annotated by each tool, and a subset of the annotated variants was kept.

The GDS files are generated from the text files by the script `make_extdata.R` in the `scripts` folder of the installed package, where `README.md` describes the provenance of each file in more detail.

Source

FAVOR: https://favor.genohub.org.

gnomAD v4.1: https://gnomad.broadinstitute.org/downloads, released under the CC0 1.0 license.

Ensembl VEP: https://www.ensembl.org/info/docs/tools/vep/; SnpEff: https://pcingola.github.io/SnpEff/; ANNOVAR: https://annovar.openbioinformatics.org/.

References

Zhou H, Arapoglou T, Li X, et al. FAVOR: functional annotation of variants online resource and annotator for variation across the human genome. Nucleic Acids Res. 2023;51(D1):D1300-D1311. DOI: 10.1093/nar/gkac966.

See Also

seqFAVOR2GDS, seqGnomAD2GDS, seqVEP2GDS, seqSnpEff2GDS, seqANNOVAR2GDS

Examples

```
# the example files
list.files(system.file("extdata", package="GDSAnnotator"))

# the annotations in the FAVOR example file
seqAnnotList(system.file("extdata", "favor_chr22_sub.gds",
    package="GDSAnnotator"))

# the script generating the GDS files
cat(readLines(system.file("scripts", "make_extdata.R",
    package="GDSAnnotator")), sep="\n")
```

---

seqAnnotate-methods *Annotate a list of variants*

Description

Annotate a list of variants using the annotation stored in GDS files.

Usage

```
## S4 method for signature 'character'
seqAnnotate(object, annot_gds, varnm, split="[-_:]",
    ..., verbose=TRUE)
## S4 method for signature 'data.frame'
seqAnnotate(object, annot_gds, varnm, col_chr="chr",
    col_pos="pos", col_ref="ref", col_alt="alt", ..., verbose=TRUE)
## S4 method for signature 'DataFrame'
seqAnnotate(object, annot_gds, varnm, col_chr="chr",
    col_pos="pos", col_ref="ref", col_alt="alt", ..., verbose=TRUE)
## S4 method for signature 'GRanges'
seqAnnotate(object, annot_gds, varnm, ..., verbose=TRUE)
## S4 method for signature 'GRangesList'
seqAnnotate(object, annot_gds, varnm, ...,
    verbose=TRUE)
## S4 method for signature 'IRanges'
seqAnnotate(object, annot_gds, varnm, chr, ...,
    verbose=TRUE)
## S4 method for signature 'ANY'
seqAnnotate(object, annot_gds, varnm, ..., verbose=TRUE)
## S4 method for signature 'SeqVarGDSClass'
seqAnnotate(object, annot_gds, varnm,
    add_to_gds=FALSE, no_sample=TRUE, root="", cleanup=FALSE,
    bsize=1000000L, ..., verbose=TRUE)
seqAnnotateGDS(gds_fn, annot_gds, varnm, add_to_gds=FALSE, no_sample=TRUE,
    root="", cleanup=TRUE, bsize=1000000L, ..., verbose=TRUE)
seqAnnotateVCF(vcf_fn, annot_gds, varnm, ..., verbose=TRUE)
```

Arguments

object	the variants to be annotated: a character vector of variant identifiers combining chromosome, position, reference and alternative alleles (e.g., "22-23413241-A-G", see split); a data.frame or DataFrame with the columns col_chr, col_pos, col_ref and col_alt; a GRanges, GRangesList or IRanges object, in which case all the variants located in the ranges are annotated; a VCF object of the package **VariantAnnotation** (CollapsedVCF or ExpandedVCF), annotated with one row per alternative allele by the method for ANY; or a SeqArray GDS object (SeqVarGDSClass)
gds_fn	character for a SeqArray GDS file name
vcf_fn	character for a VCF or BCF file name, read by readVcf
annot_gds	a character vector for GDS file names or a list of opened GDS files (SeqVarGDSClass objects), one chromosome per file; if a variant is found in more than one file, the annotation is taken from the first file
varnm	a character vector for GDS node names without the prefix "annotation/info/"; ":id" for accessing "annotation/id", ":qual" for "annotation/qual", ":filter" for "annotation/filter"; if it is missing, to use all annotations in the first file of annot_gds; if varnm=character() or NULL, return which variants are selected; see details
col_chr	the column name for chromosome
col_pos	the column name for position

col_ref	the column name for the reference alleles
col_alt	the column name for the alternative alleles
chr	a character vector of chromosome(s), used with an IRanges object; passed to seqSetFilter
add_to_gds	logical or character for an output file name; if FALSE, return DataFrame for annotation; if TRUE, add annotation to the input GDS file; if add_to_gds is a file name, output the annotation to a new GDS file
no_sample	logical, when add_to_gds is a file name and no_sample=TRUE, the output GDS file excludes per-sample data
root	character, to save the annotated GDS nodes to a root node (using "annotation/info/root"); if root="", use "annotation/info" instead
cleanup	if TRUE, call cleanup.gds on the output GDS file when add_to_gds is a file name
bsize	the block size, i.e., the number of variants to be mapped and written to the output GDS file at a time, when add_to_gds is TRUE or a file name; for in-place annotation, both variant matching and annotation data are processed block by block, so memory usage does not depend on the total number of variants
split	regular expression to split the character vector in object by using strsplit, using "-", "_" or ":" by default, e.g., "22-23413241-A-G" will be split into "22", "23413241", "A" and "G"
verbose	if TRUE, show information
...	additional arguments passed to specific methods, e.g., from seqAnnotateGDS() to the SeqVarGDSClass method, and from the VCF method to the DataFrame method

Details

If varnm=character() or NULL, seqAnnotate returns a DataFrame object with two columns 'file_idx' and 'variant_idx' to indicate where the variants are found, and NA in 'variant_idx' if the variant is not found.

seqAnnotateGDS() opens the GDS file gds_fn and calls the SeqVarGDSClass method. seqAnnotateVCF() reads the VCF file with readVcf and annotates the VCF object. The package **VariantAnnotation** should be installed for the VCF files and objects: it is a suggested package loaded on demand, so the class VCF is not used in a method signature, and a VCF object is handled by the method for ANY, which checks the class of the object.

A VCF object is annotated for each alternative allele of each row: a CollapsedVCF with multi-allelic sites gives one row per alternative allele, a row without any alternative allele (ALT is ".") is skipped, so the output can have more or fewer rows than the VCF object, and expand can be used first to get one alternative allele per row. The output is the same for a CollapsedVCF and for its expanded version.

When the annotations are written to a GDS file (add_to_gds is TRUE or a file name), the variants are matched and the annotation data are loaded and written block by block (see bsize), so that the memory usage does not depend on the total number of variants. The genotype GDS file then holds the annotations of its variants under "annotation/info" (or under "annotation/info/root"); with the annotations of FAVOR, this produces an annotated GDS (aGDS) file, see seqFAVOR2GDS.

Value

Returns a DataFrame with one row per input variant (one row per variant located in the ranges for a GRanges, GRangesList or IRanges input, and one row per alternative allele for a VCF input) and one column per annotation in varnm; the annotations of a variant not found in annot_gds are NA. If varnm=character() or NULL, the DataFrame has the two columns 'file_idx' and 'variant_idx' instead, see details. When add_to_gds is TRUE or a file name, the annotations are written to the GDS file and NULL is returned invisibly.

Author(s)

Xiuwen Zheng

See Also

seqAnnotList, seqValueCounts, seqFAVOR2GDS, seqVEP2GDS

Examples

```
# FAVOR essential dataset of chromosome 22 (subset)
annot_gds <- system.file("extdata", "favor_chr22_sub.gds",
    package="GDSAnnotator")

# list the annotations in the gds file
(ann <- seqAnnotList(annot_gds))

varnm <- ann$name  # use all annotation
snp <- c("22-10510007-T-C", "22-10510038-T-C", "22-10510282-G-C",
    "22-10510303-G-C"  # a fake SNP, return NA
    )
seqAnnotate(snp, annot_gds, varnm)

# a subset of the annotations, from a data.frame
df <- data.frame(chr="22", pos=c(10510007L, 10510038L, 10510282L),
    ref=c("T", "T", "G"), alt=c("C", "C", "C"))
seqAnnotate(df, annot_gds, varnm=c("rsid", "cadd_phred", "linsight"))

# all the variants located in the genomic ranges
gr <- GenomicRanges::GRanges(seqnames = c("chr22", "chr22"),
    ranges = IRanges::IRanges(start = c(10510100, 10510225),
        end = c(10510125, 10510250)))
seqAnnotate(gr, annot_gds, varnm)

# the variants of a VCF file, one row per alternative allele (the package
#   VariantAnnotation should be installed)
vcf_fn <- system.file("extdata", "gnomad.genomes.v4.chr22_sub.vcf.gz",
    package="GDSAnnotator")
if (requireNamespace("VariantAnnotation", quietly=TRUE))
{
    res <- seqAnnotateVCF(vcf_fn, annot_gds, varnm=c("rsid", "cadd_phred"))
    print(head(res[!is.na(res$cadd_phred), ]))

    # or from a VCF object of the package VariantAnnotation
    vcf <- VariantAnnotation::readVcf(vcf_fn)
    res2 <- seqAnnotate(vcf, annot_gds, varnm=c("rsid", "cadd_phred"))
    print(identical(res, res2))
}
```

```
# the variants of a SeqArray GDS file (the gnomAD sites of the same region)
gds_fn <- system.file("extdata", "gnomad.genomes.v4.chr22_sub.gds",
    package="GDSAnnotator")
res <- seqAnnotateGDS(gds_fn, annot_gds, varnm=c("rsid", "cadd_phred"))
head(res[!is.na(res$cadd_phred), ])

# write the annotations to a new GDS file
out_fn <- tempfile(fileext=".gds")
seqAnnotateGDS(gds_fn, annot_gds, varnm=c("rsid", "cadd_phred"),
    add_to_gds=out_fn)
f <- seqOpen(out_fn)
head(seqGetData(f, "annotation/info/cadd_phred"))
seqClose(f)

# delete the temporary file
unlink(out_fn, force=TRUE)
```

`seqAnnotGeneTable` *Per-gene annotation counts*

Description

Return the number of annotation records of each gene, broken down by impact category.

Usage

```
seqAnnotGeneTable(stat)
```

Arguments

`stat`	a `SeqAnnotStat` object returned from `seqAnnotStat`

Details

The output is the equivalent of the "genes.txt" file of SnpEff. It is `NULL` if the GDS file has no gene annotation, e.g., if the file was not annotated by Ensembl-VEP or SnpEff.

Value

Returns a `data.frame` with one row per gene, sorted by the total count in decreasing order: the column `gene` for the gene symbol, `total` for the total number of annotation records, and one column for each impact category (HIGH, MODERATE, LOW and MODIFIER).

Author(s)

Xiuwen Zheng

See Also

`seqAnnotStat`, `seqAnnotReport`

Examples

```
# SnpEff-annotated variants of chromosome 22 (subset)
gds_fn <- system.file("extdata", "example_wgs_sites_chr22_snpeff.gds",
    package="GDSAnnotator")

st <- seqAnnotStat(gds_fn, verbose=FALSE)
head(seqAnnotGeneTable(st))
```

`seqAnnotList` *List annotations*

Description

List the annotations stored in the GDS file.

Usage

```
seqAnnotList(gdsfile)
```

Arguments

`gdsfile`	a `SeqVarGDSClass` object or a GDS file name

Value

Returns a `DataFrame` with the columns 'name', 'type' (R data type), 'trait' (data type defined in GDS) and 'description'.

Author(s)

Xiuwen Zheng

See Also

`seqAnnotate`

Examples

```
# gnomAD v4.1.0 dataset of chromosome 22
gds_fn <- system.file("extdata", "gnomad.genomes.v4.chr22_sub.gds",
    package="GDSAnnotator")

# list the annotations in the gds file
seqAnnotList(gds_fn)
```

seqAnnotReport *Variant annotation summary report*

Description

Generate a summary report of the variants and their annotations stored in a SeqArray GDS file, in HTML or Markdown format, similar to the reports of Ensembl-VEP and SnpEff.

Usage

```
seqAnnotReport(stat, out_fn="annot_summary.html",
    format=c("auto", "html", "markdown", "rmd"),
    title="GDSAnnotator Summary Report", parallel=FALSE, bsize=100000L,
    verbose=TRUE)
```

Arguments

stat	a SeqAnnotStat object returned from seqAnnotStat, or a SeqVarGDSClass object, a GDS file name or a character vector for multiple file names, for which seqAnnotStat is called first
out_fn	the file name of the output report
format	the output format: "html" and "markdown" are finished reports, ready to be read as they are, and their charts are drawn by the renderer (inline SVG in HTML, block characters in Markdown); "rmd" is instead the R Markdown *source* of a report, in which the counts are embedded as R code and the charts are drawn by barplot() only when the document is knitted, so it can be edited before rendering; "auto" (by default) takes the format from the file extension of out_fn: ".md" and ".markdown" give "markdown", ".Rmd" gives "rmd", and anything else gives "html" (the extension is matched case-insensitively)
title	the title of the report
parallel	the number of processes, passed to seqAnnotStat if stat is not a SeqAnnotStat object; see seqAnnotStat
bsize	block size, passed to seqAnnotStat if stat is not a SeqAnnotStat object
verbose	if TRUE, show information

Details

The three output formats show the same content: the report is first built as a format-independent list of sections, then rendered. None of them writes a companion image directory, i.e., the report is always a single file.

The HTML output is self-contained, using plain HTML tables and inline SVG charts, so it requires no JavaScript, no pandoc and no additional package, and it can be viewed offline.

The Markdown output uses GitHub-flavored pipe tables, with the bar charts drawn using block characters, so that it is also readable as plain text.

The R Markdown output embeds the counts in the document as R code, and the charts are drawn by barplot() when the document is knitted, e.g., by rmarkdown::render(). Use this format to restyle the charts, to add text of your own, or to include the report in a larger document. The code chunks are hidden by echo=FALSE in the setup chunk.

The report contains a general information section (the annotator version and command line taken from the VCF header, the number of variants, annotation records, genes and transcripts, and the variant rate), the variant-level sections (variants per chromosome, variant classes, base changes with the Ts/Tv ratio, InDel length distribution, alleles per variant, known vs. novel, FILTER and QUAL) and, if the file was annotated by Ensembl-VEP or SnpEff, the annotation sections (consequences for all records and for the most severe consequence per variant, region, impact, functional class with the Missense/Silent ratio, transcript biotypes, feature types, SIFT and PolyPhen predictions, the warnings reported by the annotator, and the most frequently annotated genes).

Value

Return the file name of the report with an absolute path.

Author(s)

Xiuwen Zheng

See Also

seqAnnotStat, seqAnnotGeneTable

Examples

```
# Ensembl-VEP annotated variants of chromosome 22 (subset)
gds_fn <- system.file("extdata", "example_wgs_sites_chr22_vep.gds",
    package="GDSAnnotator")

# HTML report
html_fn <- tempfile(fileext=".html")
seqAnnotReport(gds_fn, html_fn, title="Ensembl-VEP, chromosome 22",
    verbose=FALSE)

# Markdown report, reusing the statistics
st <- seqAnnotStat(gds_fn, verbose=FALSE)
md_fn <- tempfile(fileext=".md")
seqAnnotReport(st, md_fn, title="Ensembl-VEP, chromosome 22", verbose=FALSE)

cat(head(readLines(md_fn), 20), sep="\n")

# R Markdown source, to be knitted by rmarkdown::render()
rmd_fn <- tempfile(fileext=".Rmd")
seqAnnotReport(st, rmd_fn, title="Ensembl-VEP, chromosome 22", verbose=FALSE)

# remove the temporary files
unlink(c(html_fn, md_fn, rmd_fn), force=TRUE)
```

---

seqAnnotStat *Summary statistics of variant annotation*

Description

Calculate the summary statistics of the variants and their annotations stored in a SeqArray GDS file, in a single pass over the file.

Usage

```
seqAnnotStat(gdsfile, parallel=FALSE, bsize=100000L, verbose=TRUE)
## S3 method for class 'SeqAnnotStat'
print(x, ...)
```

Arguments

gdsfile	a SeqVarGDSClass object, a GDS file name or a character vector for multiple file names
parallel	FALSE (serial processing), TRUE (multicore processing), numeric value or other value; parallel is passed to the argument cl in seqParallel, see seqParallel for more details
bsize	block size, i.e., the number of variants to be processed at a time
verbose	if TRUE, show information
x	a SeqAnnotStat object
...	unused arguments

Details

All statistics are collected in a single call to seqBlockApply, so that the file is scanned only once no matter how many statistics are reported. Each statistic is stored as a named vector of counts, and the counts are additive: when gdsfile is a character vector of multiple file names (e.g., one GDS file per chromosome), the files are summarized one by one and the counters are merged into a single object. The memory usage is proportional to the number of distinct categories (e.g., the number of genes), not to the number of variants.

Since the counters are additive, parallel splits the variants between the processes, each process calculates its own counters, and the results are merged. Each counter is a named numeric vector, and its categories are stored in a fixed order (e.g., the chromosomes in numerical order, the impact categories from HIGH to MODIFIER, and the other categories in alphabetical order), so that the output does not depend on parallel or on bsize.

The variant-level statistics (the number of variants per chromosome, the variant classes, the base changes and the Ts/Tv ratio, the InDel length distribution, the number of alleles, known vs. novel, FILTER and QUAL) are derived from the nodes "chromosome", "position", "allele", "annotation/id", "annotation/qual" and "annotation/filter".

The annotation statistics are calculated when the GDS file contains the sub-fields of Ensembl-VEP ("annotation/info/CSQ.list", see seqVEP2GDS) or SnpEff ("annotation/info/ANN.list", see seqSnpEff2GDS). Both annotators report Sequence Ontology terms, and an internal table ("so_consequence.csv" in the "extdata" folder) is used to rank the consequences by severity, to classify them into impact categories (HIGH, MODERATE, LOW and MODIFIER), into functional classes (MISSENSE, NONSENSE and SILENT) and into genomic regions. The consequences are counted twice: once for every annotation record (e.g., one record per transcript), and once per variant using only the most severe consequence.

Value

Returns an object of class SeqAnnotStat, a list with the components: file, filesize, n_variant, source (the annotator), profile (the annotation sub-fields in use), header (the VCF header), counts (a list of named numeric vectors of counts) and created.

Author(s)

Xiuwen Zheng

See Also

seqAnnotReport, seqAnnotGeneTable, seqValueCounts, seqAnnotList

Examples

```
# SnpEff-annotated variants of chromosome 22 (subset)
gds_fn <- system.file("extdata", "example_wgs_sites_chr22_snpeff.gds",
    package="GDSAnnotator")

(st <- seqAnnotStat(gds_fn, verbose=FALSE))

# the most severe consequence of each variant
st$counts$cons_severe

# the impact of every annotation record
st$counts$impact

# transitions & transversions
st$counts$base_change

# using two processes: the counters are additive, so the result is the same
st2 <- seqAnnotStat(gds_fn, parallel=2L, verbose=FALSE)
identical(st$counts, st2$counts)
```

---

seqANNOVAR2GDS *Reformat ANNOVAR VCF*

Description

Convert ANNOVAR-annotated VCF output (from `table_annovar.pl`) to a SeqArray GDS file. Unlike VEP or SnpEff which pack annotations into a single pipe-delimited INFO field, ANNOVAR stores each annotation as a separate INFO key-value pair. This function imports the VCF via `seqVCF2GDS`, removes ANNOVAR bookkeeping fields (`ANNOVAR_DATE`, `ALLELE_END`), and optionally adds descriptions and type information for known ANNOVAR annotation columns.

Usage

```
seqANNOVAR2GDS(vcf_fn, out_fn, compress=c("LZMA", "ZIP", "none"),
    keep=TRUE, verbose=TRUE)
```

Arguments

`vcf_fn`	the path(s) of VCF file from ANNOVAR output (generated by `table_annovar.pl -vcfinput`)
`out_fn`	the file name of output SeqArray GDS file
`compress`	compression method (LZMA by default), `"LZMA"` has a higher compression ratio than `"ZIP"`

keep	if TRUE or NA, keep all ANNOVAR annotation nodes in the output GDS file; if FALSE, remove gene-based annotation nodes (Func.*, Gene.*, GeneDetail.*, ExonicFunc.*, AAChange.*) after import
verbose	if TRUE, show information

Details

ANNOVAR's table_annovar.pl with -vcfinput adds annotations as individual INFO fields in the VCF output, such as Func.refGeneWithVer, Gene.refGeneWithVer, ExonicFunc.refGeneWithVer, AAChange.refGeneWithVer, gnomAD_exome_AF, SIFT_score, CADD_phred, etc. Each annotation becomes a separate node under annotation/info/ in the output GDS file.

The function automatically removes the ANNOVAR_DATE and ALLELE_END bookkeeping fields that ANNOVAR adds to the VCF. Known ANNOVAR annotation fields are matched against an internal metadata table to add descriptions.

Value

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

Author(s)

Xiuwen Zheng

References

Wang K, Li M, Hakonarson H. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. Nucleic Acids Research, 38(16):e164, 2010. DOI: 10.1093/nar/gkq603.

See Also

seqVCF2GDS, seqVEP2GDS, seqSnpEff2GDS

Examples

```
# ANNOVAR annotation
vcf_fn <- system.file("extdata", "example_wgs_sites_chr22_annovar.vcf.gz",
    package="GDSAnnotator")

out_fn <- tempfile(fileext=".gds")
seqANNOVAR2GDS(vcf_fn, out_fn)

# list the annotations in the output file
seqAnnotList(out_fn)

# delete the temporary file
unlink(out_fn, force=TRUE)
```

seqFAVOR2GDS *Reformat FAVOR files*

Description

Convert FAVOR CSV files to a SeqArray GDS file.

Usage

```
seqFAVOR2GDS(csv_fn, out_fn, compress=c("LZMA", "ZIP", "none"),
    root="FAVOR", use_float32=TRUE, verbose=TRUE)
seqFAVORTar2GDS(tar_fn, out_fn, fn_in_tar=NULL,
    compress=c("LZMA", "ZIP", "none"), root="", use_float32=TRUE,
    block_size=100000L, tar_cmd=Sys.getenv("TAR"), verbose=TRUE)
```

Arguments

csv_fn	the path(es) of FAVOR csv file
tar_fn	the path of a FAVOR tar file containing csv files
out_fn	the file name of output SeqArray GDS file
compress	compression method (LZMA by default), "LZMA" has a higher compression ratio than "ZIP"
root	specify the GDS node, the annotation will be stored in "annotation/info/root"; if root="", "annotation/info" is used
use_float32	if TRUE, use 32-bit float to store floating-point numbers instead of 64-bit float
fn_in_tar	a character vector for the csv file names in the tar file
block_size	the block size (# of lines) for reading a csv file
tar_cmd	the path of TAR to the command to be used
verbose	if TRUE, show information

Details

The CSV files (which may be gzip-compressed) should have the columns "variant_vcf", "chromosome", "position", "ref_vcf" and "alt_vcf"; all the other columns are stored as annotations under "annotation/info" (or under "annotation/info/root" if root is not empty). The column types of the FAVOR essential database are defined in the internal table "favor_csv_header.csv" in the "extdata" folder.

seqFAVORTar2GDS() reads the CSV files directly from a tar archive (e.g., the per-chromosome files of the FAVOR essential database) block by block, so that it does not require extracting the archive nor loading a whole CSV file in memory; it calls the external program tar_cmd via pipe(), so a tar program should be available on the system.

Value

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

Author(s)

Xiuwen Zheng

References

Zhou H, Arapoglou T, ..., Lin X. FAVOR: Functional Annotation of Variants Online Resource and Annotator for Variation across the Human Genome. NAR 2022 Nov 9; gkac966. PMID: 36350676. DOI: 10.1093/nar/gkac966.

See Also

seqVCF2GDS

Examples

```
# FAVOR essential dataset of chromosome 22 (subset) in CSV format
csv_fn <- system.file("extdata", "favor_chr22_sub.csv.gz",
    package="GDSAnnotator")

out_fn <- tempfile(fileext=".gds")
seqFAVOR2GDS(csv_fn, out_fn)

# list the annotations in the output file
seqAnnotList(out_fn)

# a tar file of FAVOR CSV files (here, the CSV file above only), which is
#   read with the external program 'tar' (this example is run on Unix only)
if (.Platform$OS.type == "unix" && nzchar(Sys.which("tar")))
{
    tmp_dir <- tempfile("favor")
    dir.create(tmp_dir)
    writeLines(readLines(csv_fn), file.path(tmp_dir, "chr22_1.csv"))
    tar_fn <- tempfile(fileext=".tar.gz")
    old_wd <- setwd(tmp_dir)
    utils::tar(tar_fn, files="chr22_1.csv", compression="gzip")
    setwd(old_wd)

    out_fn2 <- tempfile(fileext=".gds")
    seqFAVORTar2GDS(tar_fn, out_fn2, tar_cmd="tar")
    seqAnnotList(out_fn2)

    # delete the temporary files
    unlink(c(tmp_dir, tar_fn, out_fn2), recursive=TRUE, force=TRUE)
}

# delete the temporary file
unlink(out_fn, force=TRUE)
```

---

seqGnomAD2GDS *Reformat gnomAD VCF*

Description

Convert gnomAD VCF files to a SeqArray GDS file.

Usage

```
seqGnomAD2GDS(vcf_fn, out_fn, compress=c("LZMA", "ZIP", "none"),
    verbose=TRUE)
```

Arguments

vcf_fn	the path(es) of gnomAD VCF file
out_fn	the file name of output SeqArray GDS file
compress	compression method (LZMA by default), "LZMA" has a higher compression ratio than "ZIP"
verbose	if TRUE, show information

Value

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

Author(s)

Xiuwen Zheng

References

Chen S, Francioli LC, Goodrich, JK, Collins, RL, etc., Nature. 625, 92-100 (2024). doi.org/10.1038/s41586-023-06045-0. PMID: 38057664

See Also

seqVCF2GDS

Examples

```
# gnomAD v4.1.0 dataset of chromosome 22
vcf_fn <- system.file("extdata", "gnomad.genomes.v4.chr22_sub.vcf.gz",
    package="GDSAnnotator")

out_fn <- tempfile(fileext=".gds")
seqGnomAD2GDS(vcf_fn, out_fn)

# list the annotations in the output file
seqAnnotList(out_fn)

# delete the temporary file
unlink(out_fn, force=TRUE)
```

---

seqSnpEff2GDS *Reformat SnpEff VCF*

Description

Convert SnpEff VCF output to a SeqArray GDS file.

Usage

```
seqSnpEff2GDS(vcf_fn, out_fn, compress=c("LZMA", "ZIP", "none"),
    root=c("ANN", "LOF", "NMD"), keep=FALSE, bsize=50000L, verbose=TRUE)
```

Arguments

vcf_fn	the path(es) of VCF file from SnpEff output
out_fn	the file name of output SeqArray GDS file
compress	compression method (LZMA by default), "LZMA" has a higher compression ratio than "ZIP"
root	the name(s) of INFO field in the input VCF file to be parsed
keep	if TRUE, keep the original root node(s) in the output GDS file; if FALSE or NA, remove them after splitting into sub-fields
bsize	the block size (number of variants per chunk) for processing annotations; a smaller value reduces peak memory usage at the cost of more iterations
verbose	if TRUE, show information

Value

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

Author(s)

Xiuwen Zheng

References

Cingolani, P., Platts, A., Wang le L., Coon, M., Nguyen, T., Wang, L., Land, S.J., Lu, X., & Ruden, D.M. (2012). A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of Drosophila melanogaster strain w1118; iso-2; iso-3. Fly, 6(2), 80-92. PMID: 22728672.

See Also

seqVCF2GDS

Examples

```
# SnpEff annotation
vcf_fn <- system.file("extdata", "example_wgs_sites_chr22_snpeff.vcf.gz",
    package="GDSAnnotator")

out_fn <- tempfile(fileext=".gds")
seqSnpEff2GDS(vcf_fn, out_fn)

# list the annotations in the output file
seqAnnotList(out_fn)

# delete the temporary file
unlink(out_fn, force=TRUE)
```

seqValueCounts *Counts of unique values*

Description

Calculate the counts of unique values.

Usage

```
seqValueCounts(gdsfile, varnm, use_info=TRUE, FUN=NULL, per_variant=FALSE,
    parallel=FALSE, bsize=100000L, verbose=TRUE, ...)
```

Arguments

gdsfile	a SeqVarGDSClass object, a GDS file name or a character vector for multiple file names
varnm	a character vector containing GDS node name(s)
use_info	TRUE or FALSE; if TRUE, to use the node(s) in the folder "annotation/info"; otherwise, full node name(s) should be specified in varnm
FUN	NULL to use the default table, or a user-defined function to be applied
per_variant	FALSE by default; if TRUE, counting at the variant level, i.e., multiple annotations for the same variant are collapsed into a single count
parallel	FALSE (serial processing), TRUE (multicore processing), numeric value or other value; parallel is passed to the argument cl in seqParallel, see seqParallel for more details
bsize	block size
verbose	if TRUE, show information
...	additional arguments passed to FUN if it is a function

Value

Returns a contingency table, an object of class table, an array of integer values.

Author(s)

Xiuwen Zheng

See Also

seqAnnotList

Examples

```
# gnomAD v4.1.0 dataset of chromosome 22
gds_fn <- system.file("extdata", "gnomad.genomes.v4.chr22_sub.gds",
    package="GDSAnnotator")

# open the file
f <- seqOpen(gds_fn)
```

```
# list the annotations in the gds file
(a <- seqAnnotList(f))

seqValueCounts(f, "variant_type")

seqValueCounts(f, "was_mixed")

seqValueCounts(f, c("variant_type", "allele_type"))

seqValueCounts(f, c("variant_type", "allele_type", "was_mixed"))

seqClose(f)
```

`seqVEP2GDS` *Reformat Ensembl-VEP VCF*

Description

Convert Ensembl-VEP VCF output to a SeqArray GDS file.

Usage

```
seqVEP2GDS(vcf_fn, out_fn, compress=c("LZMA", "ZIP", "none"),
    root="CSQ", keep=FALSE, bsize=50000L, verbose=TRUE)
```

Arguments

`vcf_fn`	the path(es) of VCF file from Ensembl-VEP output
`out_fn`	the file name of output SeqArray GDS file
`compress`	compression method (LZMA by default), `"LZMA"` has a higher compression ratio than `"ZIP"`
`root`	the name of INFO field in the input VCF file, the default is `"CSQ"`
`keep`	if `TRUE`, keep the original `root` node in the output GDS file; if `FALSE` or `NA`, remove it after splitting into sub-fields
`bsize`	the block size (number of variants per chunk) for processing annotations; a smaller value reduces peak memory usage at the cost of more iterations
`verbose`	if `TRUE`, show information

Value

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

Author(s)

Xiuwen Zheng

References

McLaren W, Gil L, Hunt SE, Riat HS, Ritchie GR, Thormann A, Flicek P, Cunningham F. The Ensembl Variant Effect Predictor. Genome Biology Jun 6; 17(1):122. (2016) DOI:10.1186/s13059-016-0974-4.

See Also

seqVCF2GDS

Examples

```
# Ensembl-VEP annotation
vcf_fn <- system.file("extdata", "example_wgs_sites_chr22_vep.vcf.gz",
    package="GDSAnnotator")

out_fn <- tempfile(fileext=".gds")
seqVEP2GDS(vcf_fn, out_fn)

# list the annotations in the output file
seqAnnotList(out_fn)

# delete the temporary file
unlink(out_fn, force=TRUE)
```

Index

∗ **ANNOVAR**
seqANNOVAR2GDS, 14
∗ **FAVOR**
seqFAVOR2GDS, 16
∗ **SnpEff**
seqSnpEff2GDS, 18
∗ **VEP**
seqVEP2GDS, 21
∗ **WES**
seqAnnotate-methods, 5
∗ **WGS**
seqAnnotate-methods, 5
seqANNOVAR2GDS, 14
seqFAVOR2GDS, 16
seqGnomAD2GDS, 17
seqSnpEff2GDS, 18
seqVEP2GDS, 21
∗ **annotation**
GDSAnnotator-package, 2
seqAnnotate-methods, 5
seqAnnotGeneTable, 9
seqAnnotList, 10
seqAnnotReport, 11
seqAnnotStat, 12
seqANNOVAR2GDS, 14
seqFAVOR2GDS, 16
seqGnomAD2GDS, 17
seqSnpEff2GDS, 18
seqValueCounts, 20
seqVEP2GDS, 21
∗ **datasets**
GDSAnnotator-extdata, 4
∗ **gds**
GDSAnnotator-package, 2
seqAnnotate-methods, 5
seqAnnotGeneTable, 9
seqAnnotList, 10
seqAnnotReport, 11
seqAnnotStat, 12
seqANNOVAR2GDS, 14
seqFAVOR2GDS, 16
seqGnomAD2GDS, 17
seqSnpEff2GDS, 18
seqValueCounts, 20
seqVEP2GDS, 21
∗ **gnomAD**
seqGnomAD2GDS, 17
∗ **package**
GDSAnnotator-package, 2

expand, *7*

GDSAnnotator *(*GDSAnnotator-package*)*, 2
GDSAnnotator-extdata, 4
GDSAnnotator-package, 2

print.SeqAnnotStat *(*seqAnnotStat*)*, 12

readVcf, *6, 7*

seqAnnotate, *3*, *10*
seqAnnotate *(*seqAnnotate-methods*)*, 5
seqAnnotate,ANY-method
*(*seqAnnotate-methods*)*, 5
seqAnnotate,character-method
*(*seqAnnotate-methods*)*, 5
seqAnnotate,data.frame-method
*(*seqAnnotate-methods*)*, 5
seqAnnotate,DataFrame-method
*(*seqAnnotate-methods*)*, 5
seqAnnotate,GRanges-method
*(*seqAnnotate-methods*)*, 5
seqAnnotate,GRangesList-method
*(*seqAnnotate-methods*)*, 5
seqAnnotate,IRanges-method
*(*seqAnnotate-methods*)*, 5
seqAnnotate,SeqVarGDSClass-method
*(*seqAnnotate-methods*)*, 5
seqAnnotate-methods, 5
seqAnnotateGDS, *3*
seqAnnotateGDS *(*seqAnnotate-methods*)*, 5
seqAnnotateVCF, *3*
seqAnnotateVCF *(*seqAnnotate-methods*)*, 5
seqAnnotGeneTable, *3*, 9, *12*, *14*
seqAnnotList, *3*, *8*, 10, *14*, *20*
seqAnnotReport, *3*, *9*, 11, *14*
seqAnnotStat, *3, 4*, *9*, *11*, *12*, 12
seqANNOVAR2GDS, *3*, *5*, 14

seqBlockApply, *13*
seqFAVOR2GDS, *3–5*, *7, 8*, 16
seqFAVORTar2GDS, *4*
seqFAVORTar2GDS *(*seqFAVOR2GDS*)*, 16
seqGetData, *3*
seqGnomAD2GDS, *3–5*, 17
seqOpen, *3*
seqParallel, *13*, *20*
seqSetFilter, *7*
seqSnpEff2GDS, *3–5*, *13*, *15*, 18
seqValueCounts, *3*, *8*, *14*, 20
seqVCF2GDS, *3*, *15*, *17–19*, *22*
seqVEP2GDS, *3–5*, *8*, *13*, *15*, 21