

# Package ‘VariantExperiment’

November 2, 2024

**Title** A RangedSummarizedExperiment Container for VCF/GDS Data with GDS Backend

**Version** 1.20.0

**Description** VariantExperiment is a Bioconductor package for saving data in VCF/GDS format into RangedSummarizedExperiment object. The high-throughput genetic/genomic data are saved in GDSArray objects. The annotation data for features/samples are saved in DelayedDataFrame format with mono-dimensional GDSArray in each column. The on-disk representation of both assay data and annotation data achieves on-disk reading and processing and saves memory space significantly. The interface of RangedSummarizedExperiment data format enables easy and common manipulations for high-throughput genetic/genomic data with common SummarizedExperiment metaphor in R and Bioconductor.

**biocViews** Infrastructure, DataRepresentation, Sequencing, Annotation, GenomeAnnotation, GenotypingArray

**Depends** R (>= 3.6.0), S4Vectors (>= 0.21.24), SummarizedExperiment (>= 1.13.0), GenomicRanges,

**License** GPL-3

**Encoding** UTF-8

**URL** <https://github.com/Bioconductor/VariantExperiment>

**BugReports** <https://github.com/Bioconductor/VariantExperiment/issues>

**Imports** GDSArray (>= 1.11.1), DelayedDataFrame (>= 1.6.0), tools, utils, stats, methods, gdsfmt, SNPRelate, SeqArray, DelayedArray, Biostrings, IRanges

**RoxygenNote** 7.3.1

**Suggests** testthat, knitr, rmarkdown, markdown, BiocStyle

**VignetteBuilder** knitr

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

**git\_branch** RELEASE\_3\_20

**git\_last\_commit** dc5092d

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

**Repository** Bioconductor 3.20

**Date/Publication** 2024-11-01

**Author** Qian Liu [aut, cre],  
 Hervé Pagès [aut],  
 Martin Morgan [aut]

**Maintainer** Qian Liu <Qian.Liu@roswellpark.org>

## Contents

VariantExperiment-package . . . . .	2
loadVariantExperiment . . . . .	3
makeVariantExperimentFromGDS . . . . .	3
makeVariantExperimentFromVCF . . . . .	5
saveVariantExperiment . . . . .	7
showAvailable . . . . .	8
VariantExperiment-class . . . . .	9

**Index** **11**

---

VariantExperiment-package

*VariantExperiment: A package to represent VCF / GDS files using standard SummarizedExperiment metaphor with on-disk representation.*

---

## Description

The package VariantExperiment takes GDS file or VCF file as input, and save them in VariantExperiment object. Assay data are saved in GDSArray objects and annotation data are saved in DelayedDataFrame format, both of which remain on-disk until needed. Common manipulations like subsetting, mathematical transformation and statistical analysis are done easily and quickly in R.

## Author(s)

**Maintainer:** Qian Liu <Qian.Liu@roswellpark.org>

Authors:

- Hervé Pagès
- Martin Morgan

## See Also

Useful links:

- <https://github.com/Bioconductor/VariantExperiment>
- Report bugs at <https://github.com/Bioconductor/VariantExperiment/issues>

---

loadVariantExperiment *loadVariantExperiment* to load the GDS back-end SummarizedExperiment object into R console.

---

**Description**

loadVariantExperiment to load the GDS back-end SummarizedExperiment object into R console.

**Usage**

```
loadVariantExperiment(dir = tempdir())
```

**Arguments**

dir                   The directory to save the gds format of the array data, and the newly generated SummarizedExperiment object with array data in GDSArray format.

**Value**

An VariantExperiment object.

**Examples**

```
gds <- SeqArray::seqExampleFileName("gds")
## ve <- makeVariantExperimentFromGDS(gds)
## ve1 <- subsetByOverlaps(ve, GRanges("22:1-48958933"))
aa <- tempfile()
## saveVariantExperiment(ve1, dir=aa, replace=TRUE)
## loadVariantExperiment(dir = aa)
```

---

makeVariantExperimentFromGDS  
*makeVariantExperimentFromGDS*

---

**Description**

Conversion of gds files into SummarizedExperiment object.

**Usage**

```
makeVariantExperimentFromGDS(  
  file,  
  ftnode,  
  smpnode,  
  assayNames = NULL,  
  rowDataColumns = NULL,  
  colDataColumns = NULL,  
  rowDataOnDisk = TRUE,  
  colDataOnDisk = TRUE,  
  infoColumns = NULL
```

```

)

makeVariantExperimentFromSEQGDS(
  file,
  ftnode = "variant.id",
  smpname = "sample.id",
  assayNames = NULL,
  rowDataColumns = NULL,
  colDataColumns = NULL,
  infoColumns = NULL,
  rowDataOnDisk = TRUE,
  colDataOnDisk = TRUE
)

makeVariantExperimentFromSNPGDS(
  file,
  ftnode = "snp.id",
  smpname = "sample.id",
  assayNames = NULL,
  rowDataColumns = NULL,
  colDataColumns = NULL,
  rowDataOnDisk = TRUE,
  colDataOnDisk = TRUE
)

```

### Arguments

<code>file</code>	the GDS file name to be converted.
<code>ftnode</code>	the node name for feature id (e.g., "variant.id", "snp.id", etc.).
<code>smpname</code>	the node name for sample id (e.g., "sample.id").
<code>assayNames</code>	the gds node name that will be read into the assays slot and be represented as DelayedArray object.
<code>rowDataColumns</code>	which columns of <code>rowData</code> to import. The default is NULL to read in all variant annotation info.
<code>colDataColumns</code>	which columns of <code>colData</code> to import. The default is NULL to read in all sample related annotation info.
<code>rowDataOnDisk</code>	whether to save the <code>rowData</code> as DelayedArray object. The default is TRUE.
<code>colDataOnDisk</code>	whether to save the <code>colData</code> as DelayedArray object. The default is TRUE.
<code>infoColumns</code>	which columns of <code>infoColumns</code> to import for "SEQ_ARRAY" ("SeqVarGDSClass" gds class). The default is NULL to read in all available info columns.

### Value

An VariantExperiment object.

### Examples

```

## gds file from DNA-seq data

seqfile <- SeqArray::seqExampleFileName(type="gds")
ve <- makeVariantExperimentFromGDS(seqfile)

```

```

## all assay data
names(assays(ve))
showAvailable(seqfile)

## only read specific columns for feature / sample annotation.

assayNames <- showAvailable(seqfile)$assayNames
rowDatacols <- showAvailable(seqfile)$rowDataColumns
coldatacols <- showAvailable(seqfile)$colDataColumns
infocols <- showAvailable(seqfile)$infoColumns
ve1 <- makeVariantExperimentFromGDS(
  seqfile,
  assayNames = assayNames[2],
  rowDataColumns = rowDatacols[1:3],
  colDataColumns = coldatacols[1],
  infoColumns = infocols[c(1,3,5,7)],
  rowDataOnDisk = FALSE,
  colDataOnDisk = FALSE)
assay(ve1)

## the rowData(ve1) and colData(ve1) are now in DataFrame format

rowData(ve1)
colData(ve1)

## gds file from genotyping data

snpfile <- SNPRelate::snpgdsExampleFileName()
ve <- makeVariantExperimentFromGDS(snpfile)
rowData(ve)
colData(ve)
metadata(ve)

## Only read specific columns for feature annotation.

showAvailable(snpfile)
ve1 <- makeVariantExperimentFromGDS(snpfile, rowDataColumns=c("snp.allele"))
rowRanges(ve1)

## use specific conversion functions for certain gds types

veseq <- makeVariantExperimentFromSEQGDS(seqfile)
vesnp <- makeVariantExperimentFromSNPGDS(snpfile)

```

---

```
makeVariantExperimentFromVCF
```

*The function to convert VCF files directly into VariantExperiment object.*

---

## Description

makeVariantExperimentFromVCF is the function to convert a vcf file into VariantExperiment object. The genotype data will be written as GDSArray format, which is saved in the assays slot. The annotation info for variants or samples will be written as DelayedDataFrame object, and saved in the rowData or colData slot.

**Usage**

```

makeVariantExperimentFromVCF(
  vcf.fn,
  out.dir = tempfile(),
  replace = FALSE,
  header = NULL,
  info.import = NULL,
  fmt.import = NULL,
  sample.info = NULL,
  ignore.chr.prefix = "chr",
  reference = NULL,
  start = 1L,
  count = -1L,
  parallel = FALSE,
  verbose = FALSE
)

```

**Arguments**

<code>vcf.fn</code>	the file name(s) of (compressed) VCF format; or a 'connection' object.
<code>out.dir</code>	The directory to save the gds format of the vcf data, and the newly generated VariantExperiment object with array data in GDSArray format and annotation data in DelayedDataFrame format. The default is a temporary folder.
<code>replace</code>	Whether to replace the directory if it already exists. The default is FALSE.
<code>header</code>	if NULL, 'header' is set to be 'seqVCF_Header(vcf.fn)', which is a list (with a class name "SeqVCFHeaderClass", S3 object).
<code>info.import</code>	characters, the variable name(s) in the INFO field for import; default is 'NULL' for all variables.
<code>fmt.import</code>	characters, the variable name(s) in the FORMAT field for import; default is 'NULL' for all variables.
<code>sample.info</code>	characters (with) file path for the sample info data. The data must have colnames (for phenotypes), rownames (sample ID's). No blank line allowed. The default is 'NULL' for no sample info.
<code>ignore.chr.prefix</code>	a vector of character, indicating the prefix of chromosome which should be ignored, like "chr"; it is not case-sensitive.
<code>reference</code>	genome reference, like "hg19", "GRCh37"; if the genome reference is not available in VCF files, users could specify the reference here.
<code>start</code>	the starting variant if importing part of VCF files.
<code>count</code>	the maximum count of variant if importing part of VCF files, -1 indicates importing to the end.
<code>parallel</code>	'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 '?SeqArray::seqParallel' for more details. The default is "FALSE".
<code>verbose</code>	whether to print the process messages. The default is FALSE.

**Value**

An VariantExperiment object.

**Examples**

```

## the vcf file
vcf <- SeqArray::seqExampleFileName("vcf")
## conversion
ve <- makeVariantExperimentFromVCF(vcf)
ve
## the filepath to the gds file.
gdsfile(ve)

## only read in specific info columns
ve <- makeVariantExperimentFromVCF(vcf, out.dir = tempfile(),
                                   info.import=c("OR", "GP"))
ve
## convert without the INFO and FORMAT fields
ve <- makeVariantExperimentFromVCF(vcf, out.dir = tempfile(),
                                   info.import=character(0),
                                   fmt.import=character(0))
ve
## now the assay data does not include the
##"annotation/format/DP/data", and the rowData(ve) does not include
##any info columns.

```

---

`saveVariantExperiment` *saveVariantExperiment* Save all the assays in GDS format, including in-memory assays. Delayed assays with delayed operations on them are realized while they are written to disk.

---

**Description**

`saveVariantExperiment` Save all the assays in GDS format, including in-memory assays. Delayed assays with delayed operations on them are realized while they are written to disk.

**Usage**

```

saveVariantExperiment(
  ve,
  dir = tempdir(),
  replace = FALSE,
  fileFormat = NULL,
  compress = "LZMA_RA",
  chunk_size = 1000,
  rowDataOnDisk = TRUE,
  colDataOnDisk = TRUE,
  verbose = FALSE
)

```

**Arguments**

<code>ve</code>	A SummarizedExperiment object, with the array data being ordinary array structure.
<code>dir</code>	The directory to save the gds format of the array data, and the newly generated SummarizedExperiment object with array data in GDSArray format. The default is temporary directory within the R session.

replace	Whether to replace the directory if it already exists. The default is FALSE.
fileFormat	File format for the output gds file. See details.
compress	the compression method for writing the gds file. The default is "LZMA_RA".
chunk_size	The chunk size (number of columns) when reading GDSArray-based assays from input ve into memory and then write into a new gds file. Default is 1000. Can be modified to smaller value if chunk data is too big (e.g., when number of rows are large).
rowDataOnDisk	whether to save the rowData as DelayedArray object. The default is TRUE.
colDataOnDisk	whether to save the colData as DelayedArray object. The default is TRUE.
verbose	whether to print the process messages. The default is FALSE.

### Details

If the input SummarizedExperiment object has GDSArray-based assay data, there is no need to specify the argument fileFormat. Otherwise, it takes values of SEQ\_ARRAY for sequencing data or SNP\_ARRAY SNP array data.

### Value

An VariantExperiment object with the new gdsfile() ve.gds as specified in dir argument.

### Examples

```
gds <- SeqArray::seqExampleFileName("gds")
ve <- makeVariantExperimentFromGDS(gds)
gdsfile(ve)
ve1 <- subsetByOverlaps(ve, GRanges("22:1-48958933"))
ve1
gdsfile(ve1)
aa <- tempfile()
obj <- saveVariantExperiment(ve1, dir=aa, replace=TRUE)
obj
gdsfile(obj)
```

---

showAvailable

*ShowAvailable*

---

### Description

The function to show the available entries for the arguments within makeVariantExperimentFromGDS

### Usage

```
showAvailable(
  file,
  args = c("assayNames", "rowDataColumns", "colDataColumns", "infoColumns"),
  ftnode,
  smpnode
)
```



**Arguments**

file	the path to the gds.class file.
args	the arguments in makeVariantExperimentFromGDS.
ftnode	the node name for feature id (e.g., "variant.id", "snp.id", etc.). Must be provided if the file format is not SNP_ARRAY or SEQ_ARRAY.
smnnode	the node name for sample id (e.g., "sample.id"). Must be provided if the file format is not SNP_ARRAY or SEQ_ARRAY.

**Examples**

```
## snp gds file
gds <- SNPRelate::snpgdsExampleFileName()
showAvailable(gds)

## sequencing gds file
gds <- SeqArray::seqExampleFileName("gds")
showAvailable(gds)
```

---

VariantExperiment-class

*VariantExperiment-class*


---

**Description**

VariantExperiment could represent big genomic data in RangedSummarizedExperiment object, with on-disk GDS back-end data. The assays are represented by DelayedArray objects; rowData and colData could be represented by DelayedDataFrame or DataFrame objects.

**Usage**

```
VariantExperiment(
  assays,
  rowRanges = GRangesList(),
  colData = DelayedDataFrame(),
  metadata = list()
)

## S4 method for signature 'VariantExperiment'
gdsfile(object)
```

**Arguments**

assays	A 'list' or 'SimpleList' of matrix-like elements, or a matrix-like object. All elements of the list must have the same dimensions, and dimension names (if present) must be consistent across elements and with the row names of 'rowRanges' and 'colData'.
rowRanges	A GRanges or GRangesList object describing the ranges of interest. Names, if present, become the row names of the SummarizedExperiment object. The length of the GRanges or GRangesList must equal the number of rows of the matrices in 'assays'.

colData	An optional DataFrame describing the samples. Row names, if present, become the column names of the VariantExperiment.
metadata	An optional 'list' of arbitrary content describing the overall experiment.
object	a VariantExperiment object.

**Details**

VariantExperiment class and slot getters and setters.  
check "[?RangedSummarizedExperiment](#)" for more details.

**Value**

a VariantExperiment object.

# Index

`gdsfile`, `VariantExperiment`-method  
(`VariantExperiment`-class), 9

`loadVariantExperiment`, 3

`makeVariantExperimentFromGDS`, 3  
`makeVariantExperimentFromSEQGDS`  
(`makeVariantExperimentFromGDS`),  
3

`makeVariantExperimentFromSNPGDS`  
(`makeVariantExperimentFromGDS`),  
3

`makeVariantExperimentFromVCF`, 5

`saveVariantExperiment`, 7  
`showAvailable`, 8

`VariantExperiment`  
(`VariantExperiment`-class), 9  
`VariantExperiment`-class, 9  
`VariantExperiment`-package, 2