Package 'VariantExperiment'

March 30, 2021

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

Version 1.4.0

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, GDSArray (>= 1.3.0), DelayedDataFrame (>= 1.0.0)

License GPL-3

Encoding UTF-8

URL https://github.com/Bioconductor/VariantExperiment

BugReports https://github.com/Bioconductor/VariantExperiment/issues

Imports tools, utils, stats, methods, gdsfmt, SNPRelate, SeqArray, SeqVarTools, DelayedArray, Biostrings, IRanges

RoxygenNote 7.1.0

Suggests testthat, knitr

VignetteBuilder knitr

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

git_branch RELEASE_3_12

git_last_commit a279f8d

git_last_commit_date 2020-10-27

Date/Publication 2021-03-29

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

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

R topics documented:

`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_.

`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)
```

---

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

vcf.fn	the file name(s) of (compressed) VCF format; or a 'connection' object.
out.dir	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.
replace	Whether to replace the directory if it already exists. The default is FALSE.
header	if NULL, 'header' is set to be 'seqVCF_Header(vcf.fn)', which is a list (with a class name "SeqVCFHeaderClass", S3 object).

`info.import` characters, the variable name(s) in the INFO field for import; default is 'NULL' for all variables.

`fmt.import` characters, the variable name(s) in the FORMAT field for import; default is 'NULL' for all variables.

`sample.info` 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.

`ignore.chr.prefix` a vector of character, indicating the prefix of chromosome which should be ignored, like "chr"; it is not case-sensitive.

`reference` genome reference, like "hg19", "GRCh37"; if the genome reference is not available in VCF files, users could specify the reference here.

`start` the starting variant if importing part of VCF files.

`count` the maximum count of variant if importing part of VCF files, -1 indicates importing to the end.

`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 '?SeqArray::seqParallel' for more details. The default is "FALSE".

`verbose` 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 = 10000,
  rowDataOnDisk = TRUE,
  colDataOnDisk = TRUE,
  verbose = FALSE
)
```

Arguments

ve	A SummarizedExperiment object, with the array data being ordinary array structure.
dir	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 rows) when reading GDSArray-based assays from input ve into memory and then write into a new gds file.
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 fileFomat. 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`

Conversion of gds file into SummarizedExperiment.

Usage

```
showAvailable(
  file,
  args = c("name", "rowDataColumns", "colDataColumns", "infoColumns")
)

makeVariantExperimentFromGDS(
  file,
  name = NULL,
  rowDataColumns = NULL,
  colDataColumns = NULL,
  infoColumns = NULL,
  rowDataOnDisk = TRUE,
  colDataOnDisk = TRUE
)
```

Arguments

`file`	the path to the gds.class file.
`args`	the arguments in `makeVariantExperimentFromGDS`.
`name`	the components of the gds file that will be represented as `GDSArray` file.
`rowDataColumns`	which columns of `rowData` to import. The default is NULL to read in all variant annotation info.
`colDataColumns`	which columns of `colData` to import. The default is NULL to read in all sample related annotation info.
`infoColumns`	which columns of `infoColumns` to import. The default is NULL to read in all info columns.
`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.

Value

An VariantExperiment object.

Examples

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

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

file <- SNPRelate::snpgdsExampleFileName()
## se <- makeVariantExperimentFromGDS(file)
## rowData(se)
## colData(se)
## metadata(se)
## Only read specific columns for feature annotation.
showAvailable(file)
## se1 <- makeVariantExperimentFromGDS(file, rowDataColumns=c("ALLELE"))
## SummarizedExperiment::rowRanges(se1)
file <- SeqArray::seqExampleFileName(type="gds")
## se <- makeVariantExperimentFromGDS(file)
## all assay data
## names(assays(se))
## showAvailable(file)

## only read specific columns for feature / sample annotation.
names <- showAvailable(file, "name")$name
rowdatacols <- showAvailable(file, "rowDataColumns")$rowDataColumns
coldatacols <- showAvailable(file, "colDataColumns")$colDataColumns
infocols <- showAvailable(file, "infoColumns")$infoColumns
## se1 <- makeVariantExperimentFromGDS(
## file,
## name = names[2],
## rowDataColumns = rowdatacols[1:3],
## colDataColumns = coldatacols[1],
## infoColumns = infocols[c(1,3,5,7)],
## rowDataOnDisk = FALSE,
## colDataOnDisk = FALSE)
## assay(se1)

## the rowData(se1) and colData(se1) are now in DataFrame format
## rowData(se1)
## colData(se1)
```

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` 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.

`VariantExperiment-methods` *Statistical functions for* `VariantExperiment` *objects.*

Description

Statistical functions for `VariantExperiment` objects.

Usage

```
## S4 method for signature 'VariantExperiment'
seqAlleleFreq(
  gdsfile,
  ref.allele = 0L,
  minor = FALSE,
  .progress = FALSE,
  parallel = seqGetParallel(),
  verbose = FALSE
)

## S4 method for signature 'VariantExperiment'
seqAlleleCount(
  gdsfile,
  ref.allele = 0L,
  minor = FALSE,
  .progress = FALSE,
  parallel = seqGetParallel(),
  verbose = FALSE
)

## S4 method for signature 'VariantExperiment'
seqMissing(
  gdsfile,
  per.variant = TRUE,
  .progress = FALSE,
  parallel = seqGetParallel(),
  verbose = FALSE
)

## S4 method for signature 'VariantExperiment'
seqNumAllele(gdsfile)

## S4 method for signature 'VariantExperiment'
hwe(gdsobj, permute = FALSE)

## S4 method for signature 'VariantExperiment'
inbreedCoeff(gdsobj, margin = c("by.variant", "by.sample"), use.names = FALSE)

## S4 method for signature 'VariantExperiment'
pca(gdsobj, eigen.cnt = 32)

## S4 method for signature 'VariantExperiment'
titv(gdsobj, by.sample = FALSE, use.names = FALSE)

## S4 method for signature 'VariantExperiment'
refDosage(gdsobj, use.names = TRUE)

## S4 method for signature 'VariantExperiment'
altDosage(gdsobj, use.names = TRUE, sparse = FALSE)

## S4 method for signature 'VariantExperiment'
```

```
countSingletons(gdsobj, use.names = FALSE)

## S4 method for signature 'VariantExperiment'
heterozygosity(
  gdsobj,
  margin = c("by.variant", "by.sample"),
  use.names = FALSE
)

## S4 method for signature 'VariantExperiment'
homozygosity(
  gdsobj,
  allele = c("any", "ref", "alt"),
  margin = c("by.variant", "by.sample"),
  use.names = FALSE
)

## S4 method for signature 'VariantExperiment'
meanBySample(gdsobj, var.name, use.names = FALSE)

## S4 method for signature 'VariantExperiment'
isSNV(gdsobj, biallelic = TRUE)

## S4 method for signature 'VariantExperiment'
isVariant(gdsobj, use.names = FALSE)
```

Arguments

gdsfile	an VariantExperiment object that with synchronized gds file.
ref.allele	a single numeric value, a numeric vector or a character vector; see ?SeqArray::seqAlleleFreq for more details.
minor	if 'TRUE', return minor allele frequency/count
.progress	Logical, show process information if TRUE.
parallel	A logical value to indicate serial processing (FALSE) or multicore processing (TRUE). Takes numeric value or other value; see ?SeqArray::seqParallel for more details.
verbose	if 'TRUE', show progress information
per.variant	A logical value to indicate whether to calculate missing rate for variant (TRUE), or samples (FALSE).
gdsobj	same as above gdsfile argument.
permute	A logical value indicating whether to permute the genotypes. See ?SeqVarTools::hwe for more details.
margin	"by.variant" OR "by.sample" to indicate whether the calculation should be done over all samples for each variant, or over all variants for each sample. See ?SeqVarTools::inbreedCoeff for more details.
use.names	A logical value indicating whether to assign variant or sample IDs as names of the output vector.
eigen.cnt	An integer value indicating how many eigenvalues and eignvectors to return. The default is 32.

by.sample	A logical value indicating whether TiTv should be calculated by sample or overall for the entire VariantExperiment object. See ?SeqVarTools::titv for more details.
sparse	A Logical value indicating whether or not to return the alterate allele dosage as a sparse matrix. In most cases, it will dramatically reduce the size of the returned object. See ?SeqVarTools::altDosage for more details.
allele	Choose from "any", "ref," or "alt," to indicate which alleles to consider when calculating homozygosity. See ?SeqVarTools::homozygosity for more details.
var.name	Character string with name of the variable. Choose from names(assays(VE_Object)). See ?SeqVarTools::meanBySample for more details.
biallelic	A logical indicating whether to only consider biallelic SNVs. See ?SeqVarTools::isSNV for more details.

Value

Statistical results in vector or data.frame format.

Examples

```
gds <- SeqArray::seqExampleFileName("gds")
## ve <- makeVariantExperimentFromGDS(gds)
## ve

## sample missing rate
## mr.samp <- seqMissing(ve, per.variant = FALSE)
## ead(mr.samp)

## hwe
## hwe <- hwe(ve)
## head(hwe)

## titv ratio by sample / overall
## titv <- titv(ve, by.sample=TRUE)
## head(titv)
## titv(ve, by.sample=FALSE)

## countSingletons
## countSingletons(ve)
```

Index