Package ‘alabaster.files’

April 15, 2024

Title Wrappers to Save Common File Formats

Version 1.0.0

Date 2023-05-30

License MIT + file LICENSE

Description Save common bioinformatics file formats within the alabaster framework.
This includes BAM, BED, VCF, bigWig, bigBed, FASTQ, FASTA and so on.
We save and load additional metadata for each file,
and we support linkage between each file and its corresponding index.

Depends alabaster.base,

Imports methods, S4Vectors, BiocGenerics, rtracklayer, Rsamtools, Biostrings

Suggests BiocStyle, rmarkdown, knitr, testthat, VariantAnnotation

VignetteBuilder knitr

RoxygenNote 7.2.3

biocViews DataRepresentation, DataImport

git_url https://git.bioconductor.org/packages/alabaster.files

git_branch RELEASE_3_18

git_last_commit 7ed9441

git_last_commit_date 2023-10-24

Repository Bioconductor 3.18

Date/Publication 2024-04-15

Author Aaron Lun [aut, cre]

Maintainer Aaron Lun <infinite.monkeys.with.keyboards@gmail.com>

R topics documented:

BamIndexWrapper *Wrapper for a BAM index file*

Description

Wrap a BAM index file for saving and loading in the **alabaster** framework.

Usage

```
BamIndexWrapper(path)
```

Arguments

path	String containing the path to a BAM index file.

Details

The BamIndexWrapper class is a subclass of a Wrapper, so all of the methods of the latter can also be used here, e.g., path.

Value

A BamIndexWrapper instance that can be used in stageObject.

Author(s)

Aaron Lun

Examples

```
# Mocking up a BAM index file.
test_bai <- system.file("extdata", "ex1.bam.bai", package="Rsamtools")

# Creating a BamIndexWrapper.
wrapped <- BamIndexWrapper(test_bai)
wrapped
```

```
# Staging the BamIndexWrapper.
dir <- tempfile()
library(alabaster.base)
info <- stageObject(wrapped, dir, "tab")
invisible(.writeMetadata(info, dir))
list.files(dir, recursive=TRUE)

# Loading it back again:
meta <- acquireMetadata(dir, "tab/file.bam.bai")
loadObject(meta, dir)
```

`BamWrapper` *Wrapper for a BAM file*

Description

Wrap a BAM file for saving and loading in the **alabaster** framework.

Usage

```
BamWrapper(path, index = NULL)
```

Arguments

`path`	String containing the path to a BAM file.
`index`	String specifying the path to an index file in tabix format, or `NULL` if no index is available.

Details

The BamWrapper class is a subclass of a IndexedWrapper, so all of the methods of the latter can also be used here, e.g., `path`, `index`.

The `stageObject` method for BamWrapper classes will check the BAM file by scanning the header with `scanBamHeader`. If an index is present, it will additionally run `idxstatsBam`.

Value

A BamWrapper instance that can be used in `stageObject`.

Author(s)

Aaron Lun

Examples

```
# Using a BAM file from Rsamtools.
fl <- system.file("extdata", "ex1.bam", package="Rsamtools", mustWork=TRUE)

# Creating a BamWrapper.
wrapped <- BamWrapper(fl)
wrapped

# Staging the BamWrapper.
dir <- tempfile()
library(alabaster.base)
info <- stageObject(wrapped, dir, "my_bam")
invisible(.writeMetadata(info, dir))
list.files(dir, recursive=TRUE)

# Loading it back again:
meta <- acquireMetadata(dir, "my_bam/file.bam")
loadObject(meta, dir)
```

`BedWrapper` *Wrapper for a BED file*

Description

Wrap a BED file for saving and loading in the **alabaster** framework.

Usage

```
BedWrapper(path, compression = NULL, index = NULL)
```

Arguments

`path` String containing the path to a BED file.

`compression` String specifying the compression. This should be one of `"none"`, `"gzip"`, `"bzip2"` or `"bgzip"`. If `NULL`, this is inferred from the file's headers and suffix.

`index` String specifying the path to an index file in tabix format, or `NULL` if no index is available. If an index is supplied, the file should be bgzip-compressed.

Details

The BedWrapper class is a subclass of a CompressedIndexedWrapper, so all of the methods of the latter can also be used here, e.g., `path`, `index`, `compression`.

The `stageObject` method for BedWrapper classes will check the BED file by reading the first few lines and attempting to import it into a GRanges via `import.bed` or `import.bed15`. If an index file is supplied, it will attempt to use that index in `headerTabix`.

Value

A BedWrapper instance that can be used in `stageObject`.

Author(s)

Aaron Lun

Examples

```
# Mocking up a BED file.
tmp <- tempfile(fileext=".bed")
bed <- write("chr1\t2222\t33333", file=tmp)

# Creating a BedWrapper.
wrapped <- BedWrapper(tmp)
wrapped

# Staging the BedWrapper.
dir <- tempfile()
library(alabaster.base)
info <- stageObject(wrapped, dir, "my_bed")
invisible(.writeMetadata(info, dir))
list.files(dir, recursive=TRUE)

# Loading it back again:
meta <- acquireMetadata(dir, "my_bed/file.bed")
loadObject(meta, dir)
```

`BgzipIndexWrapper` *Wrapper for a Bgzip index file*

Description

Wrap a Bgzip index file for saving and loading in the **alabaster** framework.

Usage

```
BgzipIndexWrapper(path)
```

Arguments

`path`	String containing the path to a Bgzip index file.

Details

The BgzipIndexWrapper class is a subclass of a Wrapper, so all of the methods of the latter can also be used here, e.g., `path`.

Value

A BgzipIndexWrapper instance that can be used in `stageObject`.

Author(s)

Aaron Lun

Examples

```
# Mocking up a FASTA index file.
input <- system.file("extdata", "ce2dict1.fa", package="Rsamtools")
temp <- tempfile(fileext=".fa.bgz")
copy <- Rsamtools::bgzip(input, dest=temp)
Rsamtools::indexFa(copy)

# Creating a BgzipIndexWrapper.
wrapped <- BgzipIndexWrapper(paste0(copy, ".gzi"))
wrapped

# Staging the BgzipIndexWrapper.
dir <- tempfile()
library(alabaster.base)
info <- stageObject(wrapped, dir, "tab")
invisible(.writeMetadata(info, dir))
list.files(dir, recursive=TRUE)

# Loading it back again:
meta <- acquireMetadata(dir, "tab/file.fa.bgz.gzi")
loadObject(meta, dir)
```

`BigBedWrapper` *Wrapper for a bigBed file*

Description

Wrap a bigBed file for saving and loading in the **alabaster** framework.

Usage

```
BigBedWrapper(path)
```

Arguments

`path`	String containing the path to a bigBed file.

Details

The BigBedWrapper class is a subclass of a Wrapper, so all of the methods of the latter can also be used here, e.g., `path`.

The `stageObject` method for BigBedWrapper classes will check the bigBed file by searching for the bigBed magic number, i.e., 0x8789F2EB or its byte-reversed form.

Value

A BigBedWrapper instance that can be used in `stageObject`.

Author(s)

Aaron Lun

Examples

```
# Mocking up a bigBed file.
test_path <- system.file("tests", package = "rtracklayer")
test_bb <- file.path(test_path, "test.bb")

# Creating a BigBedWrapper.
wrapped <- BigBedWrapper(test_bb)
wrapped

# Staging the BigBedWrapper.
dir <- tempfile()
library(alabaster.base)
info <- stageObject(wrapped, dir, "my_bb")
invisible(.writeMetadata(info, dir))
list.files(dir, recursive=TRUE)

# Loading it back again:
meta <- acquireMetadata(dir, "my_bb/file.bb")
loadObject(meta, dir)
```

`BigWigWrapper` *Wrapper for a bigWig file*

Description

Wrap a bigWig file for saving and loading in the **alabaster** framework.

Usage

```
BigWigWrapper(path)
```

Arguments

`path`	String containing the path to a bigWig file.

Details

The BigWigWrapper class is a subclass of a Wrapper, so all of the methods of the latter can also be used here, e.g., `path`.

The `stageObject` method for BigWigWrapper classes will check the bigWig file by searching for the bigWig magic number, i.e., 0x888FFC26 or its byte-reversed form.

Value

A BigWigWrapper instance that can be used in `stageObject`.

Author(s)

Aaron Lun

Examples

```
# Mocking up a bigWig file.
test_path <- system.file("tests", package = "rtracklayer")
test_bw <- file.path(test_path, "test.bw")

# Creating a BigWigWrapper.
wrapped <- BigWigWrapper(test_bw)
wrapped

# Staging the BigWigWrapper.
dir <- tempfile()
library(alabaster.base)
info <- stageObject(wrapped, dir, "my_bw")
invisible(.writeMetadata(info, dir))
list.files(dir, recursive=TRUE)

# Loading it back again:
meta <- acquireMetadata(dir, "my_bw/file.bw")
loadObject(meta, dir)
```

`FaIndexWrapper` *Wrapper for a FASTA index file*

Description

Wrap a FASTA index file for saving and loading in the **alabaster** framework.

Usage

```
FaIndexWrapper(path)
```

Arguments

path	String containing the path to a FASTA index file.

Details

The FaIndexWrapper class is a subclass of a Wrapper, so all of the methods of the latter can also be used here, e.g., path.

Value

A FaIndexWrapper instance that can be used in stageObject.

Author(s)

Aaron Lun

Examples

```
# Mocking up a FASTA index file.
test_fai <- system.file("extdata", "ce2dict1.fa.fai", package="Rsamtools")

# Creating a FaIndexWrapper.
wrapped <- FaIndexWrapper(test_fai)
wrapped

# Staging the FaIndexWrapper.
dir <- tempfile()
library(alabaster.base)
info <- stageObject(wrapped, dir, "tab")
invisible(.writeMetadata(info, dir))
list.files(dir, recursive=TRUE)

# Loading it back again:
meta <- acquireMetadata(dir, "tab/file.fa.fai")
loadObject(meta, dir)
```

FastaWrapper *Wrapper for a FASTA file*

Description

Wrap a FASTA file for saving and loading in the **alabaster** framework.

Usage

```
FastaWrapper(
  path,
  sequence.type = "DNA",
  compression = NULL,
  index = NULL,
  gzindex = NULL
)
```

Arguments

`path` String containing the path to a FASTA file.

`sequence.type` String specifying the sequence type, should be one of `"DNA"`, `"RNA"` or `"AA"`.

`compression` String specifying the compression. This should be one of `"none"`, `"gzip"`, `"bzip2"` or `"bgzip"`. If `NULL`, this is inferred from the file's headers and suffix.

`index` String specifying the path to an FASTA index file, or `NULL` if no index is available. If an index is supplied, the file at `path` should be uncompressed or bgzip-compressed.

`gzindex` String specifying the path to a bgzip index file, or `NULL` if no index is available. If an bgzip index is supplied, the file at `path` should be bgzip-compressed. This index is mandatory if `index` is supplied and `path` is bgzip-compressed.

Details

The FastaWrapper class is a subclass of a CompressedIndexedWrapper, so all of the methods of the latter can also be used here, e.g., `path`, `index`, `compression`.

The `stageObject` method for FastaWrapper classes will check the FASTA file by reading the first few lines and attempting to import it into an XStringSet object using the relevant **Biostrings** functions, e.g., `readDNAStringSet`. If an index is supplied, the method will check its validity via `scanFaIndex`.

Value

A FastaWrapper instance that can be used in `stageObject`.

Author(s)

Aaron Lun

Examples

```
# Mocking up a FASTA file.
tmp <- tempfile(fileext=".fa")
write(">FOOBAR\nacgtacgt", tmp)

# Creating a FastaWrapper.
wrapped <- FastaWrapper(tmp)
wrapped
```

```
# Staging the FastaWrapper.
dir <- tempfile()
library(alabaster.base)
info <- stageObject(wrapped, dir, "seq")
invisible(.writeMetadata(info, dir))
list.files(dir, recursive=TRUE)

# Loading it back again:
meta <- acquireMetadata(dir, "seq/file.fa")
loadObject(meta, dir)
```

FastqWrapper *Wrapper for a FASTQ file*

Description

Wrap a FASTQ file for saving and loading in the **alabaster** framework.

Usage

```
FastqWrapper(
  path,
  encoding,
  sequence.type = "DNA",
  compression = NULL,
  index = NULL,
  gzindex = NULL
)
```

Arguments

`path` String containing the path to a FASTQ file.

`encoding` String specifying the encoding of the quality strings. This should be one of `"phred"`, `"solexa"` or `"illumina"`.

`sequence.type` String specifying the sequence type, should be one of `"DNA"`, `"RNA"` or `"AA"`.

`compression` String specifying the compression. This should be one of `"none"`, `"gzip"`, `"bzip2"` or `"bgzip"`. If `NULL`, this is inferred from the file's headers and suffix.

`index` String specifying the path to an faidx file, or `NULL` if no index is available. If an index is supplied, the file should be uncompressed or bgzip-compressed.

`gzindex` String specifying the path to a bgzip index file, or `NULL` if no index is available. If an bgzip index is supplied, the file at `path` should be bgzip-compressed. This index is mandatory if `index` is supplied and `path` is bgzip-compressed.

Details

The FastqWrapper class is a subclass of a CompressedIndexedWrapper, so all of the methods of the latter can also be used here, e.g., `path`, `index`, `compression`.

The `stageObject` method for FastqWrapper classes will check the FASTQ file by reading the first few lines and attempting to import it into an XStringSet object using the relevant **Biostrings** functions, e.g., `readDNAStringSet`.

Value

A FastqWrapper instance that can be used in `stageObject`.

Author(s)

Aaron Lun

Examples

```
# Mocking up a FASTQ file.
tmp <- tempfile(fileext=".fq")
write("@FOOBAR\nacgtacgt\n+134987382", tmp)

# Creating a FastqWrapper.
wrapped <- FastqWrapper(tmp, encoding="phred")
wrapped

# Staging the FastqWrapper.
dir <- tempfile()
library(alabaster.base)
info <- stageObject(wrapped, dir, "seq")
invisible(.writeMetadata(info, dir))
list.files(dir, recursive=TRUE)

# Loading it back again:
meta <- acquireMetadata(dir, "seq/file.fastq")
loadObject(meta, dir)
```

`GffWrapper` *Wrapper for a GFF file*

Description

Wrap a GFF2/3 file for saving and loading in the **alabaster** framework.

Usage

```
GffWrapper(path, compression = NULL, index = NULL, format = NULL)
```

Arguments

`path` String containing the path to a GFF file.

`compression` String specifying the compression. This should be one of `"none"`, `"gzip"`, `"bzip2"` or `"bgzip"`. If `NULL`, this is inferred from the file's headers and suffix.

`index` String specifying the path to an index file in tabix format, or `NULL` if no index is available. If an index is supplied, the file should be bgzip-compressed.

`format` String specifying the format of the GFF file. This should be one of `"GFF2"` (i.e., GTF) or `"GFF3"`. If `NULL`, this is inferred from the file extension of `path`.

Details

The GffWrapper class is a subclass of a CompressedIndexedWrapper, so all of the methods of the latter can also be used here, e.g., `path`, `index`, `compression`.

The `stageObject` method for GffWrapper classes will check the GFF file by reading the first few lines and attempting to import it into a GRanges via `import.gff2` or `import.gff3`. If an index file is supplied, it will attempt to use that index in `headerTabix`.

Value

A GffWrapper instance that can be used in `stageObject`.

Author(s)

Aaron Lun

Examples

```
# Using rtracklayer's example GFF file.
test_path <- system.file("tests", package = "rtracklayer")
test_gff3 <- file.path(test_path, "genes.gff3")

# Creating a GffWrapper.
wrapped <- GffWrapper(test_gff3)
wrapped

# Staging the GffWrapper.
dir <- tempfile()
library(alabaster.base)
info <- stageObject(wrapped, dir, "my_gff")
invisible(.writeMetadata(info, dir))
list.files(dir, recursive=TRUE)

# Loading it back again:
meta <- acquireMetadata(dir, "my_gff/file.gff3")
loadObject(meta, dir)
```

GmtWrapper *Wrapper for a GMT file*

Description

Wrap a GMT file for saving and loading in the **alabaster** framework.

Usage

```
GmtWrapper(path, compression = NULL)
```

Arguments

`path`	String containing the path to a GMT file.
`compression`	String specifying the compression. This should be one of `"none"`, `"gzip"`, `"bzip2"` or `"bgzip"`. If `NULL`, this is inferred from the file's headers and suffix.

Details

The GmtWrapper class is a subclass of a CompressedWrapper, so all of the methods of the latter can also be used here, e.g., `path`, `compression`.

The `stageObject` method for GmtWrapper classes will check the GMT file by reading the first few lines and checking that each line contains at least three tab-separated fields.

Value

A GmtWrapper instance that can be used in `stageObject`.

Author(s)

Aaron Lun

Examples

```
# Mocking up a GMT file.
tmp <- tempfile(fileext=".gmt")
write("SET1\tdescription\tgene1\tgene2\tgene3", file=tmp)

# Creating a GmtWrapper.
wrapped <- GmtWrapper(tmp)
wrapped

# Staging the GmtWrapper.
dir <- tempfile()
library(alabaster.base)
info <- stageObject(wrapped, dir, "my_gmt")
invisible(.writeMetadata(info, dir))
list.files(dir, recursive=TRUE)
```

```
# Loading it back again:
meta <- acquireMetadata(dir, "my_gmt/file.gmt")
loadObject(meta, dir)
```

TabixIndexWrapper *Wrapper for a Tabix file*

Description

Wrap a Tabix file for saving and loading in the **alabaster** framework.

Usage

```
TabixIndexWrapper(path)
```

Arguments

path	String containing the path to a Tabix file.

Details

The TabixIndexWrapper class is a subclass of a Wrapper, so all of the methods of the latter can also be used here, e.g., path.

Value

A TabixIndexWrapper instance that can be used in stageObject.

Author(s)

Aaron Lun

Examples

```
# Mocking up a Tabix file.
test_tbx <- system.file("extdata", "example.gtf.gz.tbi", package="Rsamtools")

# Creating a TabixIndexWrapper.
wrapped <- TabixIndexWrapper(test_tbx)
wrapped

# Staging the TabixIndexWrapper.
dir <- tempfile()
library(alabaster.base)
info <- stageObject(wrapped, dir, "tab")
invisible(.writeMetadata(info, dir))
list.files(dir, recursive=TRUE)
```

```
# Loading it back again:
meta <- acquireMetadata(dir, "tab/file.tbi")
loadObject(meta, dir)
```

VcfWrapper *Wrapper for a GFF file*

Description

Wrap a GFF2/3 file for saving and loading in the **alabaster** framework.

Usage

```
VcfWrapper(path, compression = NULL, index = NULL, header_only = FALSE)
```

Arguments

path	String containing the path to a GFF file.
compression	String specifying the compression. This should be one of "none", "gzip", "bzip2" or "bgzip". If NULL, this is inferred from the file's headers and suffix.
index	String specifying the path to an index file in tabix format, or NULL if no index is available. If an index is supplied, the file should be bgzip-compressed.
header_only	Logical scalar specifying whether this file just contains the VCF headers without any actual data. Occasionally useful for applications to perform preflight requests before downloading the full dataset.

Details

The VcfWrapper class is a subclass of a CompressedIndexedWrapper, so all of the methods of the latter can also be used here, e.g., path, index, compression.

The stageObject method for VcfWrapper classes will check the GFF file by reading the first few lines and attempting to import it into a GRanges via readVCF from the **VariantAnnotation** package. If an index file is supplied, it will attempt to use that index in headerTabix.

Value

A VcfWrapper instance that can be used in stageObject.

Author(s)

Aaron Lun

Examples

```
# Using VariantAnnotations's example file.
fl <- system.file("extdata", "structural.vcf", package="VariantAnnotation")

# Creating a VcfWrapper.
wrapped <- VcfWrapper(fl)
wrapped

# Staging the VcfWrapper.
dir <- tempfile()
library(alabaster.base)
info <- stageObject(wrapped, dir, "my_vcf")
invisible(.writeMetadata(info, dir))
list.files(dir, recursive=TRUE)

# Loading it back again:
meta <- acquireMetadata(dir, "my_vcf/file.vcf")
loadObject(meta, dir)
```

virtual-wrapper *Virtual wrapper classes*

Description

Defines some base classes for the concrete wrappers for specific file formats. This provides a standard set of methods that can be applied to all Wrapper instances.

Wrapper methods

Any instance `x` of a base Wrapper class can be used with the `path(x)` method, which returns a string containing the path to the file on the current file system.

The Wrapper class inherits from the Annotated class, so users can also get and set metadata via `metadata(x)`.

IndexedWrapper methods

The IndexedWrapper class inherits from the Wrapper class and can be used with all its methods. It additionally implements the `index(x)` method, which returns another Wrapper object for the associated index file (or `NULL`, if no index file exists).

CompressedWrapper methods

The CompressedWrapper class inherits from the Wrapper class and can be used with all its methods. It additionally implements the `compression(x)` method, which returns a string specifying the compression strategy.

CompressedIndexedWrapper methods

The CompressedIndexedWrapper class inherits from both the IndexedWrapper and CompressedWrapper classes and can be used with all their methods.

Author(s)

Aaron Lun

Index