Package ‘rhdf5client’

January 13, 2024

Title  Access HDF5 content from HDF Scalable Data Service

Description  This package provides functionality for reading data from HDF Scalable Data Service from within R. The HSDSArray function bridges from HSDS to the user via the DelayedArray interface. Bioconductor manages an open HSDS instance graciously provided by John Readey of the HDF Group.

Version  1.24.0

Suggests  knitr, testthat, BiocStyle, DT, rmarkdown

Imports  httr, rjson, utils, data.table

Depends  R (>= 3.6), methods, DelayedArray

License  Artistic-2.0

LazyLoad  yes


‘Dataset.R’ ‘RHDF5Array.R’

VignetteBuilder  knitr

Encoding  UTF-8

biocViews  DataImport, Software, Infrastructure

RoxygenNote  7.2.3

git_url  https://git.bioconductor.org/packages/rhdf5client

git_branch  RELEASE_3_18

git_last_commit  4f73a04

git_last_commit_date  2023-10-24

Repository  Bioconductor 3.18

Date/Publication  2024-01-12

Author  Samuela Pollack [aut],

Shweta Gopaulakrishnan [aut],

BJ Stubbs [aut],

Alexey Sergushichev [aut],

Vincent Carey [cre, aut]

Maintainer  Vincent Carey <stvjc@channing.harvard.edu>
R topics documented:

as ....................................................... 2
check_hsd ........................................... 3
dim ................................................... 3
dimnames .......................................... 4
extractCompoundJSON ............................... 4
extract_array ...................................... 5
getData ............................................ 5
HSDSArray .......................................... 6
HSDSArraySeed ..................................... 7
HSDSDataset ........................................ 8
HSDSDataset-class ................................. 8
HDSDFile ........................................... 9
HDSDFile-class .................................... 9
HSDSMatrix ......................................... 10
HSDSSource ........................................ 10
HSDSSource-class ................................ 11
isplit ............................................... 11
listDatasets ....................................... 12
listDomains ....................................... 12
rhdf5client ....................................... 13
rhdf5client-deprecated ......................... 14
URL_hsd ........................................... 15
[,HSDSDataset,numeric,ANY,ANY-method .......... 15
[,HSDSDataset,numeric,numeric,ANY-method ...... 16

Index 17

as  ................................................ Coercion method from HSDSMatrix to its superclass HSDSArray

Description

Coercion method from HSDSMatrix to its superclass HSDSArray

See Also

Other HSDSArray: HSDSArray, HSDSMatrix
check_hsd

---

check_hsd  a test request

Description

a test request

Usage

check_hsd()

Value

logical, TRUE if hsd behaving as expected

Examples

check_hsd()

dim

Obtain dimensions of an object of type HSDSArraySeed

---

Description

(required by DelayedArray seed contract) HDF server content is assumed transposed relative to R matrix layout. This anticipates H5 datasets on the server with rows for experimental samples and columns for *-omic features. The Bioconductor SummarizedExperiment requires *-omic features in rows and samples in columns.

Usage

## S4 method for signature 'HSDSArraySeed'
dim(x)

Arguments

x  An object of type HSDSArraySeed

Value

A numeric vector of the dimensions
dimnames

Obtain names of dimensions for an object of type HSDSArraySeed

Description

(required by DelayedArray seed contract, returns NULL list)

Usage

```r
## S4 method for signature 'HSDSArraySeed'
dimnames(x)
```

Arguments

- **x**: An object of type HSDSArraySeed

Value

A NULL list of length equal to the array dimensionality

extractCompoundJSON

compound operation

Description

compound operation

Usage

```r
extractCompoundJSON(type, value)
```

Arguments

- **type**: type
- **value**: value
extract_array

Access dataset backed by an HSDataSetSeed

Description

Access dataset backed by an HSDataSetSeed

Usage

```
# S4 method for signature 'HSDataSetSeed'
extract_array(x, index)
```

Arguments

- `x`: An object of type HSDataSetSeed
- `index`: A list of numeric vectors to be accessed, one vector for each dimension of the array object. A NULL vector indicates the entire range of indices in that dimension. A zero-length vector indicates no indices in the relevant dimension. (Accordingly, any zero-length vector of indices will result in an empty array being returned.)

Value

An array containing the data elements corresponding to the indices requested

dataGet

extract elements of a one or two-dimensional HSDataSet

Description

Fetch data from a remote dataset

Usage

```
dataGet(dataset, indices, transfermode)
```

```
# S4 method for signature 'HSDataSet,character,character'
dataGet(dataset, indices, transfermode)
```

```
# S4 method for signature 'HSDataSet,character,missing'
dataGet(dataset, indices)
```

```
# S4 method for signature 'HSDataSet,list,character'
dataGet(dataset, indices, transfermode)
```

```
# S4 method for signature 'HSDataSet,list,missing'
dataGet(dataset, indices)
```
Arguments

- **dataset**: An object of type HSDSDataset, the dataset to access.
- **indices**: The indices of the data to fetch
- **transfermode**: Either 'JSON' or 'binary' (default)

Details

The servers require data to be fetched in slices, i.e., in sets of for which the indices of each dimension are of the form start:stop:step. More complex sets of indices will be split into slices and fetched in multiple requests. This is opaque to the user, but may enter into considerations of data access patterns, e.g., for performance-tuning.

Value

an Array containing the data fetched from the server

Examples

```r
if (check_hsds()) {
  s <- HSDSSource(URL_hsds())
  f <- HSDSfile(s, '/shared/bioconductor/patelGBMSC.h5')
  d <- HSDSDataset(f, '/assay001')
  x <- getData(d, c(1:4, 1:27998), transfermode='JSON')
  xb <- getData(d, c(1:4, 1:27998), transfermode='binary')
  # x <- getData(d, c(1:4, 1:27998), transfermode='JSON') # method missing?
  x
  xb
}
```

---

HSDSArray

A DelayedArray backend for accessing a remote HDF5 server.

Description

A DelayedArray backend for accessing a remote HDF5 server.

Construct an object of type HSDSArray directly from the data members of its seed

Usage

HSDSArray(endpoint, svrtype, domain, dsetname)

Arguments

- **endpoint**: URL of remote server
- **svrtype**: type of server, must be either 'hsds' or 'h5serv'
- **domain**: HDF5 domain of H5 file on server
- **dsetname**: complete internal path to dataset in H5 file
**HSDSArraySeed**

**Value**

An initialized object of type HSDSArray

**See Also**

Other HSDSArray: **HSDSMatrix, as**()

**Examples**

```r
if (check_hsds()) {
  HSDSArray(URL_hsds(),
            "hsds", "/shared/bioconductor/darmgcls.h5", "/assay001")
}
```

---

**HSDSArraySeed for HSDSArray backend to DelayedArray**

**Description**

HSDSArraySeed for HSDSArray backend to DelayedArray

Construct an object of type HSDSArraySeed

**Usage**

`HSDSArraySeed(endpoint, svrtype, domain, dsetname)`

**Arguments**

- **endpoint** URL of remote server
- **svrtype** type of server, must be either 'hsds' or 'h5serv'
- **domain** HDF5 domain of H5 file on server
- **dsetname** complete internal path to dataset in H5 file

**Value**

An initialized object of type HSDSArraySeed

**Slots**

- **endpoint** URL of remote server
- **svrtype** type of server, must be either 'hsds' or 'h5serv'
- **domain** HDF5 domain of H5 file on server
- **dsetname** complete internal path to dataset in H5 file
- **dataset** object of type HSDSDataset for access to the H5 dataset
HSDSDataset-class

**HSDSDataset**

*Construct an object of type HSDSDataset* A HSDSDataset is a representation of a dataset in a HDF5 file.

**Description**

Construct an object of type HSDSDataset A HSDSDataset is a representation of a dataset in a HDF5 file.

**Usage**

HSDSDataset(file, path)

**Arguments**

- **file** 
  An object of type HSDSFile which hosts the dataset
- **path** 
  The complete intrafile path to the dataset

**Value**

An initialized object of type HSDSDataset

**Examples**

```r
if (check_hsds()) {
  src <- HSDSSource(URL_hsds())
  f <- HSDSFile(src, '/shared/bioconductor/patelGBMSC.h5')
  d <- HSDSDataset(f, '/assay001')
}
```

HSDSDataset-class

*An S4 class to represent a dataset in a HDF5 file.*

**Description**

An S4 class to represent a dataset in a HDF5 file.

**Slots**

- **file** 
  An object of type HSDSFile; the file in which the dataset is resident.
- **path** 
  The dataset’s path in the internal HDF5 hierarchy.
- **uuid** 
  The unique unit ID by which the dataset is accessed in the server database system.
- **shape** 
  The dimensions of the dataset
- **type** 
  The dataset’s HDF5 datatype
**HSDSFile**  

Construct an object of type HSDSFile

---

**Description**

A HSDSFile is a representation of an HDF5 file the contents of which are accessible exposed by a HDF5 server.

**Usage**

HSDSFile(src, domain)

**Arguments**

- **src**: an object of type HSDSSource, the server which exposes the file
- **domain**: the domain string; the file’s location on the server’s file system.

**Details**

This function is deprecated and will be defunct in the next release.

**Value**

an initialized object of type HSDSFile

**Examples**

```r
if (check_hsds()) {
  src <- HSDSSource(URL_hsds())
  f10x <- HSDSFile(src, '/shared/bioconductor/patelGBMSC.h5')
}
```

---

**HSDSFile-class**  

An S4 class to represent an HDF5 file accessible from a server.

---

**Description**

An S4 class to represent an HDF5 file accessible from a server.

**Slots**

- **HSDSSource**: an object of type HSDSSource
- **domain**: the file’s domain on the server; more or less, an alias for its location in the external server file system
- **dsetdf**: a data.frame that caches often-used information about the file
HSDSSource

**HSDSMatrix**

*DelayedMatrix subclass for a two-dimensional HSDSArray*

**Description**

DelayedMatrix subclass for a two-dimensional HSDSArray

**See Also**

Other HSDSArray: HSDSArray, as()

---

HSDSSource

*Construct an object of type HSDSSource.*

**Description**

A HSDSSource is a representation of a URL which provides access to a HDF5 server (either h5serv or hsd).

**Usage**

HSDSSource(endpoint, type = "hsd")

**Arguments**

- **endpoint**: URL for server
- **type**: Type of server software at the source; must be

**Details**

This function is deprecated and will be defunct in the next release.

**Value**

An object of type HSDSSource

**Examples**

```r
if (check_hsd()) {
  src.hsd <- HSDSSource(URL_hsd())
}
```
HSDSSource-class

An S4 class to represent a HDF5 server listening on a port.

Description

This class is deprecated and will be defunct in the next release.

Slots

endpoint URL for server
type Type of server software at the source; must be either ‘h5serv’ or (default) ‘hsds’

isplit

This function is deprecated and will be defunct in the next release.

Description

isplit converts a numeric vector into a list of sequences for compact reexpression

Usage

isplit(x)
sproc(spl)

Arguments

x a numeric vector (should be integers)
spl output of isplit

Value

list of vectors of integers which can be expressed as initial/final/stride triplets
list of colon-delimited strings each with initial/final/stride triplet

Examples

inds = c(1:10, seq(25,50,2), seq(200,150,-2))
sproc(isplit(inds))
**listDomains**

**Search inner file hierarchy for datasets**

**Description**

The datasets in an HDF5 file are organized internally by groups. This routine traverses the internal group hierarchy, locates all datasets and prints a list of them. Note that if the file’s group hierarchy is complex, this could be time-consuming.

**Usage**

`listDatasets(file)`

**Arguments**

- **file**: an object of type HSDSFile to be searched

**Details**

This function is deprecated and will be defunct in the next release.

**Value**

a list of inner-paths

**Examples**

```r
if (check_hsd()) {
  src <- HSDDSSource(URL_hsd())
  f <- HSDSFile(src, '/shared/bioconductor/patelGBMSC.h5')
  listDatasets(f)
}
```

---

**listDomains**

**List files and subdirectories of a domain**

**Description**

The user needs to give the domain to start in. The search will be non-recursive. I.e., output for domain `/home/jreadey/` will not return the files in `/home/jreadey/HDFLabTutorial/`
Usage

listDomains(object, rootdir)

## S4 method for signature 'HSDSSource,character'
listDomains(object, rootdir)

## S4 method for signature 'HSDSSource,missing'
listDomains(object)

Arguments

object       An object of type HSDSSource
rootdir      A slash-separated directory in the HSDSSource file system.

Details

This function is deprecated and will be defunct in the next release.

Value

a vector of domains in the rootdir

Examples

src.hsds <- HSDSSource(URL_hsds())
listDomains(src.hsds, '/shared')

Description

The rhdf5client package provides read-only access to HDF5 files maintained on a server. The HDFGroup provides two servers, an obsolescent one called ‘h5serv’ and the newer prototype called ‘hsds’.
Description

These functions are provided for compatibility with older versions of ‘rhdf5client’ only, and will be defunct at the next release.

Details

The following functions are deprecated and will be made defunct in the next release:

- URL_h5serv
- URL_hdsds
- dsmeta
- getReq
- groups
- setPath
- links
- transfermode
- dataset
- internalDim
- hdsdsInfo
- domains
- getDatasetUUIDs
- getDatasetAttrs
- getDims
- getHRDF
- H5S_dataset2
- getDatasetSlice
- fetchDatasets
- isplit
- sproc
- listDomains
- listDatasets
- getData

The following classes are deprecated and will be made defunct in the next release:

- H5S_source
**URL_hsds**

- H5S_dataset
- H5S_Array
- H5S_Matrix
- HSDSSource
- HSDFFile
- HSDSDataset

---

**Description**

manage hsds URL

**Usage**

```
URL_hsds()
```

**Value**

URL of hsds server

**Examples**

```
URL_hsds()
```

---

```r
[,HSDSDataset,numeric,ANY,ANY-method
  
  bracket method for 1d request from HSDSDataset
```

---

**Description**

bracket method for 1d request from HSDSDataset

**Usage**

```
## S4 method for signature 'HSDSDataset,numeric,ANY,ANY'

x[i, j, ..., drop = TRUE]
```

**Arguments**

- **x**: object of type HSDSDataset
- **i**: vector of indices (first dimension)
- **j**: not used
- **...**: not used
- **drop**: logical(1) if TRUE return has no array character
Value

an array with the elements requested from the HSDSDataset

Description

bracket method for 2d request from HSDSDataset

Usage

## S4 method for signature 'HSDSDataset,numeric,numeric,ANY'

x[i, j, ..., drop = TRUE]

Arguments

x object of type HSDSDataset
i vector of indices (first dimension)
j vector of indices (second dimension)
... not used
drop logical(1) if TRUE return has no array character

Value

an array with the elements requested from the HSDSDataset
Index

* HSDSArray  
  as, 2
  HSDSArray, 6
  HSDSMatrix, 10
  [,HSDDataset,numeric,ANY,ANY-method, 15
  [,HSDDataset,numeric,numeric,ANY-method, 16
  as, 2, 7, 10
  check_hsds, 3
  dim, 3
  dim,HSDSArraySeed-method(dim), 3
  dimnames,HSDSArraySeed-method
  (dimnames), 4
  extract_array, 5
  extract_array,HSDSArraySeed-method
  (extract_array), 5
  extractCompoundJSON, 4

  getData, 5
  getData,HSDDataset,character,character-method
  (getData), 5
  getData,HSDDataset,character,missing-method
  (getData), 5
  getData,HSDDataset,list,missing-method
  (getData), 5
  getData,HSDDataset,list,character-method
  (getData), 5
  getData,HSDDataset,list,missing-method
  (getData), 5

HSDSArray, 2, 6, 10
HSDSArray-class (HSDSArray), 6
HSDSArraySeed, 7
HSDSArraySeed-class (HSDSArraySeed), 7
HSDDataset, 8

HSDDataset-class, 8
HSDSFile, 9
HSDSFile-class, 9
HSDSMatrix, 2, 7, 10
HSDSMatrix-class (HSDSMatrix), 10
HSDDSource, 10
HSDDSource-class, 11

isplit, 11
listDatasets, 12
listDomains, 12
listDomains,HSDDSource,character-method
  (listDomains), 12
listDomains,HSDDSource,missing-method
  (listDomains), 12

rhdf5client, 13
rhdf5client-deprecated, 14

sproc (isplit), 11

URL_hsds, 15