Package 'AnnotationHub'

October 18, 2022

Type Package

Title Client to access AnnotationHub resources

Version 3.4.0

biocViews Infrastructure, DataImport, GUI, ThirdPartyClient

Description This package provides a client for the Bioconductor AnnotationHub web resource. The AnnotationHub web resource provides a central location where genomic files (e.g., VCF, bed, wig) and other resources from standard locations (e.g., UCSC, Ensembl) can be discovered. The resource includes metadata about each resource, e.g., a textual description, tags, and date of modification. The client creates and manages a local cache of files retrieved by the user, helping with quick and reproducible access.

License Artistic-2.0

Depends BiocGenerics (>= 0.15.10), BiocFileCache (>= 1.5.1)

Imports utils, methods, grDevices, RSQLite, BiocManager, BiocVersion, curl, rappdirs, AnnotationDbi (>= 1.31.19), S4Vectors, interactiveDisplayBase, httr, yaml, dplyr

Suggests IRanges, GenomicRanges, GenomeInfoDb, VariantAnnotation, Rsamtools, rtracklayer, BiocStyle, knitr, AnnotationForge, rBiopaxParser, RUnit, GenomicFeatures, MSnbase, mzR, Biostrings, SummarizedExperiment, ExperimentHub, gdsfmt, rmarkdown, HubPub

Enhances AnnotationHubData

Collate AnnotationHubOption.R AllGenerics.R Hub-class.R db-utils.R AnnotationHub-class.R AnnotationHubResource-class.R BEDResource-class.R ProteomicsResource-class.R EpigenomeResource-class.R EnsDbResource-class.R utilities.R sql-utils.R Hub-utils.R cache-utils.R zzz.R

VignetteBuilder knitr

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

NeedsCompilation yes

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

git_branch RELEASE_3_15

git_last_commit e74e54c

git_last_commit_date 2022-04-26

Date/Publication 2022-10-18

Author Bioconductor Package Maintainer [cre],
Martin Morgan [aut],
Marc Carlson [ctb],
Dan Tenenbaum [ctb],
Sonali Arora [ctb],
Valerie Oberchain [ctb],
Kayla Morrell [ctb],
Lori Shepherd [aut]

Maintainer Bioconductor Package Maintainer <maintainer@bioconductor.org>

R topics documented:

`AnnotationHub-package` *Light-weight AnnotationHub 3.0 Client*

Description

Client for discovery and retrieval of Bioconductor annotation resources.

Author(s)

Martin Morgan mtmorgan@fhcrc.org

See Also

`AnnotationHub-class`

Examples

```
## Not run:
library(AnnotationHub)
hub = AnnotationHub()
hub

## End(Not run)
```

AnnotationHub-objects *AnnotationHub objects and their related methods and functions*

Description

Use AnnotationHub to interact with Bioconductor's AnnotationHub service. Query the instance to discover and use resources that are of interest, and then easily download and import the resource into R for immediate use.

Use AnnotationHub() to retrieve information about all records in the hub. If working offline, add argument localHub=TRUE to work with a local, non-updated hub; It will only have resources available that have previously been downloaded. If offline, Please also see BiocManager vignette section on offline use to ensure proper funcionality. To force redownload of the hub, refreshHub(hubClass="AnnotationHub") can be utilized.

Discover records in a hub using mcols(), query(), subset(), [, and display().

Retrieve individual records using [[. On first use of a resource, the corresponding files or other hub resources are downloaded from the internet to a local cache. On this and all subsequent uses the files are quickly input from the cache into the R session. If a user wants to download the file again and not use the cache version add the argument force=TRUE.

AnnotationHub records can be added (and sometimes removed) at any time. snapshotDate() restricts hub records to those available at the time of the snapshot. possibleDates() lists snapshot dates valid for the current version of Bioconductor. You can check the status of a past record using recordStatus().

The location of the local cache can be found (and updated) with getAnnotationHubCache and setAnnotationHubCache; removeCache removes all cache resources.

For common hub troubleshooting, please see the AnnotationHub vignette entitled 'vignette("TroubleshootingTheCache", package=AnnotationHub)'.

Constructors

AnnotationHub(..., hub=getAnnotationHubOption("URL"),cache=getAnnotationHubOption("CACHE"),proxy=ge

Create an AnnotationHub instance, possibly updating the current database of records.

Accessors

In the code snippets below, x and object are AnnotationHub objects.

hubCache(x): Gets the file system location of the local AnnotationHub cache.

`hubUrl(x)`: Gets the URL for the online hub.

`isLocalHub(x)`: Get whether or not constructor was called with `localHub=TRUE`.

`length(x)`: Get the number of hub records.

`names(x)`: Get the names (AnnotationHub unique identifiers, of the form AH12345) of the hub records.

`fileName(x)`: Get the file path of the hub records as stored in the local cache (AnnotationHub files are stored as unique numbers, of the form 12345). NA is returned for those records which have not been cached.

`mcols(x)`: Get the metadata columns describing each record. Columns include:

title Record title, frequently the file name of the object.

dataprovider Original provider of the resource, e.g., Ensembl, UCSC.

species The species for which the record is most relevant, e.g., 'Homo sapiens'.

taxonomyid NCBI taxonomy identifier of the species.

genome Genome build relevant to the record, e.g., hg19.

description Textual description of the resource, frequently automatically generated from file path and other information available when the record was created.

tags Single words added to the record to facilitate identification, e.g,. TCGA, Roadmap.

rdataclass The class of the R object used to represent the object when imported into R, e.g., `GRanges`, `VCFFile`.

sourceurl Original URL of the resource.

sourectype Format of the original resource, e.g., BED file.

`dbconn(x)`: Return an open connection to the underyling SQLite database.

`dbfile(x)`: Return the full path the underyling SQLite database.

`.db_close(conn)`: Close the SQLite connection `conn` returned by `dbconn(x)`.

Subsetting and related operations

In the code snippets below, `x` is an AnnotationHub object.

`x$name`: Convenient reference to individual metadata columns, e.g., `x$species`.

`x[i]`: Numerical, logical, or character vector (of AnnotationHub names) to subset the hub, e.g., `x[x$species == "Homo sapiens"]`.

`x[[i, force=FALSE, verbose=TRUE]]`: Numerical or character scalar to retrieve (if necessary) and import the resource into R. If a user wants to download the file again and not use the cache version add the argument `force=TRUE`. `verbose=FALSE` will quiet status messages.

`query(x, pattern, ignore.case=TRUE, pattern.op= `&`)`: Return an AnnotationHub subset containing only those elements whose metadata matches `pattern`. Matching uses `pattern` as in `grepl` to search the `as.character` representation of each column, performing a logical `&` across columns. e.g., `query(x, c("Homo sapiens", "hg19", "GTF"))`.

`pattern` A character vector of patterns to search (via `grepl`) for in any of the `mcols()` columns.

`ignore.case` A logical(1) vector indicating whether the search should ignore case (TRUE) or not (FALSE).

`pattern.op` Any function of two arguments, describing how matches across pattern elements are to be combined. The default `` `&` `` requires that only records with *all* elements of `pattern` in their metadata columns are returned. `` `&` ``, `` `|` `` and `` `!` `` are most notably available. See `"?&"` or `?base::Ops` for more information.

`subset(x, subset)`: Return the subset of records containing only those elements whose metadata satisfies the *expression* in `subset`. The expression can reference columns of `mcols(x)`, and should return a logical vector of length `length(x)`. e.g., `subset(x, species == "Homo sapiens" & genome=="GRCh38")`.

`display(object)`: Open a web browser allowing for easy selection of hub records via interactive tabular display. Return value is the subset of hub records identified while navigating the display.

`recordStatus(hub, record)`: Returns a `data.frame` of the record id and status. `hub` must be a `Hub` object and `record` must be a `character(1)`. Can be used to discover why a resource was removed from the hub.

Cache and hub management

In the code snippets below, `x` is an AnnotationHub object.

`snapshotDate(x)` and `snapshotDate(x) <- value`: Gets or sets the date for the snapshot in use. `value` should be one of `possibleDates()`.

`possibleDates(x)`: Lists the valid snapshot dates for the version of Bioconductor that is being run (e.g., BiocManager::version()).

`cache(x)` and `cache(x) <- NULL`: Adds (downloads) all resources in `x`, or removes all local resources corresponding to the records in `x` from the cache. In the later case, `x` would typically be a small subset of AnnotationHub resources. If `x` is a subset hub from a larger hub, and `localHub=TRUE` was used to construct the hubs, the original object will need to be reconstructed to reflect the removed resources. See also `removeResources` for a nicer interface for removing cached resources, or `removeCache` for deleting the hub cache entirely.

`hubUrl(x)`: Gets the URL for the online AnnotationHub.

`hubCache(x)`: Gets the file system location of the local AnnotationHub cache.

`refreshHub(..., hub, cache, proxy,hubClass=c("AnnotationHub", "ExperimentHub"))`: Force redownload of Hub sqlite file. This returns a Hub object as if calling the constructor (ie. AnnotationHub()). For force redownload specifically for AnnotationHub the base call should be `refreshHub(hubClass="AnnotationHub")`

`removeResources(hub, ids)`: Removes listed ids from the local cache. ids are "AH" ids. Returns an updated hub object. To work with updated hub object suggested syntax is to reassign (ie. `hub = removeResources(hub, "AH1")`). If ids are missing will remove all previously downloaded local resources.

`removeCache(x, ask=TRUE)`: Removes local AnnotationHub database and all related resources. After calling this function, the user will have to download any AnnotationHub resources again.

Coercion

In the code snippets below, `x` is an AnnotationHub object.

`as.list(x)`: Coerce `x` to a list of hub instances, one entry per element. Primarily for internal use.

`c(x, ...)`: Concatenate one or more sub-hub. Sub-hubs must reference the same AnnotationHub instance. Duplicate entries are removed.

Author(s)

Martin Morgan, Marc Carlson, Sonali Arora, Dan Tenenbaum, and Lori Shepherd

See Also

`getInfoOnIds`

Examples

```
## create an AnnotationHub object
library(AnnotationHub)
ah = AnnotationHub()

## Summary of available records
ah

## Detail for a single record
ah[1]

## and what is the date we are using?
snapshotDate(ah)

## how many resources?
length(ah)

## from which resources, is data available?
head(sort(table(ah$dataprovider), decreasing=TRUE))

## from which species, is data available ?
head(sort(table(ah$species),decreasing=TRUE))

## what web service and local cache does this AnnotationHub point to?
hubUrl(ah)
hubCache(ah)

### Examples ###

## One can  search the hub for multiple strings
ahs2 <- query(ah, c("GTF", "77","Ensembl", "Homo sapiens"))

## information about the file can be retrieved using
ahs2[1]

## one can further extract information from this show method
## like the sourceurl using:
ahs2$sourceurl
ahs2$description
ahs2$title
```

```
## We can download a file by name like this (using a list semantic):
gr <- ahs2[[1]]
## And we can also extract it by the names like this:
res <- ah[["AH28812"]]

## the gtf file is returned as a GenomicRanges object and contains
## data about which organism it belongs to, its seqlevels and seqlengths
seqinfo(gr)

## each GenomicRanges contains a metadata slot which can be used to get
## the name of the hub object and other associated metadata.
metadata(gr)
ah[metadata(gr)$AnnotationHubName]

## And we can also use "[" to restrict the things that are in the
## AnnotationHub object (by position, character, or logical vector).
## Here is a demo of position:
subHub <- ah[1:3]

if(interactive()) {
  ## Display method involves user interaction through web interface
  ah2 <- display(ah)
}

## recordStatus
recordStatus(ah, "TEST")
recordStatus(ah, "AH7220")
```

---

convertHub *Convert old Hub to new Hub structure*

Description

The Hub class was updated to utilize BiocFileCache to allow for file level caching control. This update changed the way files were stored and named. As a convenience for AnnotationHub and ExperimentHub we have provided this helper function to try to re-download files and add them into the BiocFileCache tracking database.

Usage

```
convertHub(oldcachepath=NULL,
           newcachepath=NULL,
           hubType=c("AnnotationHub", "ExperimentHub"),
           proxy=getAnnotationHubOption("PROXY"),
           max.downloads=getAnnotationHubOption("MAX_DOWNLOADS"),
           force=FALSE, verbose=TRUE)
```

Arguments

oldcachepath	character(1) complete file path location of the old hub to be converted. If left as NULL, will use the default path of the old code, which for unix systems was in a user's home directory "~\" and for windows users was in a user's AppData directory "~\AppData".
newcachepath	character(1) complete file path to the new location for the cache. If left as NULL, will use the new default path which utilizes the rappdir::user_cache_dir to deterime the appropriate caching location.
hubType	Either AnnotationHub or ExperimentHub. By default assumes AnnotationHub.
proxy	proxy connection allowing Internet access, usually through a restrictive firewall. Default: NULL.
max.downloads	numeric(1). The integer number of downloads allowed before triggering a warning. This is to help avoid accidental download of a large number of AnnotationHub members
force	logical(1). Force re-download of a resource rather than using a cached version.
verbose	logical(1). Print out status messages.

Value

character(1). File path of new cache location. If verbose also prints status messages for downloading files and any files that were not redownloaded.

Author(s)

Lori Shepherd

See Also

AnnotationHub, getAnnotationHubOption, getInfoOnIds

Examples

```
# To transition over from old default to new default location
## Not run: convertHub()
```

---

getAnnotationHubOption

Get and set options for default AnnotationHub behavior.

Description

These functions get or set options for creation of new 'AnnotationHub' instances.

Usage

```
getAnnotationHubOption(arg)
setAnnotationHubOption(arg, value)
```

Arguments

arg	The character(1) hub options to set. see 'Details' for current options.
value	The value to be assigned to the hub option.

Details

Supported options include:

"URL": character(1). The base URL of the annotation hub. Default: https://annotationhub.bioconductor.org

"CACHE": character(1). The location of the hub cache. Default: "AnnotationHub" in the user's directory established by `tools::R_user_dir()`.

"MAX_DOWNLOADS": numeric(1). The integer number of downloads allowed before triggering an error. This is to help avoid accidental download of a large number of AnnotationHub members.

"PROXY": `request` object returned by `httr::use_proxy()`. The `request` object describes a proxy connection allowing Internet access, usually through a restrictive firewall. Setting this option sends all AnnotationHub requests through the proxy. Default: NULL.

In `setAnnotationHubOption("PROXY", value)`, `value` can be one of NULL, a `request` object returned by `httr::use_proxy()`, or a well-formed URL as character(1). The URL can be completely specified by `http://username:password@proxy.dom.com:8080`; `username:password` and port (e.g. `:8080`) are optional.

"LOCAL": logical(1). TRUE/FALSE should the AnnotationHub create a hub consisting only of previously downloaded resourcesd. Default: FALSE.

"ASK": logical(1). TRUE/FALSE should the AnnotationHub ask if the hub location should be created. If FALSE, the default location will be used and created if it doesn't exist without asking. If TRUE will ask the user and if in a non interactive session utilize a temporary directoy for the caching. Default: TRUE.

Default values may also be determined by system and global R environment variables visible *before* the package is loaded. Use options or variables preceeded by "ANNOTATION_HUB_", e.g., `options(ANNOTATION_HUB_MAX_DOWNLOADS=10)` prior to package load sets the default number of downloads to 10.

Value

The requested or successfully set option.

Author(s)

Martin Morgan and Lori Shepherd

Examples

```
getAnnotationHubOption("URL")
## Not run:
setAnnotationHubOption("CACHE", "~/.myHub")

## End(Not run)
```

Hub-utils *Get information for selected ids*

Description

Gets information from the Hub database for the given selection of ids. The information collected is ah_id, fetch_id, title, rdataclass, availablitiy status, biocversion when added, date when added, date when removed, and file size.

Usage

```
getInfoOnIds(hub, ids)
```

Arguments

hub	Hub object.
ids	List of ids to get from database. Can be left unset to use all active ids in the hub. If given, it is either a numeric or character vector. See details section.

Value

data.frame of information for selected ids. The information collected is ah_id, fetch_id, title, rdataclass, availablitiy status, biocversion when added, date when added, date when removed, and file size.

details

If a hub object is passed into the function with no ids given, it will use all active ids associated with that hub object (`names(ah)`). It is recommended to only run this option if you are using a smaller subset Hub object. The ids argument can be specified as either a character vector or a numeric vector. If using a character vector, the function assumes the 'ah_ids' were used, and each entry takes the form similar to `c("AH2", "AH5012")`. If a numeric vector is specified, the function assume the 'fetch_ids' were used. The 'fetch_id' is the identifier that is used for the file name. For older versions of the cache these were the file names directly.

This function was designed as a helper function when converting between old versions of Hubs to the newer versions that utilize BiocFileCache. If files were not able to be redownloaded, one could put the ids into this function to get more information on them. Note: Some resources may appear available but could not be redownloaded. Most likely these files are rdataclass 'OrgDb'. 'OrgDb' are only valid for a given release cycle and are masked to any future release cycle. It is recommended to update to the current 'OrgDb' but if the old file was not able to be downloaded and still desired, one could download manually download using the fetch_id. Example if the file not able to be downloaded was "~./AnnotationHub/69303" then the fetch call is: "https://annotationhub.bioconductor.org/fetch/69303". While the convertHub function will not automatically download it is still possible to keep track in the cache by doing a manually addition. Although not recommended. In reality these file will not be updated so the original file could also still be used.

This function could also be a utility function to help determine any given resources download size.

Author(s)

Lori Shepherd

See Also

AnnotationHub, convertHub

Examples

```
## Not run:
getInfoOnIds(hub, c("AH2","AH5012"))
getInfoOnIds(hub, 69303)


## End(Not run)

# If using in conjunction with convertHub,
#
# File not downloaded options:
#

## Not run:
# 1. Use the original file. In reality the file is not going to be
  updated or should change. The original file does not need to be
  tracked and could now be referenced directly for usage. It will not be
  available in the Hub.

# 2. You could simply download the file for use
# The file will not change and not be updated so its static download not
# in the cache is fine

# You could type the following into a web browswer
"https://annotationhub.bioconductor.org/fetch/69303"

# or in R
httr::GET("https://annotationhub.bioconductor.org/fetch/69303",
  write_disk(<pathToSave/69303>, overwrite=FALSE))

# 3. To add to a hub cache (not recommended)
hub <- AnnotationHub()
bfc <- AnnotationHub:::.get_cache(hub)

# the hub creates the rname is in the format of 'ah_id : fetch_id'
bfcadd(bfc, fpath="https://annotationhub.bioconductor.org/fetch/69303",
rname="AH62557 : 69303")

## End(Not run)
```

utilities *Utility functions for discovering package-specific Hub resources.*

Description

List and load resources from ExperimentHub filtered by package name and optional search terms. Not Implemented for AnnotationHub.

Details

Currently `listResources` and `loadResources` are only meaningful for `ExperimentHub` objects.

utils *Utility function to list currently available dispatchClass.*

Description

When submitting resources to AnnotationHub or ExperimentHub a valid DispatchClass field must be specified in the inst/extdata/metadata.csv file for each resource. This list the currently available DispatchClass values and briefly how that class loads a resource. If your resource does not qualify for one of these methods contact Lori Shepherd <Lori.Shepherd@RoswellPark.org> to request a new DispatchClass be added

Author(s)

Lori Shepherd

Examples

```
DispatchClassList()
```

Index