Package 'rpx'

May 16, 2024

Type Package

Title R Interface to the ProteomeXchange Repository

Version 2.12.0

Author Laurent Gatto

Maintainer Laurent Gatto <laurent.gatto@uclouvain.be>

Description The rpx package implements an interface to proteomics data submitted to the ProteomeXchange consortium.

Depends R (>= 3.5.0), methods

Imports BiocFileCache, jsonlite, xml2, RCurl, curl, utils

Suggests Biostrings, BiocStyle, testthat, knitr, tibble, rmarkdown

License GPL-2

URL https://github.com/lgatto/rpx

BugReports https://github.com/lgatto/rpx/issues

VignetteBuilder knitr

biocViews ImmunoOncology, Proteomics, MassSpectrometry, DataImport, ThirdPartyClient

Encoding UTF-8

RoxygenNote 7.2.3

Roxygen list(markdown = TRUE)

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

git_branch RELEASE_3_19

git_last_commit c6b69fd

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-15

Contents

---

cache *Package cache*

Description

Function to access and manage the cache. `rxpCache()` returns the central `rpx` cache. `pxCachedrojects()` prints the names of the cached projects and invisibly returns the cache table.

Usage

```
rpxCache()

pxCachedProjects(cache = rpxCache(), rpxprefix = "^\\.rpx(2?)")
```

Arguments

`cache`	Object of class `BiocFileCache`.
`rpxprefix`	`character(1)` defining the resourne name prefix in `cache`. Default is `"^\\.rpx(2?)"` to match objects of class `PXDataset` and `PXDataset2`.

Details

The cache is an object of class `BiocFileCache`, and created with `BiocFileCache::BiocFileCache()`. It can be either the package-wide cache as defined by `rpxCache()` or an instaned provided by the user.

When projects are cached, they are given a resource name (`rname`) composed of the `.rpx` prefix followed by the ProteomeXchange identifier. For example, project `PXD000001` is named `.rpxPXD000001` (`.rpx2PXD000001` for the `PXDataset2` class) to avoid any conflicts with other resources that user-created resources.

Value

The `rpxCache()` function returns an instance of class `BiocFileCache`. `pxCachedProjects()` invisbly returns a `tibble` of cached ProteomeXchange projects.

Author(s)

Laurent Gatto

Examples

```
## Default rpx cache
rpxCache()

## Not run:

## Set up your own cache by providing a file or a directory to
## BiocFileCache::BiocFileCache()
my_cache <- BiocFileCache::BiocFileCache(tempfile())
my_cache
px <- PXDataset("PXD000001", cache = my_cache)
pxget(px, "erwinia_carotovora.fasta", cache = my_cache)


## List of cached projects
pxCachedProjects() ## default rpx cache
pxCachedProjects(my_cache)

## To delete project a project from the default cache, first find
## its resource id (rid) in the cache
px1_cache_info <- pxCacheInfo(px)
(rid <- px1_cache_info["rid"])

## Then remove it with BiocFileCache:: bfcremove()
BiocFileCache:::bfcremove(my_cache, rid)
pxCachedProjects(my_cache)

## End(Not run)
```

`fileTypes` *Infer file type*

Description

The `pxFileTypes()` function inferres mass spectrometry and proteomics file types based on a curated table of file types and associated patterns. This table can be accessed with `fileTypes()`. See the examples below for the content and format of the table.

The types of the files in a `PXDataset` object can be accessed with the `pxfiles(as.vector = FALSE)` function. See examples in the `pxfiles()` manual page.

`updatePxFileTypes()` updates the file types of a `PXDataset` instance using `pxFileTypes()`. This function also udpates the cached object unless `cache` is set to `NULL`. This function is useful to harmonise file types when the data in `fileTypes()` is updated.

The file types table is generated by `scripts/make_fileTypes.R`.

Usage

```
fileTypes()
```

```
pxFileTypes(fls, types = fileTypes())

updatePxFileTypes(object, cache = rpxCache())
```

Arguments

fls	character() of file names whose types need to be inferred based on their file extenstion.
types	data.frame of file types. Default is fileTypes().
object	Object of class PXDataset.
cache	Object of class BiocFileCache.

Value

A data.frame with the filenames and their inferred types.

Author(s)

Laurent Gatto with contributions via mastodon from Dr. Samuel Wein, Michael MacCoss, Marc Vaudel, Phil Wilmarth and Dave Tabb to identify several file types (see inst/make_file_types.R for details).

References

- McDonald, W. *et al.* 2004. "MS1, MS2, and SQT-Three Unified, Compact, and Easily Parsed File Formats for the Storage of Shotgun Proteomic Spectra and Identifications." Rapid Communications in Mass Spectrometry 18 (18):2162–68.
- Deutsch, Eric W. 2012. "File Formats Commonly Used in Mass Spectrometry Proteomics." Molecular & Cellular Proteomics 11 (12):1612–21.
- File formats in PRIDE Archive: https://www.ebi.ac.uk/pride/markdownpage/pridefileformats.

Examples

```
fileTypes()

pxFileTypes("foo")
pxFileTypes("foo.mzML")
pxFileTypes("foo.raw")
pxFileTypes("foo.txt")
pxFileTypes("foo.R")
pxFileTypes("foo.fasta")

pxFileTypes(c("foo", "foo.mzML", "foo.R", "foo.fasta"))
```

pxannounced *Return recent PX announcements*

Description

Queries the PX rss feed file for the latest PX dataset announcements.

Usage

```
pxannounced()
```

Value

A `data.frame` with announcements data set identifiers, publication dates and annoucement messages.

Author(s)

Laurent Gatto

Examples

```
pxannounced()
```

PXDataset1 *The PXDataset to find and download proteomics data*

Description

The `rpx` package provides the infrastructure to access, store and retrieve information for ProteomeXchange (PX) data sets. This can be achieved with `PXDataset` objects can be created with the `PXDataset()` constructor that takes the unique ProteomeXchange project identifier as input.

The `PXDataset` class is replaced by `PXDataset2` and is now deprecated. It will be defunct in the next release.

Usage

```
## S4 method for signature 'PXDataset'
pxid(object)

## S4 method for signature 'PXDataset'
pxurl(object)

## S4 method for signature 'PXDataset'
pxtax(object)
```

```
## S4 method for signature 'PXDataset'
pxref(object)

## S4 method for signature 'PXDataset'
pxfiles(object)

## S4 method for signature 'PXDataset'
pxget(object, list, cache = rpxCache())

## S4 method for signature 'PXDataset'
pxCacheInfo(object, cache = rpxCache())

PXDataset1(id, cache = rpxCache())
```

Arguments

object	An instance of class PXDataset, as created by PXDataset().
list	character(), numeric() or logical() defining the project files to be downloaded. This list of files can retrieved with pxfiles().
cache	Object of class BiocFileCache. Default is to use the central rpx cache returned by rpxCache(), but users can use their own cache. See rpxCache() for details.
id	character(1) containing a valid ProteomeXchange identifier.

Details

Since version 1.99.1, rpx uses the Bioconductor BiocFileCache package to automatically cache all downloaded ProteomeXchange files. When a file is downloaded for the first time, it is added to the cache. When already available, the file path to the cached file is directly returned. The central rpx package chache, object of class BiocFileCache, is returned by rpxCache(). Users can also provide their own cache object instead of using the default central cache to pxget().

Since 2.1.1, PXDataset instances are also cached using the same mechanism as project files. Each PXDataset instance also stored the project file names, the reference, taxonomy of the sample and the project URL (see slot cache) instead of accessing these every time they are needed to reduce remote access and reliance on a stable internet connection. As for files, the default cache is as returned by rpxCache(), but users can pass their own BiocFileCache objects.

For more details on how to manage the cache (for example if some files need to be deleted), please refer to the BiocFileCache package vignette and documentation. See also rpxCache() for additional details.

Value

The PXDataset() constructor returns a cached PXDataset object. It thus also modifies the cache used to projet caching, as defined by the cache argument.

Slots

`id character(1)` containing the dataset's unique ProteomeXchange identifier, as used to create the object.

`formatVersion character(1)` storing the version of the ProteomeXchange schema. Schema versions 1.0, 1.1 and 1.2 are supported (see https://code.google.com/p/proteomexchange/source/browse/schema/).

`cache list()` storing the available files (element `pxfiles`), the reference associated with the data set (`pxref`), the taxonomy of the sample (`pxtax`) and the datasets' ProteomeXchange URL (`pxurl`). These are returned by the respective accessors. It also stores the path to the cache it is stored in (element `cachepath`).

`Data XMLNode` storing the ProteomeXchange description as XML node tree.

Accessors

- `pxfiles(object)` returns the project file names.
- `pxget(object, list, cache)`: if the file(s) in `list` have never been requested, `pxget()` downloads the files from the ProteomeXchange repository, caches them in `cache` and returns their path. If the files have previously been downloaded and are available in `cache`, their path is directly returned.

  If `list` is missing, the file to be downloaded can be selected from a menu. If `list = "all"`, all files are downloaded. The file names, as returned by `pxfiles()` can also be used. Alternatively, a `logical` or `numeric` index can be used.

  The argument `cache` can be passed to define the path to the cache. The default cache is the packages' default as returned by `rpxCache()`.
- `pxtax(object)`: returns the taxonomic name of `object`.
- `pxurl(object)`: returns the base url on the ProteomeXchange server where the project files reside.
- `pxCacheInfo(object, cache)`: prints and invisibly returns object's caching information from cache(def
  The return value is a named vector of length two containing the resourne identifier and the cache location.

Author(s)

Laurent Gatto

References

Vizcaino J.A. et al. 'ProteomeXchange: globally co-ordinated proteomics data submission and dissemination', Nature Biotechnology 2014, 32, 223 – 226, doi:10.1038/nbt.2839.

Source repository for the ProteomeXchange project: https://code.google.com/p/proteomexchange/

PXDataset2 *New PXDataset (v2) to find and download proteomics data*

Description

The rpx package provides the infrastructure to access, store and retrieve information for ProteomeXchange (PX) data sets. This can be achieved with PXDataset2 objects can be created with the PXDataset2() constructor that takes the unique ProteomeXchange project identifier as input.

The new PXDataset2 class superseeds the previous and now deprecated PXDataset version.

Usage

```
PXDataset2(id, cache = rpxCache())

PXDataset(id, cache = rpxCache())

## S4 method for signature 'PXDataset2'
pxid(object)

## S4 method for signature 'PXDataset2'
pxurl(object)

## S4 method for signature 'PXDataset2'
pxtax(object)

## S4 method for signature 'PXDataset2'
pxref(object)

pxtitle(object)

pxinstruments(object)

pxSubmissionDate(object)

pxPublicationDate(object)

pxptms(object)

pxprotocols(object, which = c("project", "samples", "data"))

## S4 method for signature 'PXDataset2'
pxfiles(object, n = 10, as.vector = TRUE)

## S4 method for signature 'PXDataset2'
pxCacheInfo(object)
```

```
## S4 method for signature 'PXDataset2'
pxget(object, list, cache = rpxCache())
```

Arguments

id	character(1) containing a valid ProteomeXchange identifier.
cache	Object of class BiocFileCache. Default is to use the central rpx cache returned by rpxCache(), but users can use their own cache. See rpxCache() for details.
object	An instance of class PXDataset2.
which	character() with one or multiple protocols defined as "project", "samples" and "data".
n	integer(1) indicating the number of files to be printed.
as.vector	logical(1) defining if the output should be a vector of character with filenames (default) or a data.frame with additional details about each file.
list	character(), numeric() or logical() defining the project files to be downloaded. This list of files can retrieved with pxfiles().

Details

The rpx packages uses caching to store ProteomeXchange projects and project files. When creating an object with PXDataset2(), the cache is first queried for the projects identifier. If a unique hit is found, the project is retrieved and returned. If no matching project identifier is found, then the remote resource is accessed to first create the new PXDataset2() project, then cache it before returning it to the user. The same mechanism is applied when project files are requested.

Caching is supported by BiocFileCache package. The PXDataset2() constructor and the px_get() function can be passed a instance of class BiocFileCache that defines the cache. The default is to use the package-wide cache defined in rpxCache(). For more details on how to manage the cache (for example if some files need to be deleted), please refer to the BiocFileCache package vignette and documentation. See also rpxCache() for additional details.

Value

The PXDataset2() returns a cached PXDataset2 object. It thus also modifies the cache used to projet caching, as defined by the cache argument.

Slots

px_id character(1) containing the dataset's unique ProteomeXchange identifier, as used to create the object.

px_rid character(1) storing the cached resource name in the BiocFileCache instance stored in cachepath.

px_title character(1) with the project's title.

px_url 'character(1) with the project's URL.

px_doi character(1) with the project's DOI.

px_ref character containing the project's reference(s).

px_ref_doi character containing the project's reference DOIs.

px_pubmed character containing the project's reference PubMed identifier.

px_files data.frame containing information about the project files, including file names, URIs and types. The files are retrieved from the project's README.txt file.

px_tax charcter (typically of length 1) containing the taxonomy of the sample.

px_metadata list containing the project's metadata, as downloaded from the ProteomeXchange site. All slots but px_files are populated from this one.

cachepath character(1) storing the path to the cache the project object is stored in.

Accessors

- pxfiles(object, n = 10, as.vector = TRUE) by default, invisibly returns all the project file names. The function prints the first n files specifying whether they are local of remote (based on the cache the object is stored in). The printing can be ignored by wrapping the call in suppressMessages(). If as.vector is set to FALSE, it returns a data.frame with variables ID, NAME, URI, TYPE, MAPPINGS and PXID. Note that the variables and their content will depend on the rpx version that was installed when these objects were created and cached.
- pxget(object, list, cache): list is a vector defining the files to be downloaded. If list = "all", all files are downloaded. The file names, as returned by pxfiles() can also be used. Alternatively, a logical or numeric index can be used. If missing, the file to be downloaded can be selected from a menu.

  The argument cache can be passed to define the path to the cache. The default cache is the packages' default as returned by rpxCache().
- pxtax(object): returns the taxonomic name of object.
- pxurl(object): returns the base url on the ProteomeXchange server where the project files reside.
- pxCacheInfo(object, cache): prints and invisibly returns object's caching information from cache(def
  The return value is a named vector of length two containing the resourne identifier and the cache location.
- 'pxtitle(object): returns the project's title.
- pxref(object): returns the project's bibliographic reference(s).
- pxinstruments(object): returns the instrument(s) used to acquire the data.
- pxptms(object): returns the PTMs searched for in the experiment.
- pxprotocols(object, which): returns a list with the project description, sample processing and/or data processing protocols.

Author(s)

Laurent Gatto

References

Vizcaino J.A. et al. 'ProteomeXchange: globally co-ordinated proteomics data submission and dissemination', Nature Biotechnology 2014, 32, 223 – 226, doi:10.1038/nbt.2839.

Source repository for the ProteomeXchange project: https://code.google.com/p/proteomexchange/

Examples

```
px <- PXDataset("PXD000001")
px
pxtax(px)
pxurl(px)
pxref(px)
pxfiles(px)
pxfiles(px, as.vector = FALSE)

pxCacheInfo(px)

fas <- pxget(px, "erwinia_carotovora.fasta")
fas
library("Biostrings")
readAAStringSet(fas)
```

Index