

rpx: an *R* interface to the ProteomeXchange repository

Laurent Gatto

lg390@cam.ac.uk

Computational Proteomics Unit*

October 13, 2015

1 Introduction

The goal of the *rpx* package is to provide programmatic access to proteomics data from *R*, in particular to the ProteomeXchange¹ (PX) central repository (see <http://www.proteomexchange.org/> and <http://central.proteomexchange.org/>). Additional repositories are likely to be added in the future.

2 The *rpx* package

PXDataset objects

The central object that handles data access is the `PXDataset` class. Such an instance can be generated by passing a valid PX experiment identifier to the `PXDataset` constructor.

```
library("rpx")
id <- "PXD000001"
px <- PXDataset(id)
px

## Object of class "PXDataset"
## Id: PXD000001 with 10 files
## [1] 'F063721.dat' ... [10] 'erwinia_carotovora.fasta'
## Use 'pxfiles(.)' to see all files.
```

*<http://cpu.sysbiol.cam.ac.uk>

¹ 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.

Data and meta-data

Several attributes can be extracted from an PXDataset instance, as described below.

The experiment identifier, that was originally used to create the PXDataset instance can be extracted with the `pxid` method:

```
pxid(px)
## [1] "PXD000001"
```

The file transfer url where the data files can be accessed can be queried with the `pxurl` method:

```
pxurl(px)
## [1] "ftp://ftp.pride.ebi.ac.uk/pride/data/archive/2012/03/PXD000001"
```

The species the data has been generated the data can be obtain calling the `pntax` function:

```
pntax(px)
## [1] "Erwinia carotovora"
```

Relevant bibliographic references can be queried with the `pxref` method:

```
strwrap(pxref(px))
## [1] "Gatto L, Christoforou A. Using R and Bioconductor for proteomics data analysis."
## [2] "Biochim Biophys Acta. 2014 Jan;1844(1 Pt A):42-51. Review"
```

All files available for the PX experiment can be obtained with the `pxfiles` method:

```
pxfiles(px)
## [1] "F063721.dat"
## [2] "F063721.dat-mztab.txt"
## [3] "PRIDE_Exp_Complete_Ac_22134.xml.gz"
## [4] "PRIDE_Exp_mzData_Ac_22134.xml.gz"
## [5] "PXD000001_mztab.txt"
## [6] "TMT_Erwinia_1uLSike_Top10HCD_isol2_45stepped_60min_01-20141210.mzML"
## [7] "TMT_Erwinia_1uLSike_Top10HCD_isol2_45stepped_60min_01-20141210.mzXML"
## [8] "TMT_Erwinia_1uLSike_Top10HCD_isol2_45stepped_60min_01.mzXML"
## [9] "TMT_Erwinia_1uLSike_Top10HCD_isol2_45stepped_60min_01.raw"
## [10] "erwinia_carotovora.fasta"
```

The complete or partial data set can be downloaded with the `pxget` function. The function takes an instance of class PXDataset as first mandatory argument.

The next argument, `list`, specifies what files to download. If missing, a menu is printed and the user can select a file. If set to "all", all files of the experiment are downloaded in the working directory. Alternatively, numerics or logicals can also be used to subset the relevant files to be downloaded based on the `pxfiles(.)` output.

The last argument, `force`, can be set to `TRUE` to force the download of files that already exists in the working directory.

```
pxget(px, "erwinia_carotovora.fasta")
## Downloading 1 file
dir(pattern = "fasta")
## [1] "erwinia_carotovora.fasta"
```

By default, `pxget` will not download and overwrite a file if already available. The last argument of `pxget`, `force`, can be set to `TRUE` to force the download of files that already exists in the working directory.

```
(i <- grep("fasta", pxfiles(px)))
## [1] 10
pxget(px, i) ## same as above
## Downloading 1 file
## erwinia_carotovora.fasta already present.
```

Finally, a list of recent PX additions and updates can be obtained using the `pxannounced()` function:

```
pxannounced()
## 15 new ProteomeXchange announcements
```

	Data.Set	Publication.Data	Message
## 1	PXD002743	2015-10-13 15:34:03	New
## 2	PXD002928	2015-10-13 14:24:53	New
## 3	PXD002775	2015-10-13 10:28:36	New
## 4	PXD002802	2015-10-13 09:16:51	Updated information
## 5	PXD001066	2015-10-13 08:42:22	New
## 6	PXD002731	2015-10-13 08:02:32	New
## 7	PXD002656	2015-10-13 07:36:07	New
## 8	PXD003024	2015-10-13 07:19:54	Updated information
## 9	PXD003040	2015-10-12 13:37:52	New
## 10	PXD002028	2015-10-12 12:20:58	New
## 11	PXD000766	2015-10-12 11:49:14	New
## 12	PXD000498	2015-10-12 11:34:40	New
## 13	PXD001934	2015-10-12 10:00:22	New
## 14	PXD001514	2015-10-12 07:44:56	New
## 15	PXD003024	2015-10-09 14:42:24	New

A simple use-case

Below, we show how to automate the extraction of files of interest (fasta and mzTab files), download them and read them using appropriate Bioconductor infrastructure.

```
(mzt <- grep("F0.+mztab", pxfiles(px), value = TRUE))
## [1] "F063721.dat-mztab.txt"

(fas <- grep("fasta", pxfiles(px), value = TRUE))
## [1] "erwinia_carotovora.fasta"

pxget(px, c(mzt, fas))

## Downloading 2 files
## erwinia_carotovora.fasta already present.

library("Biostings")
readAAStringSet(fas)

## A AAStringSet instance of length 4499
##      width seq                                     names
## [1] 147 MADITLISGSTLGS AEYVAEHLAEELLE...EIDITQHQIPEDPAEEWLGSWVNLLK ECA0001 putative
## [2] 153 VAEIYQIDNLD RGILSALMENARTPYA...IQTIDEIQSTETLISLQNPIMRTIAP ECA0002 AsnC-fami
## [3] 330 MKKQYIEKQQQISFVKSFSSQLEQLL...LQLPHIGVQCQGVWPQPLRESVSGLL ECA0003 putative
## [4] 492 MITLESLEMLLSIDENELDDLVTLM...IFDHIWRFD TGLKSRLMRRWQH GKAY ECA0004 conserved
## [5] 499 MRQTAALAERISRLSHALEHGLYERQH...PSEWLAKIEASLQQVAEQIQQSEQQD ECA0005 conserved
## ...
## [4495] 634 MSDKIIHLTDDSFDTDLKADGAILVD...EWISVRRKVDPLRVFASDMARRLELL trx-rv3790 trx-rv
## [4496] 93 MTKMNNKARRTARELKHLGASIQTTSL...KPALYRELRDEFPMGYLG DYKDDDDK TimBlower TimBlowe
## [4497] 309 MFSNLSKRWAQRTLKSFYSTATGAAS...SIWKKFKWAGIKTRKFVFNPPKPRK sp|P07143|CY1_YEA
## [4498] 231 FPTDDDDKIVGGYTCAANSIPYQVSLN...AQKNKPGVYTKVCNYVNW IQQTIAAN sp|P00761|TRYP_PI
## [4499] 269 GVGSGSCNIDVVCPEGNGHRDVIRSVAA...LSDWLDAAGTGAQFIDGLDSTGTPPV sp|Q7M135|LYSC_LY

library("MSnbase")
(x <- readMzTabData(mzt, "PEP"))

## MSnSet (storageMode: lockedEnvironment)
## assayData: 1528 features, 0 samples
## element names: exprs
## protocolData: none
## phenoData: none
## featureData
## featureNames: DGVSVAR NVVL DK ... IDPILVTMDTLPELLSQALR (1528 total)
## fvarLabels: sequence accession ... peptide_abundance_sub.6. (20 total)
## fvarMetadata: labelDescription
## experimentData: use 'experimentData(object)'
## Annotation:
## - - - Processing information - - -
```

```
## MSnbase version: 1.18.0
head(exprs(x))
##
## DGVSVAR
## NVVLDK
## VEDALHATR
## LAGGVAVIK
## LIAEAMEK
## SFGAPTITK
head(fData(x)[, 1:2])
##           sequence accession
## DGVSVAR      DGVSVAR   ECA0625
## NVVLDK       NVVLDK   ECA0625
## VEDALHATR VEDALHATR   ECA0625
## LAGGVAVIK LAGGVAVIK   ECA0625
## LIAEAMEK   LIAEAMEK   ECA0625
## SFGAPTITK SFGAPTITK   ECA0625
```

3 Session information

- R version 3.2.2 (2015-08-14), x86_64-apple-darwin13.4.0
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: Biobase 2.30.0, BiocGenerics 0.16.0, BiocParallel 1.4.0, Biostrings 2.38.0, IRanges 2.4.0, MSnbase 1.18.0, ProtGenerics 1.2.0, Rcpp 0.12.1, S4Vectors 0.8.0, XVector 0.10.0, mzR 2.4.0, rpx 1.6.0
- Loaded via a namespace (and not attached): BiocInstaller 1.20.0, BiocStyle 1.8.0, MALDIquant 1.13, MASS 7.3-44, RCurl 1.95-4.7, XML 3.98-1.3, affy 1.48.0, affyio 1.40.0, bitops 1.0-6, codetools 0.2-14, colorspace 1.2-6, digest 0.6.8, doParallel 1.0.8, evaluate 0.8, foreach 1.4.3, formatR 1.2.1, futile.logger 1.4.1, futile.options 1.0.0, ggplot2 1.0.1, grid 3.2.2, gtable 0.1.2, highr 0.5.1, impute 1.44.0, iterators 1.0.8, knitr 1.11, lambda.r 1.1.7, lattice 0.20-33, limma 3.26.0, magrittr 1.5, munsell 0.4.2, mzID 1.8.0, pcaMethods 1.60.0, plyr 1.8.3, preprocessCore 1.32.0, proto 0.3-10, reshape2 1.4.1, scales 0.3.0, stringi 0.5-5, stringr 1.0.0, tools 3.2.2, vsn 3.38.0, zlibbioc 1.16.0