rpx: an *R* interface to the ProteomeXchange repository

Laurent Gatto
lg390@cam.ac.uk
Computational Proteomics Unit*

March 31, 2017

1 Introduction

The goal of the *rpx* package is to provide programmatic access to proteomics data from *R*, in particular to the ProteomeXchange[1] (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 12 files
##  [1] 'F063721.dat' ... [12] 'generated'
##  Use 'pxfiles(.)' to see all files.
```

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

[1] Vizcaíno 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 `pxtax` function:

```
pxtax(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] "README.txt"
##  [7] "TMT_Erwinia_1uLSike_Top10HCD_isol2_45stepped_60min_01-20141210.mzML"
##  [8] "TMT_Erwinia_1uLSike_Top10HCD_isol2_45stepped_60min_01-20141210.mzXML"
##  [9] "TMT_Erwinia_1uLSike_Top10HCD_isol2_45stepped_60min_01.mzXML"
## [10] "TMT_Erwinia_1uLSike_Top10HCD_isol2_45stepped_60min_01.raw"
## [11] "erwinia_carotovora.fasta"
## [12] "generated"
```

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] 11

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 annoucements

##     Data.Set    Publication.Data             Message
## 1  PXD004959 2017-03-31 13:21:40                 New
## 2  PXD004818 2017-03-31 11:27:32                 New
## 3  PXD004911 2017-03-31 08:21:34                 New
## 4  PXD005080 2017-03-31 08:17:41                 New
## 5  PXD004962 2017-03-31 08:16:03                 New
## 6  PXD005946 2017-03-31 07:31:04                 New
## 7  PXD005173 2017-03-31 07:20:21 Updated information
## 8  PXD005172 2017-03-31 07:19:24 Updated information
## 9  PXD005176 2017-03-31 07:18:34 Updated information
## 10 PXD005174 2017-03-31 07:16:49 Updated information
## 11 PXD005177 2017-03-31 07:15:52 Updated information
## 12 PXD005171 2017-03-31 07:14:09 Updated information
## 13 PXD005175 2017-03-31 07:13:11 Updated information
## 14 PXD005163 2017-03-31 07:12:20 Updated information
## 15 PXD005159 2017-03-31 07:09:04 Updated information
```

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. (Note that we read version 0.9 of the MzTab format below. For recent data, the `version` argument would be omitted.)

```
(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("Biostrings")
readAAStringSet(fas)
```

```
##   A AAStringSet instance of length 4499
##        width seq                                                 names
##    [1]   147 MADITLISGSTLGSAEYVAEHLAELLE...EIDITQHQIPEDPAEEWLGSWVNLLK ECA0001 putative
##    [2]   153 VAEIYQIDNLDRGILSALMENARTPYA...IQTIDEIQSTETLISLQNPIMRTIAP ECA0002 AsnC-fami
##    [3]   330 MKKQYIEKQQQISFVKSFFSSQLEQLL...LQLPHIGQVQCGVWPQPLRESVSGLL ECA0003 putative
##    [4]   492 MITLESLEMLLSIDENELLDDLVVTLM...IFDHIWRFDTGLKSRLMRRWQHGKAY ECA0004 conserved
##    [5]   499 MRQTAALAERISRLSHALEHGLYERQH...PSEWLAKIEASLQQVAEQIQQSEQQD ECA0005 conserved
##     ...   ... ...
## [4495]   634 MSDKIIHLTDDSFDTDVLKADGAILVD...EWISVRRKVDPLRVFASDMARRLELL trx-rv3790 trx-rv
## [4496]    93 MTKMNNKARRTARELKHLGASIQTTSL...KPALYRELRDEFPMGYLGDYKDDDDK TimBlower TimBlowe
## [4497]   309 MFSNLSKRWAQRTLSKSFYSTATGAAS...SIWVKKFKWAGIKTRKFVFNPPKPRK sp|P07143|CY1_YEA
## [4498]   231 FPTDDDDKIVGGYTCAANSIPYQVSLN...AQKNKPGVYTKVCNYVNWIQQTIAAN sp|P00761|TRYP_PI
## [4499]   269 GVSGSCNIDVVCPEGNGHRDVIRSVAA...LSDWLDAAGTGAQFIDGLDSTGTPPV sp|Q7M135|LYSC_LY
```

```
library("MSnbase")
(x <- readMzTabData(mzt, "PEP", version = "0.9"))
```

```
## Detected a metadata section
```

```
## Detected a peptide section
```

```
## MSnSet (storageMode: lockedEnvironment)
## assayData: 1528 features, 6 samples
##   element names: exprs
## protocolData: none
## phenoData
##   sampleNames: sub[1] sub[2] ... sub[6] (6 total)
##   varLabels: abundance
```

```
##   varMetadata: labelDescription
## featureData
##   featureNames: 1 2 ... 1528 (1528 total)
##   fvarLabels: sequence accession ... uri (14 total)
##   fvarMetadata: labelDescription
## experimentData: use 'experimentData(object)'
## Annotation:
## - - - Processing information - - -
## mzTab read: Fri Mar 31 19:08:31 2017
##  MSnbase version: 2.0.2
```

```
head(exprs(x))
```

```
##     sub[1]   sub[2]   sub[3]   sub[4]   sub[5]   sub[6]
## 1 10630132 11238708 12424917 10997763  9928972 10398534
## 2 11105690 12403253 13160903 12229367 11061660 10131218
## 3  1183431  1322371  1599088  1243715  1306602  1159064
## 4  5384958  5508454  6883086  6136023  5626680  5213771
## 5 18033537 17926487 21052620 19810368 17381162 17268329
## 6  9873585 10299931 11142071 10258214  9664315  9518271
```

```
head(fData(x)[, 1:2])
```

```
##    sequence accession
## 1   DGVSVAR   ECA0625
## 2    NVVLDK   ECA0625
## 3 VEDALHATR   ECA0625
## 4 LAGGVAVIK   ECA0625
## 5  LIAEAMEK   ECA0625
## 6 SFGAPTITK   ECA0625
```

3 Session information

- R version 3.3.3 (2017-03-06), `x86_64-apple-darwin13.4.0`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: Biobase 2.34.0, BiocGenerics 0.20.0, BiocParallel 1.8.1, Biostrings 2.42.1, IRanges 2.8.2, MSnbase 2.0.2, ProtGenerics 1.6.0, Rcpp 0.12.10, S4Vectors 0.12.2, XVector 0.14.1, mzR 2.8.1, rpx 1.10.2
- Loaded via a namespace (and not attached): BiocInstaller 1.24.0, BiocStyle 2.2.1, MALDIquant 1.16.1, RCurl 1.95-4.8, XML 3.98-1.6, affy 1.52.0, affyio 1.44.0, assertthat 0.1, bitops 1.0-6, codetools 0.2-15, colorspace 1.3-2, digest 0.6.12, doParallel 1.0.10, evaluate 0.10, foreach 1.4.3, ggplot2 2.2.1, grid 3.3.3, gtable 0.2.0, highr 0.6, impute 1.48.0, iterators 1.0.8, knitr 1.15.1, lattice 0.20-35, lazyeval 0.2.0, limma 3.30.13, magrittr 1.5, munsell 0.4.3, mzID 1.12.0, pcaMethods 1.66.0, plyr 1.8.4, preprocessCore 1.36.0, reshape2 1.4.2, scales 0.4.1,

stringi 1.1.3, stringr 1.2.0, tibble 1.2, tools 3.3.3, vsn 3.42.3, zlibbioc 1.20.0