A short tutorial on using *proteoQC* for mass spectrometry-based proteomics

Laurent Gatto and Bo Wen

October 13, 2015

1 Introduction

The *proteoQC* package provides a integrated pipeline for mass spectrometry-based proteomics quality control. It allows to generate a dynamic report starting from a set of `mgf` or `mz[X]ML` format peak list files, a protein database file and a description file of the experimental design. It performs an MS/MS search against the protein data base using the `X!Tandem` search engine [1] and the *rTANDEM* package [2]. The results are then summarised and compiled into an interactive html report using the *Nozzle.R1* package [3, 4].

2 Example data

We are going to use parts a dataset from the ProteomeXchange repository (http://www.proteomexchange.org/). We will use the *rpx* package to accessed and downloaded the data.

```
library("rpx")
px <- PXDataset("PXD000864")
px
```

```
## Object of class "PXDataset"
##  Id: PXD000864 with 218 files
##  [1] 'README.txt' ... [218] 'TTE2010.zip'
##  Use 'pxfiles(.)' to see all files.
```

There are a total of 218 files available from the ProteomeXchange repository, including raw data files (`raw`), result files (`-pride.xml.gz`), (compressed) peak list files (`.mgf.gz`) and, the fasta database file (`TTE2010.zip`) and one `README.txt` file.

```
head(pxfiles(px))
```

```
## [1] "README.txt"                    "TTE-55-1-01-1.dat-pride.xml.gz"
## [3] "TTE-55-1-01-1.mgf.gz"          "TTE-55-1-01-1.raw"
```

```
## [5] "TTE-55-1-01-2.dat-pride.xml.gz" "TTE-55-1-01-2.mgf.gz"
```

```
tail(pxfiles(px))
```

```
## [1] "TTE-75-1-12-2.mgf.gz"           "TTE-75-1-12-2.raw"
## [3] "TTE-75-1-12-3.dat-pride.xml.gz" "TTE-75-1-12-3.mgf.gz"
## [5] "TTE-75-1-12-3.raw"              "TTE2010.zip"
```

The files, in particular the `mgf` files that will be used in the rest of this document are named as follows `TTE-CC-B-FR-R` where `CC` takes values 55 or 75 and stands for the bacteria culture temperature in degree Celsius, `B` stands for the biological replicate (only 1 here), `FR` represents the fraction number from 01 to 12 and the leading `R` documents one of three technical replicates. (See also http://www.ebi.ac.uk/pride/archive/projects/PXD000864 for details). Here, we will make use of a limited number of samples below. First, we create a vector that stores the file names of interest.

```
mgfs <- grep("mgf", pxfiles(px), value = TRUE)
mgfs <- grep("-0[5-6]-[1|2]", mgfs, value=TRUE)
mgfs
```

```
## [1] "TTE-55-1-05-1.mgf.gz" "TTE-55-1-05-2.mgf.gz" "TTE-55-1-06-1.mgf.gz"
## [4] "TTE-55-1-06-2.mgf.gz" "TTE-75-1-05-1.mgf.gz" "TTE-75-1-05-2.mgf.gz"
## [7] "TTE-75-1-06-1.mgf.gz" "TTE-75-1-06-2.mgf.gz"
```

These files can be downloaded[1] using the `pxget`, providing the relevant data object (here `px`) and file names to be downloaded (see `?pxget` for details). We also need to uncompress (using `gunzip`) the files.

```
mgffiles <- pxget(px, mgfs)
library("R.utils")
mgffiles <- sapply(mgffiles, gunzip)
```

To reduce the file size of the demonstration data included for this package, we have trimmed the peak lists to 1/10 of the original number of spectra. All the details are provided in the vignette source.

Similarly, below we download the database file and unzip it.

```
fas <- pxget(px, "TTE2010.zip")
fas <- unzip(fas)
fas
```

3 Running proteoQC

[1] In the interest of time, the files are not downloaded when this vignette is compiled and the quality metrics are pre-computed (see details below). These following code chunks can nevertheless be executed to reproduce the complete pipeline.

3.1 Preparing the QC

The first step in the *proteoQC* pipeline is the definition of a design file, that provides the `mgf` file names, `sample` numbers, biological (`bioRep`) and technical (`techRep`) replicates and `fraction` numbers in a simple space-separated tabular format. We provide such a design file for our 8 files of interest.

```
design <- system.file("extdata/PXD000864-design.txt", package = "proteoQC")
design
```

```
## [1] "C:/biocbld/bbs-3.2-bioc/tmpdir/RtmpSQ5nnk/Rinstde467fc65cf/proteoQC/extdata/PXD0008
```

```
read.table(design, header = TRUE)
```

```
##                file sample bioRep techRep fraction
## 1 TTE-55-1-05-1.mgf     55      1       1        5
## 2 TTE-55-1-05-2.mgf     55      1       2        5
## 3 TTE-55-1-06-1.mgf     55      1       1        6
## 4 TTE-55-1-06-2.mgf     55      1       2        6
## 5 TTE-75-1-05-1.mgf     75      1       1        5
## 6 TTE-75-1-05-2.mgf     75      1       2        5
## 7 TTE-75-1-06-1.mgf     75      1       1        6
## 8 TTE-75-1-06-2.mgf     75      1       2        6
```

3.2 Running the QC

We need to load the *proteoQC* package and call the `msQCpipe` function, providing appropriate input parameters, in particular the `design` file, the `fasta` protein database, the `outdir` output directory that will contain the final quality report and various other peptide spectrum matching parameters that will be passed to the *rTANDEM* package. See `?msQCpipe` for a more in-depth description of all its arguments. Please note that if you take mz[X]ML format files as input, you must make sure that you have installed the rTANDEM that the version is greater than 1.5.1.

```
qcres <- msQCpipe(spectralist = design,
                  fasta = fas,
                  outdir = "./qc",
                  miss  = 0,
                  enzyme = 1, varmod = 2, fixmod = 1,
                  tol = 10, itol = 0.6, cpu = 2,
                  mode = "identification")
```

The `msQCpipe` function will run each mgf input file documented in the design file and search it against the fasta database using the `tandem` function from the *rTANDEM*. This might take some time depending on the number of files to be searched and the search parameters. The code chunk above takes about 3 minutes using 2 cores (`cpu = 2` above) on a modern laptop.

You can load the pre-computed quality control directory and result data that a shipped with *proteoQC* as shown below:

```
zpqc <- system.file("extdata/qc.zip", package = "proteoQC")
unzip(zpqc)
qcres <- loadmsQCres("./qc")
```

```
print(qcres)
```

```
## An msQC results:
##  Results stored in ./qc
##  Database: TTE2010.fasta
##  Run on Wed Apr 23 14:38:04 2014
##  Design with 8 samples:
##                 mgf sample bioRep techRep fraction
## 1 TTE-55-1-05-1.mgf     55      1       1        5
## 2 TTE-55-1-05-2.mgf     55      1       2        5
##   ...
##                 mgf sample bioRep techRep fraction
## 8 TTE-75-1-06-2.mgf     75      1       2        6
##
## You can now run reportHTML() to generate the QC report.
```

3.3 Generating the QC report

The final quality report can be generated with the `reportHTML`, passing the `qcres` object produced by the `msQCpipe` function above or the directory storing the QC data, as defined as parameter to the `msQCpipe`.

```
html <- reportHTML(qcres)
```

or

```
html <- reportHTML("./qc")
```

The report can then be opened by opening the `qc/qc_report.html` file in a web browser or directly with `browseURL(html)`.

4 The QC report

The dynamic html report is composed of 3 sections: an introduction, a methods and data section and a result part. The former are purely descriptive and summarise the design matrix and analysis parameters, as passed to `msQCpipe`.

The respective sections and sub-sections can be expanded and collapsed and each figure in the report can be zoomed in. While the dynamic html report is most useful for interactive inspection, it is also possible to print the complete report for archiving.

The results section provides tables and graphics that summarise

- Summaries of identification results for individual files as well as technical and biological replicates at the protein, peptide and spectrum levels.
- Summary overview charts that describe number of missed cleavages, peptide charge distributions, peptide length, precursor and fragment ion mass deviations, number of unique spectra/peptides per proteins and protein mass distributions for each sample.
- A contamination summary table generated using the common Repository of Adventitious Proteins (*cRAP*).
- Reproducibility summaries that compare fractions, replicates and samples, representing total number of spectra, number of identified spectra, number of peptides and proteins and overlap of peptides and proteins across replicates.
- Summary histograms of mass accuracies for fragment and precursor ions.
- A summary of the separation efficiency showing the effect of accumulating fractions for all samples.
- A summary of identification-independent QC metrics.

5 Some useful functions

5.1 Protein inference

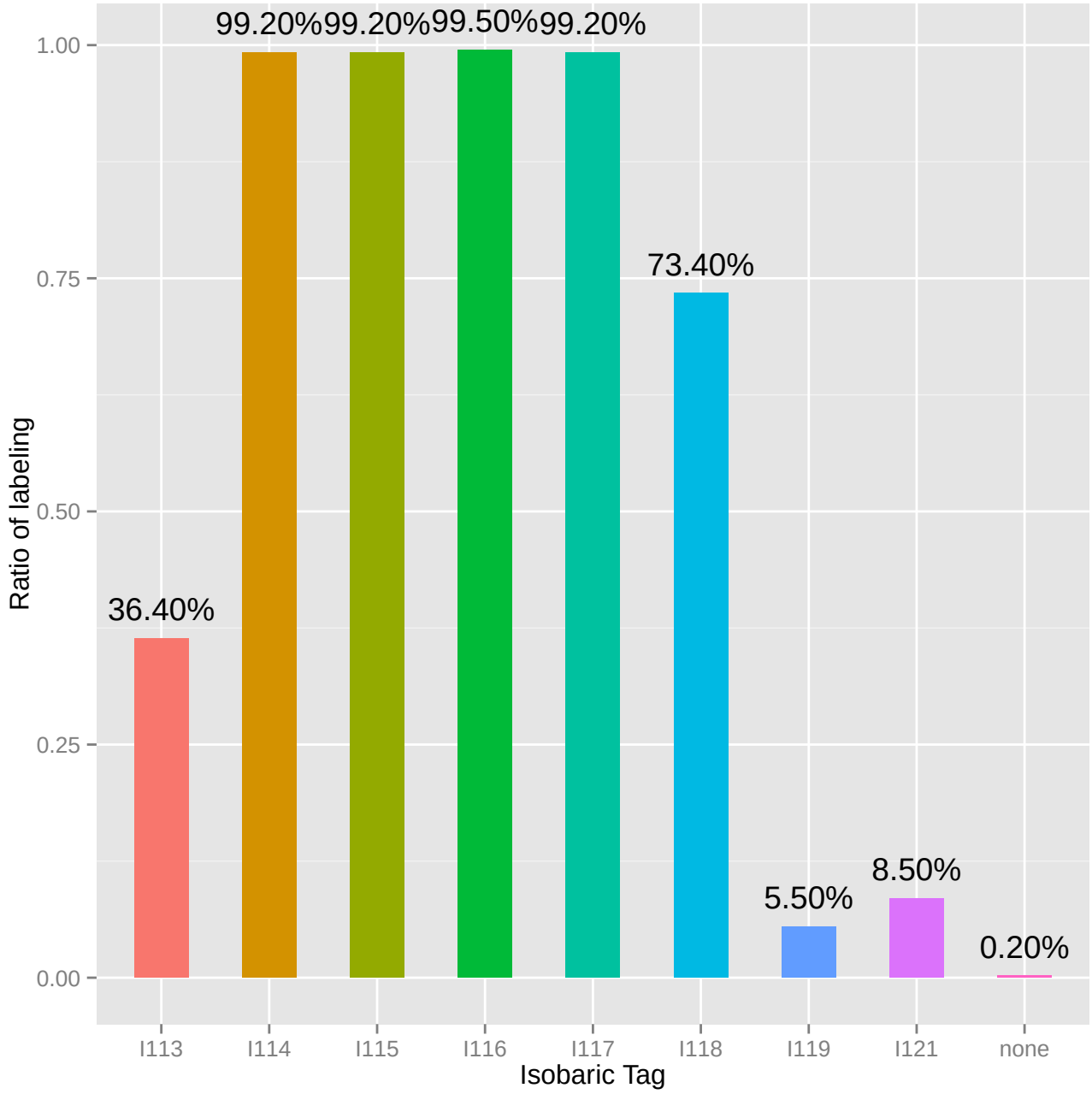
Protein inference from peptide identifications in shotgun proteomics is a very important task. We provide a function `proteinGroup` for this purpose. This function is based on the method used in our another package *sapFinder* [5]. You can use the function as below:

```
pep.zip <- system.file("extdata/pep.zip", package = "proteoQC")
unzip(pep.zip)
proteinGroup(file = "pep.txt", outfile = "pg.txt")
```

5.2 Isobaric tagging reagent labeling efficiency

The labeling efficiency of the isobaric tag reagents to peptides, such as iTRAQ and TMT, is a very important experiment quality metrics. We provide a function `labelRatio` to calculate this metrics. You can use the function as below:

```
mgf.zip <- system.file("extdata/mgf.zip", package = "proteoQC")
unzip(mgf.zip)
a <- labelRatio("test.mgf")
```



Session information

All software and respective versions used to produce this document are listed below.

- R version 3.2.2 Patched (2015-08-16 r69094), `x86_64-w64-mingw32`
- Locale: `LC_COLLATE=C`, `LC_CTYPE=English_United States.1252`, `LC_MONETARY=English_United States.1252`, `LC_NUMERIC=C`, `LC_TIME=English_United States.1252`

- Base packages: base, datasets, grDevices, graphics, grid, methods, parallel, stats, utils
- Other packages: Biobase 2.30.0, BiocGenerics 0.16.0, BiocParallel 1.4.0, MSnbase 1.18.0, ProtGenerics 1.2.0, R.methodsS3 1.7.0, R.oo 1.19.0, R.utils 2.1.0, Rcpp 0.12.1, VennDiagram 1.6.16, XML 3.98-1.3, futile.logger 1.4.1, mzR 2.4.0, proteoQC 1.6.0, rpx 1.6.0
- Loaded via a namespace (and not attached): BiocInstaller 1.20.0, BiocStyle 1.8.0, IRanges 2.4.0, MALDIquant 1.13, MASS 7.3-44, Nozzle.R1 1.1-1, RCurl 1.95-4.7, S4Vectors 0.8.0, ade4 1.7-2, 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.options 1.0.0, ggplot2 1.0.1, gtable 0.1.2, highr 0.5.1, impute 1.44.0, iterators 1.0.8, knitr 1.11, labeling 0.3, 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, rTANDEM 1.10.0, reshape2 1.4.1, scales 0.3.0, seqinr 3.1-3, stats4 3.2.2, stringi 0.5-5, stringr 1.0.0, tools 3.2.2, vsn 3.38.0, zlibbioc 1.16.0

References

[1] R Craig and R C Beavis. Tandem: matching proteins with tandem mass spectra. *Bioinformatics*, 20(9):1466–7, Jun 2004. doi:10.1093/bioinformatics/bth092.

[2] Frederic Fournier, Charles Joly Beauparlant, Rene Paradis, and Arnaud Droit. *rTANDEM: Encapsulates X!Tandem in R.*, 2013. R package version 1.2.0.

[3] Nils Gehlenborg. *Nozzle.R1: Nozzle Reports*, 2013. R package version 1.1-1. URL: http://CRAN.R-project.org/package=Nozzle.R1.

[4] N Gehlenborg, M S Noble, G Getz, L Chin, and P J Park. Nozzle: a report generation toolkit for data analysis pipelines. *Bioinformatics*, 29(8):1089–91, Apr 2013. doi:10.1093/bioinformatics/btt085.

[5] Bo Wen, Shaohang Xu, Gloria Sheynkman, Qiang Feng, Liang Lin, Quanhui Wang, Xun Xu, Jun Wang, and Siqi Liu. sapfinder: an r/bioconductor package for detection of variant peptides in shotgun proteomics experiments. *Bioinformatics*, page btu397, 2014.