

A Parser for `mzXML`, `mzData` and `mzML` files

Bernd Fischer*
Steffen Neumann†
Laurent Gatto‡

October 2, 2012

Contents

1	Introduction	1
2	Mass spectrometry raw data	2
2.1	Spectral data access	2
2.2	Metadata access	2
3	Example	2
4	Future plans	5
5	Session information	6

1 Introduction

The `mzR` package aims at providing a common interface to several mass spectrometry data formats, namely `mzData` (Orchard et al., 2007), `mzXML` (Pedrioli et al., 2004) and the latest `mzML` (Martens et al., 2010), somewhat similar to the Bioconductor package `affyio` for `affymetrix` raw data. No processing is done in `mzR`, which is left to packages such as `XCMS`¹ or `MSnbase`².

Most importantly, access to the data should be fast and memory efficient. This is made possible by allowing random file access, i.e. retrieving specific data of interest without having to sequentially browser the full content.

The actual work of reading and parsing the data files is handled by the included C/C++ libraries or “backends”. The RAMP parser, written at the Institute for Systems Biology (ISB) is a fast and lightweight parser in pure C. Later, it gained support for the `mzData` format.

*bernd.fischer@embl.de

†sneumann@ipb-halle.de

‡lg390@cam.ac.uk

¹<http://www.bioconductor.org/packages/release/bioc/html/xcms.html>

²<http://www.bioconductor.org/packages/release/bioc/html/MSnbase.html>

The C++ reference implementation for the **mzML** is the **proteowizard** library (Kessner et al., 2008) (**pwiz** in short), which in turn makes use of the **boost C++** (<http://www.boost.org/>) library. **RAMP** is able to access **mzML** files by calling **pwiz** methods.

The **mzR** package is in essence a collection of wrappers to the C++ code, and benefits from the C++ interface provided through the **Rcpp** package (Edelbuettel and François, 2011).

2 Mass spectrometry raw data

All the mass spectrometry file formats are organized similarly, where a set of metadata nodes about the run is followed by a list of spectra with the actual masses and intensities. In addition, each of these spectra has its own set of metadata, such as the retention time and acquisition parameters.

2.1 Spectral data access

Access to the spectral data is done via the **peaks** function. The return value is a list of two-column mass-to-charge and intensity matrices or a single matrix if one spectrum is queried.

2.2 Metadata access

Run metadata is available via several functions such as **instrumentInfo()** or **runInfo()**. The individual fields can be accessed via e.g. **detector()** etc.

Spectrum metadata is available via **header()**, which will return a list (for single scans) or a dataframe with information such as the **basePeakMZ**, **peaksCount**, ... or, for higher-order MS the **msLevel** and precursor information.

The availability of this metadata can not always be guaranteed, and depends on the MS software which converted the data.

3 Example

A short example sequence to read data from a mass spectrometer. First open the file.

```
> library(mzR)
> library(msdata)
> mzxml <- system.file("threonine/threonine_i2_e35_pH_tree.mzXML",
+                       package = "msdata")
> aa <- openMSfile(mzxml)
```

We can obtain different kind of header information.

```
> runInfo(aa)
```

```

$scanCount
[1] 55

$lowMz
[1] 50.0036

$highMz
[1] 298.673

$dStartTime
[1] 0.3485

$dEndTime
[1] 390.027

$msLevels
[1] 1 2 3 4

> instrumentInfo(aa)

$manufacturer
[1] "Thermo Scientific"

$model
[1] "LTQ Orbitrap"

$ionisation
[1] "ESI"

$analyzer
[1] "FTMS"

$detector
[1] "unknown"

> header(aa,1)

$seqNum
[1] 1

$acquisitionNum
[1] 1

$msLevel
[1] 1

$peaksCount
[1] 684

$totIonCurrent
[1] 341427000

```

```

$retentionTime
[1] 0.3485

$basePeakMZ
[1] 120.066

$basePeakIntensity
[1] 211860000

$collisionEnergy
[1] 0

$ionisationEnergy
[1] 0

$lowMZ
[1] 50.3254

$highMZ
[1] 298.673

$precursorScanNum
[1] 0

$precursorMZ
[1] 0

$precursorCharge
[1] 0

$precursorIntensity
[1] 0

$mergedScan
[1] 0

$mergedResultScanNum
[1] 0

$mergedResultStartScanNum
[1] 0

$mergedResultEndScanNum
[1] 0

```

Read a single spectrum from the file.

```

> pl <- peaks(aa,10)
> peaksCount(aa,10)

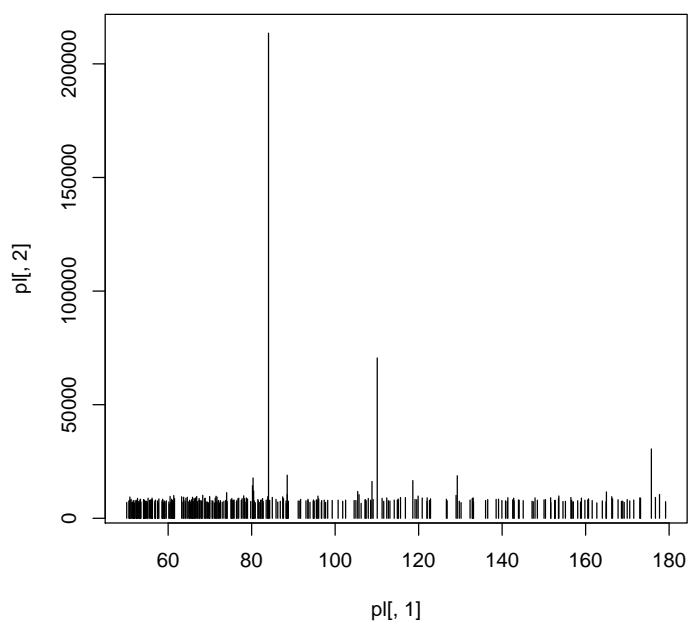
```

```
[1] 317

> head(pl)

      [,1]      [,2]
[1,] 50.08176 6984.858
[2,] 50.62267 7719.419
[3,] 50.70530 7185.290
[4,] 50.73298 7509.140
[5,] 50.83848 9366.624
[6,] 50.88303 8012.808

> plot(pl[,1], pl[,2], type="h", lwd=1)
```



One should always close the file when not needed any more. This will release the memory of cached content.

```
> close(aa)
```

4 Future plans

Right on the heels of the initial RAMP backend release, supporting little meta-data beyond e.g. the instrument model, is support to the full pwiz functionality to access the full metadata stored in an **mzML** files, or the chromatograms (which store e.g. MRM data). Other file formats supported by pwiz, such as **mzIdentML** for protein identification results are also possible in the future.

5 Session information

- R version 2.15.1 (2012-06-22), i386-pc-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, methods, stats, utils
- Other packages: Rcpp 0.9.14, msdata 0.1.10, mzR 1.4.0
- Loaded via a namespace (and not attached): Biobase 2.18.0, BiocGenerics 0.4.0, codetools 0.2-8, tools 2.15.1

References

- Dirk Eddelbuettel and Romain François. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18, 2011. URL <http://www.jstatsoft.org/v40/i08/>.
- Darren Kessner, Matt Chambers, Robert Burke, David Agus, and Parag Mallick. Proteowizard: open source software for rapid proteomics tools development. *Bioinformatics*, 24(21):2534–6, 2008. doi: 10.1093/bioinformatics/btn323.
- Lennart Martens, Matthew Chambers, Marc Sturm, Darren Kessner, Fredrik Levander, Jim Shofstahl, Wilfred H Tang, Andreas Rompp, Steffen Neumann, Angel D Pizarro, Luisa Montecchi-Palazzi, Natalie Tasman, Mike Coleman, Florian Reisinger, Puneet Souda, Henning Hermjakob, Pierre-Alain Binz, and Eric W Deutsch. mzml - a community standard for mass spectrometry data. *Molecular and Cellular Proteomics : MCP*, 2010. doi: 10.1074/mcp.R110.000133.
- Sandra Orchard, Luisa Montecchi-Palazzi, Eric W Deutsch, Pierre-Alain Binz, Andrew R Jones, Norman Paton, Angel Pizarro, David M Creasy, JÃrÃme Wojcik, and Henning Hermjakob. Five years of progress in the standardization of proteomics data 4th annual spring workshop of the hupo-proteomics standards initiative april 23-25, 2007 ecole nationale supÃrieure (ens), lyon, france. *Proteomics*, 7(19):3436–40, 2007. doi: 10.1002/pmic.200700658.
- Patrick G A Pedrioli, Jimmy K Eng, Robert Hubley, Mathijs Vogelzang, Eric W Deutsch, Brian Raught, Brian Pratt, Erik Nilsson, Ruth H Angeletti, Rolf Apweiler, Kei Cheung, Catherine E Costello, Henning Hermjakob, Sequin Huang, Randall K Julian, Eugene Kapp, Mark E McComb, Stephen G Oliver, Gilbert Omenn, Norman W Paton, Richard Simpson, Richard Smith, Chris F Taylor, Weimin Zhu, and Ruedi Aebersold. A common open representation of mass spectrometry data and its application to proteomics research. *Nat. Biotechnol.*, 22(11):1459–66, 2004. doi: 10.1038/nbt1031.