MSnbase: labelled MS2 data pre-processing, visualisation and quantification.

Laurent Gatto
lg390@cam.ac.uk
Cambridge Center for Proteomics
Kathryn S. Lilley Group
University of Cambridge

September 20, 2011

Abstract

This vignette describes the functionality implemented in the *MSnbase* package. *MSnbase* aims at (1) facilitating the upload, processing, visualisation and quantification of mass spectrometry data into the `R` environment (R Development Core Team, 2011) by providing specific data classes and methods and (2) enabling the utilisation of throughput-high data analysis pipelines provided by the Bioconductor (Gentleman et al., 2004) project.

Keywords: Mass Spectrometry (MS), proteomics, infrastructure, bioinformatics, quantitative.

Contents

Foreword

MSnbase is in an early development (see section 8 for details about packages and version used in this vignette). Although main data structures have been thought out and are meant to be compatible with other existing mature infrastructure in the Bioconductor project, changes may occur in the future. Current functionality will evolve and new one will be added. Although at an early stage, this package is released in the hope that it may foster new developments in proteomics data analysis within `R` by providing a common infrastructure. Several package developers working with mass spectrometry and proteomics data met at the Bioconductor Developer Meeting Europe[1] held in Heidelberg in November 2010, and agreed to combine efforts. This library is one attempt to do so.

You are welcome to contact me for questions, bugs, typos or suggestions about *MSnbase*. If you wish to reach a broader audience for general questions about proteomics analysis using `R`, you may want to use the Bioconductor mailing list[2].

1 Introduction

MSnbase aims are providing a reproducible research framework to proteomics data analysis. It should allow researcher to easily mine mass spectrometry data, explore the data and its statistical properties and visually display these.

MSnbase also aims at being compatible with the infrastructure implemented in Bioconductor, in particular *Biobase*. As such, classes developed specifically for proteomics mass spectrometry data are based on the `eSet` and `Expression` classes. The main goal is to assure seamless compatibility with existing meta data structure, accessor methods and normalisation techniques.

This vignette illustrates *MSnbase* utility using a dummy data sets provided with the package without describing the underlying data structures. More details can be found in the package, classes, method and function documentations. A description of the classes is provided in the `MSnbase-development` vignette.

Speed and memory requirements Raw mass spectrometry file are generally several hundreds of MB large and most of this is used for binary raw spectrum data. As such, data containers can easily grow very large and thus require large amounts of RAM. This requirement may be tackled in the future by avoiding to load the raw data into memory and using random access to the content of `mzXML`/`mzML` data files on demand. When focusing on reporter ion quantitation, a direct solution for this is to trim the spectra using the `trimMz`

[1] http://bioconductor.org/help/course-materials/2010/HeidelbergNovember2010/
[2] https://stat.ethz.ch/mailman/listinfo/bioconductor

method to select the area of interest and thus substantially reduce the size of the `Spectrum` objects. This is illustrated on page **??** of the `MSnbase-demo` vignette.

The independent handling of spectra is ideally suited for parallel processing. This will be added soon.

2 Importing raw data

Raw data is contained in `MSnExp` objects, that stores all the spectra of an experiment, as defined by one or multiple raw data files.

MSnbase imports raw MS data stored in `mzXML` format (Pedrioli et al., 2004) with the function `readMzXMLData` function. More formats, notably `mzML` (Martens et al., 2010), will be added at a later stage. Either MS1 or MS2 spectra can be loaded at a time by setting the `msLevel` parameter accordingly.

```
> library("MSnbase")
> filename <- dir(system.file(package = "MSnbase", dir =
"extdata"),
+     full.name = TRUE, pattern = "mzXML$")
> print(filename)
```

```
[1] "/tmp/Rtmpx83liC/Rinst75000477/MSnbase/extdata/dummyiTRAQ.mzXML"
```

```
> experiment <- readMzXMLData(filename, msLevel = 2, verbose =
FALSE)
> experiment
```

```
Object of class "MSnExp"
 Object size in memory: 2.13 Mb
- - - Spectra data - - -
 MSn level(s): 2
 Number of MS1 acquisitions: 3
 Number of MS2 scans: 54
 Number of precursor ions: 54
 7 unique MZs
 Precursor MZ's: 425.78 - 630.33
 MSn M/Z range: 50 2000.04
 MSn retention times: 37:40 - 49:16 minutes
- - - Processing information - - -
Data loaded: Tue Sep 20 11:20:25 2011
 MSnbase version: 1.0.7
- - - Meta data  - - -
phenoData: none
Loaded from:
  dummyiTRAQ.mzXML
```

```
protocolData: none
featureData
  featureNames: X1 X10 ... X9 (54 total)
  fvarLabels: spectum
  fvarMetadata: labelDescription
experimentData: use 'experimentData(object)'
```

As illustrated above, showing the experiment textually displays it's content:

- Information about the raw data, i.e. the spectra.
- Specific information about the experiment processing[3] and package version. This slot can be accessed with the `processingData` method.
- Other meta data, including experimental phenotype, file name(s) used to import the data, protocol data, information about features (individual spectra here) and experiment data. Most of these are implemented as in the `eSet` class and are described in more details in their respective manual pages. See `?MSnExp` and references therein for additional background information.

  The experiment meta data associated with an `MSnExp` experiment is of class `MIAPE`. It stores general information about the experiment as well as MIAPE (Minimum Information About a Proteomics Experiment, (Taylor et al., 2007, 2008)) information. This meta-data can be accessed with the `experimentData` method. When available, a summary of MIAPE-MS data can be printed with the `msInfo` method. See `?MIAPE` for more details.

3 Spectra objects

The raw data is composed of the 54 MS spectra. The spectra are named individually (X1, X10, X11, X12, X13, X14, ...) and stored in a `environment`. They can be accessed individually with `experiment[["X10"]]` or as a list with `spectra(experiment)`. As we have loaded our experiment specifying `msLevel=2`, the spectra will all be of level 2 (or higher, if available).

```
> sp <- experiment[["X43"]]
> sp


Object of class "Spectrum2"
 Precursor: 595.3597
 Retention time: 49:12
 Charge: 2
 MSn level: 2
```

[3] this part will be automatically updated when the object is modified with it's *ad hoc* methods, as illustrated later

```
 Peaks count: 9157
 Total ion count: 8809
```

Attributes of individual spectra or of all spectra of an experiment can be accessed with their respective methods: `precursorCharge` for the precursor charge, `rtime` for the retention time, `mz` for the MZ values, `intensity` for the inensities, ... see the `Spectrum`, `Spectrum1` and `Spectrum2` manuals for more details.

```
> peaksCount(sp)

[1] 9157

> head(peaksCount(experiment))

  X1  X10  X11  X12  X13  X14
1757 2631 2316 1365 1987 1938

> rtime(sp)

[1] 2952.03

> head(rtime(experiment))

     X1     X10     X11     X12     X13     X14
2259.83 2659.53 2659.87 2660.22 2660.56 2660.90
```

4 Reporter ions

Reporter ions are defined with the `ReporterIons` class. Specific peaks of interest are defined by a MZ value, a with around the expected MZ and a name (and optionally a colour for plotting, see section 5). `ReporterIons` instances are required to quantify reporter peaks in `MSnExp` experiments. Instances for the most commonly used isobaric tags like iTRAQ 4-plex and 8-plex and TMT tags are already defined in *MSnbase*. See `?ReporterIons` for details about how to generate new `ReporterIons` objects.

```
> iTRAQ4

Object of class "ReporterIons"
iTRAQ4: '4-plex iTRAQ' with 4 reporter ions
 - 114.1 +/- 0.05 (red)
 - 115.1 +/- 0.05 (green)
 - 116.1 +/- 0.05 (blue)
 - 117.1 +/- 0.05 (yellow)
```

5 Plotting raw data

Spectra can be plotted individually or as part of (subsetted) experiments with the `plot` method. Full spectra can be plotted (using `full=TRUE`), specific reporter ions of interest (by specifing with reporters with `reporters=iTRAQ4` for instance) or both (see figure 1).

```
> plot(sp, reporters = iTRAQ4, full = TRUE)
```

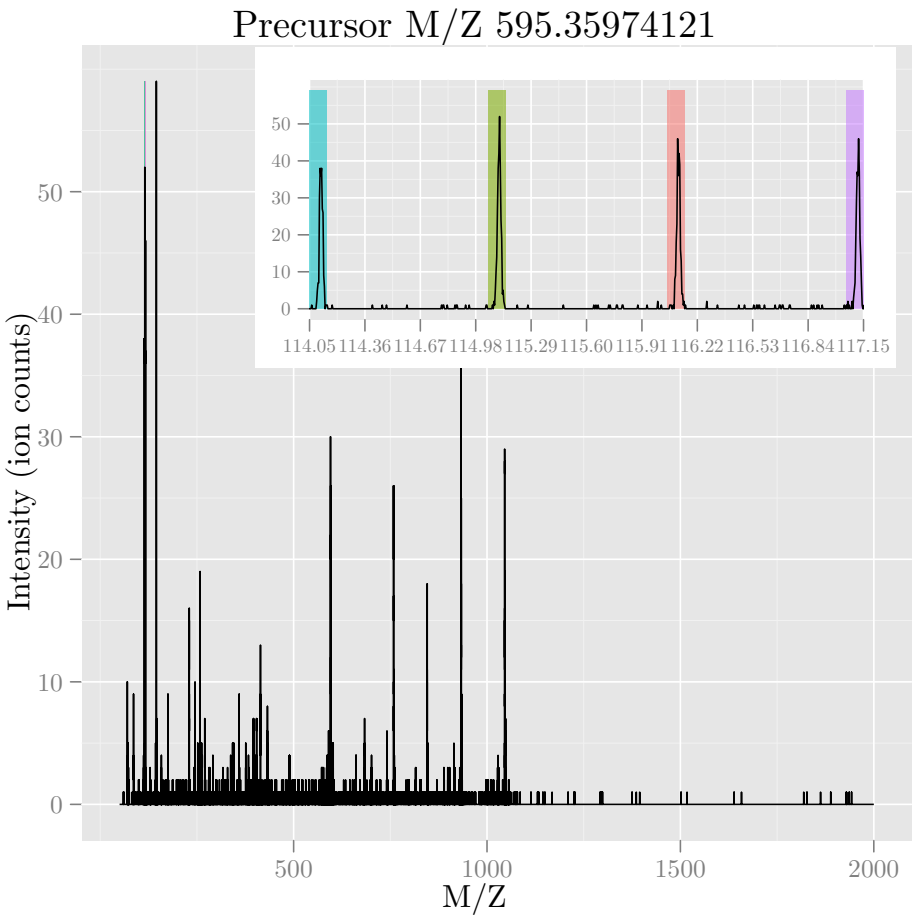

Figure 1: Raw MS2 spectrum with details about reporter ions.

It is also possible to plot all spectra of an experiment (figure 2). Lets start by extracting all spectra that have the same precursor MZ value than `sp` into a separate experiment, using the `extractPrecSpectra` method. This method takes an `MSnExp` experiment and a precursor MZ value as parameters.

```
> exp2 <- extractPrecSpectra(experiment, precursorMz(sp))
> exp2


Object of class "MSnExp"
 Object size in memory: 0.36 Mb
- - - Spectra data - - -
 MSn level(s): 2
 Number of MS1 acquisitions: 1
 Number of MS2 scans: 3
 Number of precursor ions: 3
 1 unique MZs
 Precursor MZ's: 595.36 - 595.36
 MSn M/Z range: 50 2000.04
 MSn retention times: 49:10 - 49:12 minutes
- - - Processing information - - -
Data loaded: Tue Sep 20 11:20:25 2011
1(3) precursors (spectra) extracted: Tue Sep 20 11:20:40 2011
 MSnbase version: 1.0.7
- - - Meta data  - - -
phenoData: none
Loaded from:
  dummyiTRAQ.mzXML
protocolData: none
featureData
  featureNames: X37 X40 X43
  fvarLabels: spectum
  fvarMetadata: labelDescription
experimentData: use 'experimentData(object)'
```

We see that 3 spectra have the same MZ value. These can be visulaised together by plotting the `MSnExp` object, as illustrated on figure 2.

6 Data processing

There are several methods implemented to perform basic data manipulation. Low intensity peaks can be set to 0 with the `removePeaks` method from spectra or whole experiments. The intensity threshold below which peaks are removed is defined by the `t` parameter. `t` can be specified directly as a numeric. The default value is the character `"min"`, that will remove all peaks equal to the lowest non null intensity in any spectrum. We observe the effect of the `removePeaks` method by comparing total ion count (i.e. the total intensity in a spectrum) with the `tic` method before (object `sp`) and after (object `experiment`). Figure 3 illustrated spectrum `X43` after low intensity peaks removal.

```
> plot(exp2, reporters = iTRAQ4, full = FALSE)
```

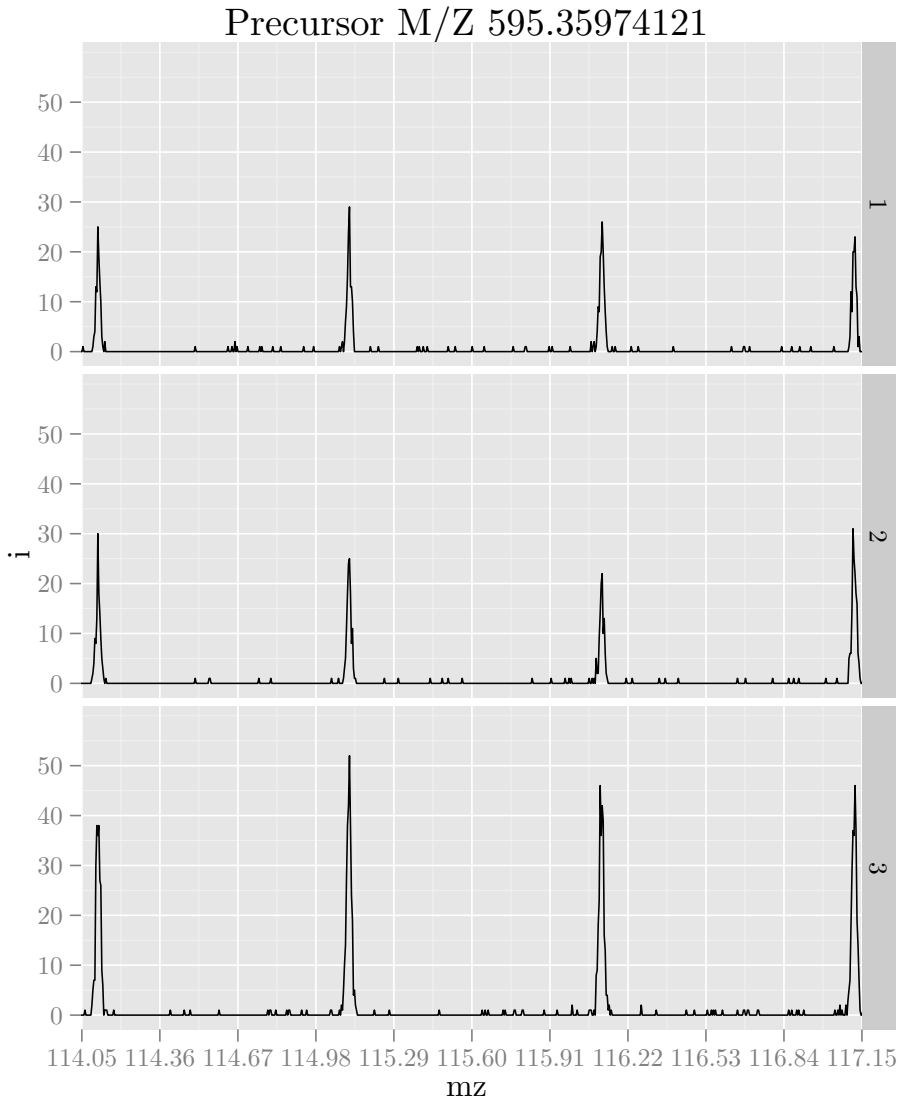

Figure 2: Experiment-wide raw MS2 spectra.

```
> experiment <- removePeaks(experiment, t = 3, verbose = FALSE)
> tic(sp)
```

```
[1] 8809
```

```
> tic(experiment[["X43"]])
```

```
[1] 4897
```

```
> plot(experiment[["X43"]], reporters = iTRAQ4, full = TRUE)
```

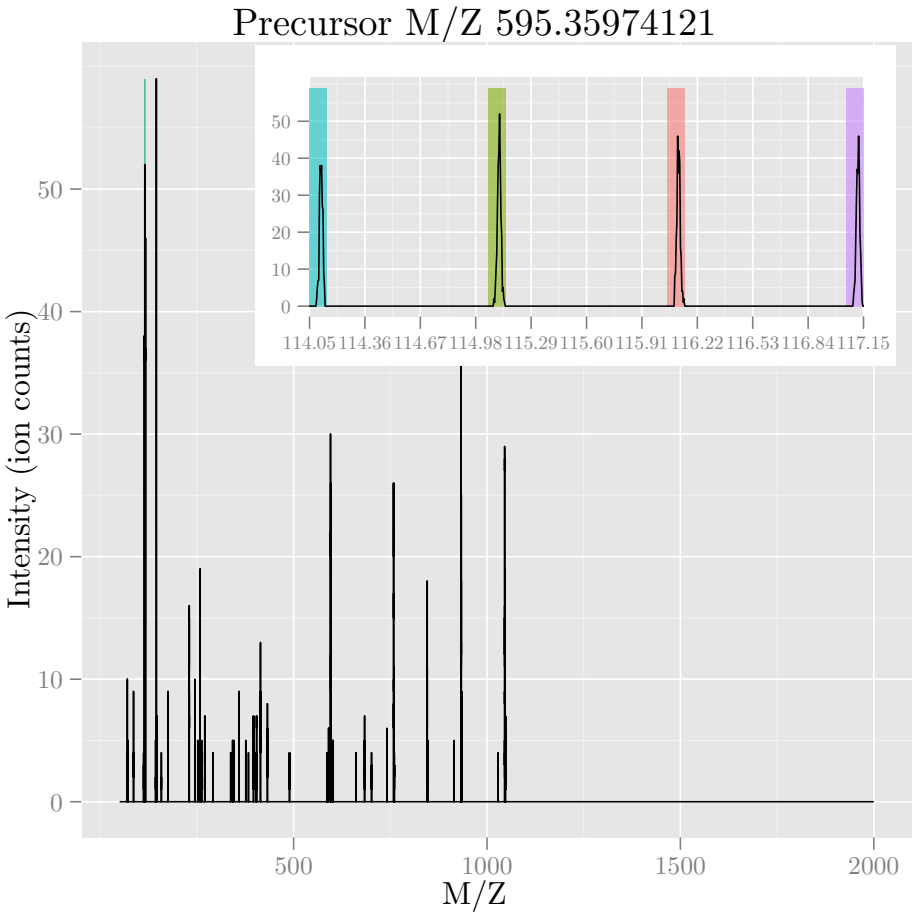

Figure 3: Same spectrum as on figure 1, after setting peaks <= 3 to 0.

Unlike the name might suggest, the `removePeaks` method does not actually remove peaks from the spectrum; they are set to 0. This can be checked using the `peaksCount` method, that returns the number of peaks (including 0 intensity

peaks) in a spectrum. To effectively remove 0 intensity peaks from spectra, and reduce the size of the data set, one can use the `clean` method.

```
> peaksCount(sp)
```

```
[1] 9157
```

```
> peaksCount(experiment[["X43"]])
```

```
[1] 9157
```

```
> experiment <- clean(experiment, verbose = FALSE)
> peaksCount(experiment[["X43"]])
```

```
[1] 793
```

Another useful manipulation method is `trimMz`, that takes as parameters and `MSnExp` (or a `Spectrum`) and a numeric `mzlim`. MZ values smaller then `min(mzlim)` or greater then `max(mzmax)` are discarded. This method is particularely useful when one wants to concentrate on a specific MZ range, as for reporter ions quantification, and generally results in substantial reduction of data size. Compare the size of the full trimmed experiment to the original 2.13 Mb.

```
> range(mz(sp))
```

```
[1]   49.99825 2000.03552
```

```
> experiment <- trimMz(experiment, mzlim = c(112, 120))
> range(mz(experiment[["X43"]]))
```

```
[1] 112.0877 117.1464
```

```
> experiment
```

```
Object of class "MSnExp"
 Object size in memory: 0.27 Mb
- - - Spectra data - - -
 MSn level(s): 2
 Number of MS1 acquisitions: 3
 Number of MS2 scans: 54
 Number of precursor ions: 54
 7 unique MZs
 Precursor MZ's: 425.78 - 630.33
```

```
 MSn M/Z range: 112.08 117.15
 MSn retention times: 37:40 - 49:16 minutes
- - - Processing information - - -
Data loaded: Tue Sep 20 11:20:25 2011
Curves <= 3 set to '0': Tue Sep 20 11:20:57 2011
Spectra cleaned: Tue Sep 20 11:21:04 2011
MZ trimmed [112..120]
 MSnbase version: 1.0.7
- - - Meta data  - - -
phenoData: none
Loaded from:
  dummyiTRAQ.mzXML
protocolData: none
featureData
  featureNames: X1 X10 ... X9 (54 total)
  fvarLabels: spectum
  fvarMetadata: labelDescription
experimentData: use 'experimentData(object)'
```

As can be seen above, all processing performed on the experiment is recorded and displayed as integral part of the experiment object.

7 Quantitation

Quantitation is performed on specific peaks in the spectra, that are specified with an `ReporterIon` object. A specific peak is defined by it's expected `mz` value and is expected to be found within `mz` $\pm$ `width`. If the peak reaches outside, a warning will be issued. If no data is found, `NA` is returned.

```
> mz(iTRAQ4)
```

```
[1] 114.1 115.1 116.1 117.1
```

```
> width(iTRAQ4)
```

```
[1] 0.05
```

The `quantify` method takes an experiment, a character describing the quantification `method` and the `reporters` to be quantified as parameters. Additionally, a progress bar can be dispaying when setting the `verbose` paramter to `TRUE`. Three quantification methods are implemented: `trapezoidation` returns the area under the peak of interest, `max` returns the apex of the peak and `sum` returns the sum of all intensities of the peak.

The `quantify` method returns `MSnSet` objects, that extend the well-known `eSet` class defined in the *Biobase* package. `MSnSet` instances are very similar

to `ExpressionSet` objects, except for the experiment meta-data that captures MIAPE specific information. The assay data is a matrix of dimensions $n \times m$ matrix, where m is the number of features/spectra originally in the `MSnExp` used as parameter in `quantify` and m is the number of reporter ions, that can be accessed with the `exprs` method. The meta data is directly inherited from the `MSnExp` instance.

```
> qnt <- quantify(experiment,
+                 method="trap",
+                 reporters=iTRAQ4,
+                 verbose=FALSE)
> qnt
```

```
MSnSet (storageMode: lockedEnvironment)
assayData: 54 features, 4 samples
  element names: exprs
protocolData: none
phenoData
  rowNames: iTRAQ4.114 iTRAQ4.115 iTRAQ4.116 iTRAQ4.117
  varLabels: mz reporters
  varMetadata: labelDescription
featureData
  featureNames: X1 X10 ... X9 (54 total)
  fvarLabels: spectum index ... collision.energy (10 total)
  fvarMetadata: labelDescription
experimentData: use 'experimentData(object)'
Annotation: No annotation
- - - Processing information - - -
Data loaded: Tue Sep 20 11:20:25 2011
Curves <= 3 set to '0': Tue Sep 20 11:20:57 2011
Spectra cleaned: Tue Sep 20 11:21:04 2011
MZ trimmed [112..120]
iTRAQ4 quantification by trapezoidation: Tue Sep 20 11:21:08 2011
 MSnbase version: 1.0.7
```

```
> head(exprs(qnt))
```

```
    iTRAQ4.114 iTRAQ4.115 iTRAQ4.116 iTRAQ4.117
X1  0.05523682 0.09904480  0.1193390 0.06391907
X10 0.11441803 0.12676620  0.1551361 0.16382599
X11 0.14594269 0.19406891  0.1471786 0.17577362
X12         NA 0.04356384         NA 0.04393768
X13 0.12229156 0.16637421  0.1710510 0.17578888
X14 0.08678055 0.13072205  0.1233521 0.12786865
```

A `MSnSet` object is meant to be compatible with further downstream packages for data normalisation and statistical analysis. Below is an example of applying variance stabilization normalisation (Huber et al., 2002) to the iTRAQ reporter intensities (Karp et al., 2010).

```
> library("vsn")
> qnt.vsn <- vsn2(exprs(qnt))
```

The results of figure 3 of Karp et al. (2010) are reproduced with the dummy experiment and show on figure 4.

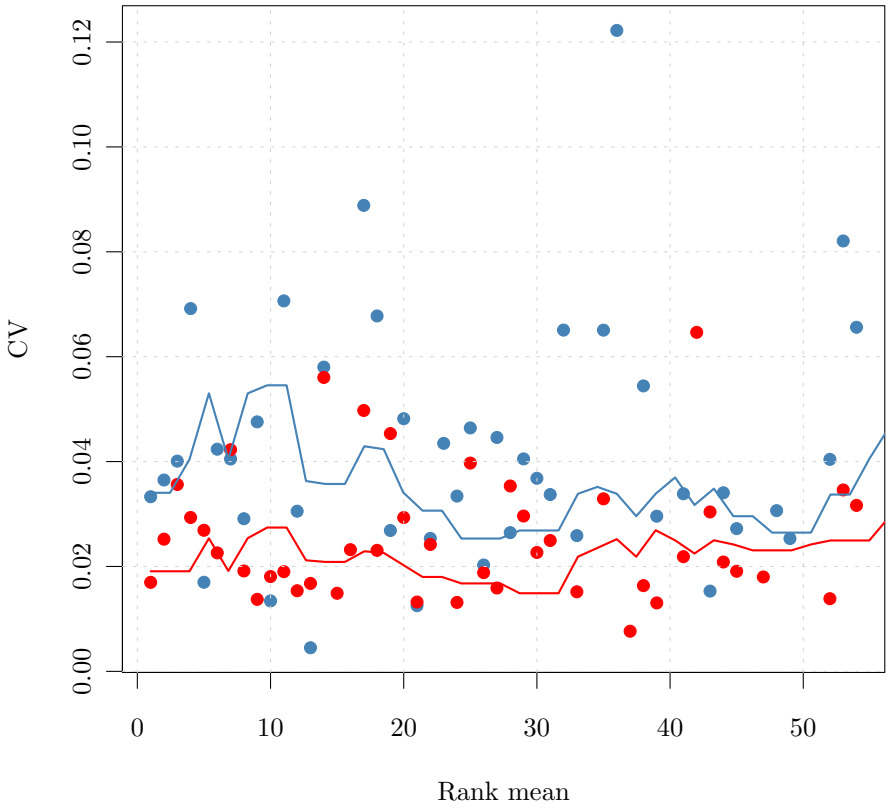

Figure 4: CV versus signal intensity comparison for log2 (blue) and vsn (red) transformed data. Lines indicate running CV medians.

8 Session information

- R version 2.13.1 (2011-07-08), `x86_64-unknown-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=C`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, grDevices, graphics, grid, methods, stats, tools, utils
- Other packages: Biobase 2.12.2, MSnbase 1.0.7, Rcpp 0.9.6, cacheSweave 0.6, codetools 0.2-8, filehash 2.2, formatR 0.2-3, getopt 1.16, ggplot2 0.8.9, highlight 0.2-5, optparse 0.9.1, parser 0.0-13, pgfSweave 1.2.1, plyr 1.6, proto 0.3-9.2, reshape 0.8.4, stashR 0.3-4, tikzDevice 0.6.1, vsn 3.20.0, zoo 1.7-4
- Loaded via a namespace (and not attached): IRanges 1.10.6, affy 1.30.0, affyio 1.20.0, digest 0.5.0, lattice 0.19-33, limma 3.8.3, preprocessCore 1.14.0, xcms 1.26.1

References

Robert C. Gentleman, Vincent J. Carey, Douglas M. Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, Laurent Gautier, Yongchao Ge, Jeff Gentry, Kurt Hornik, Torsten Hothorn, Wolfgang Huber, Stefano Iacus, Rafael Irizarry, Friedrich Leisch, Cheng Li, Martin Maechler, Anthony J. Rossini, Gunther Sawitzki, Colin Smith, Gordon Smyth, Luke Tierney, Jean Y. H. Yang, and Jianhua Zhang. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol*, 5(10):–80, 2004. doi: 10.1186/gb-2004-5-10-r80. URL http://dx.doi.org/10.1186/gb-2004-5-10-r80.

Wolfgang Huber, Anja von Heydebreck, Holger Sueltmann, Annemarie Poustka, and Martin Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18 Suppl. 1:S96–S104, 2002.

Natasha A Karp, Wolfgang Huber, Pawel G Sadowski, Philip D Charles, Svenja V Hester, and Kathryn S Lilley. Addressing accuracy and precision issues in itraq quantitation. *Mol. Cell Proteomics*, 9(9):1885–97, 2010. doi: 10.1074/mcp.M900628-MCP200.

Lennart Martens, Matthew Chambers, Marc Sturm, Darren Kes sner, Fredrik Levander, Jim Shofstahl, Wilfred H Tang, Andreas Ro mpp, Steffen Neumann, Angel D Pizarro, Lu isa Montecchi-Palazzi, Natalie Tasman, Mike Coleman, Florian Reisinger, Pune et Souda, Henning Hermjakob, Pierre-Alain Binz,

and Eric W Deutsch. mzml - a community standard for mass spectrometry data. *Molecular & Cellular Proteomics : MCP*, 2010. doi: 10.1074/mcp.R110.000133.

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.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL http://www.R-project.org/. ISBN 3-900051-07-0.

Chris F. Taylor, Norman W. Paton, Kathryn S. Lilley, Pierre-Alain Binz, Randall K. Julian, Andrew R. Jones, Weimin Zhu, Rolf Apweiler, Ruedi Aebersold, Eric W. Deutsch, Michael J. Dunn, Albert J. R. Heck, Alexander Leitner, Marcus Macht, Matthias Mann, Lennart Martens, Thomas A. Neubert, Scott D. Patterson, Peipei Ping, Sean L. Seymour, Puneet Souda, Akira Tsugita, Joel Vandekerckhove, Thomas M. Vondriska, Julian P. Whitelegge, Marc R. Wilkins, Ioannnis Xenarios, John R. Yates, and Henning Hermjakob. The minimum information about a proteomics experiment (miape). *Nat Biotechnol*, 25(8):887–893, Aug 2007. doi: 10.1038/nbt1329. URL http://dx.doi.org/10.1038/nbt1329.

Chris F Taylor, Pierre-Alain Binz, Ruedi Aebersold, Michel Affolter, Robert Barkovich, Eric W Deutsch, David M Horn, Andreas HÃijhmer, Martin Kussmann, Kathryn Lilley, Marcus Macht, Matthias Mann, Dieter MÃijller, Thomas A Neubert, Janice Nickson, Scott D Patterson, Roberto Raso, Kathryn Resing, Sean L Seymour, Akira Tsugita, Ioannis Xenarios, Rong Zeng, and Randall K Julian. Guidelines for reporting the use of mass spectrometry in proteomics. *Nat. Biotechnol.*, 26(8):860–1, 2008. doi: 10.1038/nbt0808-860.