MSnbase development

Laurent Gatto[*]

February 25, 2016

Abstract

This vignette describes the classes implemented in *MSnbase* package. It is intended as a starting point for developers or users who would like to learn more or further develop/extend *pSet*.

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

Contents

[*] lg390@cam.ac.uk

Foreword

MSnbase is under active developed; current functionality is evolving and new features will be added. This software is free and open-source software. If you use it, please support the project by citing it in publications:

> Laurent Gatto and Kathryn S. Lilley. *MSnbase - an R/Bioconductor package for isobaric tagged mass spectrometry data visualization, processing and quantitation.* Bioinformatics 28, 288-289 (2011).

Questions and bugs

You are welcome to contact me directly about *MSnbase*. For bugs, typos, suggestions or other questions, please file an issue in our tracking system[1] providing as much information as possible, a reproducible example and the output of `sessionInfo()`.

If you wish to reach a broader audience for general questions about proteomics analysis using `R`, you may want to use the Bioconductor support site: https://support.bioconductor.org/.

1 Introduction

This document is not a replacement for the individual manual pages, that document the slots of the *MSnbase* classes. It is a centralised high-level description of the package design.

MSnbase aims at being compatible with the *Biobase* infrastructure [1]. Many meta data structures that are used in *eSet* and associated classes are also used here. As such, knowledge of the *Biobase development and the new eSet* vignette[2] would be beneficial.

The initial goal is to use the *MSnbase* infrastructure for labelled quantitation using reporter ions (iTRAQ [2] and TMT [3]). Spectral counting should be trivial to apply with current features, as long as identification data is at hand. Currently, no effort is invested to streamline label-free quantitative proteomics, although some effort has been done to keep the infrastructure flexible enough to accommodate more designs.

2 MSnbase classes

All classes have a *.__classVersion__* slot, of class *Versioned* from the *Biobase* package. This slot documents the class version for any instance to be used for debugging and object update purposes. Any

[1] https://github.com/lgatto/MSnbase/issues

[2] The vignette can directly be accessed with `vignette("BiobaseDevelopment",package="Biobase")` once *Biobase* is loaded.

change in a class implementation should trigger a version change.

2.1 pSet: a virtual class for raw mass spectrometry data and meta data

This virtual class is the main container for mass spectrometry data, i.e spectra, and meta data. It is based on the *eSet* implementation for genomic data. The main difference with *eSet* is that the `assayData` slot is an environment containing any number of *Spectrum* instances (see section 2.6).

One new slot is introduced, namely `processingData`, that contains one *MSnProcess* instance (see section 2.4). and the `experimentData` slot is now expected to contain *MIAPE* data (see section 2.5). The `annotation` slot has not been implemented, as no prior feature annotation is known in shotgun proteomics.

```
getClass("pSet")

Virtual Class "pSet" [package "MSnbase"]

Slots:

Name:              assayData            phenoData
Class:           environment NAnnotatedDataFrame

Name:            featureData       experimentData
Class:   AnnotatedDataFrame                MIAxE

Name:           protocolData       processingData
Class:   AnnotatedDataFrame           MSnProcess

Name:                 .cache   .__classVersion__
Class:           environment             Versions

Extends: "Versioned"

Known Subclasses: "MSnExp"
```

Future work Currently, few setters have been implemented.

2.2 MSnExp: a class for MS experiments

MSnExp extends *pSet* to store MS experiments. It does not add any new slots to *pSet*. Accessors and setters are all inherited from *pSet* and new ones should be implemented for *pSet*. Methods that manipulate actual data in experiments are implemented for *MSnExp* objects.

```
getClass("MSnExp")

Class "MSnExp" [package "MSnbase"]

Slots:

Name:            assayData           phenoData
Class:         environment NAnnotatedDataFrame

Name:          featureData      experimentData
Class:  AnnotatedDataFrame               MIAxE

Name:         protocolData      processingData
Class:  AnnotatedDataFrame          MSnProcess

Name:               .cache   .__classVersion__
Class:         environment            Versions

Extends:
Class "pSet", directly
Class "Versioned", by class "pSet", distance 2
```

2.3 MSnSet: a class for quantitative proteomics data

This class stores quantitation data and meta data after running `quantify` on an *MSnExp* object or by creating an *MSnSet* instance from an external file, as described in the `MSnbase-io` vignette and in `?readMSnSet`, `readMzTabData`, etc. The quantitative data is in form of a $m \times n$ 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 (see section 2.7). If read from an external file, n corresponds to the number of features (protein groups, proteins, peptides, spectra) in the file and m is the number of columns with quantitative data (samples) in the file.

This prompted to keep a similar implementation as the *ExpressionSet* class, while adding the proteomics-specific annotation slot introduced in the *pSet* class, namely `processingData` for objects of class *MSnProcess* (see section 2.4).

```
getClass("MSnSet")

Class "MSnSet" [package "MSnbase"]

Slots:

Name:      experimentData     processingData               qual
Class:              MIAPE         MSnProcess         data.frame
```

```
Name:           assayData          phenoData        featureData
Class:          AssayData AnnotatedDataFrame AnnotatedDataFrame

Name:          annotation       protocolData  .__classVersion__
Class:          character AnnotatedDataFrame           Versions

Extends:
Class "eSet", directly
Class "VersionedBiobase", by class "eSet", distance 2
Class "Versioned", by class "eSet", distance 3
```

The *MSnSet* class extends the virtual *eSet* class to provide compatibility for *ExpressionSet*-like behaviour. The experiment meta-data in `experimentData` is also of class *MIAPE* (see section 2.5). The `annotation` slot, inherited from *eSet* is not used. As a result, it is easy to convert *ExpressionSet* data from/to *MSnSet* objects with the coersion method `as`.

```
data(itraqdata)
class(msnset)

[1] "MSnSet"
attr(,"package")
[1] "MSnbase"

class(as(msnset, "ExpressionSet"))

[1] "ExpressionSet"
attr(,"package")
[1] "Biobase"

data(sample.ExpressionSet)
class(sample.ExpressionSet)

[1] "ExpressionSet"
attr(,"package")
[1] "Biobase"

class(as(sample.ExpressionSet, "MSnSet"))

[1] "MSnSet"
attr(,"package")
[1] "MSnbase"
```

2.4 MSnProcess: a class for logging processing meta data

This class aims at recording specific manipulations applied to *MSnExp* or *MSnSet* instances. The `processing` slot is a `character` vector that describes major processing. Most other slots are of class `logical` that indicate whether the data has been centroided, smoothed, . . . although many of the functionality is not implemented yet. Any new processing that is implemented should be documented and logged here.

It also documents the raw data file from which the data originates (`files` slot) and the *MSnbase* version that was in use when the *MSnProcess* instance, and hence the *MSnExp*/*MSnSet* objects, were originally created.

```
getClass("MSnProcess")

Class "MSnProcess" [package "MSnbase"]

Slots:

Name:               files        processing            merged
Class:          character         character           logical

Name:             cleaned      removedPeaks          smoothed
Class:            logical         character           logical

Name:             trimmed        normalised    MSnbaseVersion
Class:            numeric           logical         character

Name:   .__classVersion__
Class:           Versions

Extends: "Versioned"
```

2.5 MIAPE: Minimum Information About a Proteomics Experiment

The Minimum Information About a Proteomics Experiment [4, 5] *MIAPE* class describes the experiment, including contact details, information about the mass spectrometer and control and analysis software.

```
getClass("MIAPE")

Class "MIAPE" [package "MSnbase"]

Slots:

Name:                      title                      url
Class:                 character                character

Name:                   abstract                pubMedIds
Class:                 character                character

Name:                    samples            preprocessing
Class:                      list                     list
```

```
Name:                      other                dateStamp
Class:                      list                character

Name:                       name                      lab
Class:                 character                character

Name:                    contact                    email
Class:                 character                character

Name:            instrumentModel   instrumentManufacturer
Class:                 character                character

Name:   instrumentCustomisations             softwareName
Class:                 character                character

Name:            softwareVersion        switchingCriteria
Class:                 character                character

Name:             isolationWidth            parameterFile
Class:                   numeric                character

Name:                  ionSource         ionSourceDetails
Class:                 character                character

Name:                   analyser          analyserDetails
Class:                 character                character

Name:               collisionGas        collisionPressure
Class:                 character                  numeric

Name:            collisionEnergy             detectorType
Class:                 character                character

Name:        detectorSensitivity        .__classVersion__
Class:                 character                 Versions

Extends:
Class "MIAxE", directly
Class "Versioned", by class "MIAxE", distance 2
```

2.6 Spectrum et al.: classes for MS spectra

Spectrum is a virtual class that defines common attributes to all types of spectra. MS1 and MS2 specific attributes are defined in the *Spectrum1* and *Spectrum2* classes, that directly extend *Spectrum*.

```
getClass("Spectrum")

Virtual Class "Spectrum" [package "MSnbase"]

Slots:

Name:            msLevel        peaksCount                rt
Class:           integer           integer           numeric

Name:     acquisitionNum         scanIndex               tic
Class:           integer           integer           numeric

Name:                 mz         intensity          fromFile
Class:           numeric           numeric           integer

Name:         centroided .__classVersion__
Class:           logical          Versions

Extends: "Versioned"

Known Subclasses: "Spectrum2", "Spectrum1"
```

```
getClass("Spectrum1")

Class "Spectrum1" [package "MSnbase"]

Slots:

Name:           polarity           msLevel        peaksCount
Class:           integer           integer           integer

Name:                 rt    acquisitionNum         scanIndex
Class:           numeric           integer           integer

Name:                tic                mz         intensity
Class:           numeric           numeric           numeric

Name:           fromFile        centroided .__classVersion__
Class:           integer           logical          Versions

Extends:
```

```
Class "Spectrum", directly
Class "Versioned", by class "Spectrum", distance 2
```

```
getClass("Spectrum2")

Class "Spectrum2" [package "MSnbase"]

Slots:

Name:               merged       precScanNum       precursorMz
Class:             numeric           integer           numeric

Name:  precursorIntensity    precursorCharge    collisionEnergy
Class:             numeric           integer           numeric

Name:              msLevel        peaksCount                rt
Class:             integer           integer           numeric

Name:       acquisitionNum         scanIndex               tic
Class:             integer           integer           numeric

Name:                   mz         intensity          fromFile
Class:             numeric           numeric           integer

Name:           centroided  .__classVersion__
Class:             logical          Versions

Extends:
Class "Spectrum", directly
Class "Versioned", by class "Spectrum", distance 2
```

2.7 ReporterIons: a class for isobaric tags

The iTRAQ and TMT (or any other peak of interest) are implemented *ReporterIons* instances, that essentially defines an expected MZ position for the peak and a width around this value as well a names for the reporters.

```
getClass("ReporterIons")

Class "ReporterIons" [package "MSnbase"]

Slots:

Name:                name     reporterNames       description
Class:          character         character         character
```

```
Name:                  mz               col             width
Class:            numeric         character           numeric

Name:  .__classVersion__
Class:           Versions

Extends: "Versioned"
```

2.8 NAnnotatedDataFrame: multiplexed AnnotatedDataFrames

The simple expansion of the *AnnotatedDataFrame* classes adds the `multiplex` and `multiLabel` slots to document the number and names of multiplexed samples.

```
getClass("NAnnotatedDataFrame")
```

```
Class "NAnnotatedDataFrame" [package "MSnbase"]

Slots:

Name:           multiplex       multiLabels       varMetadata
Class:            numeric         character        data.frame

Name:                data         dimLabels .__classVersion__
Class:         data.frame         character          Versions

Extends:
Class "AnnotatedDataFrame", directly
Class "Versioned", by class "AnnotatedDataFrame", distance 2
```

2.9 Other classes

Lists of MSnSet instances

3 Miscellaneous

Unit tests *MSnbase* implements unit tests with the *testthat* package.

Processing methods Methods that process raw data, i.e. spectra should be implemented for *Spectrum* objects first and then `eapply`'ed (or similar) to the `assayData` slot of an *MSnExp* instance in the specific method.

4 Session information

- R version 3.2.3 (2015-12-10), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `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, parallel, stats, stats4, utils
- Other packages: AnnotationDbi 1.32.3, Biobase 2.30.0, BiocGenerics 0.16.1, BiocParallel 1.4.3, IRanges 2.4.8, MLInterfaces 1.50.0, MSnbase 1.18.1, ProtGenerics 1.2.1, Rcpp 0.12.3, RcppClassic 0.9.6, Rdisop 1.30.0, S4Vectors 0.8.11, XML 3.98-1.3, annotate 1.48.0, cluster 2.0.3, ggplot2 2.0.0, gplots 2.17.0, knitr 1.12.3, mzR 2.4.0, pRoloc 1.10.1, pRolocdata 1.8.0, reshape2 1.4.1, zoo 1.7-12
- Loaded via a namespace (and not attached): BiocInstaller 1.20.1, BiocStyle 1.8.0, DBI 0.3.1, DEoptimR 1.0-4, FNN 1.1, KernSmooth 2.23-15, MALDIquant 1.14, MASS 7.3-45, Matrix 1.2-3, MatrixModels 0.4-1, R6 2.1.2, RColorBrewer 1.1-2, RCurl 1.95-4.7, RSQLite 1.0.0, SparseM 1.7, affy 1.48.0, affyio 1.40.0, assertthat 0.1, base64enc 0.1-3, biomaRt 2.26.1, bitops 1.0-6, caTools 1.17.1, car 2.1-1, caret 6.0-64, class 7.3-14, codetools 0.2-14, colorspace 1.2-6, digest 0.6.9, diptest 0.75-7, doParallel 1.0.10, dplyr 0.4.3, e1071 1.6-7, evaluate 0.8, flexmix 2.3-13, foreach 1.4.3, formatR 1.2.1, fpc 2.1-10, futile.logger 1.4.1, futile.options 1.0.0, gbm 2.1.1, gdata 2.17.0, genefilter 1.52.1, ggvis 0.4.2, gtable 0.1.2, gtools 3.5.0, highr 0.5.1, htmltools 0.3, htmlwidgets 0.5, httpuv 1.3.3, hwriter 1.3.2, impute 1.44.0, iterators 1.0.8, kernlab 0.9-23, labeling 0.3, lambda.r 1.1.7, lattice 0.20-33, limma 3.26.8, lme4 1.1-11, lpSolve 5.6.13, magrittr 1.5, mclust 5.1, mgcv 1.8-11, mime 0.4, minqa 1.2.4, mlbench 2.1-1, modeltools 0.2-21, munsell 0.4.3, mvtnorm 1.0-5, mzID 1.8.0, nlme 3.1-124, nloptr 1.0.4, nnet 7.3-12, pbkrtest 0.4-6, pcaMethods 1.60.0, pls 2.5-0, plyr 1.8.3, prabclus 2.2-6, preprocessCore 1.32.0, proxy 0.4-15, quantreg 5.21, randomForest 4.6-12, rda 1.0.2-2, rgl 0.95.1441, robustbase 0.92-5, rpart 4.1-10, sampling 2.7, scales 0.3.0, sfsmisc 1.1-0, shiny 0.13.1, splines 3.2.3, stringi 1.0-1, stringr 1.0.0, survival 2.38-3, threejs 0.2.1, tools 3.2.3, trimcluster 0.1-2, vsn 3.38.0, xtable 1.8-2, zlibbioc 1.16.0

References

[1] 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. URL: http://dx.doi.org/10.1186/gb-2004-5-10-r80, `doi:10.1186/gb-2004-5-10-r80`.

[2] Philip L. Ross, Yulin N. Huang, Jason N. Marchese, Brian Williamson, Kenneth Parker, Stephen Hattan, Nikita Khainovski, Sasi Pillai, Subhakar Dey, Scott Daniels, Subhasish Purkayastha, Peter

Juhasz, Stephen Martin, Michael Bartlet-Jones, Feng He, Allan Jacobson, and Darryl J. Pappin. Multiplexed protein quantitation in saccharomyces cerevisiae using amine-reactive isobaric tagging reagents. *Mol Cell Proteomics*, 3(12):1154–1169, Dec 2004. URL: http://dx.doi.org/10.1074/mcp.M400129-MCP200, doi:10.1074/mcp.M400129-MCP200.

[3] Andrew Thompson, Jürgen Schäfer, Karsten Kuhn, Stefan Kienle, Josef Schwarz, Günter Schmidt, Thomas Neumann, R Johnstone, A Karim A Mohammed, and Christian Hamon. Tandem mass tags: a novel quantification strategy for comparative analysis of complex protein mixtures by MS/MS. *Anal. Chem.*, 75(8):1895–904, 2003.

[4] 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. URL: http://dx.doi.org/10.1038/nbt1329, doi:10.1038/nbt1329.

[5] Chris F Taylor, Pierre-Alain Binz, Ruedi Aebersold, Michel Affolter, Robert Barkovich, Eric W Deutsch, David M Horn, Andreas Hhmer, Martin Kussmann, Kathryn Lilley, Marcus Macht, Matthias Mann, Dieter Mller, 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.