

MSnbase development

Laurent Gatto

lg390@cam.ac.uk

Cambridge Center for Proteomics

Kathryn S. Lilley Group

University of Cambridge

December 5, 2012

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

1	Introduction	2
2	MSnbase classes	2
2.1	pSet: a virtual class for raw mass spectrometry data and meta data	2
2.2	MSnExp: a class for MS experiments	3
2.3	MSnSet: a class for quantitative proteomics data	3
2.4	MSnProcess: a class for logging processing meta data	4
2.5	MIAPE: Minimum Information About a Proteomics Experiment	4
2.6	Spectrum <i>et al.</i> : classes for MS spectra	5
2.7	ReporterIons: a class for isobaric tags	7
2.8	NAnnotatedDataFrame: multiplexed AnnotatedDataFrames	7
3	Miscellaneous	8
4	Session information	8

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).

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 list¹.

¹<https://stat.ethz.ch/mailman/listinfo/bioconductor>

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 [Gentleman et al. \(2004\)](#). 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 ([Ross et al., 2004](#)) and TMT ([Thompson et al., 2003](#))). 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 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"
```

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

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. The quantitative data is in form of a $n \times m$ matrix, where m is the number of features/spectra originally in the MSnExp used as parameter in `quantify` and n is the number of reporter ions (see section 2.7).

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).

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.

```
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

```

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 ([Taylor et al., 2007](#), [2008](#)) 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

```

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 2.15.2 (2012-10-26), 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=en_US.UTF-8, LC_MESSAGES=en_US.UTF-8, LC_PAPER=C, LC_NAME=C, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=en_US.UTF-8, LC_IDENTIFICATION=C
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: Biobase 2.18.0, BiocGenerics 0.4.0, MSnbase 1.6.2, Rcpp 0.10.1, codetools 0.2-8, ggplot2 0.9.2.1, knitr 0.9, mzR 1.4.2
- Loaded via a namespace (and not attached): BiocInstaller 1.8.3, IRanges 1.16.4, MASS 7.3-22, RColorBrewer 1.0-5, affy 1.36.0, affyio 1.26.0, colorspace 1.2-0, dichromat 1.2-4, digest 0.6.0, evaluate 0.4.3, formatR 0.7, grid 2.15.2, gtable 0.1.1, labeling 0.1, lattice 0.20-10, limma 3.14.3, memoise 0.1, munsell 0.4, parallel 2.15.2, plyr 1.7.1, preprocessCore 1.20.0, proto 0.3-9.2, reshape2 1.2.2, scales 0.2.2, stats4 2.15.2, stringr 0.6.1, tools 2.15.2, vsn 3.26.0, zlibbioc 1.4.0

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>.

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. doi: 10.1074/mcp.M400129-MCP200. URL <http://dx.doi.org/10.1074/mcp.M400129-MCP200>.

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, Ioannis 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 Håjmer, Martin Kussmann, Kathryn Lilley, Marcus Macht, Matthias Mann, Dieter Måjller, 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.

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.