

# MSnID: A Convenience Tool for Handling MS/MS Identifications in R/Bioconductor

Vladislav A Petyuk<sup>1</sup>   Laurent Gatto<sup>2</sup>   Thomas L Pedersen<sup>3</sup>  
Samuel Payne<sup>1</sup>   Richard D Smith<sup>1</sup>

<sup>1</sup>Pacific Northwest National Laboratory, Richland, WA, USA

<sup>2</sup>University of Cambridge, Cambridge, UK

<sup>3</sup>Technical University of Denmark, Kgs. Lyngby, Denmark

US HUPO, Seattle, WA, April 6-9, 2014

# R as a *Lingua Franca* of Statistical Computing

- 1-2 million users. Used in any scientific discipline that requires statistically rigorous data analysis.
- Having a common language helps with reporting modern (complex) data analyses in a reproducible manner.
- 5,000+ packages that cover pretty much all of the statistics and exploratory data analysis are available at the CRAN repository.
- Bioconductor is a focused open development project providing tools for analysis of genomic data.
- Out of 749 Bioconductor packages, 48 (6-7%) are proteomics related.

# R Proteomics Packages - Building Blocks for Data Analysis

Support of community standard formats mzXML, mzML, mzIdentML

mzR, mzID

Quantification (mostly iTRAQ)

MSnbase, isobar

MS/MS search

rTANDEM, shinyTANDEM, MSGFplus, MSGFgui

Isotope distributions

BRAIN, IPPD

Statistical analysis

MSstats, msmsTests

Great starting package

RforProteomics<sup>1</sup>

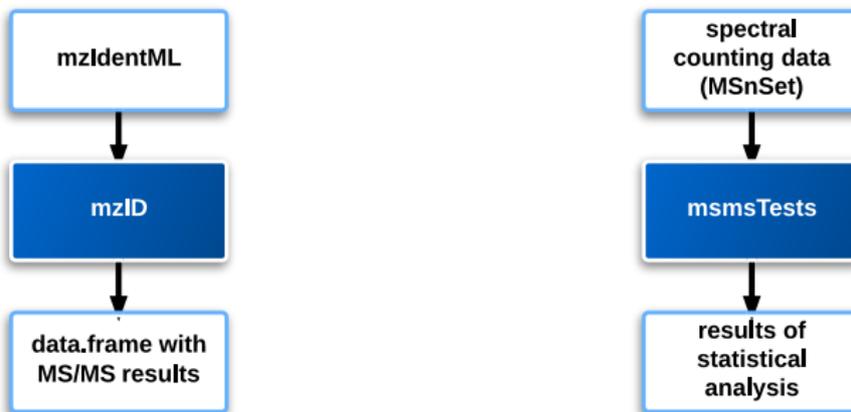


---

<sup>1</sup>Laurent Gatto and Andy Christoforou. "Using R and Bioconductor for proteomics data analysis." *Biochim Biophys Acta* 1844.1 Pt A (2014): 42–51. Web.

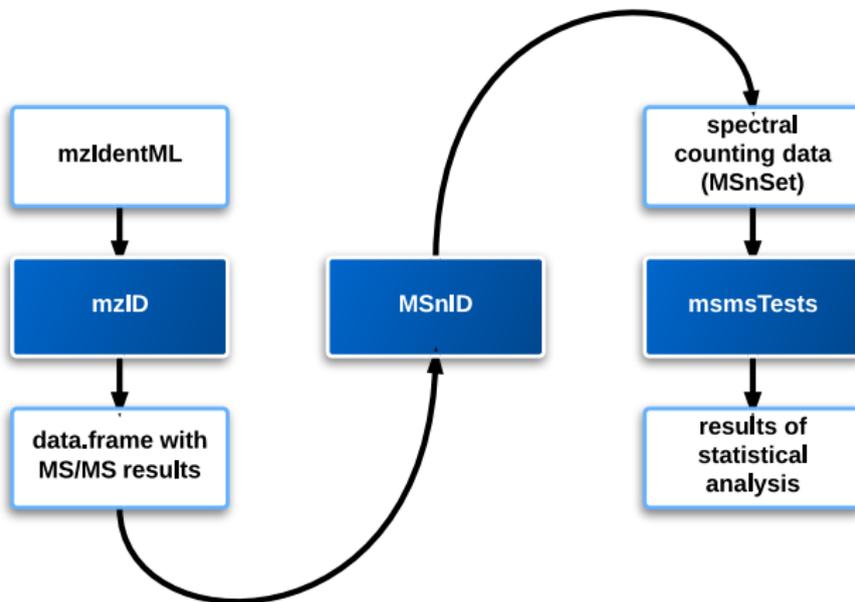
# Motivation for MSnID

Need for a capability to quickly and interactively explore, visualize and manipulate MS/MS identifications in R/Bioconductor environment.



# Motivation for MSnID

Need for a capability to quickly and interactively explore, visualize and manipulate MS/MS identifications in R/Bioconductor environment.



# Installation and Example Script

<https://github.com/vladpetyuk/MSnID/>

README.md

## MSnID

A Convenience Tool for Handling MS/MS Proteomics Identifications

To install `MSnID` package, run these commands from R prompt:

```

#installing devtools
install.packages("devtools")
# installing dependencies from d3rn0 to ensure latest versions
install_github("d3rn0", "thomaspd", quiet=TRUE)
install_github("MSnbase", "lgaetto", quiet=TRUE)
# installing the MSnID itself
install_github("MSnID", "vladpetyuk", quiet=TRUE)

```

Download an example *C. elegans* R script:

```

library("RCurl")
script_url <- "https://raw2.github.com/vladpetyuk/MSnID/master/inst/example/c_elegans.R"
script <- getURL(script_url, ssl.verifypeer=FALSE, followlocation=TRUE)
writeLines(script, "c_elegans.R")

```

A brief description:

- Input: mzIdentML files with MS/MS search results
- Utilities to explore MS/MS search results and assess data confidence metrics (MS/MS match scoring, parent ion mass measurement accuracy, missed cleavages ...)

Running the `c_elegans.R` script will reproduce the results shown in the current presentation. The example is based on 10 datasets from the study of proteomes of long-living *daf-2* and normal-living *daf-2; daf-16* strains of *C. elegans*<sup>2</sup>.

<sup>2</sup>Geert Depuydt et al. "Reduced insulin/insulin-like growth factor-1 signaling and dietary restriction inhibit translation but preserve muscle mass in *Caenorhabditis elegans*." *Mol Cell Proteomics* 12.12 (2013): 3624–3639. Web.

# MSnID Object

```
> library("MSnID")
> msnid <- MSnID(".") # provide working directory
> mzids <- list.files(".", pattern=".mzid.gz")
> msnid <- read_mzIDs(msnid, mzids)
> # alternatively
> psms(msnid) <- yourMSMSdata

> msnid
```

MSnID object

Working directory: "."

#Spectrum Files: 10

#PSMs: 190589 at 31 % FDR

#Peptides: 57662 at 75 % FDR

#Accessions: 28728 at 94 % FDR

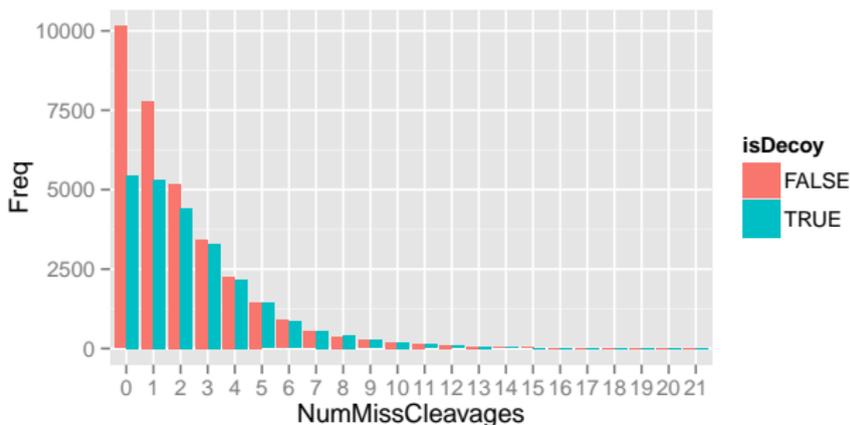
Slots inside of MSnID object

- @workDir - working directory for storing cache and output
- @psms - MS/MS search results in the form of high-performance data.table object

# Analysis of Peptide Sequences

- Irregular cleavage termini
- Missed cleavages
- Any other sequence patterns

```
> msnid <- assess_missed_cleavages(msnid,  
+ missedCleavagePattern="[KR](?=[^P$])")
```

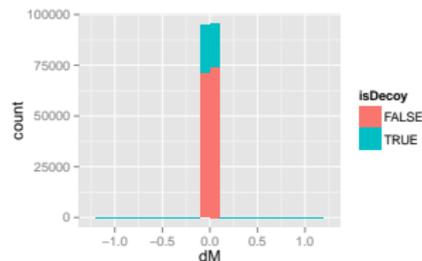
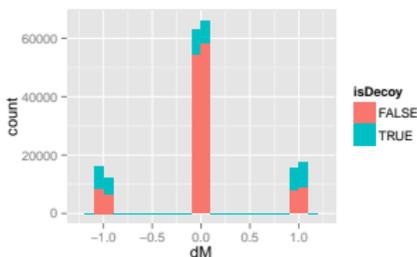


# Problem of Selection of Non-Monoisotopic Ion

## Solutions:

- Enable monoisotopic ion precursor selection (MIPS) option on the instrument
- Post-experimental deisotoping (DeconMSn<sup>3</sup>)
- Subtracting/adding  $C^{13} - C^{12}$  difference: `correct_peak_selection`

```
> msnid <- correct_peak_selection(msnid)
```



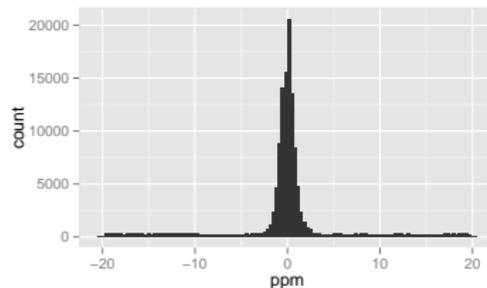
<sup>3</sup>Anoop M. Mayampurath et al. "DeconMSn: a software tool for accurate parent ion monoisotopic mass determination for tandem mass spectra." *Bioinformatics* 24.7 (2008): 1021–1023. Web.

# Mass Measurement Error Problem

Solutions:

- Instrument calibration
- Post-experimental recalibration (DtaRefinery<sup>4</sup>)
- `recalibrate` function

```
> msnid <- recalibrate(msnid)
> ppm <- mass_measurement_error(msnid)
```



---

<sup>4</sup>Vladislav A. Petyuk et al. "DtaRefinery, a software tool for elimination of systematic errors from parent ion mass measurements in tandem mass spectra data sets." *Mol Cell Proteomics* 9.3 (2010): 486–496. Web.

# Filtering Criteria

Filtering criteria can be anything as long as they are explicitly present in MSnID object. In this example we will use:

- $-\log_{10}$  of spectrum E-value
- absolute mass measurement error in ppm

adding filtering criteria to the MSnID object

```
> msnid$msmsScore <-  $-\log_{10}(\text{msnid}\$`ms-gf:specvalue`)$   
> msnid$absMassErr <-  $\text{abs}(\text{mass\_measurement\_error}(\text{msnid}))$ 
```

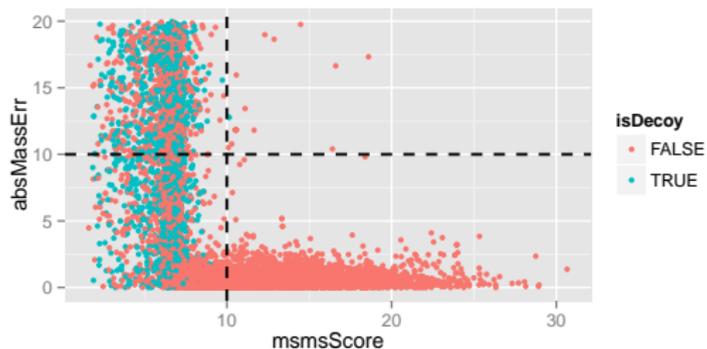
# MSnIDFilter Object

## initialization

```
> filtObj <- MSnIDFilter(msnid)
```

## setting filter criteria

```
> filtObj$absMassErr <- list(comparison="<", threshold=10.0)  
> filtObj$msmsScore <- list(comparison=">", threshold=10.0)
```



# Filter Optimization

`MSnIDFilter` object can be evolved and optimized.

## Optimization objective:

Maximize the number of IDs (spectra, unique peptide sequences or proteins) within given FDR upper limit.

## Optimization options:

- **Grid**: Simply enumerates the number `n.iter` combinations and evaluates FDR at each of them. Great for coarse optimization.
- **Nelder-Mead**: The recommended practical option.
- **SANN**: Simulated annealing, computationally very intensive optimization.

```
> filtObj.nm <- optimize_filter(filtObj, msnid,  
+                               level="Peptide",  
+                               fdr.max=0.01,  
+                               method="Nelder-Mead",  
+                               n.iter=500)
```

# Filter Optimization Results

Results of the filter optimization with the objective to achieve maximum number of unique peptide sequences, while not exceeding 1% FDR.

good guess

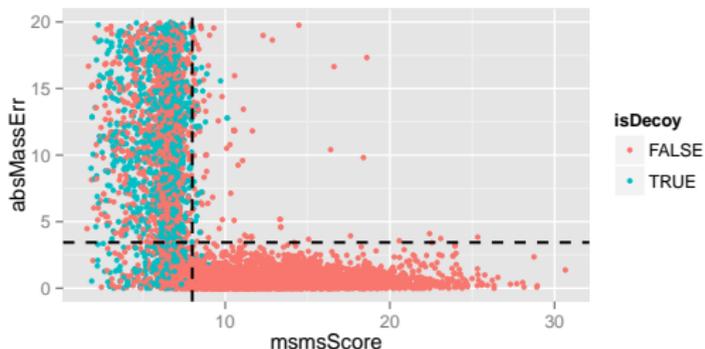
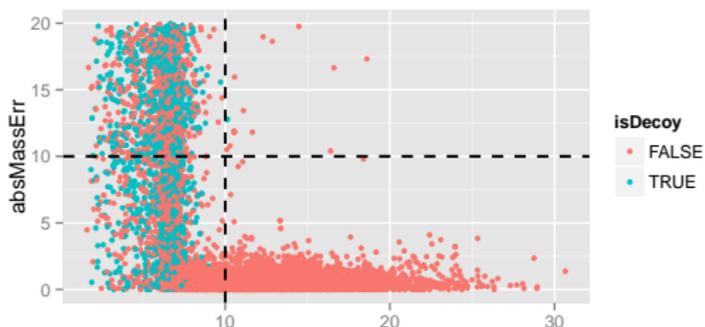
```
> filtObj
```

```
MSnIDFilter object
(absMassErr < 10) & (msmsScore > 10)
```

optimized

```
> filtObj.nm
```

```
MSnIDFilter object
(absMassErr < 3.4) & (msmsScore > 8)
```



# Filtering The Data

- `evaluate_filter`: Returns number of identifications (PSMs, peptides or proteins) and corresponding FDR.
- `apply_filter`: Returns filtered MSnID object.

## filtering spurious identifications

```
> msnid <- apply_filter(msnid, filtObj.nm)
```

```
> msnid
```

MSnID object

Working directory: "."

#Spectrum Files: 10

#PSMs: 86200 at 0.15 % FDR

#Peptides: 6846 at 0.99 % FDR

#Accessions: 2050 at 4.8 % FDR

## removing decoy and contaminant hits

```
> msnid <- apply_filter(msnid, "!isDecoy")
```

```
> msnid <- apply_filter(msnid, "!grepl('Contaminant',Accession)")
```

# Spectral Counts Quantitative Data - MSnSet

- The key is to convert MSnID to another class object available at Bioconductor that is designed for handling quantitative data.
- MSnSet class (defined in MSnbase package) is a subclass of one of the central Bioconductor class eSet.
- MSnSet has an access to dozens of eSet-aware Bioconductor packages.

```
> msnset <- as(msnid, "MSnSet")
```

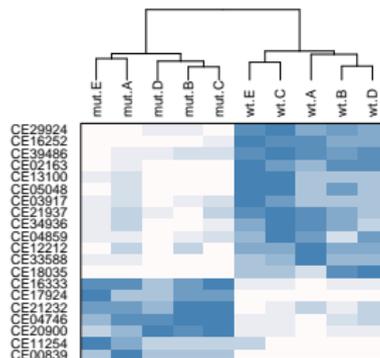
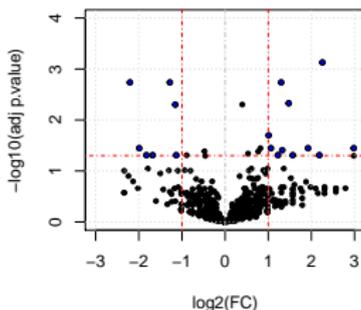
Leveraging MSnbase package functionality for summing peptide counts to protein level.

```
> msnset <- combineFeatures(msnset,  
+                           fData(msnset)$Accession,  
+                           redundancy.handler="unique",  
+                           fun="sum",  
+                           cv=FALSE)
```

# Leveraging `msmsTest` Package for Hypothesis Testing

Quasi-likelihood Poisson model for spectral count analysis<sup>5</sup>.

```
> library("msmsTests")
> alt.f <- "y ~ Daf.16.type + 1"
> null.f <- "y ~ 1"
> div <- colSums(exprs(msnset)) # normalization factor
> res <- msms.glm.qlll(msnset, alt.f, null.f, div=div)
```



<sup>5</sup>Ming Li et al. "Comparative shotgun proteomics using spectral count data and quasi-likelihood modeling." *J Proteome Res* 9.8 (2010): 4295–4305. [Web](#).

# Future Features

- Remapping from one protein sequence collection to another with help of `biomaRt` package
- Inference of parsimonious set of proteins from peptide sequences
- Post-translational modifications
- Top-down