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

Vladislav A Petyuk\textsuperscript{1}  Laurent Gatto\textsuperscript{2}  Thomas L Pedersen\textsuperscript{3}
Samuel Payne\textsuperscript{1}  Richard D Smith\textsuperscript{1}

\textsuperscript{1}Pacific Northwest National Laboratory, Richland, WA, USA
\textsuperscript{2}University of Cambridge, Cambridge, UK
\textsuperscript{3}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.
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
- RforProteomics¹

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/

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\(^2\).

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

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

```
          NumMissCleavages  Freq isDecoy
isDecoy     FALSE     TRUE
0           2500      5000
1           7500      2500
2            0        1
3            0        1
4            0        1
5            0        1
6            0        1
7            0        1
8            0        1
9            0        1
10           0        1
11           0        1
12           0        1
13           0        1
14           0        1
15           0        1
16           0        1
17           0        1
18           0        1
19           0        1
20           0        1
21           0        1
```

Bar chart showing the frequency of missed cleavages by the number of missed cleavages and whether the protein is a decoy.
Problem of Selection of Non-Monoisotopic Ion

Solutions:

- Enable monoisotopic ion precursor selection (MIPS) option on the instrument
- Post-experimental deisotoping (DeconMSn^3)
- Subtracting/adding $C^{13} - C^{12}$ difference:
  \[
  \text{correct\_peak\_selection}
  \]

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

---

Mass Measurement Error Problem

Solutions:

- Instrument calibration
- Post-experimental recalibration (DtaRefinery⁴)
- `recalibrate` function

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

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

- -log10 of spectrum E-value
- absolute mass measurement error in ppm

```
> msnid$msmsScore <- -log10(msnid$`ms-gf:specevalue``)
> msnid$absMassErr <- abs(mass_measurement_error(msnid))
```
**MSnIDFilter Object**

**Initialization**

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

**Setting filter criteria**

```r
> 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.

```r
> 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**

\[
> \text{filtObj}
\]

MSnIDFilter object
\[(\text{absMassErr} < 10) \& (\text{msmsScore} > 10)\]

**optimized**

\[
> \text{filtObj.nm}
\]

MSnIDFilter object
\[(\text{absMassErr} < 3.4) \& (\text{msmsScore} > 8)\]
Filtering The Data

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

```r
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.

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

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

```r
> 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. 

```r
> 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)
```

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