Spectra: A scalable and flexible infrastructure for mass spectrometry data in R

Johannes Rainer[1,*], Sebastian Gibb[2], Laurent Gatto[3]

[1] Institute for Biomedicine, Eurac Research, Affiliated Institute of the University of Lübeck, 39100 Bolzano, Italy.
[2] Department of Anaesthesiology and Intensive Care, University Medicine Greifswald, University of Greifswald, 17475 Greifswald, Germany.
[3] Computational Biology Unit, de Duve Institute, Université catholique de Louvain, Brussels, 1200, Belgium.

Introduction

- Easy expandability by separation of user functionality from data representation and storage:
  - `Spectra`: provides functions to handle and analyze MS data.
  - `MsBackend`: manages and provides the MS data to `Spectra`.
- **Use case**: match experimental MS2 spectra against public database (full version of the tutorial: https://github.com/jorainer/SpectraTutorials).

Import from mzML files

- Import data from an LC-MS/MS experiment (4 mzML files).
- `MsBackendMzR` supports data import from mzML/mzXML/CDF files; has a small memory footprint hence enabling analysis of large scale experiments.

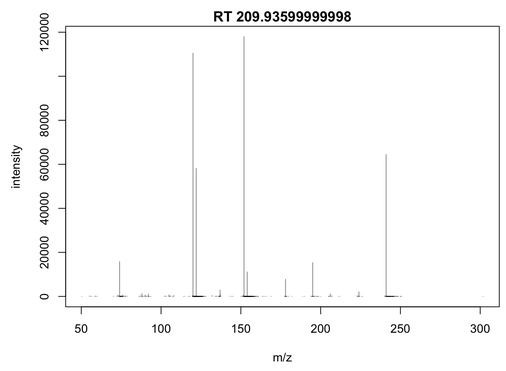
```
fls <- dir("data/mzML", pattern = "mzML$", full.names = TRUE)
sps_all <- Spectra(fls, backend = MsBackendMzR())
```

- Identify MS2 spectra with precursor m/z matching the [M+H]+ ion of Cystine.

```
mz <- 241.0311
sps <- filterPrecursorMz(sps_all, mz = mz + ppm(c(-mz, mz), 10))
```

- Plot first spectrum: raw spectra seem to be noisy.

```
plotSpectra(sps[1])
```



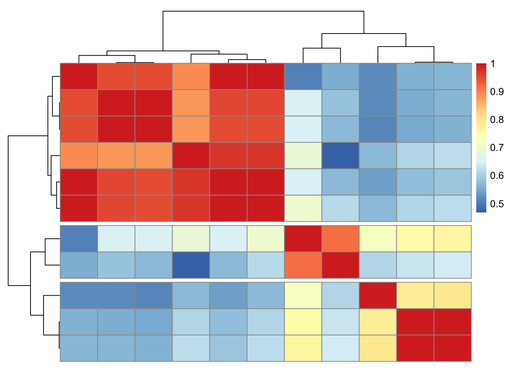
- Use `filterIntensity` to remove intensities below 5% of base peak signal.
- *Normalize* each spectrum by applying a custom function `norm_int` to each spectrum with `addProcessing`.

```
low_int <- function(x) x > max(x) * 0.05
sps <- filterIntensity(sps, intensity = low_int)

norm_int <- function(x, ...) {
    x[, "intensity"] <- 100 * x[, "intensity"] / max(x[, "intensity"])
    x
}
sps <- addProcessing(sps, norm_int)
```

- Calculate pairwise similarity between spectra and visualize.

```
cormat <- compareSpectra(sps, ppm = 20, FUN = ndotproduct)
hm <- pheatmap(cormat, cutree_rows = 3)
```



- Spectra group into 3 clusters: related to the applied collision energy.

- Proceed analysis with spectra from 20eV collision energy.

```
sps_ce20 <- split(sps, cutree(hm$tree_row, 3))[[1L]]
```

Comparison against spectra from HMDB

- Next step: compare spectra against *reference* spectra from HMDB.
- `MsBackendHmdbXml` supports import from HMDB MS/MS spectra xml files.

```
library(MsBackendHmdb)
fls <- dir("data/hmdb_all_spectra/", full.names = TRUE, pattern = "ms_ms")
hmdb <- Spectra(fls, source = MsBackend)
```

- Subset the ~ 500,000 spectra to those containing the precursor m/z.

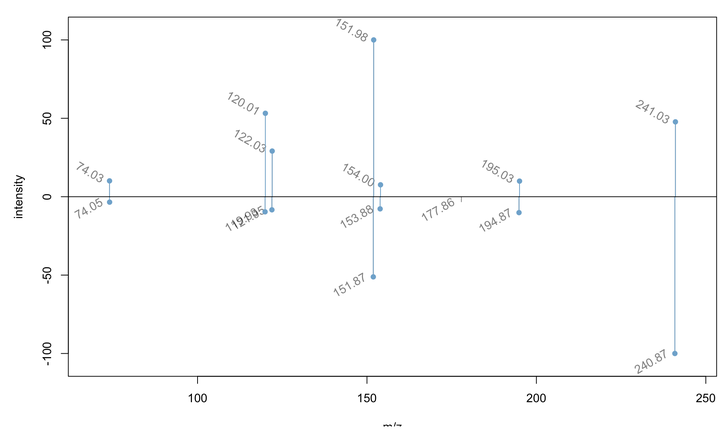
```
has_mz <- containsMz(hmdb, mz = mz, tolerance = 0.2)
hmdb_with_mz <- hmdb[has_mz]
```

- Compare the experimental spectra against the database.

```
res <- compareSpectra(sps_ce20, hmdb_with_mz, tolerance = 0.2)
```

- Highest similarity is 0.821. Plotting best matching spectra.

```
idx <- which(res == max(res), arr.ind = TRUE)
label_fun <- function(x) format(unlist(mz(x)), digits = 4)
plotSpectraMirror(sps_ce20[idx[1]], hmdb_with_mz[idx[2]], tolerance = 0.2,
                  labels = label_fun, labelPos = 2, labelSrt = -30)
```



- Best match is with HMDB0000192 (L-Cystine).

Export in MGF format

- Add annotation and collision energy to the spectra.
- Export to a file in mascot generic format (MGF) using `MsBackendMgf`.

```
sps_ce20$hmdb_id <- hmdb_with_mz[idx[2]]$compound_id
sps_ce20$collisionEnergy <- 20

library(MsBackendMgf)
export(sps_ce20, backend = MsBackendMgf(), file = "Cystin_ce20.mgf")
```

Conclusion and Outlook

- `Spectra` provides a flexible and expandable infrastructure for MS data in R.
- Enables seamless integration of MS data from different data sources or formats.
- Allows elegant MS data handling and analysis in R.
- Future backends will involve storage of data in SQL databases with possibility of remote access and eventually access to online spectral databases.