Initial directions for the `metabalize` package

Drew

November 19, 2013

1 Functions

The `metabalize` package currently exists in embryonic form. There are three functions:

- `read_mz_file`, which reads a .csv or .xlsx file generated by MAVEN, and returns the data frame containing the results. Currently this is a little silly, but the idea is to develop it in the future to create objects which are slightly less trivial. Note to self: converting m/z values from double to integer doesn't seem to help much - object size is reduced by half, and there's a performace gain of maybe 50% at lower numbers of peaks, but at higher number of peaks the performance gain goes away.
- `toy_deisotope`, a 'toy' version of the planned deisotoping function. Currently it identifies peaks that differ in median m/z by a specified amount (`mass_diff`), plus or minus a tolerance (`tolerance`), meaning that it can only identify one isotopomer. (The default is a difference of a single ^{13}C, +/- 2e-5).
- `toy_deisotope`This tests the *performance* of `toy_deisotope`(not its accuracy). The algorithm behind `toy_deisotope`scales as $O(n^2)$, i.e., the time it takes is proportional to the square of the number of peaks analyzed. I need to make the algorithm more efficient; standard testing will me to track those efforts.

2 Workflow

First, load a data set using `read_mz_file`. `"maven-output.csv"` contains a 193-peak data set of known compounds produced by MAVEN:

```
> mz_df <- read_mz_file("../data/maven-output.csv")
> nrow(mz_df)

[1] 193
```

Second, identify isotopomers. Currently `toy_deisotope`can only look for one isotopomer at a time (currently the defaults are set to a difference of a single ^{13}C with tolerance of 2×10^{-5} m/z). It will be straightforward to add multiple isotopomers; the tricky thing is finding a good search algorithm.

```
> is_short <- toy_deisotope(mz_df)
> is_short

[1] mz1 mz2
<0 rows> (or 0-length row.names)
```

OK, this data set doesn't seem to have any isotopomers, assuming I've set the tolerance correctly. Have they been removed already?

Let's check a dataset of unknown peaks to see whether we can find some isotopomers. We don't want to check the whole dataset for istopomers - this creates a 12 Gb memory object, as currently written - but we can check the first thousand elements for isotopes.

```
> unknowns <- read_mz_file("../data/maven-output-unknowns.csv")
> system.time({
+   unknown_isotopes <- toy_deisotope(unknowns[1:1000, ])})

   user  system elapsed
  1.109   0.095   1.226

> print(unknown_isotopes)

            mz1      mz2
765058 87.00810 88.01147
852077 89.02377 90.02712
852078 89.02377 90.02712
```

So this says that the peak at 87.00810 has an isotopomer at 88.01147, and the peak at 89.02377 has two isotopomers at 90.02712 (which have different retention times.)

3 Performance

```
> subset_lengths=seq(from=500, to=4000, by=250)
> system.time({
+   performance <- test_toy_deisotope(unknowns, subset_lengths=subset_lengths)
+ })

[1] 0.095 0.055 0.152
[1] 0.365 0.054 0.431
[1] 0.768 0.088 0.858
[1] 1.049 0.145 1.208
```

```
[1] 1.225 0.197 1.442
[1] 1.490 0.260 1.765
[1] 1.982 0.375 2.386
[1] 2.275 0.499 2.798
[1] 2.561 0.554 3.150
[1] 3.105 0.664 3.798
[1] 3.713 0.917 4.680
[1] 4.364 1.193 5.768
[1] 4.853 1.593 7.002
[1]  5.686  3.347 15.660
[1]  6.713  3.350 27.892
   user  system elapsed
 41.422  15.213  82.264
```

```
> plot(performance[ , "subset_lengths"], performance[, "elapsed"], xlab="number of elements"
```