Analysis of desialylated and sialylated spectra of Myozyme®

Code for data analysis and figures

Plots were created in R using packages from tidyverse and janitor.

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.0     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.1     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
v purrr     1.0.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(janitor)
```

```

Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

Loading data

Load intact sialylated Myozyme glycoform annotations.

```
mofi_results_sial <-
  read_csv(
    "data/141222_Intact_Myozyme_annotations_v3.csv",
    name_repair = "minimal",
    skip = 104,
    col_select = c(1:26)
  ) %>%
  clean_names() %>%
  separate(
    id,
    into = c("peak_id", "hit_id", "perm_id"),
    sep = "-",
    remove = FALSE
  )
```

```
Rows: 1048471 Columns: 26
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (8): ID, N177, N334, N414, N596, N826, N84, N869
dbl (18): Exp. Mass, %, Hit, Hit Score, # Perms, Theo. Mass, Da, ppm, Hex, H...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
Warning: Expected 3 pieces. Missing pieces filled with `NA` in 194 rows [32853, 32922,
33094, 33102, 33110, 33131, 33143, 33333, 33337, 33455, 33457, 33478, 33504,
33659, 33663, 33664, 33681, 33709, 33766, 33794, ...].
```

Load experimentally desialylated Myozyme glycoform annotations.

```
mofi_results_desial <-
  read_csv(
    "data/111022_Hits MoFi N-glycan + acetylated 1%Cutoff_cysteinyl_Sialidase.csv",
    skip = 72
  ) %>%
  clean_names() %>%
  separate(
    id,
    into = c("peak_id", "hit_id", "perm_id"),
    sep = "-",
    remove = FALSE
  )
```

```
Rows: 28520 Columns: 26
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (8): ID, N177, N334, N414, N596, N826, N84, N869
dbl (18): Exp. Mass, %, Hit, Hit Score, # Perms, Theo. Mass, Da, ppm, Hex, H...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Defining desialylate function

find_interval_mean function finds the mean of an interval given as string e.g. "(101,137]" -> 118

```
find_interval_mean <- function(interval) {
  span <- str_match(interval, "\\((.*),(.*)\\]")
  (as.numeric(span[,2]) + as.numeric(span[,3])) / 2
}
```

desialylate function performs computational calculations to desialylate the sialylated Myozyme glycoform annotations masses, to compare the computationally desialylated masses to the masses present in the experimentally desialylated ones, and to filter the hit score. Finally, it calculates relative abundances for the computationally desialylated, filtered spectrum.

```
desialylate <- function(peaks_sial,
                        peaks_desial_exp,
                        mass_desial_peak,
                        filter_peaks = TRUE,
                        filter_hit_score = TRUE,
                        hit_score_cutoff = 0.01,
                        mass_tolerance = 5) {
  MASS_NEU5AC <- 291.256  # mass of a sialic acid
  MASS_AC <- 42.0106  # acetylation

  # experimental desialylation
```

```r
peaks_desial_comp <-
  peaks_sial %>%
  mutate(mass_desial = exp_mass - MASS_NEU5AC * neu5ac - MASS_AC * acetyl)

# if desired, only keep MoFi hits where the computationally desialylated mass
# corresponds to a peak in the experimentally desialylated spectrum (within
# the given mass_tolerance)
if (filter_peaks) {
  peaks_desial_comp <-
    map_dfr(
      unique(peaks_desial_exp$exp_mass),
      function(mass) {
        peaks_desial_comp %>%
          filter(abs(mass_desial - mass) < mass_tolerance)
      }
    ) %>%
    distinct(id, .keep_all = TRUE) %>%
    group_by(peak_id) %>%
    mutate(hit_score = hit_score / sum(hit_score) * 100) %>%
    ungroup()
}

# if desired, only keep MoFi hits where hit_score > hit_score_cutoff
if (filter_hit_score) {
  peaks_desial_comp <-
    peaks_desial_comp %>%
    filter(hit_score > hit_score_cutoff) %>%
    group_by(peak_id) %>%
    mutate(hit_score = hit_score / sum(hit_score) * 100) %>%
    ungroup()
}

n_bins <- round(
  (max(peaks_desial_comp$mass_desial) - min(peaks_desial_comp$mass_desial))
  / mass_tolerance
)

peaks_desial_comp %>%
  mutate(mass = mass_desial %>% cut(n_bins) %>% find_interval_mean()) %>%
  group_by(mass) %>%
  summarise(intensity = sum(percent)) %>%
    mutate(intensity = intensity / max(intensity) * 100)

}
```

Running analysis with desialylate function

1. The intact glycoform annotations are computationally desialylated to obtain a desialylated *in silico* spectrum of Myozyme. No filtering of peaks and no filtering of hit score is performed. Finally, relative abundances of the *in silico* desialylated masses are calculated.

```
df_desial <- desialylate(
  mofi_results_sial,
  mofi_results_desial,
  filter_hit_score = FALSE,
  filter_peaks = FALSE
)
```

2. The *in silico* desialylated masses are filtered based on their correspondence with the experimentally desialylated masses and the hit scores are normalized to 100%. No filtering of hit score is performed. Finally, relative abundances of the filtered *in silico* desialylated masses are calculated.

```
df_desial_filtered <- desialylate(
  mofi_results_sial,
  mofi_results_desial,
  filter_hit_score = FALSE,
  filter_peaks = TRUE
)
```

3. The filtered, *in silico* desialylated masses are calculated as described in 1. and 2. Furthermore, the **hit scores** are filtered with a given cut-off value and afterwards normalized to 100%. In this step, both filtering of peaks and filtering of hit scores is performed. Finally, relative abundances of the resulting masses are calculated.

```
df_desial_filtered_cutoff <- desialylate(
  mofi_results_sial,
  mofi_results_desial,
  filter_hit_score = TRUE,
  filter_peaks = TRUE,
  hit_score_cutoff = 0.01
)
```

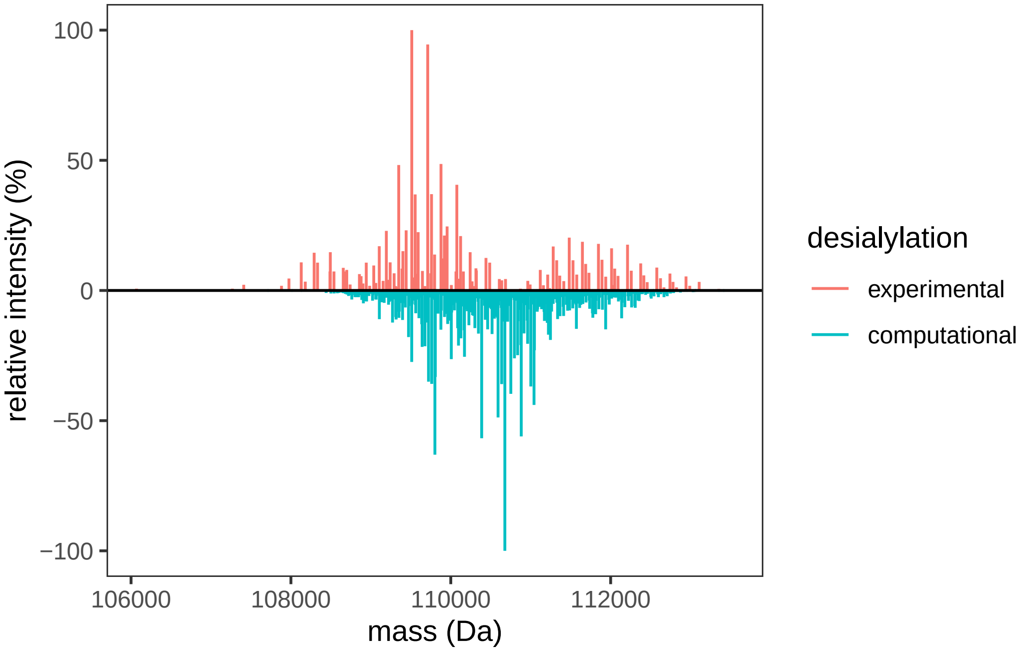
Plotting data

plot_spectrum function plots relative abundances of the experimentally desialylated spectrum and of the *in silico* desialylated Myozyme spectrum at a different step of the analysis (1., 2., or 3.).

```
plot_spectrum <- function(computational_data) {
  bind_rows(
    experimental =
      mofi_results_desial %>%
      group_by(peak_id) %>%
      summarise(across(c(exp_mass, percent), first)) %>%
      select(mass = exp_mass, intensity = percent),
    computational =
      computational_data %>%
      mutate(intensity = intensity * -1),
    .id = "desialylation"
  ) %>%
    mutate(desialylation = fct_rev(desialylation)) %>%
    ggplot(aes(mass, 0, xend = mass, yend = intensity)) +
    geom_segment(aes(color = desialylation)) +
    geom_hline(yintercept = 0) +
    xlab("mass (Da)") +
    ylab("relative intensity (%)") +
    theme_bw() +
    theme(panel.grid = element_blank())
}
```

Plots spectrum of relative abundances of the experimentally desialylated Myozyme (experimental) and that of the *in silico* desialylated Myozyme after step 1. of the analysis (computational).

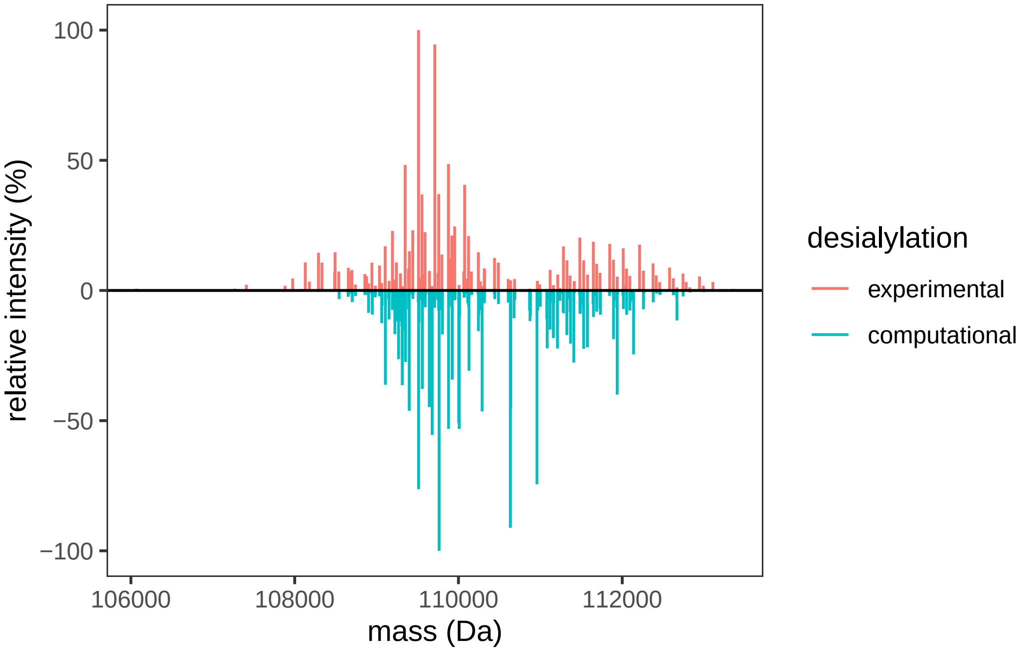
```
plot_spectrum(df_desial)
```



```
#ggsave("plots/desialylated_spectra_experimental_vs_computational_unfiltered.pdf")
```

Plots spectrum of relative abundances of the experimentally desialylated Myozyme (experimental) and that of the *in silico* desialylated Myozyme after steps 1. and 2. of the analysis (computational).

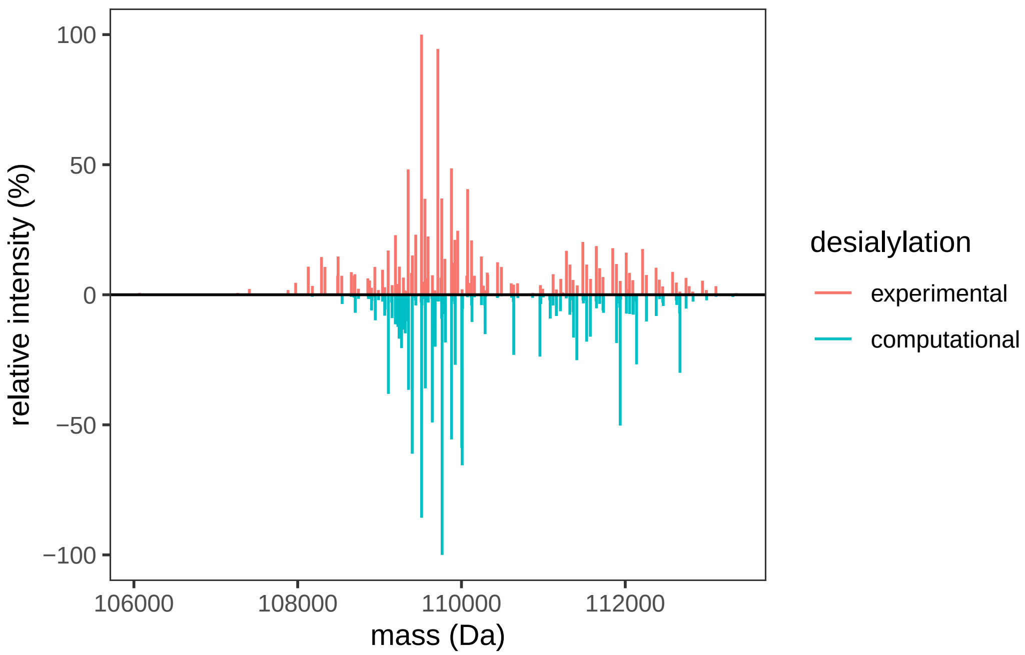
```
plot_spectrum(df_desial_filtered)
```



```
#ggsave("plots/desialylated_spectra_experimental_vs_computational_filtered.pdf")
```

Plots spectrum of relative abundances of the experimentally desialylated Myozyme (experimental) and that of the *in silico* desialylated Myozyme after steps 1., 2., and 3. of the analysis (computational).

```
plot_spectrum(df_desial_filtered_cutoff)
```



```
#ggsave("plots/desialylated_spectra_experimental_vs_computational_filtered_cutoff0.01.pdf"
```