The Growing Landscape of Protein Modifications

E. Keith Keenan & Matthew D. Hirschey

September 03, 2019

Contents

This is an R Markdown notebook accompanying a review on protein modifications. When you execute code within the notebook, the results appear beneath the code and Figures will be save to the working directory.

Load libraries

```r
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------------- tidy
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ---------------------------------------------------------------------------- tidyverse_c
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```r
library(viridis)
```

```
## Loading required package: viridisLite
```

```r
library(XML)
library(feather)
library(rmarkdown)
library(beepr) #long analysis; get some coffee, and comeback when ready

#clear environment
rm(list=ls())

#print Session information for provenance and reproducibility
utils:::print.sessionInfo(sessionInfo()[-8])
```

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] beepr_1.3         rmarkdown_1.15    feather_0.3.3
##  [4] XML_3.98-1.20     viridis_0.5.1     viridisLite_0.3.0
##  [7] janitor_1.2.0     forcats_0.4.0     stringr_1.4.0
## [10] dplyr_0.8.3       purrr_0.3.2       readr_1.3.1
## [13] tidyr_0.8.3       tibble_2.1.3      ggplot2_3.2.1
## [16] tidyverse_1.2.1
```

```r
#You can remove an item from sessionInfo(), which is a list with a class attribute, by printing the res

#Set theme
theme_set(theme_light())
```

Figure 2

Overall goal is to quanitfy known landscape of protein amino acids. Chose to get data from Uniprot, as a comprehensive and validated resouce containing data for human proteins.

```r
ptm_raw <- read_tsv("https://www.uniprot.org/docs/ptmlist.txt", col_names = FALSE, skip = 48)
```

```
## Parsed with column specification:
```

```
## cols(
##   X1 = col_character()
## )
```

```
#skip 48 first lines which contain data file dictionary

#URL points to a datafile, to increase reproducibility; datafile is also downloaded 12/21/2018 and save
#alt
#ptm_raw <- read_tsv("data/ptmlist.txt", col_names = FALSE, skip = 48)

#make working df
ptm <- ptm_raw %>%
  separate(X1, c("key", "value"), sep = 3) %>%
  mutate(id = if_else(grepl("ID", key), value, NA_character_)) %>% #must call NA_char so that fill fxn
  fill(id) #need fill fxn to populate ids across all observations, so that spread can work

#clean more
ptm$key <- str_trim(ptm$key, side = "right") #use stringr pkg to remove white space *janitor works on c

#drop rows, duplicate rows are causing problems with spread, and don't need them
ptm <- ptm %>%
  filter(!key %in% c("//", "TR", "DR", "---"))

#This is code I used to ensure that there were no duplicates
#ptm <- ptm %>%
#  unite(key_id, c("key", "id"), sep = "_", remove = FALSE)
#ptm_dup <- get_dupes(ptm, key_id)
#I check the ptm_dup df and made to sure to drop the keys that had more than one entry (immediate prece

#spread data
ptm <- ptm %>%
  spread(key, value) #not clever names, but appropriate

#gives a tibble of 645 observations, therefore 645 unique PTMs

#double check to see no duplicates
#get_dupes(ptm, id)

#more cleaning steps
ptm$MM <- as.numeric(ptm$MM)
ptm$MA <- as.numeric(ptm$MA)
ptm$KW <- str_replace(ptm$KW, "\\.", "") #need two \\ to mean literal "."
ptm$KW <- as.factor(ptm$KW)
ptm$FT <- str_trim(ptm$FT, side = "left") #use stringr pkg to remove white space
ptm$TG <- str_trim(ptm$TG, side = "left")
ptm$TG <- str_replace(ptm$TG, "\\.", "") #need two \\ to mean literal "."
ptm$KW <- fct_explicit_na(ptm$KW, na_level = "Other") #get rid of NAs in KW by making a factor
ptm <- ptm %>% select(-Cop, -Dis) #remove copyright and distribution columns

#a little bit of eda
count(ptm, FT, sort = TRUE)
```

```
## # A tibble: 4 x 2
##   FT            n
##   <chr>    <int>
## 1 MOD_RES    329
## 2 CROSSLNK   149
## 3 CARBOHYD   130
## 4 LIPID       41
```

```
#should I include crosslinks? Or just modifications?

ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  summarize(n = n())
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   500
```

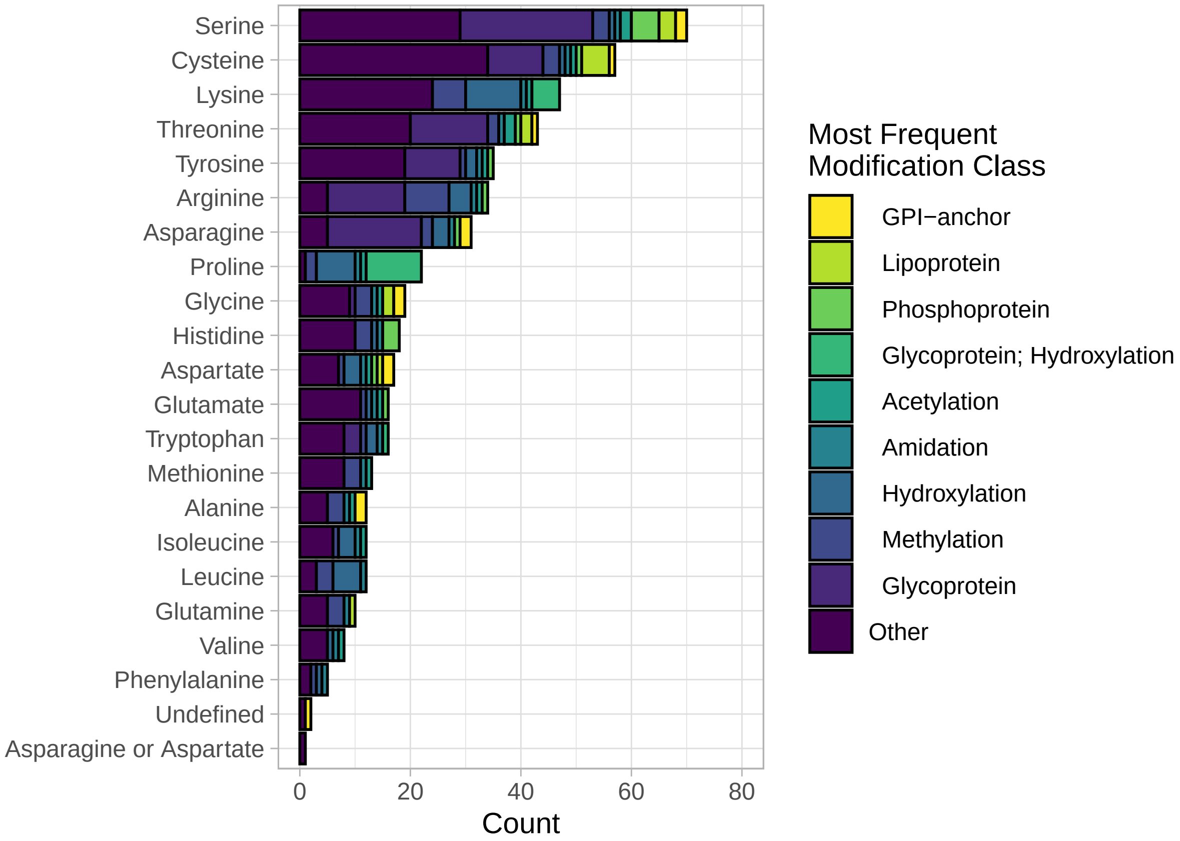
```
#This code snippet give me the total number of unique modifications, with CROSSLINK removed; total is 4

count(ptm, KW, sort = TRUE) #number of modifications by keyword
```

```
## # A tibble: 59 x 2
##    KW                                  n
##    <fct>                           <int>
##  1 Other                             164
##  2 "  Glycoprotein"                   93
##  3 "  Methylation"                    51
##  4 "  Hydroxylation"                  45
##  5 "  Thioether bond"                 27
##  6 "  Isopeptide bond"                23
##  7 "  Amidation"                      20
##  8 "  Acetylation"                    16
##  9 "  Glycoprotein; Hydroxylation"    16
## 10 "  Phosphoprotein"                 15
## # ... with 49 more rows
```

```
#a lot of glycoproteins!

#ptm %>%
#  filter(FT == "MOD_RES") %>% #include only modified AAs, no cross links, no lipids, no glycoproteins?
#  count(TG, sort = TRUE) %>% #target (TG) is exactly what I need
#  mutate(TG = fct_reorder(TG, n)) %>%
#  ggplot(aes(TG, n)) +
#  geom_col() +
#  coord_flip() +
#  labs(x = "") +
#  expand_limits(y = 40)
#commented this out because it only includes modified AAs; not sure if this is useful

ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  count(TG, sort = TRUE)
```

```
## # A tibble: 22 x 2
##    TG             n
##    <chr>      <int>
##  1 Serine        70
##  2 Cysteine      57
##  3 Lysine        47
##  4 Threonine     43
##  5 Tyrosine      35
##  6 Arginine      34
##  7 Asparagine    31
##  8 Proline       22
##  9 Glycine       19
## 10 Histidine     18
## # ... with 12 more rows
```

```
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  ggplot() +
  geom_bar(aes(fct_rev(fct_infreq(TG, ordered = TRUE)), fill = fct_rev(fct_infreq(fct_lump(KW, 10)))), 
  coord_flip() +
  labs(x = "", y = "Count") +
  expand_limits(y = 80) +
  scale_fill_viridis(discrete = TRUE, direction = -1, option = "viridis", name = "Most Frequent \nModif
  NULL
```



```
#save plot
ggsave("output/fig2.pdf", plot = last_plot(), dpi = 600)
```

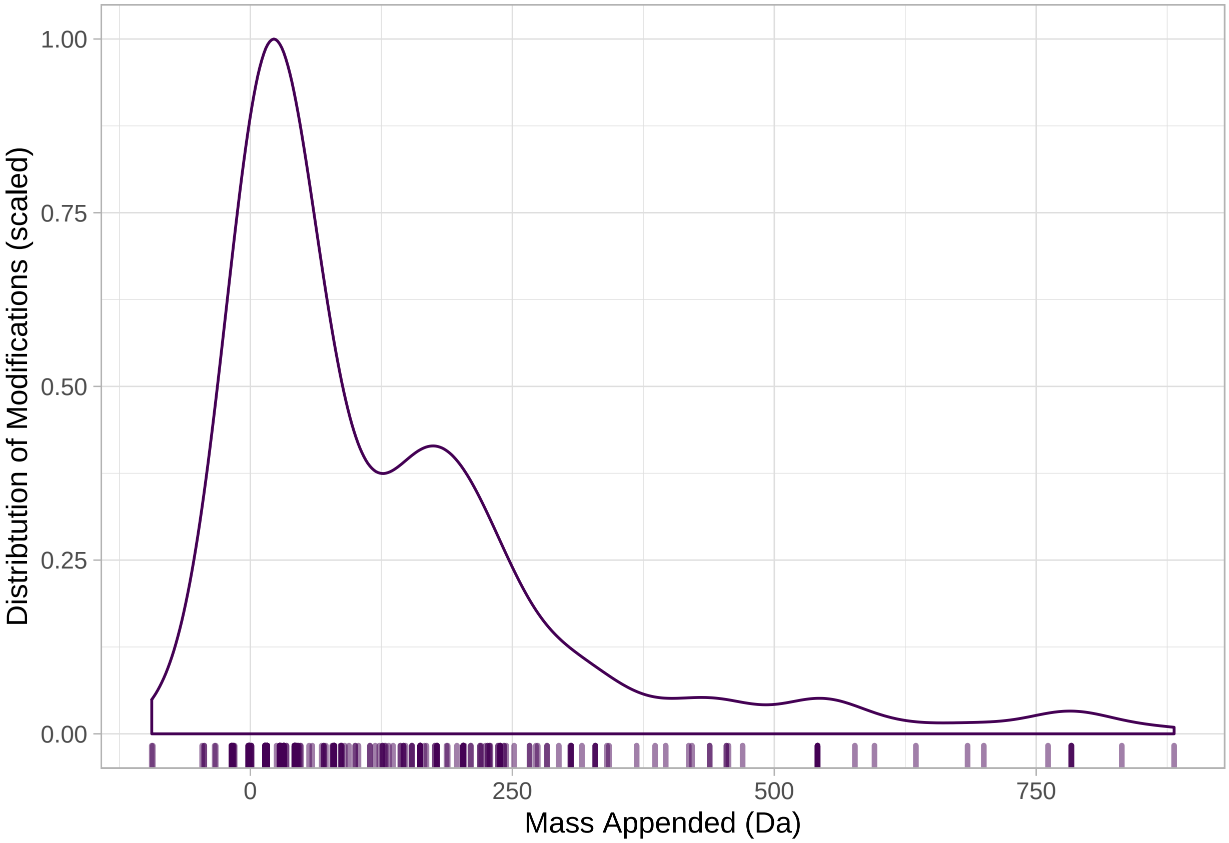
```
## Saving 6.5 x 4.5 in image
```

Figure 3

Goal is to determine how these modifications are distributed; thought it'd be interesting to visualize by average added mass (MA) to a protein, with several small changes in molecular mass, with some very large additions of mass

```
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  ggplot() +
  #geom_point(aes(x = MA, y =0, color = fct_lump(KW, 0)), shape = "|", size = 15, alpha = 1/2) +
  geom_density(aes(x = MA, ..scaled.., color = fct_lump(KW,0))) +
  geom_rug(aes(x = MA, y = 0, color = fct_lump(KW,0)), sides = "b", alpha = 1/2, position = "jitter", s
  labs(x = "Mass Appended (Da)", y = "Distribtution of Modifications (scaled)") +
  scale_color_viridis(discrete = TRUE, direction = 1) +
  scale_y_continuous(limits = c(0,1)) +
  theme(legend.position = "") +
  NULL
```

```
## Warning: Removed 144 rows containing non-finite values (stat_density).
```



```
#save plot
ggsave("output/fig3.pdf", plot = last_plot(), width = 5, height = 5, dpi = 600)
```

```
## Warning: Removed 144 rows containing non-finite values (stat_density).
```

```
#the reason some average masses (MA) are so abundant is because you find the same modifications across 
#NB several glycans and lipids are variable masses, and therefore are entered as NA, so not reflected i

ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  count(KW, sort = TRUE)
```

```
## # A tibble: 50 x 2
##    KW                                n
##    <fct>                         <int>
##  1 "  Glycoprotein"                 93
##  2 Other                            80
##  3 "  Methylation"                  50
##  4 "  Hydroxylation"                45
##  5 "  Amidation"                    20
##  6 "  Acetylation"                  16
##  7 "  Glycoprotein; Hydroxylation"  16
##  8 "  Phosphoprotein"               15
##  9 "  Lipoprotein"                  14
## 10 "  GPI-anchor"                   13
## # ... with 40 more rows
```

AA Analyses

Lysine Analysis

In this code chunk, the goal is to count and summarize lysine modifications.

```
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(TG == "Lysine") %>%
  count(ID, sort = TRUE)
```

```
## # A tibble: 47 x 2
##    ID                              n
##    <chr>                       <int>
##  1 "  (3S)-3-hydroxylysine"        1
##  2 "  (5R)-5-hydroxylysine"        1
##  3 "  (5S)-5-hydroxylysine"        1
##  4 "  4-hydroxylysine"             1
##  5 "  4,5-dihydroxylysine"         1
##  6 "  5-hydroxylysine"             1
##  7 "  Allysine"                    1
##  8 "  Hypusine"                    1
##  9 "  Lysine amide"                1
## 10 "  Lysine derivative"           1
## # ... with 37 more rows
```

```
#code chunk to make a tibble that is easy to view all attributes; no need to save as an object in envir
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(TG == "Lysine") %>%
  arrange(MA) #sorts by mass, low to high
```

```
## # A tibble: 47 x 12
##    id    AC    CF    FT    ID    KW    LC       MA      MM PA    PP    TG
##    <chr> <chr> <chr> <chr> <chr> <fct> <chr> <dbl>   <dbl> <chr> <chr> <chr>
##  1 "   A~ "   P~ "   H~ MOD_~ "   A~ Other "   E~ -1.03 -1.03  "   A~ "   A~ Lysi~
##  2 "   L~ "   P~ "   H~ MOD_~ "   L~ "   A~ "   E~ -0.98 -0.984 "   A~ "   C~ Lysi~
##  3 "   L~ "   P~ "   C~ MOD_~ "   L~ "   M~ "   I~ 14.0  14.0   "   A~ "   C~ Lysi~
##  4 "   N~ "   P~ "   C~ MOD_~ "   N~ "   M~ "   I~ 14.0  14.0   "   A~ "   A~ Lysi~
##  5 "   (~ "   P~ "   O~ MOD_~ "   (~ "   H~ "   E~ 16    16.0   "   A~ "   A~ Lysi~
##  6 "   (~ "   P~ "   O~ MOD_~ "   (~ "   H~ "   E~ 16    16.0   "   A~ "   A~ Lysi~
##  7 "   (~ "   P~ "   O~ MOD_~ "   (~ "   H~ "   E~ 16    16.0   "   A~ "   A~ Lysi~
##  8 "   4~ "   P~ "   O~ MOD_~ "   4~ "   H~ "   E~ 16    16.0   "   A~ "   A~ Lysi~
##  9 "   5~ "   P~ "   O~ MOD_~ "   5~ "   H~ "   E~ 16    16.0   "   A~ "   A~ Lysi~
## 10 "   N~ "   P~ "   C~ MOD_~ "   N~ "   F~ "   E~ 28.0  28.0   "   A~ "   A~ Lysi~
## # ... with 37 more rows
```

Cysteine Analysis

In this code chunk, the goal is to count and summarize cysteine modifications. Counted 57 (as of Feb 2019), however does not include 3 published modifiations: succination, 2,3-dicarboxylpropylation (i.e. itaconylation), or s-acetylation, so OK to conclude 60, at least.

```
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(TG == "Cysteine") %>%
  count(ID, sort = TRUE)
```

```
## # A tibble: 57 x 2
##    ID                                                        n
##    <chr>                                                 <int>
##  1 "   2-(S-cysteinyl)pyruvic acid O-phosphothioketal"       1
##  2 "   2,3-didehydroalanine (Cys)"                           1
##  3 "   3-oxoalanine (Cys)"                                   1
##  4 "   ADP-ribosylcysteine"                                  1
##  5 "   Blocked amino end (Cys)"                              1
##  6 "   Cyclo[(prolylserin)-O-yl] cysteinate"                 1
##  7 "   Cysteine amide"                                       1
##  8 "   Cysteine derivative"                                  1
##  9 "   Cysteine methyl disulfide"                            1
## 10 "   Cysteine methyl ester"                                1
## # ... with 47 more rows
```

```
#code chunk to make a tibble that is easy to view all attributes; no need to save as an object in envir
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(TG == "Cysteine") %>%
  arrange(MA) #sorts by mass, low to high
```

```
## # A tibble: 57 x 12
##    id    AC    CF    FT    ID    KW    LC        MA      MM PA    PP
##    <chr> <chr> <chr> <chr> <chr> <fct> <chr>  <dbl>   <dbl> <chr> <chr>
##  1 "  2~ "  P~ "  H~ MOD_~ "  2~ Other "  I~ -34.1  -34.0   "  A~ "  P~
##  2 "  P~ "  P~ "  H~ MOD_~ "  P~ "  P~ "  I~ -33.1  -33.0   "  A~ "  N~
##  3 "  3~ "  P~ "  H~ MOD_~ "  3~ Other "  E~ -18.1  -18.0   "  A~ "  A~
##  4 "  D~ "  P~ "  O~ MOD_~ "  D~ "  D~ "  E~ -16.1  -16.0   "  A~ "  P~
##  5 "  C~ "  P~ "  H~ MOD_~ "  C~ "  A~ "  E~  -0.98  -0.984 "  A~ "  C~
##  6 "  C~ "  P~ "  C~ MOD_~ "  C~ "  M~ "  I~  14.0   14.0   "  A~ "  C~
##  7 "  S~ "  P~ "  C~ MOD_~ "  S~ "  M~ "  I~  14.0   14.0   "  A~ "  A~
##  8 "  C~ "  P~ "  O~ MOD_~ "  C~ "  O~ "  I~  16     16.0   "  A~ "  A~
##  9 "  S~ "  P~ "  C~ MOD_~ "  S~ Other "  I~  25.0   25.0   "  A~ "  A~
## 10 "  S~ "  P~ "  H~ MOD_~ "  S~ "  S~ "  I~  29     29.0   "  A~ "  A~
## # ... with 47 more rows, and 1 more variable: TG <chr>
```

Serine Analysis

In this code chunk, the goal is to count and summarize serine modifications. Counted 70 (as of Feb 2019).

```
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(TG == "Serine") %>%
  count(ID, sort = TRUE)
```

```
## # A tibble: 70 x 2
##    ID                                      n
##    <chr>                               <int>
##  1 "  2,3-didehydroalanine (Ser)"          1
##  2 "  3-oxoalanine (Ser)"                  1
##  3 "  ADP-ribosylserine"                   1
##  4 "  Aminomalonic acid (Ser)"             1
##  5 "  Blocked amino end (Ser)"             1
##  6 "  D-alanine (Ser)"                     1
##  7 "  D-serine (Ser)"                      1
##  8 "  FMN phosphoryl serine"               1
##  9 "  GPI-anchor amidated serine"          1
## 10 "  GPI-like-anchor amidated serine"     1
## # ... with 60 more rows
```

```
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(TG == "Threonine") %>%
  count(ID, sort = TRUE)
```

```
## # A tibble: 43 x 2
##    ID                                    n
##    <chr>                             <int>
##  1 "  (E)-2,3-didehydrobutyrine"         1
##  2 "  (Z)-2,3-didehydrobutyrine"         1
##  3 "  1-amino-2-propanone"               1
##  4 "  2-oxobutanoic acid"                1
##  5 "  2,3-didehydrobutyrine"             1
##  6 "  Blocked amino end (Thr)"           1
##  7 "  D-threonine"                       1
##  8 "  Decarboxylated threonine"          1
##  9 "  FMN phosphoryl threonine"          1
## 10 "  GPI-anchor amidated threonine"     1
## # ... with 33 more rows
```

```r
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(TG == "Tyrosine") %>%
  count(ID, sort = TRUE)
```

```
## # A tibble: 35 x 2
##    ID                                       n
##    <chr>                                <int>
##  1 "  (E)-2,3-didehydrotyrosine"            1
##  2 "  (Z)-2,3-didehydrotyrosine"            1
##  3 "  2,3-didehydroalanine (Tyr)"           1
##  4 "  2,3-didehydrotyrosine"                1
##  5 "  2',4',5'-topaquinone"                 1
##  6 "  3'-nitrotyrosine"                     1
##  7 "  3',4'-dihydroxyphenylalanine"         1
##  8 "  3',4',5'-trihydroxyphenylalanine"     1
##  9 "  ADP-ribosyltyrosine"                  1
## 10 "  Diiodotyrosine"                       1
## # ... with 25 more rows
```

```r
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(TG == "Serine") %>%
  filter(str_detect(CF, "P")) %>%
  count(ID, sort = TRUE)
```

```
## # A tibble: 13 x 2
##    ID                                                    n
##    <chr>                                             <int>
##  1 "  ADP-ribosylserine"                                 1
##  2 "  FMN phosphoryl serine"                             1
##  3 "  O-(2-aminoethylphosphoryl)serine"                  1
##  4 "  O-(2-cholinephosphoryl)serine"                     1
##  5 "  O-(pantetheine 4'-phosphoryl)serine"               1
##  6 "  O-(phosphoribosyl dephospho-coenzyme A)serine"     1
##  7 "  O-(sn-1-glycerophosphoryl)serine"                  1
##  8 "  O-AMP-serine"                                      1
##  9 "  O-linked (GlcNAc1P) serine"                        1
## 10 "  O-linked (GlcNAc6P) serine"                        1
## 11 "  O-linked (Man1P) serine"                           1
## 12 "  O-UMP-serine"                                      1
## 13 "  Phosphoserine"                                     1
```

```r
#13 serine modifications contain phosphate (12 carbon-phosphate, 1 phosphate only)
```

```
#code chunk to make a tibble that is easy to view all attributes; no need to save as an object in envir
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(TG == "Serine") %>%
  arrange(MA) #sorts by mass, low to high
```

```
## # A tibble: 70 x 12
##    id    AC    CF    FT    ID    KW    LC        MA      MM PA    PP
##    <chr> <chr> <chr> <chr> <chr> <fct> <chr>  <dbl>   <dbl> <chr> <chr>
##  1 "  2~ "  P~ "  H~ MOD_~ "  2~ Other "  I~ -18.0  -18.0   "  A~ "  P~
##  2 "  P~ "  P~ "  H~ MOD_~ "  P~ "  P~ "  I~ -17.0  -17.0   "  A~ "  N~
##  3 "  D~ "  P~ "  O~ MOD_~ "  D~ "  D~ "  E~ -16    -16.0   "  A~ "  P~
##  4 "  L~ "  P~ "  H~ MOD_~ "  L~ Other "  E~ -15.0  -15.0   "  A~ "  N~
##  5 "  3~ "  P~ "  H~ MOD_~ "  3~ Other "  E~  -2.02  -2.02  "  A~ "  A~
##  6 "  S~ "  P~ "  H~ MOD_~ "  S~ "  A~ "  E~  -0.98 -0.984  "  A~ "  C~
##  7 "  A~ "  P~ "  H~ MOD_~ "  A~ Other "  E~  14.0   14.0    "  A~ "  A~
##  8 "  N~ "  P~ "  C~ MOD_~ "  N~ "  M~ "  I~  14.0   14.0    "  A~ "  N~
##  9 "  N~ "  P~ "  C~ MOD_~ "  N~ "  M~ "  I~  28.0   28.0    "  A~ "  N~
## 10 "  N~ "  P~ "  C~ MOD_~ "  N~ "  A~ "  I~  42.0   42.0    "  A~ "  N~
## # ... with 60 more rows, and 1 more variable: TG <chr>
```

Phenylalanine Analysis

In this code chunk, the goal is to count and summarize phenylalanine modifications. Counted 5 (as of Feb 2019); 1?

```
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(TG == "Phenylalanine") %>%
  arrange(MA) #sorts by mass, low to high
```

```
## # A tibble: 5 x 12
##   id     AC    CF    FT    ID    KW    LC       MA     MM PA    PP    TG
##   <chr>  <chr> <chr> <chr> <chr> <fct> <chr> <dbl>  <dbl> <chr> <chr> <chr>
## 1 "  Ph~ "  P~ "  H~ MOD_~ "  P~ "  A~ "  E~ -0.98 -0.984 "  A~ "  C~ Phen~
## 2 "  3-- "  P~ "  H~ MOD_~ "  3~ Other "  E~  0.98  0.984 "  A~ "  N~ Phen~
## 3 "  N-- "  P~ "  C~ MOD_~ "  N~ "  M~ "  E~ 14.0  14.0   "  A~ "  N~ Phen~
## 4 "  3-- "  P~ "  O~ MOD_~ "  3~ "  H~ "  E~ 16    16.0   "  A~ "  A~ Phen~
## 5 "  D-- "  P~ <NA>  MOD_~ "  D~ "  D~ "  E~ NA    NA     "  A~ "  P~ Phen~
```

Protein Backbone Analysis

In this code chunk, the goal is to count and summarize backbone modifications. First look at backbone alone; next look at the part of the protein where these are ascribed; then look at distribution of all backbone modifications on amino acids (glycine is the most); but, these are all n- or c-term modifications; if you look at protien core modifications, these are all serine/threonine/tyrosine and cysteine.

```
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(str_detect(PA, "backbone")) %>%
  arrange(MA) #sorts by mass, low to high
```

```
## # A tibble: 131 x 12
##    id     AC    CF    FT    ID    KW    LC        MA    MM PA    PP    TG
##    <chr>  <chr> <chr> <chr> <chr> <fct> <chr> <dbl> <dbl> <chr> <chr> <chr>
##  1 "  2,~ "  P~ "  C~ MOD_~ "  2~ Other "  I~ -94.1 -94.0 "  A~ "  P~ Tyro~
##  2 "  Py~ "  P~ "  C~ MOD_~ "  P~ "  P~ "  I~ -93.1 -93.1 "  A~ "  N~ Tyro~
##  3 "  1-~ "  P~ "  C~ MOD_~ "  1~ Other "  E~ -46.0 -46.0 "  A~ "  C~ Thre~
##  4 "  De~ "  P~ "  C~ MOD_~ "  D~ Other "  E~ -44.0 -44.0 "  A~ "  C~ Thre~
##  5 "  2,~ "  P~ "  H~ MOD_~ "  2~ Other "  I~ -34.1 -34.0 "  A~ "  P~ Cyst~
##  6 "  Py~ "  P~ "  H~ MOD_~ "  P~ "  P~ "  I~ -33.1 -33.0 "  A~ "  N~ Cyst~
##  7 "  (E~ "  P~ "  H~ MOD_~ "  (~ Other "  I~ -18.0 -18.0 "  A~ "  P~ Thre~
##  8 "  (Z~ "  P~ "  H~ MOD_~ "  (~ Other "  E~ -18.0 -18.0 "  A~ "  P~ Thre~
##  9 "  2,~ "  P~ "  H~ MOD_~ "  2~ Other "  I~ -18.0 -18.0 "  A~ "  P~ Seri~
## 10 "  2,~ "  P~ "  H~ MOD_~ "  2~ Other "  E~ -18.0 -18.0 "  A~ "  P~ Thre~
## # ... with 121 more rows
```

```r
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(str_detect(PA, "backbone")) %>%
  count(PA, sort = TRUE)
```

```
## # A tibble: 1 x 2
##   PA                               n
##   <chr>                        <int>
## 1 "  Amino acid backbone."   131
```

```r
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(str_detect(PA, "backbone")) %>%
  count(PP, sort = TRUE)
```

```
## # A tibble: 3 x 2
##   PP                      n
##   <chr>               <int>
## 1 "  N-terminal."      54
## 2 "  C-terminal."      51
## 3 "  Protein core."    26
```

```r
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(str_detect(PA, "backbone")) %>%
  count(TG, sort = TRUE)
```

```
## # A tibble: 21 x 2
##    TG              n
##    <chr>       <int>
##  1 Glycine        17
##  2 Serine         14
##  3 Cysteine       12
##  4 Threonine      12
##  5 Alanine        10
##  6 Tyrosine        8
##  7 Aspartate       6
##  8 Isoleucine     6
##  9 Methionine     6
## 10 Valine         6
## # ... with 11 more rows
```

```
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(str_detect(PA, "backbone")) %>%
  filter(TG == "Glycine")
```

```
## # A tibble: 17 x 12
##    id    AC    CF    FT    ID    KW    LC         MA       MM PA    PP
##    <chr> <chr> <chr> <chr> <chr> <fct> <chr>   <dbl>   <dbl> <chr> <chr>
##  1 "   1~ "   P~ "   O~ MOD_~ "   1~ Other "   I~  16.1   16.0   "   A~ "   P~
##  2 "   A~ "   P~ <NA>  MOD_~ "   A~ "   A~ "   I~  NA     NA     "   A~ "   C~
##  3 "   C~ "   P~ "   C~ LIPID "   C~ "   L~ "   E~ 369.   368.    "   A~ "   C~
##  4 "   C~ "   P~ "   C~ MOD_~ "   C~ Other "   I~ 103.   103.    "   A~ "   C~
##  5 "   G~ "   P~ "   H~ MOD_~ "   G~ "   A~ "   E~  -0.98  -0.984 "   A~ "   C~
##  6 "   G~ "   P~ "   C~ MOD_~ "   G~ "   N~ "   I~ 329.   329.    "   A~ "   C~
##  7 "   G~ "   P~ <NA>  LIPID "   G~ "   G~ "   E~  NA     NA     "   A~ "   C~
##  8 "   G~ "   P~ <NA>  LIPID "   G~ "   G~ "   E~  NA     NA     "   A~ "   C~
##  9 "   N~ "   P~ "   C~ MOD_~ "   N~ "   A~ "   I~   42.0   42.0   "   A~ "   N~
## 10 "   N~ "   P~ "   C~ MOD_~ "   N~ "   G~ "   E~ 176.   176.    "   A~ "   N~
## 11 "   N~ "   P~ "   C~ MOD_~ "   N~ "   F~ "   E~   28.0   28.0   "   A~ "   N~
## 12 "   N~ "   P~ "   C~ MOD_~ "   N~ "   M~ <NA>    14.0   14.0   "   A~ "   N~
## 13 "   N~ "   P~ "   C~ LIPID "   N~ "   M~ "   I~ 210.   210.    "   A~ "   N~
## 14 "   N~ "   P~ "   C~ LIPID "   N~ "   P~ "   I~ 238.   238.    "   A~ "   N~
## 15 "   N~ "   P~ "   C~ MOD_~ "   N~ "   M~ <NA>    28.0   28.0   "   A~ "   N~
## 16 "   N~ "   P~ "   C~ MOD_~ "   N~ "   M~ <NA>    43.1   43.1   "   A~ "   N~
## 17 "   P~ "   P~ "   C~ LIPID "   P~ "   L~ "   I~ 700.   700.    "   A~ "   C~
## # ... with 1 more variable: TG <chr>
```

```
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(str_detect(PA, "backbone")) %>%
  filter(str_detect(PP, "core")) %>%
  count(TG, sort = TRUE)
```

```
## # A tibble: 15 x 2
##    TG                n
##    <chr>         <int>
##  1 Threonine         4
##  2 Tyrosine          4
##  3 Serine            3
##  4 Cysteine          2
##  5 Isoleucine        2
##  6 Valine            2
##  7 Alanine           1
##  8 Asparagine        1
##  9 Aspartate         1
## 10 Glutamine         1
## 11 Glycine           1
## 12 Leucine           1
## 13 Methionine        1
## 14 Phenylalanine     1
## 15 Tryptophan        1
```

```
ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(str_detect(PA, "backbone")) %>%
  filter(str_detect(PP, "core")) %>%
  arrange(TG)
```

```
## # A tibble: 26 x 12
##    id    AC    CF    FT    ID    KW    LC        MA     MM PA    PP   
##    <chr> <chr> <chr> <chr> <chr> <fct> <chr>   <dbl>  <dbl> <chr> <chr>
##  1 "  D~ "  P~ <NA>  MOD_~ "  D~ "  D~ "  E~  NA     NA    "  A~ "  P~
##  2 "  D~ "  P~ <NA>  MOD_~ "  D~ "  D~ "  E~  NA     NA    "  A~ "  P~
##  3 "  (~ "  P~ "  H~ MOD_~ "  (~ Other "  I~  -2.02  -2.02 "  A~ "  P~
##  4 "  2~ "  P~ "  H~ MOD_~ "  2~ Other "  I~ -34.1  -34.0  "  A~ "  P~
##  5 "  D~ "  P~ "  O~ MOD_~ "  D~ "  D~ "  E~ -16.1  -16.0  "  A~ "  P~
##  6 "  2~ "  P~ "  C~ MOD_~ "  2~ "  M~ "  I~  14.0   14.0  "  A~ "  P~
##  7 "  1~ "  P~ "  O~ MOD_~ "  1~ Other "  I~  16.1   16.0  "  A~ "  P~
##  8 "  D~ "  P~ <NA>  MOD_~ "  D~ "  D~ "  E~  NA     NA    "  A~ "  P~
##  9 "  L~ "  P~ <NA>  MOD_~ "  L~ Other "  E~  NA     NA    "  A~ "  P~
## 10 "  D~ "  P~ <NA>  MOD_~ "  D~ "  D~ "  E~  NA     NA    "  A~ "  P~
## # ... with 16 more rows, and 1 more variable: TG <chr>
```

Figure 4

In this figure, the goal is to determine how many acyl-CoA species have been measured

```
#reload
load("data/proteins_raw.Rda")
load("data/metabolites_raw.Rda")
```

```
#as.X
proteins_raw <- as_tibble(proteins_raw)
metabolites_raw <- as_tibble(metabolites_raw)
metabolites_raw$average_molecular_weight <- as.numeric(metabolites_raw$average_molecular_weight)

#clean
metabolites <- metabolites_raw %>%
  select(one_of("accession", "name", "average_molecular_weight", "chemical_formula", "smiles", "normal_
  clean_names() %>%
  remove_empty("rows")
```

```
#count CoAs
CoA <- metabolites %>%
  filter(str_detect(name, "CoA")) %>%
  mutate(average_molecular_weight_noCoA = round(average_molecular_weight - 767.534, 2)) %>%  #substract
  arrange(average_molecular_weight_noCoA)

CoA <- CoA %>% #number of carbons
  mutate(carbon_num = str_extract(chemical_formula, "C\\d+")) %>% #extract C then digit, then one or mo
```

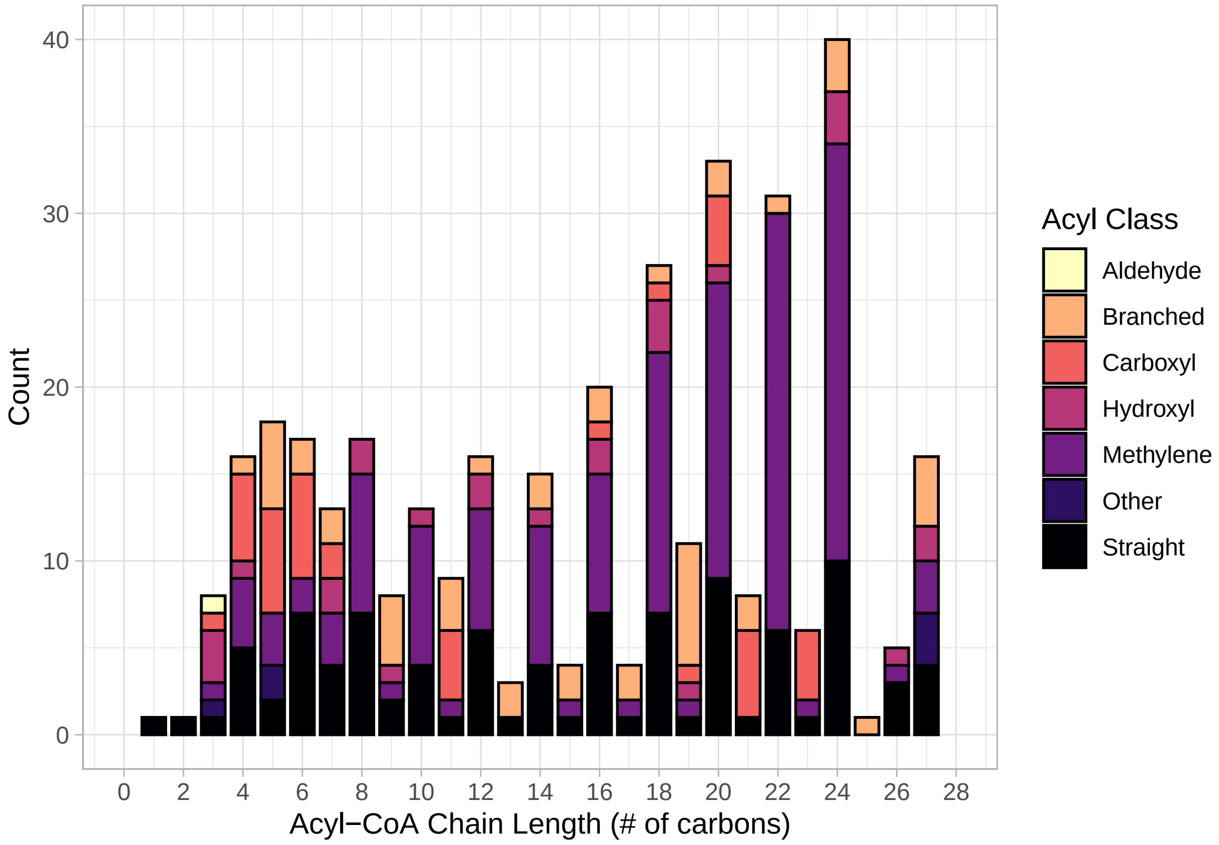
```r
  mutate(carbon_num = str_extract(carbon_num, "\\d+")) %>% #to extract digit only
  mutate(carbon_num = as.numeric(carbon_num)) %>% #numeric, to do substraction next
  mutate(carbon_num_acyl = carbon_num - 21) %>% #remove number of carbons in CoA alone, to get acyls
  slice(-1:-6) %>%  #typos in the dataset
  arrange(carbon_num_acyl) %>%
  mutate (type = "CoA")

CoA <- CoA %>% #number of oxygens
  mutate(o2_num = str_extract(chemical_formula, "O\\d+")) %>% #extract C then digit, then one or more
  mutate(o2_num = str_extract(o2_num, "\\d+")) %>% #to extract digit only
  mutate(o2_num = as.numeric(o2_num)) %>% #numeric, to do substraction next
  mutate(o2_num_acyl = o2_num - 16) %>% #remove number of oxygens in CoA alone, to get acyls
  slice(-1) #remove dephosphoCoA

CoA <- CoA %>%
  separate(smiles, into = c("smiles1", "smiles2"), sep = "S", remove = FALSE, extra = "merge") #split s
#https://en.wikipedia.org/wiki/Simplified_molecular-input_line-entry_system

CoA <- CoA %>%
  mutate(smiles_acyl = if_else(str_detect(smiles1, "P"), smiles2, smiles1)) #this is the code that pull

CoA <- CoA %>%
  mutate(acyl_description = if_else(str_detect(smiles_acyl, "\\(O\\)\\=O"), "Carboxyl",
                                    if_else(str_detect(smiles_acyl, "CO"), "Hydroxyl",
                                    if_else(str_detect(smiles_acyl, "C\\(O\\)C"), "Hydroxyl",
                                    if_else(str_detect(smiles_acyl, "C\\=C"), "Methylene",
                                    if_else(str_detect(smiles_acyl, "C\\(\\=C\\)"), "Methylene",
                                    if_else(str_detect(smiles_acyl, "CC\\=O"), "Aldehyde", #hardcode al
                                    if_else(str_detect(smiles_acyl, "C\\=O"), "Straight", #hardcode for
                                    if_else(str_detect(smiles_acyl, "C\\(C\\)"), "Branched",
                                    if_else(str_detect(smiles_acyl, "CCC"), "Straight",
                                    if_else(str_detect(smiles_acyl, "CC\\(\\=O\\)"), "Straight", #hardc
                                        "Other"))))))))))

CoA %>%
  filter(str_detect(smiles_acyl, "N")) %>% #looking
  count(smiles_acyl, sort = TRUE)
```

```
## # A tibble: 16 x 2
##    smiles_acyl                                                           n
##    <chr>                                                             <int>
##  1 [H][C@](O)(C(O)=NCCC(O)=NCC                                           6
##  2 [H][C@](O)(C(=O)NCCC(=O)NCC                                           3
##  3 C(=O)C=CCCC=CCC=CC=CC=CC(SCC(N)C(O)=O)C(O)CCCC(O)=O                   1
##  4 C(=O)C=CCCCC=CC=CC=CC(SCC(N)C(O)=O)C(O)CCCC(O)=O                      1
##  5 C(=O)CC(=O)CCC=CCC=CC=CC=CC(SCC(N)C(O)=O)C(O)CCCC(O)=O                1
##  6 C(=O)CC(=O)CCCC=CC=CC=CC(SCC(N)C(O)=O)C(O)CCCC(O)=O                   1
##  7 C(=O)CC(O)CCC=CCC=CC=CC=CC(SCC(N)C(O)=O)C(O)CCCC(O)=O                 1
##  8 C(=O)CC(O)CCCC=CC=CC=CC(SCC(N)C(O)=O)C(O)CCCC(O)=O                    1
##  9 C(=O)CC=CC=CC=CC=CC(SCC(N)C(O)=O)C(O)CCCC(O)=O                        1
## 10 C(=O)CC=CCCC=CC=CC=CC(SCC(N)C(O)=O)C(O)CCCC(O)=O                      1
## 11 C(=O)CCC=CCC=CC=CC=CC(SCC(N)C(O)=O)C(O)CCCC(O)=O                      1
## 12 C(=O)CCCCC=CCC=CC=CC=CC(SCC(N)C(O)=O)C(O)CCCC(O)=O                    1
## 13 C(=O)CCN                                                                 1
## 14 CC(N)CC(=O)                                                               1
## 15 CCN=C(O)CCN=C(O)[C@H](O)C(C)(C)COP(O)(=O)OP(O)(=O)OC[C@H]1O[C@H](~       1
## 16 CN1C2CCC1[C@@H]([C@@H](O)C2)C(=O)                                         1
```

```
CoA %>%
  count(average_molecular_weight, sort = TRUE) #code chunk to count acyl-CoAs, both total and discrete
```

```
## # A tibble: 234 x 2
##    average_molecular_weight     n
##                       <dbl> <int>
##  1                    1124.     8
##  2                     892.     5
##  3                     852.     4
##  4                     868.     4
##  5                     918.     4
##  6                     920.     4
##  7                     964.     4
##  8                    1032.     4
##  9                    1106.     4
## 10                    1122.     4
## # ... with 224 more rows
```

```
ggplot(CoA) +
  geom_bar(aes(x = carbon_num_acyl, fill = acyl_description), color = "black", width = 0.8) +
  labs(x = "Acyl-CoA Chain Length (# of carbons)", y = "Count") +
  expand_limits(y = 40) +
  scale_fill_viridis(discrete = TRUE, direction = -1, option = "magma", name = "Acyl Class") +
  scale_x_continuous(breaks = c(0,2,4,6,8,10,12,14,16,18,20,22,24,26,28), limits = c(0,28)) +
  NULL
```



```
#save plot
ggsave("output/fig4.pdf", plot = last_plot(), width = 5, height = 5, dpi = 600)

CoA %>%
  count(average_molecular_weight_noCoA, sort = TRUE)
```

```
## # A tibble: 231 x 2
##    average_molecular_weight_noCoA     n
##                              <dbl> <int>
##  1                            357.      8
##  2                            124.      5
##  3                             84.1     4
##  4                            100.      4
##  5                            150.      4
##  6                            152.      4
##  7                            180.      4
##  8                            196.      4
##  9                            264.      4
## 10                            339.      4
## # ... with 221 more rows
```

Acyl-phosphate Analysis

Code chunk to count acyl-phosphates. 10 total counted, although strangely two are listed at 266 Da. Same or different?

```
phosphate <- metabolites %>%
  filter(str_detect(smiles, "C\\(=O\\)OP")) %>% #regex the smiles code for carbonyl-phosphate bond
  arrange(average_molecular_weight) %>%
  mutate (type = "Acyl Phosphate") %>% #duplicate entry!
  mutate(pre_post = if_else(str_detect(smiles, "C\\(=O\\)OP"), "pre", "post")) %>%
  separate(smiles, into = c("pre_smiles", "post_smiles"), sep = "P", remove = FALSE, extra = "merge") %
  mutate(added_carbons = if_else(grepl("pre", pre_post), str_count(pre_smiles, "C"), str_count(post_smi
phosphate
```

```
## # A tibble: 10 x 11
##    accession name  average_molecul~ chemical_formula smiles pre_smiles
##    <chr>     <chr>            <dbl> <chr>            <chr>  <chr>
##  1 HMDB0001~ Acet~             140. C2H5O5P          CC(=O~ CC(=O)O
##  2 HMDB0001~ Carb~             141. CH4NO5P          NC(=O~ NC(=O)O
##  3 HMDB0012~ L-As~             213. C4H8NO7P         NC(CC~ NC(CC(=O)O
##  4 HMDB0001~ L-Gl~             227. C5H10NO7P        N[C@@~ N[C@@H](C~
##  5 HMDB0001~ Glyc~             266. C3H8O10P2        OC(CO~ OC(CO
##  6 HMDB0062~ 3-ph~             266. C3H8O10P2        OC(CO~ OC(CO
##  7 HMDB0006~ N-Ac~             269. C7H12NO8P        CC(=O~ CC(=O)N[C~
##  8 HMDB0006~ Acet~             389. C12H16N5O8P      CC(=O~ CC(=O)O
##  9 HMDB0006~ Prop~             403. C13H18N5O8P      CCC(=~ CCC(=O)O
## 10 HMDB0006~ L-2-~             490. C16H23N6O10P     N[C@@~ N[C@@H](C~
## # ... with 5 more variables: post_smiles <chr>,
## #   normal_concentrations <chr>, type <chr>, pre_post <chr>,
## #   added_carbons <int>
```

```
phosphate %>%
  count(average_molecular_weight, sort = TRUE) #code chunk to count acyl-CoAs, both total and discrete
```

```
## # A tibble: 9 x 2
##   average_molecular_weight     n
##                      <dbl> <int>
## 1                     266.     2
## 2                     140.     1
## 3                     141.     1
## 4                     213.     1
## 5                     227.     1
## 6                     269.     1
## 7                     389.     1
## 8                     403.     1
## 9                     490.     1
```

Figure 5

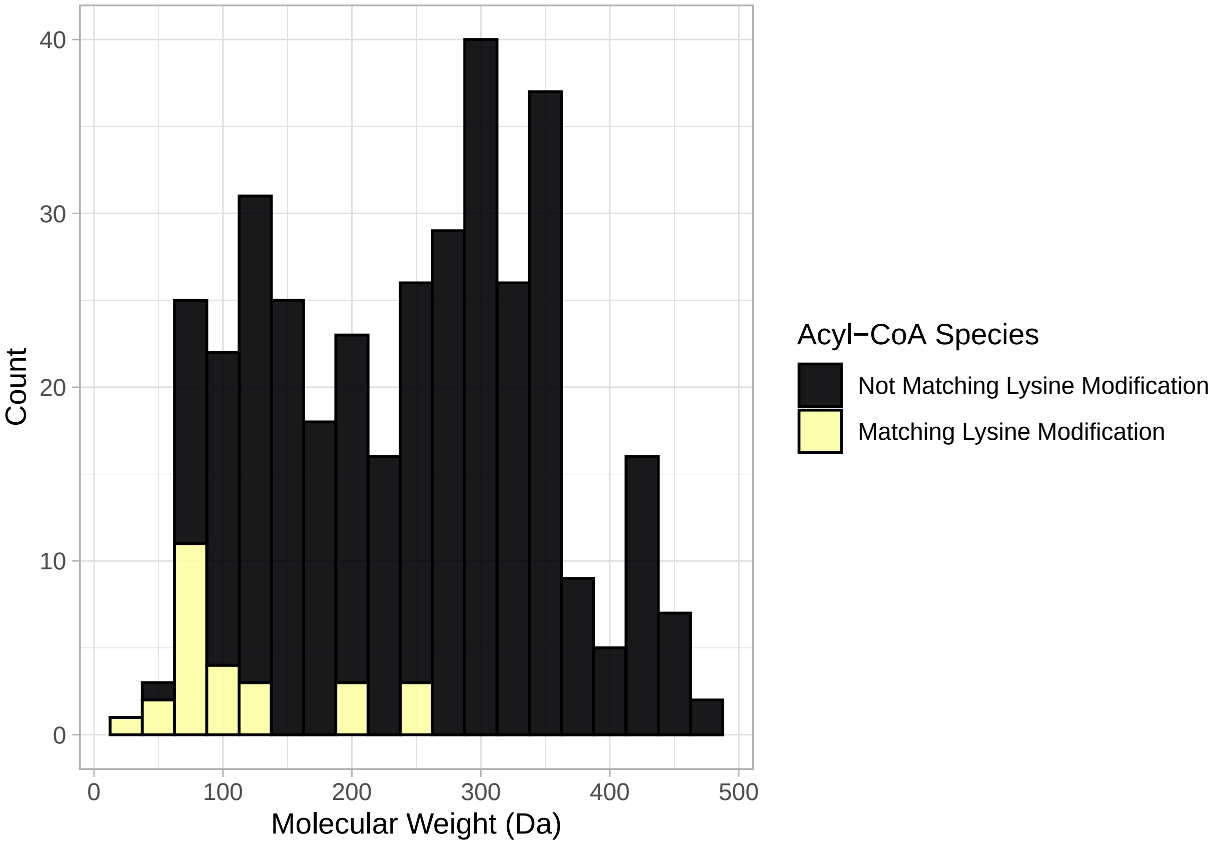
In this figure, the goal is to determine how many reactive [human] metabolites there are and to determine how many are associated with PTMs

```
#count thioesters
thioester <- metabolites %>%
  filter(str_detect(smiles, "C\\(=O\\)S") | str_detect(smiles, "SC\\(=O\\)")) %>% #regex the smiles cod
  arrange(average_molecular_weight) %>%
```

```r
  mutate(type = "Thioester") %>%
  mutate(pre_post = if_else(str_detect(smiles, "C\\(=O\\)S"), "pre", "post")) %>%
  separate(smiles, into = c("pre_smiles", "post_smiles"), sep = "S", remove = FALSE, extra = "merge" ) %
  mutate(added_carbons = if_else(grepl("pre", pre_post), str_count(pre_smiles, "C"), str_count(post_smil

#because the smiles code has thioesters with orientations that could add carbon on either sides of the

#these include all from CoA list, except "CoA-"
#anti_join(CoA, thioester, by = "name")
#semi_join(CoA, thioester, by = "name") leaves 355, which is one less than in the CoA df

#sum(str_count(thioester$smiles, "C\\(=O\\)S")) #346
#sum(str_count(thioester$smiles, "SC\\(=O\\)")) #80

#thioester$added_carbons <- as.factor(thioester$added_carbons)

match2 <- ptm %>%
  filter(!FT == "CROSSLNK") %>% #omit AA cross links only
  filter(TG == "Lysine") %>%
  select(MA) %>%
  round(2) %>%
  distinct() %>%
  pull()

match1 <- CoA %>%
  select(average_molecular_weight_noCoA) %>%
  round(2) %>%
  distinct %>%
  mutate(mod =  if_else(average_molecular_weight_noCoA %in% match2, TRUE, FALSE)) %>%
  left_join(CoA, by = "average_molecular_weight_noCoA")

match1 %>% count(mod, sort = TRUE)
```

```
## # A tibble: 2 x 2
##   mod       n
##   <lgl> <int>
## 1 FALSE   334
## 2 TRUE     27
```

```r
ggplot(match1) +
  geom_histogram(aes(x = average_molecular_weight_noCoA, fill = mod), color = "black", binwidth = 25, c
  labs(x = "Molecular Weight (Da)", y = "Count") +
  scale_fill_viridis(discrete = TRUE, direction = 1, option = "inferno", name = "Acyl-CoA Species", labe
  NULL
```

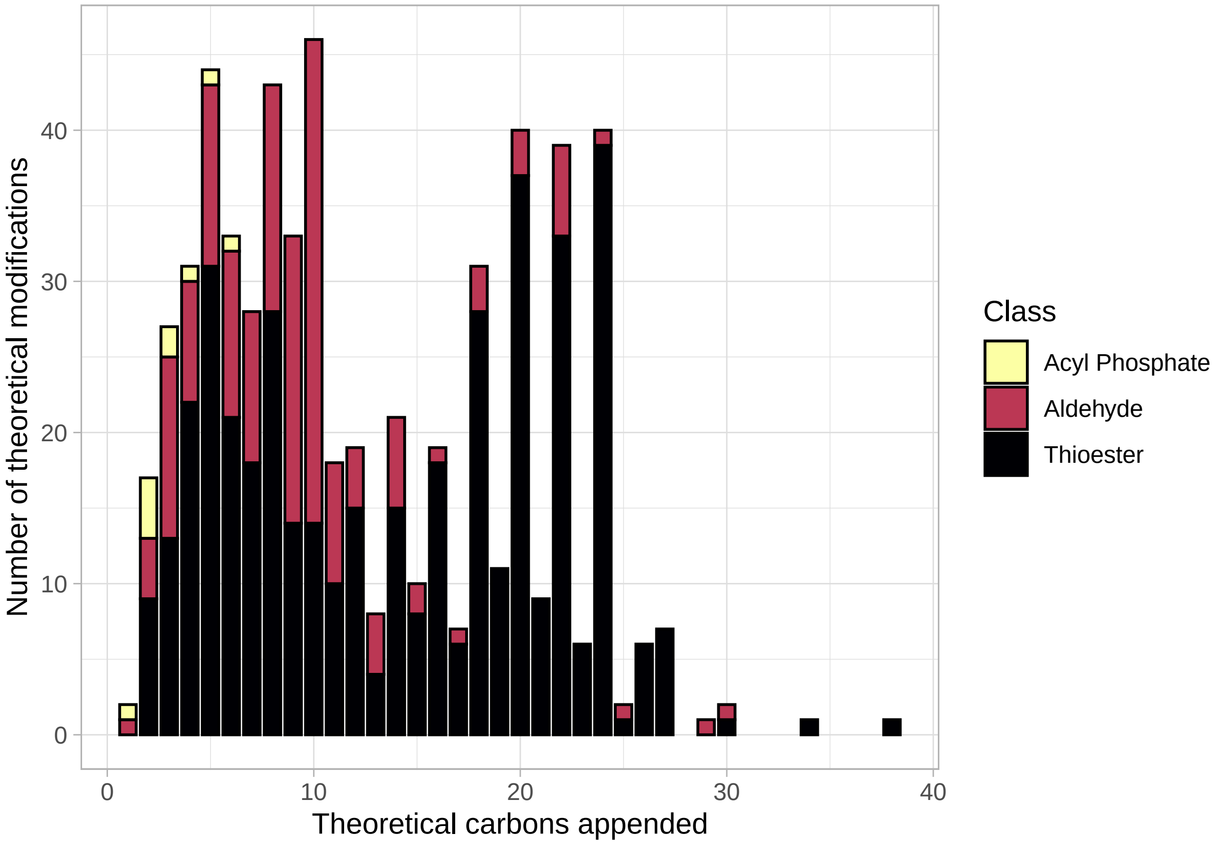


```r
#save plot
ggsave("output/fig5a.pdf", plot = last_plot(), width = 7, height = 5, dpi = 600)
```

```r
#count aldehydes
aldehyde <- metabolites %>%
  filter(str_detect(name, "aldehyde")) %>% #str_detect for aldehydes give too many false positives
  arrange(average_molecular_weight) %>%
  mutate(type = "Aldehyde") %>%
  mutate(added_carbons = str_count(smiles, "C"))

#Merge thioesters, phosphates, aldehydes
carbon <- full_join(thioester, phosphate) %>%
  full_join(aldehyde) %>%
  arrange(average_molecular_weight) %>%
  select(-c("smiles", "pre_smiles", "post_smiles", "pre_post", "normal_concentrations"))
```

```
## Joining, by = c("accession", "name", "average_molecular_weight", "chemical_formula", "smiles", "pre_

## Joining, by = c("accession", "name", "average_molecular_weight", "chemical_formula", "smiles", "norma
```

```r
ggplot(carbon) +
  geom_bar(aes(x = added_carbons, fill = type), color = "black", width = 0.8) +
  labs(x = "Theoretical carbons appended", y = "Number of theoretical modifications") +
  scale_fill_viridis(discrete = TRUE, direction = -1, option = "inferno", name = "Class") +
  NULL
```



```
#save plot
ggsave("output/fig5b.pdf", plot = last_plot(), width = 5, height = 5, dpi = 600)
```

Save final files

Code chunk to save files

```
write_delim(ptm, "output/table_s1.csv", delim = ",", na = "")
write_delim(metabolites, "output/table_s2.csv", delim = ",", na = "")
write_delim(carbon, "output/table_s3.csv", delim = ",", na = "")
beep(sound = 8) #because mario is awesome
```