

```

1  setwd("~/04_Champion_Lab/02_N-terminal_Acetylation/OnePot")
2  library(readxl)
3  library(stringr)
4  library(tidyr)
5  library(dplyr)
6  library(seqinr)
7  library(Peptides)
8  library(cleaver)
9
10 #One pot import and cleaning
11 opDF <- read.csv("Data/db.protein-peptides.csv")
12
13 opDF <- opDF[!grepl("CONTAM", opDF$Accession),]
14 opDF$Peptide <- str_remove(opDF$Peptide, "^\\.\\.\\.")
15 opDF$Peptide <- str_remove(opDF$Peptide, "\\..$")
16
17
18 opDF$NtermAcet_L <- grepl("^([A-Z]\\(\\(\\+42\\.\\.01\\.\\.\\)", opDF$Peptide)
19 opDF$NtermAcet_H <- grepl("^([A-Z]\\(\\(\\+45\\.\\.03\\.\\.\\)", opDF$Peptide)
20 opDF$UnLabelled <- !opDF$NtermAcet_H & !opDF$NtermAcet_L
21
22 names(opDF) <- str_replace_all(names(opDF), "Intensity.New", "Intensity_Sample")
23 names(opDF) <- str_replace_all(names(opDF), "Area.New", "Area_Sample")
24 names(opDF) <- str_replace_all(names(opDF), "X.Spec.New", "nSpec_Sample")
25
26
27 opDF <- opDF %>% pivot_longer(which((grepl("^Area", names(opDF)) |
28                               grepl("^Intensity", names(opDF)) |
29                               grepl("^nSpec", names(opDF)))),
30                               names_to = c(".value", "measure"),
31                               names_sep = "_")
32 drops <- c("Unique", "Top", "scan", "Source File", "AScore")
33 opDF <- opDF[, !names(opDF) %in% drops]
34
35
36 opDF$Accession2 <- str_remove(str_extract(opDF$Accession, "^[^\\|\\|]*\\|\\|"), "\\|\\|")
37
38
39 opDFsummary <- opDF %>% group_by(Peptide, Mass, m.z, z,
40                                RT, Start, End, Length) %>%
41   summarise(Accessions = paste(unique(Accession2), collapse = ";"),
42             rowsN = n(),
43             valuesArea = sum(!is.na(Area)),
44             meanArea = mean(Area, na.rm = T),
45             sdArea = sd(Area, na.rm = T),
46             maxArea = max(Area, na.rm = T),
47             valuesIntensity = sum(!is.na(Intensity)),
48             meanIntensity = mean(Intensity, na.rm = T),
49             sdIntensity = sd(Intensity, na.rm = T),
50             maxIntensity = max(Intensity, na.rm = T),
51             valuesSpec = sum(nSpec > 0))
52
53
54 opDFsummary$str_seq <- str_remove_all(opDFsummary$Peptide,
55   "\\([\\|\\+\\|\\-\\|][0-9]+\\.\\.[0-9]+\\|\\|)")
56
57 opDFsummary$IEP <- NA
58 for (i in 1:nrow(opDFsummary)) {
59   opDFsummary$IEP[i] <- computePI(s2c(opDFsummary$str_seq[i]))
60   if (i %% 500 == 0) {
61     print(i/nrow(opDFsummary) * 100)
62   }
63 }
64 #save as dataframe
65 write.csv(opDFsummary, "Data/OnePotSummarisedData.csv",
66           row.names = F, quote = F)
67
68 #enriched data (Thompson et al.) import and cleaning (trypsin only)

```

```

69 enrDF <- read_excel("Data/Cristal Marinum.xlsx", sheet = "Raw NTA Peptides")
70
71 enrDF <- enrDF[!grepl("GluC", enrDF$File),]
72
73 enrDF$Peptide <- paste0(enrDF$Sequence, "|", enrDF$Modifications)
74
75 enrDFsummary <- enrDF %>% group_by(Peptide, Sequence, `Theor m/z`, `Theor z`,
76 `Theor MW`, Length) %>%
77 summarise(Genes = paste(unique(Gene), collapse = ";"),
78 rowsN = n(),
79 meanIntensity = mean(`Intensity (Peptide)`, na.rm = T),
80 maxIntensity = max(`Intensity (Peptide)`, na.rm = T),
81 valuesIntensity = sum(!is.na(`Intensity (Peptide)`)))
82
83 #function to return start position of peptide from proteome db
84 return_start_position <- function(peptide, accession, database){
85 protein_names <- names(database)
86 protein_index <- which(grepl(accession, protein_names))
87 if(length(protein_index) > 1) {
88 return(NA)
89 } else if (length(protein_index) < 1) {
90 return(NA)
91 } else {
92 protein <- toupper(c2s(database[[protein_index[1]]]))
93 }
94 matches_df <- str_locate(protein, peptide)
95 if (length(matches_df) < 1) {
96 return(NA)
97 } else {
98 start_position <- matches_df[[1,1]]
99 }
100 return(start_position)
101 }
102
103 #read in database
104 db <- read.fasta("Data/SDW_codon_MARINUM_MetStart.fasta", seqtype = "AA")
105
106
107 enrDFsummary$IEP <- NA
108 enrDFsummary$Start <- NA
109 for (i in 1:nrow(enrDFsummary)) {
110 enrDFsummary$IEP[i] <- computePI(s2c(enrDFsummary$Sequence[i]))
111 enrDFsummary$Start[i] <- return_start_position(enrDFsummary$Sequence[i],
112 enrDFsummary$Genes[i],
113 database = db)
114 if (i %% 500 == 0) {
115 print(i/ nrow(enrDFsummary) *100)
116 }
117 }
118
119 #saving
120 write.csv(enrDFsummary, "Data/EnrichedSummarisedData.csv",
121 row.names = F, quote = F)
122
123
124 #same but with GluC
125 enrDF <- read_excel("Data/Cristal Marinum.xlsx", sheet = "Raw NTA Peptides")
126
127 enrDF <- enrDF[grepl("GluC", enrDF$File),]
128
129 enrDF$Peptide <- paste0(enrDF$Sequence, "|", enrDF$Modifications)
130
131 enrDFsummary <- enrDF %>% group_by(Peptide, Sequence, `Theor m/z`, `Theor z`,
132 `Theor MW`, Length) %>%
133 summarise(Genes = paste(unique(Gene), collapse = ";"),
134 rowsN = n(),
135 meanIntensity = mean(`Intensity (Peptide)`, na.rm = T),
136 maxIntensity = max(`Intensity (Peptide)`, na.rm = T),
137 valuesIntensity = sum(!is.na(`Intensity (Peptide)`)))

```

```

138
139
140 return_start_position <- function(peptide, accession, database){
141   protein_names <- names(database)
142   protein_index <- which(grepl(accession, protein_names))
143   if(length(protein_index) > 1) {
144     return(NA)
145   } else if (length(protein_index) < 1) {
146     return(NA)
147   } else {
148     protein <- toupper(c2s(database[[protein_index[1]]]))
149   }
150   matches_df <- str_locate(protein, peptide)
151   if (length(matches_df) < 1) {
152     return(NA)
153   } else {
154     start_position <- matches_df[[1,1]]
155   }
156   return(start_position)
157 }
158 db <- read.fasta("Data/SDW_codon_MARINUM_MetStart.fasta", seqtype = "AA")
159
160
161 enrDFsummary$IEP <- NA
162 enrDFsummary$Start <- NA
163 for (i in 1:nrow(enrDFsummary)) {
164   enrDFsummary$IEP[i] <- computePI(s2c(enrDFsummary$Sequence[i]))
165   enrDFsummary$Start[i] <- return_start_position(enrDFsummary$Sequence[i],
166                                                    enrDFsummary$Genes[i],
167                                                    database = db)
168   if (i %% 500 == 0) {
169     print(i/ nrow(enrDFsummary) *100)
170   }
171 }
172
173
174 write.csv(enrDFsummary, "Data/EnrichedSummarisedDataGLUC.csv",
175           row.names = F, quote = F)
176
177
178
179
180
181 #list of Nterminal peptides from genome
182 NtermPeptides <- list()
183
184 for (i in 1:length(db)) {
185   protein <- db[[i]]
186   seq <- toupper(c2s(protein))
187   MMAR <- str_remove(str_extract(attr(protein, 'name'), "^[^\\|]+|"), "\\|")
188   peptide <- cleave(seq, enzym = "trypsin")[[1]][1]
189   NtermPeptides[[MMAR]] <- peptide
190 }
191
192 NtermGenome <- data.frame(Accession = names(NtermPeptides),
193                           Peptide = unlist(NtermPeptides))
194
195 NtermGenome$Length <- nchar(NtermGenome$Peptide)
196 NtermGenome$secondAA <- substr(NtermGenome$Peptide, 2,2)
197 NtermGenome$metCleavedPeptide <- str_remove(NtermGenome$Peptide, "^M")
198
199
200 NtermGenome$IEP <- NA
201 for (i in 1:nrow(NtermGenome)) {
202   if (NtermGenome$secondAA[i] %in% c("G", "A", "S", "T", "V", "C", "P")) {
203     NtermGenome$IEP[i] <- computePI(s2c(NtermGenome$metCleavedPeptide[i]))
204   } else {
205     NtermGenome$IEP[i] <- computePI(s2c(NtermGenome$Peptide[i]))
206   }

```

```

207     if (i %% 500 == 0) {
208         print(i/ nrow(NtermGenome) *100)
209     }
210 }
211
212
213 write.csv(NtermGenome, "Data/GenomeSummarisedData_GASTVCP.csv", row.names = F, quote = F)
214
215
216
217
218
219
220 #all genome peptides (not just n terminal)
221
222
223 proteinDFList <- list()
224
225 #filters
226 minLength <- 5
227 maxLength <- 50
228
229
230
231 for (i in 1:length(db)) {
232     if (i %% 100 == 0) {
233         print(i / length(db))
234     }
235     protein <- db[[i]]
236     seq <- toupper(c2s(protein))
237     seq <- str_remove(seq, "\\*")
238     MMAR <- str_remove(str_extract(attr(protein, 'name'), "^[^\\|]+|"), "\\|")
239     peptides <- cleave(seq, enzym = "trypsin")[[1]]
240     peptideDF <- data.frame(Accession = MMAR,
241                             Peptide = peptides)
242     peptideDF$NtermPeptide <- c(T, rep(F, nrow(peptideDF) - 1))
243     NtermPeptide <- peptideDF$Peptide[1]
244     secondAA <- substr(NtermPeptide, 2,2)
245     if (secondAA %in% c("G", "A", "S", "T", "V", "C", "P")) {
246         peptideDF$Peptide[1] <- str_remove(peptideDF$Peptide[1], "^M")
247     }
248     peptideDF$Length <- nchar(peptideDF$Peptide)
249
250     peptideDF <- peptideDF[peptideDF$Length >= minLength &
251                             peptideDF$Length <= maxLength,]
252
253
254     if (nrow(peptideDF) > 0) {
255         peptideDF$IEP <- NA
256         peptideDF$mass <- NA
257         peptideDF$IEP <- sapply(peptideDF$Peptide, function(peptide) computePI(s2c(peptide)))
258         peptideDF$mass <- sapply(peptideDF$Peptide, mw)
259
260         proteinDFList[[MMAR]] <- peptideDF
261     } else {
262         print(i)
263         print("noPeps")
264     }
265
266 }
267
268 genomePeps <- bind_rows(proteinDFList)
269
270 write.csv(genomePeps, "Data/GenomeSummarisedData_allPeptides_GASTVCP.csv", row.names = F
271 , quote = F)

```