R code used to generate proteomic dissection profiles

```
# function to compute Dissection Profile
# modified 11/23/2020 to use chi squared instead of correlation to compare potein abundance columns
# given three lists of protein abundances from mass spec from three samples
# A, B, and C, we scan through a range of proportions p that describe how
# the first two sampes are mixed to give the third sample
# by calculating cov((pA + (1-p)B), C) and finding the p that maximizes the covariance
# this is a measure for the relative proportion of cell fragments A and B in the
# combined sample C (which could be a whole cell for example)

# format for the input is a dataframe where the first column will be gene access numbers
# and the ohter columns will be abundances in various fractions
# the three inputs A B and C will be column numbers
# p will vary from 0 to 1 in 101 increments so 0, 0.01, 0.02, etc.

# the variable holding the data frame is called proteome_data
# proteome_data <- read.table("abundance_data_for_profile.csv", sep=",", header=TRUE)

data_table <- proteome_data5

A <- 4     # fragment A (MB-less body) fraction whose contribution is weightd by p
B <- 2    # fragment B  (MB)   fraction whose contribution is weighted by (1-p)
C <-   3  # whole cell

results = c()   # setting up the array to hold the values of p

for (pindex in 0:100)
{
  p <- 0.01*pindex
  fragment_sum <- p*data_table[,A] + (1-p)*data_table[,B]
  Xsq <- chisq.test(round(fragment_sum), data_table[,C], simulate.p.value=TRUE)
  match_score <- as.numeric(Xsq$statistic)  # have to converted a named number to a number
  results <- c(results, match_score)
}

best_correlation <- min(results)
print(best_correlation)
best_p <- which.min(results)
print(best_p - 1)  #correct for offset
```