```
640  #create data frame containing correlations between control and RNase group and the absolute amount of
     fractions between control and RNase peak
641  df_lm = data.frame(row.names =  protein_names)
642
643
644  correlation_function = function(x){cor(x[1:25], x[26:50])} #creating        ...ion which will compute the
     correlation between c...        RNase of each protein
645
646  corr_vector = apply(                                    ...on on all proteins in the
     data frame
647
648  df_lm$cor = c(corr_ve
649
650  df_lm$shift = c(a...                                     ...n global peaks are added
     -> no negative val...
651
652  #create two data frames                                 ...maining 10% of rows
653
654  n = nrow(df_lm) #number of ...
655  n_new = round(0.9 * n)   #90% of ...
656
657  train_df = df_lm[1:n_new, ] #create new data frame with 90% of the rows -> train the model
658  predict_df = df_lm[4130:4588, ] #create new data frame with the remaining 10% of the rows -> test how
     well our model can predict
659
660  linear_reg = lm(shift ~ cor, train_df) #linear model is created
661  summary(linear_reg) #control all available values
```

Welcome back!

Steps of the project

1 → **2** → **3** → **4** → **5** → **6** → **7**

Data clean-up
- Normalization
- Scaling
- Smoothing

Correlation

Peak finder

Identifying shifts

T-Test

K-means

Linear Regression
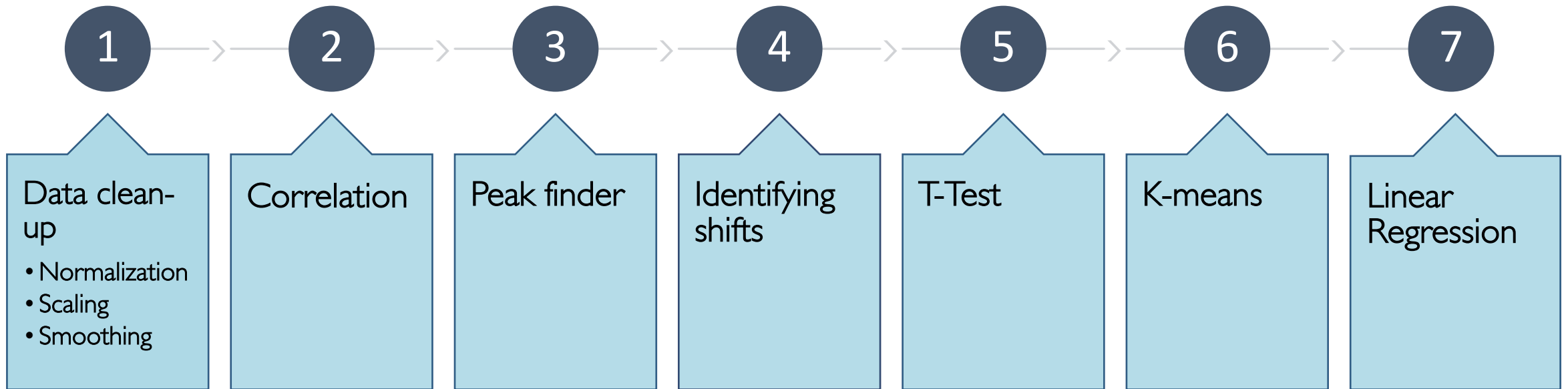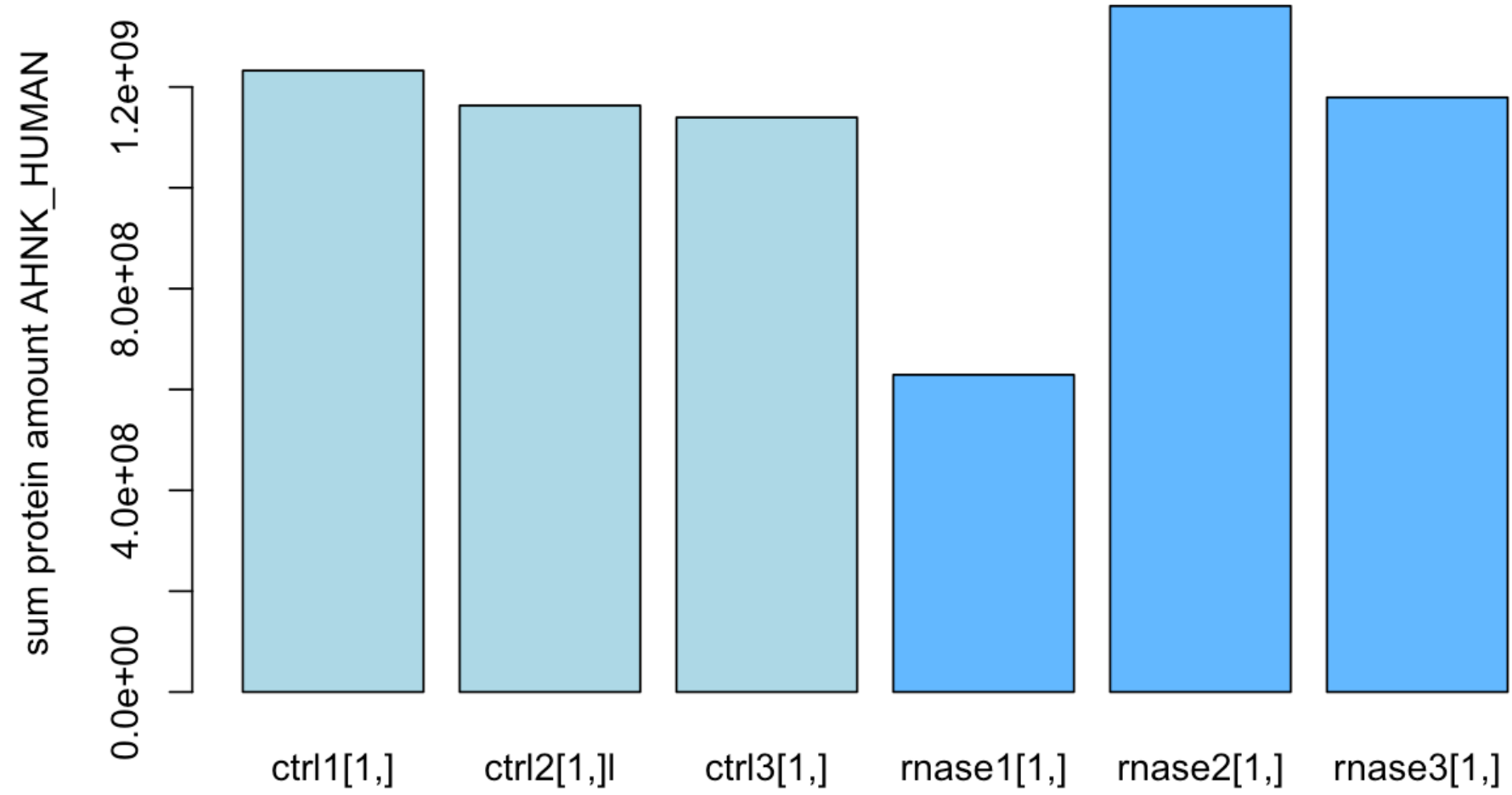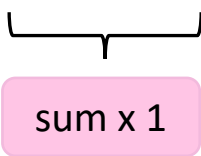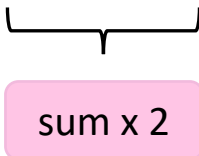
Normalization

Rep-wise normalization
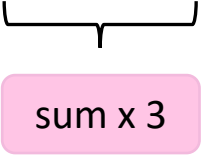
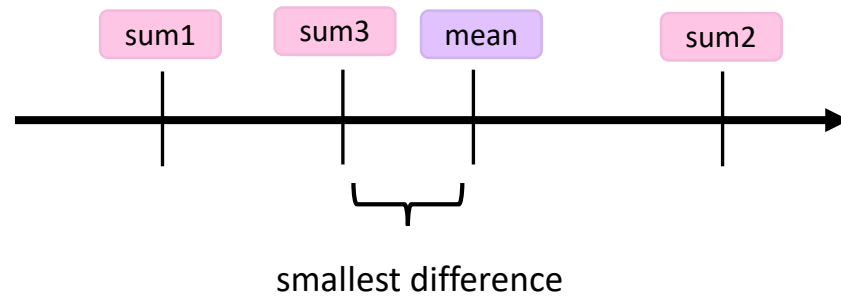| Protein	fraction x rep 1	fraction x rep 2	fraction x rep 3

sum x 1

sum x 2

sum x 3
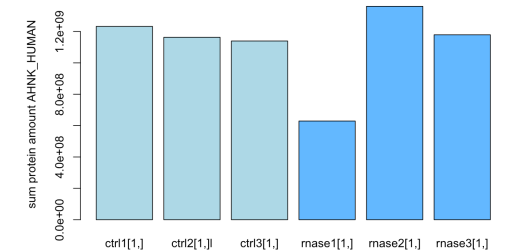
Normalization factor:

sum x 3 / sum x 1 = normalization factor x 1

sum x 3 / sum x 2 = normalization factor x 2

sum x 3 / sum x 3 = normalization factor x 3

sum1   sum3   mean   sum2

smallest difference

Fraction-wise normalization

Protein	fraction 1 rep 1	fraction 2 rep 1	fraction 3 rep 1

sum protein 1

sum protein 2

sum protein 3
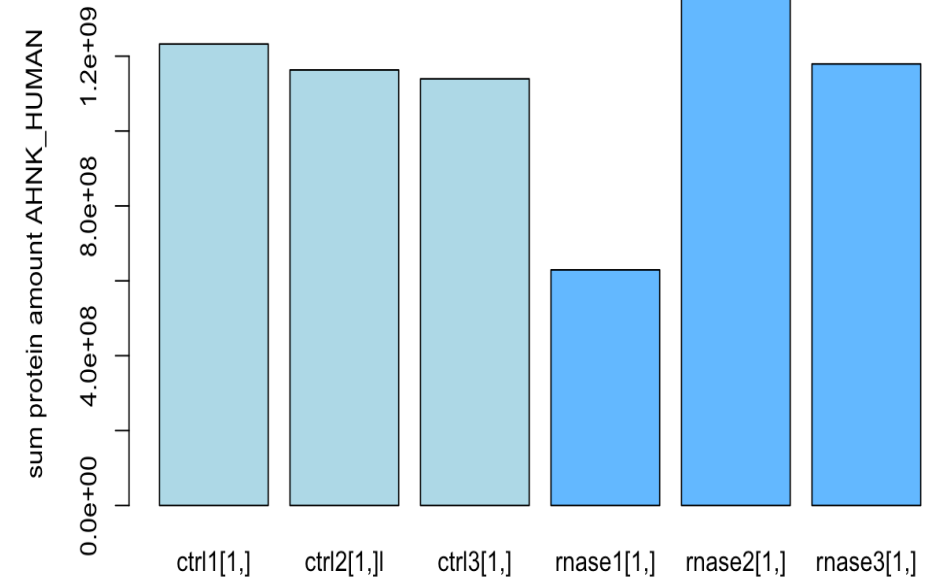
QX 1 = sum protein 1 / 100

fraction 1 rep 1 scaled = (fraction 1 rep 1) / QX 1

Scaling

sum of protein amounts **before** scaling
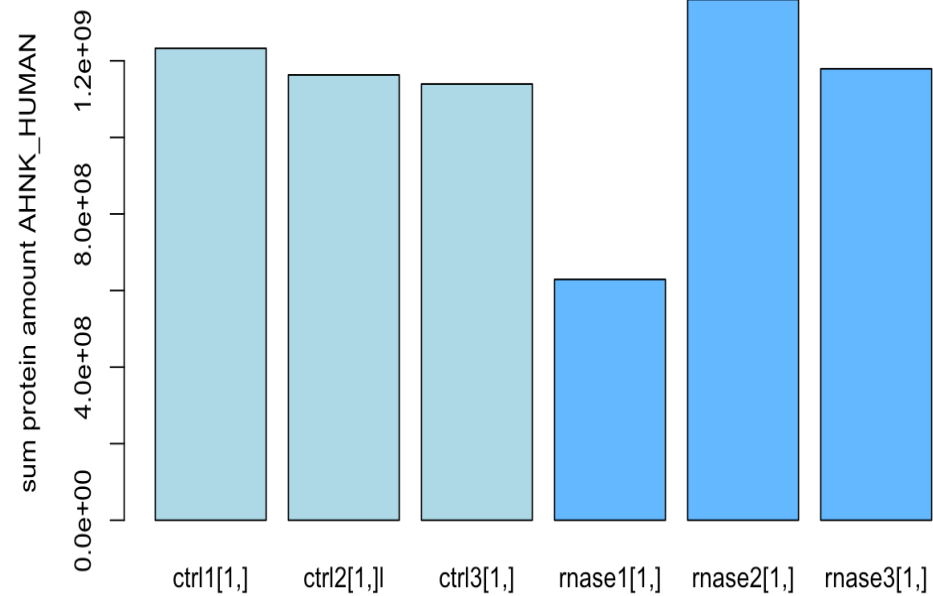
sum of protein amounts **after** scaling
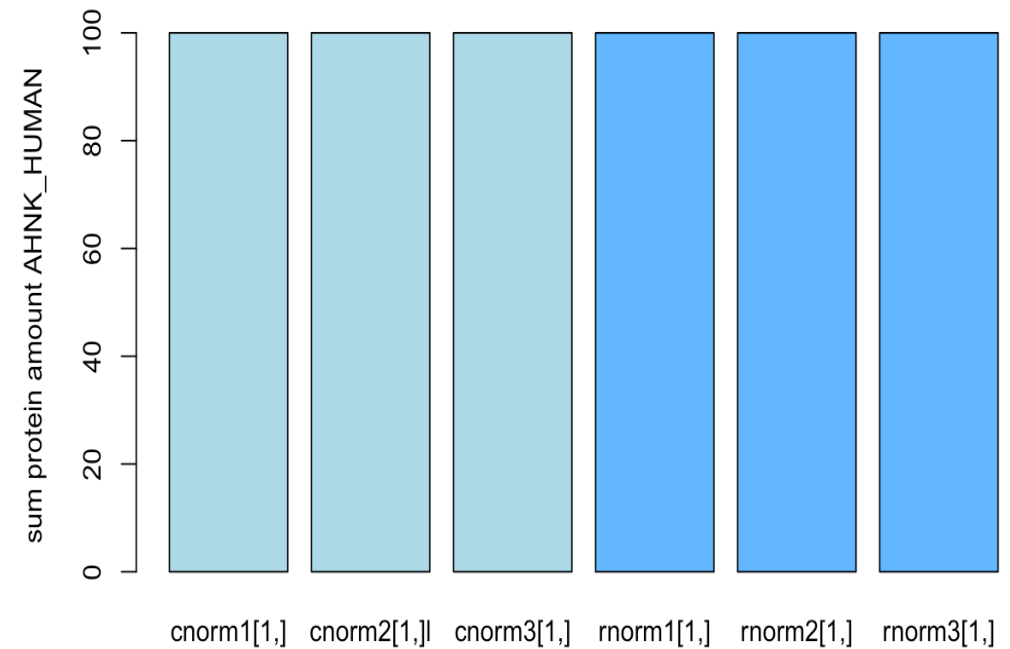
Smoothing

AHNK_HUMAN before smoothing
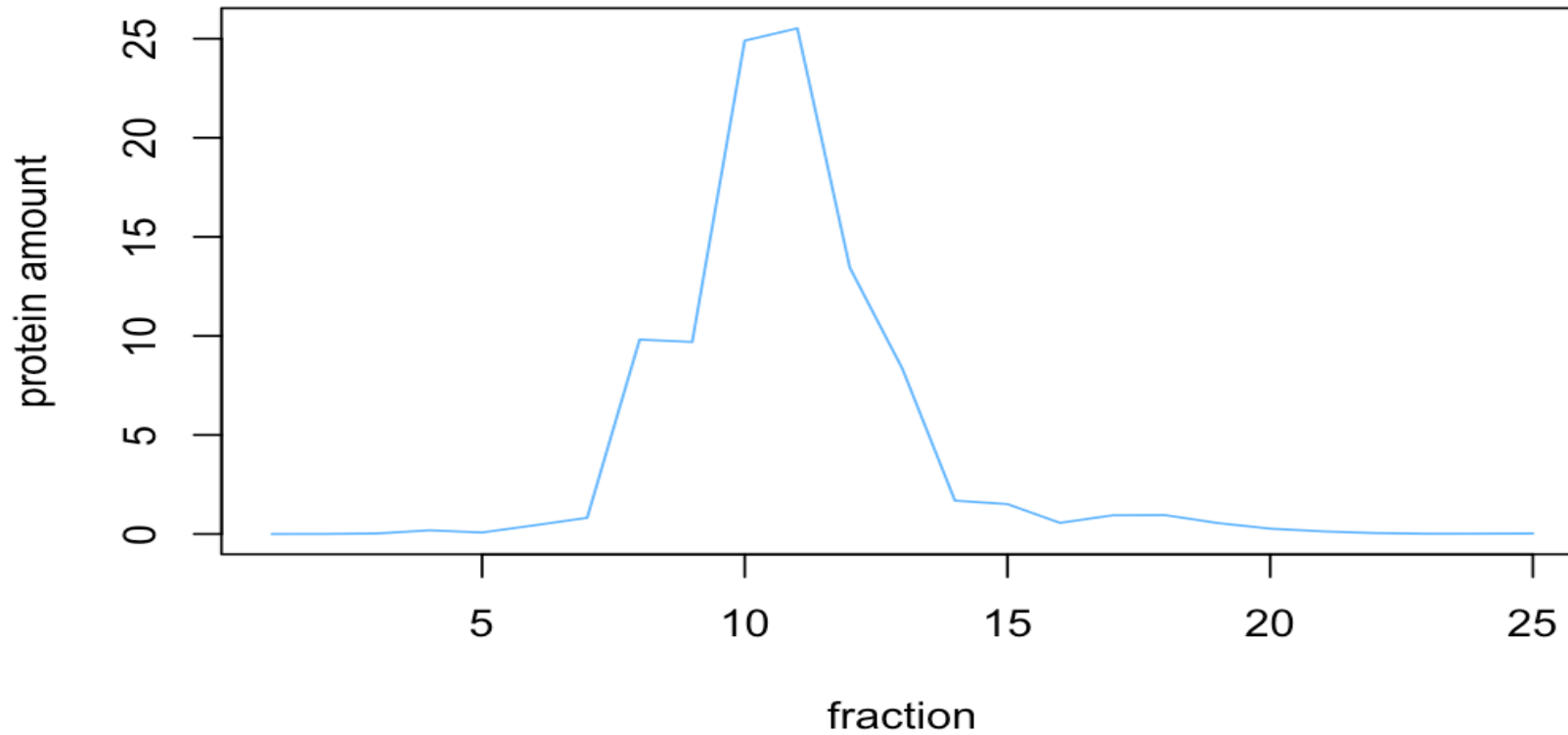
$fraction\_x\_smoothed = mean(fraction\_x-1, fraction\_x, fraction\_x+1)$

Smoothing
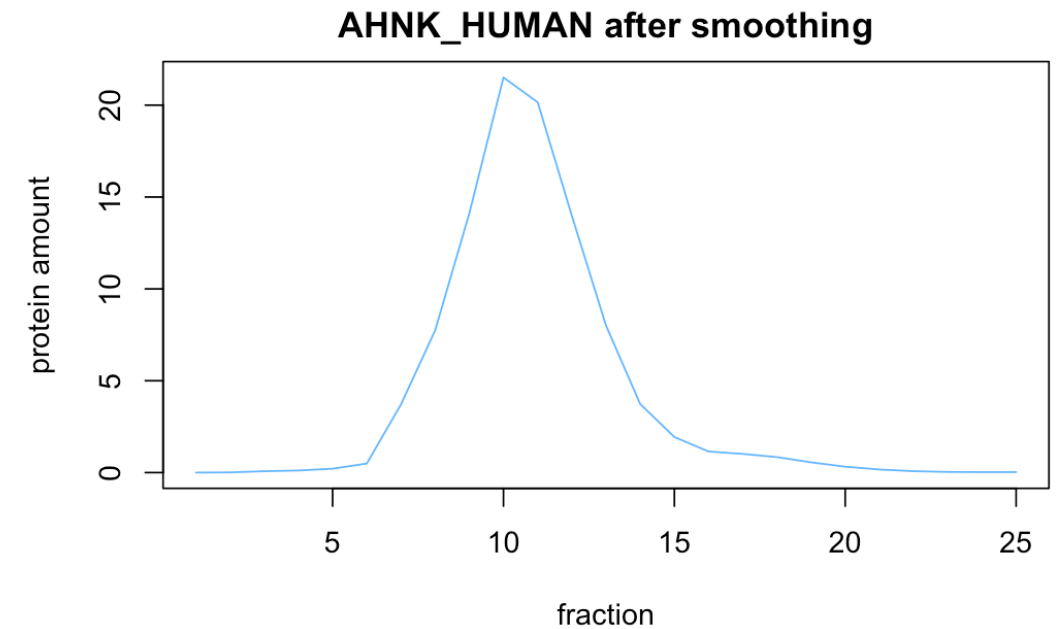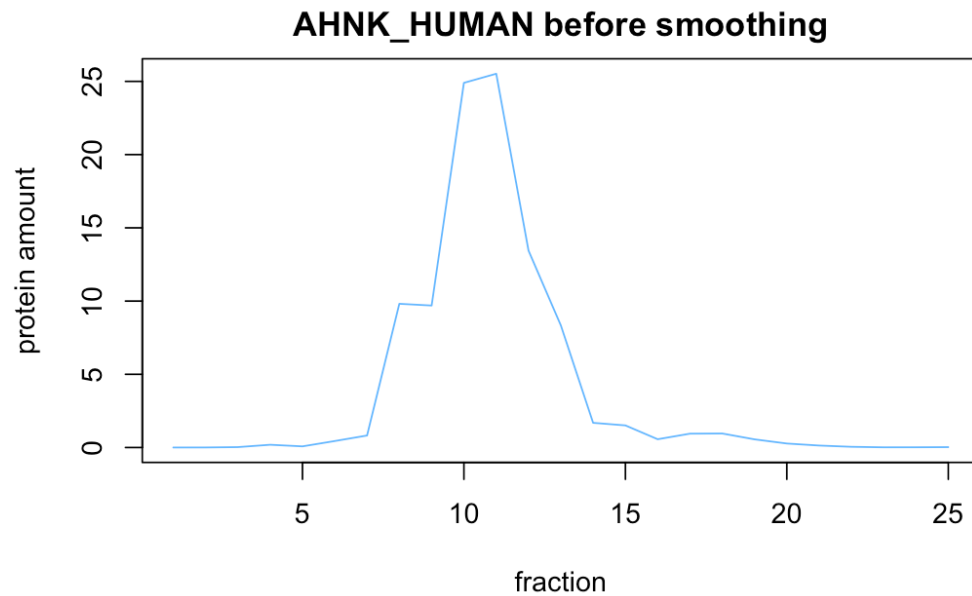
AHNK_HUMAN before smoothing

AHNK_HUMAN after smoothing
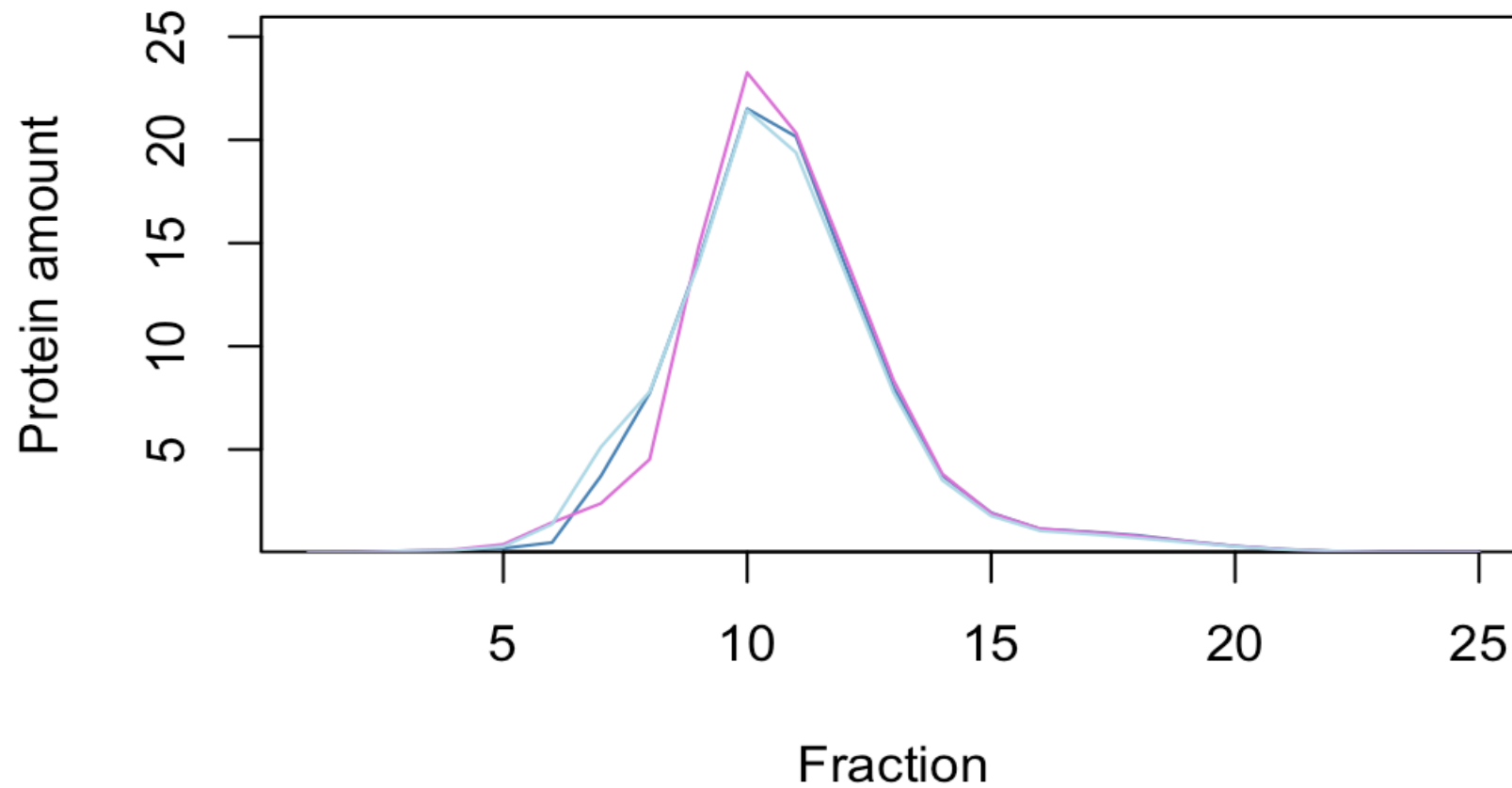
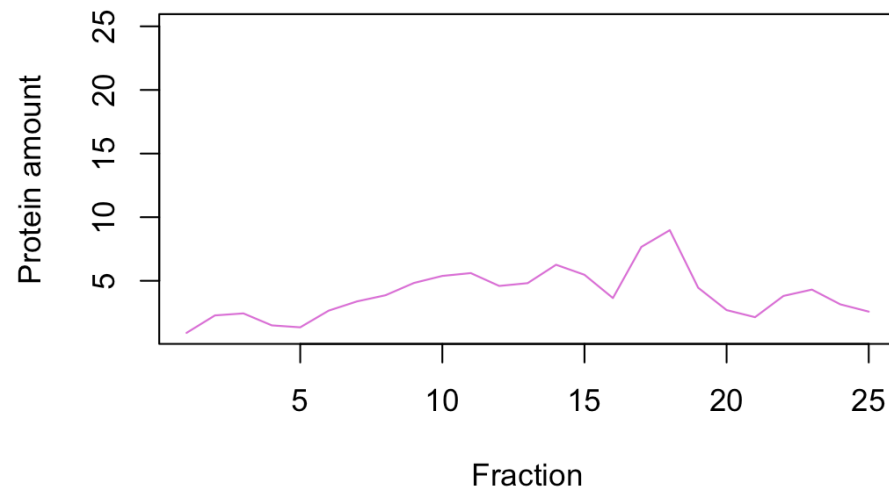fraction_x_smoothed = mean(fraction_x-1, fraction_x, fraction_x+1)
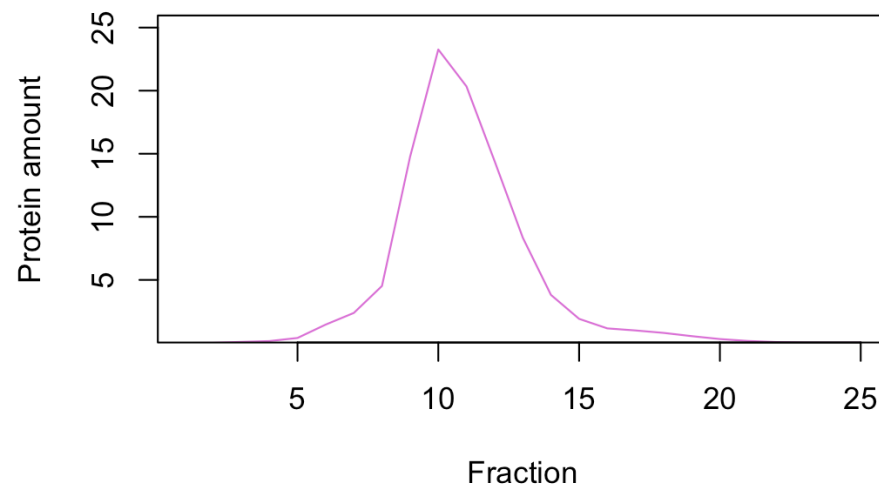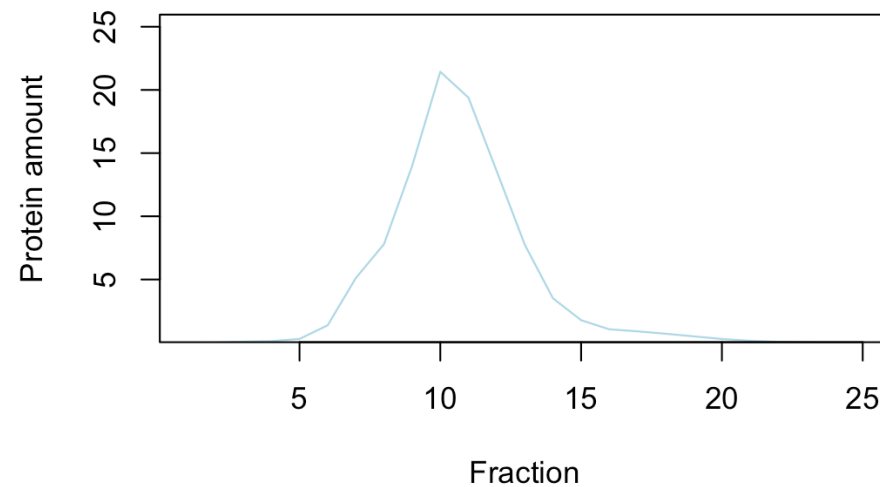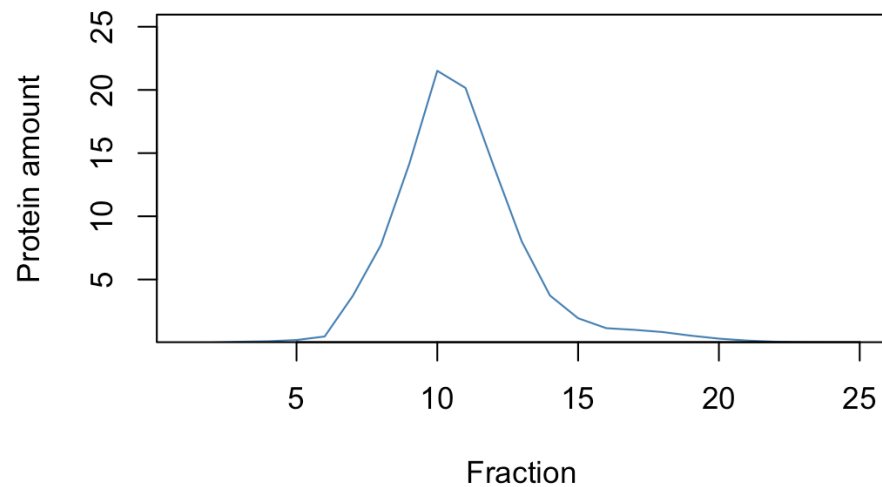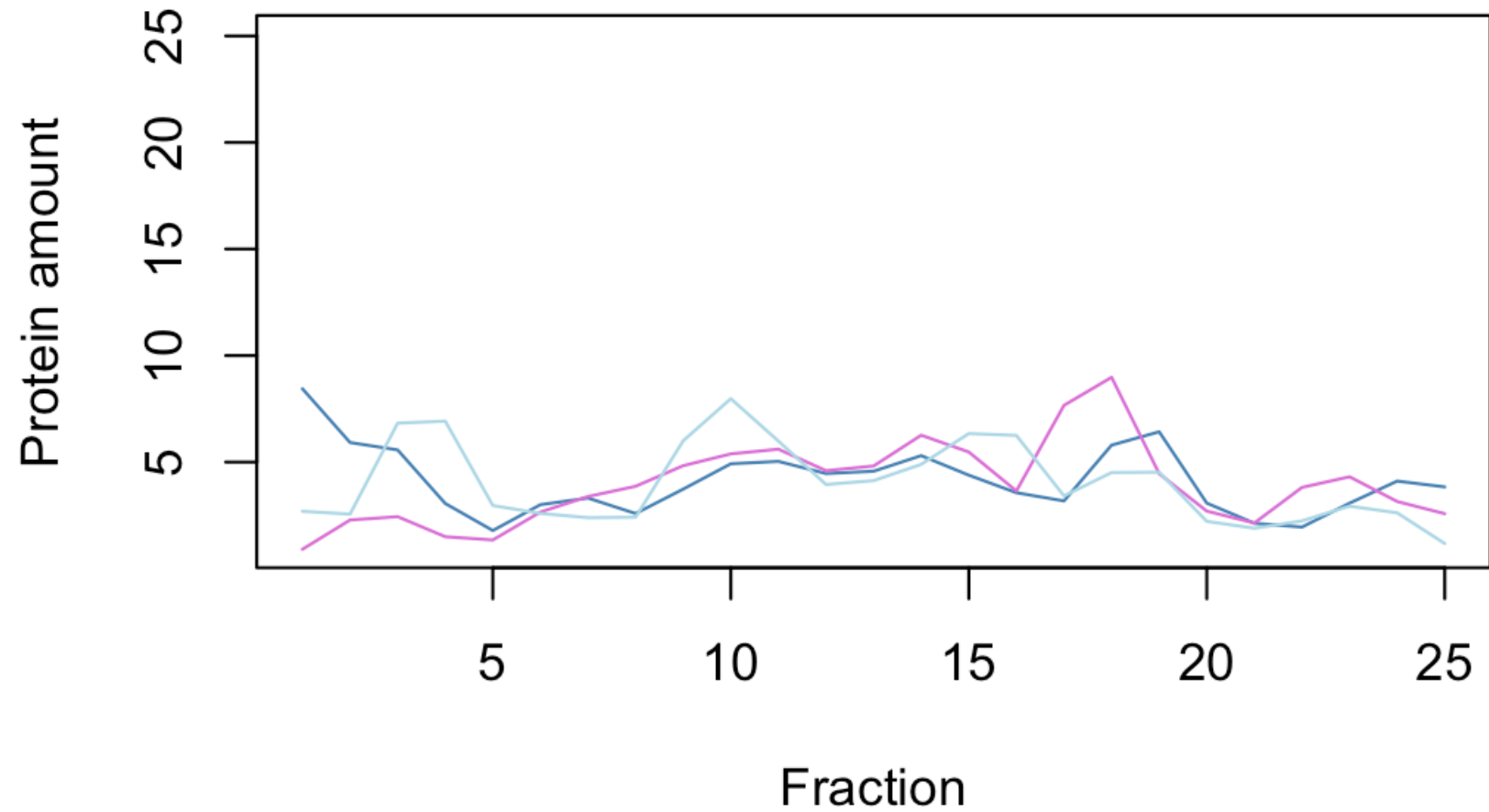
Correlation

Correlation
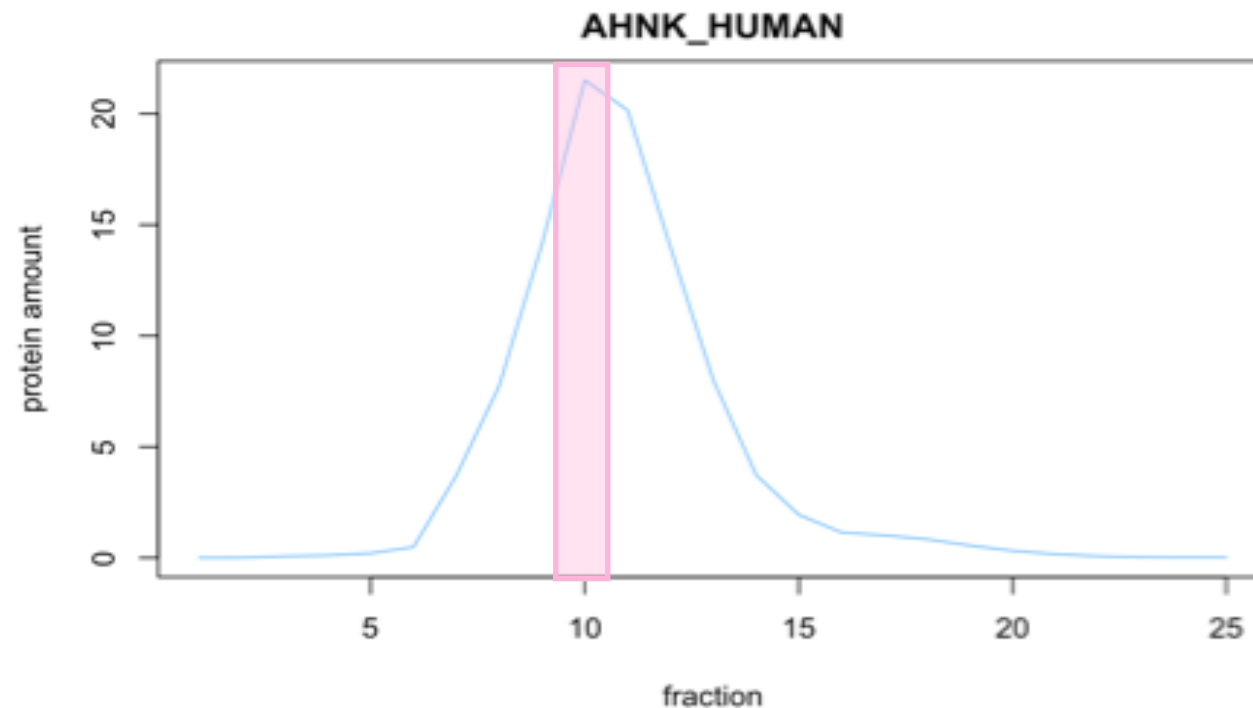

Correlation > 0,8 ➡ Keep protein

Correlation
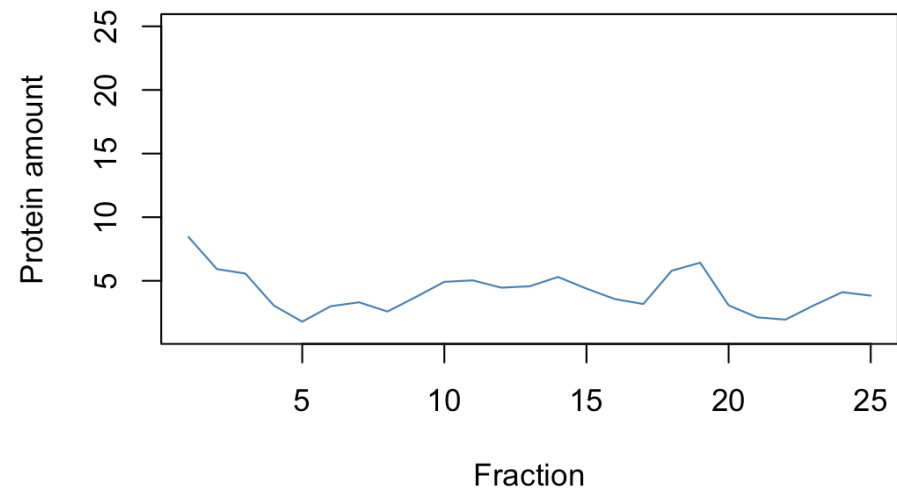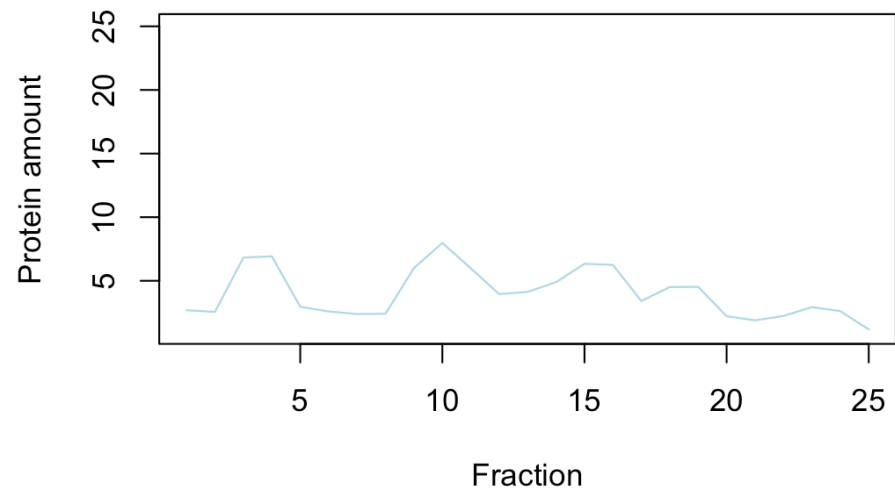
Correlation

Correlation < 0,8   →   Delete protein

Identification of global maxima

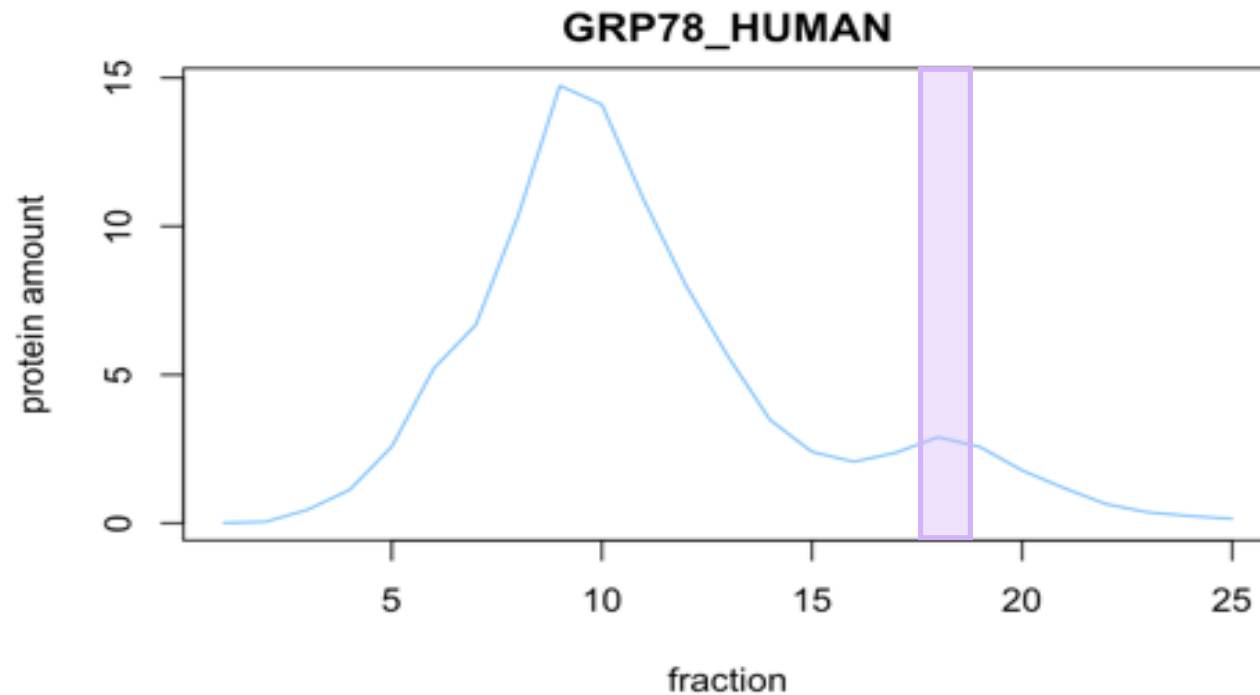|	Fraction 7	Fraction 8	Fraction 9	Fraction 10	Fraction 11
Protein amount	5	15	19	22	17

AHNK_HUMAN

Max function:

Iterates trough each row comparing the values

Identification of local maxima

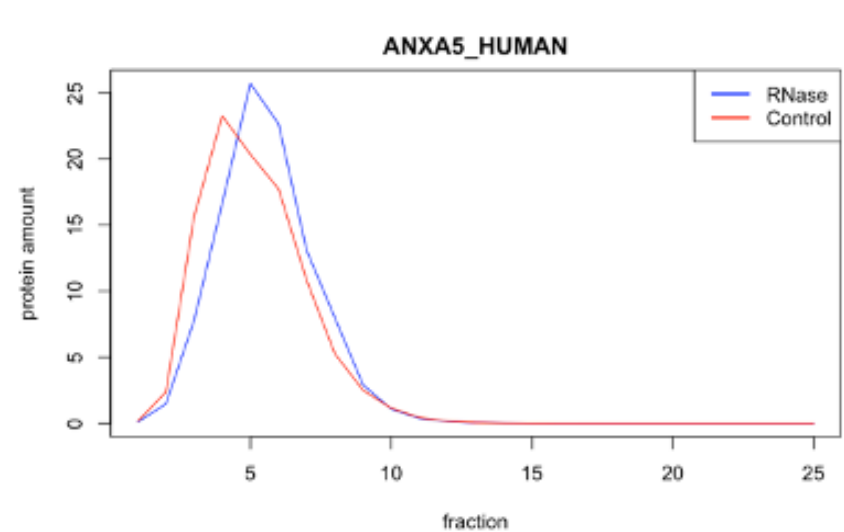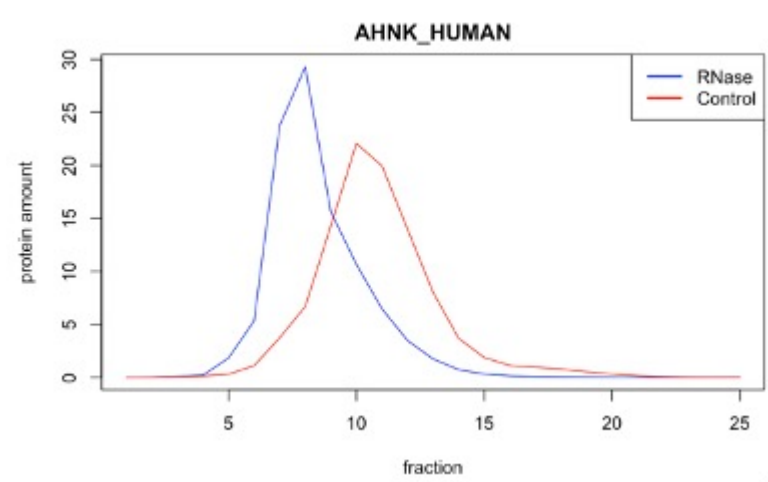|	Fraction 15	Fraction 16	Fraction 17	Fraction 18	Fraction 19
Protein amount	2.3	2.1	3	2.7	2.5

GRP78_HUMAN

Fraction x is local peak if:

$x > x - 1$
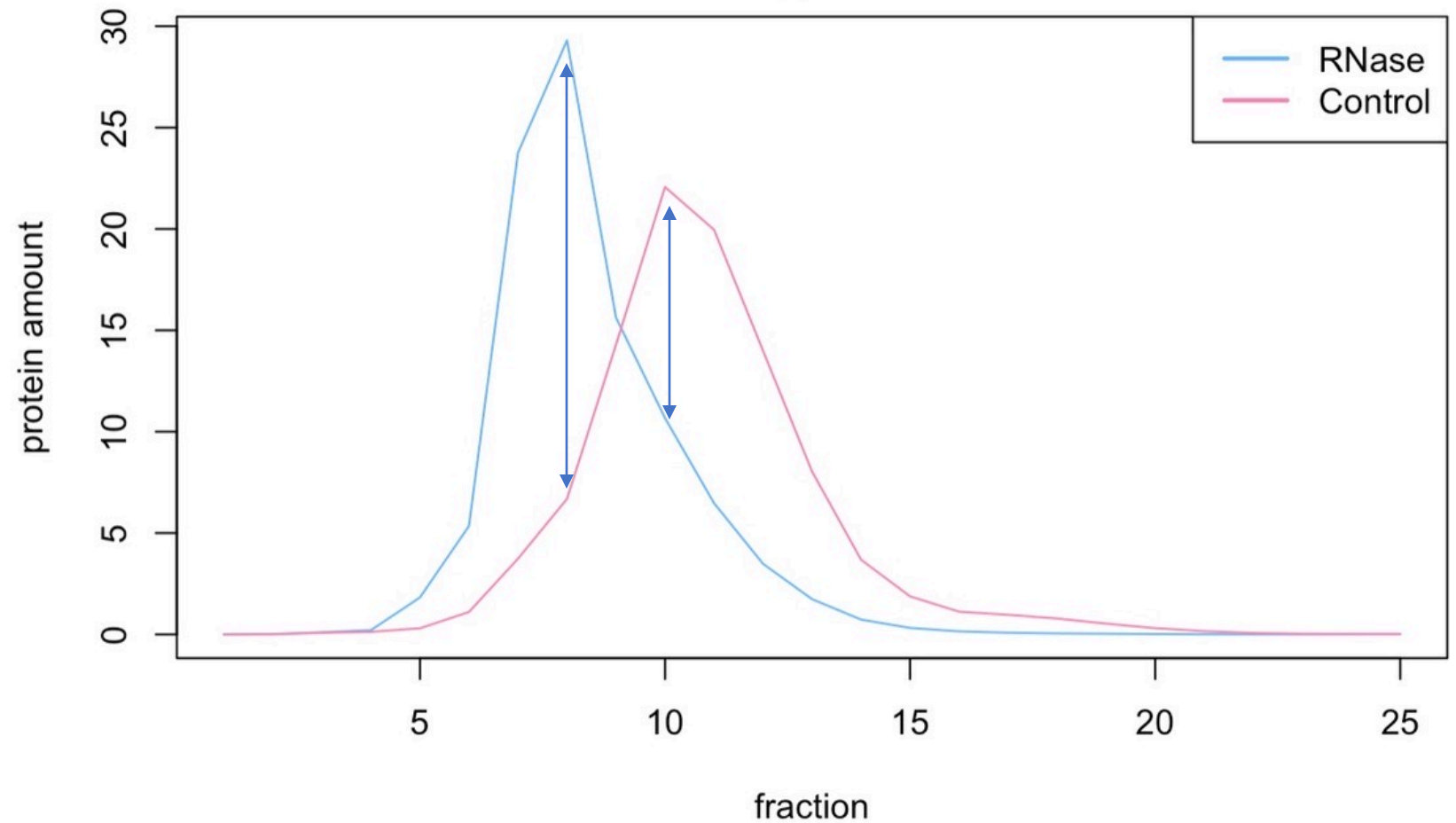
$x > x - 2$

$x > x + 1$

$x > x + 2$

Identifying shifts
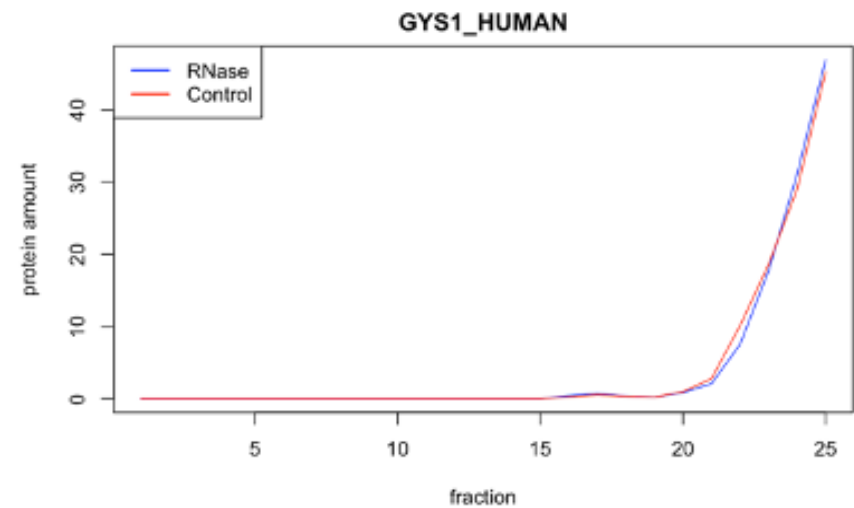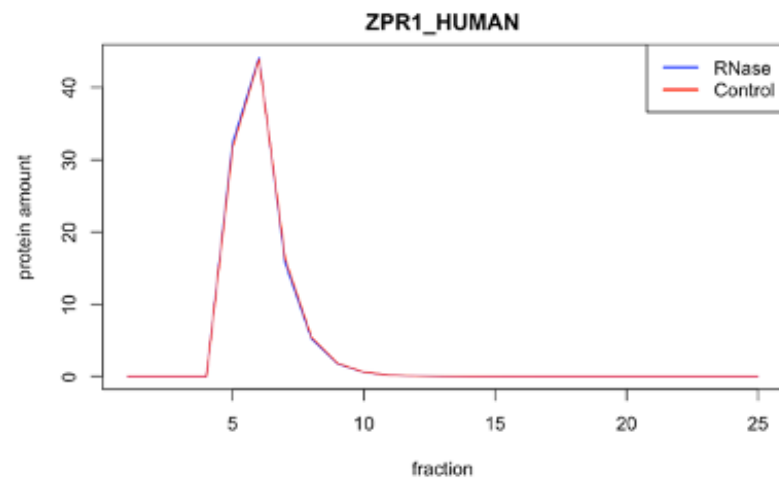
T-Test

amount of protein in global peak fraction

null hypothesis:
difference between amount of protein is zero

T-Test

P-value:

amount of protein in global peak fraction

null hypothesis:
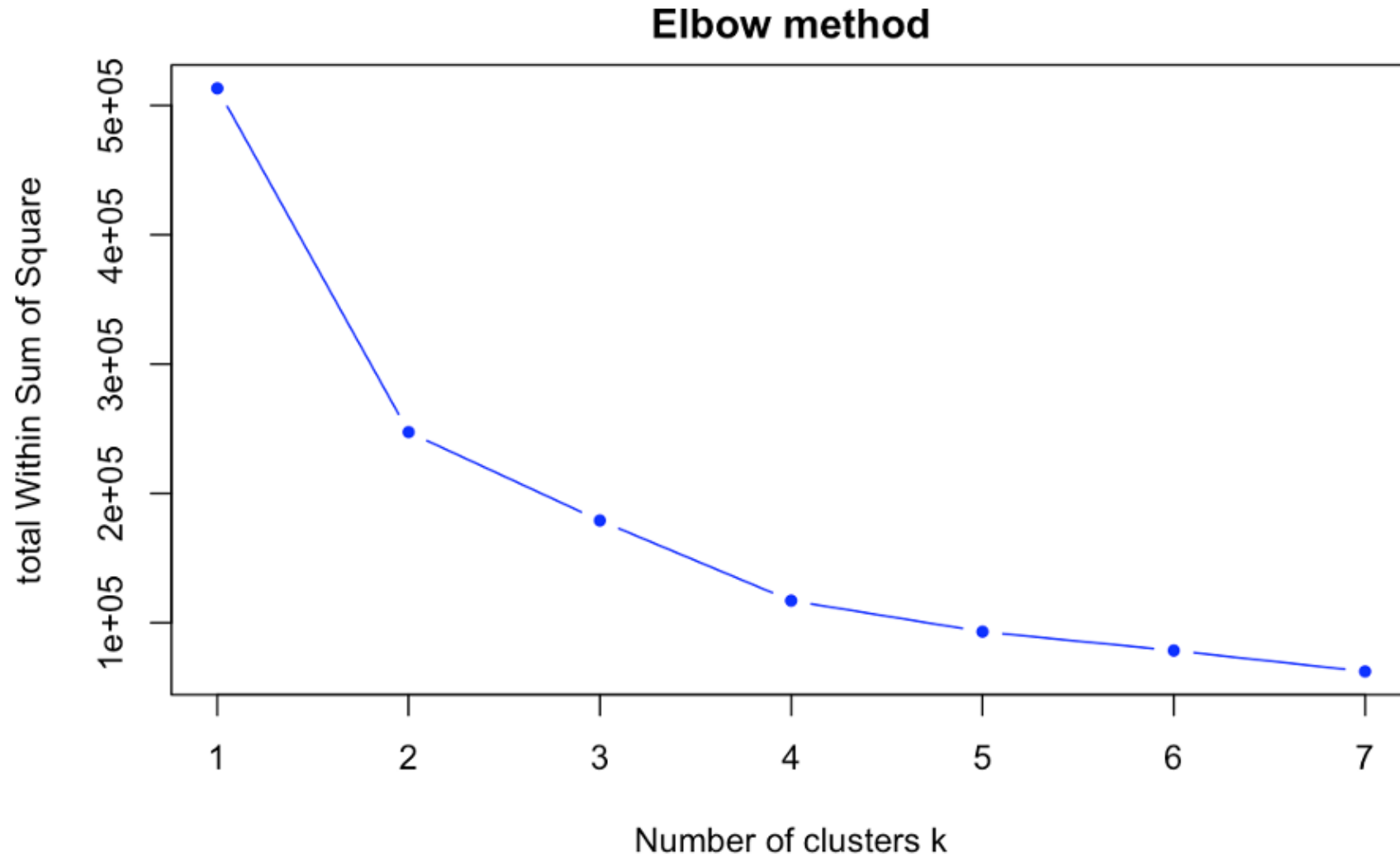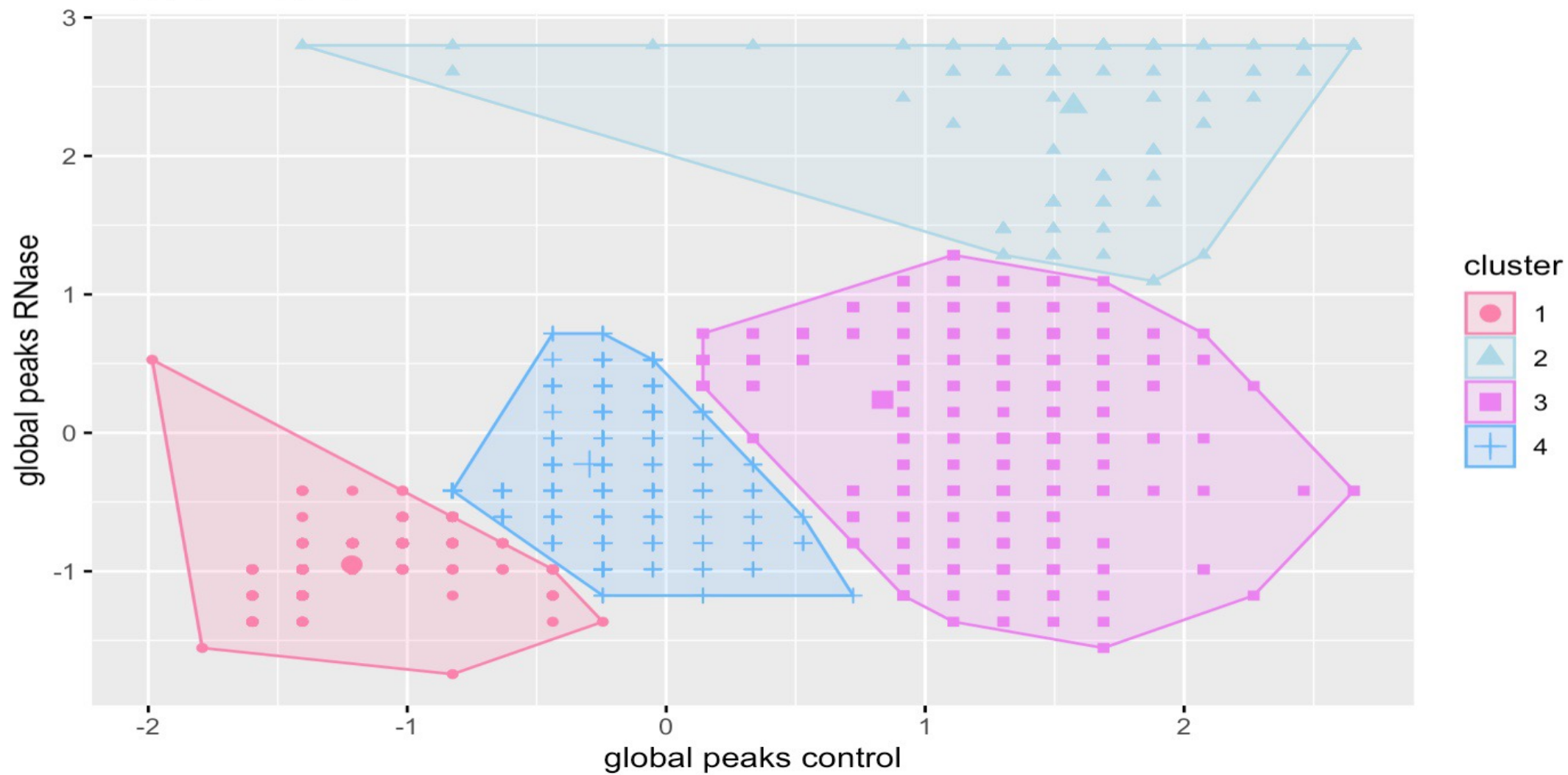difference between amount of protein is zero
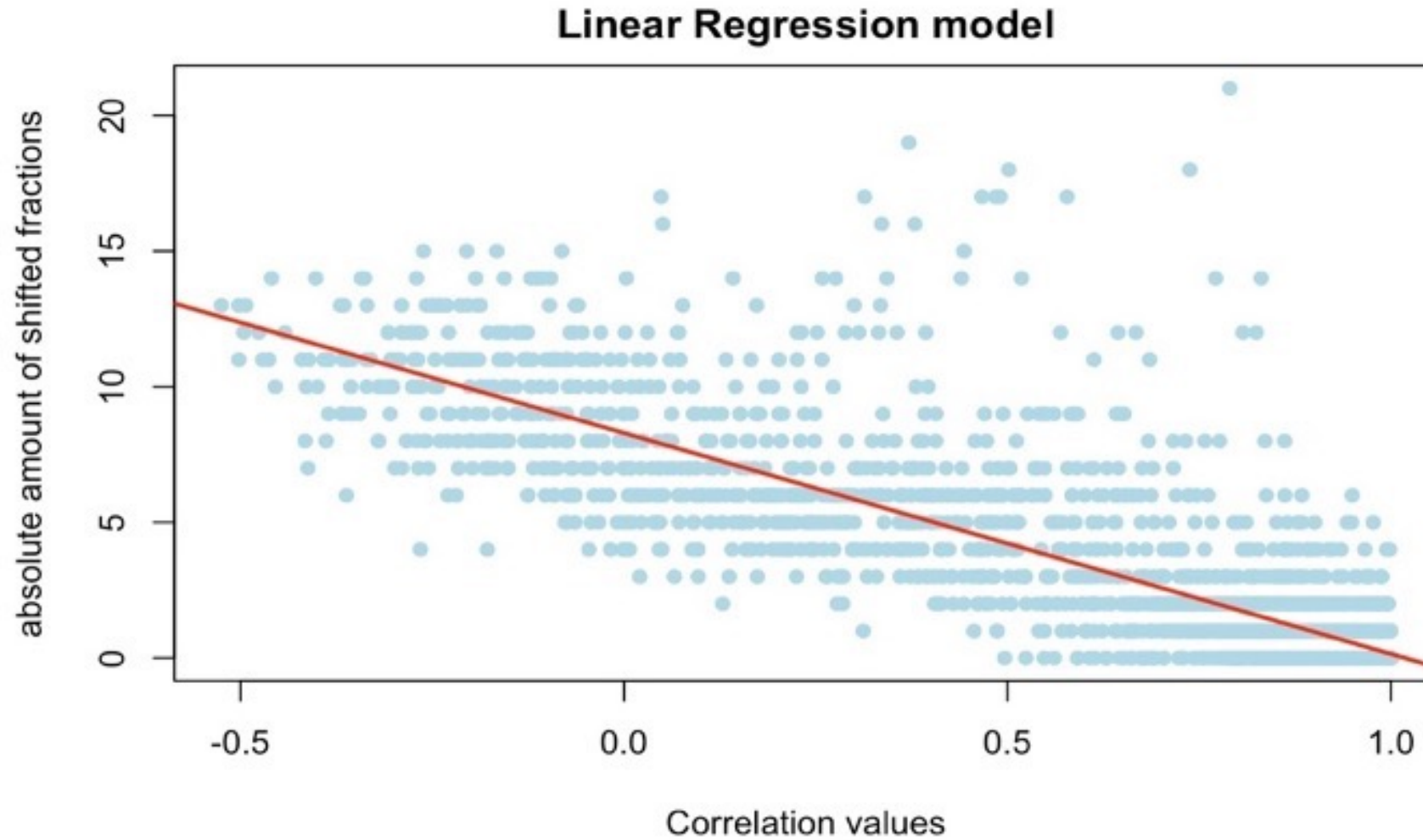
>0.05 non-significant

<0.05 significant

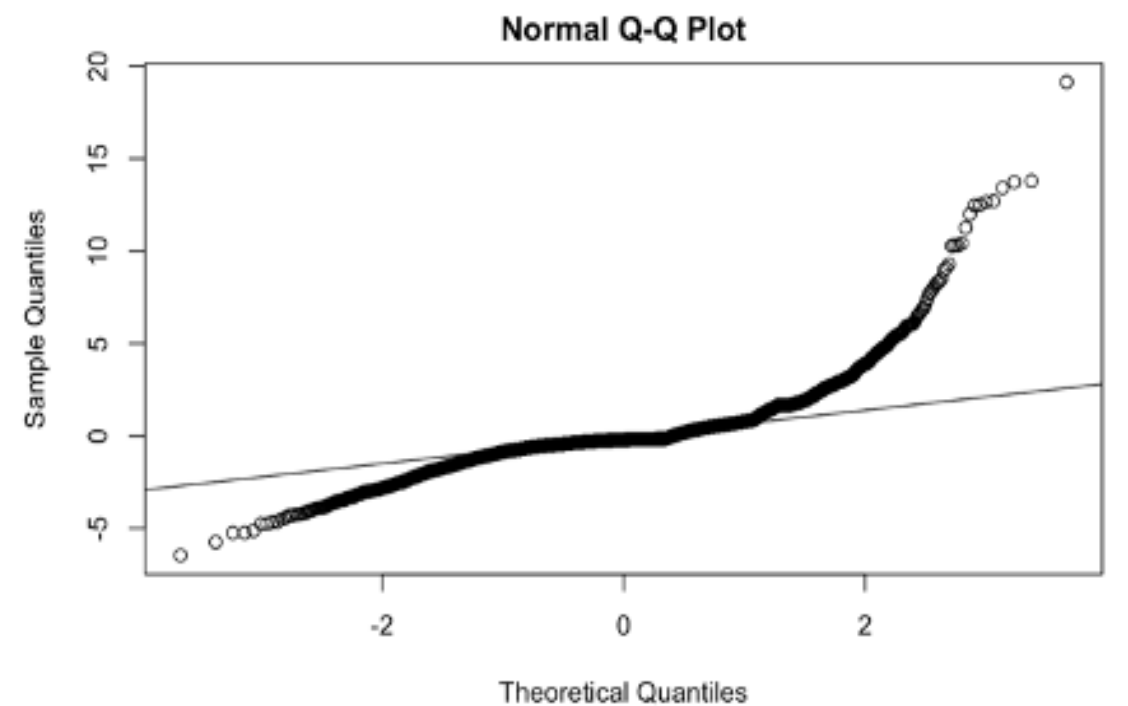# K-means – elbow method

Elbow method

K-means

Linear regression

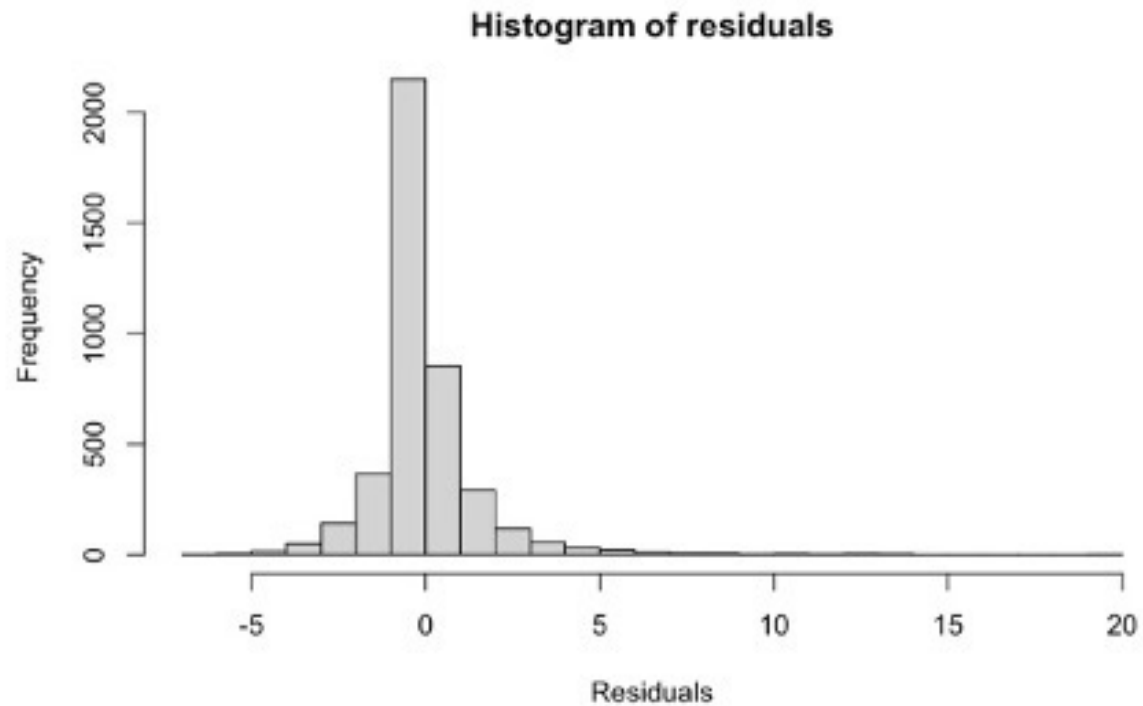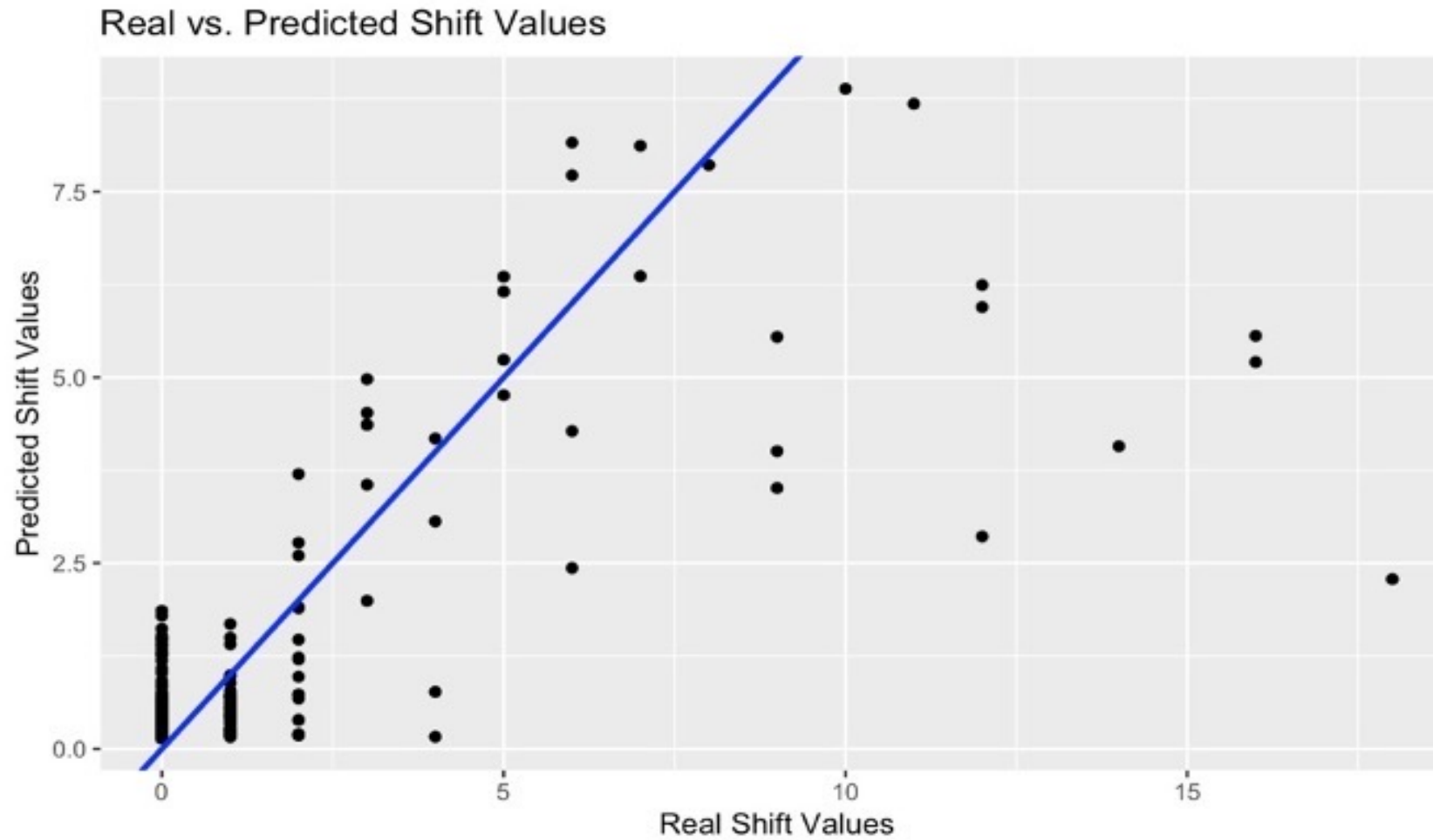# Linear Regression – compute data frame

Independent -> correlation

Dependent -> shifts

protein_name	correlation	shift
AHNK_HUMAN	0.443	2
PRKDC_HUMAN	0.902	0
ACTB_HUMAN	0.787	2
ACTG_HUMAN	0.787	2
ACTC_HUMAN	0.789	2
ACTS_HUMAN	0.789	2
ACTA_HUMAN	0.792	2
ACTH_HUMAN	0.792	2
CPSM_HUMAN	0.829	1
GRP78_HUMAN	0.801	1

Residuals

Prediction

Real vs. Predicted Shift Values

Comparison to other databases

Comparison to RBP2GO

false positive rate = 7.8%
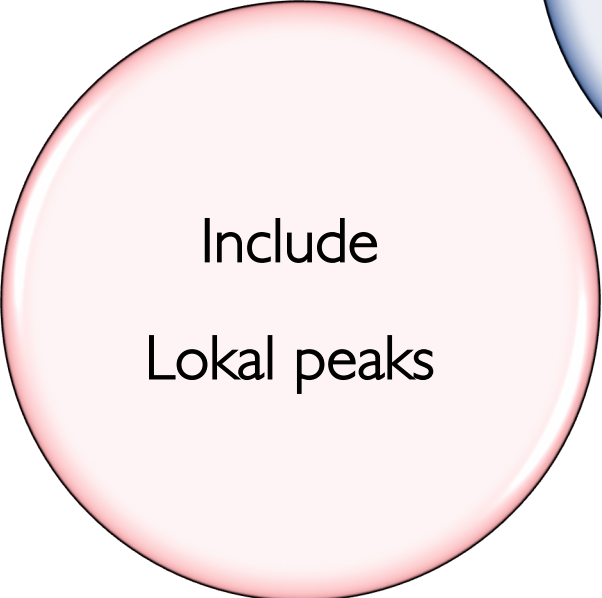false negative rate = 65.9 %

test sensitivity = 34.0%
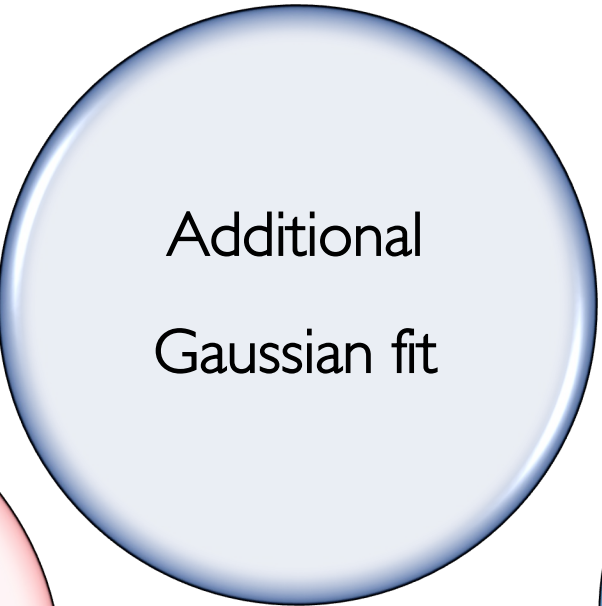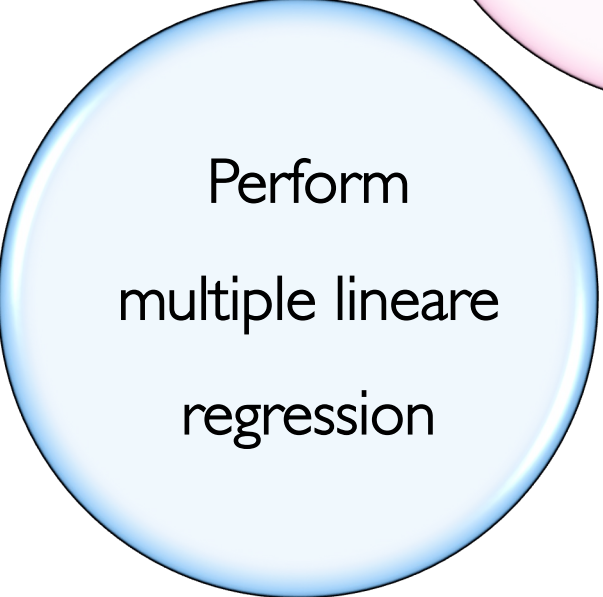test specificity = 92.2 %
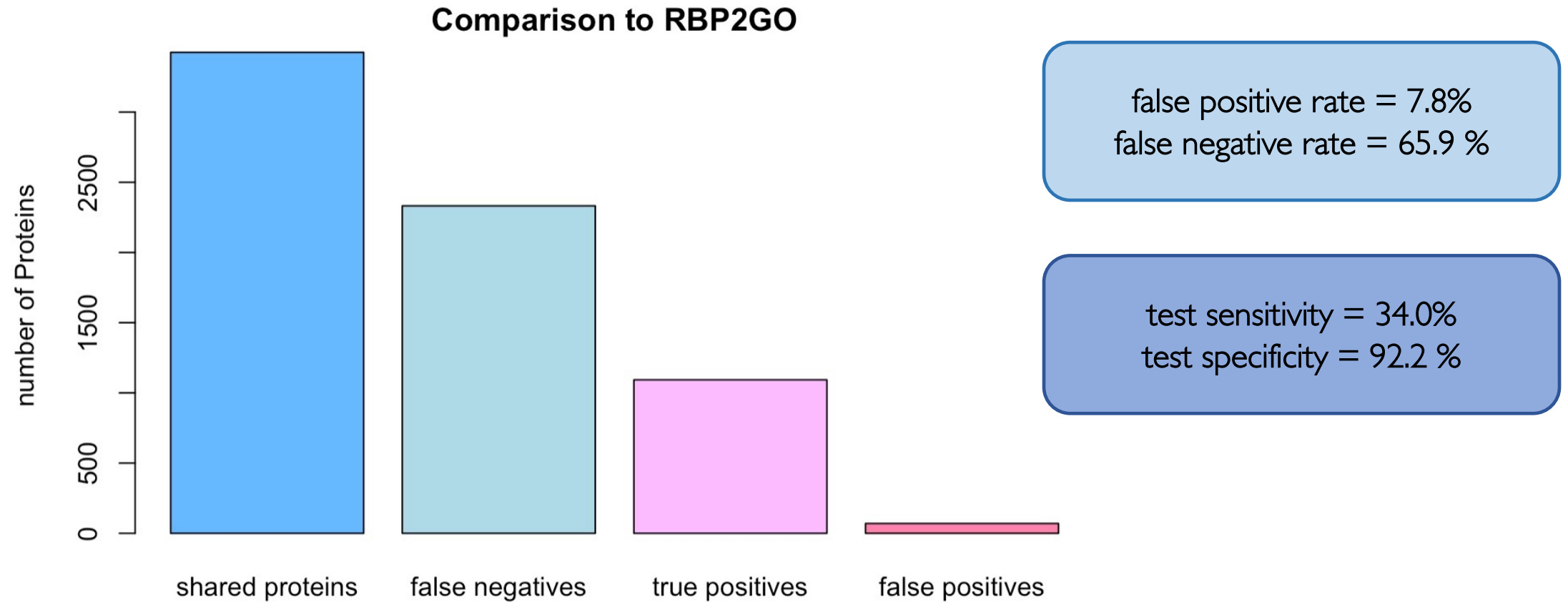
Discussion

Include
Lokal peaks

Additional
Gaussian fit

Perform
multiple lineare
regression

Adjust
Significance
level t-test