Computational Meeting: Relative Abundance Correction

Aaron Fehr

2023-04-24

The Following Script contains a script and tests thereof for the correction of peptide intensities for relative abundance changes in the protein concentration.It was specifically written for LiP data but can also be applied to PTM data.The function was written based on the approach described by Devon Kohler et al. in https://doi.org/10.1016/j.mcpro.2022.100477 and implemented in the R packages "MSstatsPTM" and "MSStatsLiP"

In brief, the relative change on the peptide level is corrected for by division with the relative change on the protein level. For the statistical test, the pooled standard deviation as well as the tvalue is calculated. The effective degree of freedom are calculated using the Satterthwaite approximation in order to take into account that the protein and peptide intensities are not fully independent measurements.

As input, the script takes two differential abundance tables on peptide and protein level. They can be obtained by using the "calculate_diff_abundance()" function in the Protti package.

The peptide table has to contain the columns:

1. Protein names
2. Peptide id's
3. log2 fold change
4. Standard error
5. Number of observations

The protein table has to contain the columns:

1. Protein names
2. log2 fold change
3. Standard error
4. Number of observations

The Data used for this demonstration is LiP-School data (Rapamycin treated HeLa lysate, quadruplicates) which was generated together with Ludovic Gillet. It was analysed using MSstatsLiP with the "LiP-MS_data_analysis_case_control" script in https://github.com/PicottiGroup/ Data-analysis-of-LiP-MS-data-for-high-throughput-applications.git without imputation by Valentina Capelletti.

```
# Import of data:

MSstats_LiP_raw = read.csv("./CM_Data/LiPschool_msstats_model_RAW.csv")

MSstats_Trp_raw = read.csv("./CM_Data/LiPschool_msstats_model_PROTEIN.csv")

MSstats_Adj_raw = read.csv("./CM_Data/LiPschool_msstats_model_ADJUSTED.csv")
```

```r
# Adapt columns so that they match protti output from calculate_diff_abundance()

MSstats_LiP = MSstats_LiP_raw  %>%
  dplyr::select(ProteinName,
                PeptideSequence,
                std_error = SE,
                diff = log2FC,
                n_obs = DF) %>%
  mutate(n_obs = n_obs + 2)

glimpse(MSstats_LiP)
```

```
## Rows: 28,211
## Columns: 5
## $ ProteinName     <chr> "A0AVT1", "A0AVT1", "A0AVT1", "A0AVT1", "A0AVT1", "A0A~
## $ PeptideSequence <chr> "AVTIHDTEK", "DGSLFWQSPK", "DLGTNFFLSEDDVVNKR", "EDFTL~
## $ std_error       <dbl> 0.20116614, 0.09968147, 0.12518565, 0.31508866, 0.3786~
## $ diff            <dbl> -0.116537415, 0.044599388, 0.008535136, 0.031606809, -~
## $ n_obs           <dbl> 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, ~
```

```r
MSstats_Trp = MSstats_Trp_raw %>%
  dplyr::select(ProteinName = Protein,
                std_error = SE,
                diff = log2FC,
                n_obs = DF) %>%
  mutate(n_obs = n_obs + 2)


glimpse(MSstats_Trp)
```

```
## Rows: 2,590
## Columns: 4
## $ ProteinName <chr> "A0AVT1", "A0FGR8", "A0MZ66", "A1X283", "A2VDF0", "A5YKK6"~
## $ std_error   <dbl> 0.07190306, 0.13238116, 0.04134834, 0.05634862, 0.22927746~
## $ diff        <dbl> 0.0002369896, -0.0199183725, -0.0324484028, -0.0415830347,~
## $ n_obs       <dbl> 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8~
```

```r
# Adapt column names for MSstats adjusted table to match the function output
MSstats_Adj = MSstats_Adj_raw %>%
  dplyr::select(ProteinName,
                PeptideSequence,
                adj_diff = log2FC,
                adj_std_error = SE,
                df = DF,
                tval = Tvalue,
                pval = pvalue, p.adj =
                  adj.pvalue)

glimpse(MSstats_Adj)
```

```
## Rows: 28,211
## Columns: 8
```

```
## $ ProteinName     <chr> "A0AVT1", "A0AVT1", "A0AVT1", "A0AVT1", "A0AVT1", "A0A~
## $ PeptideSequence <chr> "AVTIHDTEK", "DGSLFWQSPK", "DLGTNFFLSEDDVVNKR", "EDFTL~
## $ adj_diff        <dbl> -0.1167744043, 0.0443623984, 0.0082981468, 0.031369819~
## $ adj_std_error   <dbl> 0.21363021, 0.12290828, 0.14436584, 0.32318866, 0.3854~
## $ df              <dbl> 7.508464, 10.913544, 9.570259, 6.623210, 6.432095, 7.7~
## $ tval            <dbl> -0.546619335, 0.360939044, 0.057479986, 0.097063490, -~
## $ pval            <dbl> 0.60050121, 0.72503450, 0.95534520, 0.92554742, 0.4962~
## $ p.adj           <dbl> 0.9991272, 0.9991272, 0.9991272, 0.9991272, 0.9991272,~
```

```
# Manual correction for relative abundance changes:

ErrorPropagationTest = MSstats_LiP %>%
  # Join Tryptic data
  inner_join(MSstats_Trp,
             by = c("ProteinName"), suffix = c("_pep","_prot")) %>%
  # Adjust log2FC
  mutate(adj_diff = diff_pep-diff_prot) %>%
  # Perform error propagation of standard error
  mutate(adj_std_error = sqrt(std_error_pep**2 + std_error_prot**2)) %>%
  # Calculate numerator of Satterthwaite equation
  mutate(numer = (std_error_pep**2 + std_error_prot**2)**2) %>%
  # Calculate denominator of Satterthwaite equation
  mutate(denom = (std_error_pep**4/(n_obs_pep-2) +
                    std_error_prot**4/(n_obs_prot-2))) %>%
  # Calculate DF with Satterthwaite equation
  mutate(df = numer/denom) %>%
  # Calculate tvalue
  mutate(tval = adj_diff/adj_std_error) %>% # Calculate Tvalue
  #Caluculate pvalue
  mutate(pval = 2*stats::pt(abs(tval), df, lower.tail = FALSE)) %>%
  # Calculate adj. pvalue using Benjamini Hochberg
  mutate(p.adj = p.adjust(pval, method = "BH")) %>%
  #Join MSstats Adjusted data table
  left_join(MSstats_Adj, by = c("ProteinName", "PeptideSequence"),
             suffix = c("_Function","_MSstats"))


# Count identical values
# ErrorPropagationTest has 27931 rows. 2 peptides have a change of -Inf

# Degrees of freedom
sum(round(ErrorPropagationTest$df_Function,5) ==
      round(ErrorPropagationTest$df_MSstats,5), na.rm = TRUE)
```

```
## [1] 27929
```

```
# adjusted Log2FC
sum(ErrorPropagationTest$adj_diff_Function ==
      ErrorPropagationTest$adj_diff_MSstats, na.rm = TRUE)
```

```
## [1] 27931
```

```r
# Standard error
sum(ErrorPropagationTest$adj_std_error_Function ==
      ErrorPropagationTest$adj_std_error_MSstats, na.rm = TRUE)
```

```
## [1] 27929
```

```r
# T-value
sum(ErrorPropagationTest$tval_Function ==
      ErrorPropagationTest$tval_MSstats, na.rm = TRUE)
```

```
## [1] 27929
```

```r
# P-value
sum(round(ErrorPropagationTest$pval_Function,5) ==
      round(ErrorPropagationTest$pval_MSstats,5), na.rm = TRUE)
```

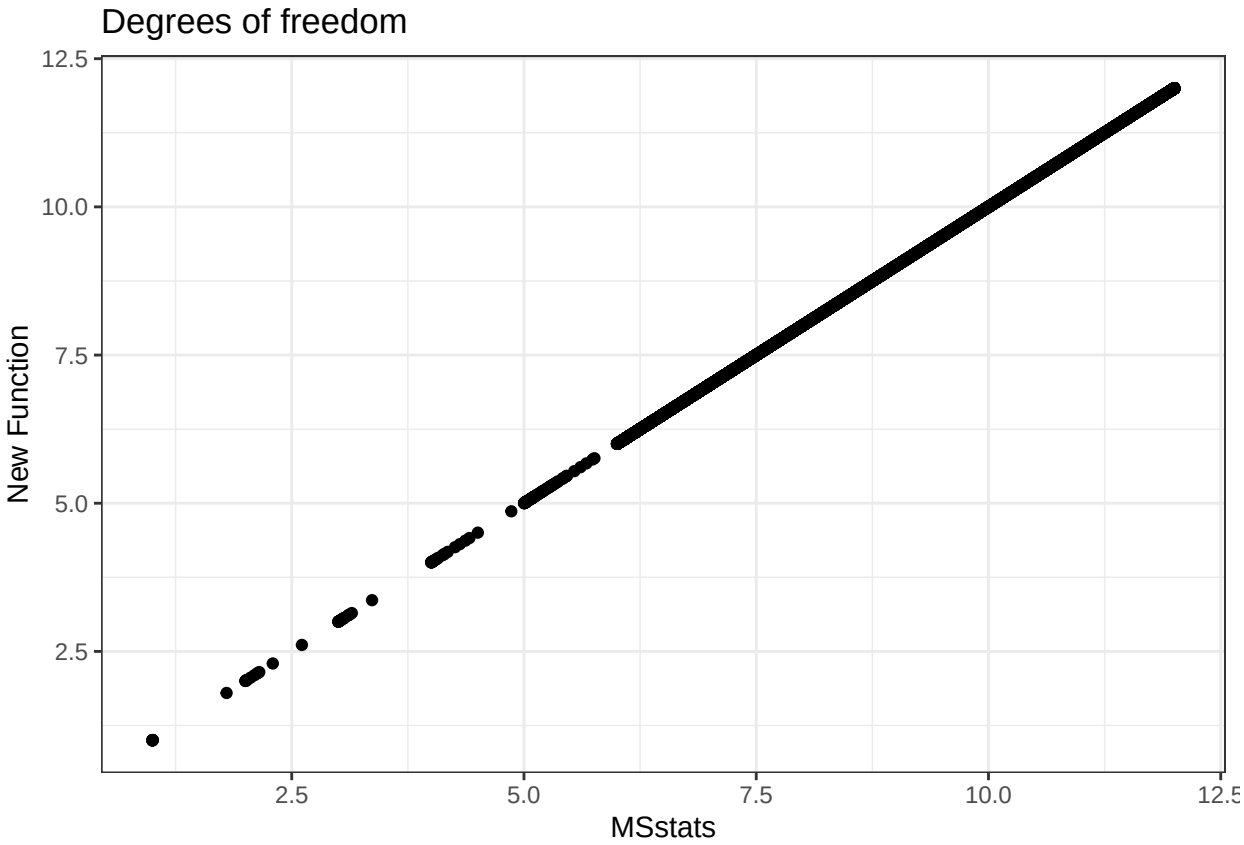
```
## [1] 27929
```

```r
# BH adjusted P-value
sum(round(ErrorPropagationTest$p.adj_Function,5) ==
      round(ErrorPropagationTest$p.adj_MSstats,5), na.rm = TRUE)
```

```
## [1] 27929
```

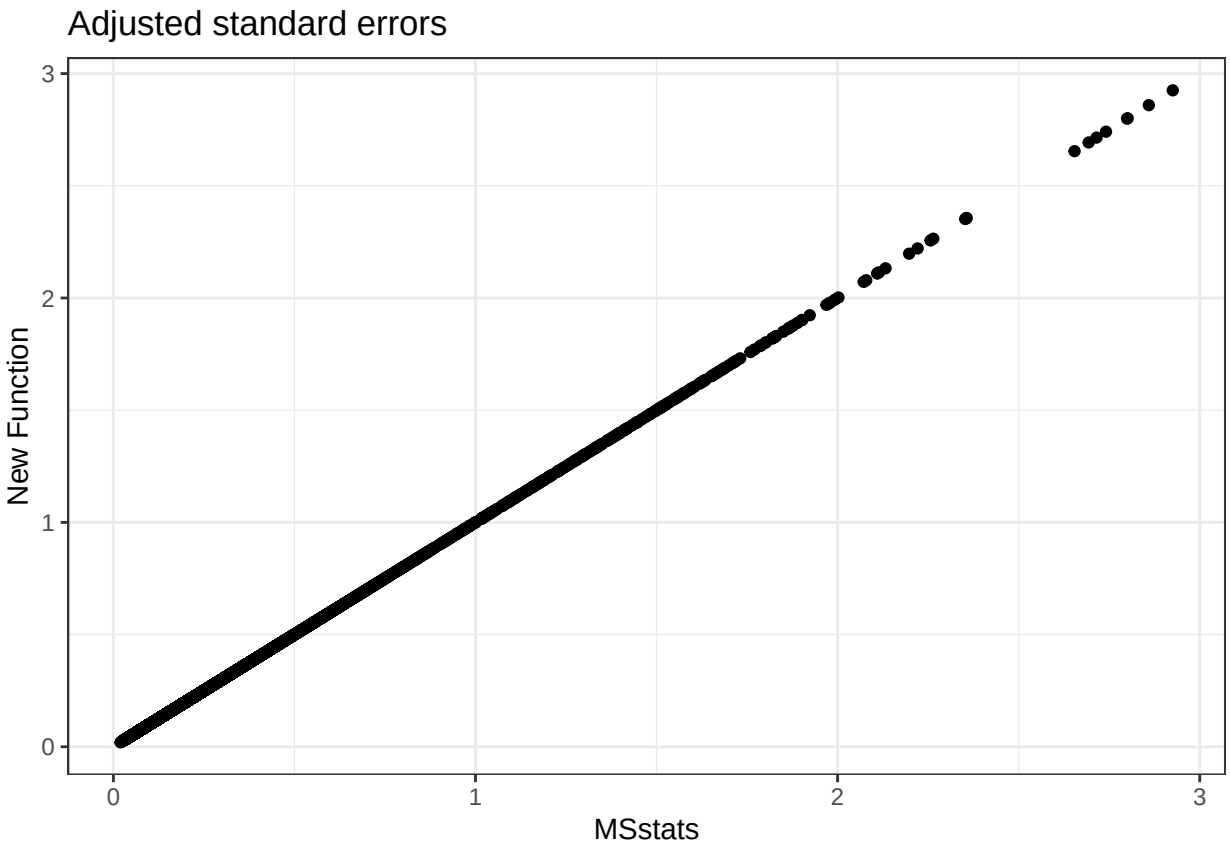
```r
# Plots of MSstats vs New function

ErrorPropagationTest %>%
  ggplot(aes(x=df_MSstats,y=df_Function)) +
  geom_point() +
  theme_bw() +
  labs(title = "Degrees of freedom", x = "MSstats", y = "New Function")
```

```
## Warning: Removed 2 rows containing missing values (‘geom_point()‘).
```

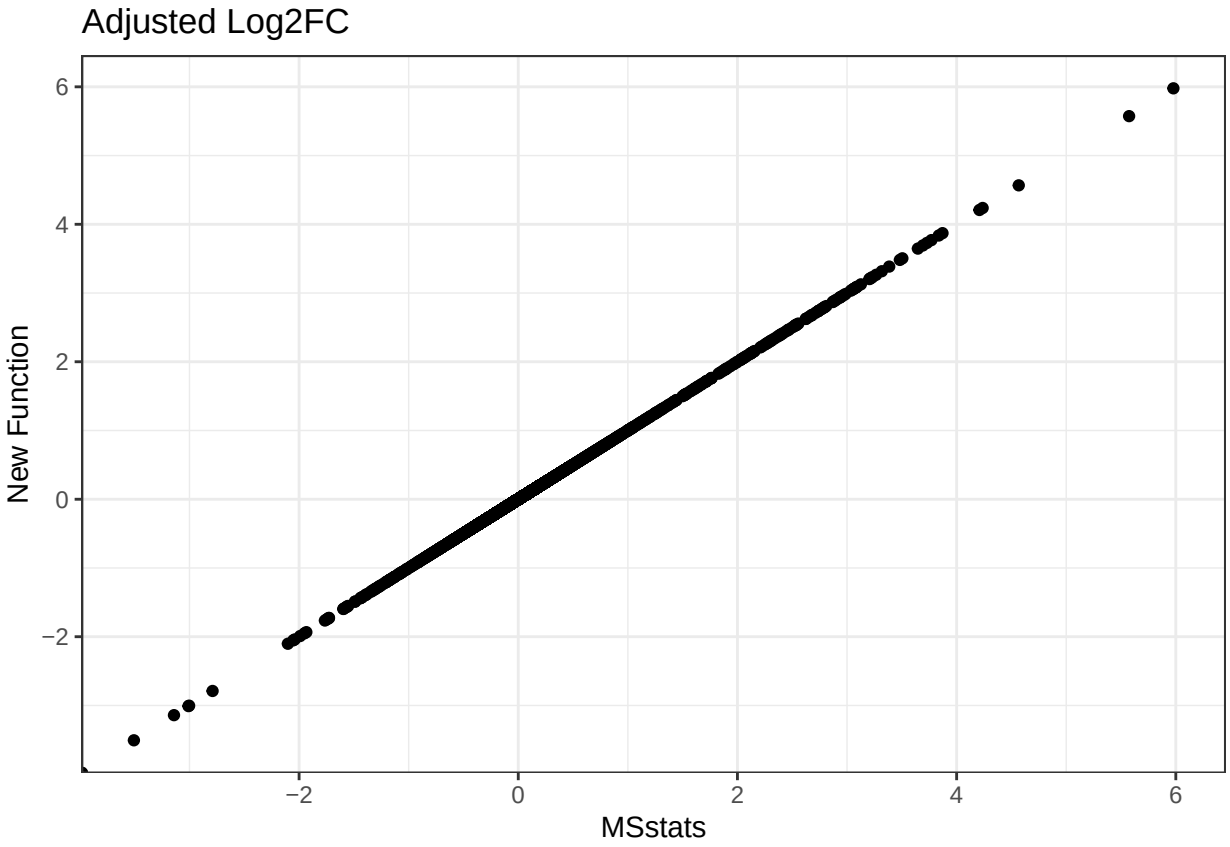



```
ErrorPropagationTest %>%
  ggplot(aes(x=adj_std_error_MSstats,y=adj_std_error_Function)) +
  geom_point() +
  theme_bw() +
  labs(title = "Adjusted standard errors", x = "MSstats", y = "New Function")
```

```
## Warning: Removed 2 rows containing missing values (‘geom_point()‘).
```

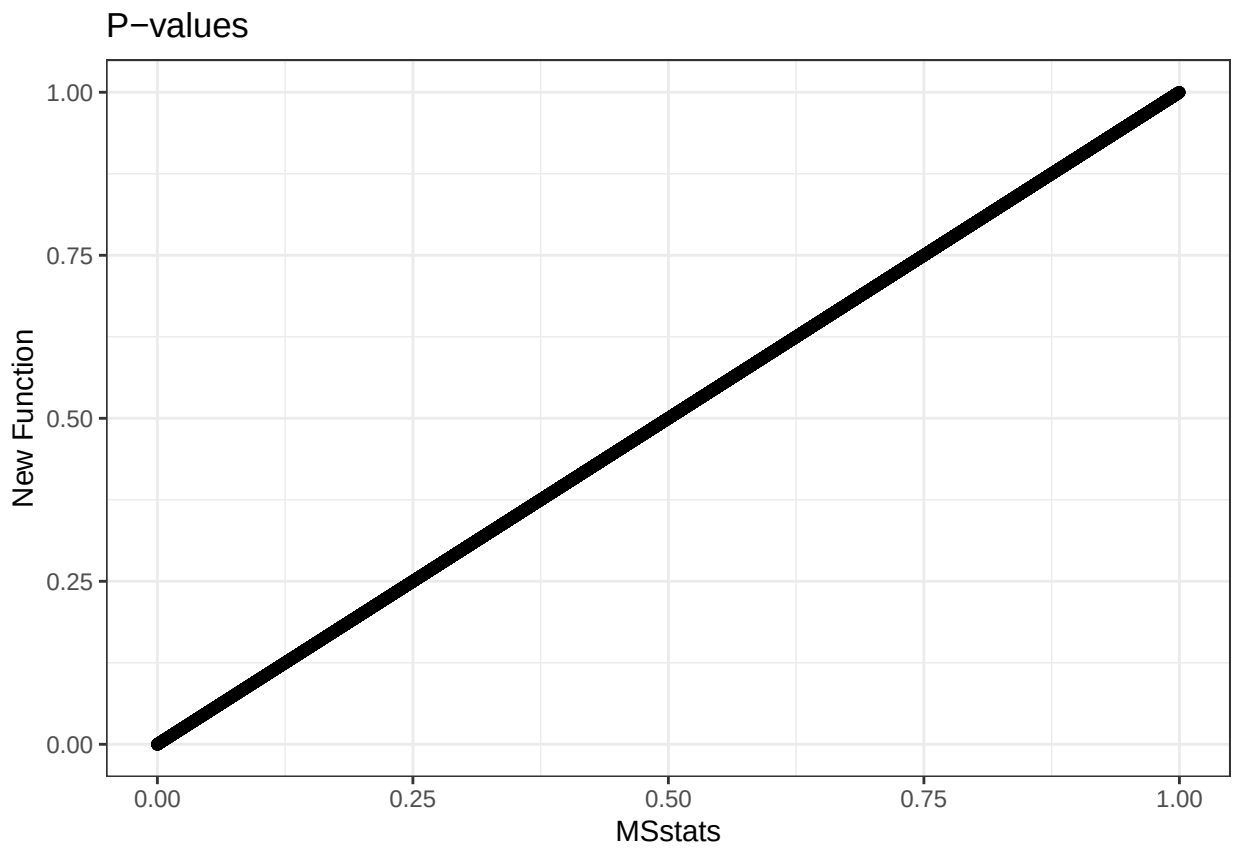



```
ErrorPropagationTest %>%
  ggplot(aes(x=adj_diff_MSstats, y=adj_diff_Function)) +
  geom_point() +
  theme_bw() +
  labs(title = "Adjusted Log2FC", x = "MSstats", y = "New Function")
```



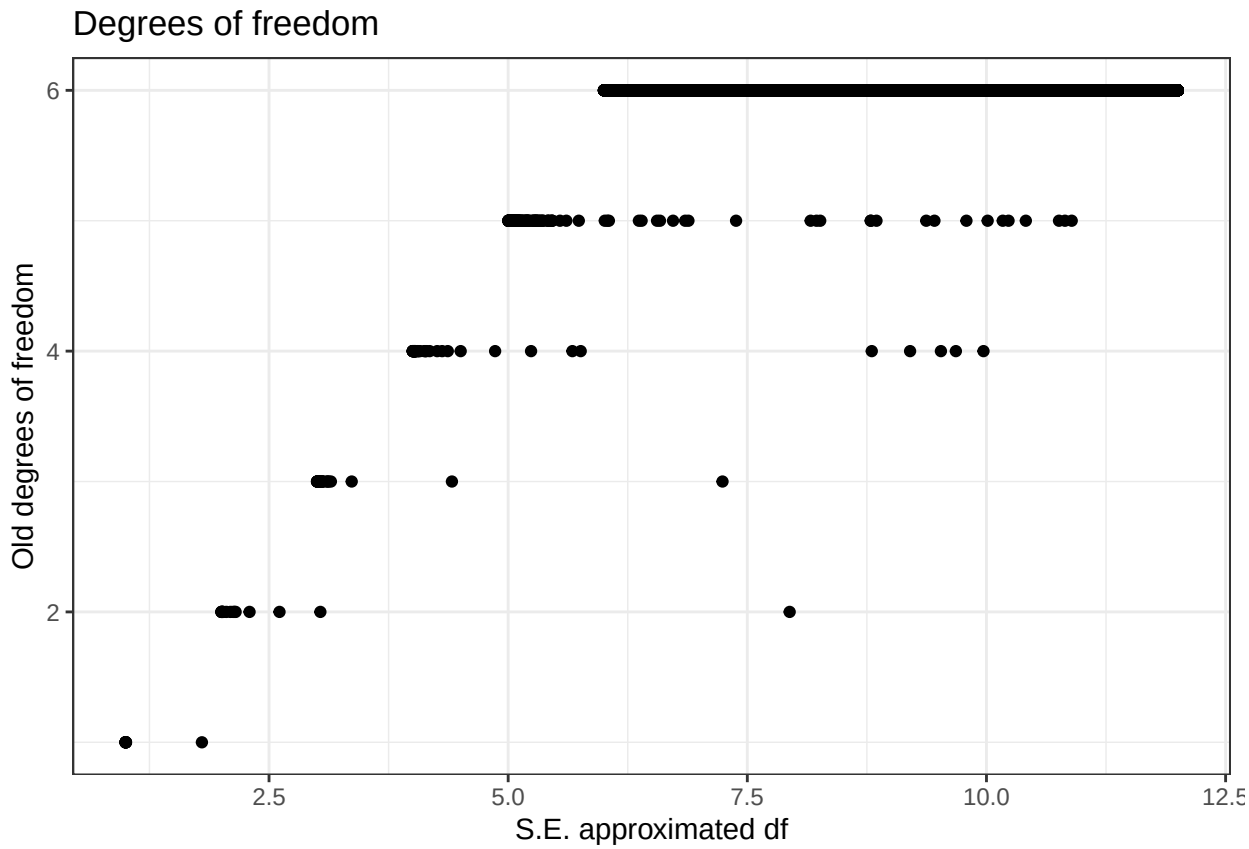

```
ErrorPropagationTest %>%
  ggplot(aes(x=pval_MSstats, y=pval_Function)) +
  geom_point() +
  theme_bw() +
  labs(title = "P-values", x = "MSstats", y = "New Function")
```

```
## Warning: Removed 2 rows containing missing values (‘geom_point()‘).
```



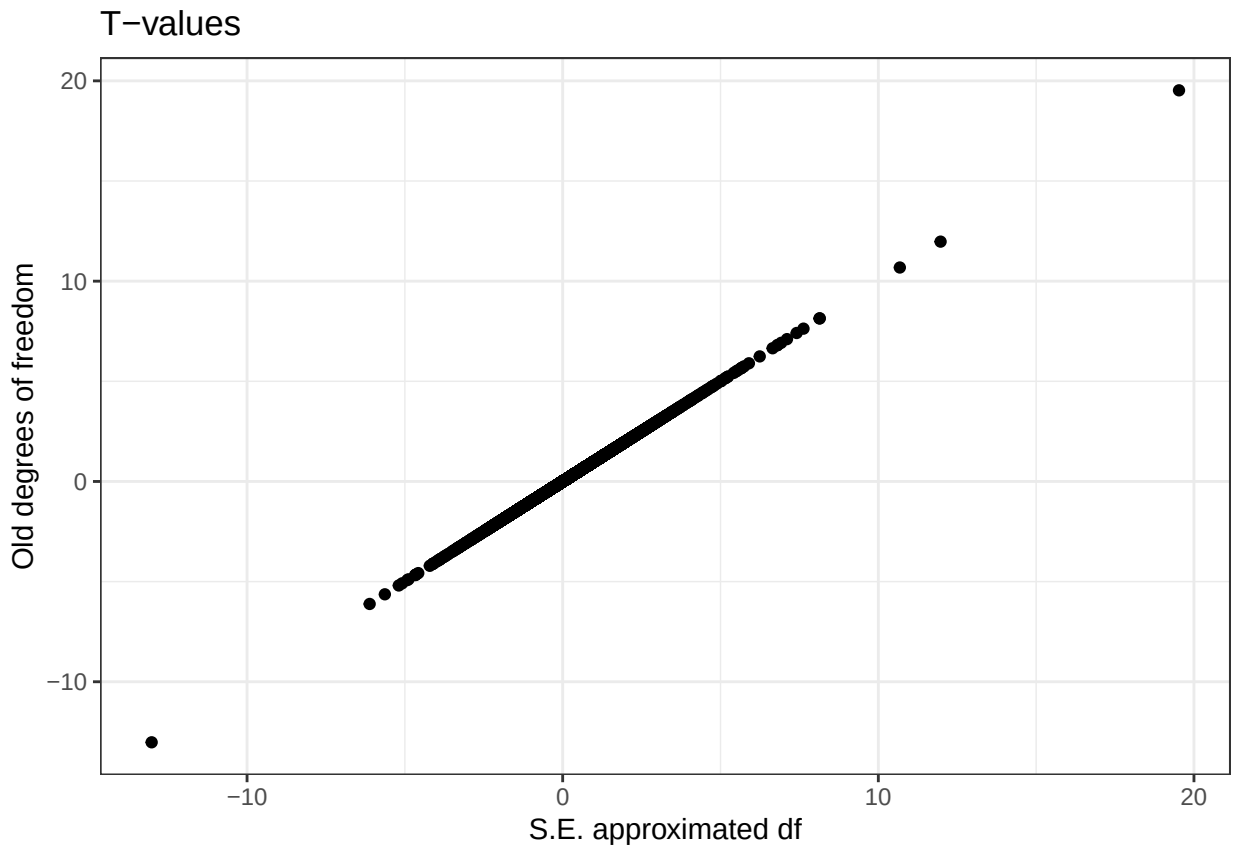

```
# Correction for abundance changes with normal calculation of degrees of freedom
# by calculatin the sum of observations -2

OldErrorPropagation = MSstats_LiP %>%
  inner_join(MSstats_Trp,
             by = c("ProteinName"), suffix = c("_pep","_prot")) %>%
  mutate(adj_diff = diff_pep-diff_prot) %>%
  mutate(adj_std_error = sqrt(std_error_pep**2 + std_error_prot**2)) %>%
  mutate(df = n_obs_pep - 2) %>% # NO SATTERTHWAITE APPROXIMATION
  mutate(tval = adj_diff/adj_std_error) %>%
  mutate(pval = 2*stats::pt(abs(tval), df, lower.tail = FALSE)) %>%
  mutate(p.adj = p.adjust(pval, method = "BH")) %>%
  left_join(MSstats_Adj, by = c("ProteinName", "PeptideSequence"),
             suffix = c("_Function","_MSstats"))



OldErrorPropagation %>%
  ggplot(aes(x=df_MSstats,y=df_Function)) +
  geom_point() +
  theme_bw() +
  labs(title = "Degrees of freedom", x = "S.E. approximated df", y = "Old degrees of freedom")
```

```
## Warning: Removed 2 rows containing missing values (‘geom_point()‘).
```

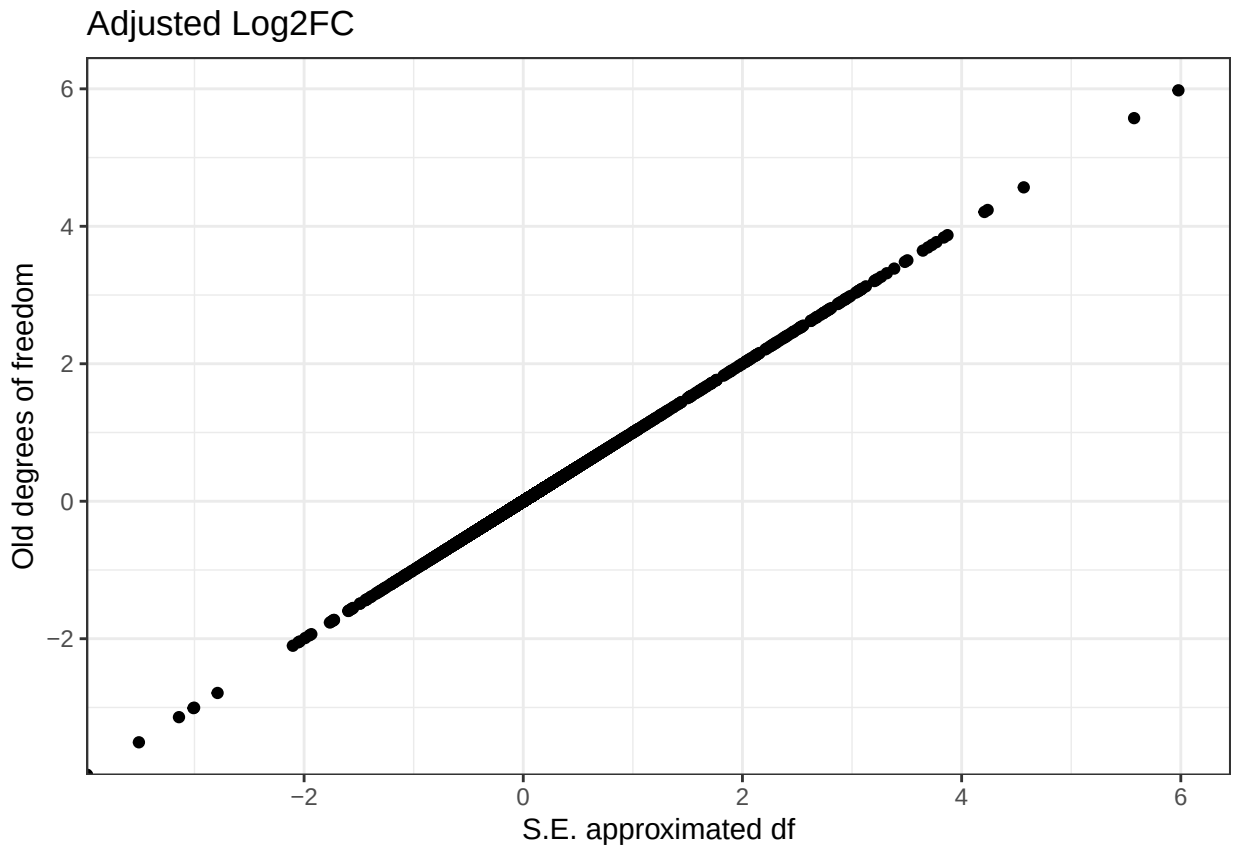



```
OldErrorPropagation %>%
  ggplot(aes(x=tval_MSstats, y=tval_Function)) +
  geom_point() +
  theme_bw() +
  labs(title = "T-values", x = "S.E. approximated df", y = "Old degrees of freedom")
```

```
## Warning: Removed 2 rows containing missing values (‘geom_point()‘).
```

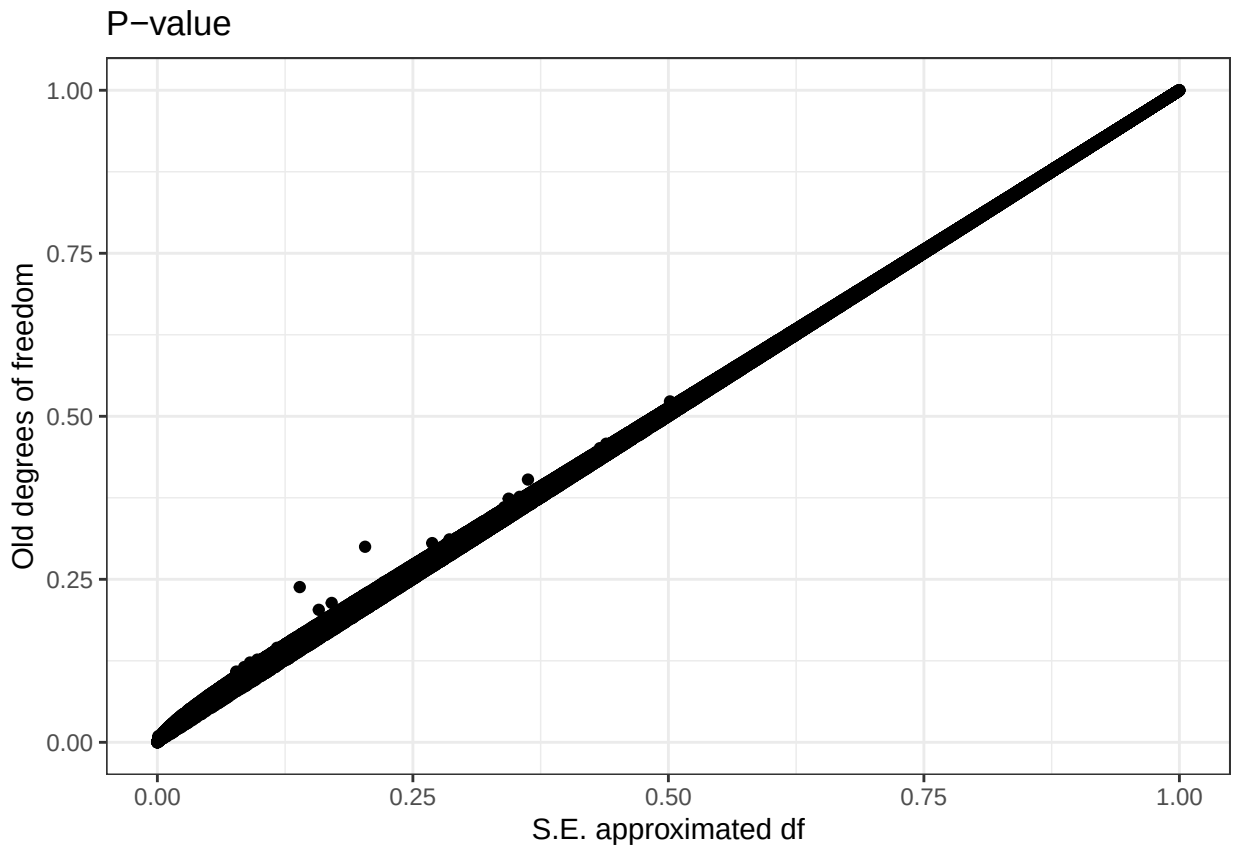



```
OldErrorPropagation %>%
  ggplot(aes(x=adj_diff_MSstats, y=adj_diff_Function)) +
  geom_point() +
  theme_bw() +
  labs(title = "Adjusted Log2FC", x = "S.E. approximated df", y = "Old degrees of freedom")
```



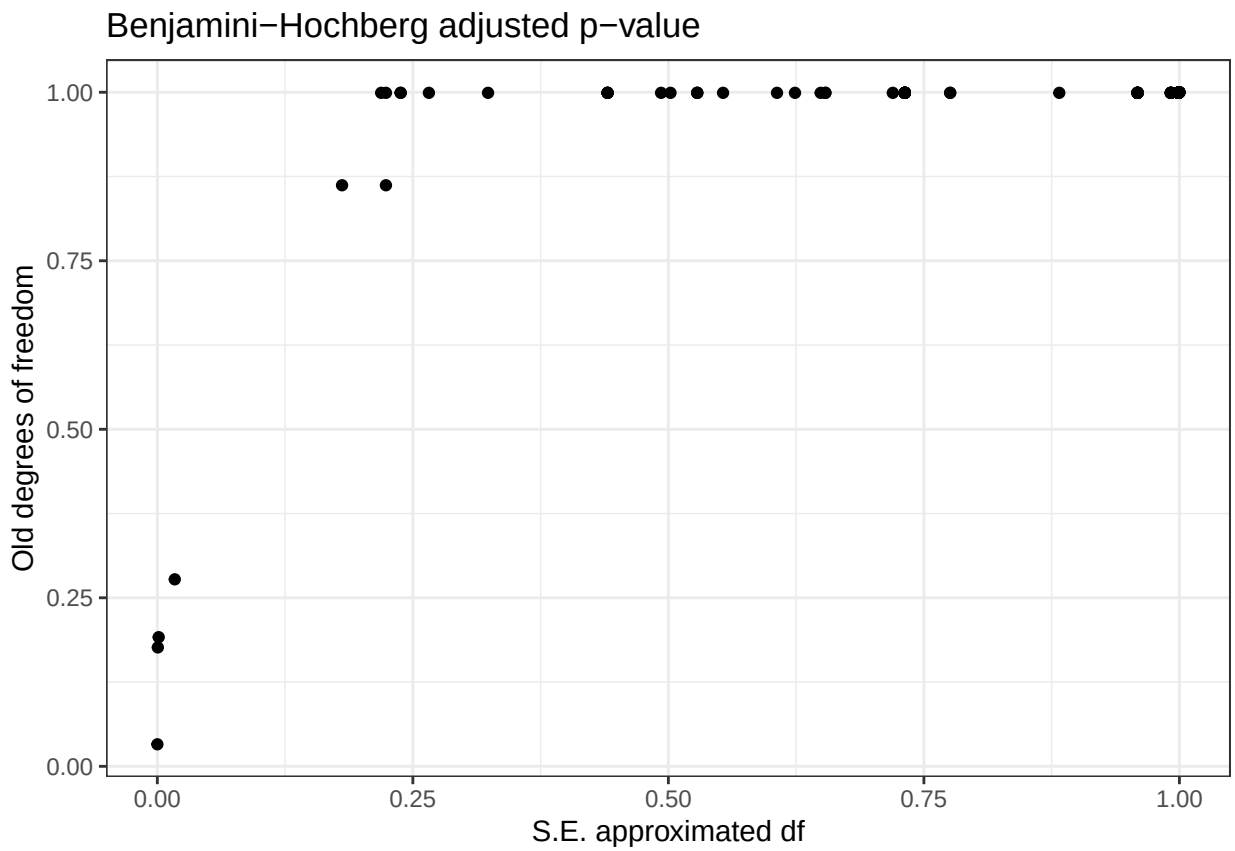

```
OldErrorPropagation %>%
  ggplot(aes(x=pval_MSstats, y=pval_Function)) +
  geom_point() +
  theme_bw() +
  labs(title = "P-value",  x = "S.E. approximated df", y = "Old degrees of freedom")
```

```
## Warning: Removed 2 rows containing missing values (‘geom_point()‘).
```



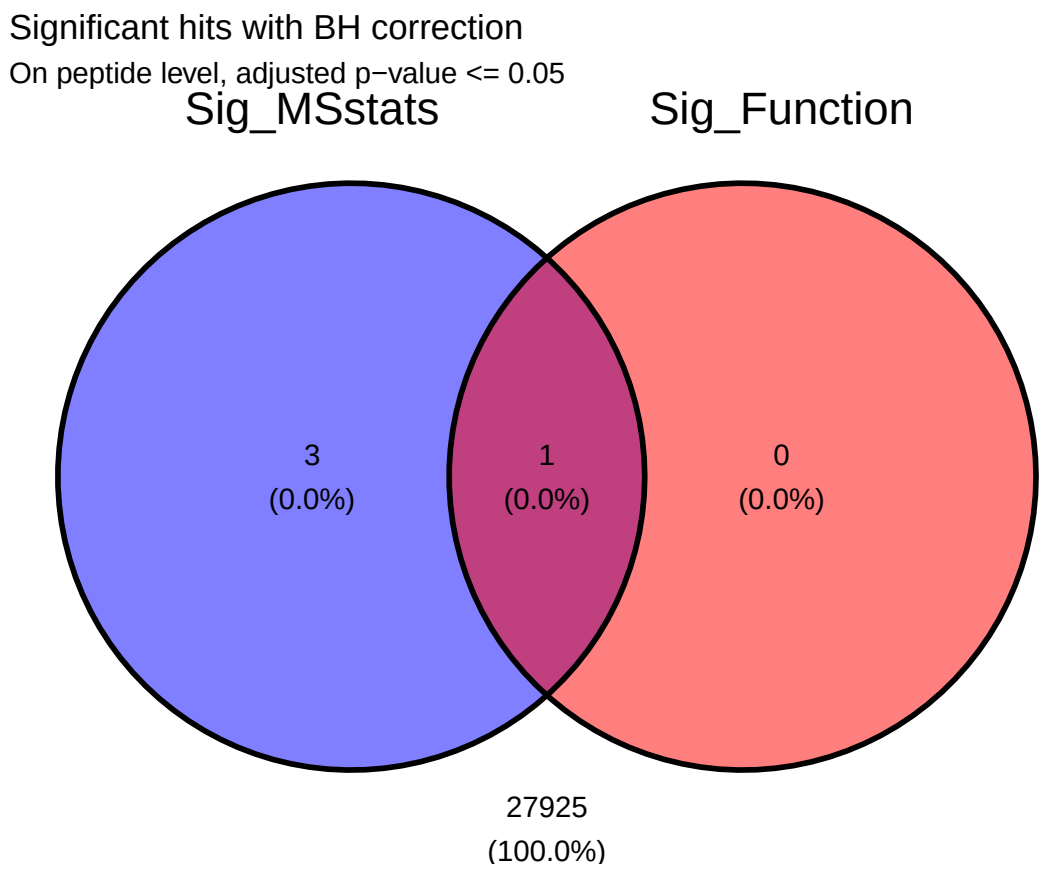

```
OldErrorPropagation %>%
  ggplot(aes(x=p.adj_MSstats, y=p.adj_Function)) +
  geom_point() +
  theme_bw() +
  labs(title = "Benjamini-Hochberg adjusted p-value", x = "S.E. approximated df", y = "Old degrees of fi
```

```
## Warning: Removed 2 rows containing missing values (‘geom_point()‘).
```

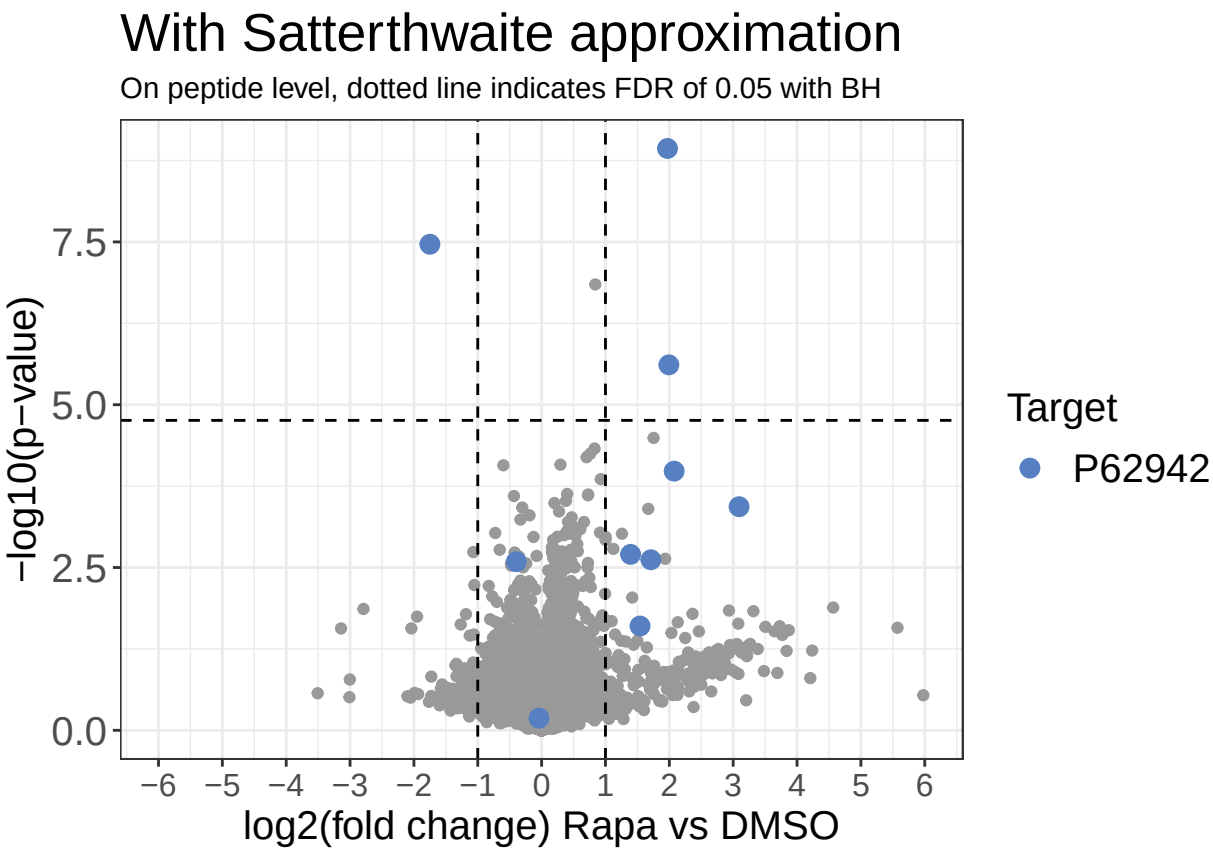


As one can see, when using the Satterthwaite equation for the approximation for the degrees of freedom, the p values are slightly lower because it also includes information from the protein measurements. The results are ranked the same way but there are slightly more significant observations

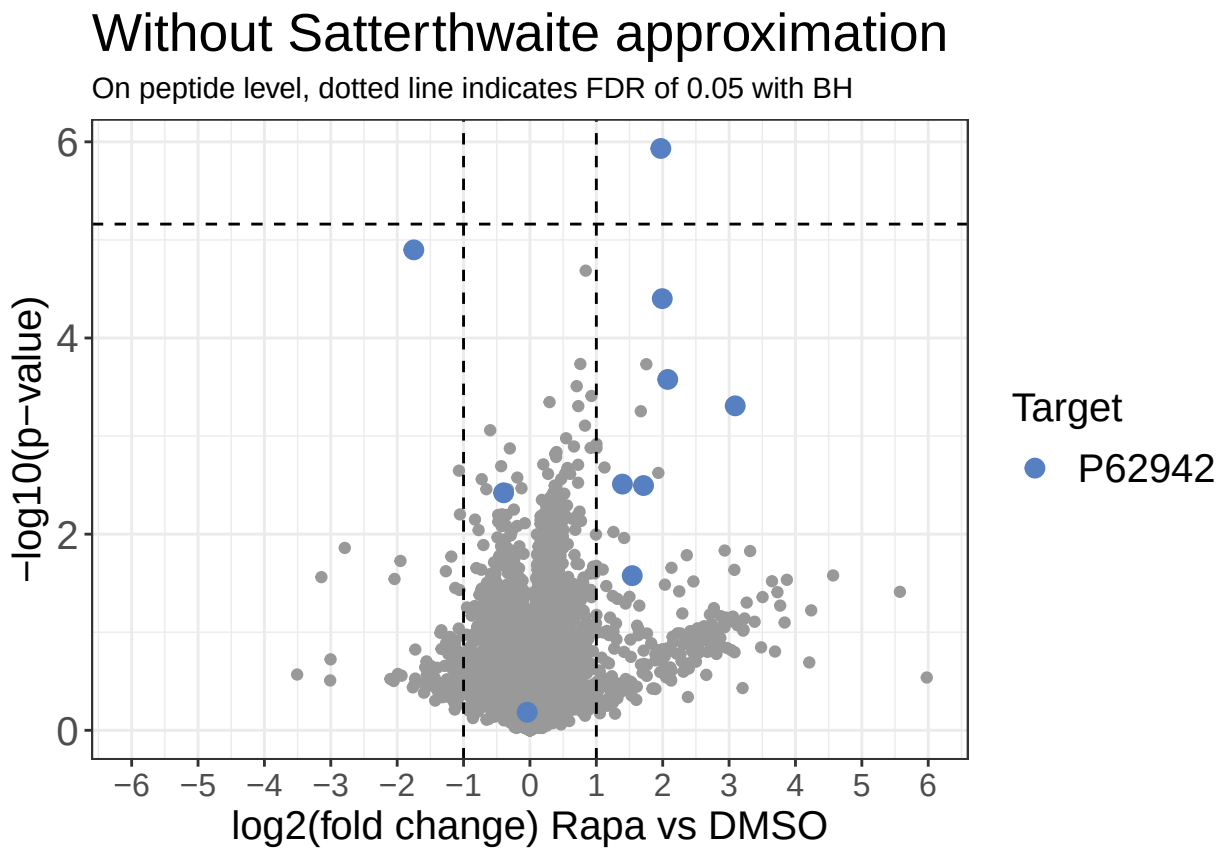
```
# Venn diagramm of significant peptides using adj pvalue <= 0.05

OldErrorPropagation %>%
  mutate(Sig_MSstats = p.adj_MSstats <= 0.05) %>%
  mutate(Sig_Function = p.adj_Function <= 0.05) %>%
  drop_na() %>%
  ggvenn(c("Sig_MSstats","Sig_Function"), fill_color = c("blue","red"))+
  labs(title = "Significant hits with BH correction",
       subtitle = "On peptide level, adjusted p-value <= 0.05")
```




```
target = "P62942"

# Volcano plot using Satterthwaite approximated columns

volcano_plot(
  data = OldErrorPropagation,
  grouping = PeptideSequence,
  log2FC = adj_diff_MSstats,
  significance = pval_MSstats,
  method = "target",
  target_column = ProteinName,
  target = target,
  x_axis_label = "log2(fold change) Rapa vs DMSO",
  significance_cutoff = c(0.05, "p.adj_MSstats"),
  interactive = FALSE
) +
  labs(title = "With Satterthwaite approximation",
       subtitle = "On peptide level, dotted line indicates FDR of 0.05 with BH")
```




```
# Volcano plot using df = n_obs(pep) -2

volcano_plot(
  data = OldErrorPropagation,
  grouping = PeptideSequence,
  log2FC = adj_diff_Function,
  significance = pval_Function,
  method = "target",
  target_column = ProteinName,
  target = target,
  x_axis_label = "log2(fold change) Rapa vs DMSO",
  significance_cutoff = c(0.05, "p.adj_Function"),
  interactive = FALSE
) +
  labs(title = "Without Satterthwaite approximation",
       subtitle = "On peptide level, dotted line indicates FDR of 0.05 with BH")
```




```
OldErrorPropagation %>%
  mutate(Sig_MSstats = pval_MSstats <= 0.05) %>%
  mutate(Sig_Function = pval_Function <= 0.05) %>%
  drop_na() %>%
  ggvenn(c("Sig_MSstats","Sig_Function"), fill_color = c("blue","red"))+
  labs(title = "Significant hits without BH correction",
       subtitle = "On peptide level, p-value <= 0.05")
```

Significant hits without BH correction

On peptide level, p−value <= 0.05