Multi-Matrix Proteomics Analysis

Yuliya Karpievitch

10/30/2017

How to run MultiMatrix pipeline: EigenMS, Model-Based Imputation, Differential Expression

EigenMS normalization

The data used in this examaple is a subset of a proteomics experiment where peptide IDs (sequences) have been shuffled and protein and gene IDs were replced by fake 'Prot_#' name. This document provides an example of the code and data structures that are necessary to run Multi-Matrix analysis, inclusding EigenMS normalization, Model-Based imputation and Multi-Matrix statistical analysis.

For non-proteomics data, such as metabolomics data, 2 columns with identical information can be provided.

Start by loading the data and defining parameter prot.info, 2 column data frame with IDs for metabolites or peptides in case of matabolites the 2 columns are identical. For peptides 1st column must contain unique peptide ID (usually sequences) 2nd column can contain protein IDs, (not used in EigenMS) and any other metadata columns that will be propagated through the analysis pipeline.

Human

```
# Load data, human, mouse
data("hs_peptides") # loads variable hs_peptides
dim(hs_peptides)  # 695 x 13
```

```
## [1] 695  13
```

```
intsCols = 8:13 # column indeces that contain intensities
m_logInts = make_intencities(hs_peptides, intsCols)

# replace 0's with NA's as NA's are more appropriate for anlysis and log2 transform
m_logInts = convert_log2(m_logInts)
metaCols = 1:7 # column indeces that contain metadata such as protein IDs and sequences
m_prot.info = make_meta(hs_peptides, metaCols)

# m_prot.info - 2+ column data frame with pepIDs, here metabolite IDs
head(m_prot.info)
```

```
##                         Sequence MatchedID  ProtID  GeneID     ProtName
## 1            CLLAASPENEAGGLKLDGR         3   Prot3   Gene3   Prot3 Name
## 2                  HNIEGIFTFVDHR         3   Prot3   Gene3   Prot3 Name
## 3 RLFSGTQISTIAESEDSQESVDSVTDSQKR       501 Prot501 Gene501 Prot501 Name
## 4             LREQYGLGPYEAVTPLTK       501 Prot501 Gene501 Prot501 Name
## 5                  LINNNPEIFGPLK       502 Prot502 Gene502 Prot502 Name
## 6                     ENMELEEKEK        14  Prot14  Gene14  Prot14 Name
##     ProtIDLong   GeneIDLong
## 1   Prot3 long   Gene3 long
## 2   Prot3 long   Gene3 long
## 3 Prot501 long Gene501 long
## 4 Prot501 long Gene501 long
## 5 Prot502 long Gene502 long
## 6  Prot14 long  Gene14 long
```

```r
dim(m_logInts) # 695 x 6
```

```
## [1] 695   6
```

```r
grps = as.factor(c('CG','CG','CG', 'mCG','mCG','mCG')) # 3 samples for CG and 3 for mCG

# check the number of missing values
m_nummiss = sum(is.na(m_logInts)) #
m_nummiss
```

```
## [1] 1597
```

```r
m_numtot = dim(m_logInts)[1] * dim(m_logInts)[2] #  total observations
m_percmiss = m_nummiss/m_numtot  # % percent missing observations
m_percmiss # 38.29% missing values, representative of the true data
```

```
## [1] 0.3829736
```

```r
# plot number of missing values for each sample
par(mfcol=c(1,1))
barplot(colSums(is.na(m_logInts)),
        main="Numbers of missing values in Human samples (group order)")
```

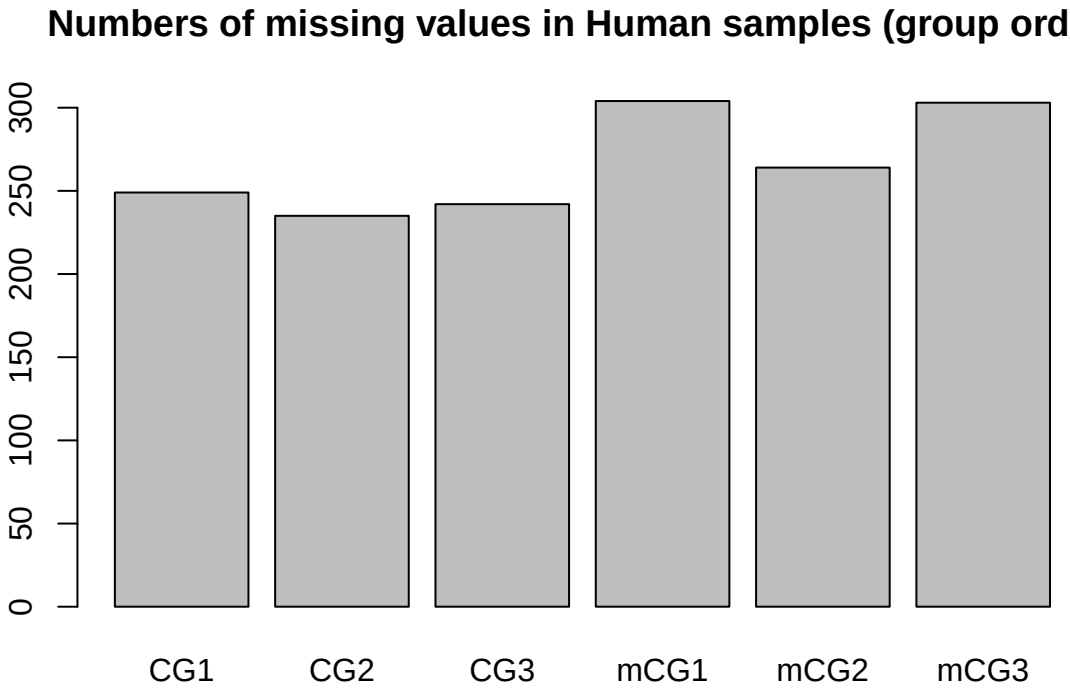


Figure 1. Numbers of missing values in each of the Human samples. mCG treatment group has more missing values.

```r
# Identify bias trends with eig_norm1()
hs_m_ints_eig1 = eig_norm1(m=m_logInts,treatment=grps,prot.info=m_prot.info)
```

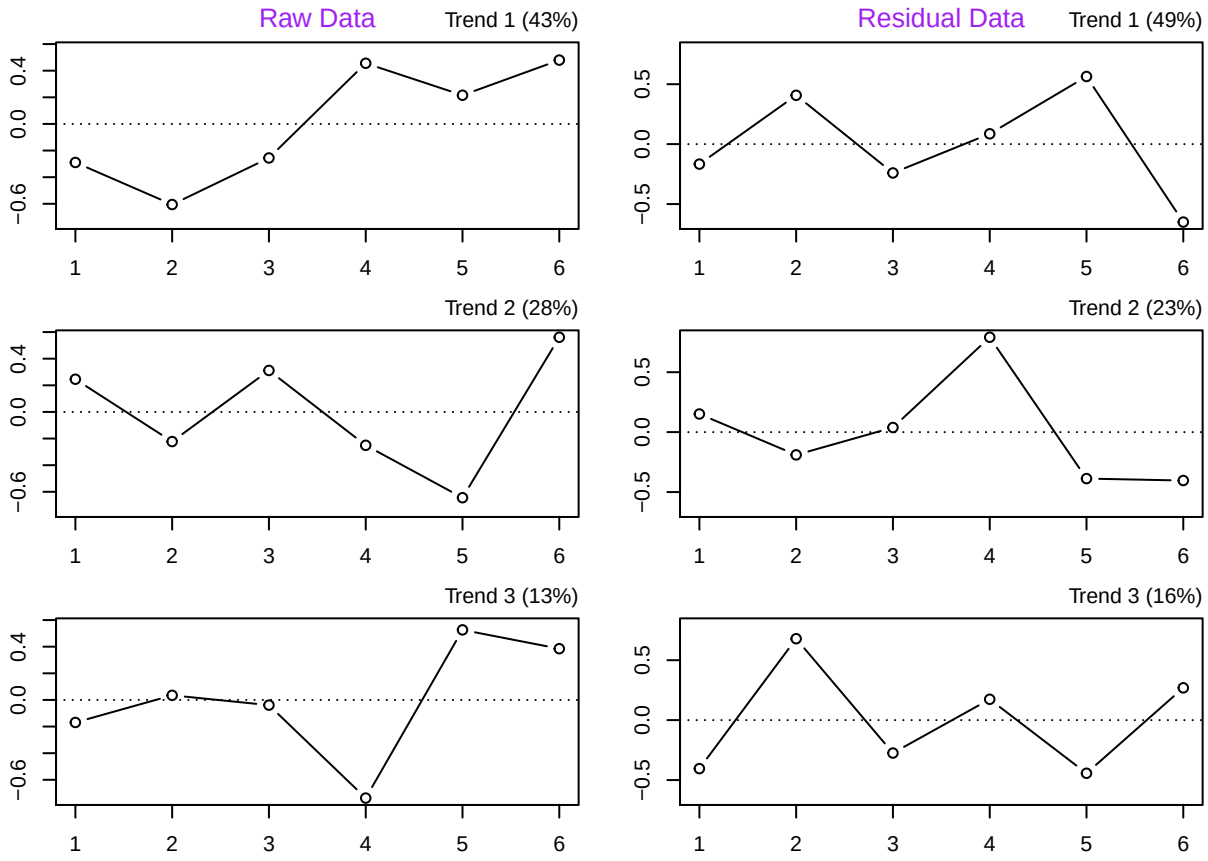


Figure 2. Eigetrends for raw and residual peptide intensities in Human samples. Dots at positions 1-6 correspond to the 6 samples. Top trend in the Raw Data (left panel) shows a pattern representative of the differences between the two groups. Top trend in the Residual Data (right panel) shows that sample 2 and 5 have higher similarity to each otehr, as well as, 1, 3, 4 and 6 whereas in reality samples 1-3 are from the same treatment group and 3-6 are from the other.

```
# check what is inside
names(hs_m_ints_eig1)
```

```
##  [1] "m"             "treatment"     "my.svd"        "pres"
##  [5] "n.treatment"   "n.u.treatment" "h.c"           "present"
##  [9] "prot.info"     "complete"      "toplot1"       "Tk"
## [13] "ncompl"        "grp"
```

```
# Our simulated dataset is small, only 1 bias trend was identified in the
# peptides with no missing values. But visually it seems that there are at least 2.
hs_m_ints_eig1$h.c # 1
```

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

```
# Run EigenMS normalization to eliminate 1 bias trend
hs_m_ints_norm_1bt = eig_norm2(rv=hs_m_ints_eig1)
```

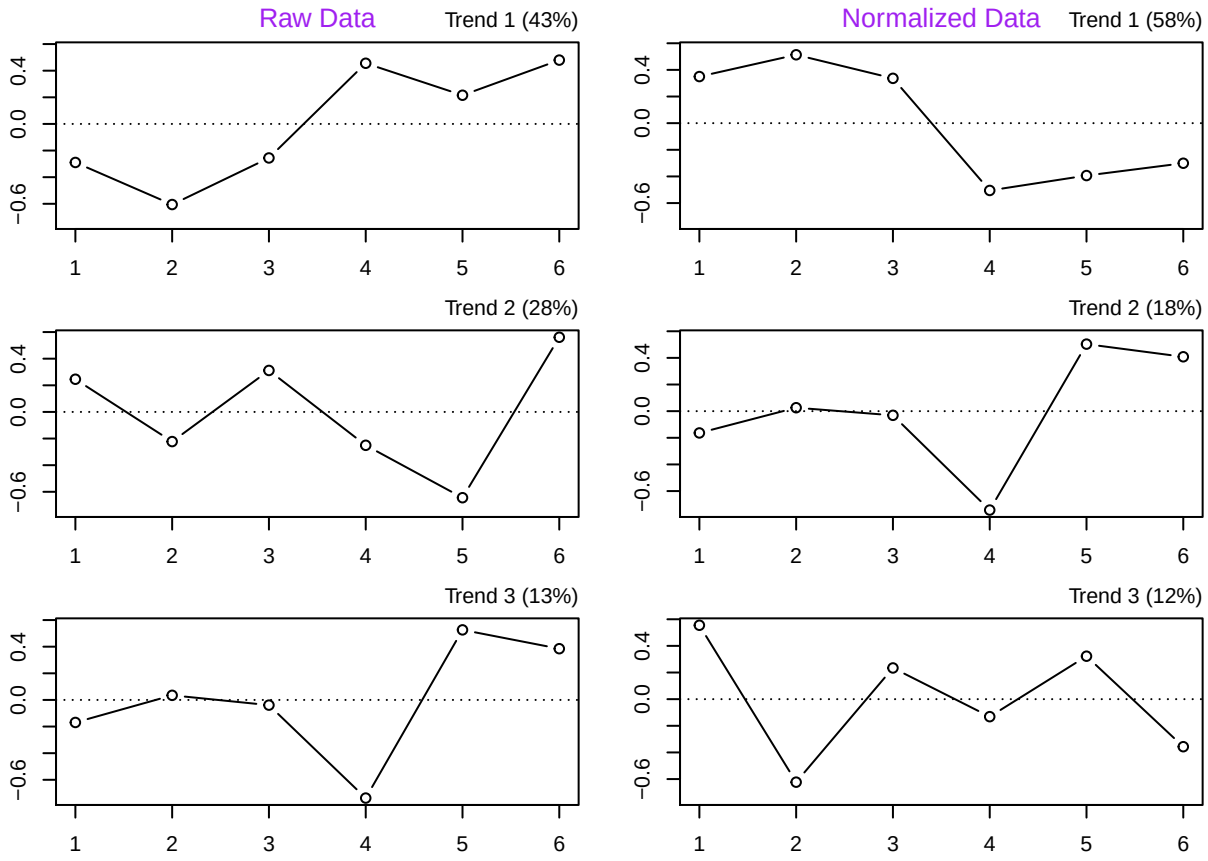


Figure 3. Eigetrends for raw and normalized peptide intensities in Human samples. Dots at positions 1-6 correspond to the 6 samples. Top trend in the Normalizaed Data (right panel) shows a pattern representative of the differences between the two groups (eigen trends can be rotated around x-axis). There is a 15% increase in percent variance explained by the trend as is indicated by the percentage in the upper right corner. But the next (middle) trend explains 18% of variation, so bias effect of this trend may need to be removed.

```
# check what is inside
names(hs_m_ints_eig1)
```

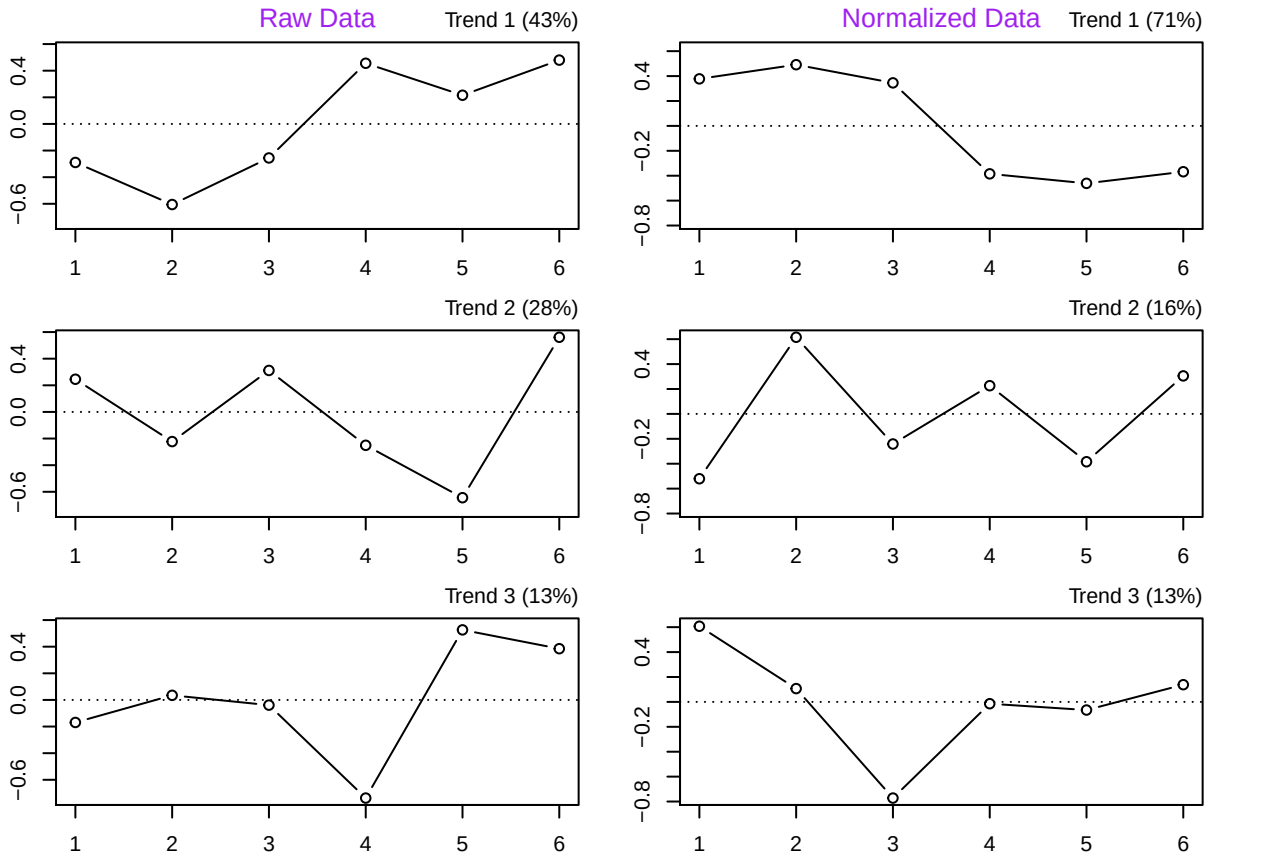
```
##  [1] "m"             "treatment"     "my.svd"        "pres"
##  [5] "n.treatment"   "n.u.treatment" "h.c"           "present"
##  [9] "prot.info"     "complete"      "toplot1"       "Tk"
## [13] "ncompl"        "grp"
```

```
# how many peptides with no missing values (complete) are in the data?
dim(hs_m_ints_eig1$complete)  # bias trend identification is based on 196 peptides
```

```
## [1] 196   6
```

```
# Our simulated dataset is small, with 196 peptides with no missing values.
# Only 1 bias trend was identified, but visually it seems that there are at least 2.
# So set h.c to 2 trestnds to be eliminates
hs_m_ints_eig1$h.c = 2 # visibly there are more than 1 bias trend, set to 2
```

```
# 190 petides with no missing values were ussed for bais trend identification
hs_m_ints_norm = eig_norm2(rv=hs_m_ints_eig1)
```




```
# names(hs_m_ints_norm)
```

Figure 4. Eigetrends for raw and normalized peptide intensities in Human samples with the effects of two bias trends removed. Dots at positions 1-6 correspond to the 6 samples. Top trend in the Normalizaed Data (right panel) shows a pattern representative of the differences between the two groups (eigen trends can be rotated around x-axis).

Figure 4 shows a 28% increase in percent variance explained by the trend where differences between the groups explaining 71% of total variation in the data as is indicated by the percentage in the upper right corner. The next (middle) trend explains 16% of variation, but removing the effect of more trends may overnormalize, thus this we will use normalized data with two bias trends eliminated.

Mouse

```
data("mm_peptides") # loads variable mm_peptides
dim(mm_peptides)
```

```
## [1] 1102   13
```

```
dim(mm_peptides) # 1102 x 13
```

```
## [1] 1102   13
```

```
head(mm_peptides)
```

```
##                              Sequence MatchedID ProtID GeneID    ProtName
## 1             GFAYVQFEDVRDAEDALYNLNRK        64 Prot64 Gene64 Prot64 Name
## 2                   SKCEELSSLHGQLKEAR        61 Prot61 Gene61 Prot61 Name
## 3    QDAGSEPVTPASLAALQSDVQPVGHDYVEEVR        61 Prot61 Gene61 Prot61 Name
## 4         TGDQEERQDYINLDESEAAAFDDEWRR         1  Prot1  Gene1  Prot1 Name
## 5              IPAYFITVHDPAVPPGEDPDGR        60 Prot60 Gene60 Prot60 Name
## 6 GGTPGSGAAAAAGSKPPPSSSASASSSSSSFAQQR        60 Prot60 Gene60 Prot60 Name
##    ProtIDLong  GeneIDLong      CG1      CG2      CG3     mCG1     mCG2
## 1 Prot64 long Gene64 long  3725900 11642000  4872400        0 12850000
## 2 Prot61 long Gene61 long 19699000 38055000 30661000 15896000 55187000
## 3 Prot61 long Gene61 long        0        0        0  5277500        0
## 4  Prot1 long  Gene1 long        0        0        0        0        0
## 5 Prot60 long Gene60 long  9391200        0        0  4689800  8305300
## 6 Prot60 long Gene60 long        0        0 20406000  5809800        0
##       mCG3
## 1  3751700
## 2 20356000
## 3 38698000
## 4        0
## 5        0
## 6        0
```

```
intsCols = 8:13 # may differ for each dataset
m_logInts = make_intencities(mm_peptides, intsCols)  # will reuse the name m_logInts
m_logInts = convert_log2(m_logInts)
metaCols = 1:7
m_prot.info = make_meta(mm_peptides, metaCols)

head(m_prot.info)
```

```
##                              Sequence MatchedID ProtID GeneID    ProtName
## 1             GFAYVQFEDVRDAEDALYNLNRK        64 Prot64 Gene64 Prot64 Name
## 2                   SKCEELSSLHGQLKEAR        61 Prot61 Gene61 Prot61 Name
## 3    QDAGSEPVTPASLAALQSDVQPVGHDYVEEVR        61 Prot61 Gene61 Prot61 Name
## 4         TGDQEERQDYINLDESEAAAFDDEWRR         1  Prot1  Gene1  Prot1 Name
## 5              IPAYFITVHDPAVPPGEDPDGR        60 Prot60 Gene60 Prot60 Name
## 6 GGTPGSGAAAAAGSKPPPSSSASASSSSSSFAQQR        60 Prot60 Gene60 Prot60 Name
##    ProtIDLong  GeneIDLong
## 1 Prot64 long Gene64 long
## 2 Prot61 long Gene61 long
## 3 Prot61 long Gene61 long
## 4  Prot1 long  Gene1 long
## 5 Prot60 long Gene60 long
## 6 Prot60 long Gene60 long
```

```
dim(m_logInts)# 1102 x 6
```

```
## [1] 1102    6
```

```
# check numbers of missing values in Mouse samples
m_nummiss = sum(is.na(m_logInts)) #
m_nummiss
```

```
## [1] 2698
```

```
m_numtot = dim(m_logInts)[1] * dim(m_logInts)[2] #  total observations
m_percmiss = m_nummiss/m_numtot  # % percent missing observations
m_percmiss # 40.8% missing values, representative of the true data
```

```
## [1] 0.408046
```

```
# plot number of missing values for each sample
par(mfcol=c(1,1))
barplot(colSums(is.na(m_logInts)),
        main="Numbers of missing values in Mouse samples (group order)")
```

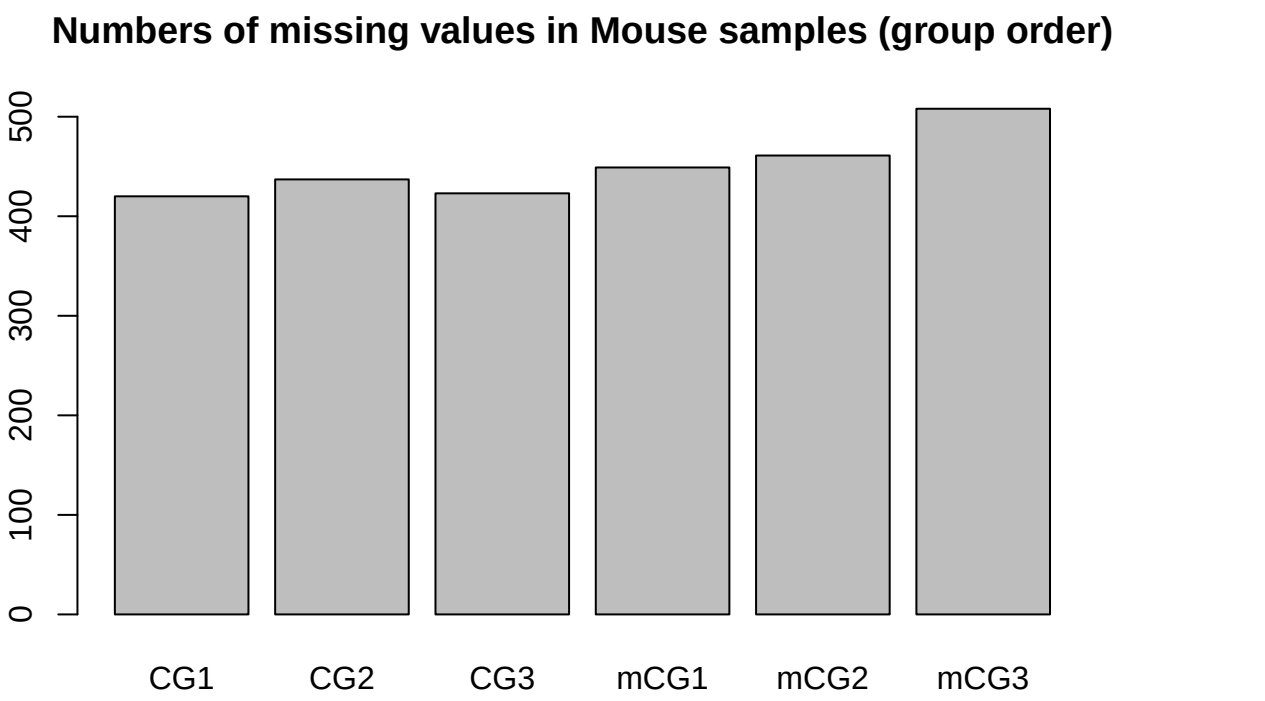


Figure 5. Numbers of missing values in each of the Human samples. mCG treatment group has more missing values.

```
mm_m_ints_eig1 = eig_norm1(m=m_logInts,treatment=grps,prot.info=m_prot.info)
```

```
## The following object is masked from TREAT (pos = 3):
##
##     TREAT
```

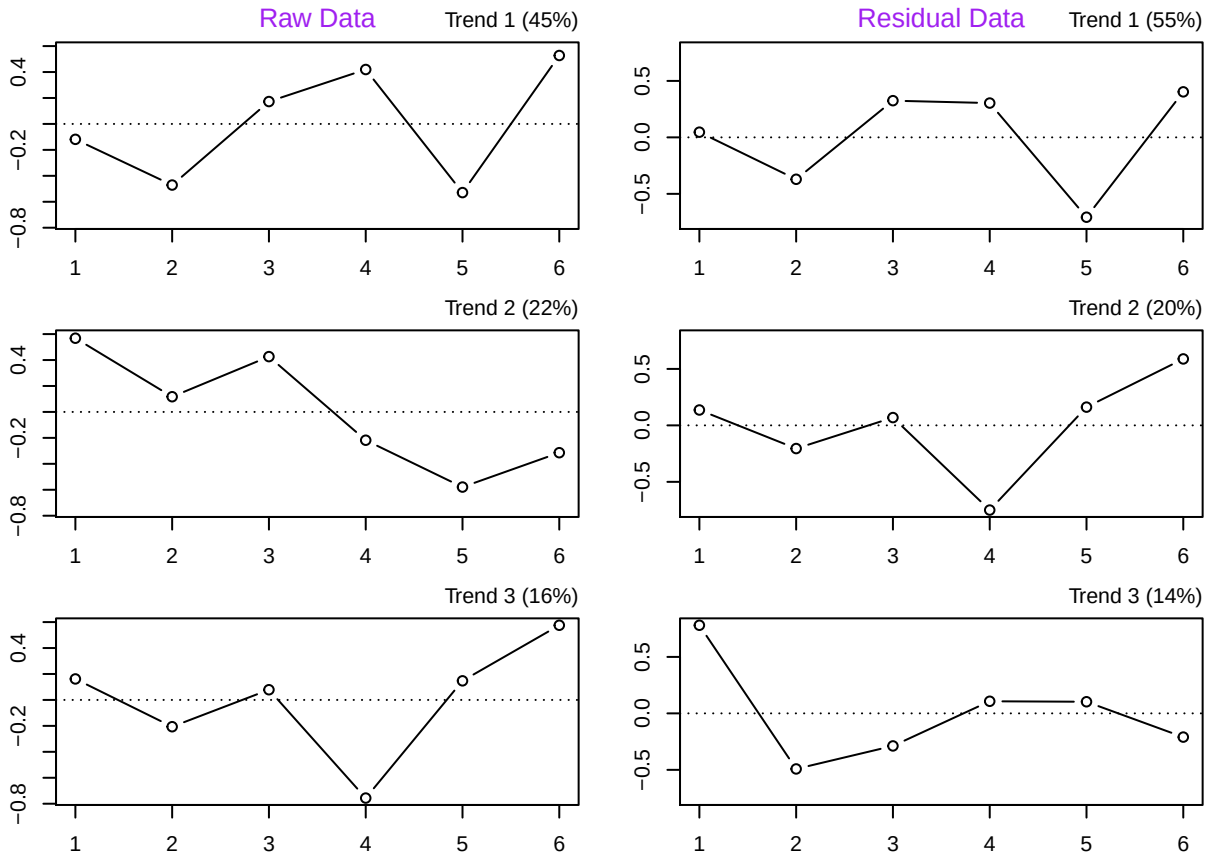


Figure 5. Eigetrends for raw and residual peptide intensities in Mouse samples. Dots at positions 1-6 correspond to the 6 samples. Top trend in the Normalizaed Data (right panel) shows a pattern representative of the differences between the two groups (eigen trends can be rotated around x-axis).

The eigntrend that explains most of the variation (45%) in the Mouse data is nto representative of the treatment groupt differences (Figure 5). The second trend in the raw data explains only 22% of the total variation that resambles treatmetn group differences necesitating normalization. variation in the data as is indicated by the percentage in the upper right corner.

```
mm_m_ints_eig1$h.c
```

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

```
mm_m_ints_norm_1bt = eig_norm2(rv=mm_m_ints_eig1)  # 700 x 560 resolution
```

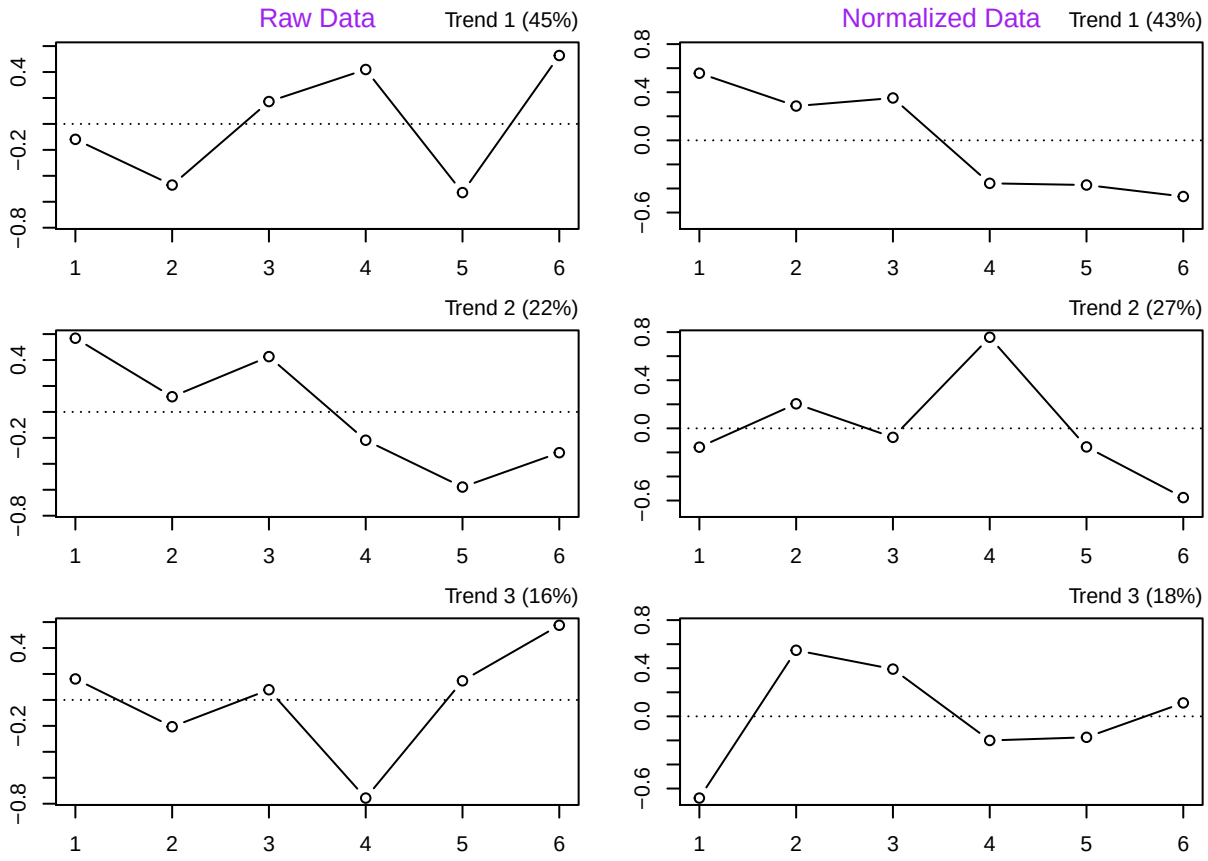
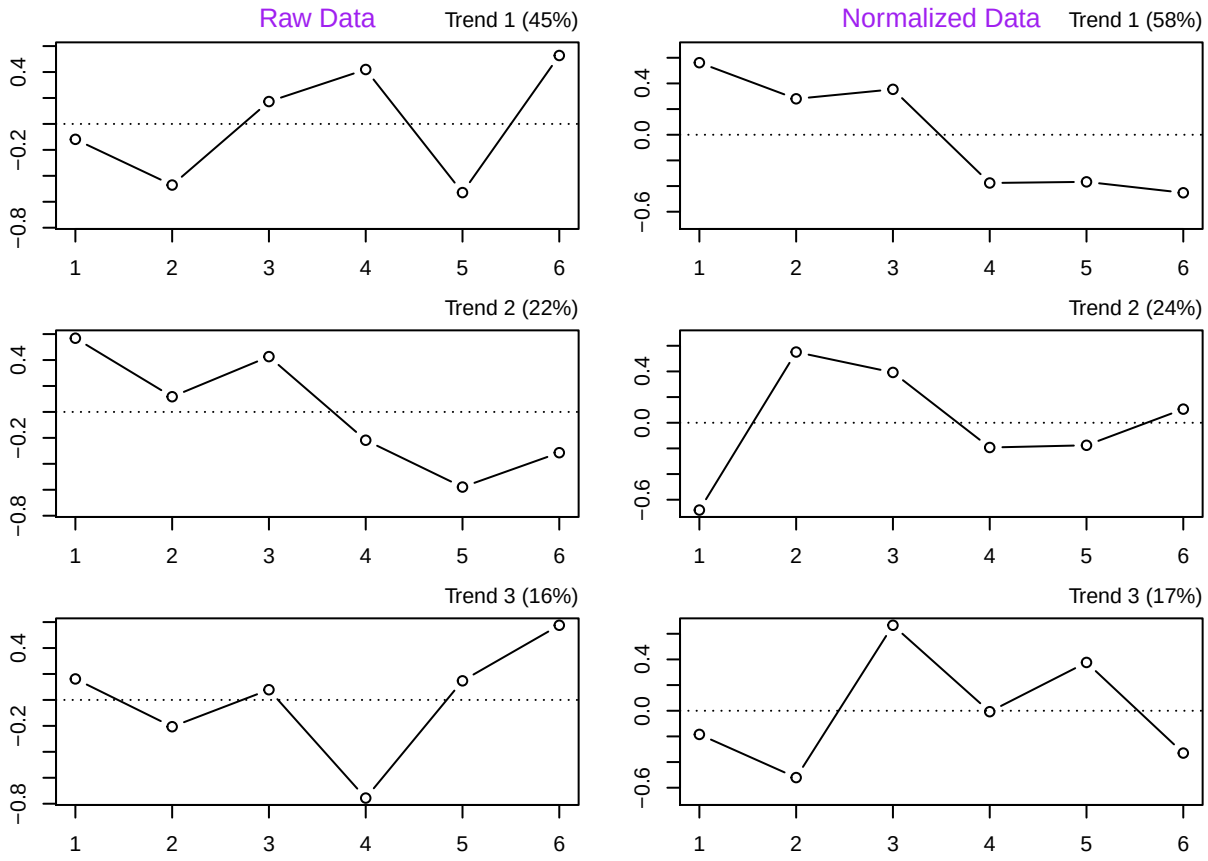


Figure 6. Eigetrends for raw and normalized peptide intensities in Mouse samples with the effects of one bias trends removed. Dots at positions 1-6 correspond to the 6 samples. Top trend in the Normalizaed Data Dots at positions 1-6 correspond to the 6 samples. Top trend in the Normalizaed Data (right panel) shows a pattern representative of the differences between the two groups.

The eigntrend that explains most of the variation (43%) in the normalized Mouse data is representative of the treatment groupt differences. The second trend in the raw data explains only 27% of the total variation and should be concidered as bias.

```
mm_m_ints_eig1$h.c = 2
```

```
mm_m_ints_norm = eig_norm2(rv=mm_m_ints_eig1)
```




```
# 190 petides with no missing values were used for bais trend identification ($complete)
```

Figure 7. Eigetrends for raw and normalized peptide intensities in Mouse samples with the effects of two bias trends removed. Dots at positions 1-6 correspond to the 6 samples. Top trend in the Normalizaed Data (right panel) shows a pattern representative of the differences between the two groups.

The eigntrend that explains most of the variation in the normalized Mouse data representative of the treatment groupt differences now explains 58% of variation. The second trend in the normalized data explains less of variaion that in Figure 6 (24%) whch is still a bit high, but we will use these data for analysis to avoid overfitting.

```
length(mm_m_ints_eig1$prot.info$MatchedID)          # 1102 - correct
```

```
## [1] 1102
```

```
length(hs_m_ints_eig1$prot.info$MatchedID)          # 695 - all are able to normalize
```

```
## [1] 695
```

```
length(unique(mm_m_ints_eig1$prot.info$MatchedID) ) # 69
```

```
## [1] 69
```

```
length(unique(hs_m_ints_eig1$prot.info$MatchedID) ) # 69
```

```
## [1] 69
```

```
# 787 peptides were normalized, rest eliminated due to low # of observations
dim(mm_m_ints_norm$norm_m)
```

```
## [1] 787   6
```

```r
dim(hs_m_ints_norm$norm_m) # 480 peptides were normalized
```

```
## [1] 480   6
```

Model-based imputation

Human

```
# Set up mata data and intensities to use for the imputation
hs_prot.info = hs_m_ints_norm$normalized[,metaCols]
hs_norm_m =  hs_m_ints_norm$normalized[,intsCols]
head(hs_prot.info)
```

```
##                                                 Sequence MatchedID
## CLLAASPENEAGGLKLDGR                     CLLAASPENEAGGLKLDGR         3
## HNIEGIFTFVDHR                             HNIEGIFTFVDHR         3
## RLFSGTQISTIAESEDSQESVDSVTDSQKR RLFSGTQISTIAESEDSQESVDSVTDSQKR       501
## LINNNPEIFGPLK                             LINNNPEIFGPLK       502
## ENMELEEKEK                                   ENMELEEKEK        14
## GHEFYNPQKK                                   GHEFYNPQKK        14
##                                 ProtID  GeneID     ProtName   ProtIDLong
## CLLAASPENEAGGLKLDGR              Prot3   Gene3   Prot3 Name   Prot3 long
## HNIEGIFTFVDHR                    Prot3   Gene3   Prot3 Name   Prot3 long
## RLFSGTQISTIAESEDSQESVDSVTDSQKR Prot501 Gene501 Prot501 Name Prot501 long
## LINNNPEIFGPLK                  Prot502 Gene502 Prot502 Name Prot502 long
## ENMELEEKEK                      Prot14  Gene14  Prot14 Name  Prot14 long
## GHEFYNPQKK                      Prot14  Gene14  Prot14 Name  Prot14 long
##                                  GeneIDLong
## CLLAASPENEAGGLKLDGR               Gene3 long
## HNIEGIFTFVDHR                     Gene3 long
## RLFSGTQISTIAESEDSQESVDSVTDSQKR Gene501 long
## LINNNPEIFGPLK                  Gene502 long
## ENMELEEKEK                      Gene14 long
## GHEFYNPQKK                      Gene14 long
```

```
head(hs_norm_m)
```

```
##                                      CG1      CG2      CG3     mCG1
## CLLAASPENEAGGLKLDGR            24.16344 25.11800 25.39066 24.73530
## HNIEGIFTFVDHR                  21.81538       NA 21.42956 21.90027
## RLFSGTQISTIAESEDSQESVDSVTDSQKR 23.52846 22.73723 23.53173 23.03903
## LINNNPEIFGPLK                        NA 22.34531 21.88714       NA
## ENMELEEKEK                     27.31511 26.85826 27.39201 27.89371
## GHEFYNPQKK                     24.69609 24.27661 24.96221 24.42590
##                                     mCG2     mCG3
## CLLAASPENEAGGLKLDGR            24.47494 24.65338
## HNIEGIFTFVDHR                  21.74596       NA
## RLFSGTQISTIAESEDSQESVDSVTDSQKR 23.51463 22.95478
## LINNNPEIFGPLK                  21.09684 21.24429
## ENMELEEKEK                     28.18741 27.83388
## GHEFYNPQKK                     24.74535 24.34182
```

```
dim(hs_norm_m) # 480 x 6, raw: 695, 215 peptides were elimunaed due to lack of observations
```

```
## [1] 480   6
```

```
length(unique(hs_prot.info$MatchedID)) # 59
```

```
## [1] 59
```

```r
length(unique(hs_prot.info$ProtID))    # 59
```

```
## [1] 59
```

```r
set.seed(1213)
# impute based on ProtID - position in the matrix for the Protein Identifyer
imp_hs = MBimpute(hs_norm_m, grps, prot.info=hs_prot.info, pr_ppos=3, my.pi=0.05,
                  compute_pi=FALSE, sseed=171717) # pi already computed...
```

```r
# check some nuumbers
length(unique(imp_hs$imp_prot.info$MatchedID)) # 59 - MatchedID IDs
```

```
## [1] 59
```

```r
length(unique(imp_hs$imp_prot.info$ProtID))    # 59 - Protein IDs
```

```
## [1] 59
```

```r
length(unique(imp_hs$imp_prot.info$GeneID))    # 59
```

```
## [1] 59
```

```r
dim(imp_hs$imp_prot.info) # 480 x 7 imputed peptides
```

```
## [1] 480   7
```

```r
dim(imp_hs$y_imputed)     # 480 x 6
```

```
## [1] 480   6
```

```r
# plot one of the protiens to check normalization and imputation visually
mylabs = c( 'CG','CG','CG', 'mCG','mCG','mCG') # same as grps but this one is a string
prot_to_plot = 'Prot32' # 43
gene_to_plot = 'Gene32'
plot_3_pep_trends_NOfile(as.matrix(hs_m_ints_eig1$m), hs_m_ints_eig1$prot.info,
                         as.matrix(hs_norm_m), hs_prot.info, imp_hs$y_imputed,
                         imp_hs$imp_prot.info, prot_to_plot, 3, gene_to_plot, 4, mylabs)
```

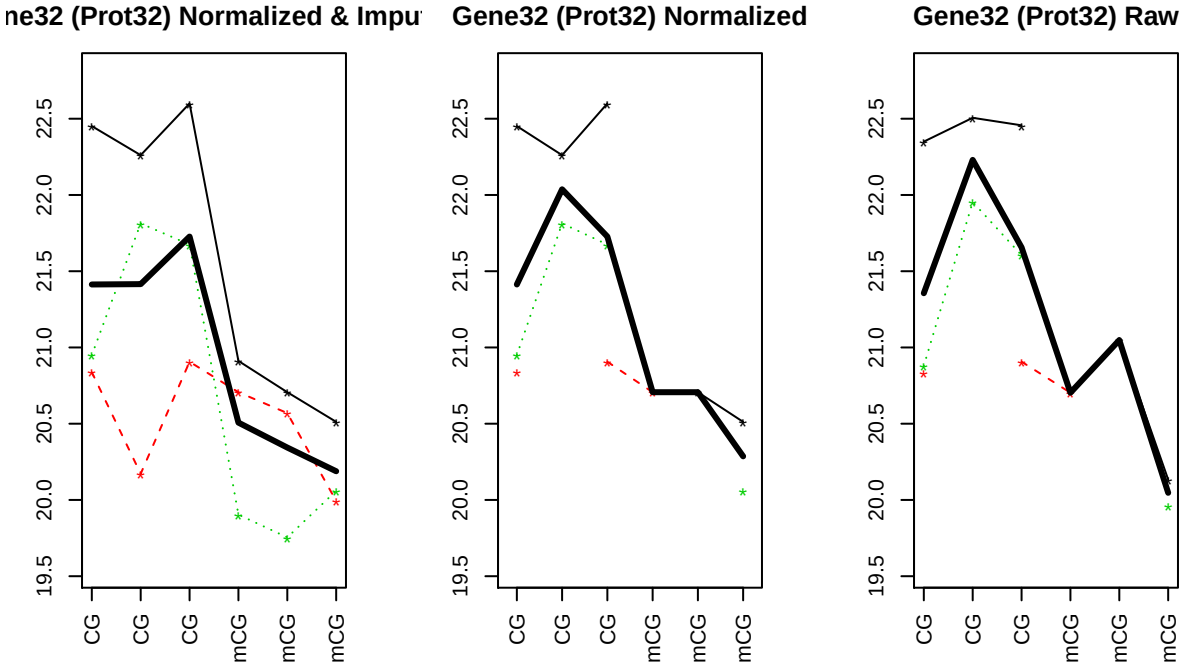


Figure 8. All peptides within protein Prot32 in raw, noramlized, and imputed form.

Mouse

```
mm_prot.info = mm_m_ints_norm$normalized[,1:7]
mm_norm_m =  mm_m_ints_norm$normalized[,8:13]
head(mm_prot.info)
```

```
##                                                                  Sequence
## GFAYVQFEDVRDAEDALYNLNRK                                GFAYVQFEDVRDAEDALYNLNRK
## SKCEELSSLHGQLKEAR                                            SKCEELSSLHGQLKEAR
## IPAYFITVHDPAVPPGEDPDGR                                IPAYFITVHDPAVPPGEDPDGR
## GGTPGSGAAAAAGSKPPPSSSASASSSSSSFAQQR GGTPGSGAAAAAGSKPPPSSSASASSSSSSFAQQR
## NLGGNYPEK                                                      NLGGNYPEK
## ISCAGPQTYKEHLEGQKHK                                    ISCAGPQTYKEHLEGQKHK
##                                    MatchedID ProtID GeneID    ProtName
## GFAYVQFEDVRDAEDALYNLNRK                   64 Prot64 Gene64 Prot64 Name
## SKCEELSSLHGQLKEAR                         61 Prot61 Gene61 Prot61 Name
## IPAYFITVHDPAVPPGEDPDGR                    60 Prot60 Gene60 Prot60 Name
## GGTPGSGAAAAAGSKPPPSSSASASSSSSSFAQQR       60 Prot60 Gene60 Prot60 Name
## NLGGNYPEK                                 28 Prot28 Gene28 Prot28 Name
## ISCAGPQTYKEHLEGQKHK                       53 Prot53 Gene53 Prot53 Name
##                                     ProtIDLong  GeneIDLong
## GFAYVQFEDVRDAEDALYNLNRK             Prot64 long Gene64 long
## SKCEELSSLHGQLKEAR                   Prot61 long Gene61 long
## IPAYFITVHDPAVPPGEDPDGR              Prot60 long Gene60 long
## GGTPGSGAAAAAGSKPPPSSSASASSSSSSFAQQR Prot60 long Gene60 long
## NLGGNYPEK                           Prot28 long Gene28 long
## ISCAGPQTYKEHLEGQKHK                 Prot53 long Gene53 long
```

```
head(mm_norm_m)
```

```
##                                         CG1      CG2      CG3      mCG1
## GFAYVQFEDVRDAEDALYNLNRK             21.99076 22.78591 22.74153       NA
## SKCEELSSLHGQLKEAR                   24.24259 24.78175 25.25876 24.56999
## IPAYFITVHDPAVPPGEDPDGR              23.13090       NA       NA 22.56945
## GGTPGSGAAAAAGSKPPPSSSASASSSSSSFAQQR       NA       NA 24.28249 22.47006
## NLGGNYPEK                           24.19505 24.89556 24.52888       NA
## ISCAGPQTYKEHLEGQKHK                       NA 22.50866 23.56617 23.18408
##                                        mCG2     mCG3
## GFAYVQFEDVRDAEDALYNLNRK             22.68752 22.83741
## SKCEELSSLHGQLKEAR                   24.76317 24.58578
## IPAYFITVHDPAVPPGEDPDGR              22.63033       NA
## GGTPGSGAAAAAGSKPPPSSSASASSSSSSFAQQR       NA       NA
## NLGGNYPEK                                 NA 24.74703
## ISCAGPQTYKEHLEGQKHK                       NA 22.84175
```

```
dim(mm_norm_m) # 787 x 6, raw had: 1102
```

```
## [1] 787   6
```

```
length(unique(mm_prot.info$MatchedID)) # 56 (69?)
```

```
## [1] 56
```

```
length(unique(mm_prot.info$ProtID))    # 56
```

```
## [1] 56
```

```r
set.seed(12131)
# impute based on ProtID - position in the matrix for the Protein Identifier
imp_mm = MBimpute(mm_norm_m, grps, prot.info=mm_prot.info, pr_ppos=3, my.pi=0.05,
                  compute_pi=FALSE, sseed=17171) # pi already computed...
```

```r
# check if returned number of rows corresponds to same number of rows in normalized data
dim(imp_mm$imp_prot.info) # 787 x 7 - imputed peptides & 787 were normalized
```

```
## [1] 787   7
```

```r
dim(imp_mm$y_imputed)     # 787 x 6
```

```
## [1] 787   6
```

Model-Based Differential Expression Analysis

Combined Model-Based Differential Expression Analysis

```
# make lists to pass as parameters in real data Use Mouse_gene_stable_ID as Protein ID
# OR Human
# Multi Matrix anlysis is generalizable to 2+ datasets thus parallel list are used to
# store intensities, metadata, and treatment group information
mms = list()
treats = list()
protinfos = list()
mms[[1]] = imp_mm$y_imputed
mms[[2]] = imp_hs$y_imputed
treats[[1]] = grps
treats[[2]] = grps
# 2nd column is PROTEIN IDENTIFYER - matchedID, have to be present in both datasets,
# in this simulated dataset ProtIDs will match across Human and Mouse, in reality
# protein IDs will differ, sometimes only by upper vs lower case, other times names
# will be different entirely, thus ProtID is not a good identifier to use across
# different organizms.
protinfos[[1]] = imp_mm$imp_prot.info
protinfos[[2]] = imp_hs$imp_prot.info

# divide data into a list of proteins that are common to both datasets (can be more than 2)
# and proteins present only in one or the other (unique to one or the other)
# here we will analyse the proteins that were observed only in one of the datasets
# grps variable does not change
subset_data = subset_proteins(mm_list=mms, prot.info=protinfos, 'MatchedID')
names(subset_data)
```

```
## [1] "sub_mm_list"          "sub_prot.info"        "sub_unique_mm_list"
## [4] "sub_unique_prot.info" "common_list"
```

```
mm_dd_only = subset_data$sub_unique_prot.info[[1]]
hs_dd_only = subset_data$sub_unique_prot.info[[2]]

ugene_mm_dd = unique(mm_dd_only$MatchedID)
ugene_hs_dd = unique(hs_dd_only$MatchedID)
length(ugene_mm_dd) # 24 - in Mouse only
```

```
## [1] 24
```

```
length(ugene_hs_dd) # 27 - Human only
```

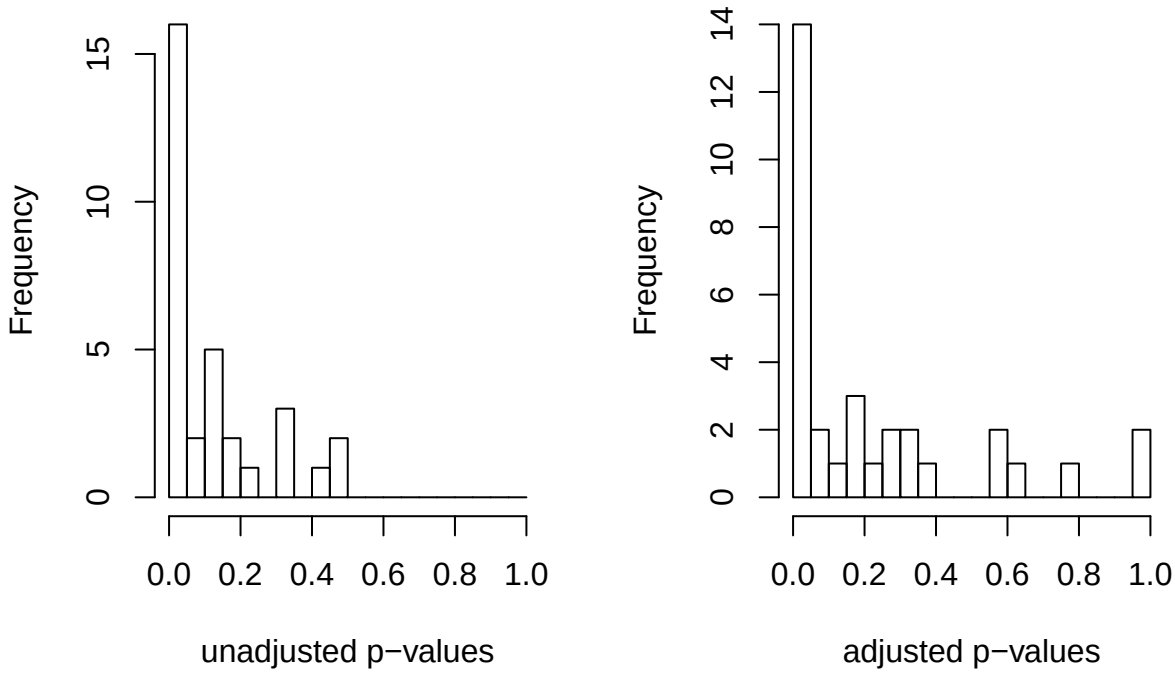
```
## [1] 27
```

```
nsets = length(mms)
nperm = 50   # number of permutations should be 500+ for publication quality permutation
```

```
ptm = proc.time()
comb_MBDE = prot_level_multi_part(mm_list=mms, treat=treats, prot.info=protinfos,
                                  prot_col_name='ProtID', nperm=nperm,
                                  setseed=123, dataset_suffix=c('MM', 'HS'))
proc.time() - ptm  # shows how long it takes to run the test
```

```
mybreaks = seq(0,1, by=.05)
# adjustment for permutation test is done by stretching out values on the interval [0 1]
```

```r
# as expected in a theoretical p-value distribution
par(mfcol=c(1,2)) # always check out p-values
# bunched up on interval [0 .5]
hist(comb_MBDE$P_val, breaks=mybreaks, xlab='unadjusted p-values', main='')
# adjusted p-values look good
hist(comb_MBDE$BH_P_val, breaks=mybreaks, xlab='adjusted p-values', main='')
```




```r
# bunched up on interval [0 .5]
hist(p.adjust(comb_MBDE$P_val, method='BH'), breaks=mybreaks, xlab='BH adjusted p-values', main='')
```

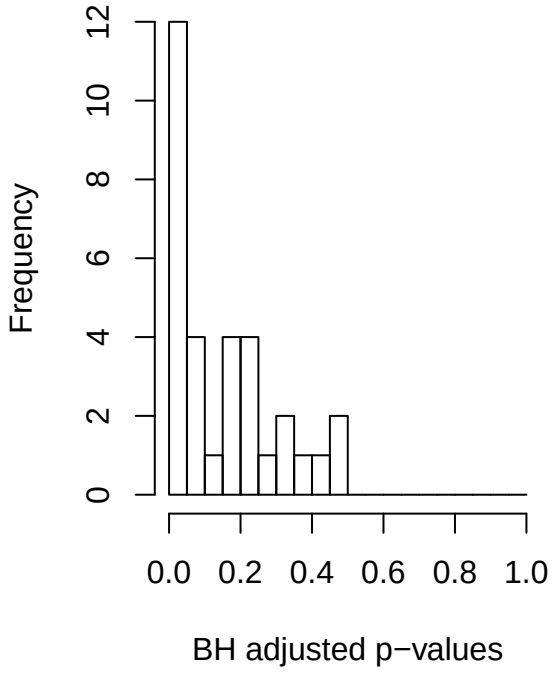


Figure 9. P-value distributions for unadjusted and adjusted p-values. Adjusted p-values (top

right) look as expected according to the theory with a peak near 0 and an approximately uniform distribution throughout the interval [0 1]. Benjamini-Hochberg adjusted p-values (bottom left) do not look according to the theoretical distribution, thus Benjamini-Hochberg adjusted is not appropriate.

```
# horizontal streaks correspond to where a permutation test produces 0 or very small value
# these are reset to improve visualization
par(mfcol=c(1,1)) # Volcano will looc better for larger dataset...
plot_volcano_wLab(comb_MBDE$FC, comb_MBDE$BH_P_val, comb_MBDE$GeneID, FC_cutoff=1.2,
                  PV_cutoff=.05, 'CG vs mCG')
```

```
## Loading required package: ggplot2
```

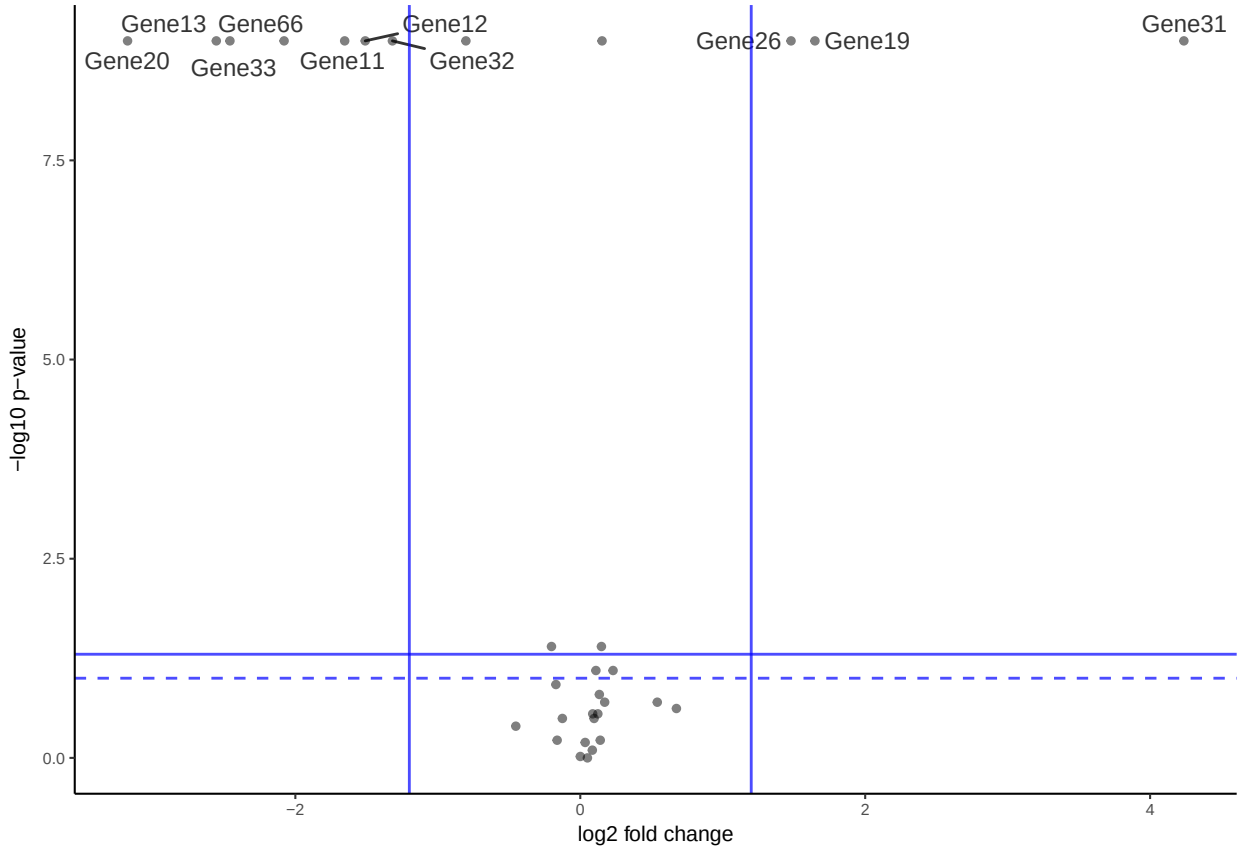


Figure 10. Distribution of p-values and fold changes for combined multi-matrix analysis of Mouse and Human.

Human Only Model-Based Differential Expression Analysis No Human (HS) specific protiens that can be analysed with Model-Based Differential Expression Analysis, so no analysis for that subset.

Mouse Only Model-Based Differential Expression Analysis

```
# subset_data contains "sub_unique_mm_list"  "sub_unique_prot.info" lists for each dataset
# in the order provided to subset function
mms_mm_dd = subset_data$sub_unique_mm_list[[1]] # Mouse
dim(mms_mm_dd)  # 258 x 6,
```

```
## [1] 258   6
```

```
protinfos_mm_dd = subset_data$sub_unique_prot.info[[1]]

length(unique(protinfos_mm_dd$ProtID))    # 24
```

```
## [1] 24
```

```
length(unique(protinfos_mm_dd$GeneID))    # 24
```

```
## [1] 24
```

```
length(unique(protinfos_mm_dd$MatchedID)) # 24
```

```
## [1] 24
```

```
DE_mCG_CG_mm_dd = peptideLevel_DE(mms_mm_dd, grps, prot.info=protinfos_mm_dd, pr_ppos=2)
```

```
## Warning in summary.lm(res): essentially perfect fit: summary may be
## unreliable
```

```
# volcano plot
FCval = 1.2 # change this value for alternative fold change cutoff
plot_volcano_wLab(DE_mCG_CG_mm_dd$FC, DE_mCG_CG_mm_dd$BH_P_val, DE_mCG_CG_mm_dd$GeneID,
                  FC_cutoff=FCval, PV_cutoff=.05, 'Mouse specific - CG vs mCG')
```

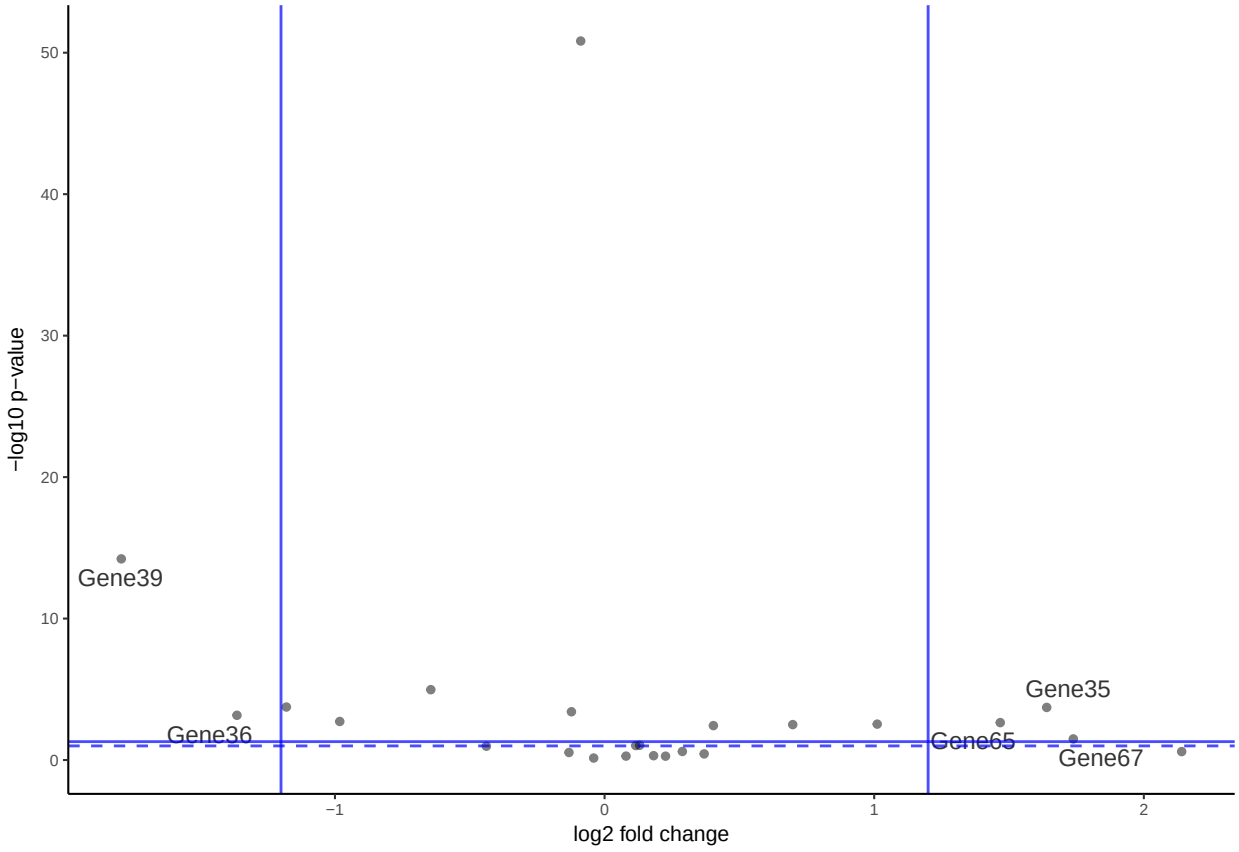


Figure 11. Distribution of p-values and fold changes for differential expression in Mouse.

Presence-Ansence Analysis

Combined Analysis

In the Presence-Ansence Analysis we use only proteins that are NOT in the normalized data. For example, some peptides may have been eliminated for some proteins due to many missing values, but if some peptides remained in the Model-Based Diffential Expression Analysis, we do not analyse a subset of pepties in the Presence-Absence Analysis as we would obtain 2 p-values. We strongly beleave that Model-Based Diffential Expression Analysis is a more sencitive approach and thus it is a preferred method of analysis for proteins that have sufficient number of observations in both treatment groups.

```
# make data structures suitable for get_presAbs_prots() function
raw_list = list()
norm_imp_prot.info_list = list()
raw_list[[1]] = mm_m_ints_eig1$m
raw_list[[2]] = hs_m_ints_eig1$m
norm_imp_prot.info_list[[1]] = mm_m_ints_eig1$prot.info
norm_imp_prot.info_list[[2]] = hs_m_ints_eig1$prot.info

protnames_norm_list = list()
protnames_norm_list[[1]] = unique(mm_m_ints_norm$normalized$MatchedID) #56/69 raw proteins
protnames_norm_list[[2]] = unique(hs_m_ints_norm$normalized$MatchedID) #59

presAbs_dd = get_presAbs_prots(mm_list=raw_list, prot.info=norm_imp_prot.info_list,
                  protnames_norm=protnames_norm_list, prot_col_name=2)
```

```
## [1] "Number of peptides normalized: 1072"
## [1] "Number of peptides Pres/Abs: 30"
## [1] "Number of peptides normalized: 663"
## [1] "Number of peptides Pres/Abs: 32"
```

```
ints_presAbs = list()
protmeta_presAbs = list()
ints_presAbs[[1]] = presAbs_dd[[1]][[1]] # Mouse
ints_presAbs[[2]] = presAbs_dd[[1]][[2]] # HS
protmeta_presAbs[[1]] = presAbs_dd[[2]][[1]]
protmeta_presAbs[[2]] = presAbs_dd[[2]][[2]]

dim(protmeta_presAbs[[2]]) # 32 x 7 peptides
```

```
## [1] 32  7
```

```
length(unique(protmeta_presAbs[[2]]$MatchedID))  # 10 - proteins
```

```
## [1] 10
```

```
dim(protmeta_presAbs[[1]]) # 30 x 7 peptides
```

```
## [1] 30  7
```

```
length(unique(protmeta_presAbs[[1]]$MatchedID))  # 13 - proteins
```

```
## [1] 13
```

```
 # grps do not change
subset_presAbs = subset_proteins(mm_list=ints_presAbs,prot.info=protmeta_presAbs,'MatchedID')
names(subset_presAbs)
```

```
## [1] "sub_mm_list"          "sub_prot.info"        "sub_unique_mm_list"
```

```
## [4] "sub_unique_prot.info" "common_list"
```

```
dim(subset_presAbs$sub_unique_prot.info[[1]])
```

```
## [1] 17  7
```

```
dim(subset_presAbs$sub_unique_prot.info[[2]])
```

```
## [1] 14  7
```

```
dim(subset_presAbs$sub_prot.info[[1]])
```

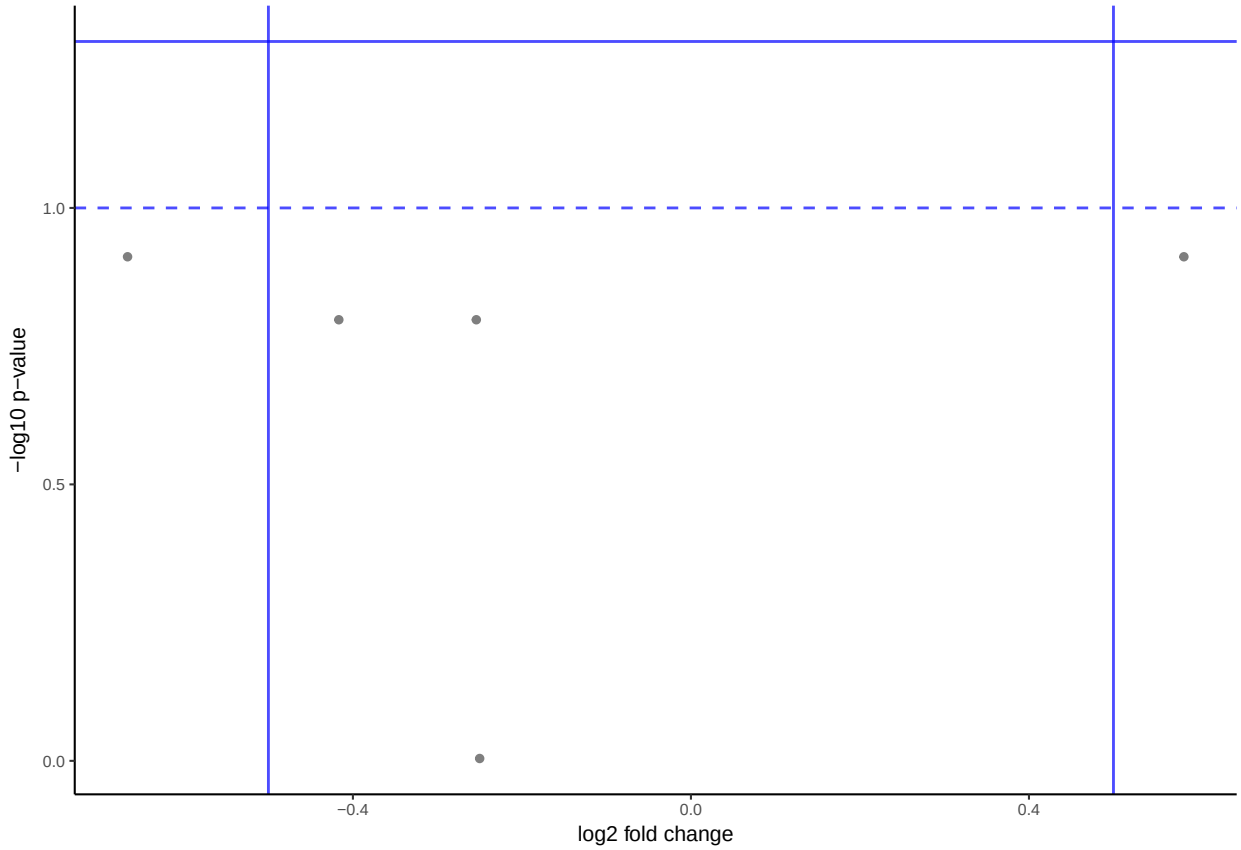
```
## [1] 13  7
```

```
dim(subset_presAbs$sub_prot.info[[2]])
```

```
## [1] 18  7
```

```
nperm = 50  # set to 500+ for publication
ptm <- proc.time()
presAbs_comb = prot_level_multiMat_PresAbs(mm_list=subset_presAbs$sub_mm_list,treat=treats,
                                           prot.info=subset_presAbs$sub_prot.info,
                                           prot_col_name='MatchedID', nperm=nperm,
                                           setseed=123372, dataset_suffix=c('MM', 'HS') )
proc.time() - ptm  #
```

```
plot_volcano_wLab(presAbs_comb$FC, presAbs_comb$BH_P_val, presAbs_comb$GeneID,
                  FC_cutoff=.5, PV_cutoff=.05, 'Combined Pres/Abs CG vs mCG')
```



** Figure 10. **

```
# Presence / Absence analysis for proteins found only in  one or the other dataset
dim(subset_presAbs$sub_unique_mm_list[[1]])
```

```
## [1] 17  6
```

```
dim(subset_presAbs$sub_unique_mm_list[[2]])
```

```
## [1] 14  6
```

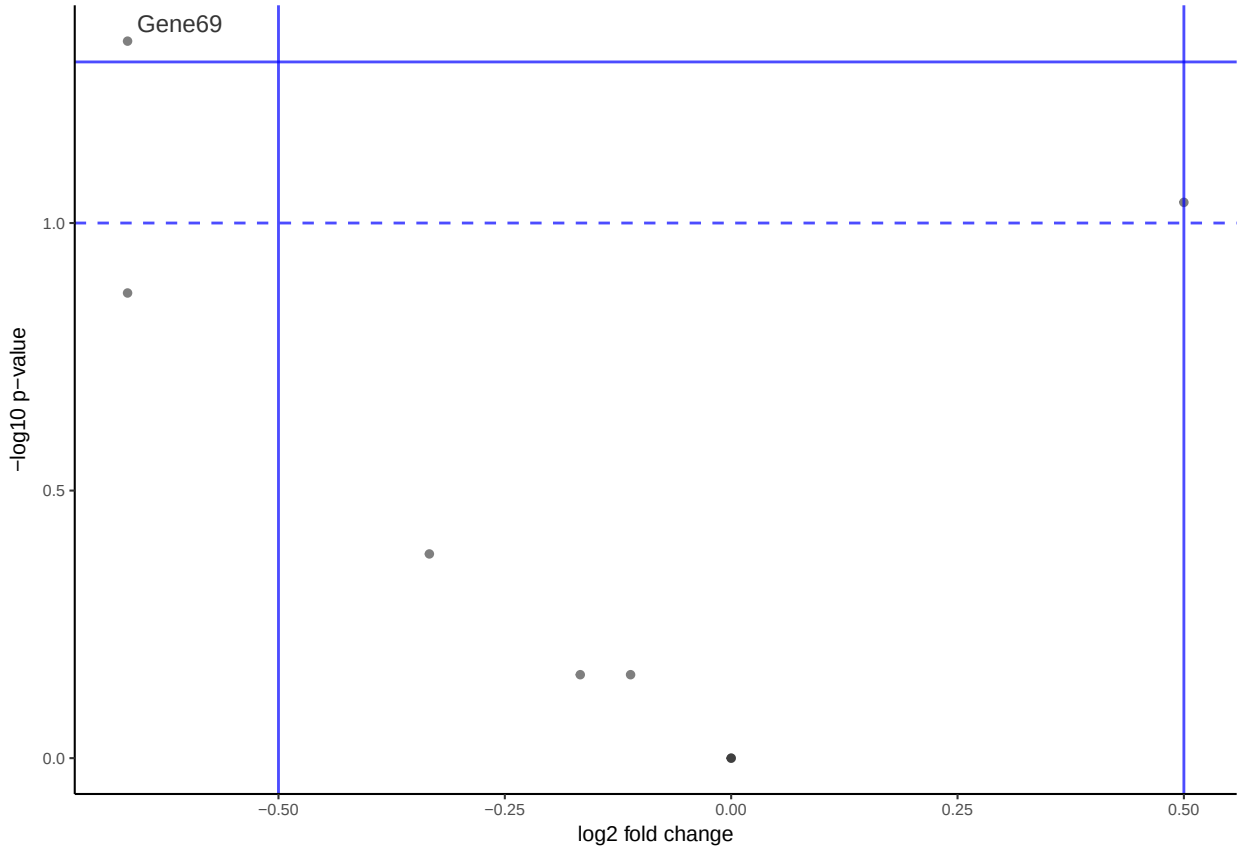
```
unique(subset_presAbs$sub_unique_prot.info[[1]]$ProtID)# 8
```

```
## [1] Prot55 Prot58 Prot45 Prot37 Prot46 Prot69 Prot63 Prot62
## 69 Levels: Prot1 Prot10 Prot11 Prot12 Prot13 Prot14 Prot15 ... Prot9
```

```
unique(subset_presAbs$sub_unique_prot.info[[2]]$ProtID)# 5
```

```
## [1] Prot523 Prot525 Prot527 Prot529 Prot530
## 69 Levels: Prot1 Prot10 Prot11 Prot12 Prot13 Prot14 Prot15 ... Prot9
```

```
## Multi matrix analysis
# Mouse
mm_presAbs = peptideLevel_PresAbsDE(subset_presAbs$sub_unique_mm_list[[1]], treats[[1]],
                                    subset_presAbs$sub_unique_prot.info[[1]], pr_ppos=3)
#mm_presAbs$FC = mm_presAbs$FC * -1
plot_volcano_wLab(mm_presAbs$FC, mm_presAbs$BH_P_val, mm_presAbs$GeneID, FC_cutoff=.5,
                  PV_cutoff=.05, 'MM Pres/Abs CG vs mCG') # look reasonable
```




```
# Human
hs_presAbs = peptideLevel_PresAbsDE(subset_presAbs$sub_unique_mm_list[[2]], treats[[2]],
                                    subset_presAbs$sub_unique_prot.info[[2]], pr_ppos=3)
```

```
# hs_presAbs$FC = hs_presAbs$FC * -1
plot_volcano_wLab(hs_presAbs$FC, hs_presAbs$BH_P_val, hs_presAbs$GeneID, FC_cutoff=.5,
                  PV_cutoff=.05, 'HS Pres/Abs CG vs mCG')
```