Supplementary Statistical Methods

Genetic Disruption of WASHC4 Drives Endo-lysosomal Dysfunction and Cognitive-Movement Impairments in Mice and Humans

Jamie Courtland[1*], Tyler W. A. Bradshaw[1*], Greg Waitt[2], Erik J. Soderblom[2,3], Tricia Ho[2], Anna Rajab[4], Ricardo Vancini[5], Il Hwan Kim[2†], Ting Huang[6], Olga Vitek[6], Scott H. Soderling[3]

[1]Department of Neurobiology, Duke University School of Medicine, Durham, NC 27710, USA; [2]Proteomics and Metabolomics Shared Resource, Duke University School of Medicine, Durham, NC 27710, USA; [3]Department of Cell Biology, Duke University School of Medicine, Durham, NC 27710, USA; [4]Burjeel Hospital, VPS Healthcare, Muscat, Oman; [5]Department of Pathology, Duke University School of Medicine, Durham, NC 27710, USA; [6]Khoury College of Computer Sciences, Northeastern University, Boston, MA 02115, USA

Author correspondence: jamie.courtland@duke.edu (JC); tyler.w.bradshaw@duke.edu (TWAB); greg.waitt@duke.edu (GW); erik.soderblom@duke.edu (EJB); tricia.ho@duke.edu (TH); drannarajab@gmail.com (DR); ricardo.vancini@duke.edu (RV); ikim9@uthsc.edu (IK); huang.tin@northeastern.edu (TH); o.vitek@northeastern.edu (OV); scott.soderling@duke.edu (SHS)

*These authors contributed equally to this work.

Present address:
[†]Department of Anatomy and Neurobiology, University of Tennessee Health Science Center, Memphis, TN 38163, USA

Summary

Here we address concerns about the statistical validity of our previous approach to assess differential protein abundance in the **WASH-iBioID** and **SWIP-TMT** proteomics datasets. Our previous approach depended upon the R package `edgeR`. We used `edgeR` to perform both protein- and module-level inference—assessing differential abundance of individual proteins as well as protein groups in SWIP$^{\mathrm{P1019R}}$ mouse brain. `edgeR` utilizes a negative binomial (NB) statistical framework originally developed for analysis of RNA-Seq read count data. Previously, we failed to fully consider the validity of `edgeR's` NB assumption for proteomics data. We evaluate the goodness-of-fit of the negative binomial model for our TMT dataset and find evidence of a lack-of-fit. Thus, we revise our statistical approach and reanalyze our data, making use of ***Huang et al.*** (***2020***)'s recently published R package `MSstatsTMT`. `MSstatsTMT` uses a flexible linear mixed-model (LMM) statistical framework which we extend to re-evaluate both protein- and module-level statistical comparisions in our SWIP-TMT spatial proteomics dataset.

Goodness-of-fit of the NB Model for TMT MS

Our previous method can be summarized as the *Sum + IRS* approach (***Huang et al., 2020***). Following protein summarization and internal reference scaling (IRS) normalization (***Plubell et al., 2017***), we applied `edgeR` (***McCarthy et al., 2012***) to assess differential abundance of individual proteins and protein-groups. The use of `edgeR` for protein-level comparisons was based on work by ***Plubell et al.*** (***2017***) who describe IRS normalization and the use of `edgeR` for statistical testing in TMT MS experiments (***Plubell et al., 2017***). We failed however, to consider the overall adequecy of `edgeR's` NB GLM model for our TMT proteomics data.

Statistical inference in `edgeR` is performed for each gene or protein using a negative binomial, generalized linear model framework. The data are assumed to be adequately described by a NB distribution parameterized by a dispersion parameter, ϕ. The dispersion parameter ϕ accounts for mean-variance relationships in proteomics and transcriptomics data. As signal intensity in protein MS is fundamentally related to the number of ions generated from an ionized, fragmented protein, we incorrectly inferred that TMT mass spectrometry data can be modeled as NB count data. Based on this assumption, we justified our use of `edgeR`.

To evaluate the overall adequacy of the negative binomial model for TMT proteomics data, we plot the residual protein deviance statistics of all proteins fit with `edgeR's` NB GLM against their theoretical normal quantiles in a quantile-quantile (QQ) plot (FIG:gof). The QQ plot addresses the question of how similar the observed data are to the theoretical distribution given by the NB model. A linear relationship between the observed and theoretical values is a goodness-of-fit indicator. Deviation from this linear trend is evidence of a lack-of-fit.

Following protein summarization and normalization with `MSstatsTMT`, the SWIP-TMT data were fit with a NB GLM using `edgeR::glmFit`. ***Figure 2*** illustrates the divergence of the observed and theoretical quantiles for our SWIP-TMT dataset fit with `edgeR's` NB GLM. Given our experimental design, `MSstatsTMT` fits an appropriate linear-mixed model to the data. The quantile-quantile plot in ***Figure 2*** indicates that the data are well described by `MSstatsTMT's` LMM, which does not depend upon the negative binomial assumption.

Protein-wise Linear Mixed-Models

Huang et al. (***2020***) created `MSstatsTMT`, an R package for data normalization and hypothesis testing in multiplex TMT proteomics experiments. `MSstatsTMT` performs statistical inference in two steps. First, each protein in the dataset is fit with a LMM expressing the major sources of variation in the experimental design. Second, given the fitted model, a model-based comparison is made between pairs of treatment conditions. Using LMMs we can untangle the variance attributable to the biological effect we are interested in from the experimental and biological covariates which mask this response.

Huang et al. (***2020***) outline a common vocabulary for describing TMT MS experimental design. An experiment consists of `m = 1 ... M` concatenations of isobarically labeled samples or `Mixtures`. This mixture is then analyzed by the mass spectrometer in a single MS `Run`. This mixture is often fractionated into multiple liquid chromotography `Fractions` to decrease sample complexity, and thereby increase the depth of proteome coverage. Within a mixture, each of the unique TMT channels is dedicated to the analysis of `c = 1 ... C` individual biological or treatment `Conditions`. There may then be `b = 1 ... B` biological replicates or `Subjects`. Finally, a single TMT mixture may be repeatedly analyzed in `t = 1 ... T` technical replicate mass spectrometry runs.

Equation 1 is a LMM describing protein abundance as a function of the major sources of variation in a general TMT experiment composed of `M` mixtures, `T` technical replicates of mixture, `C` conditions, and `B` biological subjects.

$$Y_{mcbt} = \mu + Condition_c + Mixture_m + TechRep(Mixture)_{m(t)} + Subject_{mcb} + \epsilon_{mcbt} \tag{1}$$

$$\begin{gathered}
\sum_{c=1}^{C} Condition_c = 0 \\
Mixture_m \overset{iid}{\sim} N(0, \sigma_M^2) \\
TechRep(Mixture)_{t(m)} \overset{iid}{\sim} N(0, \sigma_T^2) \\
Subject_{mcb} \overset{iid}{\sim} N(0, \sigma_S^2) \\
\epsilon mtcb \overset{iid}{\sim} N(0, \sigma^2)
\end{gathered} \tag{2}$$

The model's constraints 2 distinguish fixed- and mixed-effect components of variation in the response, Y_{mcbt}. `Mixture` is a mixed-effect and represents variation between different TMT mixtures. By definition mixed-effects are assumed to be normally and independently distributed (`iid`). The term `TechRep(Mixture)` represents random variation between replicates of a single MS `Run`. `Subject` corresponds to each unique biological replicate and represents biological variation among the levels of the fixed-effect term `Condition`. The term ϵ_{mtcb} is a mixed-effect representing both biological and technical variation, quantifying any remaining error. If a component of the model is not estimable, then it is removed. For example, if there is no technical replication of mixture `(T=0)`, then the model is reduced to equation 3.

$$Y_{mcbt} = \mu + Condition_c + Mixture_m + Subject_b + \epsilon_{mcb} \tag{3}$$

SWIP-TMT Spatial Proteomics

We analyzed the brains of mice with the SWIP^{P1019R} mutation by subcellular fractionation and TMT MS profiling. We aimed to reveal how this pathogenic mutation may perturb the organization of the subcellular proteome. We adapted the subcellular fractionation method of ***Geladaki et al.*** (***2019***) to prepare seven subcellular fractions from the brains of control and SWIP$^{\mathrm{P1019R}}$ mutant mice. Our experimental design is summarized in ***Figure 7***.

Each 16-plex TMT `Mixture` was composed of fourteen biological fractions or `BioFractions` obtained from subcellular fractionation of a control and SWIP$^{\mathrm{P1019R}}$ mutant mouse brain. We refer to these subcellular preparations as a `BioFractions` to distinguish them from an MS `Fraction`. The term `Condition` of equation 3 represents these fourteen combinations of `Genotype` and `BioFraction`.

In our design, `Mixture` is confounded with `Subject`. We analyzed all seven `BioFractions` from a single control and mutant mouse in the same `Mixture`. We choose to model the effect of `Mixture` and not `Subject` based on the assumption that the experimental batch effect represented by the term `Mixture` is greater than the error inherent in the repeated measures of each `Subject`. We omit the unestimable terms `TechRep(Mixture)` and `Subject` from equation (1). The reduced linear mixed-model describing our experimental design is given by equation 4.

$$Y_{mcbt} = \mu + Condition_c + Mixture_m + \epsilon_{mcb} \tag{4}$$

Statistical Inference with MSstatsTMT

`MSstatsTMT` performs protein-wise comparisons between pairs of `Conditions` by comparing the estimates obtained from the LMM fit by restricted maximum likelihood (***Bates et al., 2015***). We are interested in testing the hypothesis:

$$H0 : l^T * \beta = 0 \tag{5}$$

Where l^T is a vector of $\sum = 1$ specifying the positive and negative coefficients of a contrast. β is the model-based estimates of `Condition`. The null hypothesis (5) is that the fold change, $\mathit{lll}^T * \beta$, is 0. A test statistic for such a two-way contrasts is given by ***Kuznetsova et al.*** (***2017***):

$$t = \frac{l^T \hat{\beta}}{\sqrt{l \sigma^2 \hat{V} l^T}} \tag{6}$$

We obtain the model's estimates, $\hat{\beta}$, error,σ^2, and variance-covariance matrix,$\hat{V}$, from the fit LMM. Given a contrast, l^T, the numerator of equation (6) is the fold change of a comparison. The product of σ^2 and $\hat{V}$ is the scaled variance-covariance matrix describing error estimates of the model's fixed- and mixed-effect parameters. Together the denominator represents the standard error of the comparison. The degrees of freedom for the contrast are derived using the Satterthwaite moment of approximation method (***Kuznetsova et al., 2017***). Finally, a p-value is calculated given the t-statistic and degrees of freedom. P-values for the protein-wise tests are adjusted using the Benjamini-Hochberg FDR method (***Huang et al., 2020***).

We used MSstatsTMT to assess two types of contrasts. Intra-BioFraction comparisons are the seven pairwise comparisons of control and mutant protein abundance for each BioFraction. We also assessed the overall Mutant-Control comparison. Each of these contrasts is represented by a vector, l^T, which specifies a contrast between coefficients of Condition in the LMM (4). ***Figure 8*** illustrates the two types of protein-level statistical comparisons we implement with MSstatsTMT.

Module-level Inference with Mixed-Models

The strength of linear mixed-models lies in their flexibility. In a mixed-model the response variable is taken to be a function of both fixed- and random-effects. If the set of possible levels of a covariate is fixed and reproducible, then the factor is modeled as a fixed-effect parameter. In contrast, if the levels of an observation reflect a sampling of the set of all possible levels, then the covariate is modeled as a random-effect. Random or mixed-effects represent categorical variables that reflect experimental or observational units within the dataset. As such, mixed-effect parameters account for the variation occurring among lower levels of an upper level unit in the data (***Bates et al., 2015***).

We wish to extend the LMM framework developed by MSstatsTMT to perform inference at the level of protein groups. Given a map partitioning the proteome into modules of covarying proteins, we wish to assess the module-level difference between control and SWIP$^{\mathrm{P1019R}}$ conditions. We fit the data for each module in the dataset with a LMM. We represent the proteins within each module as the mixed-effect term Protein, capturing variation among a module's constintuent proteins.

$$\begin{gathered} Y_{mcbt} = \mu + Condition_c + Mixture_m + Protein_p + \epsilon_{mcb} \\ Protein_p \overset{iid}{\sim} N(0, \sigma_P^2) \end{gathered} \tag{7}$$

The term Protein in equation 7 quantifies the variance σ_P attributable to heterogenity among a modules proteins.

LMM Goodness-of-fit

It is useful to consider the goodness-of-fit of our models. A straight forward measure of a LMM's quality is the Nakagawa coefficient of determination (***Nakagawa and Schielzeth, 2012***). ***Nakagawa and Schielzeth*** (***2012***)'s conditional R_c^2 is interpreted as the total variance explained by a LMM (R_{total}^2). The marginal R_m^2 is interpreted as the variance explained by the LMM's fixed-effects (R_{fixed}^2). We implement ***Nakagawa and Schielzeth*** (***2012***)'s coeffficient of determination using the `r.squaredGLMM` function taken from the `MuMin` package (***Wang and Merkle, 2018***).

In addition to considering the total variance explained by a module, it is helpful to consider the variance explained by each of its factors. The R package `variancePartition` enables us to calculate the percent variance explained by a LMM's parameters cite(variancePartition).

Spatial Proteomics Network Construction

Using our SWIP-TMT dataset, we aim to identify modules or groups of proteins that covary together across subcellular space. Prior to building the co-variation network, other sources of variation should be removed. Although `MSstatsTMT` handles the batch effect inherent in experiments with multiple TMT mixtures, it is necessary to remove this effect prior to building the network. We removed the effect of `Mixture` using `limma::RemoveBatchEffect`. These adjusted data are used for network construction and plotting, but not statistical modeling.

Prior to network construction, we removed protein models with poor fit ($R^2_{total} < 0.7$; n=791 proteins). Removing this noisey proteins facilitation module identification and improves overall module quality.

The final network was constructed using data from both Control and Mutant samples after adjusting for batch (Mixture). The final dataset included 42 samples and 6,119 proteins. The protein covariation network was build by calculating the Pearson correlation for all pairwise comparisons of proteins.

We performed network enhancement to remove biological noise from the network prior to clustering. This step is essential in large, and dense networks for module detection. Network enhancement reweights the network's edges and has the overall effect of making the network sparse. Conceptually this step is related to the soft-thresholding approach taken by WGCNA or WPCNA analysis workflows (REFS), but has the benefit of not assuming that the network has an overall scale-free topology. Without reweighting or enhancing the network, most extant clustering algorithms fail to detect communities in the dataset. Network enhancement has the effect of making the network sparse and facilitates the identification of network structure.

Community Detection with Leidenalg

To reveal the structure of our spatial proteomics network we used the recently published Leiden algorithm (***Traag et al., 2019***).

Implementation

In order to understand and extend the function of `MSstatsTMT`, we extracted `MSstatsTMT's` core model-fitting and statistical testing steps. At the core of the model fitting-step is the R package `lme4` which implements mixed-effects models with its function `lme4::lmer`(***Bates et al., 2015***). The package `lmerTest` extends `lme4's` functionality and enables the computation of Sattertwaite degrees of freedom (***Kuznetsova et al., 2017***).

Fit WASHC4

As a means of example, we demonstrate the model fitting statistical testing steps for both and protein- and module-level statistical comparisons. First, we fit the LMM in (4) the normalized protein level data from `MSstatsTMT` for WASHC4.

```
## fit the protein-level model to WASHC4

# load dependencies
library(dplyr)
library(lmerTest)

# load SwipProteomics
data(swip)
data(msstats_prot)

# LMM formula
fx0 <- 'Abundance ~ 0 + Genotype:BioFraction + (1|Mixture)'

# fit the model
fm0 <- lmer(fx0, data = msstats_prot %>% subset(Protein == swip))

# examine the model's summary
summary(fm0, ddf = "Satterthwaite")
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: fx0
##    Data: msstats_prot %>% subset(Protein == swip)
##
## REML criterion at convergence: 3.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.5030 -0.6089 -0.1463  0.7474  1.4302
```

```
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Mixture  (Intercept) 0.009596 0.09796
##  Residual             0.034418 0.18552
## Number of obs: 42, groups:  Mixture, 3
##
## Fixed effects:
##                                Estimate Std. Error      df t value Pr(>|t|
## GenotypeMutant:BioFractionF4     5.4043     0.1211 17.3059   44.62   <2e-1
## GenotypeControl:BioFractionF4    6.7110     0.1211 17.3059   55.41   <2e-1
## GenotypeMutant:BioFractionF5     5.5674     0.1211 17.3059   45.96   <2e-1
## GenotypeControl:BioFractionF5    6.9456     0.1211 17.3059   57.34   <2e-1
## GenotypeMutant:BioFractionF6     5.6402     0.1211 17.3059   46.56   <2e-1
## GenotypeControl:BioFractionF6    7.2401     0.1211 17.3059   59.77   <2e-1
## GenotypeMutant:BioFractionF7     5.6317     0.1211 17.3059   46.49   <2e-1
## GenotypeControl:BioFractionF7    7.3211     0.1211 17.3059   60.44   <2e-1
## GenotypeMutant:BioFractionF8     5.4928     0.1211 17.3059   45.35   <2e-1
## GenotypeControl:BioFractionF8    7.1296     0.1211 17.3059   58.86   <2e-1
## GenotypeMutant:BioFractionF9     5.7810     0.1211 17.3059   47.73   <2e-1
## GenotypeControl:BioFractionF9    6.9545     0.1211 17.3059   57.41   <2e-1
## GenotypeMutant:BioFractionF10    5.7844     0.1211 17.3059   47.76   <2e-1
## GenotypeControl:BioFractionF10   7.6187     0.1211 17.3059   62.90   <2e-1
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

’MutantF7-ControlF7’

We assess the contrast between BioFraction seven (F7) mutant and control conditions.

```
## compare 'Mutant:F7' and 'Control:F7' conditions

# create a contrast
coeff <- lme4::fixef(fm0)
contrast7 <- setNames(rep(0,length(coeff)), nm = names(coeff))
contrast7["GenotypeMutant:BioFractionF7"] <- +1 # positive coeff
contrast7["GenotypeControl:BioFractionF7"] <- -1 # negative coeff

# evaluate contrast
lmerTestContrast(fm0, contrast7)
```

```
##                                                      Contrast    log2FC
## 1 GenotypeMutant:BioFractionF7-GenotypeControl:BioFractionF7 -1.689393
##   percentControl        SE Tstatistic      Pvalue DF isSingular
## 1      0.3100573 0.1514779  -11.15274 2.08622e-11 26      FALSE
```

'Mutant-Control'

Provided the correct contrast, we easily evaluate the overall difference between mutant and control mice.

```
# create a contrast to compare 'Mutant' versus 'Control'
contrast8 <- getContrast(fm0, "Mutant","Control")

# evaluate contrast
lmerTestContrast(fm0, contrast8)
```

```
##                                                        Contrast    log2FC
## 1   GenotypeMutant:BioFractionF4-GenotypeControl:BioFractionF4 -1.516956
## 2   GenotypeMutant:BioFractionF5-GenotypeControl:BioFractionF5 -1.516956
## 3   GenotypeMutant:BioFractionF6-GenotypeControl:BioFractionF6 -1.516956
## 4   GenotypeMutant:BioFractionF7-GenotypeControl:BioFractionF7 -1.516956
## 5   GenotypeMutant:BioFractionF8-GenotypeControl:BioFractionF8 -1.516956
## 6   GenotypeMutant:BioFractionF9-GenotypeControl:BioFractionF9 -1.516956
## 7 GenotypeMutant:BioFractionF10-GenotypeControl:BioFractionF10 -1.516956
##   percentControl         SE Tstatistic       Pvalue DF isSingular
## 1      0.3494225 0.05725328  -26.49552 2.423534e-20 26      FALSE
## 2      0.3494225 0.05725328  -26.49552 2.423534e-20 26      FALSE
## 3      0.3494225 0.05725328  -26.49552 2.423534e-20 26      FALSE
## 4      0.3494225 0.05725328  -26.49552 2.423534e-20 26      FALSE
## 5      0.3494225 0.05725328  -26.49552 2.423534e-20 26      FALSE
## 6      0.3494225 0.05725328  -26.49552 2.423534e-20 26      FALSE
## 7      0.3494225 0.05725328  -26.49552 2.423534e-20 26      FALSE
```

Fit WASH Complex

Next we fit a LMM to the five WASH complex proteins.

```
# the module-level formula to be fit:
fx1 <- 'Abundance ~ 0 + Condition + (1|Mixture) + (1|Protein)'

# load WASH Complex proteins
data(washc_prots)
```

```
fm1 <- lmer(fx1, data=msstats_prot %>% subset(Protein %in% washc_prots))

# assess 'Mutant-Control' comparison
lmerTestContrast(fm1, contrast8)
```

```
##                                                        Contrast    log2FC
## 1   GenotypeMutant:BioFractionF4-GenotypeControl:BioFractionF4 0.2305437
## 2   GenotypeMutant:BioFractionF5-GenotypeControl:BioFractionF5 0.2305437
## 3   GenotypeMutant:BioFractionF6-GenotypeControl:BioFractionF6 0.2305437
## 4   GenotypeMutant:BioFractionF7-GenotypeControl:BioFractionF7 0.2305437
## 5   GenotypeMutant:BioFractionF8-GenotypeControl:BioFractionF8 0.2305437
## 6   GenotypeMutant:BioFractionF9-GenotypeControl:BioFractionF9 0.2305437
## 7 GenotypeMutant:BioFractionF10-GenotypeControl:BioFractionF10 0.2305437
##   percentControl         SE Tstatistic       Pvalue  DF isSingular
## 1       1.173277 0.03698955   6.232669 2.894644e-09 190      FALSE
## 2       1.173277 0.03698955   6.232669 2.894644e-09 190      FALSE
## 3       1.173277 0.03698955   6.232669 2.894644e-09 190      FALSE
## 4       1.173277 0.03698955   6.232669 2.894644e-09 190      FALSE
## 5       1.173277 0.03698955   6.232669 2.894644e-09 190      FALSE
## 6       1.173277 0.03698955   6.232669 2.894644e-09 190      FALSE
## 7       1.173277 0.03698955   6.232669 2.894644e-09 190      FALSE
```

We evaluate the goodness of fit of our module-level model.

```
# assess gof with Nakagawa coefficient of determination
r.squaredGLMM.merMod(fm0)
```

```
##            R2m      R2c
## [1,] 0.9353344 0.949433
```

```
r.squaredGLMM.merMod(fm1)
```

```
##            R2m       R2c
## [1,] 0.7620866 0.8928053
```

varianceParition

We compute the variance explained using `variancePartition`.

```
# load variancePartition
suppressPackageStartupMessages({
        library(variancePartition)
})
```

```
# calculate partitioned variance
form <- "Abundance ~ (1|Genotype) + (1|BioFraction) + (1|Mixture) + (1|Protei
fit <- lmer(form, data = msstats_prot %>% filter(Protein %in% washc_prots))

calcVarPart(fit)
```

References

Bates D, Mächler M, Bolker B, Walker S. Fitting Linear Mixed-Effects Models Usinglme4. Journal of Statistical Software. 2015; 67(1). https://doi.org/10.18637%2Fjss.v067.i01, doi: 10.18637/jss.v067.i01.

Geladaki A, Britovšek NK, Breckels LM, Smith TS, Vennard OL, Mulvey CM, Crook OM, Gatto L, Lilley KS. Combining LOPIT with differential ultracentrifugation for high-resolution spatial proteomics. Nature Communications. 2019 jan; 10(1). https://doi.org/10.1038%2Fs41467-018-08191-w, doi: 10.1038/s41467-018-08191-w.

Huang T, Choi M, Tzouros M, Golling S, Pandya NJ, Banfai B, Dunkley T, Vitek O. MSstatsTMT: Statistical Detection of Differentially Abundant Proteins in Experiments with Isobaric Labeling and Multiple Mixtures. Molecular & Cellular Proteomics. 2020 Jul; 19(10):1706–1723. https://doi.org/10.1074/mcp.ra120.002105, doi: 10.1074/mcp.ra120.002105.

Kuznetsova A, Brockhoff PB, Christensen RHB. lmerTest Package: Tests in Linear Mixed Effects Models. Journal of Statistical Software. 2017; 82(13). https://doi.org/10.18637/jss.v082.i13, doi: 10.18637/jss.v082.i13.

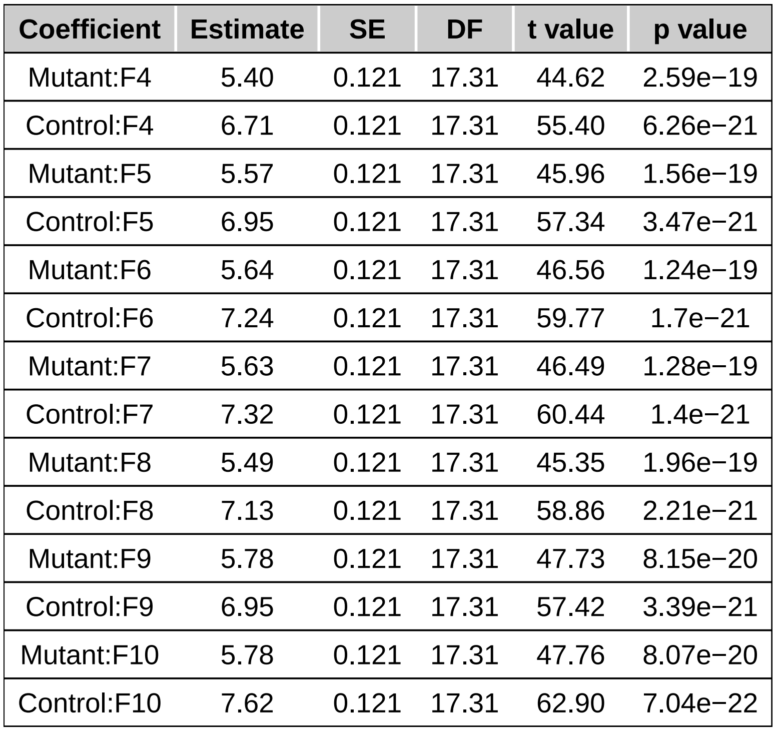
McCarthy DJ, Chen Y, Smyth GK. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. Nucleic Acids Research. 2012 jan; 40(10):4288–4297. https://doi.org/10.1093%2Fnar%2Fgks042, doi: 10.1093/nar/gks042.

Nakagawa S, Schielzeth H. A general and simple method for obtainingR2from generalized linear mixed-effects models. Methods in Ecology and Evolution. 2012 dec; 4(2):133–142. https://doi.org/10.1111%2Fj.2041-210x.2012.00261.x, doi: 10.1111/j.2041-210x.2012.00261.x.

Plubell DL, Wilmarth PA, Zhao Y, Fenton AM, Minnier J, Reddy AP, Klimek J, Yang X, David LL, Pamir N. Extended Multiplexing of Tandem Mass Tags (TMT) Labeling Reveals Age and High Fat Diet Specific Proteome Changes in Mouse Epididymal Adipose Tissue. Molecular & Cellular Proteomics. 2017 Mar; 16(5):873–890. https://doi.org/10.1074/mcp.m116.065524, doi: 10.1074/mcp.m116.065524.

Traag VA, Waltman L, van Eck NJ. From Louvain to Leiden: guaranteeing well-connected communities. Scientific Reports. 2019 mar; 9(1). https://doi.org/10.1038%2Fs41598-019-41695-z, doi: 10.1038/s41598-019-41695-z.

Wang T, Merkle EC. merDeriv: Derivative Computations for Linear Mixed Effects Models with Application to Robust Standard Errors. Journal of Statistical Software. 2018; 87(Code Snippet 1). https://doi.org/10.18637%2Fjss.v087.c01, doi: 10.18637/jss.v087.c01.



Coefficient	Estimate	SE	DF	t value	p value
Mutant:F4	5.40	0.121	17.31	44.62	2.59e−19
Control:F4	6.71	0.121	17.31	55.40	6.26e−21
Mutant:F5	5.57	0.121	17.31	45.96	1.56e−19
Control:F5	6.95	0.121	17.31	57.34	3.47e−21
Mutant:F6	5.64	0.121	17.31	46.56	1.24e−19
Control:F6	7.24	0.121	17.31	59.77	1.7e−21
Mutant:F7	5.63	0.121	17.31	46.49	1.28e−19
Control:F7	7.32	0.121	17.31	60.44	1.4e−21
Mutant:F8	5.49	0.121	17.31	45.35	1.96e−19
Control:F8	7.13	0.121	17.31	58.86	2.21e−21
Mutant:F9	5.78	0.121	17.31	47.73	8.15e−20
Control:F9	6.95	0.121	17.31	57.42	3.39e−21
Mutant:F10	5.78	0.121	17.31	47.76	8.07e−20
Control:F10	7.62	0.121	17.31	62.90	7.04e−22

Figure 1. This is a caption.

Supplemental Tables

Supplemental Figures

- gof
- design
- contrasts
- ...

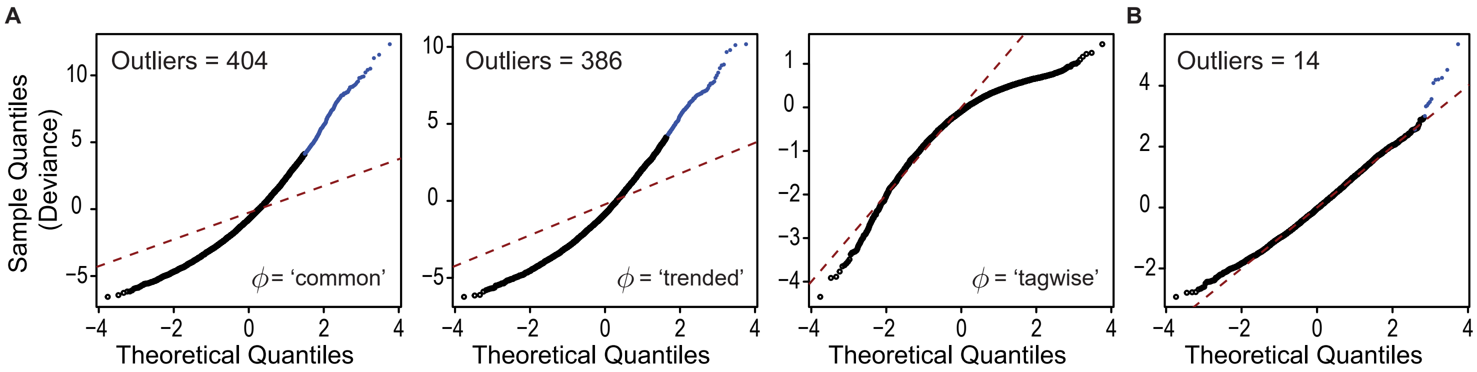


Figure 2. Goodness-of-fit of edgeR (A), and MSstatsTMT (B) statistical approaches. The overall adequacy of the linear models fit to the data were assessed by plotting the residual deviance for all proteins as a quantile-quantile plot (McCarthy *et al.*, (2012)). **(A)** For analysis with edgeR, The normalized protein data from MSstatsTMT were fit with a negative binomal generalized linear model of the form: Abundance ∼ Mixture + Condition. Where Mixure is an additive blocking factor that accounts for variablity between experiments. The NB framework used by edgeR utilizes a dispersion parameter ψ to account for mean-variance relationships in the data. The dispersion parameter can take several forms including: 'common', 'trended', and 'tagwise'. We plot the deviance statistics for the data fit with each of the three disperions parameters against their theoretical normal quantiles using the edgeR::gof function. **(B)** For analysis with MSstatsTMT, the normalized protein data were fit with a linear mixed-effects model (LMM) of the form: Abundance ∼ 0 + Condition + (1|Mixture). Where Mixture represents the mixed-effect of Mixture. The residual deviance and degrees of freedom were extracted from the fitted models, z-score normalized, and plotted as in (A). Proteins with a significantly poor fit are indicated as outliers in blue (Holm-adjusted P-value < 0.05).

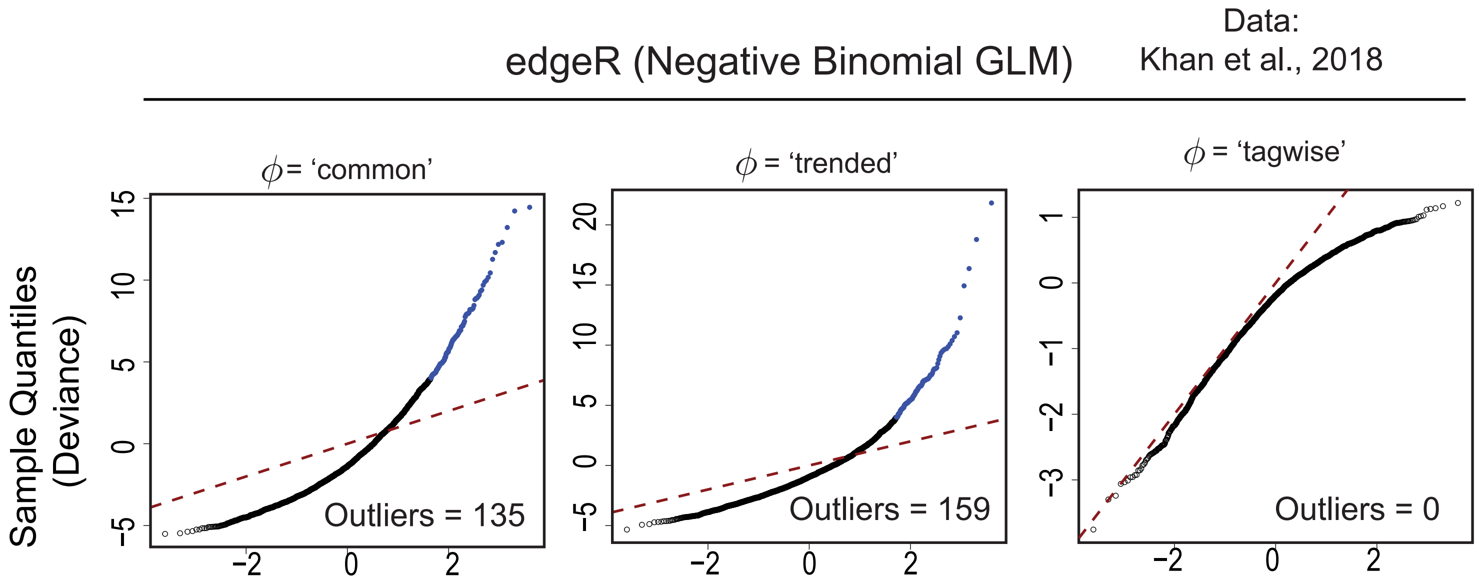


Figure 3. Goodness-of-fit for the edgeR NB GLM for the Khan *et al*, (2018) dataset.

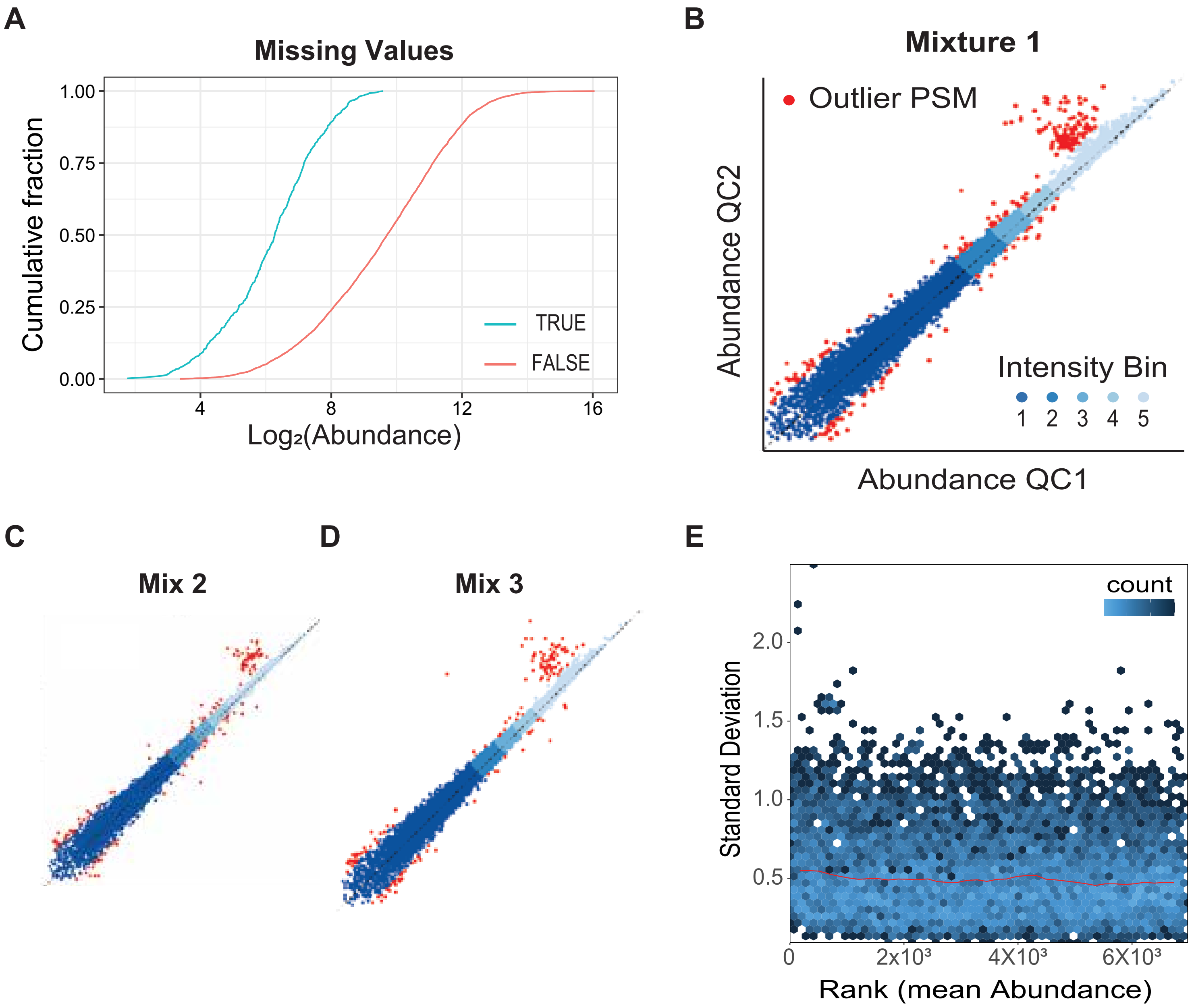


Figure 4. Missing value imputation and PSM outlier removal. A B C D

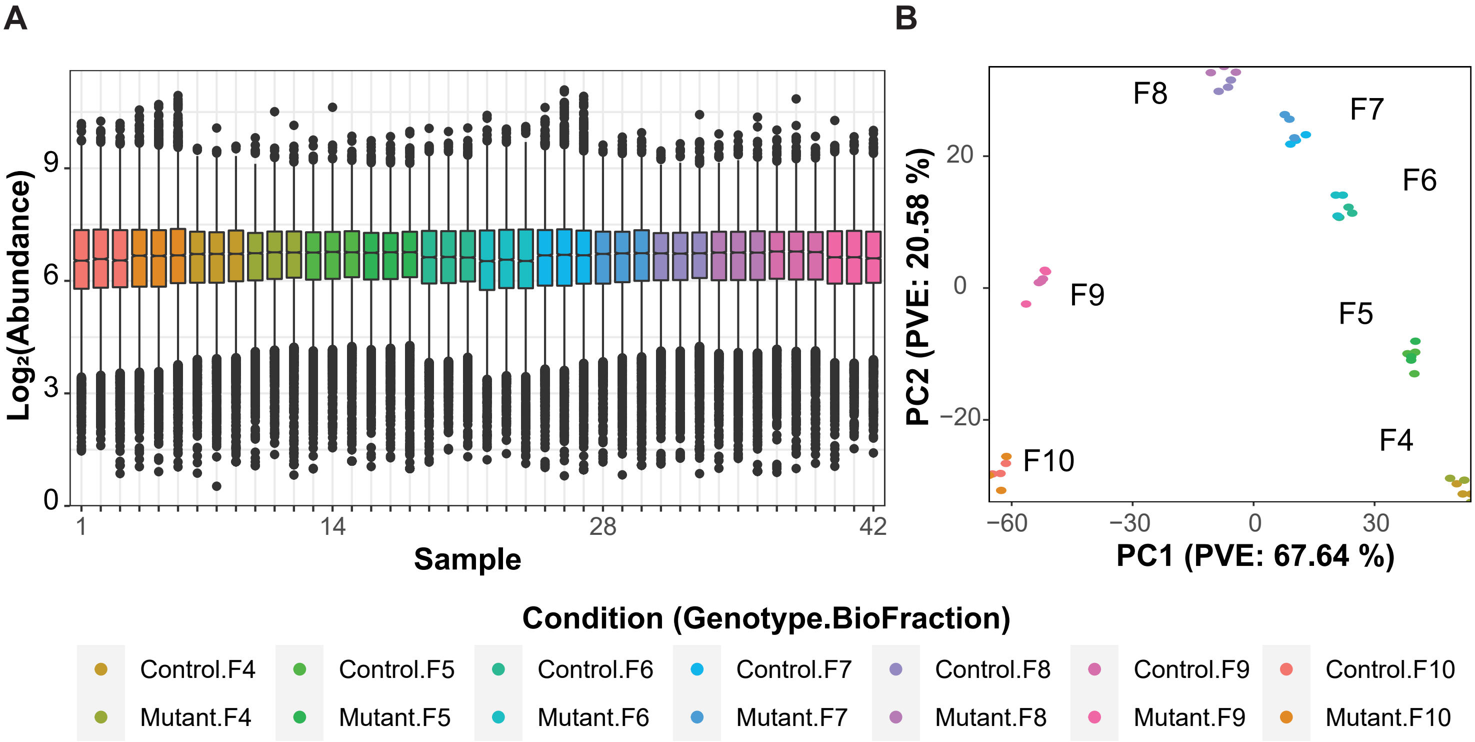


Figure 5. Data Normalization and PCA. A B

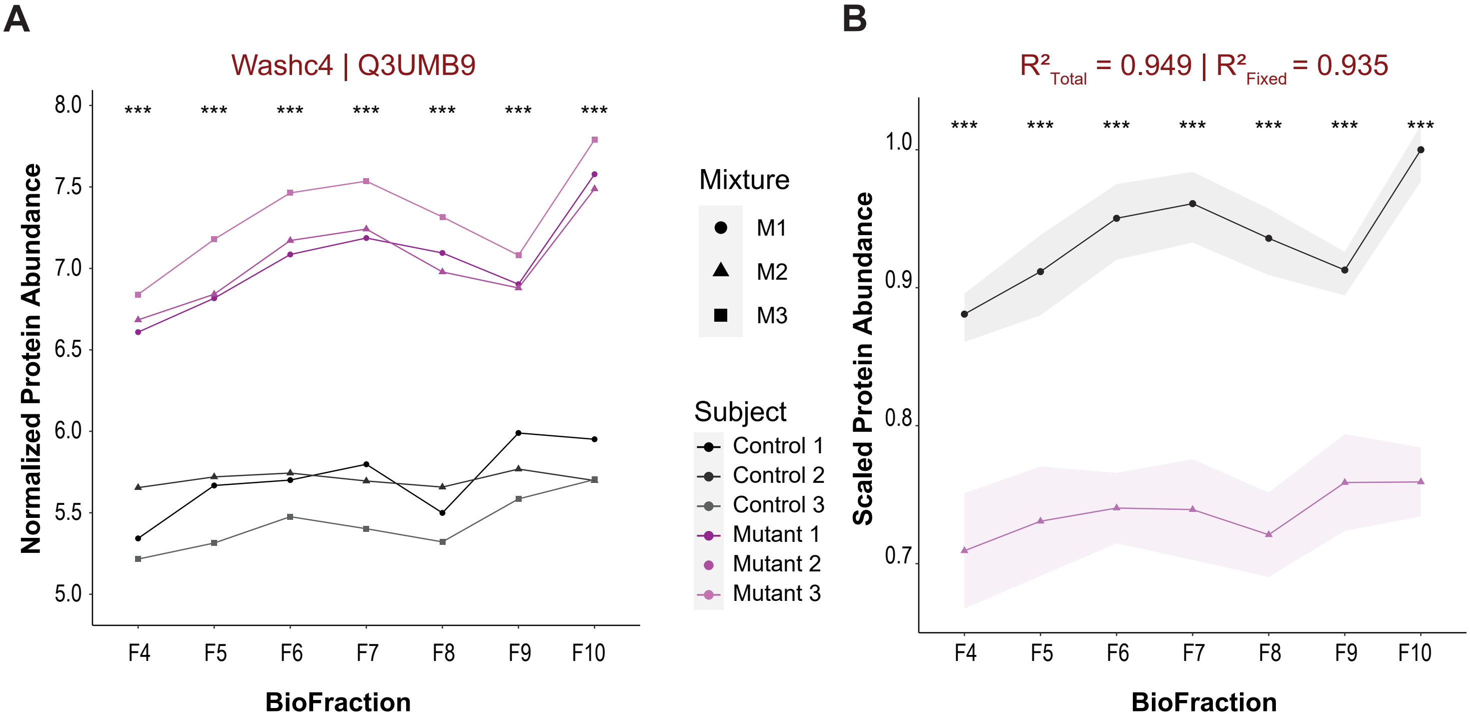
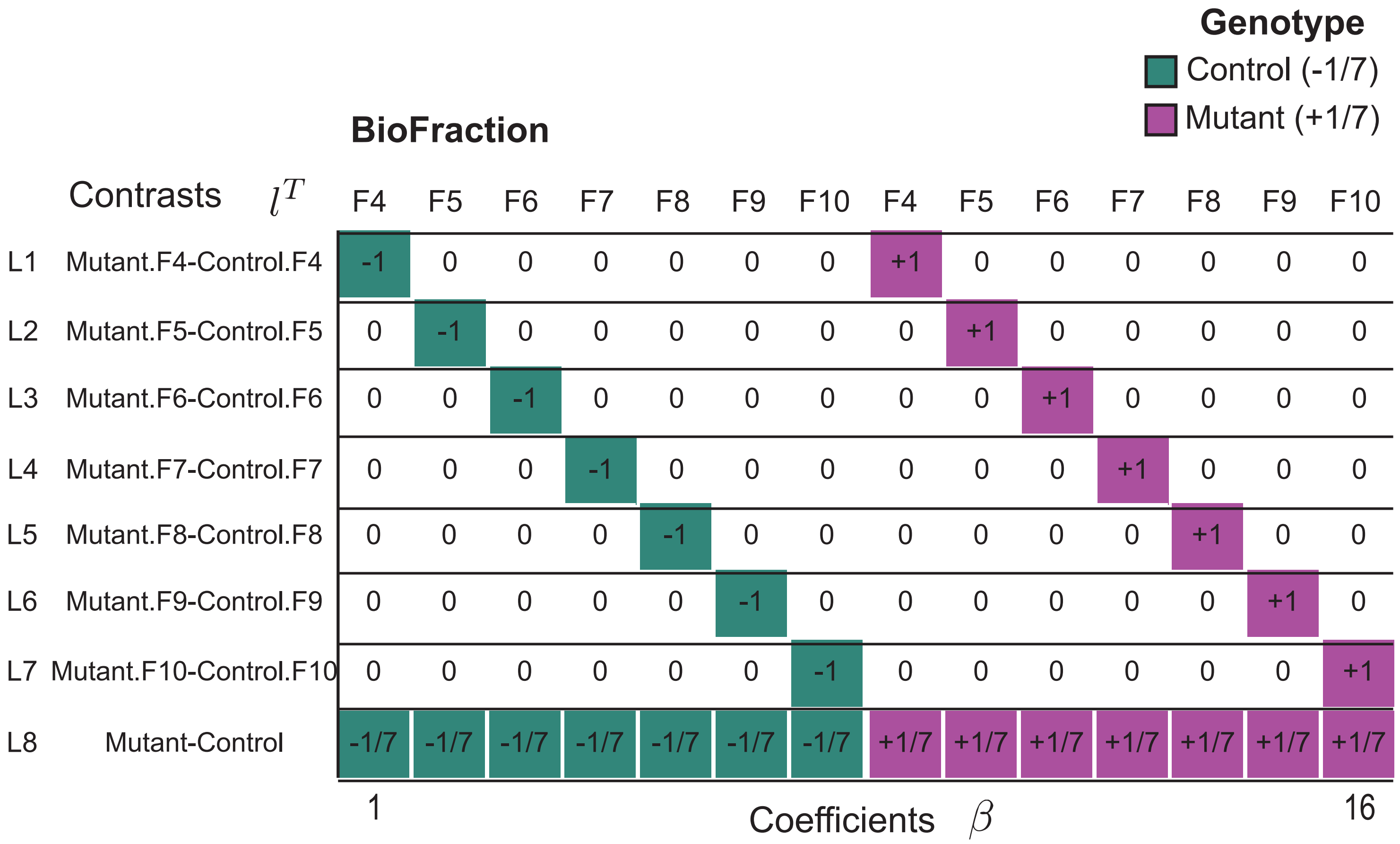


Figure 6. Data Normalization and PCA. A B

TMT Channel

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16
Mix1	WT-5K	WT-9K	WT-12K	WT-15K	WT-30K	WT-79K	WT-129K	QC1	MUT-5K	MUT-9K	MUT-12K	MUT-15K	MUT-30K	MUT-79K	MUT-129K	QC2
Mix2	WT-5K	WT-9K	WT-12K	WT-15K	WT-30K	WT-79K	WT-129K	QC1	MUT-5K	MUT-9K	MUT-12K	MUT-15K	MUT-30K	MUT-79K	MUT-129K	QC2
Mix3	WT-5K	WT-9K	WT-12K	WT-15K	WT-30K	WT-79K	WT-129K	QC1	MUT-5K	MUT-9K	MUT-12K	MUT-15K	MUT-30K	MUT-79K	MUT-129K	QC2

Figure 7. Experimental Design. We performed three 16-plex TMT experiments. Each TMT mixture is a concatenation of 16 labeled samples. In each experiment we analyzed seven subcellular `BioFractions` prepared from the brain of a single Control and 'Mutant' mouse. In all, we analyzed three `Subjects` from each Condition. Each `Mixture` includes two `Channels` dedicated to the analysis of a common quality control (QC) sample for normalization between MS runs.



Genotype

🟩 Control (-1/7)

🟪 Mutant (+1/7)

BioFraction

	Contrasts l^T	F4	F5	F6	F7	F8	F9	F10	F4	F5	F6	F7	F8	F9	F10
L1	Mutant.F4-Control.F4	-1	0	0	0	0	0	0	+1	0	0	0	0	0	0
L2	Mutant.F5-Control.F5	0	-1	0	0	0	0	0	0	+1	0	0	0	0	0
L3	Mutant.F6-Control.F6	0	0	-1	0	0	0	0	0	0	+1	0	0	0	0
L4	Mutant.F7-Control.F7	0	0	0	-1	0	0	0	0	0	0	+1	0	0	0
L5	Mutant.F8-Control.F8	0	0	0	0	-1	0	0	0	0	0	0	+1	0	0
L6	Mutant.F9-Control.F9	0	0	0	0	0	-1	0	0	0	0	0	0	+1	0
L7	Mutant.F10-Control.F10	0	0	0	0	0	0	-1	0	0	0	0	0	0	+1
L8	Mutant-Control	-1/7	-1/7	-1/7	-1/7	-1/7	-1/7	-1/7	+1/7	+1/7	+1/7	+1/7	+1/7	+1/7	+1/7

1 Coefficients β 16

Figure 8. Statistical Comparisons. We assessed two types of contrasts. Each row of the matrix specifies a contrast between positive and negative coefficients in the mixed-effects model fit to each protein. Contrasts1-7 are intra-BioFraction contrasts that specify the pairwise comparisons of Control and Mutant groups for a single fraction. In Contrast 8 we compare Mutant-Control and assess the overall difference of Control and Mutant conditions. Each contrast is a vector of sum 1.

$$Y_{mtcb} = \mu + Mixture_m + Condition_{cb} + \epsilon_{mcb}$$

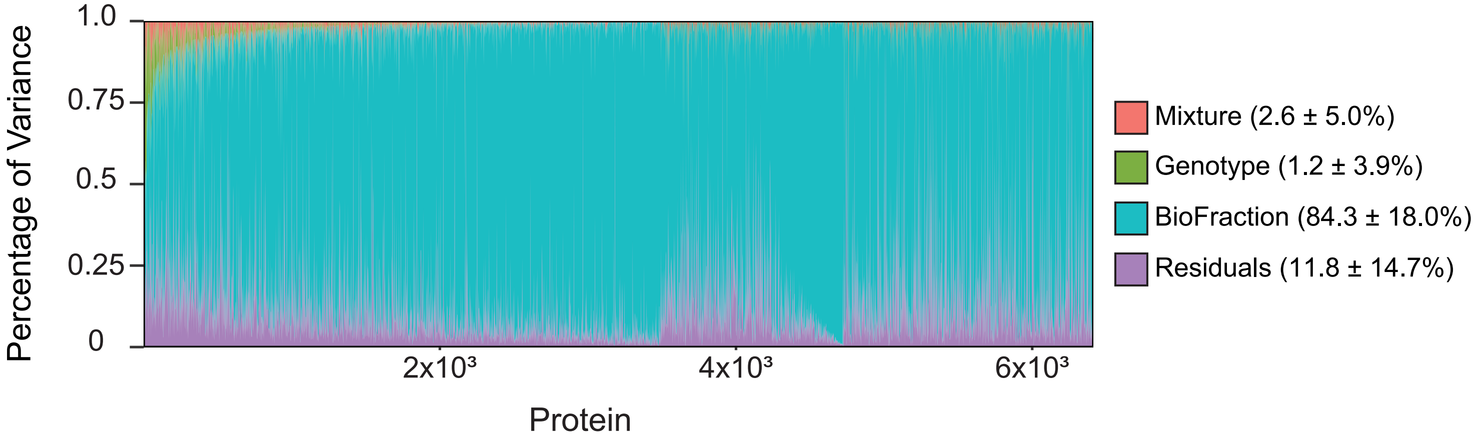


Figure 9. Analysis of Variance Components. The proportion of variance explained by Genotype, BioFraction, Mixture, and remaining residual error (subplot error) for all proteins. Note while the contribution of Mixture seems negligible, its average for all proteins is approximately twice the average percent variance explained by Genotype. BioFraction explains the majority of the variance for all proteins. Analysis done with `variancePartition::calcVarPart`.