Supplementary 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, Northeaster University, Boston, MA 02115, USA

Author correspondence:
jlc123@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^{P1019R} mouse brain. `edgeR` utilizes a negative binomial (NB) generalized linear model (GLM) framework originally developed for analysis of RNA-Seq data. Previously, we failed to fully consider the validity of `edgeR's` NB assumption for proteomics data. Here, we evaluate the goodness-of-fit of the NB GLM and find evidence of a lack-of-fit. Thus, we revise our statistical approach and reanalyze our data, making use of Huang *et al.*'s recently published R package `MSstatsTMT`. `MSstatsTMT` models the complex sources of variation in TMT mass spectrometry (MS) experiments using linear mixed-models (LMM). We extend the LMM framework used by `MSstatsTMT` to re-evaluate both protein- and module-level statistical comparisions in our **SWIP-TMT** proteomics dataset.

edgeR Lack-of-fit for TMT Proteomics

Our previous approach can be summarized as the 'Sum + IRS' method (Huang2020). Following protein summarization by summing its features, we performed Internal Reference Scaling (IRS) normalization (Plubell2017). We then applied `edgeR` 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.* who describe IRS normalization and the use of `edgeR` for statistical testing in TMT MS experiments (Plubell2017). We failed however, to consider the overall adequacy of the NB GLM model for our TMT proteomics data.

Statisitical inference in `edgeR` is performed for each gene or protein using a negative binomial framework. The data are assumed to be adequately described by a NB distribution parameterized by a dispersion parameter, ϕ. Practically, the dispersion parameter accounts for the observed mean-variance relationship 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 negative binomial count data. Based on this assumption, we justified the use of `edgeR`. Here, we reconsider the overall adequacy of the `edgeR` NB GLM model for TMT mass spectrometry data.

To evaluate the overall adequacy of the `edgeR` model, we plot the residual protein deviance statistics of all proteins against their theoretical, normal quantiles in a quantile-quantile (QQ) plot (***Figure 1***). The QQ plot addresses the question of how similar the observed data are to the theoretical NB distribution. A linear relationship between the observed and theoretical values is an indicator of goodness-of-fit. Deviation from this linear trend is evidence of a lack-of-fit.

Following protein summarization and normalization with `MSstatsTMT`, the data were fit with a NB GLM using `edgeR::glmFit`. ***Figure 1*** illustrates the divergence of the observed deviance statistics from the theoretical NB distribution for **SWIP-TMT** data fit with `edgeR's` NB GLM.

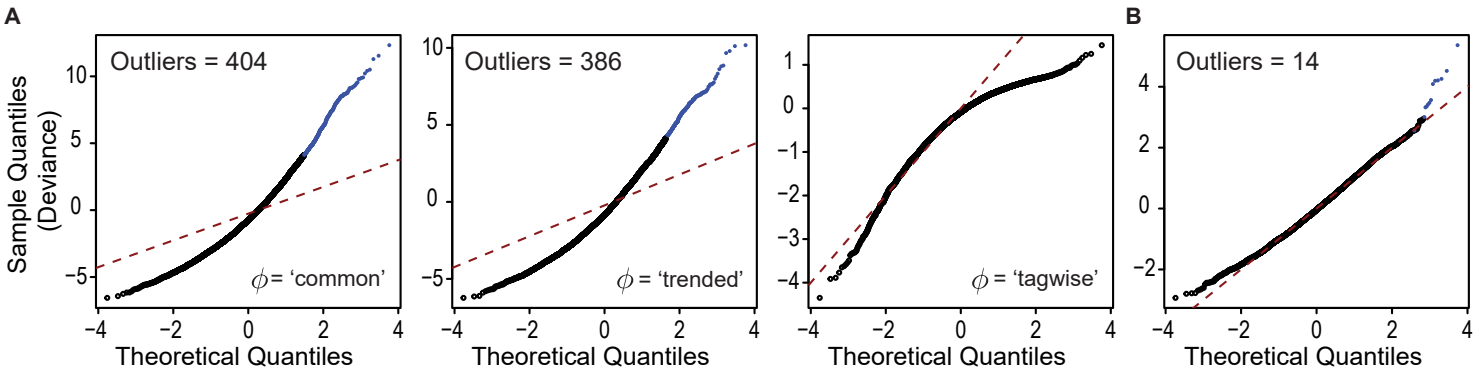

Figure 1. Goodness-of-fit of `edgeR` (A), and `MSstatsTMT` (B) statistical approaches. The overall adequacy of the linear models fit to the data were assessed by plotting the residual deviance for all proteins as a quantile-quantile plot (McCarthy *et al.*, (2012)). **(A)** For analysis with `edgeR`, 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 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 stattistics 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).

Statistical Inference with MSstatsTMT

Given our experimental design, `MSstatsTMT` fits an appropriate linear-mixed model expressing the major sources of variation in our experiment. The quantile-quantile plot in ***Figure 1*** indicates that the data are well described by `MSstatsTMT's` LMM, which does not depend upon the negative binomial assumption. These plots emphasize the overall lack-of-fit for proteomics data fit with the `edgeR` model.

The strength of LMMs lies in their flexibility. In mixed-models, 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 (Bates2015). As such, mixed-effect parameters account for the variation occurring among the lower levels of an upper level unit in the data (Bates2015). 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.* created `MSstatsTMT`, an R package for data normalization and hypothesis testing in multiplex TMT proteomics experiments. They outline a common vocabulary for describing the experimental design of a general TMT MS experiment.

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 mass spectrometry `Run` to quantify protein abundance. 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` or more `B` biological replicates or `Subjects`. Finally, a single TMT mixture may be repeatedly analyzed in `t = 1 ... T` technical replicate mass spectrometry runs.

The following equation is a LMM formula which describes protein abundance in an experiment composed of `M` mixtures, `T` technical replicates of mixture, `C` conditions, and `B` biological subjects.

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

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

The model's constraints (2) distinguish fixed- and mixed-effect components of variation in the response. `Mixture` is a mixed-effect and represents the variation between TMT mixtures. By definition mixed-effects are assumed to be independent and normally distributed (iid). `TechRep(Mixture)` represents random variation between replicates of a single MS `Run`. The term `Subject` cooresponds 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:

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

`MSstatsTMT` performs protein-wise comparisons between pairs of `Conditions` by comparing the estimates obtained from the fit LMM. We are interested in testing the null hypothesis:

$$H0 : l^T * \beta = 0. \tag{4}$$

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 the levels of `Condition`. A test statistic for such a two-way contrasts is given by Kutzenova *et al.,*(Kutzenova2017):

$$t = \frac{l^T \hat{\beta}}{\sqrt{l\sigma^2 \hat{V} l^T}} \tag{5}$$

We obtain the models estimates $\hat{\beta}$, error σ^2, and variance-covariance matrix $\hat{V}$ from the model fitted by restricted maximum likelihood. Given a contrast, l^T, the numerator of equation (5) is the fold change of a comparison. In the denominator, 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 (Satterthwaite1946, Kutzenova2017). 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 (Benjamini1995,Huang2020).

SWIP-TMT Experimental Design

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 2. 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.

In our experiment, the fixed-effect term `Condition` in equation 3 represents the fourteen combinations of `Genotype` and `BioFraction` obtained from subcellular fractionation of Control and SWIP^{P1019R} mouse brain. We refer to these as `BioFractions` to distinguish them from an MS `Fraction`.

Each 16-plex TMT mixture contains seven repeated measurements made from each biological subject (***Figure 2***). To account for this repeated measures design, we should include the random-effect term `Subject`. However, in our experimental design, `Mixture` is confounded with `Subject`. In each `Mixture` we analyzed all seven `BioFractions` from a single `Subject` (a Control or Mutant mouse). Thus we can choose to account for the effect of `Mixture` or `Subject`, but not both. We choose to account for variability of `Mixture` based on the assumption that the variance associated with this experimental batch effect is greater than the intra-Subject variance inherent in the repeated measures of each subject.

We omit the un-estimable terms `TechRep(Mixture)` and `Subject` from equation (1) and the reduced model is then:

$$Y_{mcbt} = \mu + Mixture_m + Condition_c + \epsilon_{mcb} \tag{6}$$

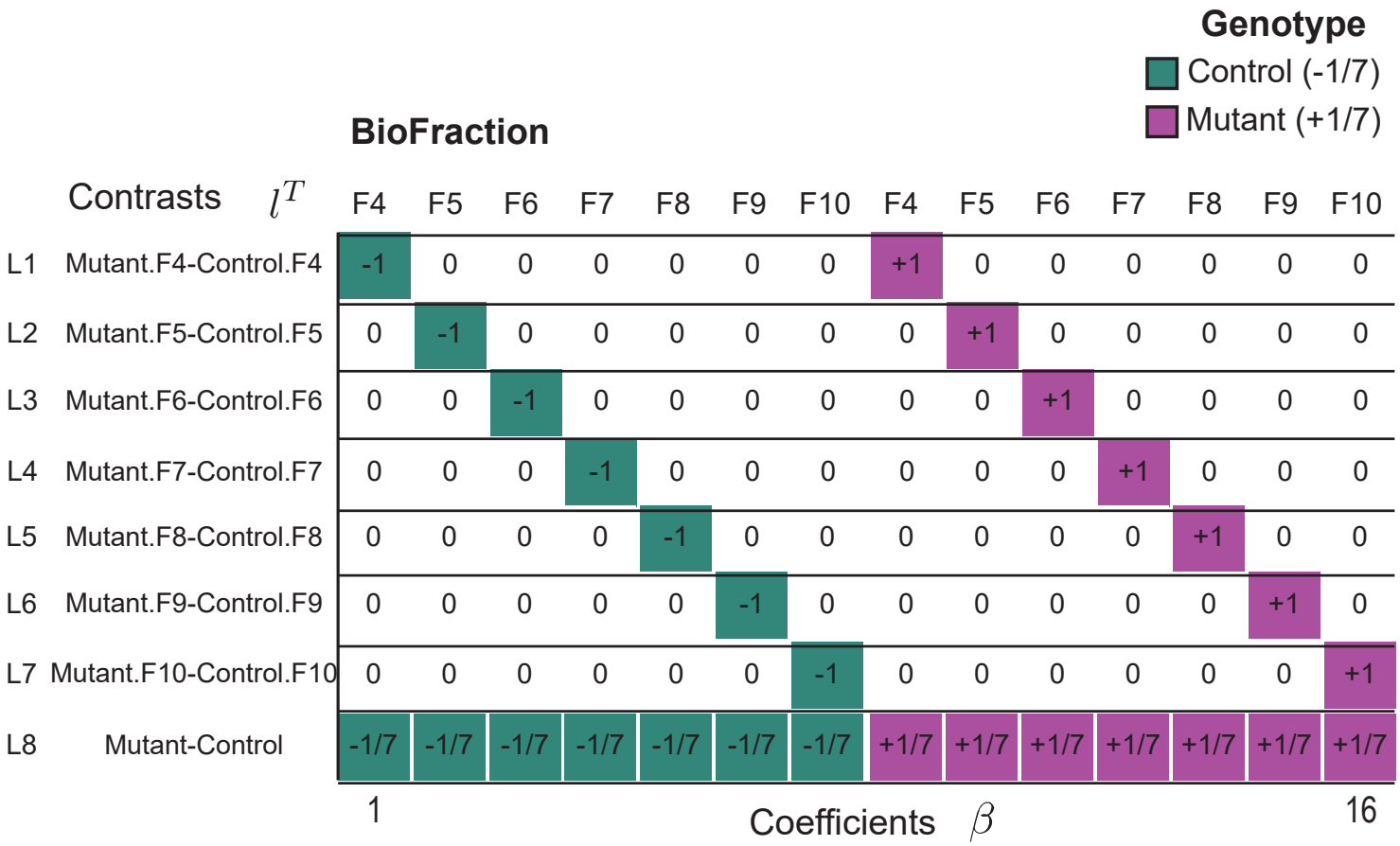

Figure 3. Statistical Comparisons. We assessed two types of contrasts. Each row of the matrix specifies a contrast between positive and negative coefficients in the mixed-effects model fit to each protein. Contrasts1-7 are intra-BioFraction contrasts that specify the pairwise comparisons of Control and Mutant groups for a single fraction. In Contrast 8 we compare Mutant-Control and asses the overall difference of Control and Mutant conditions. Each contrast is a vector of sum 1.

Protein-level comparisions

Following data preprocessing, summarization, and normalization, statistical inference by `MSstatsTMT` can be summarized in two steps:

- Fit each protein with an appropriate LMM based on the design.
- Given the fitted model, assess a contrast of interest.

Using `MSstatsTMT` we assesssed two types of protein comparisons:

- `intra-BioFraction` comparisons (7)
- `Mutant-Control` contrast (1)

`Intra-BioFraction` comparisons are the seven pairwise comparisons of Control and Mutant protein abundance for each subcellular `BioFraction`. We also assessed differential abundance for the overall `Mutant-Control` comparison. Each of these contrasts is represented by a vector, l^T_{1-8}, which specifies a comparison between coefficients in the LMM (6). ***Figure 3*** illustrates a matrix defining all eight unique comparisons.

`MSstatsTMT` attempts to automatically parse the experimental design and fit the appropriate LMM to each protein in the dataset. In order to understand and extend the function of `MSstatsTMT`, we extracted `MSstatsTMT's` core model-fitting and statistical testing steps and illustrate them here.

At the core of the model fitting-step is the R package `lme4` which implements mixed-effects models with its function `lmer`(Bates2015). The package `lmerTest` extends `lme4's` functionality and enables the computation of Sattertwaite degrees of freedom (Kutzenova2017).

As an example, we illustrate the analysis of WASHC4. First, we fit the model (6) to the normalized protein data from `MSstatsTMT`.

```
# load dependencies
library(dplyr)
library(lmerTest)

#library(SwipProteomics)
data(swip)
data(msstats_prot)

# formula to be fit to WASHC4, aka SWIP:
fx0 <- 'Abundance ~ 0 + Genotype:BioFraction + (1|Mixture)'

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

```
# examine the model's summary
summary(fm0, ddf = "Satterthwaite") %>% knitr::kable()

## Error in as.data.frame.default(x):  cannot coerce class 'c("summary.
"summary.merMod")' to a data.frame
```

The model's estimates (β) represent our best estimate of the mean protein abundance in the fourteen conditions of `Genotype:BioFraction`. To illustrate an `intra-BioFraction` comparison, we define a contrast comparing the `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) %>% knitr::kable()
```

Contrast	log2FC
GenotypeMutant:BioFractionF7-GenotypeControl:BioFractionF7	-1.689393

The function `lmerTestContrast` performs the statstical comparison given a fitted model and a convrast vector defining a comparison between the models coefficients. While the work done by this function is the same as the work done internally by `MSstatsTMT's groupComparisonsTMT` function, `lmerTestContrast` is more flexible. Provided the correct contrast, we easily assess the overall `Mutant-Control` comparison.

```
# use convenience function to contruct a contrast
contrast8 <- getContrast(fm0, "Mutant","Control"))

# assess the comparison
lmerTestContrast(fm0, contrast8) %>% knitr::kable()

## Error:  <text>:3:50:  unexpected ')'
## 2:  # use convenience function to contruct a contrast
## 3:  contrast8 <- getContrast(fm0, "Mutant","Control"))
##                                                      ^
```

LMM Goodness-of-fit

It is useful to consider the goodness-of-fit of our LMM. A straight forward measure of a LMM's quality is the Nakagawa coefficient of determination (Nakagawa2013,Nakagawa2017). Nakagawa's conditional R^2 is interpreted as the total variance explained by a LMM (R^2_{total}). The marginal R^2 is interpreted as the variance explained by the LMM's fixed-effects (R^2_{fixed}). We implment Nakagawa's coeffficient of determination using the `r.squaredGLMM` function taken from the `MuMin` package (WangMerkel2018).

```
# assess gof with Nakagawa coefficient of determination
r.squaredGLMM.merMod(fm0) %>% knitr::kable()
```

R2m	R2c
0.9353344	0.949433

The total variation explained, R^2_c, for the LMM fit to WASHC4 is `0.949`. The variance explained by fixed-effects, represents a large fraction of this total (R^2m=0.935). Only about 1.5% of the remaing variance is attributable to residuals and the mixed-effect `Mixture`.

Module-level analysis

We wish to extend the LMM framework developed by `MSstatsTMT` to perform inference at the level of protein groups or modules. That is, for module-level comparisons, we are interested in the overall affect of Genotype on a group of proteins. Where modules are groups of covarying proteins which represent biological niches defined by proteins that localized together in subcellular space.

Here we hypothesize that the proteins within a module, which are a subset of the overall proteome, are a part of a common group, a module, with a common mean effect. Proteins within a module are correlated observations which we model as a mixed-effect as we are primarily interested in making inference about the overall distribution of the responses for a module rather than among its sublevels. The following LMM includes the additional mixed effect term `Protein`, capturing variance among a module's constintuent proteins.

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

The term `Protein` quantifies the variance σ_P attributable to all proteins in a module. As a means of example, we demonstrate an ideal module, by fitting LMM (7) to the five WASH complex proteins. As before, we calculate the coefficient of determination for LMM's with the `r.squaredGLMM` function (WangMerkel2018).

```
# 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, msstats_prot %>% subset(Protein %in% washc_prots))

r.squaredGLMM.merMod(fm1) %>% knitr::kable()
```

R2m	R2c
0.7620866	0.8928053

Again, we consider the total variance explained as a measure of the model's overall quality. Our model explains 89.2% of the total variance among these five proteins. The fixed-effect term `Genotype:BioFraction` explains the majority of variance ($R_m^2 = 0.762$). The remaining 13.0% variance is attributable to a combination of mixed-effects `Mixture` and `Protein` as well as the residual variance.

We assess the overall `Mutant-Control` difference between responses of of 'Mutant' and Control groups as before.

The R package `variancePartition` enables us to calculate the percent variance explained by a LMM's parameters. To do so, it expects all terms to be mixed-effects. ***Figure 4***.

```
# load variancePartition
library(variancePartition)
```

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

calcVarPart(fit)

## BioFraction    Genotype     Mixture     Protein   Residuals
## 0.032960635 0.822069159 0.002843637 0.074146798 0.067979772
```

We can see that the majority of the variance explained by the LMM fit to the WASH complex is attributable to `Genotype`. The mixed-effect terms `Protein` and `Mixture` account for a small fraction of the overall variance explained by the model.

As our overall goal is to identify groups or modules of proteins that strongly covary together, our clustering approach should maximize the variance explained by a module's fixed-effect parameters (Genotype + BioFraction) while minimizing the variance among its individual proteins. An ideal module is a perfect summary of its protein constituents, $PVE_{Protein} = 0$. We use this idea of a module's quality to supervise our clustering approach.

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

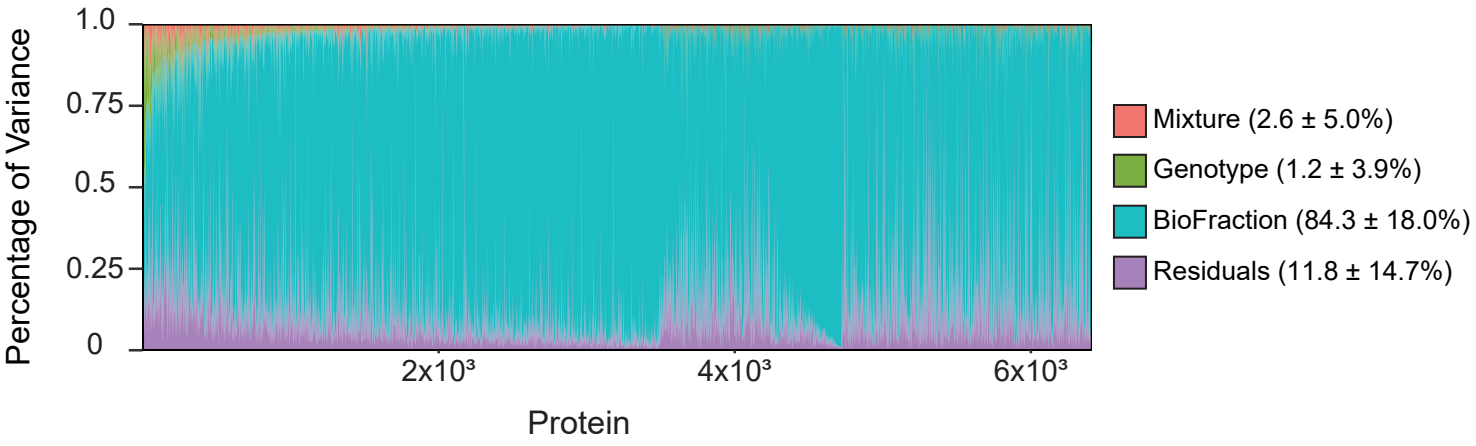

Figure 4. Analysis of Variance Components. The proportion of variance explained by Genotype, BioFraction, Mixture, and remaining residual error (subplot error) for all proteins. Note while the contribution of Mixture seems negligiable, its average for all proteins is approximately twice the average percent variance explained by Genotype. BioFraction explains the majority of the variance for all proteins. Analysis done with `variancePartition::calcVarPart`.

$$Quality_{Module} = \frac{PVE_{Genotype} + PVE_{BioFraction}}{PVE_{Protein}} \quad (8)$$

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 covariation 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. This step is essential 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 befinit 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 enhancment has the effect of making the network sparse and facilitates the identification of network structure (FIG).

(***Figure 5***)
(***Figure 6***)

(***Figure 7***)

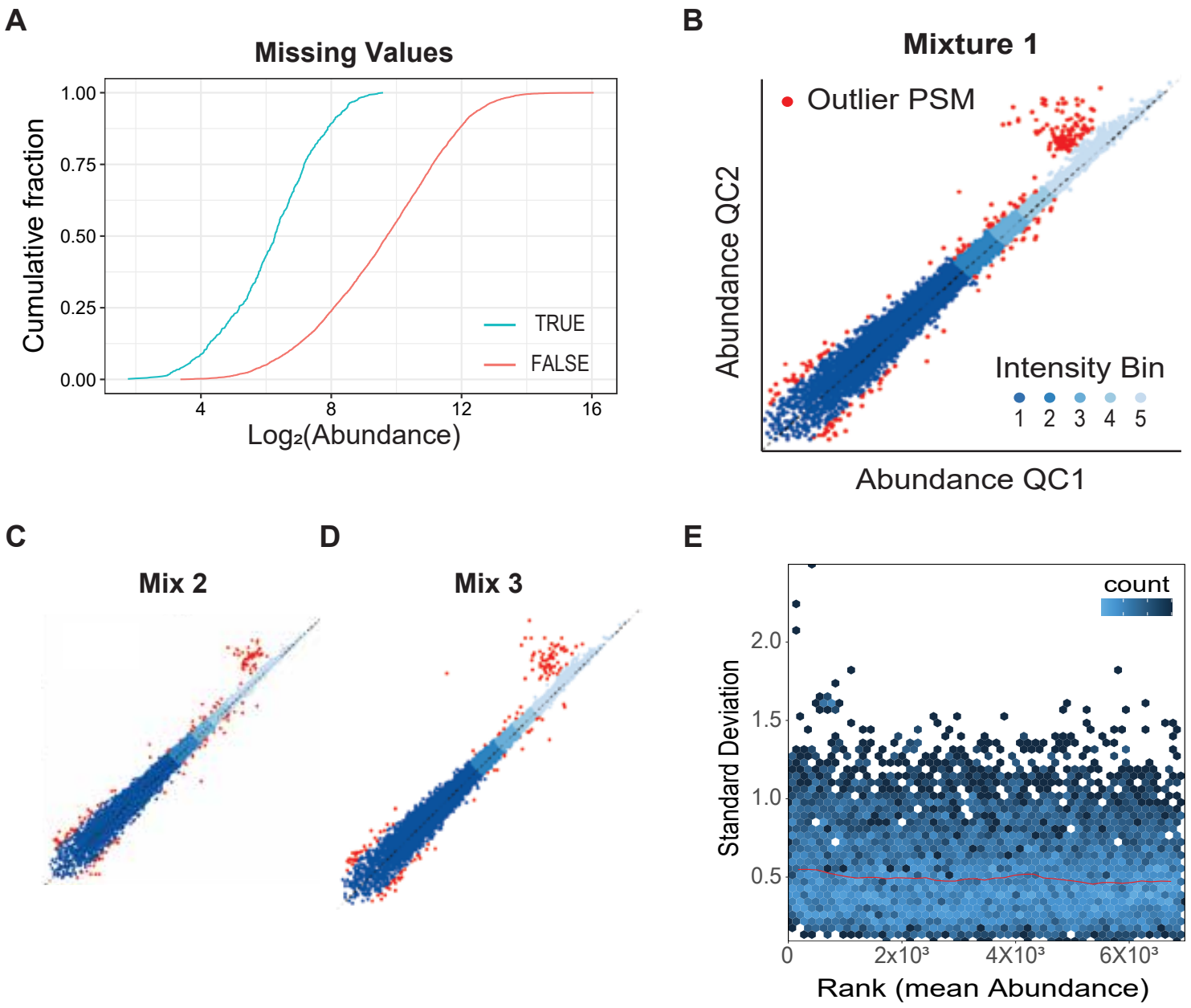

Figure 5. Missing value imputation and PSM outlier removal. A B C D

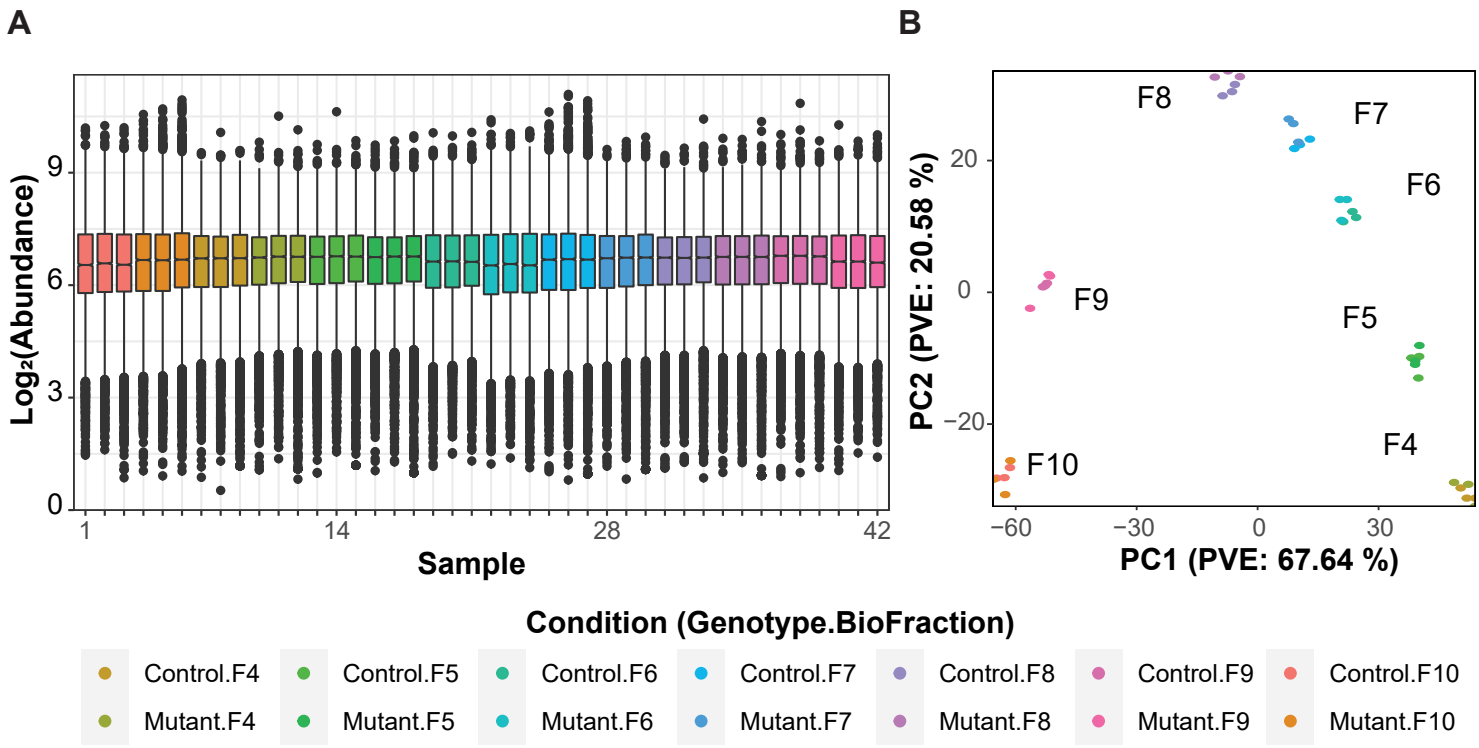

Figure 6. Data Normalization and PCA. A B

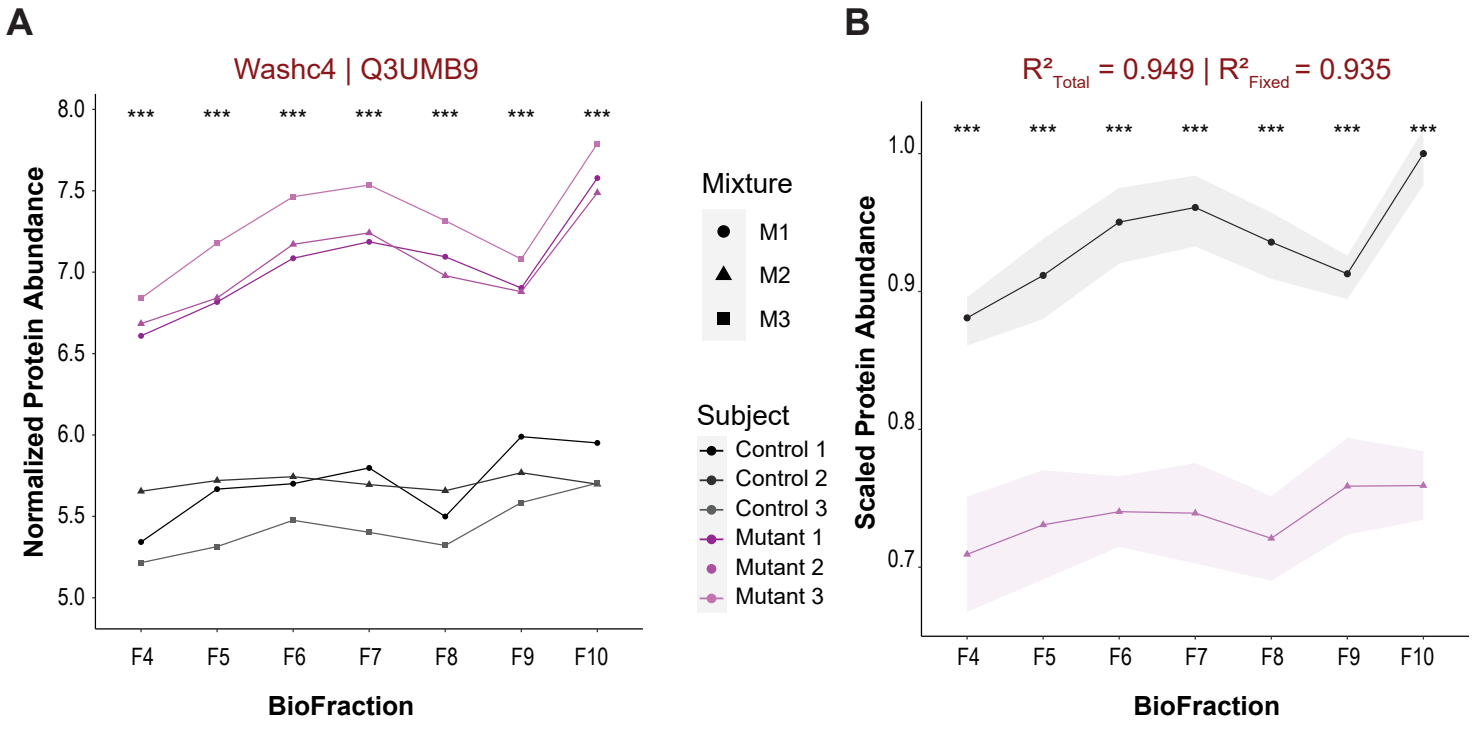

Figure 7. Data Normalization and PCA. A B