eLife Response

Tyler W. A. Bradshaw
Duke University, Department of Neurobiology
Durham, NC USA

October 29, 2020

1 Reanalysis of Proteomics Data

In our submitted manuscript, *"Genetic Disruption of WASHC4 Drives Endo-lysosomal Dysfunction and Cognitive-Movement Impairments in Mice and Humans"*, we described the analysis of two protein mass spectrometry experiments:

1. **WASH1 iBioID**
2. **WASHC4 spatial TMT-Proteomics**

At the center of the cogent critique of our manuscript was the questioned statistical validity of our previously described approach. Succinctly, the issue at question is whether or not the R package `edgeR` is an appropriate tool for analysis of protein mass spectrometry data.

Statistical inference in edgeR is built on a negative binomial (NB), generalized linear model (GLM) framework. Therefore, the data are assumed to be adequately described by a NB distribution parameterized by a dispersion parameter, ϕ. [1]

Previously we used a customized workflow [2] to preprocess and normalize the data prior to performing statistical testing using `edgeR's` flexible GLM framework. However, we failed to thoughougly consider the overall adequacy of the NB framework for mass spectrometry data. Here we reconsider its appropriatness for our TMT proteomics dataset.

[1] The dispersion parameter can take several forms. `edgeR` supports three dispersion models: 'common', 'trended', and 'tagwise'. However, when using `edgeR's` robust quasi-likelihood test methods, only global (i.e. 'common' or 'trended') dispersion metrics are appropriate (see `edgeR::glmQLFit's` documentation).

[2] The most important step in our normalization approach is IRS normalization. MS2 random sampling results in identification and quantification of proteins by different peptides in each MS experiment. To account for this source of variability, protein measurements are adjusted by a scaling factor such that the geometric mean of all internal reference standards are equal (Plubell et al., 2017). This is essential to account for the stochasticisity of peptide quantification in MS experiments. Phillip Wilmarth's GitHub offers an excellent exploration of IRS normalization.

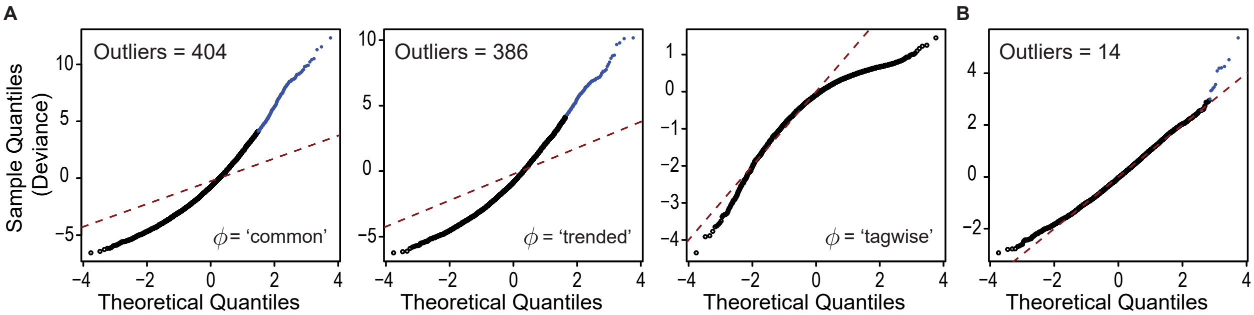


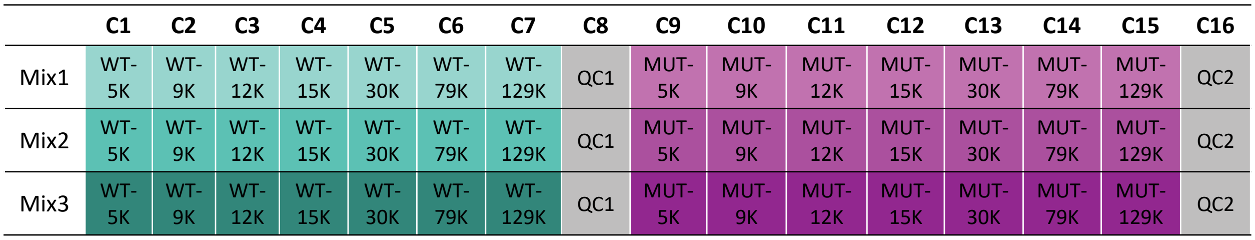
Figure 1: **Goodness-of-fit of `edgeR` (A), and `MSstats` (B) statistical approaches.** The overall adequacy of the linear models fit to the data were assessed by plotting the residual deviance for all proteins as a quantile-quantile plot (McCarthy *et al.*, (2012)). **(A)** The normalized protein data were fit with a NB GLM of the form: `Abundance` ∼ `Mixture + Condition`. Where `Mixure` is a blocking factor that accounts for sources of variablity between experiments. Protein-wise deviance statistics were transformed to normality and plotted aganis theoretical normal quantiles using `edgeR::gof`. **(B)** The normalized protein data were fit with a linear mixed-effects model (LMM) of the form: `Abundance` ∼ `0 + Condition + (1|Mixture)`. Where `Mixture` indicates the random 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).

We evaluated the overall adequacy of the `edgeR` model by plotting the residual deviance of all proteins against their theoretical, normal quantiles in a quantile-quantile plot. **Figure 1** illustrates the overall lack of fit for the three disperion models fit by `edgeR`. As an alternative to `edgeR` we considered `MSstatsTMT`, an extension of MSstats for analysis of TMT proteomics experiments.

`MSstatsTMT` utilizes a linear mixed-model framework. The strength of linear mixed models (LMMs) is in their ability to account for complex sources of variation in an experimental design.

> In a mixed model one or more covariates are a categorical variable representing experimental or observational "units" in the data set. [...] If the set of possible levels of the covariate is fixed and reproducible we model the covariate using fixed-effects parameters. If the levels that we observed represent a random sample from the set of all possible levels we incorporate random effects in the model.

A TMT proteomics experiment consists of `m = 1...M` concatensions of isobaric-TMT labeled samples or `Mixtures`. Each TMT channel is dedicated to the analysis of `c = 1...C` individual biological or treatment `Conditions` prepared from one `b = 1...B` biological replicates or `Subjects`. A single mixture may be profiled in `t = 1...T` technical replicate mass spectrometry runs.
We prepared 7 subcellular fractions (BioFraction) from 2 Conditions: Control and SWIP$^{\mathrm{P1019R}}$ Mutant mice. There were 6 Subjects, three bioreplicate Control and SWIP$^{\mathrm{P1019R}}$ Mutant mice.



	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 utilized 16-plex TMT tags to label samples prepared from 6 mice.

In an experiment such as ours with multiple mixtures and biological replicates, but no technical replication of mixture (T = 1) `MSstatsTMT` fits a linear mixed model of the following form to each protein:
Where Mixture is a mixed-effect and quantifies variation between TMT mixtures. Condition is a fixed effect (mean = 0) and in our experiment represents the interaction of terms `Genotype` and `BioFraction`. ϵ is a random effect representing both biological and technical variation, quantifying any remaining error.

$$Y_{mcbt} = \mu + Mixture_m + Condition_c + \epsilon_{mcbt} \tag{1}$$

Where Mixture represents the random-effect of mixture and Condition is a fixed-effect and in our experiment is interaction of Genotype and BioFraction–the 14 combinations of 7 BioFractions fractions from Control and Mutant mice. ϵ_{mcbt} is the residual error (σ^2).

In our experimental design, we made measurments from seven BioFractions from each subject. Thus, we should include the term Subject, representing the 6 individual mice or subjects analyzed in our experiment.

$$Y_{mcbt} = \mu + Mixture_m + Condition_c + Subject_b + \epsilon_{mcbt} \tag{2}$$

However, in our design `Mixture` is confounded with the term `Subject` – in each mixture we analyzed all `BioFractions` from a single Control and Mutant mouse. Thus we can choose to account for the effect of `Mixture` or `Subject`, but not both. Assuming Mixture contributes greater to the variance, we drop the term Subject, and the reduced model is equivalent to 1.

Model based testing of differential abundance between pairs of conditions is assessed through contrast of conditioned means estimated by fitting the parameters of the model by REML to obtain $\hat{\beta}$, σ^2 and $\hat{V}$.
The degrees of freedom are determined by the Satterthwaite approximation[REF], and the T-statistic for the contrast is taken to be (lmerTest ref):

$$t = \frac{l^T * \hat{\beta}}{sqrt(l * \sigma^2 * \hat{V} * l^T)} \tag{3}$$

σ^2 is the error from **Equation** 1. l^T is a vector specifying a contrast between positive and negative coefficients in the model.
Together, the denominator $\sqrt(l * \sigma^2 * \hat{V} * l^T)$ is the standard error of the contrast.

```
library(dplyr)

## load SwipProteomics data
data(swip)
data(gene_map)
data(msstats_prot)
data(alt_contrast)
data(msstats_contrasts)

## formula to be fit:
fx0 <- formula("Abundance ~ 0 + Condition + (1|Mixture)")

# fit the model
idx <- msstats_prot$Protein == swip
fm <- lmerTest::lmer(fx0, msstats_prot[idx,])

# calculate model statistics
model_summary <- summary(fm,ddf="Satterthwaite")

df <- model_summary$coefficients
df %>% as.data.table(keep.rownames="Coefficient") %>% knitr::kable()

## Error in as.data.table(., keep.rownames = "Coefficient"):  could not find
function "as.data.table"

# evaluate goodness-of-fit
r2_nakagawa <- r.squaredGLMM.merMod(fm)
knitr::kable(rbind(c("marginal/fixef", "conditional/total"), r2_nakagawa))
```

R2m	R2c
marginal/fixef	conditional/total
0.935142291105874	0.94934629533883

```
contrast <- msstats_contrasts[1,]
lmerTestContrast(fm,contrast) %>% knitr::kable()
```

Contrast	log2FC	percentControl	Pvalue	Tstatistic	SE	DF	isSing
Mutant.F4-Control.F4	-0.9075446	0.5330916	2.5e-06	-5.986412	0.1516008	25.99986	FALS