An Introduction Mixture Model Normalization with the *metabomxtr* Package

Michael Nodzenski, Anna C. Reisetter, Denise M. Scholtens

October 17, 2016

1 Introduction

Controlling technical variability in metabolite abundance, or normalization, is a critical step in the analysis and interpretation of non-targeted gas-chromatography/mass-spectrometry (GC/MS) data. In large scale metabolomics studies requiring sample processing in many analytic batches, technical artifacts due to batch and run-order within batch are common. In these cases, repeated assays of a set of control samples may be used to estimate and account for these artifacts. The *metabomxtr* package implements a mixture model normalization approach via the function *mixnorm* for studies implementing this quality control measure. Based on control sample variability, *mixnorm* allows for per-metabolite modeling of both batch and run-order effects, while allowing for bach specific thresholds of metabolite detectability.

2 Sample Mixture Model Normalization

The following commands demonstrate typical usage of *mixnorm*. First, load the package.

```
>   library(metabomxtr)
```

Next, load euMetabData, a sample data frame containing metabolite data for a total of 40 mother-baby pairs of Northern European ancestry. A total of 3 blood samples are included for each pair: mother fasting, mother 1-hour, and newborn cord blood. Mother samples were obtained during an oral glucose tolerance test (OGTT) at 28 weeks gestation, and baby samples were collected at birth. Sample types are indicated by row names, with 'mf' and 'm1' indicating maternal fasting and 1-hour samples, respectively, and 'bc' indicating baby samples. Note that while euMetabData is a data frame, *mixnorm* also accomodates metabolite data in matrix and ExpressionSet objects.

```
>  data(euMetabData)
>  class(euMetabData)

[1] "data.frame"

>  dim(euMetabData)

[1] 120   6

>  head(euMetabData)

        batch pheno betahydroxybutyrate pyruvic_acid malonic_acid aspartic_acid
MBP1_mf     1   MOM            20.14544     18.47593     15.52949      17.27488
MBP1_m1     1   MOM            19.30312     18.55794     16.89087      14.42220
MBP1_bc     1  BABY            22.83122     17.79843     14.77859            NA
MBP2_mf     1   MOM            20.55216     17.46991           NA            NA
MBP2_m1     1   MOM            19.76286     18.21836     16.13184            NA
MBP2_bc     1  BABY            21.62902     16.05125     14.58549            NA
```

Also load euMetabCData, a data frame containing GC/MS data from separate mom and baby control pools. Control pool aliquots were run at the beginning, middle and end of each batch with placement indicated by -1, -2 and -3 appended to the sample name, respectively.

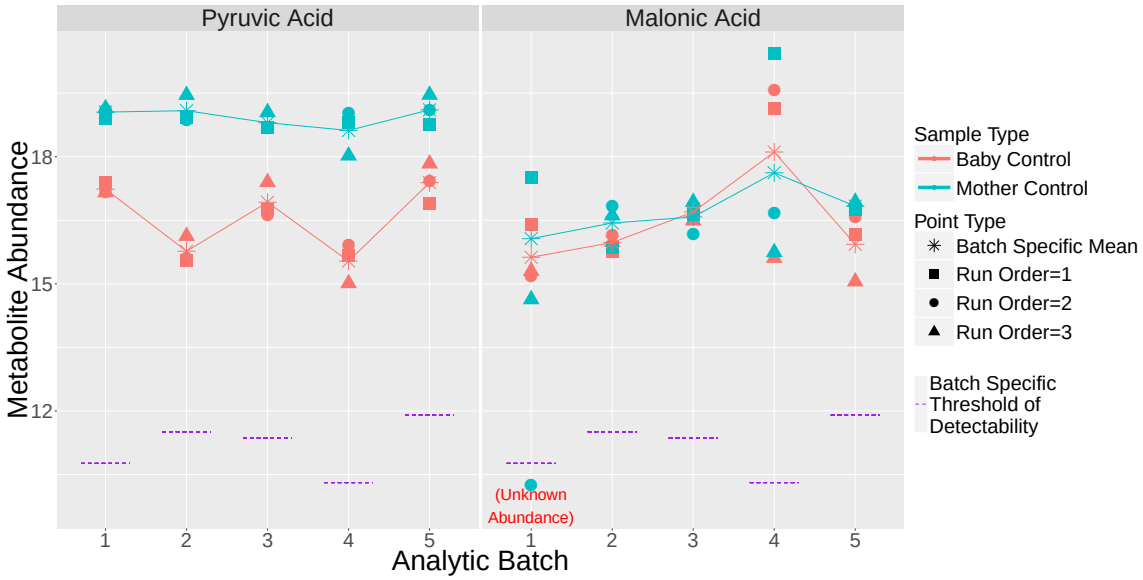
```
> data(euMetabCData)
> class(euMetabCData)

[1] "data.frame"

> dim(euMetabCData)

[1] 30  6

> head(euMetabCData)
```

	batch	pheno	betahydroxybutyrate	pyruvic_acid	malonic_acid	aspartic_acid
B-01-1	1	BABY	23.03775	17.38712	16.40161	NA
B-01-2	1	BABY	22.96725	17.16013	15.17785	NA
B-01-3	1	BABY	23.02668	17.15702	15.30278	NA
B-02-1	2	BABY	22.60945	15.55872	15.76774	NA
B-02-2	2	BABY	22.94031	15.62062	16.16459	12.98353
B-02-3	2	BABY	23.25504	16.12824	15.98942	NA

Pyruvic acid and malonic acid are included in both example data sets. We'll assume they are of analytical interest, and a define a character vector of the corresponding column names.

```
> ynames<-c('pyruvic_acid','malonic_acid')
```

Now we'll plot metabolite abundances from the control data set. In the absence of technical variability, we would expect to see constant mean abundance across batches for each metabolite. Also indicated in the plots are batch specific thresholds of metabolite detectability, based on experimental evidence not available here.



Both mother and baby control samples show considerable variability within and across batches, including one instance where abundance fell below the detectable threshold. To account for these techinical artifacts, we will use mixture model normalization implemented in the function *mixnorm*. This function takes as required arguments a character vector of target metabolite column names, the name of the variable corresponding to analytic batch in the input data objects, a data object (data frame, matrix, or ExpressionSet) with quality control data, and a data object with experimental data. In the example data sets, the variable corresponding to analytic batch is 'batch', the target metabolite columns are 'pyruvic_acid' and 'malonic_acid' (specified previously), the control data set is euMetabCData, and the experimental data set is euMetabData. By default, *mixnorm* implements a mixture model with batch as the only covariate. For this analysis, we also want to account for sample phenotype (mother vs. baby), and can do this by specifying a mixture model formula including both batch and phenotype. Note that mixnorm will not run if mixture model covariates are missing values. Additionally, we will specify the experimentally determined thresholds of metabolite detectability in optional argument batchTvals. If not specified, the default detectable batch threshold is set to the minimum observed metabolite abundance for that batch, across all metabolites of analytic interest.

```
> #execute normalization
> euMetabNorm <- mixnorm(ynames, batch="batch", mxtrModel=~pheno+batch|pheno+batch,
+                                batchTvals=c(10.76,11.51,11.36,10.31,11.90), cData=euMetabCData,
+                                data=euMetabData)
> #this produces warnings about NaNs produced, but this is the expected behavior of the function
```

The output of *mixnorm* is a list of three data frames. The first, normParamsZ, contains parameter estimates for the variables included in the mixture model for each metabolite specified. All estimates except for the intercept are subtracted from the raw metabolite values to produce the normalized data.

```
> euMetabNorm$normParamsZ
```

	zInt	z_phenoMOM	z_batch2	z_batch3	z_batch4	z_batch5
pyruvic_acid	16.96261	2.3637846	-0.7162706	-0.2801062	-1.066426	0.1029639
malonic_acid	15.71043	0.2309038	0.3788837	0.8101881	2.040327	0.5576339

The second element of the output list, ctlNorm, contains normalized values for the control samples.

```
> head(euMetabNorm$ctlNorm)
```

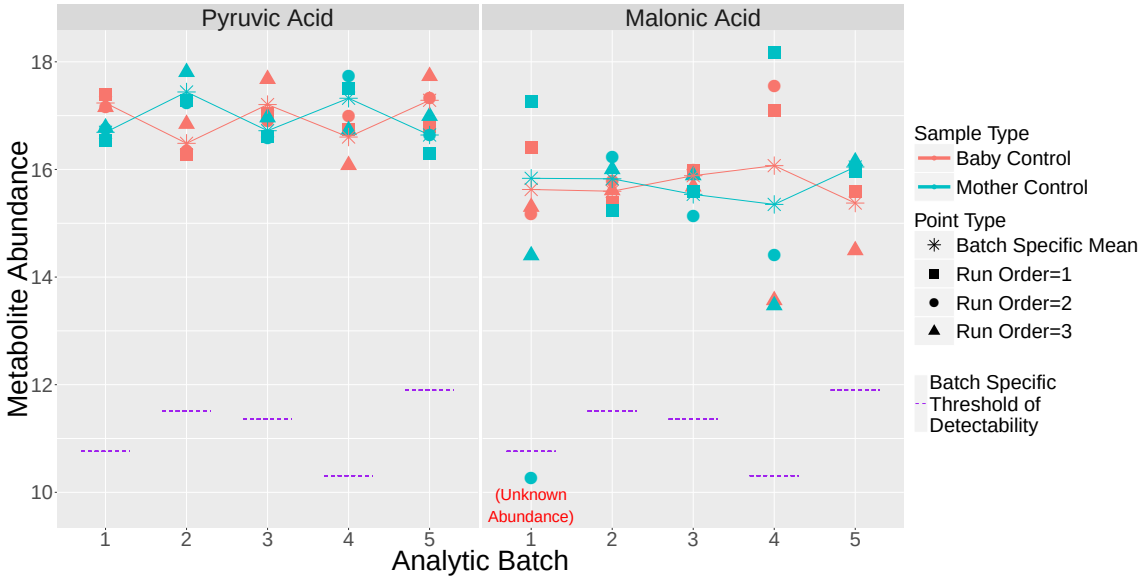
	pyruvic_acid	malonic_acid
B-01-1	17.38712	16.40161
B-01-2	17.16013	15.17785
B-01-3	17.15702	15.30278
B-02-1	16.27499	15.38886
B-02-2	16.33689	15.78571
B-02-3	16.84451	15.61053

The third element of the output list, obsNorm, contains normalized values for the experimental samples. Note that when metabolite abundance falls below the detectable threshold, indicated by missing metabolite values, values will remain missing in the normalized data set.

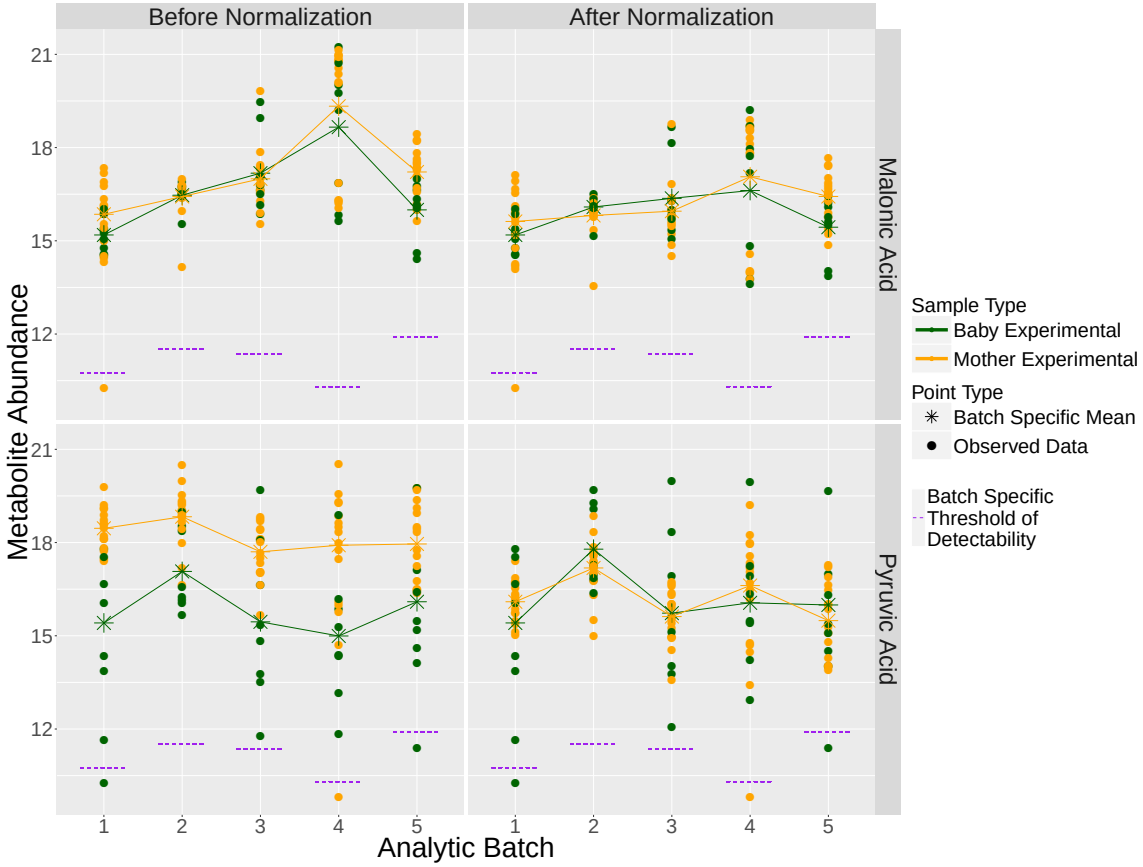
```
> head(euMetabNorm$obsNorm)
```

	pyruvic_acid	malonic_acid
MBP1_mf	16.11214	15.29859
MBP1_m1	16.19416	16.65997
MBP1_bc	17.79843	14.77859
MBP2_mf	15.10613	NA
MBP2_m1	15.85458	15.90093
MBP2_bc	16.05125	14.58549

After normalization, mean abundance values are much more stable across batches in the control samples:



In the experimental data, mean metabolite values are more variable, even after normalization. This is expected, as characteristics of biological interest are not expected to be uniform across batches, and normalization aims to preserve this true biological variability.



By default, *mixnorm* subtracts the effects of all variables included in the mixture model from the raw data to produce the normalized data. However, in certain instances, it may be desirable to include covariates in the mixture model to accurately estimate batch effects, but not actually remove the effects of those covariates. For instance, in the plots above, mother samples tended to have higher levels of pyruvic acid than baby samples across batches. We can account for sample type (mom vs. baby) in estimating batch effects while preserving metabolite variability based on sample type by specifying the name of the covariate column (or a character vector of names) to optional argument removeCorrection as follows:

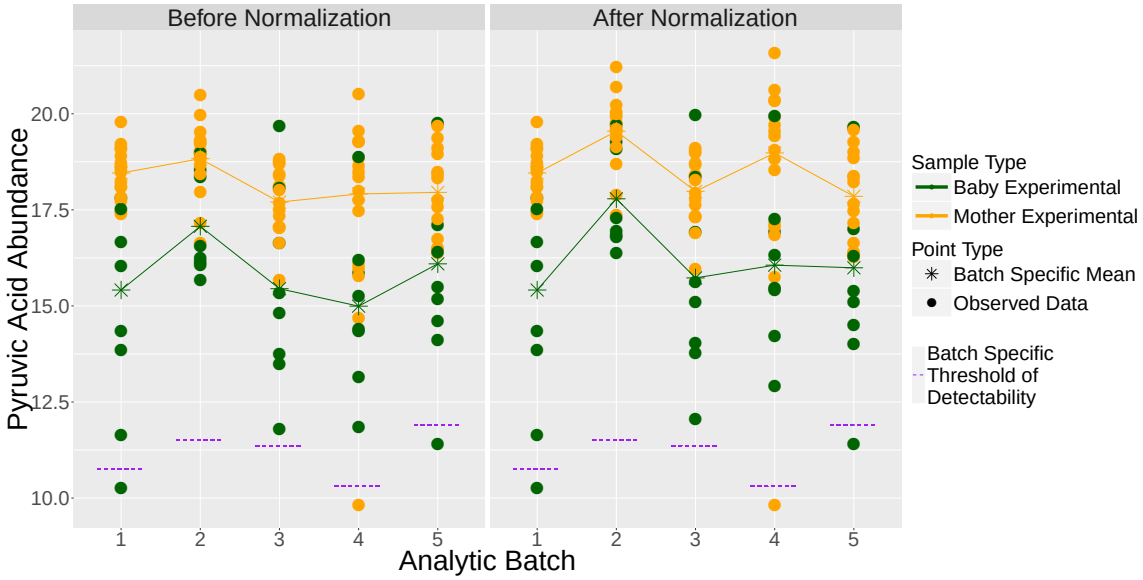
```
> euMetabNormRC <- mixnorm("pyruvic_acid", batch="batch", mxtrModel="pheno+batch|pheno+batch,
+                                  batchTvals=c(10.76,11.51,11.36,10.31,11.90), cData=euMetabCData,
+                                  removeCorrection="pheno",data=euMetabData)
```

The parameter estimates in normParamsZ will be identical to those had removeCorrection not been specified:

```
> head(euMetabNormRC$normParamsZ)
```

```
                 zInt z_phenoMOM   z_batch2   z_batch3  z_batch4  z_batch5
pyruvic_acid 16.96261   2.363785 -0.7162706 -0.2801062 -1.066426 0.1029639
```

However, the normalized data will not include a location shift for sample type. As seen below, the differences in pyruvic acid abundance between mother and baby samples are preserved.



3 Session Information

- R version 3.3.1 (2016-06-21), `x86_64-w64-mingw32`
- Locale: `LC_COLLATE=C`, `LC_CTYPE=English_United States.1252`, `LC_MONETARY=English_United States.1252`, `LC_NUMERIC=C`, `LC_TIME=English_United States.1252`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, utils
- Other packages: Biobase 2.34.0, BiocGenerics 0.20.0, ggplot2 2.1.0, metabomxtr 1.8.0, plyr 1.8.4, reshape2 1.4.1, xtable 1.8-2
- Loaded via a namespace (and not attached): BB 2014.10-1, Formula 1.2-1, MASS 7.3-45, Matrix 1.2-7.1, Rcgmin 2013-2.21, Rcpp 0.12.7, Rvmmin 2013-11.12, colorspace 1.2-7, dfoptim 2016.7-1, digest 0.6.10, grid 3.3.1, gtable 0.2.0, labeling 0.3, lattice 0.20-34, magrittr 1.5, minqa 1.2.4, multtest 2.30.0, munsell 0.4.3, numDeriv 2016.8-1, optextras 2016-8.8, optimx 2013.8.7, quadprog 1.5-5, scales 0.4.0, setRNG 2013.9-1, splines 3.3.1, stats4 3.3.1, stringi 1.1.2, stringr 1.1.0, survival 2.39-5, svUnit 0.7-12, tools 3.3.1, ucminf 1.1-4

4 References

Nodzenski M, Muehlbauer MJ, Bain JR, Reisetter AC, Lowe WL Jr, Scholtens DM. Metabomxtr: an R package for mixture-model analysis of non-targeted metabolomics data. Bioinformatics. 2014 Nov 15;30(22):3287-8.

Moulton LH, Halsey NA. A mixture model with detection limits for regression analyses of antibody response to vaccine. Biometrics. 1995 Dec;51(4):1570-8.