Supplementary File 1

1 Benchmarking log-ratio transformation methods for compositional differential analysis of metabolic pathways

We wanted to determine which log-ratio transformation method is best for the Compositional Data Analysis of our metabolic pathway dataset.

Methods

We evaluated two groups of log-ratio transformation methods, centered-log ratio (clr) transformation-like methods and additive-log ratio (alr) transformation methods. All of these methods were tested using the ALDEx2 R package (1). Both clr and alr transformation involve setting all measurements of a composition (i.e. each sample vector *j* containing relative measurements of, for example, microbiome metabolic pathways count) relative to (i.e. divided by) a reference and taking the logarithm of this ratio. The reference of alr is a single feature (i.e. index *D*, with *D* being the total number of features) whereas the reference of clr is the geometric mean, denoted as g(x), of the total components x*j* (of a composition that sum to a unity). An additional step to normalize by library size is not required for alr and clr transformation because both are technically equivalent to a normalization (2).

$$alr(x_j) = \left[ln\frac{x_{1j}}{x_{Dj}}, \ldots, ln\frac{x_{D-1j}}{x_{Dj}} \right] \quad (1)$$

$$clr(x_j) = \left[ln\frac{x_{1j}}{g(x_j)}, \ldots, ln\frac{x_{Dj}}{g(x_j)} \right] \quad (2)$$

Similar to clr, other transformations exist that use instead use the geometric mean of a *subset* of the composition. Taking a geometric mean of a subset may be attractive to statisticians if, for example, want to exclude unreliable singletons from their analyses. The log-ratio transformation methods we benchmarked are listed in Supp. Table 7.

Supplementary Table 7: List of log-ratio transformation methods assessed in the benchmarking simulation

Name	**Description and reference**	**ALDEx2 "denom" parameter**
Feature name	Additive-log ratio (alr) transformation. Reference: One of the features that is ubiquitous (have an abundance of at least 1 raw count among all samples) and among the features with a variance within the	A single row (feature) index

	first bottom quartile (< 25th percentile) for the variance of all clr transformed features.	
All	Centered-log ratio (clr) transformation. Reference: geometric mean of all components	all
iqlr	Subset of clr: inter-quartile log-ratio (iqlr). Reference: geometric mean of features with a variance between the 1st and 3rd quartile for the variance of all clr transformed features.	iqlr
lvha	Subset of clr: low variance and high relative abundance log ratio (lvha). Reference: geometric mean of features with a variance in the bottom quartile for the variance of all clr transformed features and in the top quartile for relative abundance for each sample across the entire dataset.	lvha
clr_var_below_25p	Subset of clr. Reference: geometric mean of features with a variance in the bottom quartile for the variance of all clr transformed features.	Vector of selected feature indices
coverage_above_25p.clr_var_below_25p	Subset of clr: low variance and sample coverage above 25%. Reference: geometric mean of features with a variance in the bottom quartile for the variance of all clr transformed features and feature present in more than 25 percent of samples across the entire dataset.	Vector of selected feature indices
coverage_above_50p.clr_var_below_25p	Subset of clr: low variance and sample coverage above 50%. Reference: geometric mean of features with a variance in the bottom quartile for the variance of all clr transformed features and feature present in more than 50 percent of	Vector of selected feature indices

	unique species across the entire dataset.	

We simulated 100 unique datasets based on our metabolic pathway abundance data, which consist of 20 MS cases, 20 controls, and 427 features (i.e. metabolic pathways). Each dataset contains two groups: group A consists of randomly selected 10 cases and 10 controls and group B consists of the other 10 cases and the other 10 controls. 20 random features are spiked in group B to have a mean effect size of 2 times relative to group A in simulation 1 and a mean effect size of 1.5 times in simulation 2, all while maintaining the original pairwise ratios of the abundances of features within group B. Features were not spiked if they were absent in more than 50% of the samples (sparsity) in each group or if the feature was among the clr_var_below_25p (described in Supp. Table X). Box 1 demonstrates how selected features were spiked to have a mean effect size of 2 on group B relative to group A using a demo dataset. Features present in less than 10% of samples in each group were excluded in the simulations.

Box 1: demonstrating the feature spike method for benchmarking log-ratio transformations using R programming.

```
Example demo dataset with three features, six samples split evenly into two
group, groups A (samples A1-3) and group B (samples B1-3). Spike feature X such
that feature X will have 2 times the mean relative abundance in group B
relative to group A (mean effect size of 2).

> library(dplyr)
> set.seed(1234)

Create a table with 3 rows (features) and 6 columns (samples) by randomly
selecting counts from a Poisson distribution.

> table <- rpois(n=3*6, lambda=8) %>% matrix(nrow = 3,ncol = 6)
> lib <- colSums(table) # save library sizes by summing columns
> row.names(table) <- c("feature_X", "feature_Y", "feature_Z") # label feature
names
```

	A1	A2	A3	B1	B2	B3
Feature X	5	9	2	8	6	11
Feature Y	9	11	6	9	12	6
Feature Z	9	9	9	8	6	6
Sample sum	23	29	17	25	24	23

```
Step 1: for each sample convert raw counts to proportions (normalize by library
size)

Divide each column (feature) by the total sum of that column
> table_norm <- sweep(table, MARGIN=2, lib, `/`)
```

	A1	A2	A3	B1	B2	B3

Feature X	0.22	0.31	0.12	0.32	0.25	0.48
Feature Y	0.39	0.38	0.35	0.36	0.50	0.26
Feature Z	0.39	0.31	0.53	0.32	0.25	0.26

Step 2: calculate the amount z needed to add to group B to have an effect size of 2

Original effect size of group A vs. group B for Feature X is 1.624

```
> Feature_x_A.norm <- table_norm["feature_X",1:3] #Feature X abundance in group A
> Feature_x_B.norm <- table_norm["feature_X",4:6] #Feature X abundance in group B
> mean(Feature_x_B.norm) / mean(Feature_x_A.norm) # calculate effect size
1.624246

> n_B = 3 # set number of components (features) in group B.

Set effect size (ES) to 2.
> ES = 2 # Effect size

Calculate z such that mean(Feature_x_B.norm) + z  /  mean(Feature_x_A.norm) = 2
> z = (ES*mean(Feature_x_A.norm)*n_B) - sum(Feature_x_B.norm) # Solve for z.
> z
0.2425055
```

Need to add a total of 0. 0.2425 to group B in feature X to have an effect size of 2 for feature X.

Step 3: Add a proportion of z to each component in B equal to the proportions of the component within group B

This will ensure that the ratios between each count are maintained.

Convert groups B feature X to proportions of the sum of feature X components in group B only.

```
> Feature_x_B.prop = Feature_x_B.norm / sum(Feature_x_B.norm)
0.3052675   0.2384903   0.4562422

Proportion of z to add to each component in B
> Feature_x_B.prop*z
0.07402906   0.05783520   0.11064125

> Feature_x_B.spiked = Feature_x_B.norm + Feature_x_B.prop*z # add z to group B
0.3940291   0.3078352   0.5889021

Mean effect size now equal to 2
> mean(Feature_x_B.spiked) / mean(Feature_x_A.norm)
2
```

Step 4: For feature X abundances in group B, replace the normalized abundances with the spiked normalized abundances.

```
> table_norm["feature_X", 4:6] <- Feature_x_B.spiked
```

Step 5: Convert normalized abundances to unnormalized counts.

```
> table_spiked <- sweep(table_norm, MARGIN=2, lib, `*`) # multiple proportions by library size.
```

```
Counts of feature X after spiked: 5    9    2    9.85   7.39  13.54
Original counts of feature X:     5    9    2    8      6     6

Notice that the ratio between the counts of feature X before and after they
were spiked remains the same.

> table["feature_X",4] / table["feature_X",5] == table_spiked["feature_X",4] /
table_spiked["feature_X",5]
TRUE
> table["feature_X",5] / table["feature_X",6] == table_spiked["feature_X",5] /
table_spiked["feature_X",6]
TRUE
```

Difference in the relative abundance of the simulated metabolic pathways were assessed using the ALDEx2 with the Wilcoxon Rank Sum test and only 16 Monte Carlo samples were modeled for the sake of speed ($p< 0.05$ was deemed nominally significant) of the. True positives are the 20 spiked features.

Each log-ratio transformation method was evaluated for the following metrics: area under the curve (AUC), false discovery rate (FDR), and F1 score and a metric combining all three metrics into one score by adding AUC average score + F1 score + (1 – FDR score). Significance was tested for the 2-groups comparisons using the Welch's t-test and between all groups using ANOVA.

For the top 5 performing references of the alr transformation, we listed their variance rank (lowest to greatest) among the variance of all clr transformed features that were present in all samples. We also listed the "strain prevalence," which is the number of strains from the total strains found among all 40 participants in which the pathway was present in.

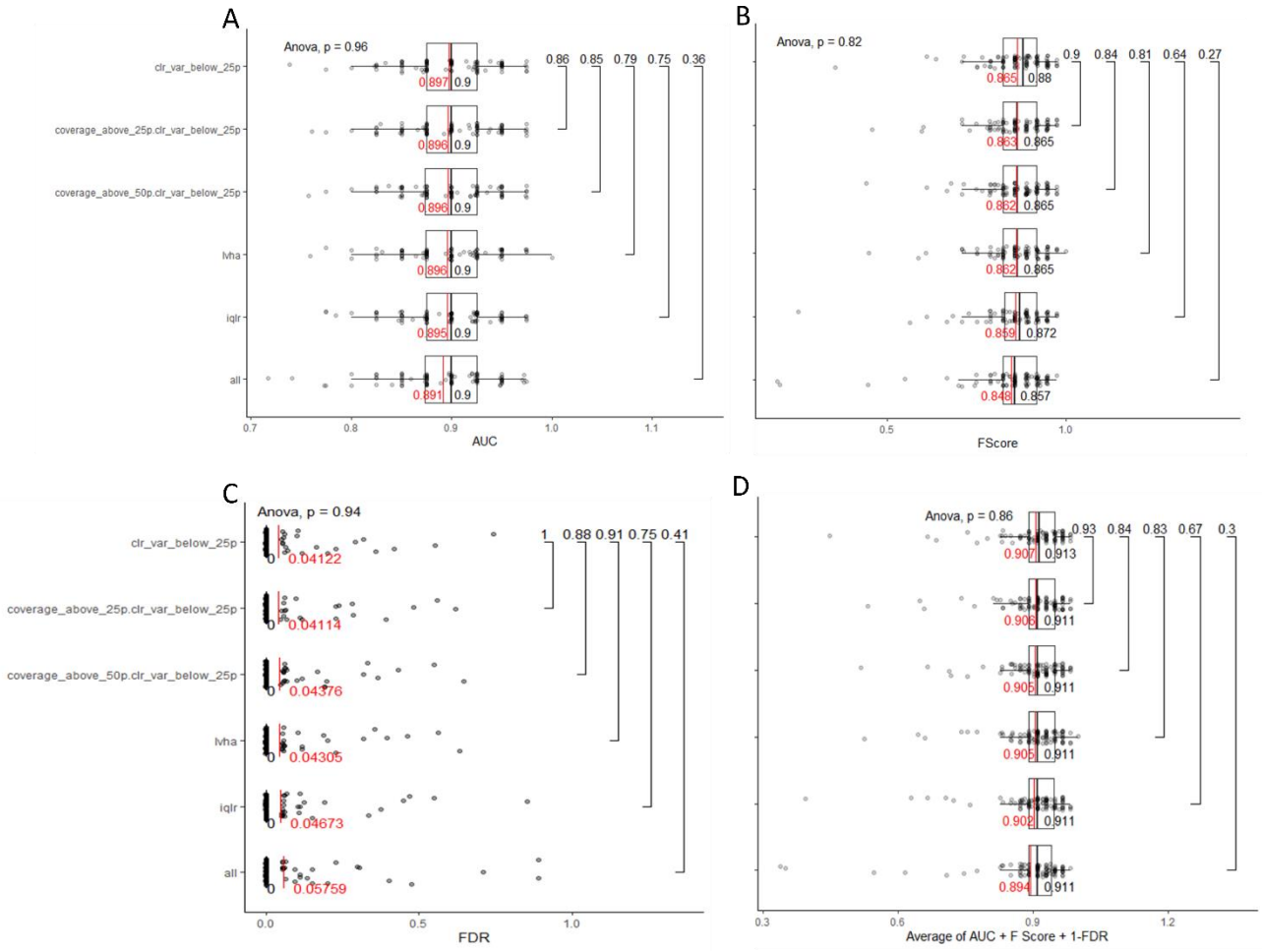
Results

When benchmarking methods with features spiked to have a mean effect size of 2 times, clr_var_below_25p performed the best and the classical clr method ('all') performed the worst among the clr-like methods (Supp. Fig. 1.A-D). Among the all the references compared for alr transformation, HEXITOLDEGSUPER-PWY performed the best (Supp. Fig. 2.A-D). None of the pair-wise comparisons of the top of performing methods in either the clr-based methods or alr references were significantly different ($p>0.05$, Supp. Fig. 1-2). Interestingly, out of the 143 pathways that are present in all samples, HEXITOLDEGSUPER-PWY had the smallest variance of the variance of clr transformed features that were present among all samples. Strain prevalence didn't seem to be a factor of performance (Supp. Fig. 2-D). HEXITOLDEGSUPER-PWY performed better than clr_var_below_25p when comparing false discovery rate, but were not significant (Supp. Fig. 3). The results were similar when evaluating mean effect size of 1.5 (data not shown).

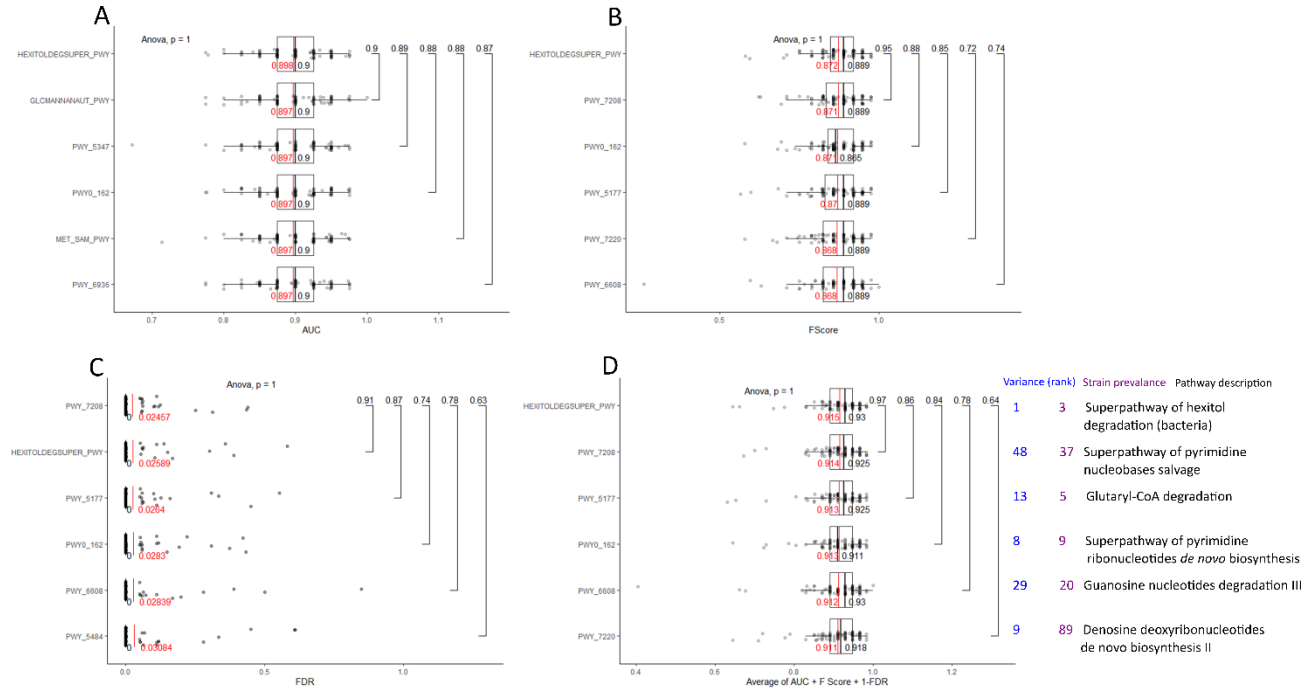
Conclusion

Our results indicate that all the clr-like methods and the top 5 performing alr references performed similarly (were not significantly different from each other). Nonetheless, the best performing log-ratio transformation method was alr using the feature reference with the smallest variance of the clr transformed features whereas the worst performing method was the traditional clr transformation.

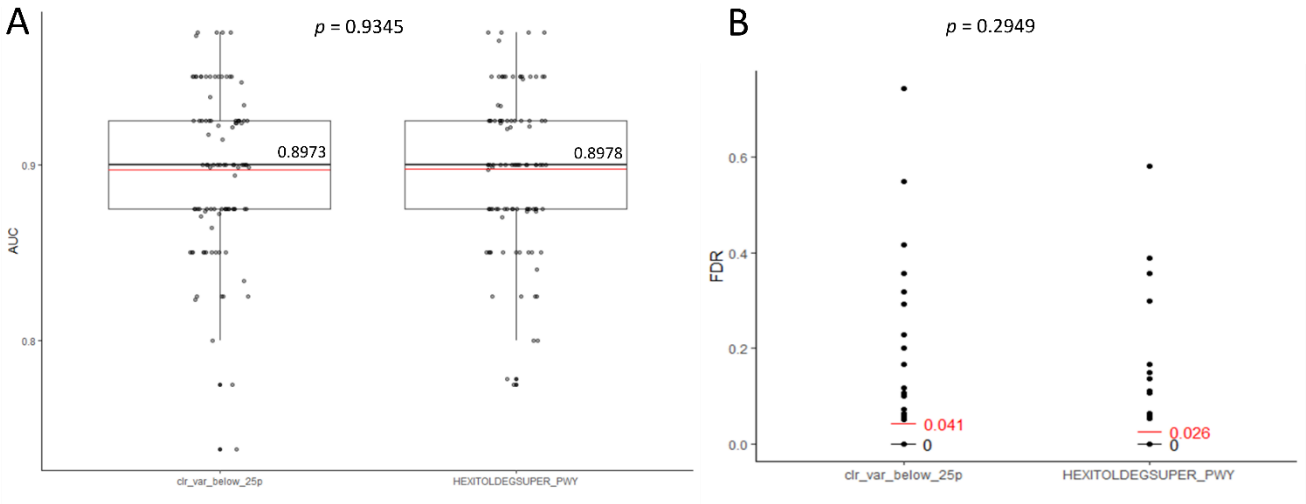
Supplementary Figure 1. Benchmarking centered-log ratio-like transformation methods using two-group comparisons of simulated pathway relative abundance datasets. Y-axis labels are ordered from the best performing to the worst. Red lines and texts indicate the median and the black lines and texts indicate the mean.



Supplementary Figure 2. Benchmarking references for the additive-log ratio transformation methods using two-group comparisons of simulated pathway relative abundance datasets. Y-axis labels are ordered from the best performing to the worst. Red lines and texts indicate the median and the black lines and texts indicate the mean.



Supplementary Figure 3. Comparing between the top performing additive-log ratio transformation methods and clr-like methods. Y-axis labels are ordered from the best performing to the worst. Red lines and texts indicate the median and the black lines and texts indicate the mean.



1. Fernandes AD, Reid JN, Macklaim JM, McMurrough TA, Edgell DR, Gloor GB. Unifying the analysis of high-throughput sequencing datasets: characterizing RNA-seq, 16S rRNA gene sequencing and selective growth experiments by compositional data analysis. Microbiome. 2014;2:15.

2. Quinn TP, Erb I, Richardson MF, Crowley TM. Understanding sequencing data as compositions: an outlook and review. Bioinformatics. 2018;34(16):2870-8.