ANOVA-Like Differential Expression tool for high throughput sequencing data

Greg Gloor*[1]

[1]Dep't of Biochemistry, University of Western Ontario

*ggloor@uwo.ca

27 July 2018

Abstract

Instructions on installing and using probabilisitic modelling and compositional data analysis using ALDEx2.

Package

ALDEx2 1.11.0

Contents

1 Why the *ALDEx2* package?

Fundamentally, many high throughput sequencing approaches generate similar data: reads are mapped to features in each sample, these features are normalized, then statistical difference between the features composing each group or condition is calculated[1]. The standard statistical tools used to analyze RNA-seq, ChIP-seq, 16S rRNA gene sequencing, metagenomics, etc. are fundamentally different for each approach despite the underlying similarity in the data structures. In most cases the values expected by, and modelled by, these tools is counts[2].

[1] Fernandes et al. (2014)

[2] Gierliński et al. (2015)

ALDEx2 breaks with this approach. Fundamentally, *ALDEx2* models the data as the *probability* of observing the count. In general, the observed data are single technical replicates, and the single observed count of a feature is one example from a distribution of examples that could have been observed under a repeated sampling model. The total read depth for a sample contains only information on the precision, and nothing else. *ALDEx2* provides a consistent framework that encompasses essentially all high throughput sequencing data types by modelling the data as a log-ratio transformed probability distribution rather than as counts[3].

[3] Fernandes et al. (2013);Fernandes et al. (2014)

2 Introduction to *ALDEx2*

This guide provides an overview of the R package ALDEx version 2 (*ALDEx2*) for differential (relative) abundance analysis of proportional data[4]. The package was developed and used initially for multiple-organism RNA-Seq data generated by high-throughput sequencing platforms (meta-RNA-Seq)[5], but testing showed that it performed very well with traditional RNA-Seq datasets[6], 16S rRNA gene variable region sequencing[7] and selective growth-type (SELEX) experiments[8]. In principle, the analysis method should be applicable to nearly any type of data that is generated by high-throughput sequencing that generates tables of per-feature counts for each sample(Fernandes et al. 2014): in addition to the examples outlined above, this would include ChIP-Seq or metagenome sequencing. We will be including examples and citations for application on these types of problems as we move forward.

[4] all high throughput sequencing data are compositional (Gloor et al. 2017) because of constraints imposed by the instruments

[5] Macklaim et al. (2013)

[6] Quinn, Crowley, and Richardson (2018)

[7] Bian et al. (n.d.)

[8] McMurrough et al. (2014);Wolfs et al. (2016)

The *ALDEx2* package in Bioconductor is modular and is suitable for the comparison of many different experimental designs. This is achieved by exposing the underlying centre log-ratio transformed Dirichlet Monte-Carlo replicate values to make it possible for anyone to add the specific R code for their experimental design — a guide to these values is outlined below.

ALDEx2 estimates per-feature technical variation within each sample using Monte-Carlo instances drawn from the Dirichlet distribution. This distribution maintains the proportional nature of the data and returns a multivariate probability distribution. *ALDEx2* uses the centred log-ratio (clr) transformation that ensures the data are scale invariant and sub-compositionally coherent[9]. The scale invariance property removes the need for a between sample data normalization step since the data are all placed on a consistent numerical co-ordinate. The sub-compositional coherence property ensures that the answers obtained are consistent when parts of the dataset are removed (e.g., removal of rRNA reads from RNA-seq studies or rare OTU species from 16S rRNA gene amplicon studies). All feature abundance values are expressed relative to the geometric mean abundance of all features in a sample. This is conceptually similar to a quantitative PCR where abundances are expressed relative to a standard: in the case of the clr transformation, the standard is the per-sample geometric mean abundance. See Aitchison (1986) for a complete description.

[9] Aitchison (1986)

3 Installation

There are multiple ways to download and install the most current of *ALDEx2*. *ALDEx2* will run with only the base R packages and is capable of running several functions with the 'parallel' package if installed. It has been tested with version R version 3, but should run on version 2.12 onward. It is recommended that the package be run on the most up-to-date R and Bioconductor versions. *ALDEx2* will make use of the BiocParallel package if possible, otherwise, *ALDEx2* will run in serial mode.

4 Quick Start: `aldex` with 2 groups:

ALDEx2 contains an `aldex` wrapper function that can perform many simple analyses. This wrapper will link the modular elements together to emulate *ALDEx2* prior to the modular approach. Note that if the test is 'kw', then `effect` should be FALSE. If the test is 't', then effect should be set to `TRUE`. The 't' option evaluates the data as a two-factor experiment using both the Welch's t and the Wilcoxon rank tests. The 'kw' option evaluates the data as a one-way ANOVA using the glm and Kruskal-Wallace tests. All tests include a Benjamini-Hochberg correction of the raw P values. The data can be plotted onto Bland-Altmann[10] (MA) or effect (MW) plots[11] for for two-way tests using the `aldex.plot` function. See the end of the modular section for examples of the plots.

[10] Altman and Bland (1983)

[11] Gloor, Macklaim, and Fernandes (2016)

Case study a growth selection type (SELEX) experiment[12]. This section contains an analysis of a dataset collected where a single gene library was made that contained 1600 sequence variants at 4 codons in the sequence. These variants were cloned into an expression vector at equimolar amounts. The wild-type version of the gene conferred resistance to a topoisomerase toxin. Seven independent growths of the gene library were conducted under selective and non-selective conditions and the resulting abundances of each variant was read out by sequencing a pooled, barcoded library on an Illumina MiSeq. The data table is included as selex_table.txt in the package. In this data table, there are 1600 features and 14 samples. The analysis takes approximately 2 minutes and memory usage tops out at less than 1Gb of RAM on a mobile i7 class processor when we use 128 Dirichlet Monte-Carlo Instances (DMC). For speed concerns we use only the first 400 features and perform only 16 DMC. The command used for *ALDEx2* is presented below:

[12] McMurrough et al. (2014)

First we load the library and the included selex dataset. Then we set the comparison groups. This must be a vector of conditions in the same order as the samples in the input counts table. The `aldex` command is calling several other functions in the background, and each of them returns diagnostics.

```
library(ALDEx2)
data(selex)
#subset only the last 400 features for efficiency
selex.sub <- selex[1:400,]

conds <- c(rep("NS", 7), rep("S", 7))
x.all <- aldex(selex.sub, conds, mc.samples=16, test="t", effect=TRUE,
     include.sample.summary=FALSE, denom="all", verbose=FALSE)

par(mfrow=c(1,2))
aldex.plot(x.all, type="MA", test="welch", xlab="Log-ratio abundance",
```

```
    ylab="Difference")
aldex.plot(x.all, type="MW", test="welch", xlab="Dispersion",
    ylab="Difference")
```

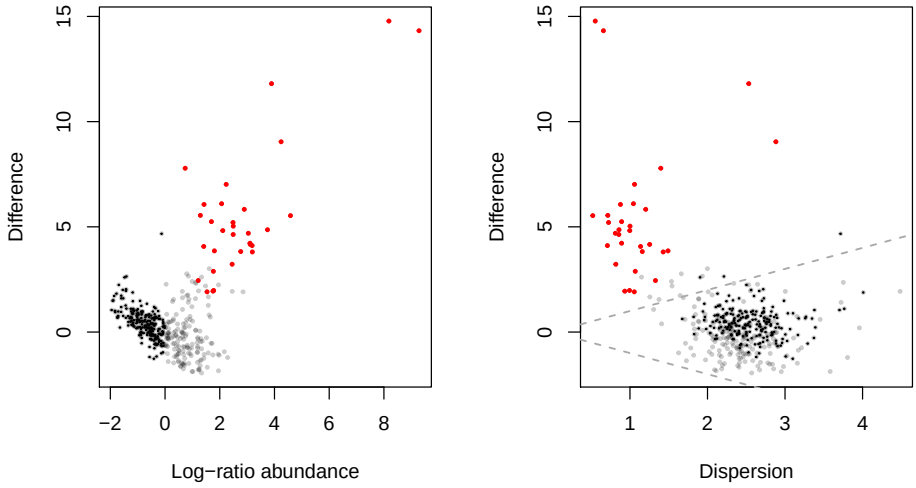


Figure 1: MA and Effect plots of ALDEx2 output
The left panel is an Bland-Altman or MA plot that shows the relationship between abundance and Difference. The right panel is an effect plot that shows the relationship between Difference and Dispersion. In both plots features that are not significant are in grey or black. Features that are statistically significant are in red. The Log-ratio abundance axis is the clr value for the feature.

5 Using *ALDEx2* modules

The modular approach exposes the underlying intermediate data so that users can generate their own tests. The simple approach outlined above just calls `aldex.clr, aldex.ttest, aldex.effect` in turn and then merges the data into one object. We will show these modules in turn, and then examine additional modules.

5.1 The `aldex.clr` module

The workflow for the modular approach first generates random instances of the centred log-ratio transformed values. There are three inputs: counts table, a vector of conditions, the number of Monte-Carlo instances, a string indicating if iqlr, zero or all feature are used as the denominator is required, and level of verbosity (TRUE or FALSE). We recommend 128 or more mc.samples for the t-test, 1000 for a rigorous effect size calculation, and at least 16 for ANOVA.

This operation is fast.

```
x <- aldex.clr(selex.sub, conds, mc.samples=16, denom="all", verbose=F)
```

5.2 The `aldex.ttest` module

The next operation performs the Welch's t and Wilcoxon rank test for the instance when there are only two conditions. There are three inputs: the aldex object from `aldex.clr`, the vector of conditions, whether a paired test should be conducted or not (TRUE or FALSE).

This operation is reasonably fast.

```
x.tt <- aldex.ttest(x, paired.test=FALSE, verbose=FALSE)
```

5.3 The `aldex.kw` module

Alternatively to the t-test, the user can perform the glm and Kruskal Wallace tests for one-way ANOVA of two or more conditions. Here there are only two inputs: the aldex object from aldex.clr, and the vector of conditions. Note that this is slow! and is not evaluated for this documentation.

```
x.kw <- aldex.kw(x)
```

5.4 The `aldex.effect` module

Finally, we estimate effect size and the within and between condition values in the case of two conditions. This step is required for plotting. There are four inputs: the aldex object from aldex.clr, the vector of conditions, a flag as to whether to include values for all samples or not are used as the denominator, and the level of verbosity.

```
x.effect <- aldex.effect(x, verbose=FALSE)
```

5.5 The `aldex.plot` module

Finally, the t-test and effect data are merged into one object.

```
x.all <- data.frame(x.tt,x.effect)
```

And the data are plotted. We see that the plotted data in Figure 1 and 2 are essentially the same.

```
par(mfrow=c(1,2))
aldex.plot(x.all, type="MA", test="welch")
aldex.plot(x.all, type="MW", test="welch")
```

5.6 Complex study designs and the aldex.glm module

The aldex.glm module has been included so that the probabilistic compositional approach can be used for complex study designs. This module is substantially slower than the two-comparison tests above, but we think it is worth it if you have complex study designs.

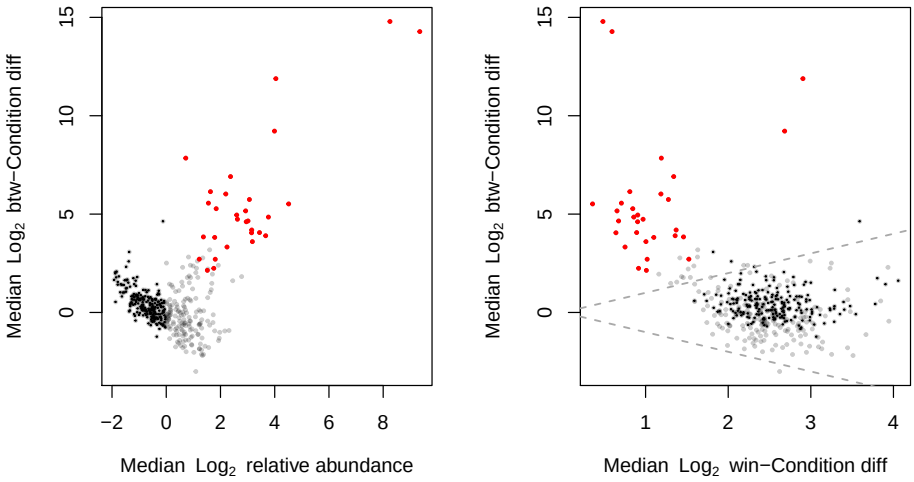


Figure 2: Output from aldex.plot function
The left panel is the MA plot, the right is the MW (effect) plot. In both plots red represents features called as differentially abundant with q < 0.1; grey are abundant, but not non-differentially abundant; black are rare, but not differentially abundant. This function uses the combined output from the aldex.ttest and aldex.effect functions

Essentially, the approach is the modular approach above but using a model matrix and covariates supplied to the glm function in R. The values returned are the expected values of the glm function given the inputs. In the example below, we are measuring the predictive value of variables A and B independently. See the documentation for the R formula function, or http://faculty.chicagobooth.edu/richard.hahn/teaching/formulanotation.pdf for more information.

Validation of features that are differential under any of the variables identified by the aldex.glm function should be performed using the aldex.effect function as a post-hoc test.

```
 covariates <- data.frame("A" = sample(0:1, 14, replace = TRUE),
     "B" = c(rep(0, 7), rep(1, 7)))
 mm <- model.matrix(~ A + B, covariates)
 x <- aldex.clr(selex.sub, mm, mc.samples=8, denom="all")
## operating in serial mode
## Warning in aldex.clr.function(reads, conds, mc.samples, denom, verbose, :
## values are unreliable when estimated with so few MC smps
## computing center with all features
 glm.test <- aldex.glm(x, mm)
## Warning in if (verbose) message("running tests for each MC instance:"): the
## condition has length > 1 and only the first element will be used
## running tests for each MC instance:
## Warning in if (verbose) numTicks <- progress(i, k, numTicks): the condition
## has length > 1 and only the first element will be used
## |------
## Warning in if (verbose) numTicks <- progress(i, k, numTicks): the condition
## has length > 1 and only the first element will be used
## -----
## Warning in if (verbose) numTicks <- progress(i, k, numTicks): the condition
## has length > 1 and only the first element will be used
## -(25%)----
## Warning in if (verbose) numTicks <- progress(i, k, numTicks): the condition
```

```
## has length > 1 and only the first element will be used
## -----
## Warning in if (verbose) numTicks <- progress(i, k, numTicks): the condition
## has length > 1 and only the first element will be used
## -(50%)----
## Warning in if (verbose) numTicks <- progress(i, k, numTicks): the condition
## has length > 1 and only the first element will be used
## -----
## Warning in if (verbose) numTicks <- progress(i, k, numTicks): the condition
## has length > 1 and only the first element will be used
## -(75%)----
## Warning in if (verbose) numTicks <- progress(i, k, numTicks): the condition
## has length > 1 and only the first element will be used
## ------|
 par(mfrow=c(1,2))
 plot(glm.test[,"model.A Pr(>|t|).BH"], x.all$we.eBH, log="xy",
     xlab="glm model A", ylab="Welch's t-test")
 plot(glm.test[,"model.B Pr(>|t|).BH"], x.all$we.eBH, log="xy",
     xlab="glm model A", ylab="Welch's t-test")
```

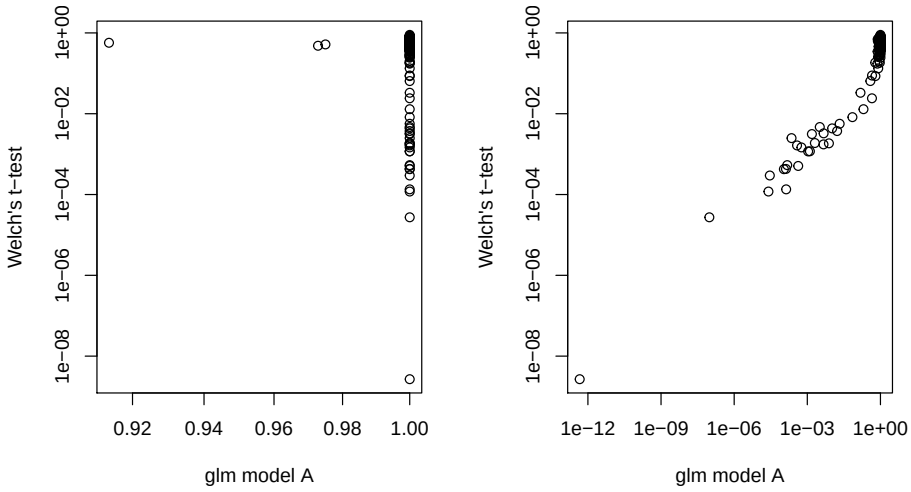


Figure 3: **Comparing the glm and t-test**
Plots of the expected P values for variables output from the glm function with model A (random predictors), and model B (actual study design groups) compared to the P values output by the Welch's t-test. This toy model shows how to use a generalized linear model within theALDEx2 framework. All values are the expected value of the test statistic after correction for multiple hypothesis testing.

6 ALDEx2 outputs

6.1 Expected values

ALDEx2 returns expected values for summary statistics. It is important to note that ALDEx uses Bayesian sampling from a Dirichlet distribution to estimate the underlying technical variation. This is controlled by the number of `mc.samples`, in practice we find that setting this to 16 or 128 is sufficient for most cases as *ALDEx2* is estimating the expected value of the distributions[13].

[13] Fernandes et al. (2013);Fernandes et al. (2014);Gloor et al. (2016)

In practical terms, *ALDEx2* takes the biological observations as given, but infers technical variation (sequencing the same sample again) multiple times using the `aldex.clr` function. Thus, the expected values returned are those that would likely have been observed *if the same samples had been run multiple times*. The user is cautioned that the number of features called as differential will vary somewhat between runs because of this sampling procedure. However, only features with values close to the chosen significance cutoff will vary between runs.

Several papers have suggested that *ALDEx2* is unable to properly control for the false discovery rate since the P values returned do not follow a random uniform distribution, but rather tend to cluster near a value of 0.5[14]Thorsen et al. (2016). These studies indicate that point estimate approaches are very sensitive to particular experimental designs and differences in sparsity and read depth. However, *ALDEx2* is not sensitive to these characteristics of the data, but seem to under-report the true FDR. The criticisms miss the mark on *ALDEx2* because *ALDEx2* reports the *expected* P value across the Dirichlet Monte-Carlo replicates. Features that are differential simply because of the vagaries of random sampling will indeed have a random uniform P value as point estimates, but will have an expected P value after repeated random sampling of 0.5. In contrast, features that are differential because of true biological variation are robust to repeated random sampling. Thus, *ALDEx2* identifies as differential only those features where simple random sampling (the minimal NULL hypothesis) cannot explain the difference.

[14] Hawinkel et al. (2018);

In our experience, we observe that *ALDEx2* returns a set of features that is very similar to the set returned as the intersect of multiple independent tools—a common recommendation when examining HTS datasets[15]

[15] Soneson and Delorenzi (2013)

6.2 Explaining the outputs

variant	we.ep	we.eBH	wi.ep	wi.eBH	kw.ep
A:D:A:D	4.03010e-01	0.63080705	0.239383012	0.43732819	0.21532060
A:D:A:E	1.15463e-01	0.34744596	0.040901806	0.15725841	0.03745315
A:E:A:D	8.98797e-05	0.00329076	0.000582750	0.00820759	0.00174511

variant	kw.eBH	glm.ep	glm.eBH	rab.all	rab.win.NS	rab.win.S
A:D:A:D	0.3932743	3.61061e-01	5.23582e-01	1.42494	1.30886	2.45384
A:D:A:E	0.1486590	8.12265e-02	1.92292e-01	1.71230	1.49767	4.23315
A:E:A:D	0.0245786	7.73660e-08	3.35492e-06	3.97484	1.41163	11.02154

variant	diff.btw	diff.win	effect	overlap
A:D:A:D	1.12261	1.72910	0.471043	0.267260701
A:D:A:E	2.73090	2.38134	1.034873	0.135857781
A:E:A:D	9.64287	2.85008	3.429068	0.000156632

In the list below, the `aldex.ttest` function returns the values highlighted with $*$, the `aldex.kw` function returns the values highlighted with $\circ$, and the `aldex.effect` function returns the values highlighted with $\diamondsuit$.

1. $*$ we.ep - Expected P value of Welch's t test
2. $*$ we.eBH - Expected Benjamini-Hochberg corrected P value of Welch's t test
3. $*$ wi.ep - Expected P value of Wilcoxon rank test
4. $*$ wi.eBH - Expected Benjamini-Hochberg corrected P value of Wilcoxon test
5. $\circ$ kw.ep - Expected P value of Kruskal-Wallace test
6. $\circ$ kw.eBH - Expected Benjamini-Hochberg corrected P value of Kruskal-Wallace test
7. $\circ$ glm.ep - Expected P value of glm test
8. $\circ$ glm.eBH - Expected Benjamini-Hochberg corrected P value of glm test
9. $\diamondsuit$ rab.all - median clr value for all samples in the feature
10. $\diamondsuit$ rab.win.NS - median clr value for the NS group of samples
11. $\diamondsuit$ rab.win.S - median clr value for the S group of samples
12. $\diamondsuit$ rab.X1_BNS.q50 - median expression value of features in sample X1_BNS if `include.item.summary=TRUE`
13. $\diamondsuit$ dif.btw - median difference in clr values between S and NS groups
14. $\diamondsuit$ dif.win - median of the largest difference in clr values within S and NS groups
15. $\diamondsuit$ effect - median effect size: diff.btw / max(diff.win) for all instances
16. $\diamondsuit$ overlap - proportion of effect size that overlaps 0 (i.e. no effect)

6.3 A word about effect size and overlap

The effect size metric used by *ALDEx2* is a standardized distributional effect size metric developed specifically for this package. The measure is somewhat robust, allowing up to 20% of the samples to be outliers before the value is affected, returns an effect size that is 71% the size of Cohen's d on a Normal distribution, and requires at worst twice the number of samples as a fully parametric method (which are not robust) would to estimate values with the same precision. The metric is equally valid for Normal, random uniform and Cauchy distributions^((**???**) submitted).

We prefer to use the effect size whenever possible rather than statistical significance since an effect size tells the scientist what they want to know—"what is reproducibly different between groups"; this is emphatically not something that P values deliver. We find that using the effect size returns a consistent set of true positive features irregardless of sample size, unlike P value based methods. Furthermore, over half of the the false positive features that are observed at low sample sizes have and effect size $> 0.5 \times \mathrm{E}$ the chosen effect size cutoff E. This is true regardless of the source of the dataset ((**???**) submitted).

We suggest that an effect size cutoff of 1 or greater be used when analyzing HTS datasets. If preferred the user can also set a fold-change cutoff as is commonly done with P value based methods.

The plot below shows the relationship between effect size and P values and BH-adjusted P values in the test dataset.

```
par(mfrow=c(1,2))
plot(x.all$effect, x.all$we.ep, log="y", cex=0.7, col=rgb(0,0,1,0.2),
  pch=19, xlab="Effect size", ylab="P value", main="Effect size plot")
points(x.all$effect, x.all$we.eBH, cex=0.7, col=rgb(1,0,0,0.2),
  pch=19)
abline(h=0.05, lty=2, col="grey")
legend(15,1, legend=c("P value", "BH-adjusted"), pch=19, col=c("blue", "red"))

plot(x.all$diff.btw, x.all$we.ep, log="y", cex=0.7, col=rgb(0,0,1,0.2),
  pch=19, xlab="Difference", ylab="P value", main="Volcano plot")
points(x.all$diff.btw, x.all$we.eBH, cex=0.7, col=rgb(1,0,0,0.2),
  pch=19)
abline(h=0.05, lty=2, col="grey")
```

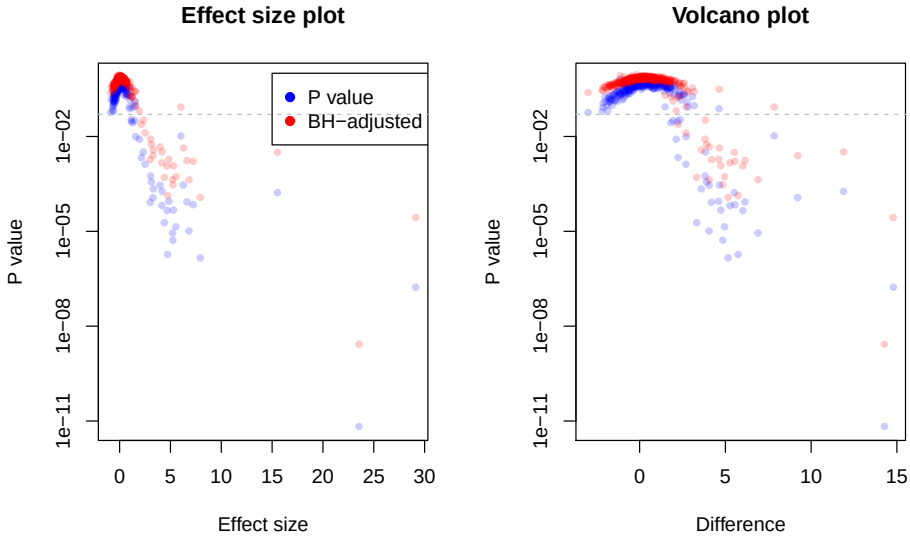


Figure 4: **Relationship between effect, difference, and P values**
We can see that the effect size has a much tighter relationship to the P value than does the raw difference. The effect size is relatively stable across datasets, but the P value become progressively smaller as the sample size is increased.

6.3.1 Alternative plotting of outputs

The built-in aldex.plot function described above will usually be sufficient, but for more user control the example in Figure 4 shows a plot that shows which features are found by the Welchs' or Wilcoxon test individually (blue) or by both (red).

```
# identify which values are significant in both the t-test and glm tests
found.by.all <- which(x.all$we.eBH < 0.05 & x.all$wi.eBH < 0.05)

# identify which values are significant in fewer than all tests
found.by.one <- which(x.all$we.eBH < 0.05 | x.all$wi.eBH < 0.05)

# plot the within and between variation of the data
```

```
plot(x.all$diff.win, x.all$diff.btw, pch=19, cex=0.3, col=rgb(0,0,0,0.3),
 xlab="Dispersion", ylab="Difference")
points(x.all$diff.win[found.by.one], x.all$diff.btw[found.by.one], pch=19,
 cex=0.7, col=rgb(0,0,1,0.5))
points(x.all$diff.win[found.by.all], x.all$diff.btw[found.by.all], pch=19,
 cex=0.7, col=rgb(1,0,0,1))
abline(0,1,lty=2)
abline(0,-1,lty=2)
```

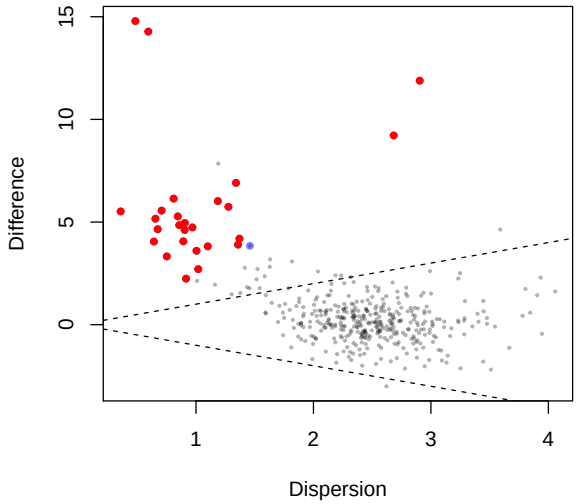


Figure 5: **Differential abundance in the selex dataset using the Welch's t-test or Wilcoxon rank test**
Features identified by both tests shown in red. Features identified by only one test are shown in blue dots. Non-significant features represent rare features if black and abundant features if grey dots.

7 Correcting for asymmetric datasets

In some cases we observe that the data returned by the centre log-ratio can be asymmetric. This occurs when the data are extremely asymmetric, such as when one group is largely composed of features that are absent in the other group. In this case the geometric mean will not accurately represent the appropriate basis of comparison for each group. An asymmetry can arise for many reasons: in RNA-seq it could arise because samples in one group contain a plasmid and the samples in the other group do not; in metagenomics or 16S rRNA gene sequencing it can arise when the samples in the two groups are taken from different environments; in a selex experiment it can arise because the two groups are under different selective constraints. The asymmetry can manifest either as a difference in sparsity (i.e., one group contains more 0 value features than the other) or as a systematic difference in abundance. When this occurs the geometric mean of each sample and group can be markedly different, and thus an inherent skew in the dataset can occur that leads to false positive and false negative feature calls. Asymmetry generally shows as a the centre of mass of the histogram for the x.all$diff.btw or x.all$effect being not centred around zero[16]. We recommend that all datasets be examined for asymmetry.

[16] Wu et al (in prep)

The approach taken by *ALDEx2* is to identify those features that are relatively invariant in all features in the entire dataset even though many features could be asymmetric between the groups. Fundamentally, the log-ratio approach requires that the denominator across all samples be comparable. The output of `aldex.clr` contains the offset of the features used for the denominator in the `@denom` slot.

7.1 Methods to correct for asymmetry

The `aldex.clr` function incorporates multiple approaches to deal with asymmetric datasets:

IMPORTANT: all rows must contain one or more counts when the user defines the row indices to ensure the appropriate rows are chosen

1. *all*: The default is to calculate the geometric mean of all features using the centred log-ratio of Aitchison[17]. This is the usual method for the compositional data analysis approach.

[17] Aitchison:1986

2. _iqlr: The *iqlr* method identifies those features that exhibit reproducible variance in the entire dataset. This is called the inter-quartile log-ratio or *iqlr* approach. For this, a uniform prior of 0.5 is applied to the dataset, the clr transformation is applied, and the variance of each feature is calculated. Those features that have variance values that fall between the first and third quartiles of variance for all features in all groups in the dataset are retained. When aldex.clr is called, the geometric mean abundance of only the retained features is calculated and used as the denominator for log-ratio calculations. Modelling shows that this approach is effective in dealing with datasets with up to 25% of the features being asymmetric. The approach has the advantage it has little or no effect on symmetric datasets and so is a safe approach if the user is unsure if the data is mildly asymmetric.

3. *lvha*: This method identifies those features that in the bottom quartile for variance in each group and the top quartile for relative abundance for each sample and across the entire dataset. This method is appropriate when the groups are very asymmetric, but there are some features that are expected to be relatively constant. The basic idea here is to identify those features that are relatively constant across all samples, similar to

the features that would be chosen as internal standards in qPCR. Experience suggests that meta-genomic and meta-transcriptomic datasets can benefit from this method of choosing the denominator. This method does not work with the selex dataset, since no features fit the criteria.

4. *zero*: This approach identifies and uses only those features that are non-zero in each group. In this approach the per-group non-zero features are used when `aldex.clr` calculates the geometric mean in the clr transformation. This method is appropriate when the groups are very asymmetric, but the experimentalist must ask whether the comparison is valid in these extreme cases.

5. *user*: The last new approach is to let the user define the set of 'invariant' features. In the case of meta-rna-seq, it could be argued that the levels of housekeeping genes should be standard for all samples. In this case the user could define the row indices that correspond to the particular set of housekeeping genes to use as the standard. *It is important that no row contain all 0 values for any feature when this method is used.*

6. *iterate*: This method identifies those features that are not statistically significantly different between groups using the statistical test of choice.

Figure 5 shows the effect of the iqlr correction on the example dataset. When the denominator is all, we see that the bulk of the points fall off the midpoint (dotted line), but that the bulk of the points are centered around 0 for the iqlr and lvha analysis. Thus, we have a demonstrably better centring of the data in the latter two methods. Practically speaking, we alter the p values and effect sizes of features near the margin of significance following iqlr or lvha transformation. The effect is largest for those features that are close to the bulk datapoints.

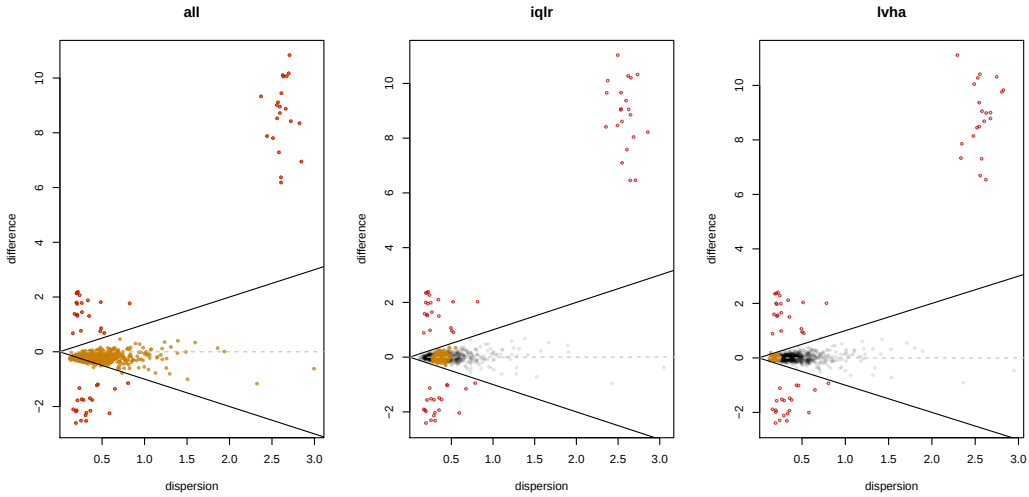
First the code:

```
# small synthetic dataset for illustration
# denominator features in x@denominator

data(synth2)
blocks <- c(rep("N", 10),rep("S", 10))
x <- aldex.clr(synth2, blocks, denom="all")
x.e <- aldex.effect(x)
plot(x.e$diff.win, x.e$diff.btw, pch=19, col=rgb(0,0,0,0.1), cex=0.5, xlab="dispersion", ylab="difference", r
points(x.e$diff.win[x@denom], x.e$diff.btw[x@denom], pch=19, col=rgb(0.8,0.5,0,0.7), cex=0.5)
points(x.e$diff.win[47:86], x.e$diff.btw[47:86], col=rgb(0.8,0,0,0.7), cex=0.5)
points(x.e$diff.win[980:1000], x.e$diff.btw[980:1000], col=rgb(0.8,0,0,0.7), cex=0.5)
abline(0,1)
abline(0,-1)
abline(h=0, col="gray", lty=2)
```

```
## operating in serial mode
## computing center with all features
## operating in serial mode
## sanity check complete
## rab.all  complete
## rab of samples complete
## within sample difference calculated
## between group difference calculated
## group summaries calculated
```

```
## effect size calculated
## summarizing output
## operating in serial mode
## computing iqlr centering
## operating in serial mode
## sanity check complete
## rab.all  complete
## rab of samples complete
## within sample difference calculated
## between group difference calculated
## group summaries calculated
## effect size calculated
## summarizing output
## operating in serial mode
## computing center with housekeeping features
## operating in serial mode
## sanity check complete
## rab.all  complete
## rab of samples complete
## within sample difference calculated
## between group difference calculated
## group summaries calculated
## effect size calculated
## summarizing output
```



\begin{figure}
\caption{*Differential abundance in a synthetic dataset using different denominators for the clr calculation. In this data 2% of the features are modelled to be sparse in one group but not the other. Features modelled to be different between two groups are shown in red. Features that are non-significant are in grey (or brown). Features used in the denominator shown in brown: the geometric mean of these features is used as the denominator when calculating the clr transformation. Lines of constant effect are drawn at 0, and ± 1. Note that the iqlr and lvha denominators place the middle of the non-asymmetric features at a between group difference of 0.*} \end{figure}

8 Contributors

I am grateful that *ALDEx2* has taken on a life of its own.

- Andrew Fernandes wrote the original ALDEx code, and designed *ALDEx2*.
- Jean Macklaim found and squished many bugs, performed unit testing, did much of the original validation. Jean also made seminal contributions to simplifying and explaining the output of *ALDEx2*.
- Matt Links incorporated several *ALDEx2* functions into a multicore environment.
- Adrianne Albert wrote the correlation and the one-way ANOVA modules in aldex.kw.
- Ruth Grace Wong added function definitions and made the parallel code functional with BioConducter.
- Jia Rong Wu developed and implemented the alternate denominator method to correct for asymmetric datasets.
- Andrew Fernandes, Jean Macklaim and Ruth Grace Wong contributed to the SumFunctionsAitchison.R code.
- Tom Quinn rewrote the t-test and Wilcoxon functions to make them substantially faster. He also wrote the current aldex.glm function. Tom's *propr*[18] R package is able to use the output from aldex.clr.
- Vladimir Mikryukov and everyone named above have contributed bug fixes
- Greg Gloor currently maintains *ALDEx2* and had and has roles in documentation, design, testing and implementation.

[18] Quinn et al. (2017)

9 Version information

Version 1.04 of ALDEx was the version used for the analysis in Macklaim et al.[19]. This version was suitable only for two-sample two-group comparisons, and provided only effect size estimates of difference between groups. ALDEx v1.0.4 is available at:

[19] Macklaim et al. (2013);Fernandes et al. (2013)

```
https://github.com/ggloor/ALDEx2/blob/master/ALDEx\_1.0.4.tar.gz
```

. No further changes are expected for that version since it can be replicated completely within *ALDEx2* by using only the `aldex.clr` and `aldex.effect` commands.

Versions 2.0 to 2.05 were development versions that enabled P value calculations. Version 2.06 of *ALDEx2* was the version used for the analysis in[20]. This version enabled large sample comparisons by calculating effect size from a random sample of the data rather than from an exhaustive comparison.

[20] Fernandes et al. (2014)

Version 2.07 of *ALDEx2* was the initial the modular version that exposed the intermediate calculations so that investigators could write functions to analyze different experimental designs. As an example, this version contains an example one-way ANOVA module. This is identical to the version submitted to Bioconductor as 0.99.1.

Future releases of *ALDEx2* now use the Bioconductor versioning numbering.

```
sessionInfo()
## R version 3.5.0 (2018-04-23)
## Platform: x86_64-apple-darwin17.6.0 (64-bit)
## Running under: macOS High Sierra 10.13.6
##
```

```
## Matrix products: default
## BLAS: /opt/local/Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.dylib
## LAPACK: /opt/local/Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_CA.UTF-8/en_CA.UTF-8/en_CA.UTF-8/C/en_CA.UTF-8/en_CA.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] ALDEx2_1.11.0   BiocStyle_2.8.2
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.18               compiler_3.5.0
##  [3] GenomeInfoDb_1.16.0        XVector_0.20.0
##  [5] bitops_1.0-6               tools_3.5.0
##  [7] zlibbioc_1.26.0            digest_0.6.15
##  [9] evaluate_0.11              lattice_0.20-35
## [11] Matrix_1.2-14              DelayedArray_0.6.2
## [13] yaml_2.1.19                parallel_3.5.0
## [15] xfun_0.3                   GenomeInfoDbData_1.1.0
## [17] stringr_1.3.1              knitr_1.20
## [19] S4Vectors_0.18.3           IRanges_2.14.10
## [21] stats4_3.5.0               rprojroot_1.3-2
## [23] multtest_2.36.0            grid_3.5.0
## [25] Biobase_2.40.0             survival_2.41-3
## [27] BiocParallel_1.14.2        rmarkdown_1.10
## [29] bookdown_0.7               magrittr_1.5
## [31] MASS_7.3-49                backports_1.1.2
## [33] htmltools_0.3.6            matrixStats_0.54.0
## [35] BiocGenerics_0.26.0        GenomicRanges_1.32.6
## [37] splines_3.5.0              SummarizedExperiment_1.10.1
## [39] stringi_1.2.4              RCurl_1.95-4.11
```

Aitchison, J. 1986. *The Statistical Analysis of Compositional Data*. London, England: Chapman & Hall.

Altman, D. G., and J. M. Bland. 1983. "Measurement in Medicine: The Analysis of Method Comparison Studies." *Journal of the Royal Statistical Society. Series D (the Statistician)* 32 (3). Wiley for the Royal Statistical Society:pp. 307–17. http://www.jstor.org/stable/2987937.

Bian, Gaorui, Gregory B Gloor, Aihua Gong, Changsheng Jia, Wei Zhang, Jun Hu, Hong Zhang, et al. n.d. "The Gut Microbiota of Healthy Aged Chinese Is Similar to That of the Healthy Young." *mSphere* 2 (5):e00327–17. https://doi.org/10.1128/mSphere.00327-17.

Fernandes, Andrew D, Jean M Macklaim, Thomas G Linn, Gregor Reid, and Gregory B Gloor. 2013. "ANOVA-Like Differential Expression (Aldex) Analysis for Mixed Population Rna-Seq." *PLoS One* 8 (7):e67019. https://doi.org/10.1371/journal.pone.0067019.

Fernandes, Andrew D, Jennifer Ns Reid, Jean M Macklaim, Thomas A McMurrough, David R Edgell, and Gregory B Gloor. 2014. "Unifying the Analysis of High-Throughput Sequencing Datasets: Characterizing RNA-Seq, 16S RRNA Gene Sequencing and Selective Growth Experiments by Compositional Data Analysis." *Microbiome* 2:15.1–15.13. https://doi.org/10.1186/2049-2618-2-15.

Gierliński, Marek, Christian Cole, Pietà Schofield, Nicholas J Schurch, Alexander Sherstnev, Vijender Singh, Nicola Wrobel, et al. 2015. "Statistical Models for Rna-Seq Data Derived from a Two-Condition 48-Replicate Experiment." *Bioinformatics* 31 (22):3625–30. https://doi.org/10.1093/bioinformatics/btv425.

Gloor, Gregory B., Jean M. Macklaim, Vera Pawlowsky-Glahn, and Juan J. Egozcue. 2017. "Microbiome Datasets Are Compositional: And This Is Not Optional." *Frontiers in Microbiology* 8:2224. https://doi.org/10.3389/fmicb.2017.02224.

Gloor, Gregory B, Jean M. Macklaim, and Andrew D. Fernandes. 2016. "Displaying Variation in Large Datasets: Plotting a Visual Summary of Effect Sizes." *Journal of Computational and Graphical Statistics* 25 (3C):971–79. https://doi.org/10.1080/10618600.2015.1131161.

Gloor, Gregory B, Jean M Macklaim, Michael Vu, and Andrew D Fernandes. 2016. "Compositional Uncertainty Should Not Be Ignored in High-Throughput Sequencing Data Analysis." *Austrian Journal of Statistics* 45:73–87. https://doi.org/doi:10.17713/ajs.v45i4.122.

Hawinkel, Stijn, Federico Mattiello, Luc Bijnens, and Olivier Thas. 2018. "A Broken Promise : Microbiome Differential Abundance Methods Do Not Control the False Discovery Rate." *BRIEFINGS IN BIOINFORMATICS*. http://dx.doi.org/10.1093/bib/bbx104.

Macklaim, M Jean, D Andrew Fernandes, M Julia Di Bella, Jo-Anne Hammond, Gregor Reid, and Gregory B Gloor. 2013. "Comparative Meta-RNA-Seq of the Vaginal Microbiota and Differential Expression by *Lactobacillus Iners* in Health and Dysbiosis." *Microbiome* 1:15. https://doi.org/doi: 10.1186/2049-2618-1-12.

McMurrough, Thomas A, Russell J Dickson, Stephanie M F Thibert, Gregory B Gloor, and David R Edgell. 2014. "Control of Catalytic Efficiency by a Coevolving Network of Catalytic and Noncatalytic Residues." *Proc Natl Acad Sci U S A* 111 (23):E2376–83. https://doi.org/10.1073/pnas.1322352111.

Quinn, Thomas P, Tamsyn M Crowley, and Mark F Richardson. 2018. "Benchmarking Differential Expression Analysis Tools for Rna-Seq: Normalization-Based Vs. Log-Ratio Transformation-Based Methods." *BMC Bioinformatics* 19 (1):274. https://doi.org/10.1186/s12859-018-2261-8.

Quinn, Thomas, Mark F Richardson, David Lovell, and Tamsyn Crowley. 2017. "Propr: An R-Package for Identifying Proportionally Abundant Features Using Compositional Data Analysis." *bioRxiv*. Cold Spring Harbor Labs Journals. https://doi.org/10.1101/104935.

Soneson, Charlotte, and Mauro Delorenzi. 2013. "A Comparison of Methods for Differential Expression Analysis of RNA-seq Data." *BMC Bioinformatics* 14:91. https://doi.org/10.1186/1471-2105-14-91.

Thorsen, Jonathan, Asker Brejnrod, Martin Mortensen, Morten A Rasmussen, Jakob Stokholm, Waleed Abu Al-Soud, Søren Sørensen, Hans Bisgaard, and Johannes Waage. 2016. "Large-Scale Benchmarking Reveals False Discoveries and Count Transformation Sensitivity in 16S RRNA Gene Amplicon Data Analysis Methods Used in Microbiome Studies." *Microbiome* 4 (1):62. https://doi.org/10.1186/s40168-016-0208-8.

Wolfs, Jason M, Thomas A Hamilton, Jeremy T Lant, Marcon Laforet, Jenny Zhang, Louisa M Salemi, Gregory B Gloor, Caroline Schild-Poulter, and David R Edgell. 2016. "Biasing Genome-Editing Events Toward Precise Length Deletions with an Rna-Guided Tevcas9 Dual Nuclease." *Proc Natl Acad Sci U S A*, December. https://doi.org/10.1073/pnas.1616343114.