LDM Package Version 2.0

Yi-Juan Hu and Glen A. Satten

January 23, 2021

Changes from Version 1.0

- Added functionalities for analyzing matched-set data
- Added functionalities for testing presence-absence associations
- Added useful summary statistics to the output
- Fixed some minor bugs
- The results below were generated from R 3.6.0, while the results in vignette 1.0 were generated from R 3.4.4.

1 Overview

The LDM package implements the Linear Decomposition Model (Hu and Satten 2020), which provides a single analysis path that includes global tests of any effect of the microbiome, tests of the effects of individual OTUs (operational taxonomic units) or ASVs (amplicon sequence variants) while accounting for multiple testing by controlling the false discovery rate (FDR), and a connection to distance-based ordination. It accommodates multiple covariates (e.g., clinical outcomes, environmental factors, treatment groups), either continuous or discrete (with ≥ 2 levels), as well as interaction terms to be tested either singly or in combination, allows for adjustment of confounding covariates, and uses permutation-based p-values that can control for clustered data (e.g., repeated measurements on the same individual). It gives results for both the frequency and arcsine-root-transformed data, and can give an "omnibus" test that combines results from analyses conducted on the two scales. In follow-up articles, we extended the LDM to test presence-absence associations (Hu, Lane, and Satten 2020) and to analyze matched sets of samples (Zhu, Satten, and Hu 2020).

The core function of the LDM package is `ldm`, which can be used to test association between covariates of interest and the microbiome, possibly after adjusting for confounding covariates. Several additional functions are included in the LDM package. The function

`permanovaFL` implements the PERMANOVA-FL test (our version of the PERMANOVA test); like the function `ldm`, `permanovaFL` allows adjustment of confounding covariates, control of clustered data, and multiple sets of covariates to be tested in the way that the sets are entered sequentially and the variance explained by each set is that part that remains after the previous sets have been fit; unlike the implementation of PERMANOVA in `adonis` and `adonis2` in the R package `vegan`, `permanovaFL` adopts the permutation scheme described by Freedman and Lane (1983). The function `adjust.data.by.covariates` produces the adjusted distance matrix and OTU table after removing the effects of covariates (e.g., confounders).

The function `ldm` uses two sequential stopping criteria for the permutation procedure. For the global test, LDM uses the stopping rule of Besag and Clifford (1991), which stops permutation when a pre-specified minimum number (default = 100) of rejections (i.e., the permutation statistic exceeded the observed test statistic) has been reached. For the OTU-specific tests, LDM uses the stopping rule of Sandve et al. (2011), which stops permutation when every OTU test has either reached the pre-specified number (default = 100) of rejections or yielded a q-value that is below the nominal FDR level (default = 0.1). As a convention, we call a test "*stopped*" if the corresponding stopping criterion for that test has been satisfied. All tests are terminated when a user-specified maximum number of permutations have been generated; if this number is "too small", some tests may not have "stopped". Tests that require a large number of permutations are those that are close to the threshold for acceptance or rejection. The function `permanovaFL` only uses the stopping rule of Besag and Clifford (1991) as it only implements a global test.

2 Example dataset 1: throat microbiome data

The throat microbiome data included in this package contain 60 subjects with 28 smokers and 32 non-smokers. Microbiome data were collected from right and left nasopharynx and oropharynx region to form an OTU table with 856 OTUs. The phylogenetic tree was constructed using UPGMA on the K80 distance matrix of the OTUs. Metadata include smoking status, packs smoked per year, age, gender, and antibiotic use. For further information on these data see Hu and Satten (2020) and references therein. We first prepare this dataset:

```
> library(LDM)
> library(GUniFrac) # for calculating UniFrac distances
>
> data(throat.tree)    # phylogenetic tree; only needed if UniFrac distances are requested
> data(throat.otu.tab) # OTU table
> data(throat.meta)    # metadata (covariates of interest, confounding covariates, etc.)
>
> throat.meta$AntibioticUse <-
            (throat.meta$AntibioticUsePast3Months_TimeFromAntibioticUsage != "None")
            # create the antibiotic use variable used in Hu and Satten (2020)
```

The OTU table should have rows corresponding to samples and columns corresponding to OTUs; `ldm` will transpose the OTU table if the number of rows is not equal to the length of the covariates in the metadata but this consistency check will fail in the unlikely case that the number of OTUs and samples are equal. Note that samples with zero reads for *every* OTU, possibly representing negative controls, are analyzed as they are, but a warning message will be sent. OTUs with zero reads in *every* sample will be automatically removed by `ldm` before any further analysis is carried out.

It is *essential* that the OTU table and the metadata have the same number of observations (samples) and that these occur *in the same order* in the OTU table and all covariates. Note also that fitting the LDM assumes no missing data; samples having a NA value for any covariate in the formula (including confounders) are removed by `ldm` before any further analysis is carried out. Similarly, if a `cluster.id` variable is specified (see section 8), samples having `cluster.id=NA` are removed.

As in the analysis presented in Hu and Satten (2020), we filtered out OTUs with presence in fewer than 5 samples, leaving 233 OTUs for analysis. Although this criterion is commonly used, its use is not necessary to run `ldm`. Note that the taxonomy table, if it exists, should be subsetted accordingly.

```
> otu_presence = which(colSums(throat.otu.tab>0)>=5)
> throat.otu.tab = throat.otu.tab[,otu_presence]
> dim(throat.otu.tab)
[1]  60 233
```

3 Ordination using distances adjusted for covariates

(Users who read for a quick introduction to the testing purpose of the LDM can skip this section). The function `adjust.data.by.covariates` produces (1) the adjusted OTU table that has the effect of the covariates (e.g., confounders) removed using projection and (2) the adjusted distance matrix that has the directions of the covariates projected off, without fitting or testing the LDM. These are also part of the adjustment of covariates in the LDM. The adjusted distance matrix can be used to perform ordination in which the effects of confounding covariates are removed. As an example, we perform distance-based ordination and visualize whether the samples from the throat data are clustered by smoking status (i.e., smokers vs. non-smokers) after removing the confounding effects from gender and antibiotic use.

```
> adj.data <- adjust.data.by.covariates(formula=~Sex+AntibioticUse, data=throat.meta,
                                        otu.table=throat.otu.tab, dist.method="bray")
> PCs <- eigen(adj.data$adj.dist, symmetric=TRUE)
```

Now we are ready to generate the ordination plot.

```
> color = rep("blue", length(throat.meta$SmokingStatus))
> w = which(throat.meta$SmokingStatus=="Smoker")
> color[w] = "red"
> plot(PCs$vectors[,1], PCs$vectors[,2], xlab="PC1", ylab="PC2",
       col=color, main="Ordination based on Bray-Curtis distance")
> legend(x="topleft", legend=c("smokers","non-smokers"), pch=c(21,21),
        col=c("red","blue"), lty=0)
```

The resulting plot is shown in Figure 1. By removing confounding directions from the distance matrix, any clustering pattern by smoking status is not due to correlation between smoking and gender and/or antibiotic use.

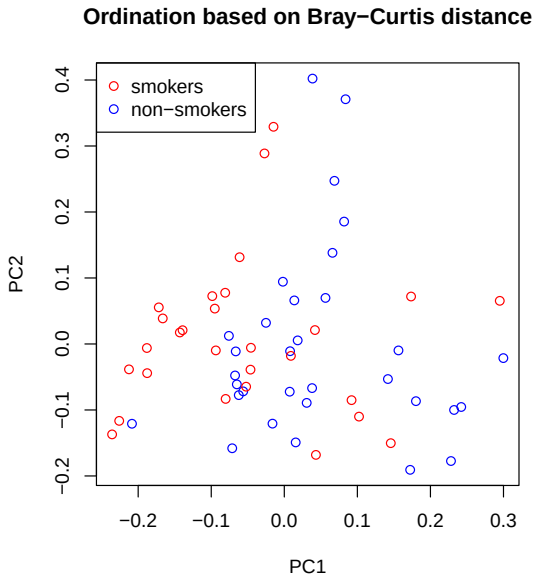

Figure 1 Ordination plot from analysis of the throat microbiome data after removing the effects of gender and antibiotic use.

4 Writing formulas for testing hypotheses using `ldm` and `permanovaFL`

The core function that implemented the LDM method is `ldm`. The function `ldm` uses a formula interface that is similar to other regression models in R, but with some differences. The general form of a formula for `LDM` is:

```
>  otu.table | (confounders) ~ (first set of covariates) + (second set of covariates)
                                                  ... + (last set of covariates)
```

The left hand side of the formula gives the OTU table and any confounders that need to be controlled for. The generic formula in the absence of confounders is

```
> otu.table ~ (first set of covariates) + (second set of covariates)
                                                ... + (last set of covariates)
```

On the right hand side of the formula, parentheses are used to group covariates that should be considered together as a single "submodel". The covariates in each submodel are considered jointly, and in order from left to right such that the variance explained by each submodel is that part that remains after the previous submodels have been fit (i.e., are controlled for). If a submodel contains only a single covariate, then the parentheses are optional. Similarly, since the confounders are always treated as a single submodel, the parentheses around the confounders is also optional. The formula

```
> otu.table ~ (confounders) + (first set of covariates) + (second set of covariates)
                                                ... + (last set of covariates)
```

will fit the same model as the first formula statement in this section. The only difference is that moving `(confounders)` to the right hand side of the model will produce a p-value for testing the confounders; specifying the confounders on the left hand side of the formula suppresses this p-value, which can speed up the `ldm`, both by reducing the number of test statistics calculated in each permutation and by possibly reducing the number of permutations required to satisfy all the stopping criteria.

The standard syntax for formulas in R can be used to expand single covariates or single terms into a model matrices with multiple columns. For example, choosing the first set of covariates to consist of a single factor variable with 2 levels is equivalent to specifying that the first set of covariates consist of 2 indicator variables (one for each factor level). Similarly, the formula

```
> otu.table ~ as.factor(a) + b*c
```

will have two submodels, the first with as many columns (degrees of freedom) as `a` has levels, and the second (using the standard R syntax for interactions) with columns corresponding to `(b, c, b×c)`.

Note that `permanovaFL` uses the same formula interface as `ldm`. In addition, `permanovaFL` allows the OTU table on the left hand side of the formula to be replaced by a pre-calculated distance matrix.

We now give an example of a formula that can be fit to the throat data. We first wish to test the association of the microbiome with smoking status and then test the association of the microbiome with pack per year after removing the association with smoking status, both tests accounting for the confounding effects of sex and antibiotic use. Thus we specify this formula in the `ldm`:

```
> throat.otu.tab | (Sex+AntibioticUse) ~ SmokingStatus + PackYears
```

5 Fitting the LDM

(Users who read for a quick introduction to the testing purpose of the LDM can skip this section). We can fit the LDM to the observed data (i.e., obtaining the variance explained [VE] without testing their significance) by calling `ldm` with `n.perm.max=0`. This is useful if we only want to see how much variability is explained by each set of covariates, or if we want factor loadings from OTUs. Unless specified otherwise by the user, the LDM automatically fits the OTU data at both the original frequency (i.e., relative abundance) scale and the arcsin-root-transformed frequency scale. Because we would also like to compare the variance explained by covariates to those by an ordination of the data using distance, an externally-calculated distance matrix can be used, or LDM can calculate the distance internally. Current distances supported include all of those supported by `vegdist` in the `vegan` package (i.e., "manhattan", "euclidean", "canberra", "bray", "kulczynski", "jaccard", "gower", "altGower", "morisita", "horn", "mountford", "raup" , "binomial", "chao", "cao", "mahalanobis") as well as "hellinger", "unwt-unifrac" (unweighted UniFrac), and "wt-unifrac" (weighted UniFrac). The default choice for distance is the Bray-Curtis distance. Note that when we internally call `vegdist` for its supported distances without changing any of the default settings, the OTU table is converted to the frequency scale if `scale.otu.table=TRUE`; the parameter `binary` is set to `TRUE` if a presence-absence-type of distance (e.g., "jaccard") is requested and `FALSE` otherwise (e.g., "bray"). The Hellinger distance measure ("hellinger") takes the form 0.5*E, where E is the Euclidean distance between the square-root-transformed frequency data. The UniFrac distances ("unwt-unifrac" and "wt-unifrac") are calculated by internally calling `GUniFrac` in the `GUniFrac` package. For the throat microbiome data, we choose the Bray-Curtis distance:

```
> fit <- ldm(throat.otu.tab|(Sex+AntibioticUse)~SmokingStatus+PackYears,
             data=throat.meta, dist.method="bray", n.perm.max=0)

> # frequency scale
> fit$VE.global.freq.submodels        # VE for submodels 1 and 2
[1] 0.13891274 0.02751291
> fit$VE.otu.freq.submodels[1,1:3]    # Contribution of OTUs 1-3 to VE for submodel 1
        4695         4194         5160
3.780166e-07 1.138872e-07 2.666018e-03

> fit$F.global.freq     # F statistics for VE of submodels 1 and 2
            [,1]
[1,] 0.049056264
[2,] 0.009716031
> fit$F.otu.freq[1,1:3] # F statistics for contribution of OTUs 1-3 in submodels 1 and 2
           4695        4194       5160
[1,] 0.09829664 0.000170493 0.04114451
[2,] 0.03139611 0.021585241 0.02909457

> # arcsin-root scale
```

```
> fit$VE.global.tran.submodels       # VE for submodels 1 and 2
[1] 1.2829126 0.3691146
```

The LDM performs a full decomposition (similar to SVD) of the OTU table; the singular values are returned in `d.freq` and `d.tran` (frequency and arcsin-root-transformed scale). It can be informative to make a scree plot to see how much variability is explained by submodel variables compared to the residual variability. The following code accomplishes this; the VE corresponding to the submodels is plotted in red, while residual components are plotted in black. In the code that follows, we handle the possibility of multiple terms per submodel by plotting a single point for each submodel; the value plotted is the average VE (i.e., averaged over each term in each submodel). We could also plot the VE for each term in each submodel by plotting the appropriate values of d.freq[2].

```
> scree.VE.freq <- c(fit$VE.global.freq.submodels/fit$VE.df.submodels,
  fit$VE.global.freq.residuals)
> color <- c(rep("red", 2), rep("black", length(scree.VE.freq)-2))
> plot(scree.VE.freq/sum(scree.VE.freq), main="Frequency Scale",
       xlab="Component", ylab="Proportion of total sum of squares", col=color)

> scree.VE.tran <- c(fit$VE.global.tran.submodels/fit$VE.df.submodels,
  fit$VE.global.tran.residuals)
> color <- c(rep("red", 2), rep("black", length(scree.VE.tran)-2))
> plot(scree.VE.tran/sum(scree.VE.tran), main="Arcsin-Root Scale",
       xlab="Component", ylab="", col=color)
```

The resulting plots are shown in Figure 2. We can see that although the VE by `SmokingStatus` does not rank the highest compared to other directions in the whole space (i.e., the augmented X matrix in the manuscript, which is denoted by x in the output of `ldm`), it still appears substantial. By contrast, the VE by `PackYears` seems very small. Although these results are suggestive, we must use permutation to decide which VEs are significant. The function `ldm` also returns the right singular vectors from the decomposition (denoted `v.freq` on the frequency scale and `v.tran` on the transformed scale) which can be used to construct biplots (see Satten et al. 2017 for additional details).

Finally, some users may prefer to specify the formula statement in a separate line:

```
> form <- throat.otu.tab | (Sex + AntibioticUse) ~ SmokingStatus + PackYears
```

and then call `ldm` using

```
> fit <- ldm(formula=form, data=throat.meta, dist.method="bray", n.perm.max=0)
```

which can be convenient when calling `ldm` multiple times with different settings but the same formula.

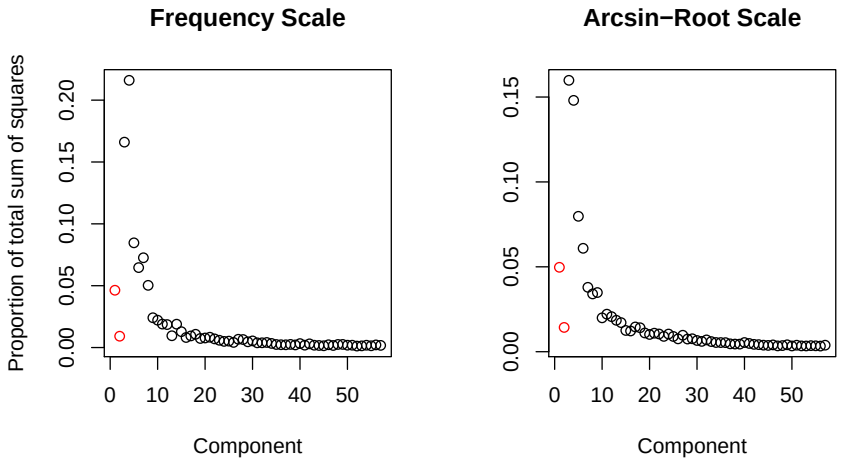

Figure 2 VEs when fitting the LDM to the throat microbiome data. The first two red circles represent VEs by `SmokingStatus` and `PackYears`, respectively. The black circles represent VEs (ordered by their Bray-Curtis eigenvalues) by residual components, usually referred to as the scree plot.

6 Testing hypotheses about the microbiome (based on relative abundance data) using `ldm` and `permanovaFL`

The LDM simultanuously provides global tests of any effect of the microbiome and OTU-specific tests. For the OTU-specific tests, there exists an upper bound for the number of permutations for the algorithm to stop, which is $J \times L_{\min} \times \alpha^{-1}$, where J is the number of OTUs, $L_{\min}$ is the pre-specified minimum number of rejections (i.e., the permutation statistic exceeded the observed test statistic), and α is the nominal FDR. For example, when $L_{\min} = 100$ (default) and $\alpha = 0.1$ (default), the default upper bound is $1000J$, which is 233000 for this example dataset. If the default value `n.perm.max=NULL` is used then the maximum number of permutations for testing OTUs is determined in this way, while the maximum number of permutations for the global test remains 5000. Conversely, if a numeric value for `n.perm.max` is specified, this value is used for both the global tests and the OTU-specific tests. Further, unless specified otherwise by the user, the LDM automatically performs tests on two data scales: the original frequency (i.e., relative abundance) scale and the arcsin-root-transformed frequency scale. The omnibus test reports the most powerful of the two scales (after properly adjusting for the comparison).

```
> fdr.nominal <- 0.1
```

```
> (seed=sample.int(100000, size=1))   # use seed=67817 to reproduce the results below
[1] 67817
```

```
> res.ldm <- ldm(throat.otu.tab|(Sex+AntibioticUse)~SmokingStatus+PackYears,
                  data=throat.meta, seed=seed)
> res.ldm$n.perm.completed
[1] 128000
> res.ldm$global.tests.stopped
[1] FALSE
> res.ldm$otu.tests.stopped
[1] TRUE
> res.ldm$p.global.omni
           [,1]
[1,] 0.00339932
[2,] 0.70120000
> res.ldm$seed
[1] 67817
> w1 = which(res.ldm$q.otu.omni[1,] < fdr.nominal)
> (n.otu.omni.m1 = length(w1))
[1] 4
> (otu.omni.m1 = colnames(res.ldm$q.otu.omni)[w1])
[1] "2434" "1490" "4703" "3538"
> w2 = which(res.ldm$q.otu.omni[2,] < fdr.nominal)
> (n.otu.omni.m1 = length(w2))
[1] 0
```

With 5000 permutations, the stopping criterion of obtaining at least 100 rejections has not been met by the global test. The OTU tests stopped (met the early stopping criteria) after 128000 permutations. According to the omnibus test results, smoking status (after accounting for the confounders) is significantly associated with the microbiome (p-value = 0.0034) while pack per year after accounting for smoking status (and the confounders) is not (p-value = 0.701). Note that, when the true global p-value is very small, a larger number of permutations may be required to "accurately" estimate it and there is no upper bound for `n.perm.max` to guarantee the stop of the algorithm; if the value of `global.tests.stopped=FALSE`, then the reported p-value may be used as an upper bound. By identifying those OTUs for which the q-value is less than the nominal FDR 0.1, we find four OTUs (with OTU IDs "2434", "'1490", "4703", and "3538") that contribute to the association with smoking status and no OTU for pack per year. These results are consistent with the pattern of significance in the corresponding global tests.

Using the code below, we can provide a summary table for the significant OTUs, which includes the raw p-value, adjusted p-value (by the Benjamini-Hochberg [1995] procedure), mean relative abundance, directions of covariate effects, and OTU name (or taxonomy assignment); the OTUs are ordered by the raw p-values. We first obtain the relative abundance data adjusted for the confounding covariates by using `adjust.data.by.covariates`.

```
> res <- adjust.data.by.covariates(formula= ~ Sex+AntibioticUse,
                                   data=throat.meta, otu.table=throat.otu.tab,
```

```
                                    center.otu.table=FALSE) # suppress centering
                                                            # to retain the original range

# order by the p-values of significant OTUs
> o = order(res.ldm$p.otu.omni[1,w1])
# If no OTU is significant, we can still provide a summary table for the top (e.g., 10) OTUs
# by defining o = order(res1$p.otu.omni[1,]) and replacing w1[o] below by o[1:10]

> mean.freq = colMeans(res$y.freq)
> direction = t(ifelse(res.ldm$beta[1,]>0, "+", "-"))
> name = colnames(throat.otu.tab)
> summary.tab = data.frame(raw.pvalue=signif(res.ldm$p.otu.omni[1,w1[o]],3),
                      adj.pvalue=signif(res.ldm$q.otu.omni[1,w1[o]],3),
                      mean.freq=mean.freq[w1[o]],
                      direction=direction[w1[o],],
                      name=name[w1[o]],
                      row.names = NULL)
> colnames(summary.tab)[4] = paste("direction.", rownames(res.ldm$beta[1,]), sep="")
> summary.tab
  raw.pvalue adj.pvalue  mean.freq direction.SmokingStatusNonSmoker name
1   0.000279     0.0583 0.06375758                                 - 1490
2   0.001000     0.0745 0.03726427                                 - 2434
3   0.001070     0.0745 0.01085440                                 - 3538
4   0.001700     0.0887 0.01292580                                 - 4703
```

For example, `direction.SmokingStatusNonSmoker = '-'` means that non-smokers (as indicated in the column name) have lower relative abundance than smokers, for all four OTUs.

We can also contrast the adjusted relative abundances by smoking status in the box plot, which are displayed in Figure 3.

```
boxplot(res$y.freq[,w1[o[1]]]~throat.meta$SmokingStatus, main="Top 1 OTU",
        ylab="Relative Abundance", xlab="Smoking Status")
boxplot(res$y.freq[,w1[o[2]]]~throat.meta$SmokingStatus, main="Top 2 OTU",
        ylab="", xlab="Smoking Status")
boxplot(res$y.freq[,w1[o[3]]]~throat.meta$SmokingStatus, main="Top 3 OTU",
        ylab="", xlab="Smoking Status")
boxplot(res$y.freq[,w1[o[4]]]~throat.meta$SmokingStatus, main="Top 4 OTU",
        ylab="", xlab="Smoking Status")
```

The `permanovaFL` function implements the PERMANOVA-FL test (our implementation of the PERMANOVA test). Like the PERMANOVA test implemented elsewhere (e.g., the `adonis` or `adonis2` functions in the R package `vegan`), our `permanovaFL` only allows testing of global hypotheses. Here we use the default `n.perm.max=5000`, i.e., 5000 for the maximum number of permutations.

```
> (seed=sample.int(100000, size=1)) # use seed=82955 to reproduce the results below
[1] 82955
> res.perm <- permanovaFL(throat.otu.tab|(Sex+AntibioticUse)~SmokingStatus+PackYears,
                          data=throat.meta, dist.method="bray", seed=seed)
```

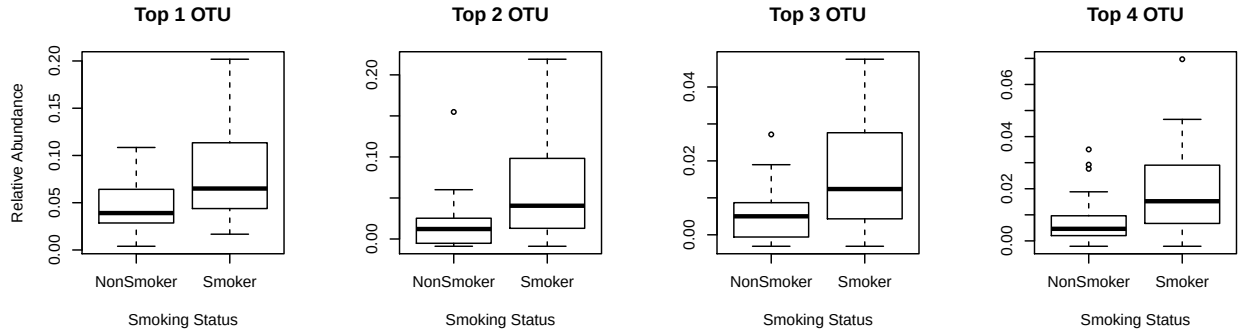

Figure 3 Box plot of relative abundances (adjusted for sex and antibiotic use) by smoking status.

```
> res.perm
$F.statistics
          [,1]
[1,] 2.9354949
[2,] 0.8020703

$p.permanova
            [,1]
[1,] 0.00219956
[2,] 0.64500000

$n.perm.completed
[1] 5000

$permanova.stopped
[1] FALSE

$seed
[1] 82955
```

The global test results are consistent with those from the LDM: smoking status (after accounting for the confounders) is significantly associated with the microbiome (p-value = 0.0022) while pack per year after accounting for smoking status (and the confounders) is not (p-value = 0.645).

7 Testing hypotheses about the microbiome (based on presence-absence data) using `ldm`

The function `ldm` has parameters to allow testing hypotheses about the presence and absence of bacteria. This type of tests are particularly confounded by the library size (total number of reads per sample). A frequently used solution to this confounding effect is *rarefaction*, which corresponds to subsampling reads of all samples to a common rarefaction depth that is often the lowest observed library size (after removing outliers). On the other hand, rarefaction typically results in a substantial

loss of reads and hence statistical power. Therefore, multiple rarefaction has been proposed to recover some of the information lost through rarefaction. However, it remains unclear how to aggregate the information from multiple rarefied datasets, and how many rarefactions recover sufficient information. In Hu, Lane, and Satten (2020), we presented two ways of extending the LDM F-statistic to aggregate information over multiple rarefactions: the LDM-F(R), which uses the average of the F-statistic over R rarefaction replicates as a test statistic, and the LDM-A, which averages the numerator and denominator terms of the F statistic separately and then uses the ratio of these averages as a test statistic. In particular, in LDM-A, the average of the numerator (or denominator) over *all* rarefaction replicates (equivalently, the expectation of the numerator over the rarefaction operation) can be calculated in a closed form, so it is not necessary to perform any actual rarefactions and it essentially uses all data in a non-stochastic manner. We recommend LDM-A over LDM-F(R) because it 1) bypasses the selection on the number of rarefaction replicates R, 2) yielded more power (and more sensitivity for detecting individual taxa) than LDM-F(R) in our simulation studies, and 3) is computationally much more efficient. We have implemented both LDM-A and LDM-F(R) in the `ldm` function. Note that there is only one data scale, which is presence or absence (i.e., binary), so there is not an omnibus test. First, we illustrate the application of the LDM-A method.

```
> res.ldmA <- ldm(formula=throat.otu.tab | (Sex+AntibioticUse) ~ SmokingStatus+PackYears,
                  data=throat.meta, seed=67817,
                  n.rff="all") # parameter for requesting LDM-A
> res.ldmA$p.global.pa
[1] 0.00859828 0.75900000

> w1 = which(res.ldmA$q.otu.pa[1,] < fdr.nominal)
> (n.otu.ldmA.m1 = length(w1))
[1] 3
> (otu.ldmA.m1 = colnames(res.ldmA$q.otu.pa)[w1])
[1] "4363" "411"  "3954"
```

Using the code below, we can provide a summary table for the significant OTUs, which includes the raw p-value, adjusted p-value (by the Benjamini-Hochberg [1995] procedure), mean relative abundance, directions of covariate effects, and OTU name (or taxonomy assignment); the OTUs are ordered by the raw p-values. We first obtain the expected proportion of presence for each OTU in each sample over rarefaction (`phi`), adjusted for the confounding covariates.

```
> phi.adj = t(t(lm(res.ldmA$phi~Sex+AntibioticUse, data=throat.meta)$resid) + colMeans(phi))

> o = order(res.ldmA$p.otu.pa[1,w1])

> prop.presence = colMeans(phi.adj)
> direction = t(ifelse(res.ldmA$beta[1,]>0, "+", "-"))
> name = colnames(throat.otu.tab)
> summary.ldmA.tab = data.frame(raw.pvalue=signif(res.ldmA$p.otu.pa[1,w1[o]],3),
                           adj.pvalue=signif(res.ldmA$q.otu.pa[1,w1[o]],3),
                           prop.presence=prop.presence[w1[o]],
                           direction=direction[w1[o],],
                           name=name[w1[o]],
```

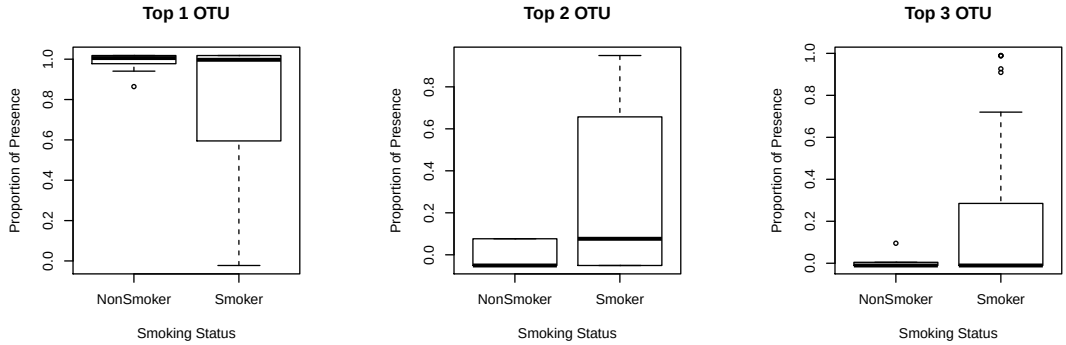

Figure 4 Box plot of proportion of presence (adjusted for sex and antibiotic use) by smoking status.

```
                          row.names = NULL)
> colnames(summary.ldmA.tab)[4] = paste("direction.", rownames(res.ldmA$beta[1,]), sep="")
> summary.ldmA.tab
  raw.pvalue adj.pvalue prop.presence direction.SmokingStatusNonSmoker name
1   0.000719     0.0995     0.8808345                                + 3954
2   0.001370     0.0995     0.1185416                                -  411
3   0.001420     0.0995     0.1004344                                - 4363
```

For example, `direction.SmokingStatusNonSmoker = '+'` means that non-smokers (as indicated in the column name) have higher proportions of presence than smokers for the top 1 OTU "3954".

We can also contrast the adjusted proportions of presence by smoking status in the box plot, which are displayed in Figure 4.

```
for (i in c(1:n.otu.ldmA.m1)) {
    boxplot(phi.adj[,w1[o[i]]] ~ throat.meta$SmokingStatus, main=paste("Top", i, "OTU"),
            ylab="Proportion of Presence", xlab="Smoking Status")
}
```

Next, we illustrate the application of the LDM-F(R) method.

```
> res.ldmF <- ldm(formula=throat.otu.tab | (Sex+AntibioticUse) ~ SmokingStatus+PackYears,
                  data=throat.meta, seed=67817,
                  binary=TRUE, n.rff=5) # parameters for requesting LDM-F(5)
> res.ldmF$p.global.pa # columns:R=1-5; rows: SmokingStatus, PackYears
          [,1]       [,2]       [,3]       [,4]       [,5]
[1,] 0.0079984 0.00839832 0.00739852 0.00739852 0.00739852
[2,] 0.7414000 0.74720000 0.77620000 0.82960000 0.83360000

> w1 = which(res.ldmF$q.otu.pa[1,] < fdr.nominal)
> (n.otu.ldmF.m1 = length(w1))
[1] 0
```

8 Complex (restricted) permutations for correlated samples

The functions `ldm` and `permanovaFL` accommodate a variety of complex (i.e., restricted) permutations through the `permute` package. For unclustered data, `perm.within.type` governs the type of permutation. The default value, `permute.within.type="free"` performs unrestricted permutations, equivalent to using `sample(n.obs, replace=FALSE)`. Other allowed values are "series", "grid" and "none". If `perm.within.type="grid"` is specified, then values of `perm.within.nrow` and `perm.within.ncol` must be specified. For more information on the effect of choosing `perm.within.type` to a value other than "free", see the documentation for the `permute` package. If desired, permutations can also be restricted to groups of observations by specifying `strata=strata.id`, which restricts permutations to those that preserve the value of `strata.id` for each observation.

Clustered data may also be analyzed using `ldm` and `permanovaFL`. Clusters in the data are identified by having a common value of `cluster.id`. There are two variables that control the type of permutations possible with clustered data: `perm.between.type` and `perm.within.type`. The allowable values of `perm.within.type` are as described in the previous paragraph, while the allowable values of `perm.between.type` are "free", "none" and "series". To permute only *within* clusters, choose `perm.between.type="none"` and `perm.within.type="free"`. To permute only *across* clusters, choose `perm.between.type="free"` and `perm.within.type="none"`. To permute *both within and between* clusters, set both to "free". Note that if `perm.between.type` has a value other than "none" then all clusters must have the same size. Permutations of clustered data may be further restricted to preserve the value of `strata.id` by specifying `strata=strata.id`. Note that the value of `strata.id` must be the same for each member of a single cluster. If there are only a few observed cluster sizes, permuting within strata defined by cluster size is a way to analyze unbalanced, clustered data.

Users may also use the `how` function in the `permute` package to manually specify a restricted permutation. This is accomplished by setting `how=how.var` where `how.var` is an object of class `how` as produced by the `permute` package. Any permutation scheme allowed by `permute` can be specified in this way.

8.1 Example dataset 2: simulated data with repeated samples

To illustrate the analysis of repeated samples using `ldm` and `permanovaFL`, we simulated a dataset based on the relative abundances and overdispersion observed in the throat data. It consists of data from 50 individuals, each of whom contributed 2 samples. Fifty individuals were cases (Y.between = 1), and the remaining ones were controls (Y.between = 0). Metadata include the case-control status (Y.between), a confounding covariate (X.between), and a unique individual identifier (ID). We use "between" in variable names to indicate that these variables are between-individual variables. See Hu and Satten (2020) for more details in generating this dataset. We first prepare this dataset:

```
> library(LDM)
> data(sim.otu.tab) # zero columns have been excluded
> data(sim.meta)
>
> otu_presence = which(colSums(sim.otu.tab>0)>=5)    # optionally remove OTUs that occur
> sim.otu.tab = sim.otu.tab[,otu_presence]           # in fewer than 5 individuals
> dim(sim.otu.tab)
```

```
[1] 100 593
```

The OTU table contains 593 OTUs after excluding OTUs having less than 5 presence in the sample. To test the association of the microbiome with the case-control status Y.between while controlling for the confounder X.between, while also accounting for the clustering structure (so that the case-control status is always the same for all observations from the same individual), we specify `cluster.id=ID`, `perm.between.type="free"`, and `perm.within.type="none"` and use the R code:

```
> (seed=sample.int(100000, size=1)) # use seed=34794 to reproduce the results below
[1] 34794
> res.ldm.repeated <- ldm(formula=sim.otu.tab | X.between ~ Y.between,
                          data=sim.meta, seed=34794,
                          cluster.id=ID, perm.within.type="none", perm.between.type="free")
# the last line specifies unique parameters for analyzing repeated samples
> res.ldm.repeated$n.perm.completed
[1] 43100
> res.ldm.repeated$global.tests.stopped
[1] FALSE
> res.ldm.repeated$otu.tests.stopped
[1] TRUE
> res.ldm.repeated$p.global.omni
[1] 0.00019996
> w1 = which(res.ldm.repeated$q.otu.omni[1,] < fdr.nominal)
> (n.otu.omni.m1 = length(w1))
[1] 19
> (otu.omni.m1 = colnames(res.ldm.repeated$q.otu.omni)[w1])
 [1] "1583" "3067" "5522" "5443" "2122" "2334" "2245" "2213" "330"  "2434" "3619" "4036"
[13] "922"  "3954" "3108" "799"  "3957" "4599" "596"
```

With 43100 permutations, the OTU-specific tests have met the stopping criterion and examination of the q-values detects 19 OTUs at the FDR = 0.1 level. The global test has not met the stopping criterion of obtaining at least 100 rejections with 5000 permutations; however, we know the p-value is going to be very small around 0.00019996 and there is no need to add more permutations to make it more precise.

The syntax of using `permanovaFL` for this type of data is very similar to `ldm`, except that the method for calculating the distance matrix (`dist.method`) should be specified:

```
> res.perm.repeated <- permanovaFL(formula=sim.otu.tab | X.between ~ Y.between,
                       data=sim.meta, dist.method="bray", seed=34794,
                       cluster.id=ID, perm.within.type="none", perm.between.type="free")
> res.perm.repeated$p.permanova
           [,1]
[1,] 0.00019996
```

8.2 Example dataset 3: simulated data with matched sets of samples

To illustrate the analysis of matched sets of samples using `ldm` and `permanovaFL`, we added to the dataset in Section 8.1 a within-individual case-control status Y.`within` and a within-individual

covariate X.`within`. The case-control status Y.`within` is always 0 for one sample and 1 for the other sample from the same individual. To test the association of the microbiome with the case-control status Y.`within` while controlling for the potential confounder X.`within`, while also accounting for the clustering structure (so that the case-control status is always different for the two samples from the same individual), we specify `cluster.id=ID`, `perm.between.type="none"`, and `perm.within.type="free"`. As described in Zhu et al. (2020), the indicator variables for individuals should be adjusted as covariates to constrain the test within individuals.

```
> res.ldm.matched <- ldm(formula=sim.otu.tab | as.factor(ID) + X.within ~ Y.within,
                         data=sim.meta, seed=34794,
                         cluster.id=ID, perm.within.type="free", perm.between.type="none")
# the last line specifies unique parameters for analyzing matched-set samples
> res.ldm.matched$n.perm.completed
[1] 70200
> res.ldm.matched$global.tests.stopped
[1] TRUE
> res.ldm.matched$otu.tests.stopped
[1] TRUE
> res.ldm.matched$p.global.omni
      [,1]
[1,] 0.435
> w1 = which(res.ldm.matched$q.otu.omni[1,] < fdr.nominal)
> (n.otu.omni.var1 = length(w1))
[1] 0
```

With 70200 permutations, both the global test and OTU-specific tests have met the stopping criterion. The global test is non-significant with p-value 0.435 and the OTU-specific tests detect 0 significant OTUs.

Again, the syntax of using `permanovaFL` for this type of data is very similar to `ldm`, except that the method for calculating the distance matrix (`dist.method`) should be specified:

```
> res.perm.matched <- permanovaFL(formula=sim.otu.tab | as.factor(ID) + X.within ~ Y.within,
                 data=sim.meta, dist.method="bray", seed=34794,
                 cluster.id=ID, perm.within.type="free", perm.between.type="none")
> res.perm.matched$p.permanova
          [,1]
[1,] 0.3215434
```

References

Hu YJ, Satten GA (2020). Testing hypotheses about the microbiome using the linear decomposition model (LDM). *Bioinformatics*, 36(14):4106–4115.

Hu YJ, Lane A, Satten GA (2020). A Rarefaction-Based Extension of the LDM for Testing Presence-Absence Associations in the Microbiome. *bioRXiv*, doi.org/10.1101/2020.05.26.117879.

Zhu Z, Satten GA, Caroline M, and Hu YJ (2020). Analyzing matched sets of microbiome data using the LDM and PERMANOVA. *bioRxiv*, 2020.03.06.980367; doi: https://doi.org/10.1101/2020.03.06.980367.

Satten GA et al. (2017) Restoring the duality between principal components of a distance matrix and linear combinations of predictors, with application to studies of the microbiome. *PLoS One*, 12, e0168131.

Besag J, Clifford P (1991). Sequential Monte Carlo p-values. *Biometrika*, 78(2):301–304.

Benjamini Y and Hochberg Y (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society*, Series B (Methodological), 289–300.

Sandve GK, Ferkingstad E, Nygard S (2011). Sequential Monte Carlo multiple testing. *Bioinformatics*, 27(23):3235–3241.

Freedman D, Lane D (1983). A nonstochastic interpretation of reported significance levels. *Journal of Business & Economic Statistics*, 1(4):292–298.