16s Microbiome Analysis with phyloseq and LDM

Jessica Randall

Last compiled 02 October, 2020

Briefly, phyloseq takes in data from data processing programs like QIIME, mothur, and Pyrotagger. While QIIME2 offers richness estimates and other exploratory data analysis (ex: alpha and beta diversity metrics) we believe that phyloseq in combination with ggplot2 offers greater flexibility for generating customizable data visualizations. To see a few examples of graphs we've generated using phyloseq, check out our data visualization menu here.

Once exploratory data analysis in phyloseq is complete, we use the LDM package from Hu and Satten to perform statistical analyses. LDM takes in a table of operational taxonomic units (OTUs) or amplicon sequence variants (ASVs) along with a table of data about the samples (i.e. covariates) and uses a linear decomposition model to associate experimental conditions and covariates of interest with microbial abundance. Data about the samples typically includes sample names, some experimental condition of interest, and other variables as collected by the experimenters. LDM can accommodate both continuous and categorical data.

There are many ways to import data into phyloseq. We typically import the .qza files produced by QIIME2 but in this walk-through we will be using a built-in dataset that you can use anytime so you can follow along with this walk-through if desired. For more examples to import data using other programs, see the phyloseq vignette here Example data comes from the Global Patterns dataset is described in PNAS (Caporoso, 2011). This dataset compares the microbial communities of 25 environmental samples and three known "mock communities"-a total of 9 sample types- at 3.1 million reads/sample. It is natively available within phyloseq.

Definition of terms

0.0.0.1 Linear Decomposition Model

The LDM models microbial abundance in the form of counts transformed into relative abundances as an outcome of interest given experimental covariates of interest. LDM provides users with both global and local hypothesis tests of differential abundance given covariates of interest and microbial count data. LDM decomposes the model sum of squares into parts explained by each variable in the model. From these sub-models we can see the amount of variability that each variable is contributing to the overall variability explained by the model's covariates of interest.

0.0.0.2 False Discovery Rate (FDR)

The FDR tells you how likely it is that all taxa identified as differentially abundant (DA) are false positives. A FDR of 5% means that among all taxa called DA, an average of 5% of those are truly not DA. The q-value is the local significance threshold adjusted for the fact that we have assessed multiple taxa. Accurate interpretation of unadjusted p-values assumes that each taxa is assessed for DA on its own. However, most, if not all 16s microbiome experiments assess multiple taxa for differential abundance at once. In order to account for the number of taxa we are testing, we must calculate and interpret the adjusted p-value for each taxa.

Our very first step is to load the packages we need from Bioconductor. Please see Bioconductor for information about installation and use of Bioconductor and its packages.

Loading R packages and data

We will be using the Global Patterns dataset for this walk through but typically we would use the .qza files we obtain from QIIME2 and the qza_to_phyloseq function in phyloseq to import the data as a phyloseq object.

```
pacman::p_load(
  "knitr", "devtools", "phyloseq", "tidyverse", "qiime2R",
  "reshape2", "xml2", "GUniFrac", "vegan", "LDM",
  "ggfortify", "janitor", "tinytex"
)

data("GlobalPatterns")
```

Prepare data

In order to improve statistical power to detect differentially abundant taxa we typically remove taxa which have counts of 0 in at least one (or more) samples.

With this dataset we would also like to define a categorical variable for sample type which will tell us if the sample is from a human or not since this is a covariate of interest to us and we expect it to account for a large percentage of the variability between the samples. This variable will be project specific.

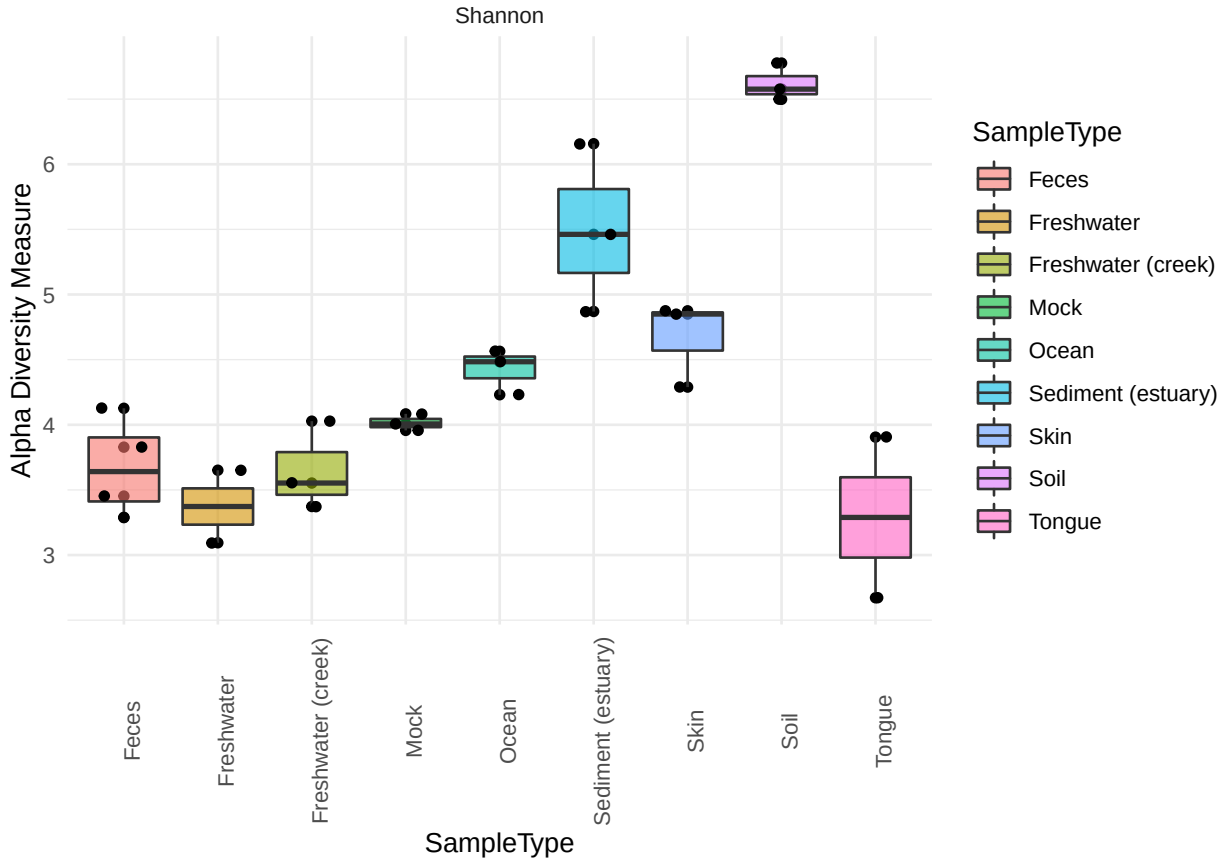
```
GP <- prune_taxa(taxa_sums(GlobalPatterns) > 0, GlobalPatterns)

human <- get_variable(GP, "SampleType") %in% c(
  "Feces", "Mock", "Skin",
  "Tongue"
)

sample_data(GP)$human <- factor(human)

sample_data(GP)$human <- ifelse(human == TRUE, "Human", "Not-Human")
```

Exploratory Data Analysis with phyloseq

Richness estimation

0.0.0.2.1 Alpha Diversity

To compare richness estimates within samples we typically provide the Shannon metric but phyloseq also offers the Chao1, ACE, Simpson, or Inverse Simpson metrics. While this dataset does not include control samples we would suggest including Zymo controls, Positive Controls, and Negative Controls in your sequencing along with your experimental samples.



Each box groups the sample types by whether they come from a human or not. Each sample type is represented by a different color and we see that among the non-human samples, soil has the highest alpha diversity score and Freshwater samples have the lowest. Among the human samples, skin samples have the highest alpha diversity and tongue samples have the lowest alpha diversity.

Alpha diversity scores tell us how similar in microbial composition each individual sample is to other samples of the same sample type or other grouping variable of interest, or the within-sample variability. We would expect experimental samples to have scores like those we see here, between 2-7. This makes sense given the amount of biological diversity between individuals even of the same sample type. It seems reasonable that the microbial composition of skin cells would be more different from one skin cell to the next than the microbial composition of one tongue cell to the next.

For control samples, we would expect them to have much lower alpha diversity scores than the experimental samples, closer to zero, because we expect that all of the samples in each control type are extremely similar to each other and more similar to one another than to other sample types.

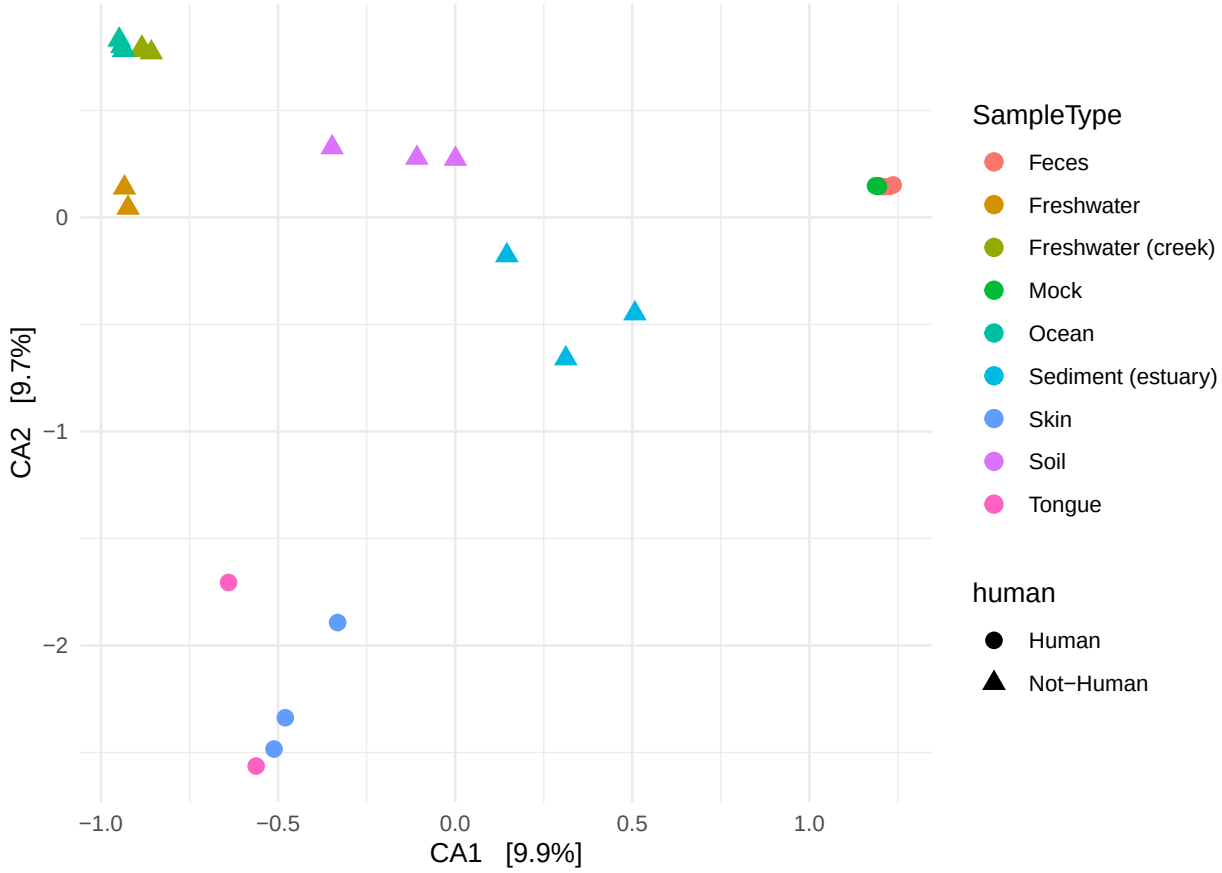
Correspondence Analysis

0.0.0.2.2 Beta Diversity

Similarly to a Principal Components Analysis a Correspondence Analysis tells us how much variability is being contributed to the data set by each axis (like a principal component). In the case of this data, we only want to look at the top 200 most represented taxa in the top 5 Phyla so first we have to subset the data to reflect that. These are parameters which will vary based on your research question.

There are many metrics for assessing beta diversity and phyloseq includes options for Unifrac distance, Jaccard, Manhattan, Euclidean, or Chao1 metrics among others. Beta Diversity tells us how similar in microbial composition each sample is to other samples of different types, or between sample variability.

The amount of variability we would expect to see between sample types will vary based on your specific project. In general we do expect that control samples will be very different in their microbial composition than the experimental samples.

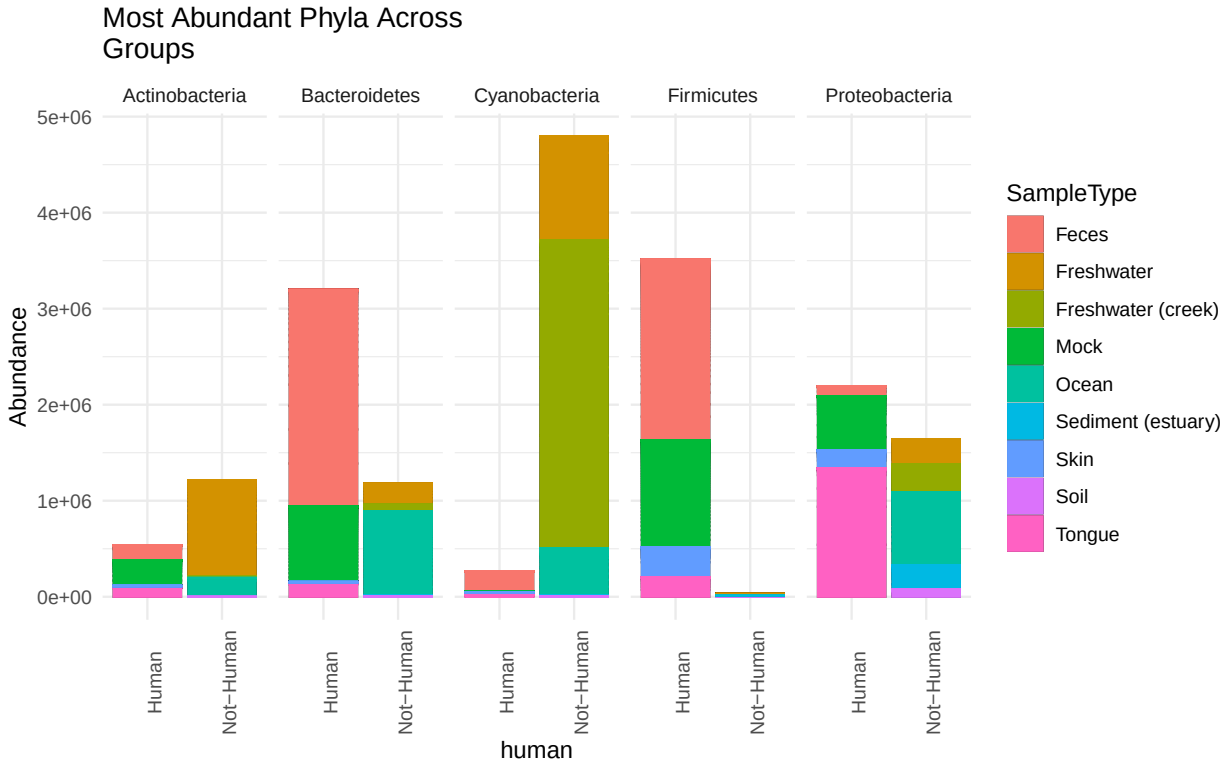


Looking at the first and second CAs we see that the Ocean and Freshwater (creek) samples and the Tongue and Skin samples appear to be more similar to one another than they are different. The Freshwater, Soil, and Sediment (estuary) samples are each more similar to other samples of the same type than samples of other types since they appear to cluster relatively closely by sample type. Finally we note that the Mock community samples and the Feces samples cluster very closely.

Looking at the top half of the graph compared to the bottom half of the graph it suggests that the most important covariate in explaining the differences between these samples is whether or not they come from human samples.This seems biologically reasonable since we would expect samples from humans would be more similar to one another than samples from waterways or sediment.

Phyla-specific abundance plot

Next, we want to see which particular phyla differ between human and non-human samples. We can do this with a bar plot with abundance graphed along the y-axis and whether the samples are human or not along the x-axis.



Here we see that Cyanobacteria and Actinobacteria are most abundant in the non-human samples and the Bacteriodetes and Firmicutes are most abundant in the human samples.

There are many ways to visualize data in phyloseq, these are just a few examples of the typical products of our analytical pipeline at EICC. We would be happy to customize graphs according to your project-specific goals. Check out some additional examples from previous projects here

Statistical Analysis with LDM

We need to recover the data from the phyloseq object and transform the taxa table, asvs, and the metadata objects from phyloseq data objects to data frames. We will use the janitor library's clean_names function to put all column names into lower case by default and remove any characters R cannot easily work with. For example, the variable "X.Sample.ID" will be converted to "x_sample_id" and since the sample IDs in the asv table are all lower case, we have converted those in the metadata to be all lowercase as well. Next, since the psmelt function treats each taxa as its own row and has multiple rows with the same sample, and we're only interested in one row per sample, we reverse engineer the sample_data information originally in the GP_LDM phyloseq object. This will make it easier for use with LDM later on.

```
GP_LDM <- prune_taxa(taxa_sums(GlobalPatterns) > 0, GlobalPatterns)

meta <- as.data.frame(psmelt(GP_LDM)) %>%
  clean_names(case = "snake") %>%
  select(
    x_sample_id, sample_type, description,
    primer, final_barcode, barcode_truncated_plus_t, barcode_full_length
  ) %>%
  mutate(sample_id = as.factor(tolower(x_sample_id))) %>%
  group_by(x_sample_id) %>%
  slice(1)

asvs <- as.data.frame(otu_table(GP_LDM)) %>%
  clean_names()

asvs_t <- as.data.frame(t(asvs)) %>%
  mutate(sample_id = as.factor(levels(as.factor(meta$x_sample_id))))

row.names(asvs_t) <- asvs_t$sample_id
```

Since the ASVs have the taxa as rows and we need them as columns, we can use the t function to transpose the columns into rows and rows into columns. Note that at this point we have 26 samples and 18989 taxa.

Next, to remove any statistical noise we may still have to detect relationships between the covariates and microbial composition, we decide to keep only those taxa which appear in at least 5 samples. This is a parameter that will vary by project.

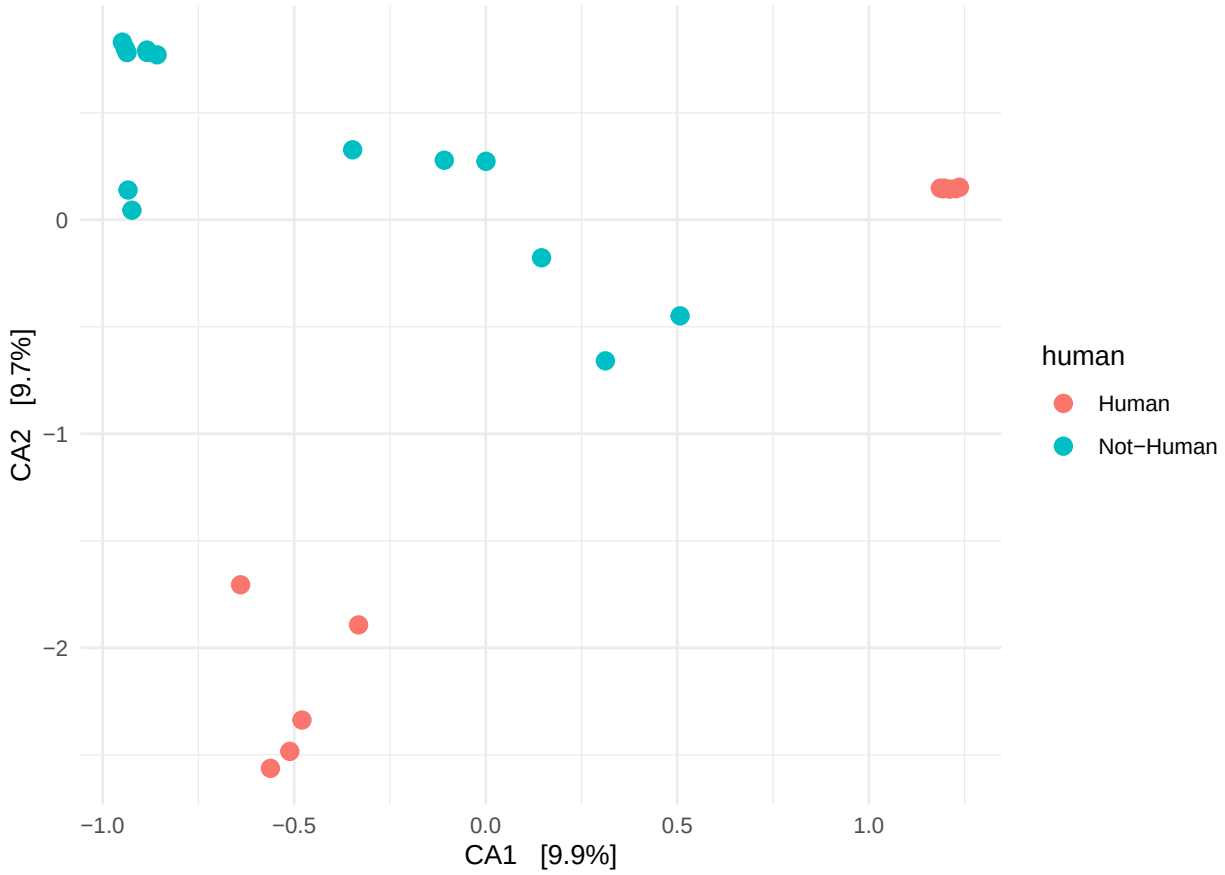
```
otu_pres <- which(colSums(asvs_t[1:18988] > 0) >= 5)

asvs_filt <- asvs_t[, otu_pres]
```

At this point we have 26 samples and 8367 taxa.

Our exploratory data analysis has suggested that a sample being either from a human or not is an important determination of its microbial composition. Let's test whether or not this relationship is statistically significant. In some cases this variable might be considered a confounder but in this case, since we want to examine the relationship between human and non-human samples and microbial composition, it will be the only covariate of interest. LDM can handle covariates both categorical and continuous and can control for confounders.

Correspondence Analysis

Above, we colored samples by their type and used different shapes to indicate whether they were from humans or not. For the purposes of this example we are only interested in assessing the impact of whether or not a sample comes from a human so let's look at a PCoA or Beta Diversity plot with samples colored by this variable.



By coloring the samples by whether the come from humans or not we see that they do appear to have some substantial separation based on this variable. We should investigate further to determine whether or not this apparent separation results in statistically significantly different microbial communities between human and non-human samples.

To examine this relationship we first specify our model, called by stating we would like to assess whether or not a sample being from a human significantly impacts its microbiome composition.

Fit the LDM

```
form <- asvs_filt ~ human


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

Global Hypothesis Testing

In order to determine whether a sample being from a human or non-human source is statistically significantly contributing to observed differences in microbial composition we first perform a test of the global hypothesis. We want to know, overall, are there differences between human and non-human samples with regard to the composition of the ASVs expressed in each group?

Since we are performing permutations, we set a seed to be able to reproduce our work each time we run the model. This is key to reproducibility and must be specified. You can use R's built-in seed function or you can pick a number you like as long as you use the same one every time you need a seed in this analysis and don't mind sharing it.

Now we fit our model, specifying that we only want to do a global test with test.global=TRUE and test.otu=FALSE. We also specify that we want to use the Bray Curtis distance but LDM offers many options to customize this parameter.

```
seed <- 22310

fit2 <- ldm(
  formula = form,
  data = meta,
  dist.method = "bray",
  test.global = TRUE,
  test.otu = FALSE,
  seed = seed
)
```

```
## permutations: 1
## permutations: 1001
## permutations: 2001
## permutations: 3001
## permutations: 4001
```

```
(global_p <- fit2$p.global.omni)
```

```
## [1] 0.00019996
```

ASV-specific Hypothesis Testing

Since this global hypothesis test has a global p-value of 1.9996001×10^{-4} it appears to suggest that there are statistically significant differences in the microbiome compositions of human and non-human samples. However, since we tested 8367 ASVs with each one representing an individual hypothesis we must correct for with a multiple testing correction. Here we use the Benjamini-Hochberg False Discovery Rate to control the FDR at 0.05.

Let's look more closely at which ASVs are responsible for this. Note that in this model we fit we specify test.otu=TRUE and the FDR is controlled at 0.05. We also specify the same seed as above for reproducibility of results. The FDR specified will vary by project. More exploratory or pilot studies may wish to have a higher FDR where more specific studies may wish to have a smaller one.

```
fit3 <- ldm(
  formula = form,
  data = meta,
  dist.method = "bray",
  test.global = TRUE,
  test.otu = TRUE,
  fdr.nominal = 0.05,
  seed = seed
)
```

```
## permutations: 1
## permutations: 1001
## permutations: 2001
## permutations: 3001
## otu test stopped at permutation 3400
## permutations: 4001
```

After fitting our model, we can find out how many and which ASVs and associated taxa are significantly differently abundant between human and non-human samples. Here we create a dataframe of results ordered by smallest local FDR.

```
##         Kingdom         Phylum           Class              Order
## 245969 Bacteria Actinobacteria  Actinobacteria   Acidimicrobiales
## 23235  Bacteria Actinobacteria  Actinobacteria    Actinomycetales
## 12567  Bacteria Actinobacteria  Actinobacteria    Actinomycetales
## 471185 Bacteria Actinobacteria  Actinobacteria  Bifidobacteriales
## 542934 Bacteria    Tenericutes Erysipelotrichi Erysipelotrichales
##                  Genus                 Species        pvals      qvals
## 245969            <NA>                    <NA> 0.0011761247 0.01659461
## 23235      Actinomyces                    <NA> 0.0008820935 0.01659461
## 12567      Actinomyces                    <NA> 0.0011761247 0.01659461
## 471185 Bifidobacterium Bifidobacteriumanimalis 0.0002940312 0.01659461
## 542934     Clostridium                    <NA> 0.0011761247 0.01659461
```

The most significantly abundant bacteria may or may not be the most clinically relevant and this will vary by project. Note that due to the fidelity of the reference microbiome we use (GreenGenes, Silva, or Human Oral Microbiome), some levels of a given taxa may not be available. Some taxa may have genus and species level information and some may not. Additionally, p-values and q-values are from the global omnibus tests, p and q values based on frequency scale data or arcsin-root transformed frequency data are also available.

We look forward to working with you to customize this pipeline to your experimental design and parameters of interest. For a more specific application and interpretation of these or additional tools and visualizations

for your own data, please find our contact information on our website and examples of some previous graphs here

Session information

```
## [1] "Fri Oct  2 11:53:12 2020"
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Debian GNU/Linux 10 (buster)
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/libopenblasp-r0.3.5.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] tinytex_0.24       janitor_2.0.1      ggfortify_0.4.10   LDM_1.0
##  [5] GUniFrac_1.1       matrixStats_0.56.0 ape_5.4            vegan_2.5-6
##  [9] lattice_0.20-41    permute_0.9-5      xml2_1.3.2         reshape2_1.4.4
## [13] qiime2R_0.99.34    forcats_0.5.0      stringr_1.4.0      dplyr_1.0.0
## [17] purrr_0.3.4        readr_1.3.1        tidyr_1.1.0        tibble_3.0.3
## [21] ggplot2_3.3.2      tidyverse_1.3.0    phyloseq_1.27.6    devtools_2.3.0
## [25] usethis_1.6.1      knitr_1.29
##
## loaded via a namespace (and not attached):
##   [1] colorspace_1.4-1    ellipsis_0.3.1      rprojroot_1.3-2
##   [4] snakecase_0.11.0    htmlTable_2.0.1     XVector_0.28.0
##   [7] base64enc_0.1-3     fs_1.4.2            rstudioapi_0.11
##  [10] farver_2.0.3        remotes_2.1.1       DT_0.14
##  [13] fansi_0.4.1         lubridate_1.7.9     codetools_0.2-16
##  [16] splines_4.0.2       pkgload_1.1.0       ade4_1.7-15
##  [19] Formula_1.2-3       jsonlite_1.7.0      broom_0.7.0
##  [22] cluster_2.1.0       dbplyr_1.4.4        png_0.1-7
##  [25] compiler_4.0.2      httr_1.4.2          backports_1.1.9
##  [28] assertthat_0.2.1    Matrix_1.2-18       cli_2.0.2
##  [31] acepack_1.4.1       htmltools_0.5.0     prettyunits_1.1.1
##  [34] tools_4.0.2         igraph_1.2.5        gtable_0.3.0
##  [37] glue_1.4.1          Rcpp_1.0.5          Biobase_2.48.0
##  [40] cellranger_1.1.0    zCompositions_1.3.4 vctrs_0.3.2
##  [43] Biostrings_2.56.0   multtest_2.44.0     nlme_3.1-148
##  [46] iterators_1.0.12    xfun_0.15           ps_1.3.4
##  [49] testthat_2.3.2      rvest_0.3.5         lifecycle_0.2.0
```

```
##  [52] pacman_0.5.1        zlibbioc_1.34.0     MASS_7.3-51.6
##  [55] scales_1.1.1        hms_0.5.3           parallel_4.0.2
##  [58] biomformat_1.16.0   rhdf5_2.32.2        RColorBrewer_1.1-2
##  [61] yaml_2.2.1          NADA_1.6-1.1        gridExtra_2.3
##  [64] memoise_1.1.0       rpart_4.1-15        latticeExtra_0.6-29
##  [67] stringi_1.4.6       S4Vectors_0.26.1    desc_1.2.0
##  [70] foreach_1.5.0       checkmate_2.0.0     BiocGenerics_0.34.0
##  [73] pkgbuild_1.1.0      truncnorm_1.0-8     rlang_0.4.7
##  [76] pkgconfig_2.0.3     evaluate_0.14       Rhdf5lib_1.10.1
##  [79] labeling_0.3        htmlwidgets_1.5.1   processx_3.4.3
##  [82] tidyselect_1.1.0    plyr_1.8.6          magrittr_1.5
##  [85] bookdown_0.20       R6_2.4.1            IRanges_2.22.2
##  [88] generics_0.0.2      Hmisc_4.4-0         DBI_1.1.0
##  [91] foreign_0.8-80      pillar_1.4.6        haven_2.3.1
##  [94] withr_2.2.0         mgcv_1.8-31         nnet_7.3-14
##  [97] survival_3.2-3      modelr_0.1.8        crayon_1.3.4
## [100] rmarkdown_2.3       jpeg_0.1-8.1        grid_4.0.2
## [103] readxl_1.3.1        data.table_1.13.0   blob_1.2.1
## [106] callr_3.4.3         reprex_0.3.0        digest_0.6.25
## [109] stats4_4.0.2        munsell_0.5.0       sessioninfo_1.1.1
```