Microbiome: 16S Data Visualization/Analysis

Hardik I Parikh

This markdown outlines instructions for visualization and analysis of OTU-clustered amplicon sequencing data, primarily using the *phyloseq* package.

Prerequisites

- R basics
- Data manipulation with dplyr and %>%
- Data visualization with ggplot2

R packages

CRAN packages

- tidyverse (readr, dplyr, ggplot2)
- magrittr
- reshape2
- vegan
- ape
- ggpubr
- RColorBrewer

Bioconductor packages

- phyloseq
- DESeq2

Required Data Files

We will use the output files generated during the sequence processing steps.

- OTU table (abundance table) : `relman2017_samples.otu_table.txt`
- Taxonomy table : `relman2017_samples.tax_table.txt`
- Sample metadata : `relman2017_samples.sample_data.txt`
- OTU phylogenetic tree : `relman2017_samples.rep_set.tre`

Load data

Let's read in the data files using `read_tsv()` function of **readr** package

```
dataDir = "../../static/data/"
# import OTU table
otu <- read_tsv(paste(dataDir, "relman2017_samples.otu_table.txt", sep="/"))
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   OTUId = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
otu
```

```
## # A tibble: 1,151 x 101
##    OTUId SRR5944960 SRR5944971 SRR5944984 SRR5944985 SRR5944986 SRR5944990
##    <chr>      <int>      <int>      <int>      <int>      <int>      <int>
##  1 OTU_~      10050          5        110        527          2        102
##  2 OTU_1      48542      51618         22          3          8         13
##  3 OTU_~       9091          4         47        123         11         32
##  4 OTU_~       4330          0          1          0          0          0
##  5 OTU_~       7501          0          1          1          2          1
##  6 OTU_~       6794          0          0          0          2          0
##  7 OTU_~       1572          0          0          0          0          0
##  8 OTU_~      10450         14       1120        647         58        564
##  9 OTU_~       4077          0        992        608         55         40
## 10 OTU_~        514          0          0          0          2          0
## # ... with 1,141 more rows, and 94 more variables: SRR5944998 <int>,
## #   SRR5945004 <int>, SRR5945006 <int>, SRR5945007 <int>,
## #   SRR5945019 <int>, SRR5945036 <int>, SRR5945037 <int>,
## #   SRR5945040 <int>, SRR5945041 <int>, SRR5945042 <int>,
## #   SRR5945062 <int>, SRR5945067 <int>, SRR5945072 <int>,
## #   SRR5945075 <int>, SRR5945093 <int>, SRR5945100 <int>,
## #   SRR5945101 <int>, SRR5945103 <int>, SRR5945121 <int>,
## #   SRR5945137 <int>, SRR5945141 <int>, SRR5970594 <int>,
## #   SRR5970602 <int>, SRR5970613 <int>, SRR5970633 <int>,
## #   SRR5970696 <int>, SRR5970727 <int>, SRR5970786 <int>,
## #   SRR5970843 <int>, SRR5970952 <int>, SRR5971020 <int>,
## #   SRR5971077 <int>, SRR5971094 <int>, SRR5971121 <int>,
## #   SRR5971134 <int>, SRR5971142 <int>, SRR5971178 <int>,
## #   SRR5971244 <int>, SRR5971252 <int>, SRR5971253 <int>,
## #   SRR5971296 <int>, SRR5971310 <int>, SRR5971321 <int>,
## #   SRR5971331 <int>, SRR5971332 <int>, SRR5971345 <int>,
## #   SRR5971351 <int>, SRR5971359 <int>, SRR5971376 <int>,
## #   SRR5971381 <int>, SRR5971392 <int>, SRR5971402 <int>,
## #   SRR5971406 <int>, SRR5971407 <int>, SRR5971412 <int>,
## #   SRR5971417 <int>, SRR5971495 <int>, SRR5971651 <int>,
## #   SRR5971741 <int>, SRR5971848 <int>, SRR5971925 <int>,
## #   SRR5971970 <int>, SRR5972033 <int>, SRR5972053 <int>,
## #   SRR5972132 <int>, SRR5972155 <int>, SRR5972183 <int>,
## #   SRR5972195 <int>, SRR5972207 <int>, SRR5972216 <int>,
## #   SRR5972221 <int>, SRR5972236 <int>, SRR5972239 <int>,
## #   SRR5972242 <int>, SRR5972243 <int>, SRR5972254 <int>,
## #   SRR5972258 <int>, SRR5972274 <int>, SRR5972280 <int>,
## #   SRR5972285 <int>, SRR5972290 <int>, SRR5972301 <int>,
## #   SRR5972305 <int>, SRR5972316 <int>, SRR5972337 <int>,
## #   SRR5972364 <int>, SRR5972387 <int>, SRR5972400 <int>,
## #   SRR5972417 <int>, SRR5972503 <int>, SRR5972529 <int>,
## #   SRR5972633 <int>, SRR5972683 <int>, SRR5972686 <int>
```

```
# import Taxonomy table
taxonomy <- read_tsv(paste(dataDir, "relman2017_samples.tax_table.txt", sep="/"))
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
```

```
##   X1 = col_character(),
##   domain = col_character(),
##   phylum = col_character(),
##   class = col_character(),
##   order = col_character(),
##   family = col_character(),
##   genus = col_character()
## )
```

```
taxonomy
```

```
## # A tibble: 1,225 x 7
##    X1     domain   phylum         class          order    family    genus
##    <chr>  <chr>    <chr>          <chr>          <chr>    <chr>     <chr>
##  1 OTU_1  Bacteria Firmicutes     Bacilli        Lactoba~ Lactobac~ Lacto~
##  2 OTU_2  Bacteria Firmicutes     Bacilli        Lactoba~ Lactobac~ Lacto~
##  3 OTU_3  Bacteria Actinobacteria Actinobacteria Bifidob~ Bifidoba~ Gardn~
##  4 OTU_4  Bacteria Firmicutes     Bacilli        Lactoba~ Lactobac~ Lacto~
##  5 OTU_5  Bacteria Firmicutes     Bacilli        Lactoba~ Lactobac~ Lacto~
##  6 OTU_6  Bacteria Firmicutes     Bacilli        Lactoba~ Lactobac~ Lacto~
##  7 OTU_7  Bacteria Actinobacteria Actinobacteria Bifidob~ Bifidoba~ Gardn~
##  8 OTU_8  Bacteria Firmicutes     Bacilli        Lactoba~ Lactobac~ Lacto~
##  9 OTU_9  Bacteria Firmicutes     Negativicutes  Selenom~ Veillone~ Megas~
## 10 OTU_10 Bacteria Firmicutes     Bacilli        Lactoba~ Lactobac~ Lacto~
## # ... with 1,215 more rows
```

```
# import Sample metadata
metadata <- read_tsv(paste(dataDir, "relman2017_samples.sample_data.txt", sep="/"))
```

```
## Parsed with column specification:
## cols(
##   sample = col_character(),
##   age = col_integer(),
##   gest_day_collection = col_integer(),
##   indication_for_PTB = col_character(),
##   race = col_character(),
##   term_vs_preterm_delivery = col_character()
## )
```

```
metadata
```

```
## # A tibble: 100 x 6
##    sample    age gest_day_collect~ indication_for_~ race  term_vs_preterm~
##    <chr>   <int>             <int> <chr>            <chr> <chr>
##  1 SRR594~    38                45 PPROM            White Preterm
##  2 SRR597~    35                77 not_applicable   Asian Term
##  3 SRR594~    41                79 not_applicable   White Term
##  4 SRR597~    31                83 not_applicable   White Term
##  5 SRR594~    32                91 Other            White Preterm
##  6 SRR597~    35                92 not_applicable   Asian Term
##  7 SRR594~    32                98 Other            White Preterm
##  8 SRR597~    31               101 not_applicable   White Term
##  9 SRR594~    32               107 PPROM            White Preterm
## 10 SRR597~    31               116 not_applicable   White Term
## # ... with 90 more rows
```

```
# import phylogenetic tree
phytree <- read.tree(paste(dataDir, "relman2017_samples.rep_set.tre", sep="/"))
phytree
```

```
##
## Phylogenetic tree with 1224 tips and 1222 internal nodes.
##
## Tip labels:
##  OTU_206, OTU_823, OTU_1169, OTU_1146, OTU_701, OTU_1067, ...
## Node labels:
##  , 0.919, 0.825, 0.874, 0.771, 0.761, ...
##
## Unrooted; includes branch lengths.
```

Create phyloseq object

Read More: Importing data into phyloseq

```
# create OTU table
otu <- otu %>%
  as.data.frame() %>%
  column_to_rownames("OTUId")
OTU <- otu_table(otu, taxa_are_rows = TRUE)

# create TAX table
taxonomy <- taxonomy %>%
  as.data.frame() %>%
  column_to_rownames("X1") %>%
  as.matrix()
TAX <- tax_table(taxonomy)

# create sample metadata
metadata <- metadata %>%
  as.data.frame() %>%
  mutate_if(sapply(metadata, is.character), as.factor) %>%
  column_to_rownames("sample")
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
SDATA <- sample_data(metadata)

# create the phyloseq object
physeq <- phyloseq(OTU, TAX, SDATA, phytree)
```

Data pruning

Read More: (Pre)Processing Data
Phyloseq package provides a plethora of functions for filtering, subsetting and merging abundance data. It is beyond the scope of this workshop to discuss their usage in detail and downstream implications of chosen threshold. I strongly recommend referring the *phyloseq* vignette for answers!
For this tutorial, we will remove taxa that have read count less than 100 in at least 10% of samples. This protects againsts OTUs with small mean and trivially large C.V.
Do not use these measures to filter your dataset!

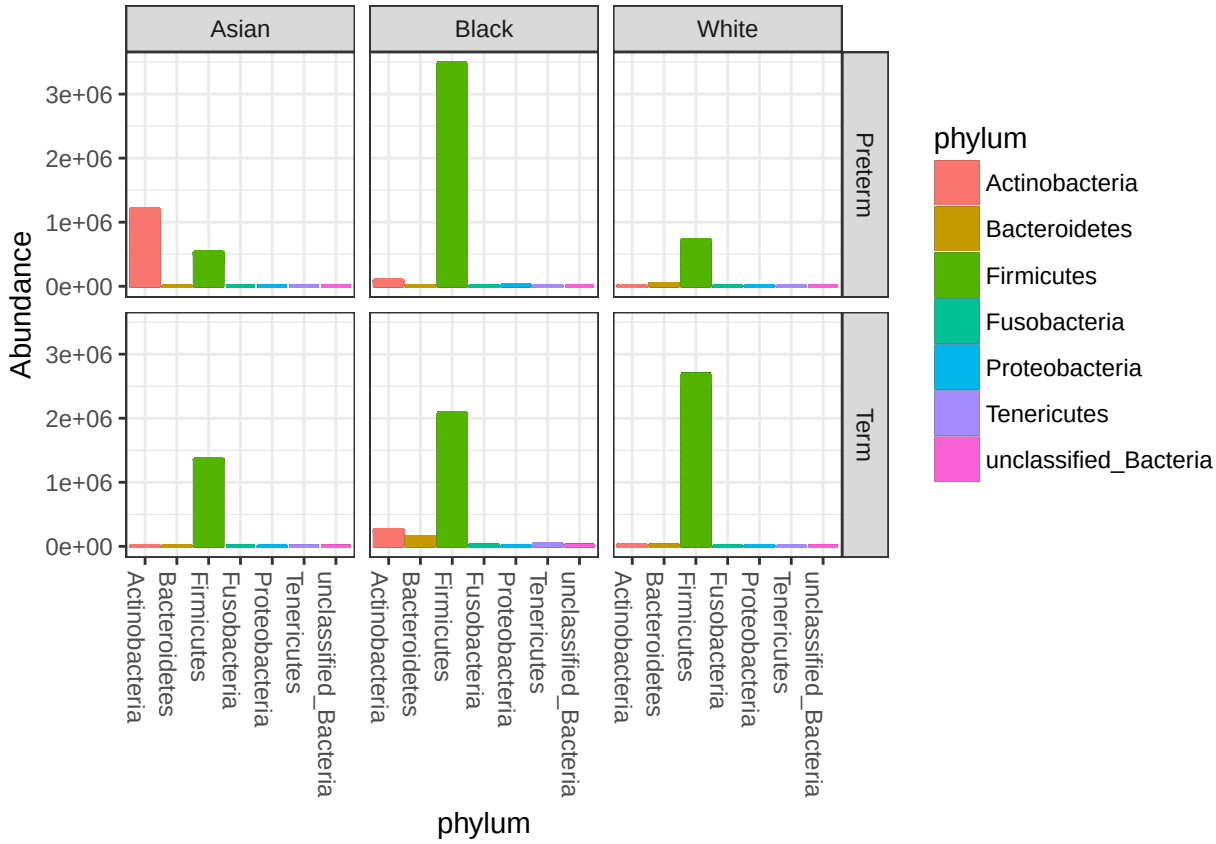
```
# For this tutorial, we will only look into high-abundance/high-prevelance OTUs
physeq.f <- filter_taxa(physeq,
                        function(x) sum(x >= 100) > (0.01*length(x)),
                        prune = TRUE)
```

Stacked Bars

Read More: Phyloseq bar plots

Let's plot phylum-level abundances using the `plot_bar()` function of *phyloseq*

```
plot_bar(physeq.f, "phylum", fill="phylum", facet_grid = term_vs_preterm_delivery~race) +
  geom_bar(aes(color=phylum, fill=phylum), stat="identity", position = "stack")
```




```
# Go over the phyloseq tutorials to explore additional features
```

Next, we will plot stacked bars by clustering samples on their bray-curtis dissimilarities.

```
# transform counts to relative abundance
physeq.f.ra <- transform_sample_counts(physeq.f, function(x) x*100/sum(x))

# agglomerate counts at genus-level
physeq.f.ra.genus <- tax_glom(physeq.f.ra, "genus")

# next get the otu_table
propData <- as.data.frame(t(otu_table(physeq.f.ra.genus)))

# get melted dataframe
```

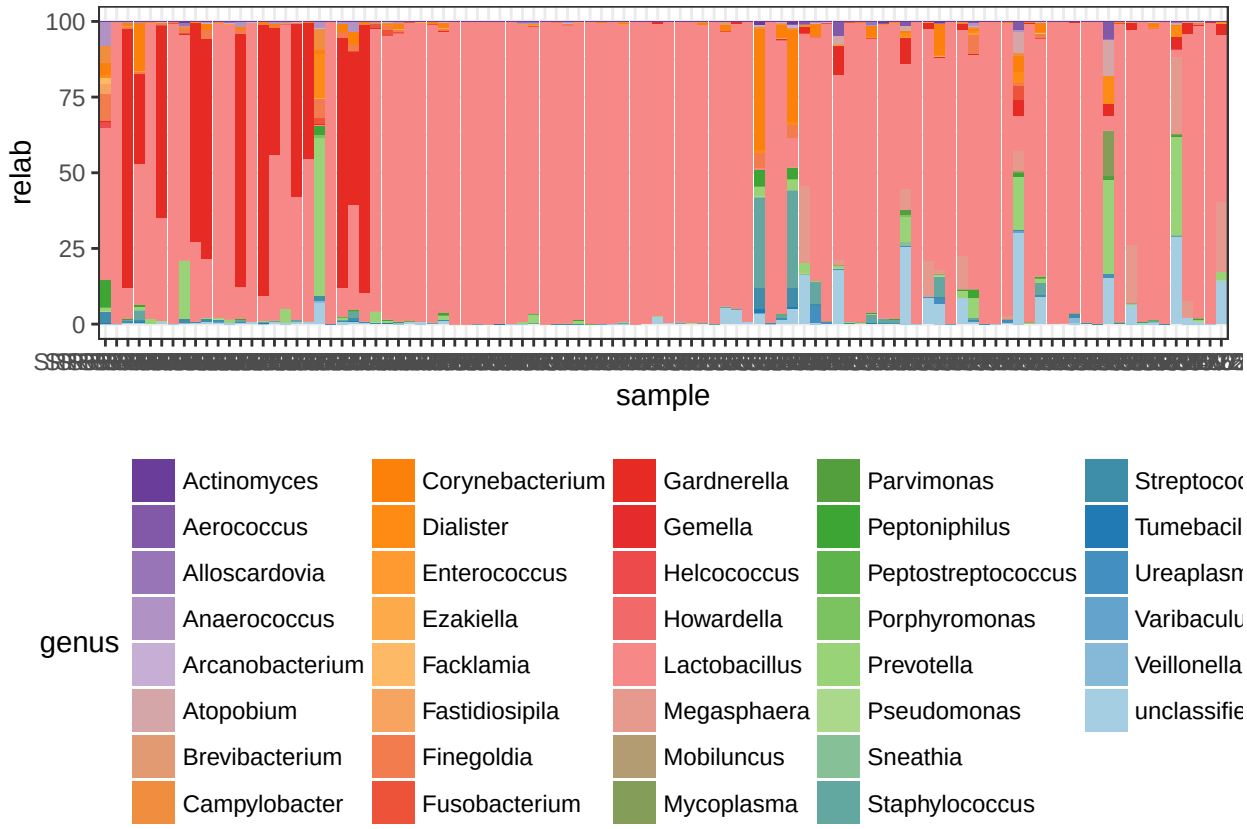
```
plotDF <- propData %>%
  rownames_to_column(var="sample") %>%
  melt() %>%
  magrittr::set_names(c("sample", "taxa", "relab"))
```

```
## Using sample as id variables
```

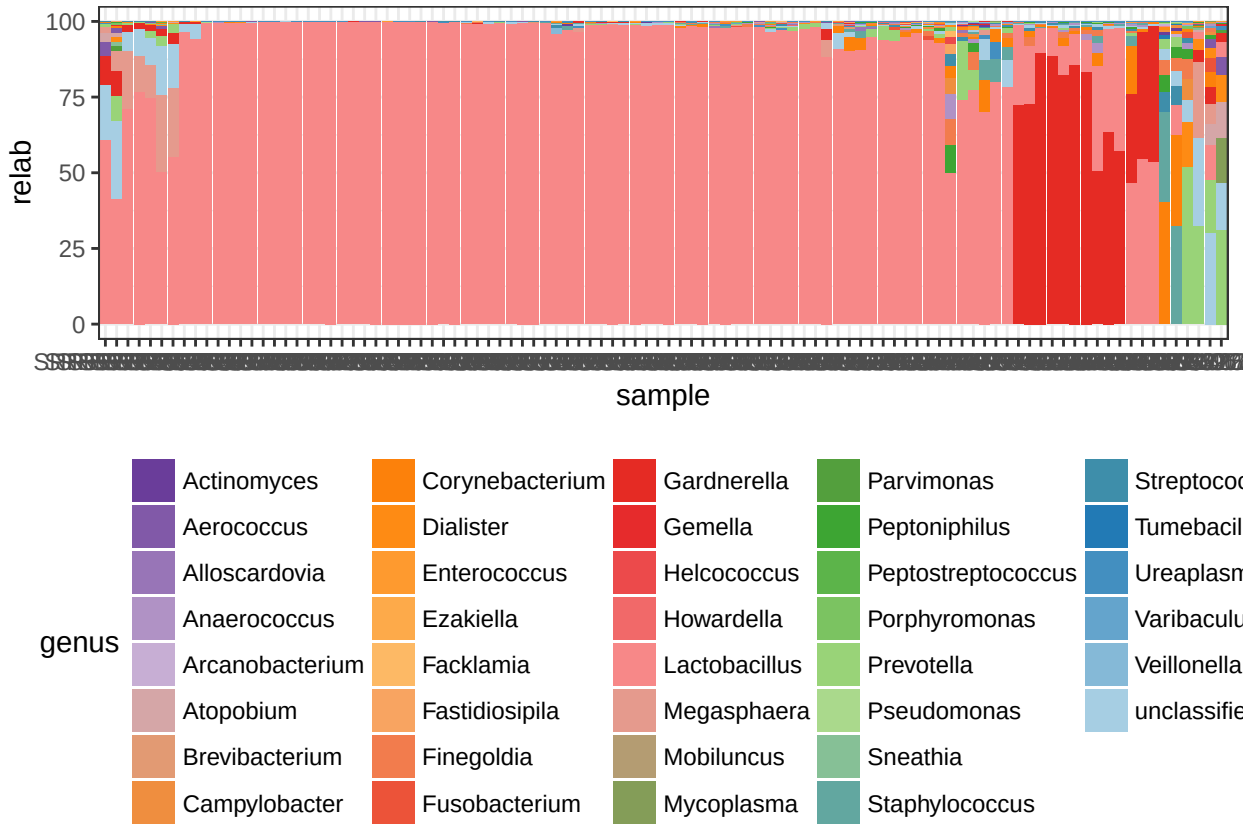
```
# add taxonomy to plotting DF
taxonomy.2 <- taxonomy %>%
  as.data.frame() %>%
  rownames_to_column(var="taxa")
plotDF <- left_join(plotDF, taxonomy.2, by="taxa") %>%
  select(sample, taxa, relab, genus)
```

```
## Warning: Column `taxa` joining factor and character vector, coercing into
## character vector
```

```
# aggregate all unclassified taxa levels
tot_unc <- plotDF %>%
  group_by(sample) %>%
  filter(grepl("unclassified", genus)) %>%
  summarise(relab=sum(relab)) %>%
  mutate(taxa = "unclassified", genus = "unclassified")

# add the unclassified aggregated rel ab to plotting data frame
plotDF <- plotDF %>%
  filter(!grepl("unclassified", genus)) %>%
  rbind(tot_unc)

# add metadata to plotting DF
metadata.2 <- metadata %>%
  rownames_to_column(var="sample")
plotDF <- left_join(plotDF, metadata.2, by="sample")

# lets plot stacked bars using ggplot2
mycolors <- rev(colorRampPalette(brewer.pal(10, "Paired"))(length(unique(plotDF$genus))))
ggplot(plotDF, aes(sample, relab, fill=genus)) +
  geom_bar(stat="identity", position = "stack") +
  scale_fill_manual(values = mycolors) +
  theme(legend.position = "bottom")
```




```
# sort samples on bray-curtis distances
bcdist <- vegdist(propData, method="bray")
hclustBC <- hclust(bcdist, method="ward.D2")
# set sample factor levels
plotDF$sample <- factor(plotDF$sample, levels = hclustBC$labels[c(hclustBC$order)])

# plot again
ggplot(plotDF, aes(sample, relab, fill=genus, group=relab)) +
  geom_bar(stat="identity", position = "stack") +
  scale_fill_manual(values = mycolors) +
  theme(legend.position = "bottom")
```



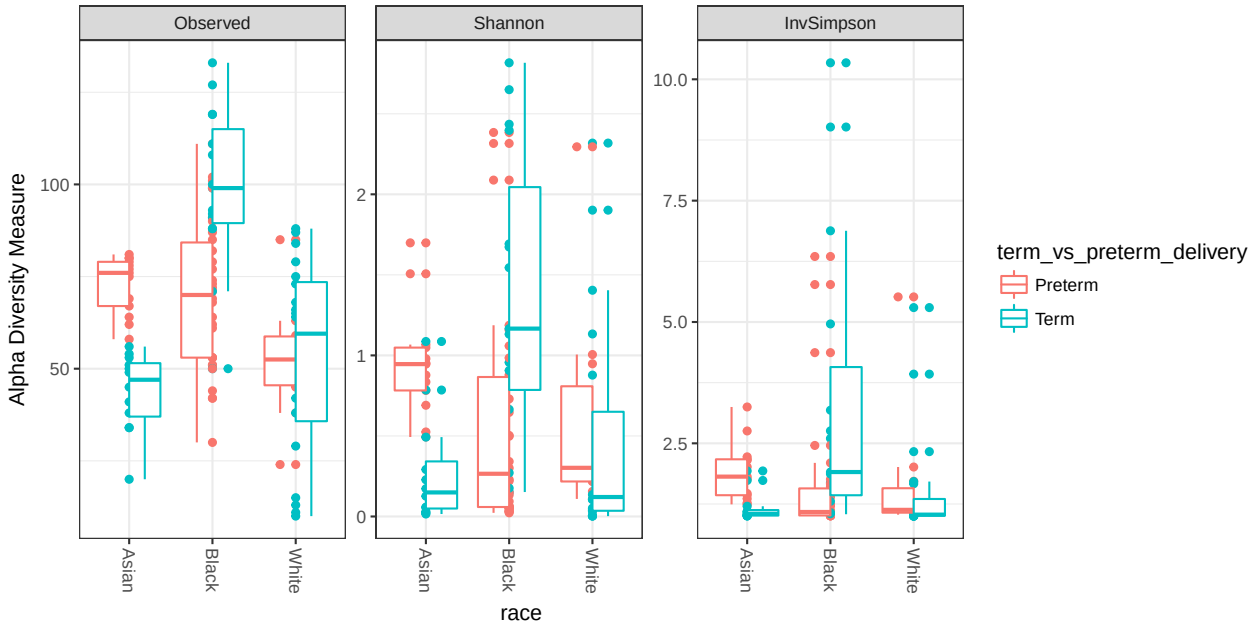
Alpha Diversity

Read More: Phyloseq Alpha diversity

Alpha diversity refers to community *richness*, i.e. how many different *types* of organisms are present, and *evenness*, i.e. how even/uneven the distribution of species abundance is, in a sample. Alpha-diversity analyses are useful for examining patterns of dominance, rarity and community complexity.

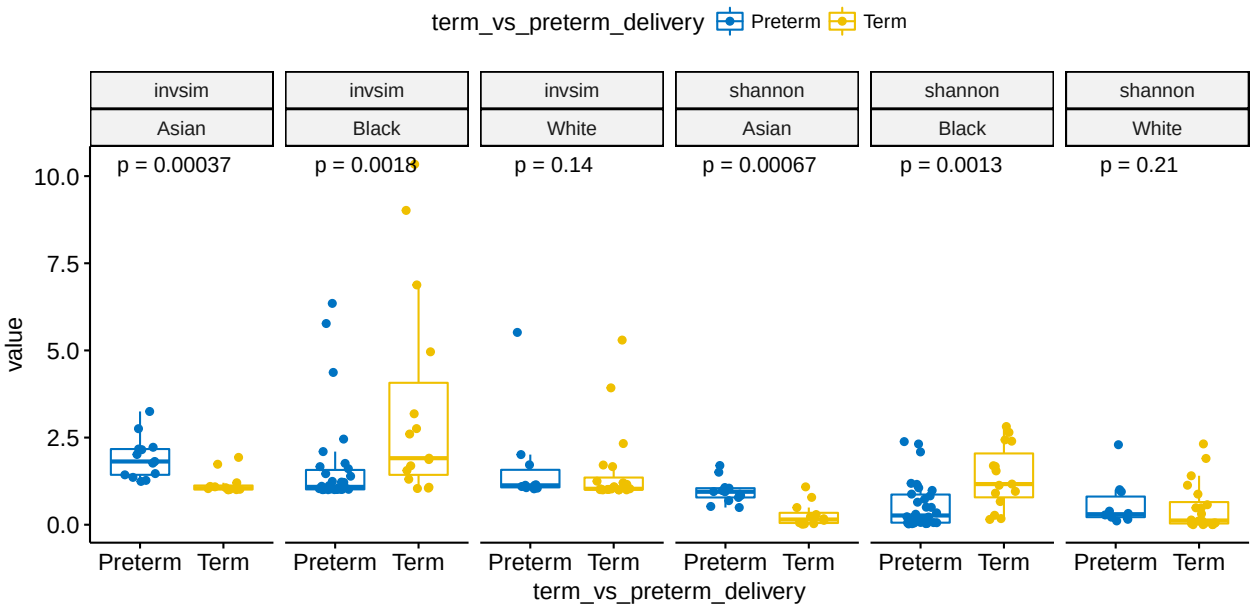
Plot alpha diversity using the `plot_richness()` function

```
# alpha diversity is measured on count data in phyloseq
plot_richness(physeq.f,
              x = "race",
              color = "term_vs_preterm_delivery",
              measures = c("Observed", "Shannon", "InvSimpson")) +
  geom_boxplot()
```



In this section, we will calculate the same measures of alpha diversity using the *vegan* package, followed by comparing means within groups using *ggpubr* package

```
# calculating alpha diversity on proportional data
# R package: vegan

propData <- as.data.frame(t(otu_table(physeq.f.ra)))

# calculate shannon index, manipulate data frame for plotting
adivDF.shannon <- as.data.frame(diversity(propData, index = "shannon")) %>%
  rownames_to_column(var="sample") %>%
  magrittr::set_names(c("sample", "value")) %>%
  mutate(index = "shannon")

# calculate invsim index, manipulate data frame for plotting
adivDF.invsim <- as.data.frame(diversity(propData, index = "invsim")) %>%
  rownames_to_column(var="sample") %>%
  magrittr::set_names(c("sample", "value")) %>%
  mutate(index = "invsim")

# create plotting dataframe
adivDF <- rbind(adivDF.shannon, adivDF.invsim) %>%
  left_join(metadata.2, by="sample")

# plot using ggpubr
ggboxplot(adivDF, "term_vs_preterm_delivery", "value",
          color = "term_vs_preterm_delivery", palette = "jco",
          add = "jitter", outlier.shape=NA) +
  facet_grid(~index+race) +
  stat_compare_means(label = "p.format")
```



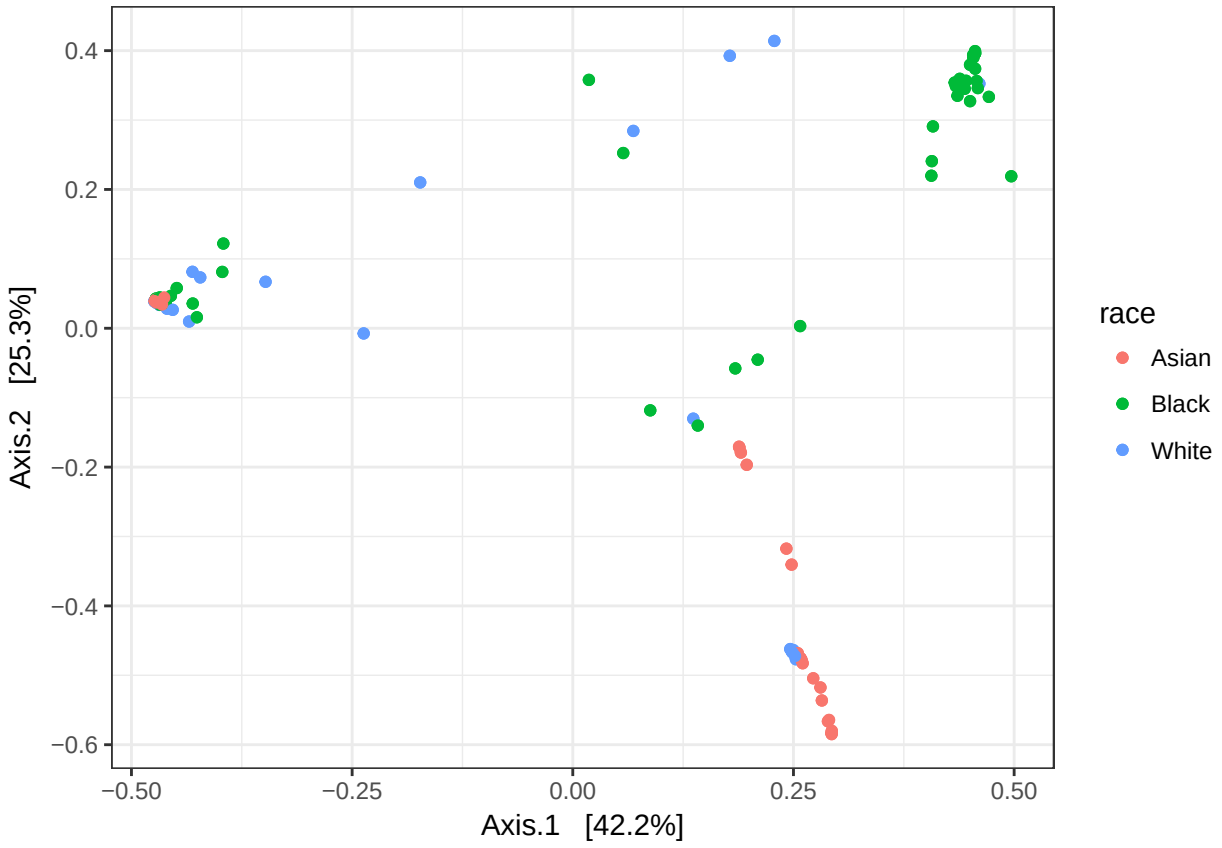
Beta Diversity, Ordinations

Read More: Ordination Plots

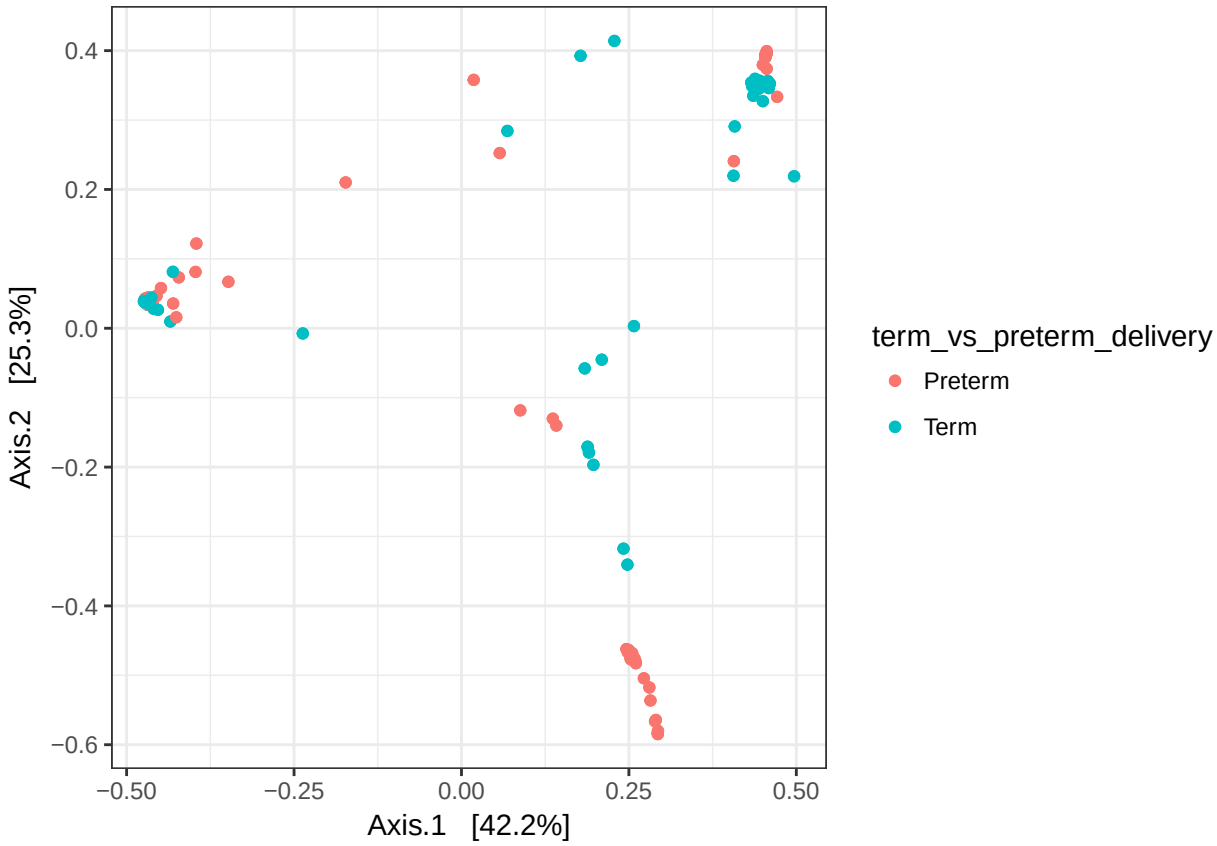
Beta diversity refers to between sample diversity, summerize how similar/dissimilar two samples are. The taxa abundances in each sample gets compared to every other sample in the dataset, generating a distance matrix, which can be visualized using Principal Coordinate Analysis.

Bray-Curtis distances

```
# first calculate bray-curtis distance
dist.mat <- t(data.frame(otu_table(physeq.f)))
physeq.f.distBC <- vegdist(dist.mat, method="bray")
physeq.f.distBC.ord <- ordinate(physeq.f, method = "PCoA", distance = physeq.f.distBC)

# color by metadata
plot_ordination(physeq.f, physeq.f.distBC.ord, color="race") + geom_point()
```

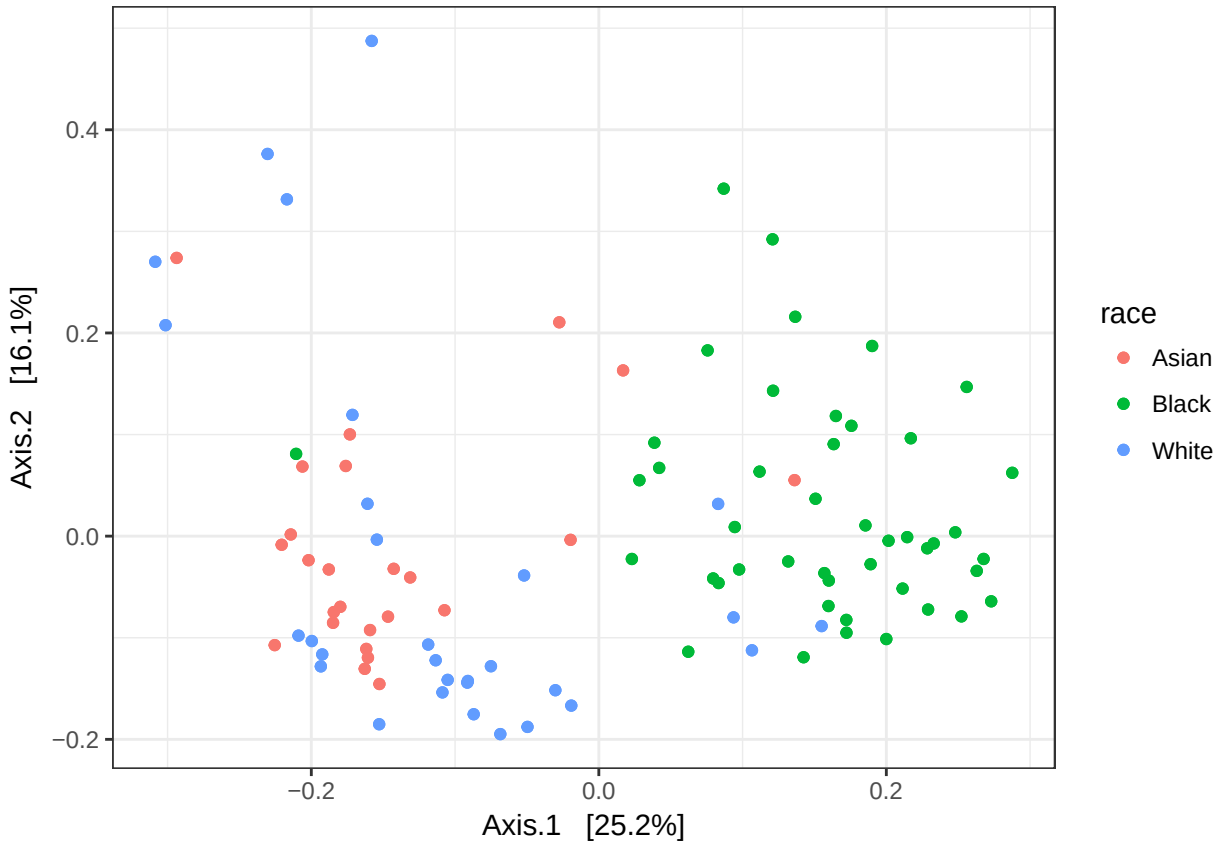



```
plot_ordination(physeq.f, physeq.f.distBC.ord, color="term_vs_preterm_delivery") + geom_point()
```

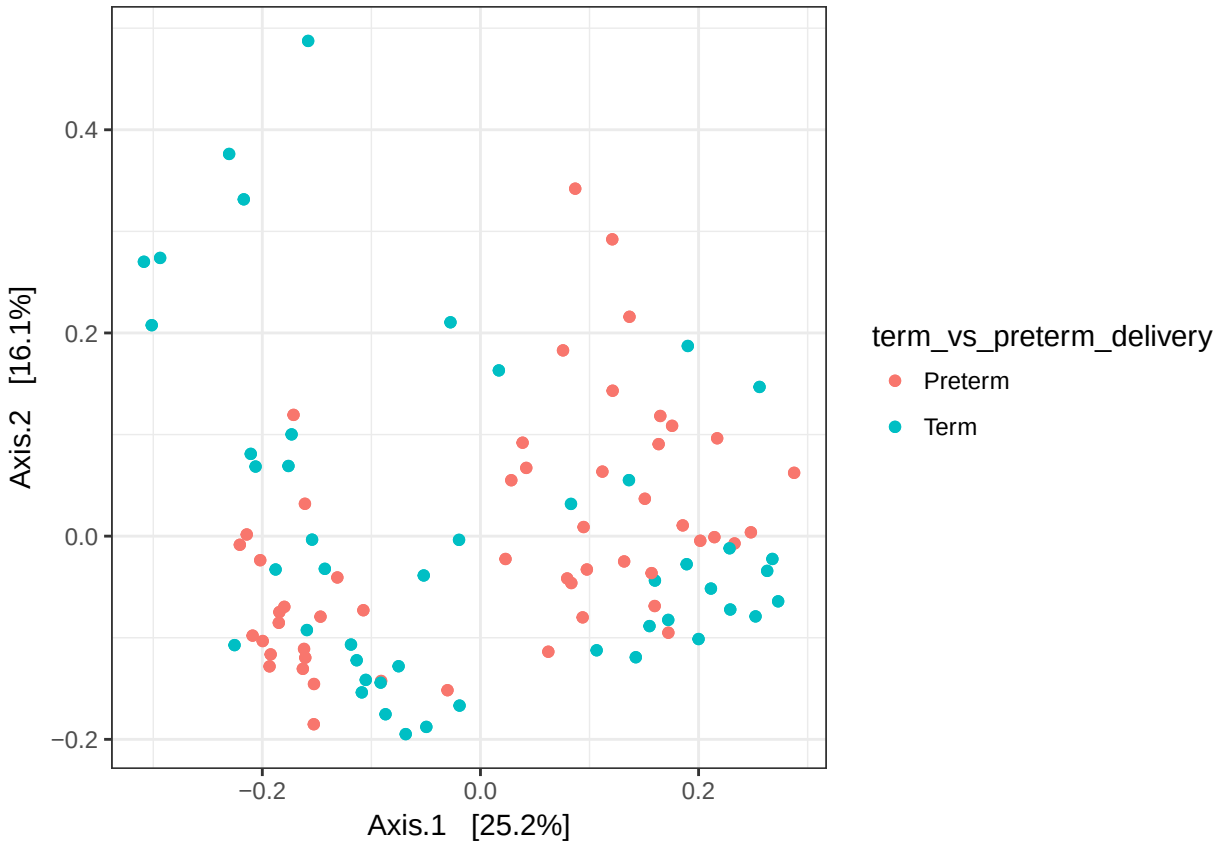


UniFrac distances

```
# first calculate unifrac distance
physeq.f.distUF <- phyloseq::distance(physeq.f, method="unifrac")
physeq.f.distUF.ord <- ordinate(physeq.f, method = "PCoA", distance = physeq.f.distUF)

# color by metadata
plot_ordination(physeq.f, physeq.f.distUF.ord, color="race") + geom_point()
```




```
plot_ordination(physeq.f, physeq.f.distUF.ord, color="term_vs_preterm_delivery") + geom_point()
```



DESeq2

Which taxa are important?
The ordination plots reveal microbiome levels shifts within groups-of-interest. A univariate analysis of individual taxa abudances can be performed using DESeq2, to measure *significant fold changes*.

`phyloseq_to_deseq2()` function provides a convenient function to convert `phyloseq` object to `DESeq2DataSet` class.

```
library(DESeq2)
```

```
## Warning: package 'DESeq2' was built under R version 3.4.2
## Loading required package: S4Vectors
## Warning: package 'S4Vectors' was built under R version 3.4.2
## Loading required package: stats4
## Loading required package: BiocGenerics
## Warning: package 'BiocGenerics' was built under R version 3.4.2
## Loading required package: parallel
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
```

```
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, cbind, colMeans,
##     colnames, colSums, do.call, duplicated, eval, evalq, Filter,
##     Find, get, grep, grepl, intersect, is.unsorted, lapply,
##     lengths, Map, mapply, match, mget, order, paste, pmax,
##     pmax.int, pmin, pmin.int, Position, rank, rbind, Reduce,
##     rowMeans, rownames, rowSums, sapply, setdiff, sort, table,
##     tapply, union, unique, unsplit, which, which.max, which.min

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:dplyr':
##
##     first, rename

## The following object is masked from 'package:tidyr':
##
##     expand

## The following object is masked from 'package:base':
##
##     expand.grid

## Loading required package: IRanges

## Warning: package 'IRanges' was built under R version 3.4.2

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:phyloseq':
##
##     distance

## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

## The following object is masked from 'package:purrr':
##
##     reduce

## Loading required package: GenomicRanges

## Warning: package 'GenomicRanges' was built under R version 3.4.3

## Loading required package: GenomeInfoDb

## Warning: package 'GenomeInfoDb' was built under R version 3.4.2
```

```
## Loading required package: SummarizedExperiment
## Warning: package 'SummarizedExperiment' was built under R version 3.4.3
## Loading required package: Biobase
## Warning: package 'Biobase' was built under R version 3.4.2
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
##
## Attaching package: 'Biobase'
## The following object is masked from 'package:phyloseq':
##
##     sampleNames
## Loading required package: DelayedArray
## Warning: package 'DelayedArray' was built under R version 3.4.2
## Loading required package: matrixStats
##
## Attaching package: 'matrixStats'
## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians
## The following object is masked from 'package:dplyr':
##
##     count
##
## Attaching package: 'DelayedArray'
## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges
## The following object is masked from 'package:base':
##
##     apply
```

```
# convert to DESeq2's data class
diagdds <- phyloseq_to_deseq2(physeq.f, ~ term_vs_preterm_delivery)
```

```
## converting counts to integer mode
```

```
# perform the testing
diagdds <- DESeq(diagdds, test="Wald", fitType="parametric")
```

```
## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
```

```
## -- note: fitType='parametric', but the dispersion trend was not well captured by the
##    function: y = a/x + b, and a local regression fit was automatically substituted.
##    specify fitType='local' or 'mean' to avoid this message next time.

## final dispersion estimates

## fitting model and testing

## -- replacing outliers and refitting for 112 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing
```

```
# look at the results
res <- results(diagdds, cooksCutoff = FALSE)
sigtab <- res[which(res$padj < 0.01), ]
sigtab <- cbind(as(sigtab, "data.frame"), as(tax_table(physeq.f)[rownames(sigtab), ], "matrix"))
head(sigtab, n = 20)
```

```
##             baseMean log2FoldChange     lfcSE       stat       pvalue
## OTU_111    1767.51244     -11.982176 1.5800953  -7.583198 3.371385e-14
## OTU_303    1178.45592     -10.742514 1.5008538  -7.157602 8.210050e-13
## OTU_5    258736.22001      -4.154370 0.9431855  -4.404616 1.059716e-05
## OTU_302     138.38654      -6.686948 1.2958589  -5.160244 2.466285e-07
## OTU_541     196.99919     -15.118544 1.2568358 -12.029053 2.500124e-33
## OTU_91     1354.41348     -14.692651 1.3425713 -10.943665 7.125963e-28
## OTU_1175    125.14917       6.460472 0.9576249   6.746349 1.516112e-11
## OTU_48     1026.28441      19.719049 2.9039429   6.790440 1.117924e-11
## OTU_468     223.81014      -8.622400 1.4992114  -5.751291 8.856463e-09
## OTU_104    1365.99562     -10.256643 0.9675319 -10.600832 2.953403e-26
## OTU_38       78.50563       3.883498 1.2676009   3.063660 2.186474e-03
## OTU_45      799.85893      -2.843298 0.8708630  -3.264920 1.094950e-03
## OTU_313     172.22250     -12.221676 1.3187338  -9.267735 1.901399e-20
## OTU_338      79.38121      -4.557586 1.1232006  -4.057677 4.956318e-05
## OTU_190    1357.30018      -5.415318 0.9902232  -5.468786 4.531297e-08
## OTU_58     1831.12441      -3.460724 0.9652698  -3.585240 3.367685e-04
## OTU_619     160.35637     -13.037735 1.4797737  -8.810628 1.244468e-18
## OTU_83       62.37140      -5.395945 1.3548102  -3.982805 6.810651e-05
## OTU_941     186.98325     -12.453022 1.1432983 -10.892189 1.255844e-27
## OTU_131     187.67197      -9.537642 1.0042962  -9.496842 2.163520e-21
##                  padj   domain                phylum                  class
## OTU_111  4.015014e-13 Bacteria            Firmicutes               Bacilli
## OTU_303  8.273204e-12 Bacteria            Firmicutes               Bacilli
## OTU_5    5.845259e-05 Bacteria            Firmicutes               Bacilli
## OTU_302  1.615417e-06 Bacteria            Firmicutes               Bacilli
## OTU_541  1.091721e-31 Bacteria unclassified_Bacteria unclassified_Bacteria
## OTU_91   2.333753e-26 Bacteria            Firmicutes               Bacilli
## OTU_1175 1.324071e-10 Bacteria            Firmicutes               Bacilli
## OTU_48   1.046058e-10 Bacteria            Firmicutes               Bacilli
## OTU_468  6.824686e-08 Bacteria            Firmicutes             Clostridia
## OTU_104  6.448262e-25 Bacteria        Actinobacteria        Actinobacteria
## OTU_38   8.183658e-03 Bacteria        Actinobacteria        Actinobacteria
## OTU_45   4.218779e-03 Bacteria        Actinobacteria        Actinobacteria
## OTU_313  2.767592e-19 Bacteria        Actinobacteria        Actinobacteria
```

```
## OTU_338  2.404732e-04 Bacteria        Actinobacteria        Actinobacteria
## OTU_190  3.297777e-07 Bacteria        Actinobacteria        Actinobacteria
## OTU_58   1.423119e-03 Bacteria        Actinobacteria        Actinobacteria
## OTU_619  1.630253e-17 Bacteria        Actinobacteria        Actinobacteria
## OTU_83   3.076535e-04 Bacteria        Actinobacteria        Actinobacteria
## OTU_941  3.290310e-26 Bacteria        Actinobacteria        Actinobacteria
## OTU_131  3.542764e-20 Bacteria        Actinobacteria        Actinobacteria
##                         order                           family
## OTU_111        Lactobacillales                Lactobacillaceae
## OTU_303        Lactobacillales                Lactobacillaceae
## OTU_5          Lactobacillales                Lactobacillaceae
## OTU_302        Lactobacillales                Lactobacillaceae
## OTU_541  unclassified_Bacteria           unclassified_Bacteria
## OTU_91         Lactobacillales    unclassified_Lactobacillales
## OTU_1175       Lactobacillales                Lactobacillaceae
## OTU_48         Lactobacillales                Lactobacillaceae
## OTU_468          Clostridiales Clostridiales_Incertae Sedis XI
## OTU_104      Bifidobacteriales              Bifidobacteriaceae
## OTU_38         Actinomycetales                  Micrococcaceae
## OTU_45         Actinomycetales    unclassified_Actinomycetales
## OTU_313        Actinomycetales              Corynebacteriaceae
## OTU_338        Actinomycetales              Corynebacteriaceae
## OTU_190        Actinomycetales              Corynebacteriaceae
## OTU_58         Actinomycetales              Corynebacteriaceae
## OTU_619        Actinomycetales    unclassified_Actinomycetales
## OTU_83         Actinomycetales               Brevibacteriaceae
## OTU_941      Bifidobacteriales              Bifidobacteriaceae
## OTU_131      Bifidobacteriales              Bifidobacteriaceae
##                                genus
## OTU_111                 Lactobacillus
## OTU_303                 Lactobacillus
## OTU_5                   Lactobacillus
## OTU_302                 Lactobacillus
## OTU_541         unclassified_Bacteria
## OTU_91   unclassified_Lactobacillales
## OTU_1175                Lactobacillus
## OTU_48                  Lactobacillus
## OTU_468                    Finegoldia
## OTU_104                  Gardnerella
## OTU_38    unclassified_Micrococcaceae
## OTU_45   unclassified_Actinomycetales
## OTU_313               Corynebacterium
## OTU_338               Corynebacterium
## OTU_190               Corynebacterium
## OTU_58                Corynebacterium
## OTU_619  unclassified_Actinomycetales
## OTU_83                 Brevibacterium
## OTU_941                  Gardnerella
## OTU_131                  Gardnerella
```

Popular biomarker discovery tools, utilizing multivariate analysis:
- LEfSe
- Indicator Analysis

```
installed.packages()[names(sessionInfo()$otherPkgs), "Version"]
```

```
##               DESeq2 SummarizedExperiment         DelayedArray
##            "1.18.1"              "1.8.1"             "0.4.1"
##          matrixStats              Biobase       GenomicRanges
##            "0.52.2"             "2.38.0"             "1.30.1"
##        GenomeInfoDb              IRanges            S4Vectors
##            "1.14.0"             "2.12.0"             "0.16.0"
##        BiocGenerics             bindrcpp         RColorBrewer
##            "0.24.0"              "0.2.2"              "1.1-2"
##              ggpubr            magrittr                  ape
##             "0.1.6"               "1.5"               "5.0"
##            phyloseq                vegan             lattice
##            "1.22.3"              "2.4-5"           "0.20-35"
##             permute             reshape2             forcats
##             "0.9-4"              "1.4.3"             "0.2.0"
##             stringr               dplyr               purrr
##             "1.3.1"              "0.7.6"             "0.2.4"
##               readr               tidyr              tibble
##             "1.1.1"              "0.7.2"             "1.4.2"
##             ggplot2           tidyverse
##             "2.2.1"              "1.2.1"
```