Vignette for phyloseq: A Bioconductor package for handling and analysis of high-throughput phylogenetic sequence data

Paul J. McMurdie and Susan Holmes*
Statistics Department, Stanford University,
Stanford, CA 94305, USA
*E-mail: mcmurdie@stanford.edu
https://github.com/joey711/phyloseq

March 15, 2012

Contents

1 Summary

The analysis of microbiological communities brings many challenges: the integration of many different types of data with methods from ecology, genetics, phylogenetics, network analysis, visualization and testing. The data itself may originate from widely different sources, such as the microbiomes of humans, soils, surface and ocean waters, wastewater treatment plants, industrial facilities, and so on; and as a result, these varied sample types may have very different forms and scales of related data that is extremely dependent upon the experiment and its question(s). The phyloseq package is a tool to import, store, analyze, and graphically display complex phylogenetic sequencing data that has already been clustered into Operational Taxonomic Units (OTUs), especially when there is associated sample data, phylogenetic tree, and/or taxonomic assignment of the OTUs. This package leverages many of the tools available in R for ecology and phylogenetic analysis (vegan, ade4, ape, picante), while also using advanced/flexible graphic systems (ggplot2) to easily produce publication-quality graphics of complex phylogenetic data. phyloseq uses a specialized system of S4 classes to store all related phylogenetic sequencing data as single experiment-level object, making it easier to share data and reproduce analyses. In general, phyloseq seeks to facilitate the use of R for efficient interactive and reproducible analysis of OTU-clustered high-throughput phylogenetic sequencing data.

2 About this vignette

A separate vignette is included within the phyloseq-package that describes the basics of importing pre-clustered phylogenetic sequencing data, data filtering, as well as some transformations and some additional details about the package and installation. A quick way to load it is:

```
> vignette("phyloseq_basics")
```

By contrast, this vignette is intended to provide functional examples of the analysis tools and wrappers included in phyloseq. All necessary code for performing the analysis and producing graphics will be included with its description, and the focus will be on the use of example data that is included and documented within the phyloseq-package.

Let's start by loading the `phyloseq-package`:

```
> library("phyloseq")
```

3 Data

To facilitate testing and exploration of tools in phyloseq, this package includes example data from published studies. Many of the examples in this vignette use either the `GlobalPatterns` or `enterotype` datasets as source data. The `GlobalPatterns` data was described in an article in PNAS in 2011 [1], and compares the microbial communities of 25 environmental samples and three known "mock communities" — a total of 9 sample types — at a depth averaging 3.1 million reads per sample. The `enterotype` dataset was described in a 2011 article in Nature [2], which compares, the faecal microbial communities from 22 subjects using complete shotgun DNA sequencing. The authors further compare these microbial communities with the faecal communities of subjects from other studies, for a total of 280 faecal samples / subjects, and 553 genera. Sourcing data from different studies invariable leads to gaps in the data for certain variables, and this is easily handled by `R`'s core `NA` features.

Because this data is included in the package, the examples can easily be run on your own computer using the code shown in this vignette. The data is loaded into memory using the `data` command. Let's start by loading the `GlobalPatterns` data.

```
> data(GlobalPatterns)
```

Later on we will use an additional categorical designation — human versus non-human associated samples — that was not in the original dataset. Now is a good time to add it as an explicit variable of the `sampleData`, and because we don't want to type long words over and over, we'll choose a shorter name for this modified version of `GlobalPatterns`, call it `GP`, and also remove a handful of taxa that are not present in any of the samples included in this dataset (probably an artifact):

```
> # prune OTUs that are not present in at least one sample
> GP <- prune_species(speciesSums(GlobalPatterns) > 0, GlobalPatterns)
> # Define a human-associated versus non-human categorical variable:
> human.levels <- levels( getVariable(GP, "SampleType") ) %in%
+         c("Feces", "Mock", "Skin", "Tongue")
> human <- human.levels[getVariable(GP, "SampleType")]
> names(human) <- sample.names(GP)
> # Add new human variable to sample data:
> sampleData(GP)$human <- factor(human)
```

4 Simple exploratory graphics

4.1 Easy Richness Estimates

We can easily create a complex graphic that compares the richness estimates of samples from different environment types in the `GlobalPatterns` dataset, using the `plot_richness_estimates` function. Note that it is important to use raw (untrimmed) OTU-clustered data when performing richness estimates, as they are highly dependent on the number of singletons in a sample.

```
> (p <- plot_richness_estimates(GlobalPatterns, "SampleType", "SampleType"))
```

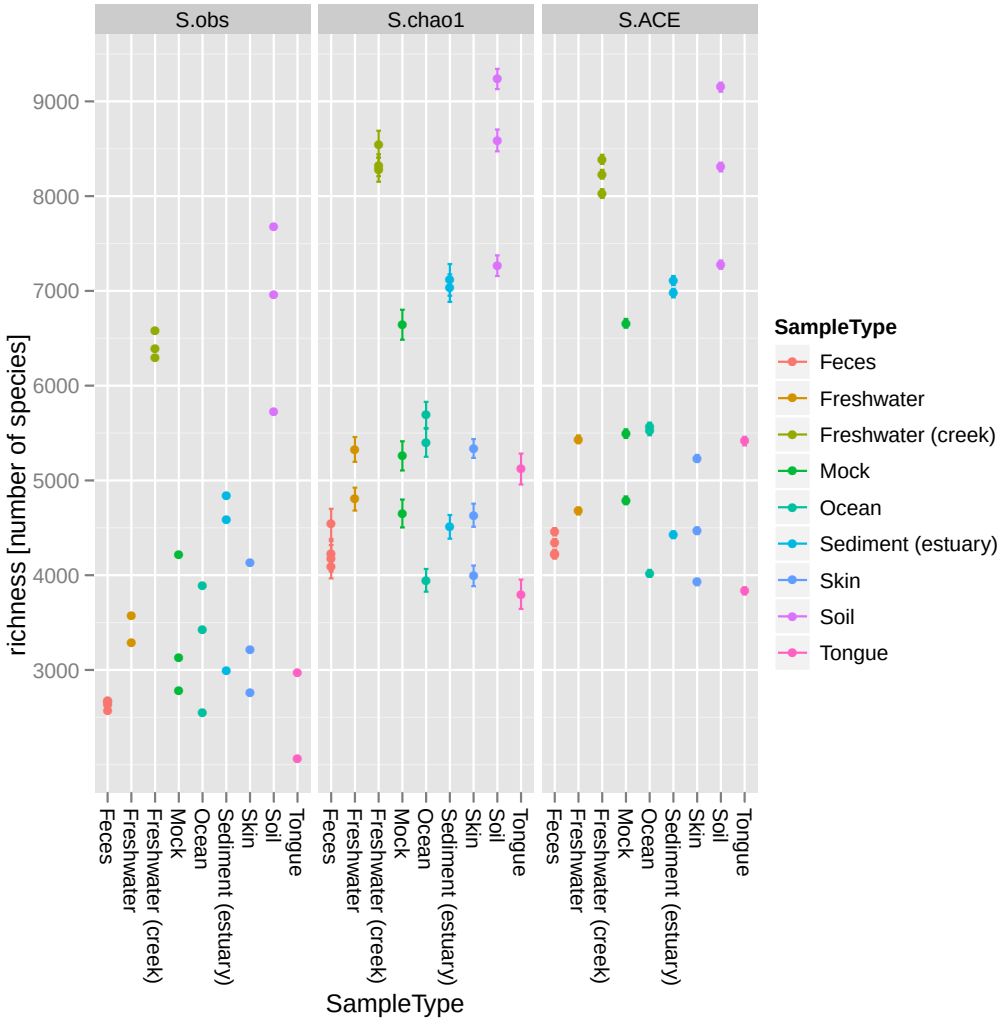


Figure 1: Estimates of the species richness of samples in the "Global Patterns" dataset.

4.2 Exploratory tree plots

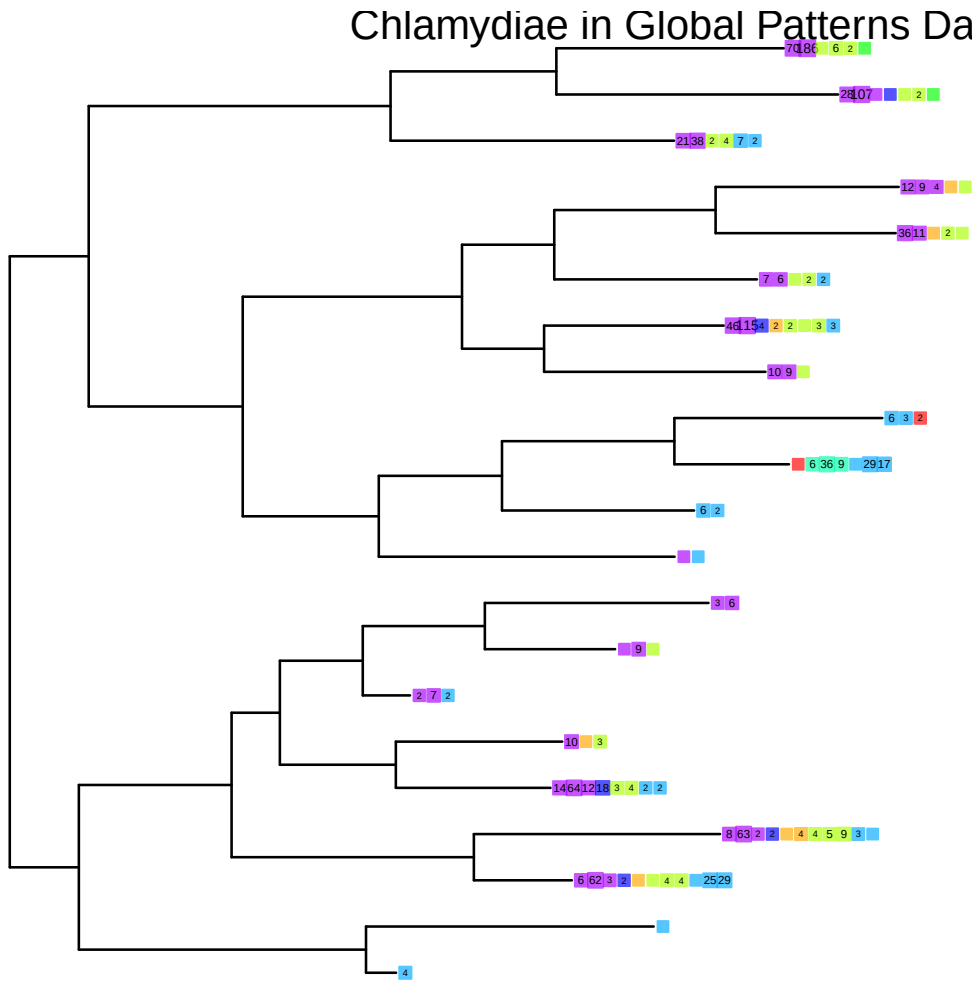
phyloseq also contains a method for easily annotating a phylogenetic tree with information regarding the sample in which a particular taxa was observed, and optionally the number of individuals that were observed.

For the sake of creating a readable tree, let's subset the data to just the Chlamydiae phylum, which consists of obligate intracellular pathogens and is present in only a subset of environments in this dataset.

```
> GP.chl <- subset_species(GP, Phylum=="Chlamydiae")
```

And now we will create the tree graphic form this subset of `GlobalPatterns`, shading by the `"SampleType"` variable, which indicates the environment category from which the microbiome samples originated. The following command also takes the option of labelling the number of individuals observed in each sample (if at all) of each taxa. The symbols are slightly enlarged as the number of individuals increases.

```
> plot_tree_phyloseq(GP.chl, color_factor="SampleType",
+                          type_abundance_value=TRUE, treeTitle="Chlamydiae in Global Patterns Data")
```



4.3 Exploratory bar plots

In the following example we use the included "enterotype" dataset [2].

```
> data(enterotype)
```

We start with a simple rank-abundance barplot, using the cumulative fractional abundance of each OTU in the dataset. In the enterotype dataset, the available published data are simplified as sample-wise fractional occurrences, rather than counts of individuals[a], and OTUs are clustered/labeled at the genus level, but no other taxonomic assignment is available. For the barplot in Figure 2, we further normalize by the total number of samples (280).

```
> par(mar = c(10, 4, 4, 2) + 0.1) # make more room on bottom margin
> N <- 30
> barplot(sort(speciesSums(enterotype), TRUE)[1:N]/nsamples(enterotype), las=2)
```

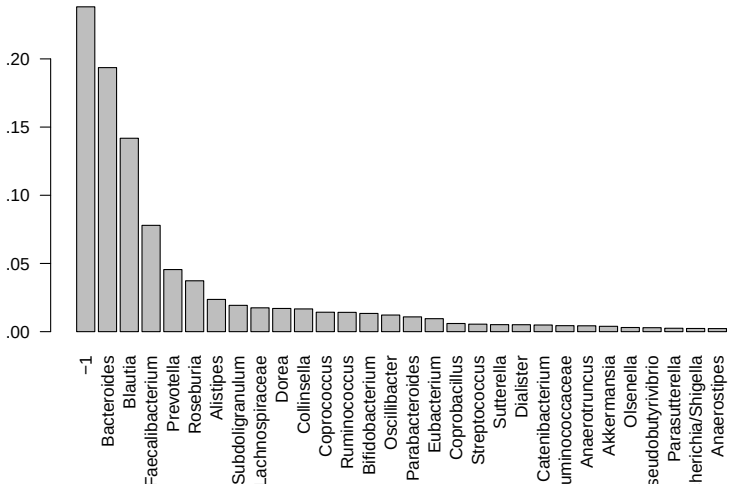


Figure 2: An example exploratory barplot using base `R` graphics and the `speciesSums` and `nsamples` functions.

Note that this first barplot is clipped at the 30th OTU. This was chosen because `nspecies(enterotype)` = 553 OTUs would not be legible on the plot. As you can see, the relative abundances have decreased dramatically by the 10th-ranked OTU.

So what are these OTUs? In the `enterotype` dataset, only a single taxonomic rank type is present:

```
> rank.names(enterotype)
```

```
[1] "Genus"
```

This means the OTUs in this dataset have been grouped at the level of genera, and no other taxonomic grouping/transformation is possible without additional information (like might be present in a phylogenetic tree, or with further taxonomic classification analysis).

We need to know which taxonomic rank classifiers, if any, we have available to specify in the second barplot function in this example, `plot_taxa_bar()`. We have already observed how quickly the abundance decreases with rank, so wo we will subset the enterotype dataset to the most abundant `N` taxa in order to make the barplot legible on this page.

[a] Unfortunate, as this means we lose information about the total number of reads and associated confidences, ability to do more sophisticated richness estimates, etc. For example, knowing that we observed 1 sequence read of a species out of 100 total reads means something very different from observing 10,000 reads out of 1,000,000 total.

```
> TopNOTUs <- names(sort(speciesSums(enterotype), TRUE)[1:10])
> ent10   <- prune_species(TopNOTUs, enterotype)
> print(ent10)

phyloseq-class experiment-level object
OTU Table:          [10 species and 280 samples]
                     species are rows
Sample Map:         [280 samples by 9 sample variables]:
Taxonomy Table:     [10 species by 1 taxonomic ranks]:
```

Note also that there are 280 samples in this dataset, and so a remaining challenge is to consolidate these samples into meaningful groups. A good place to look is the available sample variables, which in most cases will carry more "meaning" than the sample names alone.

```
> sample.variables(ent10)

[1] "Enterotype"     "Sample_ID"      "SeqTech"        "SampleID"
[5] "Project"        "Nationality"    "Gender"         "Age"
[9] "ClinicalStatus"
```

The parameters to `plot_taxa_bar` in the following code-chunk were chosen after various trials. We suggest that you also try different parameter settings while you're exploring different features of the data. In addition to the variables names of `sampleData`, the `plot_taxa_bar()` function recognizes a special parameter name `"TaxaGroup"`, which is not (should not be) a sample variable name in `sampleData(enterotype)`, but instead indicates that the particular graphic parameter should group values by the taxanomic rank specified in the `taxavec` argument. In this example we have also elected to separate the samples by "facets" (separate, adjacent sub-plots) according to the enterotype to which they have been assigned. Within each enterotype facet, the samples are further separated by sequencing technology, and the genera is indicated by fill color. Multiple samples having the same enterotype designation and sequencing technology are plotted side-by-side as separate bars.

```
> (p <- plot_taxa_bar(ent10, "Genus", x="SeqTech", fill="TaxaGroup") +
+                          facet_wrap(~Enterotype) )
```

Figure 3 summarizes quantitatively the increased abundances of *Bacteroides* and *Prevotella* in the Enterotypes 1 and 2, respectively. Interestingly, a large relative abundance of *Blautia* was observed for Enterotype 3, but only from 454-pyrosequencing data sets, not the Illumina or Sanger datasets. This suggests the increased *Blautia* might actually be an artifact. Similarly, *Prevotella* appears to be one of the most abundant genera in the Illumina-sequenced samples among Enterotype 3, but this is not reproduced in the 454-pyrosequencing or Sanger sequencing data.

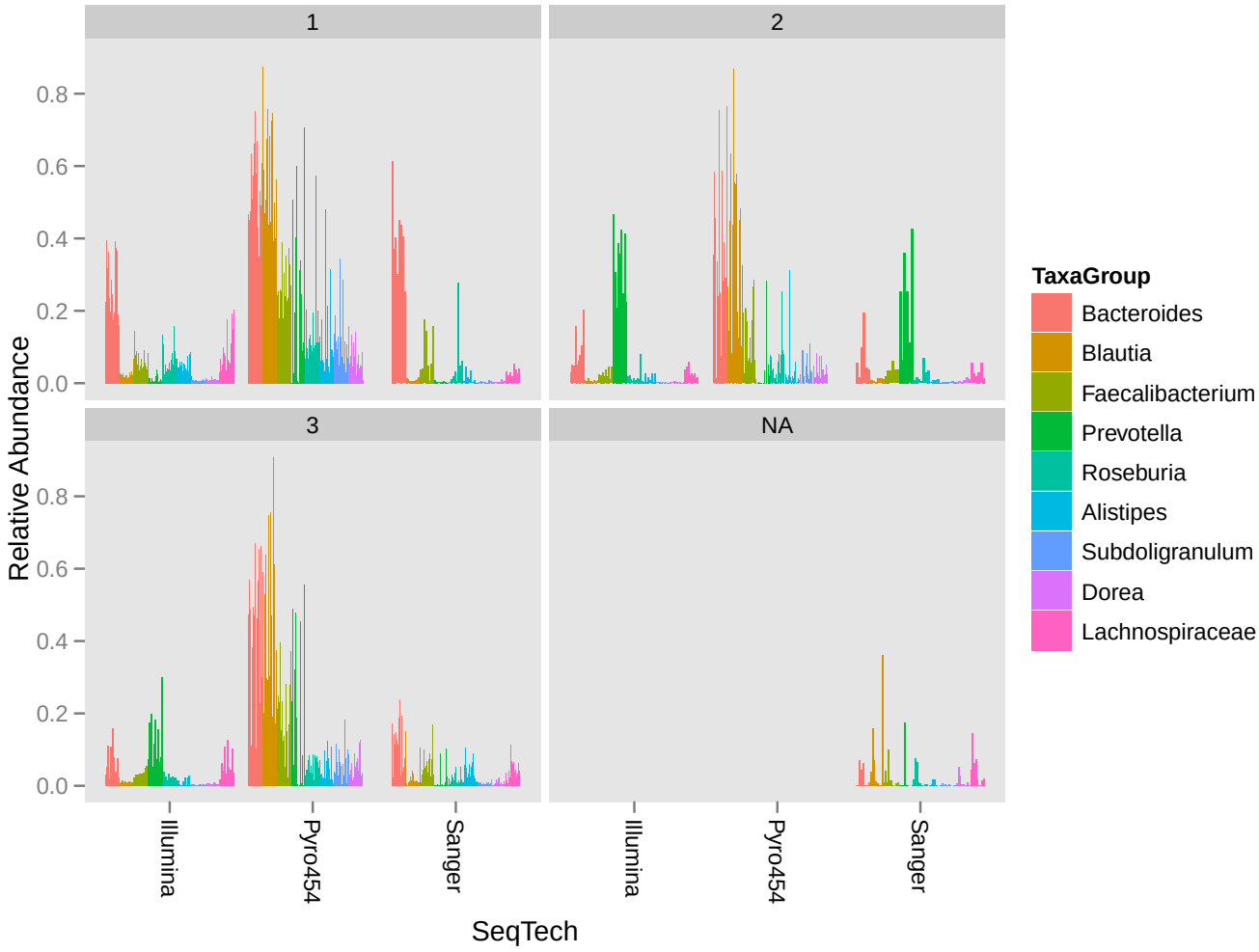


Figure 3: An example exploratory barplot using the `plot_taxa_bar()` function. In this case we have faceted the samples according to their assigned Enterotype. The small subset of samples in the dataset that do not have an assigned Enterotype are shown in the `NA` panel. Within each Enterotype facet, the samples are further separated by sequencing technology, and each genera is shaded a different color. Multiple samples from the same Enterotype and sequencing technology are plotted side-by-side as separate bars (dodged).

5 Exploratory analysis and graphics

5.1 Microbiome Network Representation

Continuing with the `enterotype` dataset, here are some examples for creating a custom network representation of the relationship between microbiome samples in an experiment. This relies heavily on the `igraph` and `ggplot2` packages to create a network display of the "connectedness" of samples according to some user-provided ecological similarity. By default, the position of points (samples) are determined using an algorithm that optimizes the clarity of the display of network "edges", but the spatial position of points does not imply any continuous similarity information like would be the case in an ordination. In this example, the default dissimilarity index was used (Jaccard, co-occurrence), with a maximum distance of `0.3` required to create an edge. Any function that can operate on phyloseq-objects and return a sample-wise distance can be provided as the `dist.fun` argument, or a character string of the name of the distance function already supported in phyloseq. Other distances may result in very different clustering, and this is a choice that should be understood and not taken too lightly, although there is little harm in trying many different distances.

```
> data(enterotype)
> ig <- make_sample_network(enterotype, FALSE, max.dist=0.3)
> (p  <- plot_sample_network(ig, enterotype, color="SeqTech",
+     shape="Enterotype", line_weight=0.4, label=NULL))
```

Interestingly, at this level of analysis and parameter-settings the two major sub-graphs appear to be best explained by the sequencing technology and not the subject enterotype (Figure 4), suggesting that the choice of sequencing technology has a major effect on the microbial community one can observe. This seems to differ somewhat with the inferences described in the "enterotype" article [2]. However, there could be some confounding or hidden variables that might also explain this phenomenon, and the well-known differences in the sequence totals between the technologies may also be an important factor. Furthermore, since this is clearly an experimental artifact (and they were including data from multiple studies that were not orginally planned for this purpose), it may be that the enterotype observation can also be shown in a network analysis of this data after removing the effect of sequencing technology and related sequencing effort. Such an effort would be interesting to show here, but is not yet included.

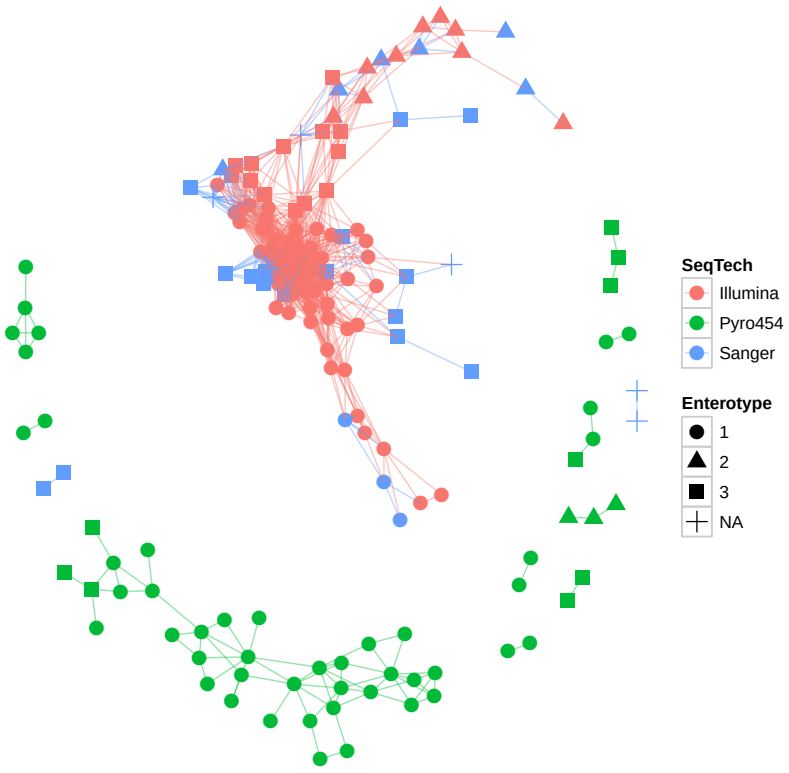


Figure 4: Network representation of the relationship between microbiome samples in the "Enterotype" dataset [2].

5.2 Ordination Methods

5.2.1 Principal Coordinates Analysis (PCoA)

We take as our first example, a reproduction of Figure 5 from the "Global Patterns" article [1]. The authors show a 3-dimensional representation of the first three axes of a Principal Coordinates Analysis (PCoA[b]) performed on the unweighted-UniFrac distance (see section 5.3.3) using all of the available sequences (their approach included both 5' and 3' sequences). According to the authors, "the first axis [appears to be associated with a] host associated/free living [classification]," and similarly the third axis with "saline/nonsaline environment[s]."

The following reproduces the unweighted UniFrac distance calculation on the full dataset. Note that this calculation can take a long time because of the large number of OTUs. Parallelization is recommended for large datasets, typically if they are as large as `GlobalPatterns`, or larger. For details on parallelization, see the details section and examples in the `UniFrac()` documentation, and also the page dedicated to the topic on the phyloseq-wiki:

`https://github.com/joey711/phyloseq/wiki/Fast-Parallel-UniFrac`

```
> data(GlobalPatterns)

> GPUF <- UniFrac(GlobalPatterns)

> GloPa.pcoa <- pcoa(GPUF)
```

Before we look at the results, let's first investigate how much of the total distance structure we will capture in the first few axes. We can do this graphically with a "scree plot", an ordered barplot of the relative fraction of the total eigenvalues associated with each axis (Fig. 5).

```
> barplot(GloPa.pcoa$values$Relative_eig)
```

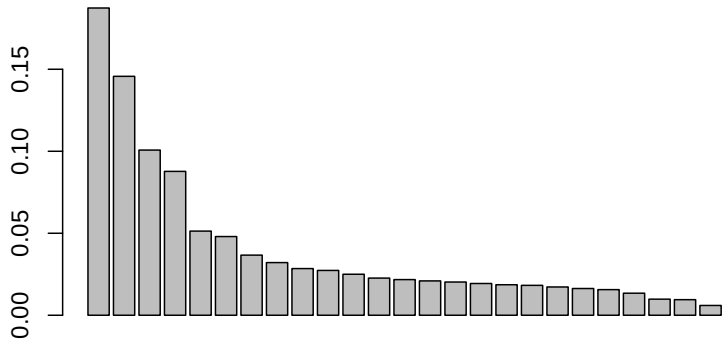
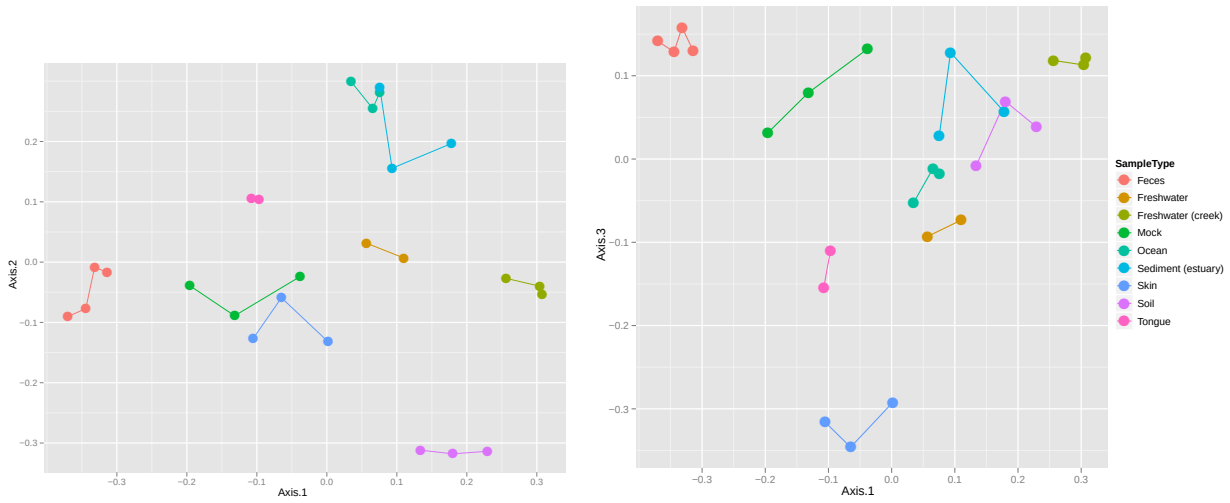


Figure 5: Scree plot of the PCoA used to create Figure 5 from the "Global Patterns" article [1]. The first three axes represent 43% of the total variation in the distances. Interestingly, the fourth axis represents another 9%, and so may warrant exploration as well. A scree plot is an important tool for any ordination method, as the relative importance of axes can vary widely from one dataset to another.

[b] This is also sometimes referred to as "Multi-Dimensional Scaling", or "MDS"

Next, we will reproduce Figure 5 from the "Global Patterns" article [1], but separating the three axes into 2 plots using `plot_ordination()` (Fig. 6).

```
> (p12 <- plot_ordination(GlobalPatterns, GloPa.pcoa, "samples", color="SampleType") +
+   geom_line() + geom_point(size=5) + scale_colour_hue(legend = FALSE) )
> (p13 <- plot_ordination(GlobalPatterns, GloPa.pcoa, "samples", axes=c(1, 3),
+   color="SampleType") + geom_line() + geom_point(size=5) )
```



(a) Axes 1, 2 **(b)** Axes 1, 3

Figure 6: A reproduction in *phyloseq* / `R` of the main panel of Figure 5 from the "Global Patterns" article [1], on two plots. The horizontal axis represents the first axis in the PCoA ordination, while the top and bottom vertical axes represent the second and third axes, respectively. Different points represent different samples within the dataset, and are shaded according to the environment category to which they belong. The color scheme is the default used by *ggplot*.

5.2.2 non-metric Multi-Dimensional Scaling (nmMDS)

We repeat the previous example, but instead using non-metric multidimensional scaling (nmMDS - `metaMDS()`) limited to just two dimensions. This approach limits the amount of residual distance "not shown" in the first two (or three) axes, but forefeits some mathematical properties and does not always converge within the specified number of axes.

```
> # (Re)load UniFrac distance matrix and GlobalPatterns data
> data(GlobalPatterns)
> load("Unweighted_UniFrac.RData") # reloads GPUF variable
> GP.nmMDS <- metaMDS(GPUF, k=2) # perform nmMDS, set to 2 axes

Run 0 stress 0.1432785
Run 1 stress 0.1809287
Run 2 stress 0.167022
Run 3 stress 0.1432799
... procrustes: rmse 0.0007108802  max resid 0.002686656
*** Solution reached

> (p <- plot_ordination(GlobalPatterns, GP.nmMDS, "samples", color="SampleType") + geom_line() + geom_poi
```

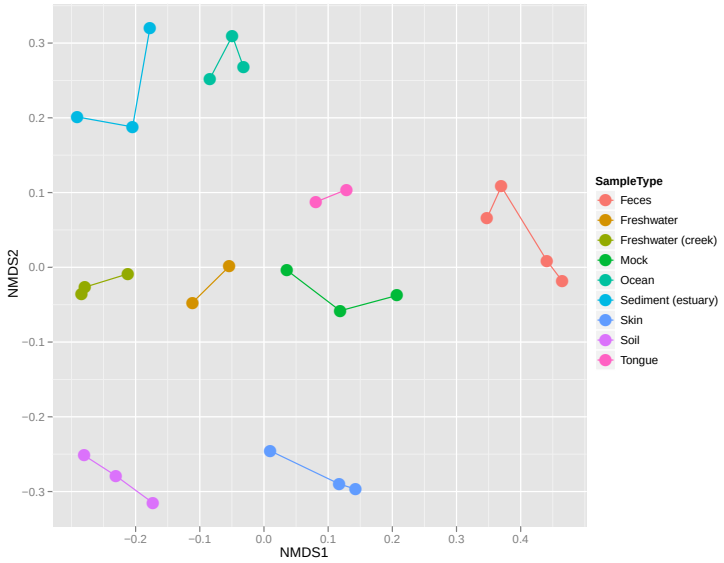


Figure 7: An example exploratory ordination using non-metric multidimensional scaling (nmMDS) on the unweighted UniFrac distance between samples of the "Global Patterns" dataset. Sample points are shaded by environment type, and connected by a line if they belong to the same type. Compare with Figure 5 from the "Global Patterns" article [1].

Figure 7 nicely shows the relative dissimilarities between microbial communities from different habitats. However, it fails to indicate *what* was different between the communities. For an ordination method that provides information on the taxa that explain differences between samples (or groups of samples), we use Correspondence Analysis (Section 5.2.3).

5.2.3 Correspondence Analysis (CA)

In the following section we will show continue our exploration of the "GlobalPatterns" dataset using various features of an ordination method called Correspondence Analysis. We give special emphasis to exploratory interpretations using the biplot, because it provides additional information that is not available from PCoA or nmMDS.

Let's start by performing a Correspondence Analysis and investigating the scree plot (Figure 8). Both interestingly and challengingly, the scree plot suggests that the `GlobalPatterns` abundance data is quite high-dimensional, with the first two CA axes accounting for not quite 17% of the total (chi-square) variability. Note the absence of a steep decline in eigenvalue fraction as axis number increases. Each additional axis represents only marginally less variability than the previous. It is often more convenient if the first two (or three) axes account for most of the variability.

```
> data(GlobalPatterns)
> # Need to clean the zeros from GlobalPatterns:
> GP <- prune_species(speciesSums(GlobalPatterns)>0, GlobalPatterns)
> # Get the names of the taxa from Phyla with over 1-million individuals observed
> top.TaxaGroup <- sort(
+   tapply(speciesSums(GP), taxTab(GP)[, "Phylum"], sum, na.rm=TRUE), decreasing=TRUE)
> top.TaxaGroup <- top.TaxaGroup[top.TaxaGroup > 1*10^6]
> # Now prune further, to just the most-abundant phyla
> GP <- subset_species(GP, Phylum %in% names(top.TaxaGroup))
> # Now do the correspondence analysis
> gpca  <- cca.phyloseq(GP)
> barplot(gpca$CA$eig/sum(gpca$CA$eig), las=2)
```

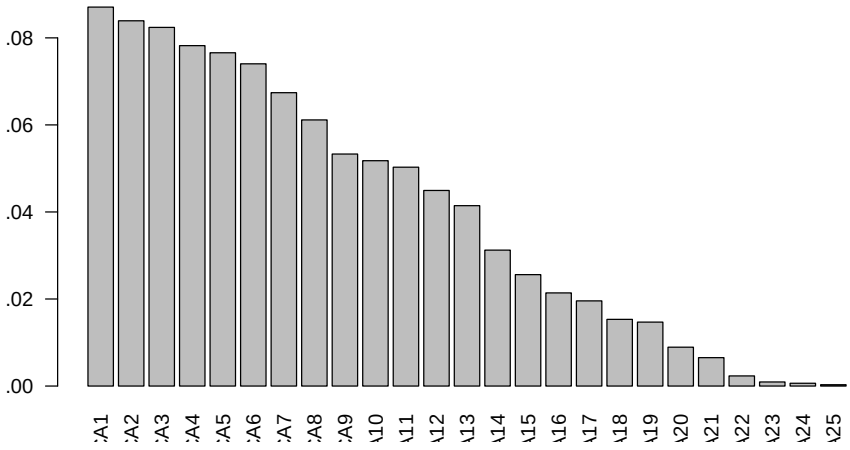
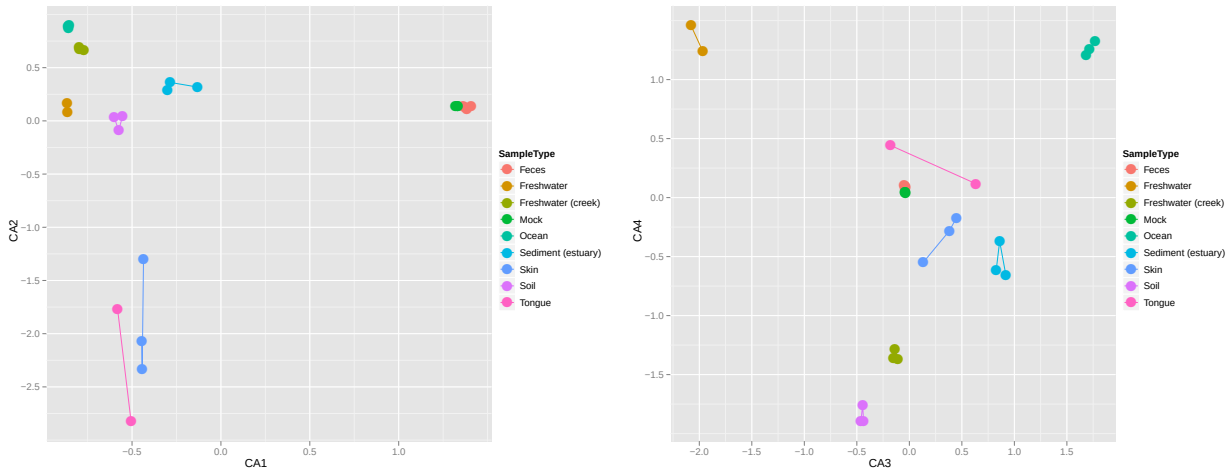


Figure 8: The correspondence analysis (CA) scree plot of the "Global Patterns" dataset [1].

Now let's investigate how the samples behave on the first few CA axes.

```
> (p12 <- plot_ordination(GP, gpca, "samples", color="SampleType") +
+   geom_line() + geom_point(size=5) )
> (p34 <- plot_ordination(GP, gpca, "samples", axes=c(3, 4), color="SampleType") +
+   geom_line() + geom_point(size=5) )
```



(a) Axes 1, 2 **(b)** Axes 3, 4

Figure 9: First 4 axes of Correspondence Analysis (CA) of the "Global Patterns" dataset [1].

A clear feature of these plots is that the feces and mock communities cluster tightly together, far away from all other samples on the first axis (CA1) in Fig. 9a. The skin and tongue samples separate similarly, but on the second axis. Taken together, it appears that the first two axes are best explained by the separation of human-associated "environments" from the other non-human environments in the dataset, with a secondary separation of tongue and skin samples from feces.

We will now investigate further this top-level structure of the data, using an additional feature of correspondence analysis that allows us to compare the relative contributions of individual taxa on the same graphical space: the "biplot". However, because we just displayed the position of samples in the ordination and there are many thousands of OTUs, we will focus on creating an interpretable plot of the OTUs. For creating graphics that combine the two plots, try the "`biplot`" or "`split`" option for `type` in `plot_ordination()`.

```
> p1 <- plot_ordination(GP, gpca, "species", color="Phylum")
> # Re-draw this as topo without points
> p1 <- ggplot(p1$data, p1$mapping) + geom_density2d() + facet_wrap(~Phylum)
> # Add layer. Subset of species-points, beyond threshold dist from origin.
> p2 <- p1 + geom_point(data=subset_ord_plot(p1, 1.0, "square"), size=1) +
+   scale_colour_hue(legend = FALSE)
```

While Fig 10 reveals some useful patterns and a few interesting outliers, but what if we want a complete summary of how each phylum is represented along each axis? The following code is a way to show this using boxplots, while still avoiding the occlusion problem (points layered on top of each other), and also conveying some useful information about the pattern of taxa that contribute to the separation of human-associated samples from the other sample types. It re-uses the data that was stored in the `ggplot2` plot object created in the previous figure, `p`, and creates a new boxlot graphical summary of the positions of each phylum (Fig 11).

```
> # Melt the species-data.frame, DF, to facet each CA axis separately
> mdf <- melt(p1$data[, c("CA1", "CA2", "Phylum", "Family", "Genus")],
+             id=c("Phylum", "Family", "Genus") )
> # Select taxonomic-Family labels of special outliers
> LF <- subset(mdf, Phylum=="Cyanobacteria" & variable=="CA2" & value < -2.5)
> # build plot: boxplot summaries of each CA-axis, with labels
```

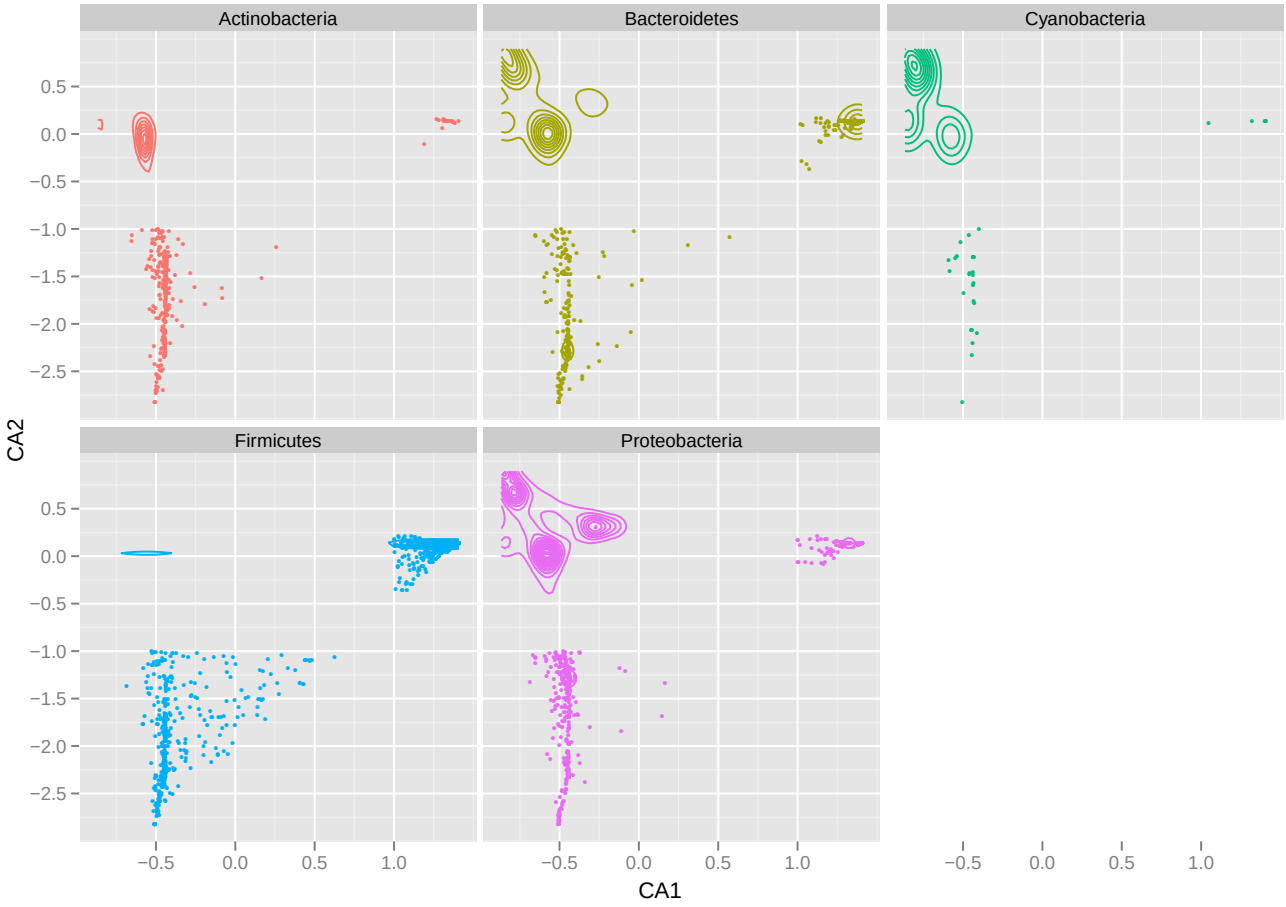


Figure 10: Species plot of the "Global Patterns" CA first two axes, with each phylum on a different panel ("facet"). Only the phyla with at least one million individuals observed (cumulative, all samples) are included. The topo lines indicate regions of the plot with a high density of points that would otherwise be difficult to discern

```
> p <- ggplot(mdf, aes(Phylum, value, color=Phylum)) + geom_boxplot() +
+   facet_wrap(~variable, 2) + scale_colour_hue(legend = FALSE) +
+   theme_bw() + opts( axis.text.x = theme_text(angle = -90, hjust = 0) )
> # Add the text label layer, and render ggplot graphic
> (p <- p + geom_text(aes(Phylum, value+0.1, color=Phylum, label=Family),
+                     data=LF, vjust=0, size=2) )
```

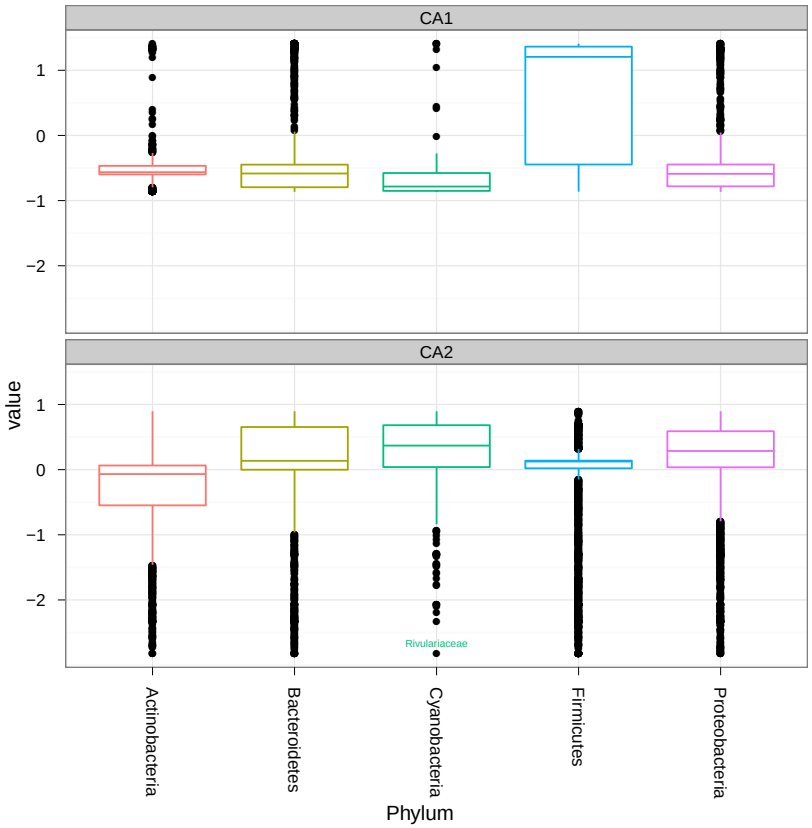


Figure 11: Boxplot of taxa (species in this case) of the "Global Patterns" CA first two axes, shaded/separated by phylum. Through this approach it is much easier to see particular species that cluster unusually relative to the rest of their phylum, for example the cyanobacteria (*Rivulariaceae*) that is positioned most in the negative CA2 direction toward the Tongue/Skin samples.

One way to relate some of the high-level patterns we observed from correspondence analysis is to visualize the relative abundances of the relevant phylogenetic groups, to see if this does in fact support / explain the human/environment microbiome differences. Here is an example using the `plot_taxa_bar` function described earlier in Section 4.3.

```
> (p <- plot_taxa_bar(GP, "Phylum", NULL, threshold=0.9, "human", "SampleType",
+                                                    facet_formula= TaxaGroup ~ .) )
```

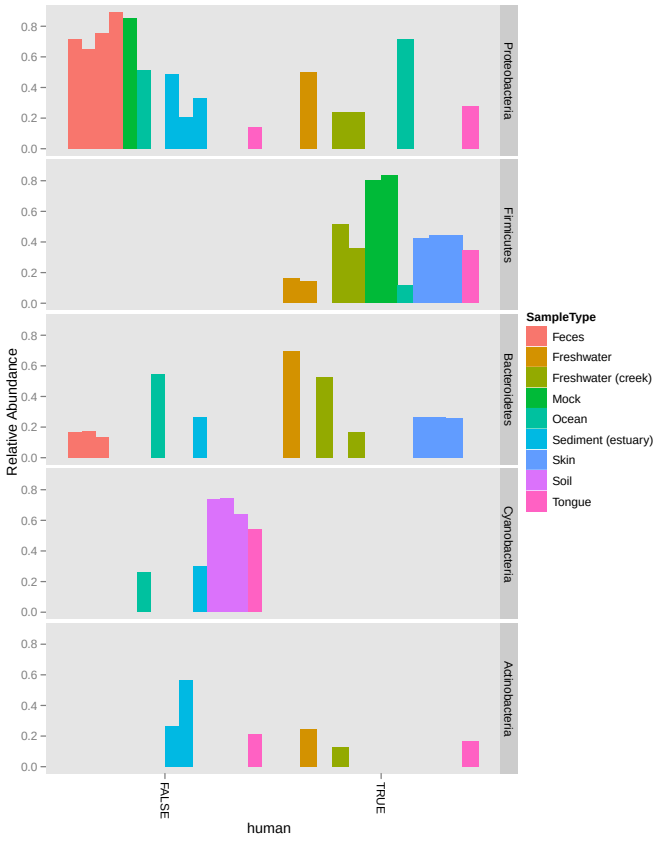


Figure 12: Phylum-level comparison of relative abundance of taxa in samples that are from human microbiomes (or not).

In Fig 12 we've used the `threshold` parameter to omit all but phyla accounting for the top 90% of phyla in any one sample. Some patterns emerging from this display appear to be: (1) Cyanobacteria, Actinobacteria appear under-represented in human samples; (2) conversely, Firmicutes appear over-represented in human samples; (3) Acidobacteria, Verrucomicrobia appear over-represented in the fecal samples; (4) the only Crenarchaeota were observed in the Mock sample, which is not really a community but a simulated community used as a control. These are not hugely surprising based on previous biological observations from the field, but it is hopefully useful code that can be applied on other datasets that you might have.

5.2.4 Double Principle Coordinate Analysis (DPCoA)

`DPCoA()` [3].

5.3 Distance Methods

5.3.1 `distance()`: Central Distance Function

Many comparisons of microbiome samples, including the graphical model (Section 5.1) and the PCoA analysis (Section 5.2.1), require a calculation for the relative dissimilarity of one microbial community to another, or "distance". Although not fully implemented yet, the phyloseq-package intends to provide a unified ecological distance function for calculating a matrix of microbial community distances between the samples in an experiment. This will surely include a wrapper for the several dozen distance calculations provided via the three distance functions in the `vegan`-package, many of the distance methods supported in the `ade4`-package, as well as the included `UniFrac` distance function (Section 5.3.3), a method for calculating Double Principal Coordinate Analysis (DPCoA), as well as an extension to the vegan interface for arbitrary, user-defined distances.

The function will take a `phyloseq-class` object and an argument indicating the distance type; and it will return a `dist`-class distance matrix.

5.3.2 `vegdist()` extension

The *phyloseq* package includes an extension for the `vegdist()` function from the *vegan* package [4], which in-turn can calculate 14 or so ecologically relevant distances / dissimilarity indices. The primary argument should be a `phyloseq-class` object, and the expected result is a sample-wise distance matrix.

```
> data(esophagus)
> vegdist(esophagus)
```

```
     B    C
C 0.406
D 0.498 0.591
```

The available distances/dissimilarity indices calculated by `vegdist()` currently include the following:

```
 [1] "manhattan"  "euclidean"  "canberra"   "bray"       "kulczynski"
 [6] "jaccard"    "gower"      "altGower"   "morisita"   "horn"
[11] "mountford"  "raup"       "binomial"   "chao"
```

These are specified by the `method=` argument. For example, if one alternatively wants to calculate the Jaccard distance instead of the default (Bray-Curtis), the following command will work:

```
> data(esophagus)
> vegdist(esophagus, "jaccard")
```

```
     B    C
C 0.578
D 0.665 0.743
```

We also have plans to extend the `designdist`, `betadiver`, and `dist` functions, which provide even further options for distance type. `designdist` also provides a way to define a custom distance calculation, while `betadiver` calculates all 24 ecological distances reviewed in Koleff et al. 2003 [5]. We plan to wrap all of these methods into one *central* distance calculator method, say "`sampleDistance()`", that would also include UniFrac, DPCoA, etc. This will be implemented soon.

5.3.3 UniFrac and weighted UniFrac

UniFrac is a recently-defined [6] and popular distance metric to summarize the difference between pairs of ecological communities. All UniFrac variants use a phylogenetic tree of the relationship among taxa as central information to calculating the distance between two samples/communities. An unweighted UniFrac distance matrix only considers the presence/absence of taxa, while weighted UniFrac accounts for the relative

abundance of taxa as well as their phylogenetic distance. Prior to *phyloseq*, a non-parallelized, non-Fast implementation of the unweighted UniFrac was available in R packages (`picante::unifrac` [7]). In the *phyloseq* package we provide optionally-parallelized implementations of Fast UniFrac [8] (both weighted and unweighted, with plans for additional UniFrac variants), all of which return a sample-wise distance matrix from any `phyloseq-class` object that contains a phylogenetic tree component.

The following is an example calculating the UniFrac distance (both weighted and unweighted) matrix using the "esophagus" example dataset:

```
> data(esophagus)
> UniFrac(esophagus, weighted=TRUE)
> UniFrac(esophagus, weighted=FALSE)
```

```
      B     C
C 0.204
D 0.260 0.248
```

```
      B     C
C 0.518
D 0.518 0.542
```

See the wiki-page devoted to details about calculating the UniFrac distances for your experiment. In particular, some example run-times are provided for comparison, as well as details for initializing a parallel "back end" to perform the computation with multiple processor cores simultaneously:

`https://github.com/joey711/phyloseq/wiki/Fast-Parallel-UniFrac`

5.4 Hierarchical Clustering

Another potentially useful and popular way to visualize/decompose sample-distance matrices is through hierarchical clustering (e.g. `hclust()`). In the following example, we reproduce Figure 4 from the "Global Patterns" article [1], using the unweighted UniFrac distance and the UPGMA method (`hclust` parameter `method="average"`). Try `help("hclust")` for alternative clustering methods included in standard `R`.

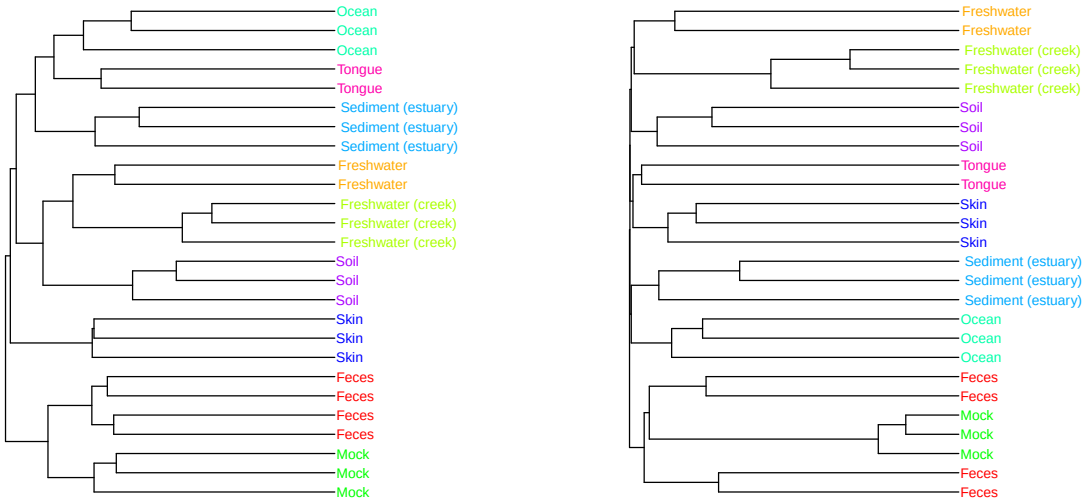
```
> # (Re)load UniFrac distance matrix and GlobalPatterns data
> data(GlobalPatterns)
> load("Unweighted_UniFrac.RData") # reloads GPUF variable
> # Manually define color-shading vector based on sample type.
> colorScale    <- rainbow(length(levels(getVariable(GlobalPatterns, "SampleType"))))
> cols          <- colorScale[getVariable(GlobalPatterns, "SampleType")]
> GP.tip.labels <- as(getVariable(GlobalPatterns, "SampleType"), "character")
> GP.hclust     <- hclust(GPUF, method="average")
```

Plot the hierarchical clustering results as a dendrogram, after first converting the `hclust-class` object to `phylo-class` tree using the `as.phylo()` function.

```
> plot(as.phylo(GP.hclust), show.tip.label=TRUE, tip.color="white")
> tiplabels(GP.tip.labels, col=cols, frame="none", adj=-0.05, cex=0.7)
```

Create an alternative plot, using the Jaccard distance and complete-linkage clustering (the default) instead of UPGMA.

```
> jaccCLC <- hclust(vegdist(GlobalPatterns, "jaccard"))
> plot( as.phylo(jaccCLC), show.tip.label=TRUE, tip.color="white" )
> tiplabels(GP.tip.labels, col=cols, frame="none", adj=-0.05, cex=0.7)
```



(a) UniFrac / UPGMA

(b) Jaccard / CLC

Figure 13: An alternative means of summarizing a distance matrix via hierarchical clustering and plotting as an annotated dendrogram. Compare with Figure 4 from the "Global Patterns" article [1]. Panel 13a represents a faithful reproduction of the original approach from the article using `R` utilities, while Panel 13b is an illustration of slightly different results with different choices of distance measure and clustering algorithm. Some differences in Panel 13a from the original article might be explained by the `GlobalPatterns` dataset in *phyloseq* includes the summed observations from both directions (5' and 3'), while in the article they show the results separately. Furthermore, in the article the "mock" community is not included in the dataset, but an extra fecal sample is included.

6 Validation

6.1 Multiple Inference Correction

The *phyloseq* package includes support for significance testing with correction for multiple inference. This is particularly important when testing for significance of the abundance patterns among thousands of microbes (OTUs). This is a common question of phylogenetic sequence data, that is, "what is the subset of microbes that significantly correlate with a scientifically-interesting sample variable". Although we plan to include support for other types of multiple-inference corrected tests, this "which taxa?" test is the only directly supported test at the moment.

Our initial implementation of this support is via an extension to the `mt.maxT()` and `mt.minP` functions in the *multtest* package [9] (Bioconductor repo). This uses permutation-adjusted p-values in a multiple testing procedure that provides strong control of the Family-Wise Error Rate (FWER) among the taxa being tested. The user specifies a sample-variable among the `sampleData` component, or alternatively provides a sample-wise vector or factor that classifies the samples into groups. Additional optional parameters can be provided that specify the type of test ("`test=`"), the sidedness of the test ("`side=`"), as well as some additional technical/computational parameters.

In the following example we test whether a particular genera correlates with the Enterotype classification of each sample. Note that we have to specify an alternate test, `test="f"`, because the default test (t-test) can only handle up to 2 classes, and there are three enterotype classes.

```
> data(enterotype)
> # Filter samples that don't have Enterotype classification.
> x <- subset_samples(enterotype, !is.na(Enterotype))
> # Calculate the multiple-inference-adjusted P-values
> ent.p.table <- mt(x, "Enterotype", test="f")
> print(head(ent.p.table, 10))
```

genera	index	teststat	rawp	adjp	plower
Prevotella	207	344.73	0.0001	0.0158	0.0001
Bacteroides	203	85.01	0.0001	0.0158	0.0001
Blautia	187	19.52	0.0001	0.0158	0.0001
Bryantella	503	16.38	0.0001	0.0158	0.0001
Parabacteroides	205	12.89	0.0001	0.0158	0.0001
Alistipes	208	8.71	0.0002	0.0301	0.0158
Bifidobacterium	240	9.29	0.0004	0.0560	0.0430
Holdemania	201	7.64	0.0009	0.1146	0.1031
Dorea	182	7.44	0.0009	0.1146	0.1031
Phascolarctobacterium	513	7.01	0.0014	0.1695	0.1585

Table 1: For computational efficiency this calculation was run separately, and results embedded here.

Not surprisingly, *Prevotella* and *Bacteroides* top the list, since they were major components of the "Enterotype" classification.

Please also note that we are planning to incorporate other tools from *multtest* that would allow for other types of multiple-inference correction procedures, for instance, strong control of the False Discovery Rate (FDR). These additional options will be made available shortly.

7 Further Examples

This vignette is limited in size and scope because of constraints on the cumulative-size of packages allowed in Bioconductor.

7.1 phyloseq Wiki

For further examples presented in html/wiki format, please see:
`https://github.com/joey711/phyloseq/wiki/Graphics-Examples`

7.2 phyloseq Vignette Gallery

For additional and/or updated vignettes not present in the official Bioconductor release, please see:
`https://github.com/joey711/phyloseq/wiki/Vignettes`

7.3 phyloseq Feedback

For feature requests, bug reports, and other suggestions and issues, please go to:
`https://github.com/joey711/phyloseq/issues`

References

[1] Caporaso, J. G. *et al.* Global patterns of 16s rrna diversity at a depth of millions of sequences per sample. *Proceedings of the National Academy of Sciences* **108**, 4516–4522 (2011). URL `http://www.pnas.org/content/108/suppl.1/4516.abstract`. `http://www.pnas.org/content/108/suppl.1/4516.full.pdf+html`.

[2] Arumugam, M. *et al.* Enterotypes of the human gut microbiome. *Nature* **473**, 174–180 (2011). URL `http://dx.doi.org/10.1038/nature09944`.

[3] Pavoine, S., Dufour, A.-B. & Chessel, D. From dissimilarities among species to dissimilarities among communities: a double principal coordinate analysis. *Journal of Theoretical Biology* **228**, 523 – 537 (2004). URL `http://www.sciencedirect.com/science/article/pii/S0022519304000736`.

[4] Oksanen, J. *et al. vegan: Community Ecology Package* (2011). URL `http://CRAN.R-project.org/package=vegan`. R package version 1.17-10.

[5] Koleff, P., Gaston, K. J. & Lennon, J. J. Measuring beta diversity for presence–absence data. *Journal of Animal Ecology* **72**, 367–382 (2003). URL `http://dx.doi.org/10.1046/j.1365-2656.2003.00710.x`.

[6] Lozupone, C. & Knight, R. UniFrac: a new phylogenetic method for comparing microbial communities. *Applied and environmental microbiology* **71**, 8228–8235 (2005).

[7] Kembel, S. W. *et al.* Picante: R tools for integrating phylogenies and ecology. *Bioinformatics (Oxford, England)* **26**, 1463–1464 (2010).

[8] Hamady, M., Lozupone, C. & Knight, R. Fast unifrac: facilitating high-throughput phylogenetic analyses of microbial communities including analysis of pyrosequencing and phylochip data. *ISME J* **4**, 17–27 (2009). URL `http://dx.doi.org/10.1038/ismej.2009.97`.

[9] Pollard, K. S., Gilbert, H. N., Ge, Y., Taylor, S. & Dudoit, S. *multtest: Resampling-based multiple hypothesis testing.* R package version 2.10.0.