

Introducing R and the RStudio IDE

Hena R. Ramay, IMC
Bioinformatics, University of
Calgary



UNIVERSITY OF
CALGARY

International
Microbiome
Centre



UNIVERSITY OF CALGARY
CUMMING SCHOOL OF MEDICINE

What is R?

- R (since 1995) is a programming language developed to teach statistics
- R is open source (i.e. free), widely used, flexible, and powerful





Packages for everything

`car` – car's `Anova` function is popular for making type II and type III Anova tables

`mgcv` – Generalized Additive Models

`lme4/nlme` – Linear and non-linear mixed effects models

`randomForest` – Random forest methods from machine learning

`multcomp` – Tools for multiple comparison testing

`vcd` – Visualization tools and tests for categorical data

`glmnet` – Lasso and elastic-net regression methods with cross validation

`survival` – Tools for survival analysis

`caret` – Tools for training regression and classification models

What is R? What is RStudio?

R is a programming language



RStudio is an Integrated Development Environment (IDE) that allows users to run R in a more user-friendly way



What is R? What is RStudio?

R: Engine



RStudio: Dashboard



<http://moderndive.com/>



RStudio looks like this

RStudio interface screenshot showing a workflow for alpha diversity analysis.

Code Editor: Contains R code for alpha diversity calculations and ggplot2 boxplots.

```
119 ## Take relative abundance
120 rel <- transform_sample_counts(ps, function(x) x / sum(x))
121
122 ## Execute filter
123 relf <- prune_taxa(keptaxa, rel)
124 psf <- prune_taxa(keptaxa, ps)
125
126
127
128 ###### Alpha diversity
129
130 ## Calculate alpha diversity using unfiltered data because rare variants influence measures of alpha div
131
132 ## Make table of alpha diversity calculations
133 alpha <- estimate_richness(ps)
134 alpha_info <- sample_data(ps)
135 aa <- cbind(alpha, alpha_info)
136
137 ## Check for outliers
138 aqplot(alpha$Shannon, binwidth = 0.05) + xlab("Shannon diversity")
139 aqplot(alpha$Simpson, binwidth = 0.005) + xlab("Simpson diversity")
140
141 ## Plot
142 a1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Shannon)") + xlab("Timepoint")
143 a1
144
145 a2 <- ggplot(aa, aes(x = timepoint, y = Simpson, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Simpson)") + xlab("Timepoint")
146 a2
147
148 a3 <- ggplot(aa, aes(x = timepoint, y = Simpson, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Simpson)") + xlab("Timepoint")
149 a3
150
```

Console: Shows the R session history, including the creation of phyloseq objects and the execution of ggplot2 commands.

```
> s
> taxa_names(ps) <- asv_names
> colnames(otu_table(ps)) <- asv_names
> rownames(tax_table(ps)) <- asv_names
>
>
> ## Remove control samples
> ps <- prune_samples(sample_data(ps)$treatment != "NA", ps)
> ps
phyloseq-class experiment-level object
otu_table() OTU Table: [ 2171 taxa and 95 samples ]
sample_data() Sample Data: [ 95 samples by 7 sample variables ]
tax_table() Taxonomy Table: [ 2171 taxa by 7 taxonomic ranks ]
phy_tree() Phylogenetic Tree: [ 2171 tips and 2170 internal nodes ]
>
> ## Add group variable
> sample_data(ps)$group <- factor(paste(sample_data(ps)$timepoint, sample_data(ps)$treatment, sep = "-"))
> alpha <- estimate_richness(ps)
> alpha_info <- sample_data(ps)
> aa <- cbind(alpha, alpha_info)
> a1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Shannon)") + xlab("Timepoint")
> a1
>
```

Plots: Displays four boxplots comparing Alpha diversity (Shannon) and Alpha diversity (Simpson) across different timepoints (base and week3) for four treatment groups (control, pre, pro, syn).

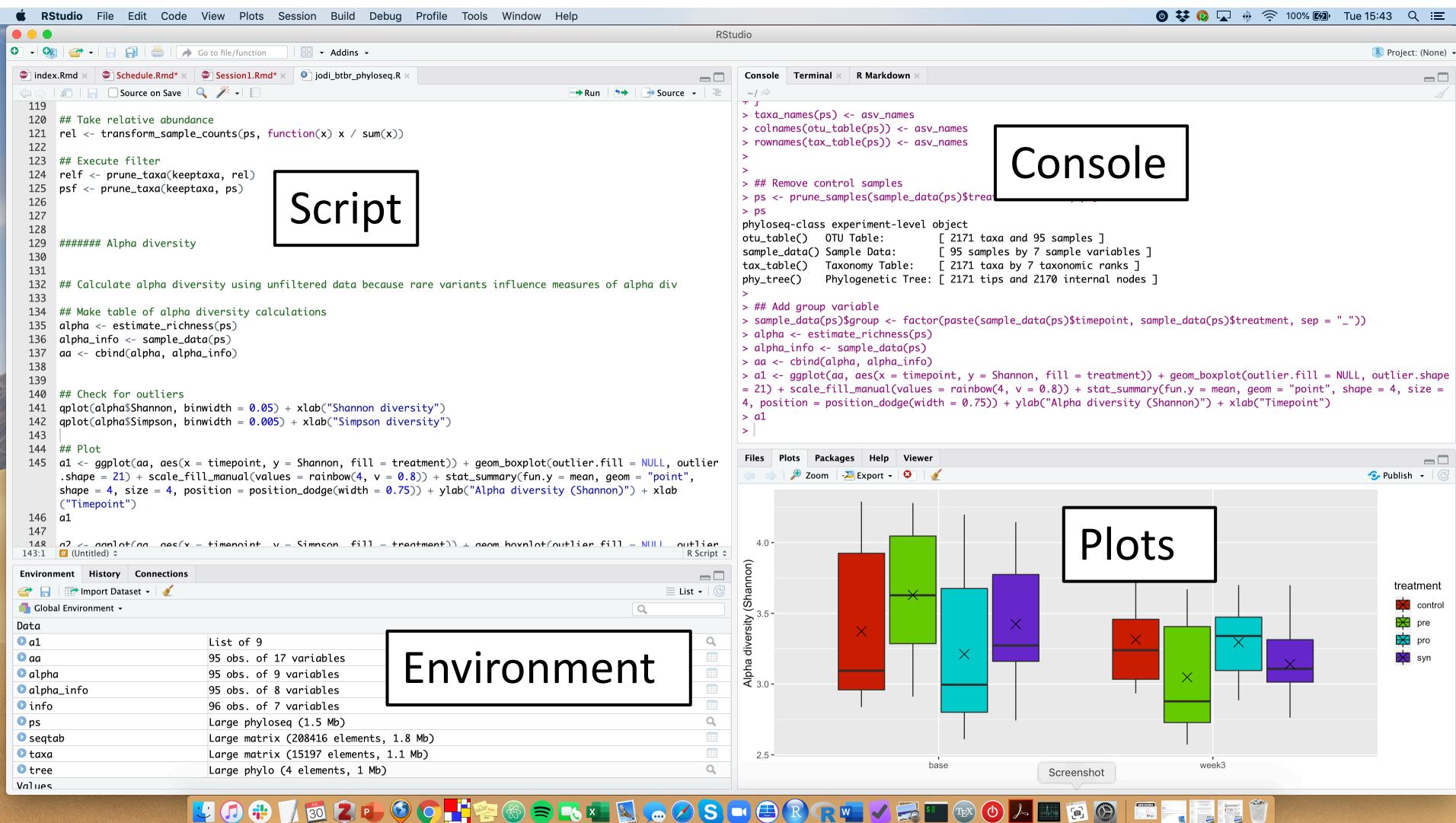
The figure consists of two rows of boxplots. The top row shows Alpha diversity (Shannon) and the bottom row shows Alpha diversity (Simpson). Each row has two boxplots: one for the 'base' timepoint and one for 'week3'. The x-axis for both rows is labeled 'timepoint' with categories 'base' and 'week3'. The y-axis is labeled 'Alpha diversity (Shannon)' or 'Alpha diversity (Simpson)' depending on the row. Each boxplot is colored by treatment: control (red), pre (green), pro (cyan), and syn (purple). The median value is indicated by a horizontal line inside each box, and individual data points are shown as 'x' marks. The 'control' group consistently shows the highest diversity across all timepoints and metrics.

Environment: Shows the global environment with objects like a1, aa, alpha, alpha_info, info, ps, seqtab, taxa, tree, and Values.

File Bar: Shows open files: index.Rmd, Schedule.Rmd*, Session1.Rmd*, and jodi_btbr_phlyoseq.R.

System: Shows the Mac OS X dock with various application icons.

RStudio screen



RStudio screen

RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help

Project: (None)

index.Rmd Schedule.Rmd Session1.Rmd jodi_btbr_phlyoseq.R

Run Source

Script

```
119  
120 ## Take relative abundance  
121 rel <- transform_sample_counts(ps, function(x) x / sum(x))  
122  
123 ## Execute filter  
124 relf <- prune_taxa(keptaxa, rel)  
125 psf <- prune_taxa(keptaxa, ps)  
126  
127  
128 ##### Alpha diversity  
129  
130 ## Calculate alpha diversity using unfiltered data because rare variants influence measures of diversity  
131  
132 ## Make table of alpha diversity calculations  
133 alpha <- estimate_richness(psf)  
134 alpha_info <- sample_data(psf)  
135 aa <- cbind(alpha, alpha_info)  
136  
137 ## Check for outliers  
138 qplot(alpha$Shannon, x = timepoint, y = Shanno, fill = treatment) + xlab("Shannon diversity")  
139 qplot(alpha$Simpson, x = timepoint, y = Simpson, fill = treatment) + xlab("Simpson diversity")  
140  
141 ## Plot  
142 a1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Shannon)") + xlab("Timepoint")  
143 a1  
144 o1 <- ggplot(aa, aes(x = timepoint, y = Simpson, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Simpson)") + xlab("Timepoint")  
145 o1  
146 o2 <- ggplot(aa, aes(x = timepoint, y = Simpson, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Simpson)") + xlab("Timepoint")  
147 o2  
148 o3 <- ggplot(aa, aes(x = timepoint, y = Simpson, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Simpson)") + xlab("Timepoint")  
149 o3  
150
```

Console Terminal R Markdown

```
> s  
> taxa_names(ps) <- asv_names  
> colnames(otu_table(ps)) <- asv_names  
> rownames(tax_table(ps)) <- asv_names  
>  
>  
> ## Remove control samples  
> ps <- prune_samples(sample_data(ps)$tre  
> ps  
phyloseq-class experiment-level object  
otu_table() OTU Table: [ 2171 taxa and 95 samples ]  
sample_data() Sample Data: [ 95 samples by 7 sample variables ]  
tax_table() Taxonomy Table: [ 2171 taxa by 7 taxonomic ranks ]  
phy_tree() Phylogenetic Tree: [ 2171 tips and 2170 internal nodes ]  
>  
> ## Add group variable  
> sample_data(ps)$group <- factor(paste(sample_data(ps)$timepoint, sample_data(ps)$treatment, sep = "-"))  
> alpha <- estimate_richness(ps)  
> alpha_info <- sample_data(ps)  
> aa <- cbind(alpha, alpha_info)  
> a1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Shannon)") + xlab("Timepoint")  
> a1  
>
```

Files Plots Packages Help Viewer

Plots

Environment

Global Environment

Data

- a1 List of 9
- aa 95 obs. of 17 variables
- alpha 95 obs. of 9 variables
- alpha_info 95 obs. of 8 variables
- info 96 obs. of 7 variables
- ps Large phyloseq (1.5 Mb)
- seqtab Large matrix (208416 elements, 1.8 Mb)
- taxa Large matrix (15197 elements, 1.1 Mb)
- tree Large phylo (4 elements, 1 Mb)

Values

Screenshot

control pre pro syn

RStudio screen

The screenshot shows the RStudio interface on a Mac OS X desktop. The window title is "RStudio". The menu bar includes "File", "Edit", "Code", "View", "Plots", "Session", "Build", "Debug", "Profile", "Tools", "Window", and "Help". The status bar at the bottom right shows "100% 16:27 Tue 16:27".

Script: The left pane displays an R script with code for phylogenetic analysis. It includes sections for relative abundance, filtering, alpha diversity calculations, and creating a phyloseq object. A box labeled "Script" highlights the script area.

```
119 ## Take relative abundance
120 rel <- transform_sample_counts(ps, function(x) x / sum(x))
121
122 ## Execute filter
123 relf <- prune_taxa(keptaxa, rel)
124 psf <- prune_taxa(keptaxa, ps)
125
126
127
128 ###### Alpha diversity
129
130
131
132 ## Calculate alpha diversity using unfiltered data because rare variants influence measures of alpha div
133
134 ## Make table of alpha diversity calculations
135 alpha <- estimate_richness(ps)
136 alpha_info <- sample_data(ps)
137 aa <- cbind(alpha, alpha_info)
138
138:1 (Untitled):
```

Console: The top right pane shows the R console output. A box labeled "Console" highlights the output area.

```
> taxa_names(ps) <- asv_names
> colnames(otu_table(ps)) <- asv_names
> rownames(tax_table(ps)) <- asv_names
>
>
> ## Remove control samples
> ps <- prune_samples(sample_data(ps)$tre
> ps
phyloseq-class experiment-level object
otu_table() OTU Table: [ 2171 taxa and 95 samples ]
sample_data() Sample Data: [ 95 samples by 7 sample variables ]
tax_table() Taxonomy Table: [ 2171 taxa by 7 taxonomic ranks ]
phy_tree() Phylogenetic Tree: [ 2171 tips and 2170 internal nodes ]
>
> ## Add group variable
> sample_data(ps)$group <- factor(paste(sample_data(ps)$timepoint, sample_data(ps)$treatment, sep = "_"))
> alpha <- estimate_richness(ps)
> alpha_info <- sample_data(ps)
> aa <- cbind(alpha, alpha_info)
> a1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape =
21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size =
4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Shannon)") + xlab("Timepoint")
> a1
> |
```

History: The bottom left pane shows the history of R commands run in the session. A box labeled "History" highlights the history area.

```
# MARCH Sample names
rownames(seqtab)
# Make a phyloseq object
ps <- phyloseq(otu_table(seqtab, taxa_are_rows=FALSE), sample_data(info), tax_table(taxa))
## Make a tree and add the tree to a new phyloseq object
tree <- rtree(ntaxa(ps), rooted = TRUE, tip.label = taxa_names(ps))
ps <- phyloseq(otu_table(seqtab, taxa_are_rows=FALSE), sample_data(info), tax_table(taxa), phy_tree(tree))
asv_names <- vector(dim(otu_table(ps))[2], mode = "character")
for (i in 1:dim(otu_table(ps))[2]){
  asv_names[i] <- paste("ASV", i, sep = "_")
}
taxa_names(ps) <- asv_names
colnames(otu_table(ps)) <- asv_names
rownames(tax_table(ps)) <- asv_names
## Remove control samples
ps <- prune_samples(sample_data(ps)$treatment != "NA", ps)
ps
## Add group variable
sample_data(ps)$group <- factor(paste(sample_data(ps)$timepoint, sample_data(ps)$treatment, sep = "_"))
alpha <- estimate_richness(ps)
alpha_info <- sample_data(ps)
aa <- cbind(alpha, alpha_info)
a1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape =
a1
```

Files: The bottom right pane shows the file browser. A box labeled "Files" highlights the file browser area.

Files	Plots	Packages	Help	Viewer
New Folder	Home	Applications	Help	Viewer
1.1 KB	Feb 11, 2019, 5:41 PM			

Screenshot

The dock at the bottom of the screen contains icons for various Mac OS X applications, including Mail, Safari, and Finder.

How to R – 2 ways

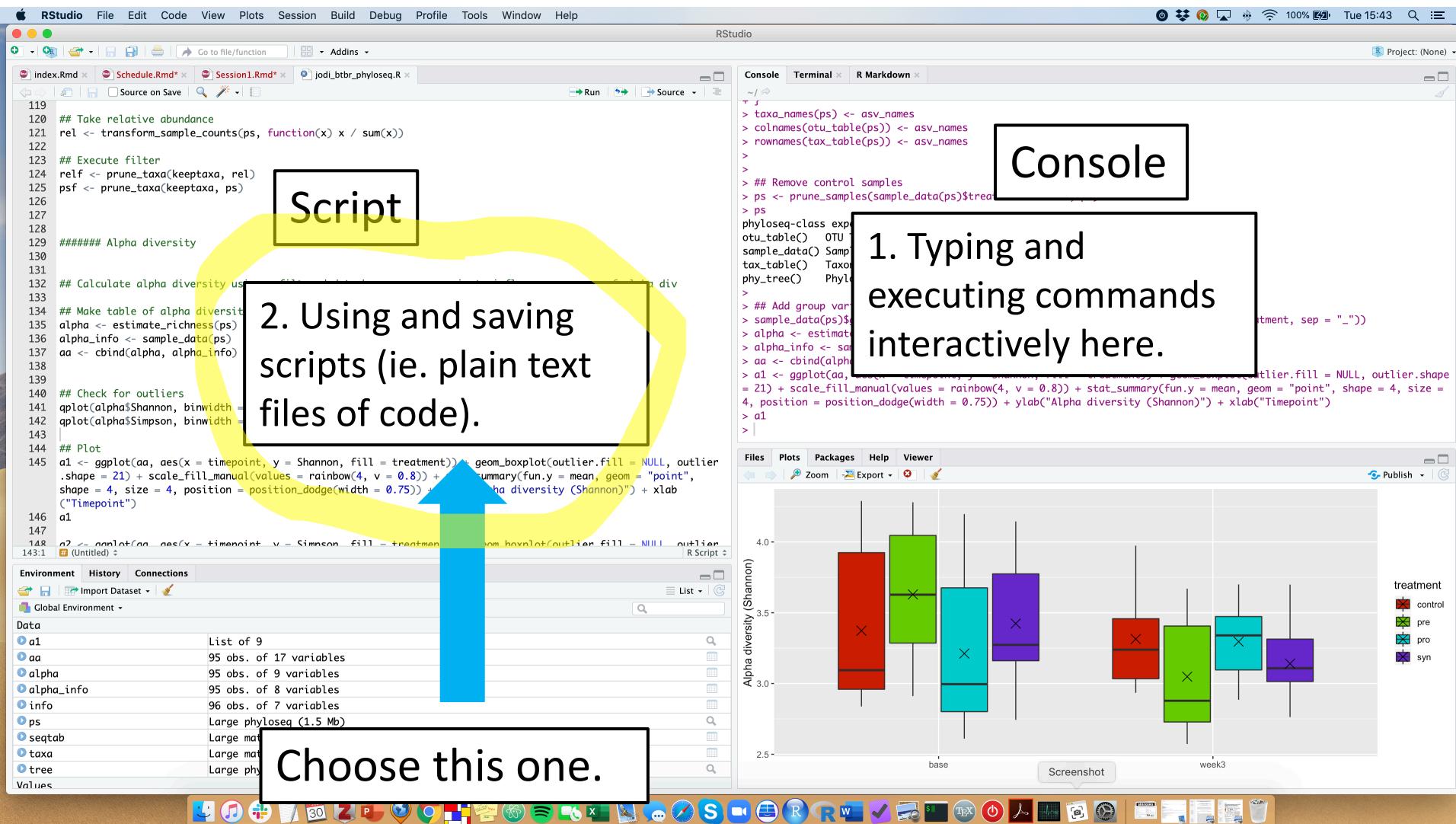
The screenshot shows the RStudio interface with several panels:

- Script Editor:** Shows a script named "jodi_btbr_phyloseq.R" with code related to alpha diversity calculations and phylogenetic tree analysis.
- Console:** Shows the R command history, including the execution of various functions like `taxa_names`, `colnames`, `rownames`, and `phyloseq-class`.
- Plots:** A boxplot titled "Alpha diversity (Shannon)" comparing diversity across timepoints (base, week3) and treatments (control, pre, pro, syn).
- Environment:** Shows the global environment with objects like `a1`, `aa`, `alpha`, `alpha_info`, `info`, `ps`, `seqtab`, `taxa`, and `tree`.

Annotations highlight sections of the script and the plot:

- A box labeled "Script" highlights the first section of the script.
- A box labeled "Console" highlights the command history.
- A box labeled "1. Typing and executing commands interactively here." highlights the command history area.
- A box labeled "2. Using and saving scripts (ie. plain text files of code)." highlights the script editor area.

How to R – 2 ways



Scripts

```
1 # jodi_btbr project, Alana Schick, April 2019
2 # This is a script to analyze the output tables of the DADA2 workflow in phyloseq
3 # Have two output files from dada2 - a sequence table and a taxonomy table, read them into R using the readRDS() function
4 # The formatted sample metadata is in a table called "jodi_btbr_metadata.txt"
5
6 library(phyloseq)
7 #packageVersion("phyloseq")
8 library(ggplot2)
9 #packageVersion("ggplot2")
10 library(ape)
11 library(viridis)
12 library(grid)
13 library(gridExtra)
14 library(reshape2)
15 library(DESeq2)
16 library(fields)
17 library(vegan)
18 library(ggpubr)
19 library(plyr)
20 library(RColorBrewer)
21
22 path_to_project <- "/Users/alanaschick/Dropbox/Jodi_BTBR"
23
24 # Read in files
25 seqtab <- readRDS(file.path(path_to_project, "seqtab.rds"))
26 taxa <- readRDS(file.path(path_to_project, "taxa.rds"))
27 info <- read.table(file.path(path_to_project, "jodi_btbr_metadata.txt"), header = TRUE)
28
29 # Match sample names
30 rownames(info) <- rownames(seqtab)
31
32 # Make a phyloseq object
```

Everything in the console will be forgotten when you close the session.

Scripts are saved, keeping a complete record of the commands you ran so you can run them again (ie. completely reproducible).

Can execute parts of this or the entire script.

Scripts - commenting

```
1 # jodi_btbr project, Alana Schick, April 2019
2 # This is a script to analyze the output tables of the DADA2 workflow in phyloseq
3 # Have two output files from dada2 - a sequence table and a taxonomy table, read them into R using the readRDS()
4 # The formatted sample metadata is in a table called "jodi_btbr_metadata.txt"
5
6 library(phyloseq)
7 #packageVersion("phyloseq")
8 library(ggplot2)
9 #packageVersion("ggplot2")
10 library(ape)
11 library(viridis)
12 library(grid)
13 library(gridExtra)
14 library(reshape2)
15 library(DESeq2)
16 library(fields)
17 library(vegan)
18 library(ggpubr)
19 library(plyr)
20 library(RColorBrewer)
21
22 path_to_project <- "/Users/alanaschick/Dropbox/time/projects/jodi_btbr"
23
24 # Read in files
25 seqtab <- readRDS(file.path(path_to_project, "results/seqtan_final.rds"))
26 taxa <- readRDS(file.path(path_to_project, "results/taxa_final.rds"))
27 info <- read.table(file.path(path_to_project, "jodi_btbr_metadata2.txt"), header = TRUE)
28
29 # Match sample names
30 rownames(info) <- rownames(seqtan)
31
32 # Make a phyloseq object
```

Comment out lines of your scripts by using the `#` symbol. R will not run these.

Be descriptive. You will not remember what you did a year later.

Packages

```
1 # jodi_btbr project, Alana Schick, April 2019
2 # This is a script to analyze the output tables of the DADA2 workflow in phyloseq
3 # Have two output files from dada2 - a sequence table and a taxonomy table, read them into R using the readRDS() function
4 # The formatted sample metadata is in a table called "jodi_btbr_metadata.txt"
5
6 library(phyloseq)
7 #packageVersion("phyloseq")
8 library(ggplot2)
9 #packageVersion("ggplot2")
10 library(ape)
11 library(viridis)
12 library(grid)
13 library(gridExtra)
14 library(reshape2)
15 library(DESeq2)
16 library(fields)
17 library(vegan)
18 library(ggpubr)
19 library(plyr)
20 library(RColorBrewer)
21
22 path_to_project <- "/Users/alanaschick/"
23
24 # Read in files
25 seqtab <- readRDS(file.path(path_to_project, "results/seqtan_final.rds"))
26 taxa <- readRDS(file.path(path_to_project, "results/taxa_final.rds"))
27 info <- read.table(file.path(path_to_project, "jodi_btbr_metadata2.txt"), header = TRUE)
28
29 # Match sample names
30 rownames(info) <- rownames(seqtan)
31
32 # Make a phyloseq object
```

Packages are collections of R functions developed for a specific task.

Packages need to first be installed on your computer.

After installed, `library()` is the command used to load a package.

Packages

A screenshot of the RStudio interface showing a code editor, console, and plot area. A large blue arrow points from the top left towards the 'Tools' menu.

The code editor shows R script content:

```
119
120 ## Take relative abundance
121 rel <- transform_sample_counts(ps, function(x) x / sum(x))
122
123 ## Execute filter
124 relf <- prune_taxa(keptaxa, rel)
125 psf <- prune_taxa(keptaxa, ps)
126
127
128 ###### Alpha diversity
129
130 ## Calculate alpha diversity using unfiltered data because
131
132 ## Make table of alpha diversity calculations
133 alpha <- estimate_richness(ps)
134 alpha.info <- sample_data(ps)
135 aa <- cbind(alpha, alpha.info)
136
137 ## Check for outliers
138 qplot(alpha$Shannon, binwidth = 0.05) + xlab("Shannon diversity")
139 qplot(alpha$Simpson, binwidth = 0.005) + xlab("Simpson diversity")
140
141 ## Plot
142 o1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment,
143 .shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) +
144 shape = 4, size = 4, position = position_dodge(width = 0.75)) +
145 ("Timepoint")
146 o1
147
148 o2 <- ggplot(aa, aes(x = timepoint, y = Simpson, fill = treatment,
149 .shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) +
150 shape = 4, size = 4, position = position_dodge(width = 0.75)) +
151 ("Timepoint")
152 o2
```

The console shows R session output:

```
119
120 ## Take relative abundance
121 rel <- transform_sample_counts(ps, function(x) x / sum(x))
122
123 ## Execute filter
124 relf <- prune_taxa(keptaxa, rel)
125 psf <- prune_taxa(keptaxa, ps)
126
127
128 ###### Alpha diversity
129
130 ## Calculate alpha diversity using unfiltered data because
131
132 ## Make table of alpha diversity calculations
133 alpha <- estimate_richness(ps)
134 alpha.info <- sample_data(ps)
135 aa <- cbind(alpha, alpha.info)
136
137 ## Check for outliers
138 qplot(alpha$Shannon, binwidth = 0.05) + xlab("Shannon diversity")
139 qplot(alpha$Simpson, binwidth = 0.005) + xlab("Simpson diversity")
140
141 ## Plot
142 o1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment,
143 .shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) +
144 shape = 4, size = 4, position = position_dodge(width = 0.75)) +
145 ("Timepoint")
146 o1
147
148 o2 <- ggplot(aa, aes(x = timepoint, y = Simpson, fill = treatment,
149 .shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) +
150 shape = 4, size = 4, position = position_dodge(width = 0.75)) +
151 ("Timepoint")
```

The plot area displays two boxplots side-by-side. The x-axis categories are 'base' and 'week3'. The y-axis ranges from 2.5 to 3.5. The legend indicates four treatment groups: 'control' (red), 'pre' (green), 'pro' (cyan), and 'syn' (purple). Each boxplot shows the median, quartiles, and whiskers, with individual data points overlaid as 'x' marks.

A modal dialog box titled 'Install Packages' is open in the foreground:

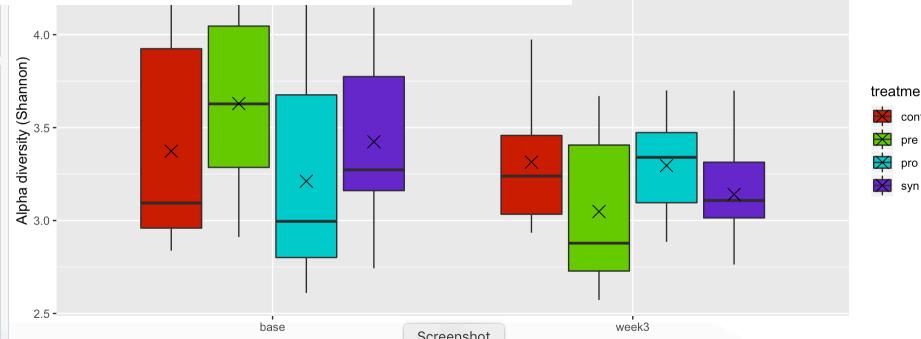
- Install from:** Repository (CRAN)
- Packages (separate multiple with space or comma):** (empty input field)
- Install to Library:** /Library/Frameworks/R.framework/Versions/3.6/Resources/library
- Install dependencies:**
- Buttons:** Install, Cancel

Working directory

Every time you open RStudio, it goes to a default directory, usually your home directory.

You can use the command **setwd()** to change the working directory.

```
setwd("home/aschick/projects/workshop")
```



Working directory

RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help RStudio Addins Go to file/function Addins ~ Run Source ~ j Project: (None) 100% Tue 15:43

```
119 ## Take relative abundance
120 rel <- transform_sample_counts(ps, function(x)
121
122 ## Execute filter
123 relf <- prune_taxa(keptaxa, rel)
124 psf <- prune_taxa(keptaxa, ps)
125
126
127
128 ###### Alpha diversity
129
130 ## Calculate alpha diversity using unfiltered data
131
132 ## Make table of alpha diversity calculations
133 alpha <- estimate_richness(psf)
134 alpha_info <- sample_data(psf)
135 aa <- cbind(alpha, alpha_info)
136
137
138 ## Check for outliers
139 qplot(alpha$Shannon, binwidth = 0.05) + xlab("Timepoint")
140 qplot(alpha$Simpson, binwidth = 0.005) + xlab("Timepoint")
141
142 ## Plot
143 o1 <- ggplot(aa, aes(x = timepoint, y = Shannon))
144 o1 + geom_boxplot(outlier.size = 0.75) + stat_summary(fun.y = mean, geom = "point",
145 .shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point",
146 shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Shannon)") + xlab(
147 "Timepoint")
148 o2 <- ggplot(aa, aes(x = timepoint, y = Simpson))
149 o2 + geom_boxplot(outlier.size = 0.75) + stat_summary(fun.y = mean, geom = "point",
150 .shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point",
151 shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Simpson)") + xlab(
152 "Timepoint")
153
```

However: you may want to run your script on a different computer with a different directory structure where that directory does not exist.

Or you may want to work in multiple directories.

Alpha diversity (Shannon)

control pre pro syn

Screenshot

RStudio Project

File > New Project...

Clicking on New Directory will create an RStudio Project.

This directory will have all the data, files, plots, etc. for that project as well as a .Rproj file.

The screenshot shows the RStudio interface with a blue arrow pointing to the 'File' menu. A callout box highlights 'File > New Project...'. Below it, another callout box highlights the text 'Clicking on New Directory will create an RStudio Project.' and 'This directory will have all the data, files, plots, etc. for that project as well as a .Rproj file.' In the center, a 'New Project' dialog box is open, showing three options: 'New Directory' (Start a project in a brand new working directory), 'Existing Directory' (Associate a project with an existing working directory), and 'Version Control' (Checkout a project from a version control repository). The background shows code in the editor and a plot in the viewer pane.

Error messages

Console Terminal R Markdown

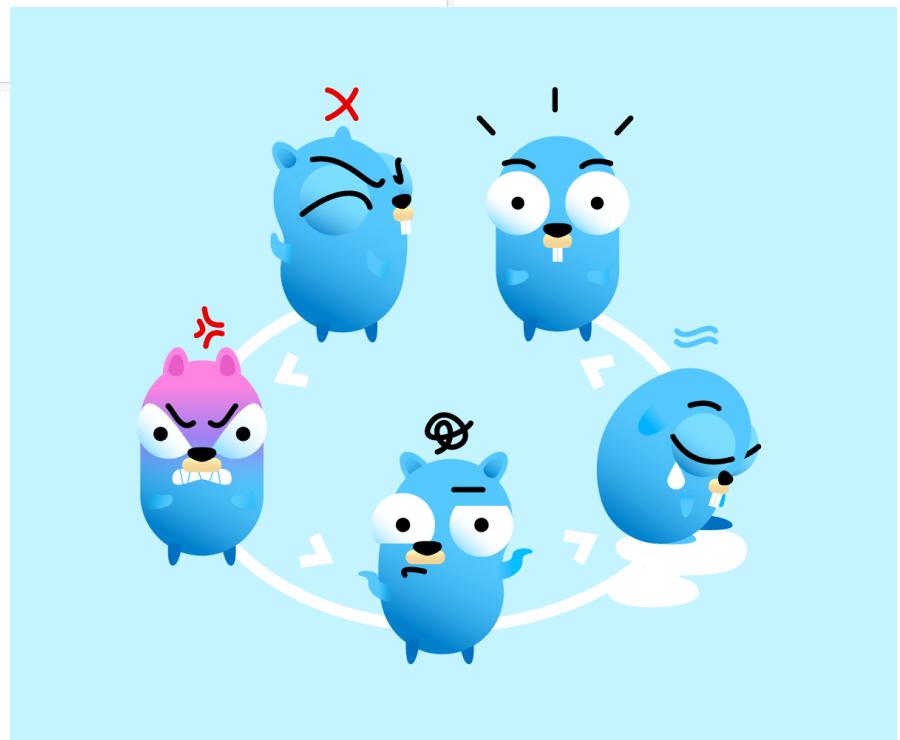
~ / ↻

```
>
> ## Remove control samples
> ps <- prune_samples(sample_data(ps)$treatment != "NA", ps)
> ps
phyloseq-class experiment-level object
otu_table() OTU Table: [ 2171 taxa and 95 samples ]
sample_data() Sample Data: [ 95 samples by 7 sample variables ]
tax_table() Taxonomy Table: [ 2171 taxa by 7 taxonomic ranks ]
phy_tree() Phylogenetic Tree: [ 2171 tips and 2170 internal nodes ]
>
> ## Add group variable
> sample_data(ps)$group <- factor(paste(sample_data(ps)$timepoint, sample_data(ps)$treatment, sep = "_"))
> alpha <- estimate_richness(ps)
> alpha_info <- sample_data(ps)
> aa <- cbind(alpha, alpha_info)
> a1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Shannon)") + xlab("Timepoint")
> a1
> ord1 <- ordinate(relf, method = "NMDS", distance = "bray")
Error in ordinate(relf, method = "NMDS", distance = "bray") :
  object 'relef' not found
> b1 <- plot_ordination(relf, ord1, color = "timepoint", shape = "treatment", title = "NMDS - Bray") + scale_colour_manual(values = viridis(3))
Error in plot_ordination(relf, ord1, color = "timepoint", shape = "treatment", :
  object 'relef' not found
> b1
Error: object 'b1' not found
> |
```

Error messages

```
> aa <- cobra_alpha_intro  
> a1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape  
= 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size =  
4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Shannon)") + xlab("Timepoint")  
> a1  
> ord1 <- ordinate(relf, method = "NMDS", distance = "bray")  
Error in ordinate(relf, method = "NMDS", distance = "bray") :  
  object 'relf' not found  
> b1 <- plot_ordination(relf, ord1, color = "timepoint", shape = "treatment", title = "NMDS - Bray") + scale_colour_man  
ual(values = viridis(3))  
Error in plot_ordination(relf, ord1, color = "timepoint", shape = "treatment", :  
  object 'relf' not found  
> b1  
Error: object 'b1' not found  
> |
```

Error handling stages



Getting help

RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help

Project: (None)

```
index.Rmd * Schedule.Rmd * Session1.Rmd * jodi_btbr_phlyoseq.R
```

Source on Save Run Source

119
120 ## Take relative abundance
121 rel <- transform_sample_counts(ps, function(x) x / sum(x))
122
123 ## Execute filter
124 relf <- prune_taxa(keptaxa, rel)
125 psf <- prune_taxa(keptaxa, ps)
126
127
128 ##### Alpha diversity
129
130
132 ## Calculate alpha diversity using unfiltered data because rare variants influence measures of alpha div
133
134 ## Make table of alpha diversity calculations
135 alpha <- estimate_richness(ps)
136 alpha_info <- sample_data(ps)
137 aa <- cbind(alpha, alpha_info)
138
139 ## Check for outliers
140 qplot(alpha\$Shannon, binwidth = 0.05) + xlab("Shannon diversity")
141 qplot(alpha\$Simpson, binwidth = 0.005) + xlab("Simpson diversity")
143
144 ## Plot
145 a1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Shannon)") + xlab("Timepoint")
146 a1
147
148 a2 <- ggplot(aa, aes(x = timepoint, y = Simpson, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Simpson)") + xlab("Timepoint")
143:1 (Untitled) R Script

Console Terminal R Markdown

```
> s  
> taxa_names(ps) <- asv_names  
> colnames(otu_table(ps)) <- asv_names  
>  
> ps  
phyloseq-class experiment-level object  
@OTU_table() OTU Table: [ 2171 taxa and 95 samples ]  
@Sample_data() Sample Data: [ 95 samples by 7 sample variables ]  
@Table() Taxonomy Table: [ 2171 taxa by 7 taxonomic ranks ]  
@Phylo() Phylogenetic Tree: [ 2171 tips and 2170 internal nodes ]  
  
> s  
> sample_data(ps)$group <- factor(paste(sample_data(ps)$timepoint, sample_data(ps)$treatment, sep = "-"))  
> alpha <- estimate_richness(ps)  
> alpha_info <- sample_data(ps)  
> aa <- cbind(alpha, alpha_info)  
> a1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Shannon)") + xlab("Timepoint")  
> a1
```

Files Plots Packages Help Viewer

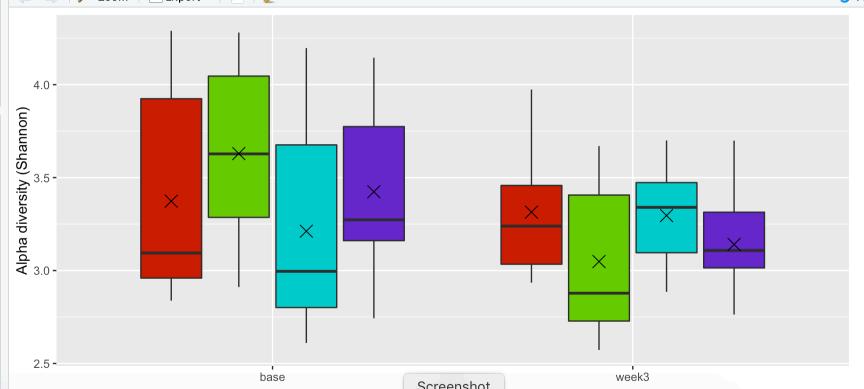
Zoom Export Publish

Alpha diversity (Shannon)

base week3

treatment

- control
- pre
- pro
- syn



Screenshot

Environment History Connections

Import Dataset

Global Environment

Data

- a1 List of 9
- aa 95 obs. of 17 variables
- alpha 95 obs. of 9 variables
- alpha_info 95 obs. of 8 variables
- info 96 obs. of 7 variables
- ps Large phyloseq (1.5 Mb)
- seqtab Large matrix (208416 elements, 1.8 Mb)
- taxa Large matrix (15197 elements, 1.1 Mb)
- tree Large phylo (4 elements, 1 Mb)

Values

Mac OS X Dock icons: Mail, Music, Calendar, Google Chrome, Spotify, iMovie, iPhoto, iWork, R, RStudio, WPS Office, TeX, Arrows, Screenshots, etc.

Getting help

The screenshot shows the RStudio interface with several panels:

- Code Editor:** Displays R code for alpha diversity calculations and boxplots.
- Console:** Shows the command `?geom_point` being typed.
- Help Tab:** A large callout box highlights the first two methods for finding help: "Search in Help tab" and "Type ? followed by the function name in the console (or ?? for installed packages)".
- Environment:** Shows the global environment with objects like `a1`, `aa`, `alpha`, etc.
- Plots:** Two boxplots comparing Alpha diversity (Shannon and Simpson) across timepoints (base and week3) for four treatments (control, pre, pro, syn).

Getting help

- 1) Search in Help tab
- 2) Type ? followed by the function name in the console (or ?? for installed packages)
- 3) Google the error message



See website for tips and resources!

The internet will make those bad words go away



Essential

Googling the
Error Message

ORLY?

*The Practical Developer
@ThePracticalDev*

Summary and best practices



*See <http://adv-r.had.co.nz/Style.html> for tips.

