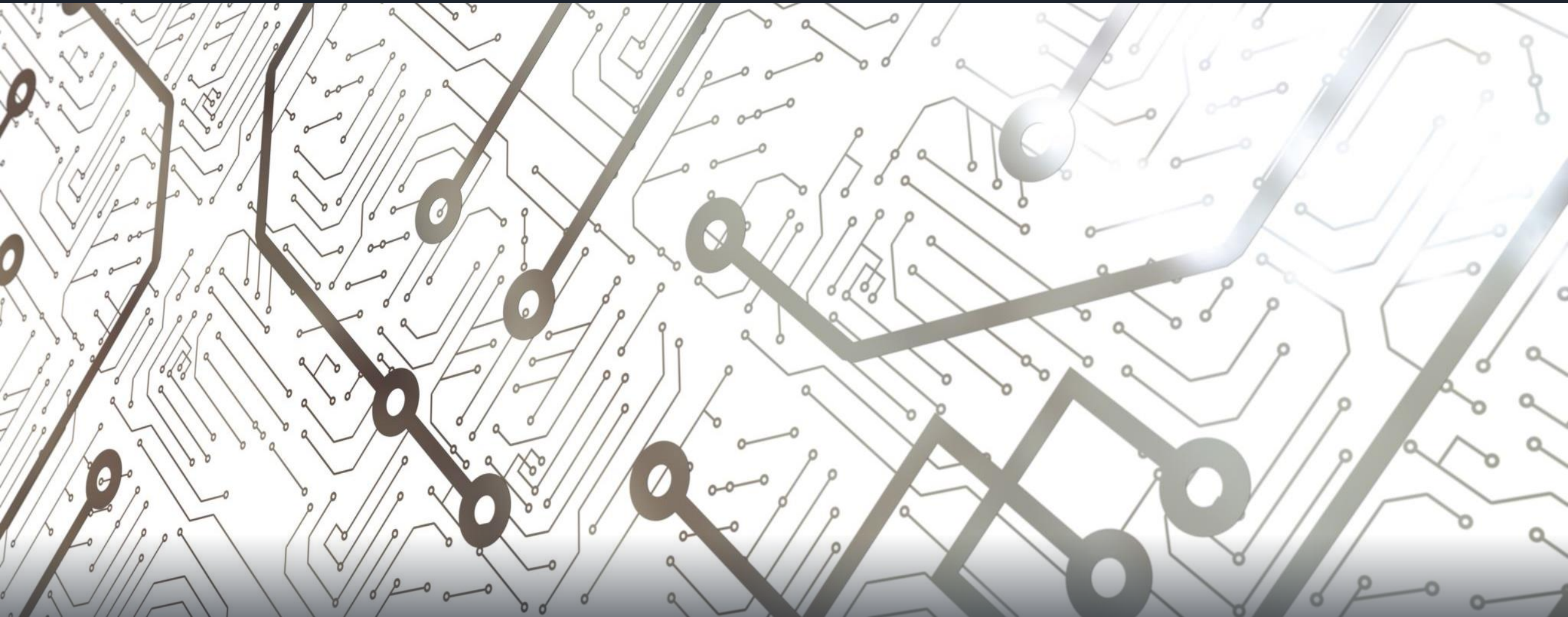


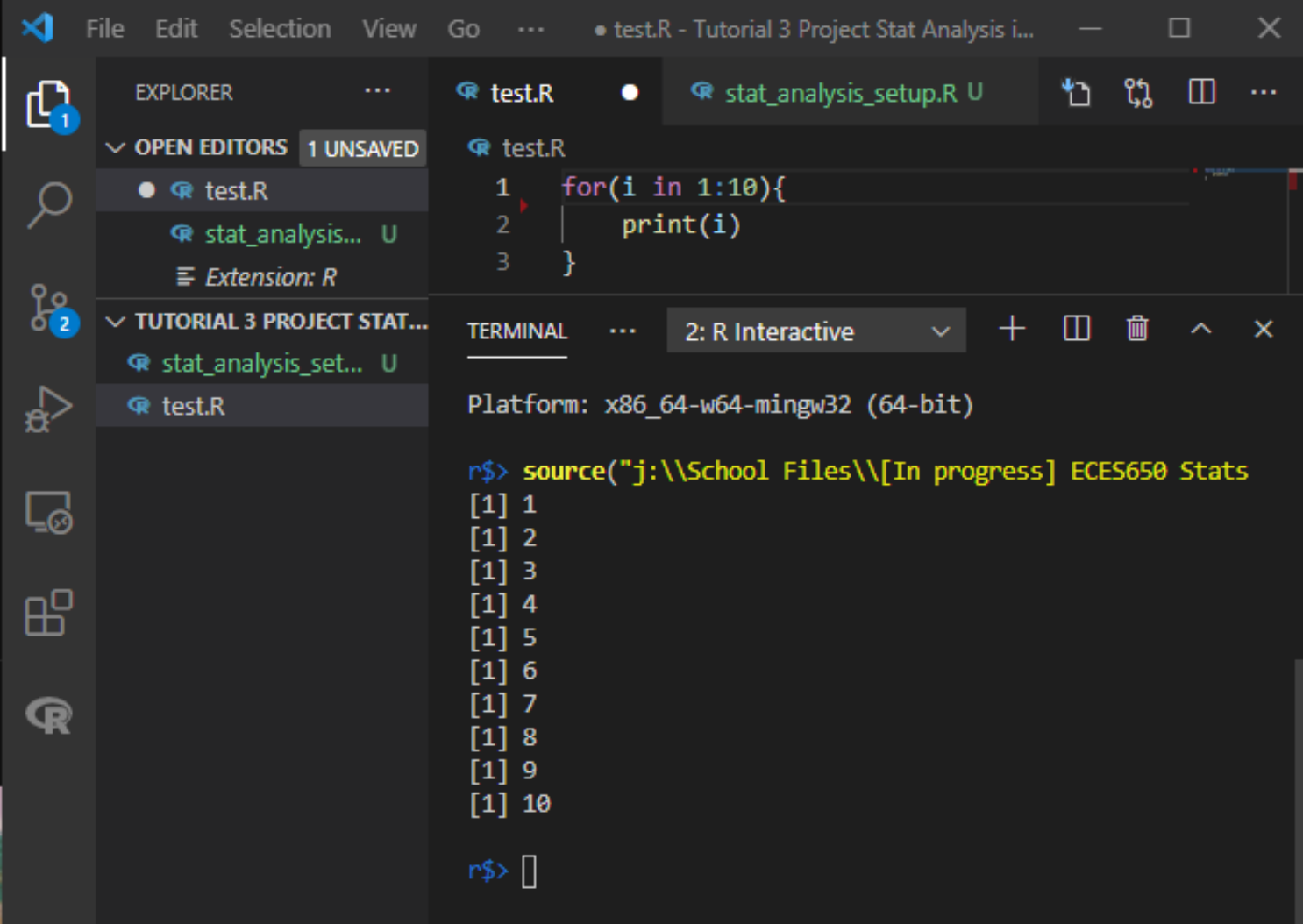
Tutorial 3: Statistical Testing: Group Comparisons and Multiple Comparison Corrections in R

Roberto Baratta
&
Cameron Calv



Installing R

- Choose OS from this link and install : <https://cran.r-project.org/bin/>
- Decide upon an environment to use (This slide is in VSCode)
 - VSCode uses two extensions for R, namely the R Extension and the R LSP Client Extension for the *radian* terminal
- Test for proper installation:



The screenshot shows the Visual Studio Code (VS Code) interface. The top menu bar includes File, Edit, Selection, View, Go, and a search icon. The Explorer sidebar on the left shows a file named 'test.R' and a folder named 'TUTORIAL 3 PROJECT STAT...'. The Open Editors sidebar shows 'test.R' and 'stat_analysis_setup.R U'. The main editor area displays the code for 'test.R':

```
1 for(i in 1:10){
2   print(i)
3 }
```

The Terminal panel at the bottom shows the output of the R code execution. The platform is 'x86_64-w64-mingw32 (64-bit)'. The command executed is 'source("j:\\School Files\\[In progress] ECES650 Stats', and the output is a list of numbers from 1 to 10:

```
r$> source("j:\\School Files\\[In progress] ECES650 Stats
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
r$> 
```



Packages for R 4.0.5

```
## Installing packages
.cran_packages <- c("tidyverse", "cowplot", "picante", "vegan", "HMP", "dendextend", "rms", "devtools")
.bioc_packages <- c("phyloseq", "DESeq2", "microbiome", "metagenomeSeq", "ALDEx2")
.inst <- .cran_packages %in% installed.packages()
if(any(!.inst)) {
  install.packages(.cran_packages[!.inst])
}
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install(.bioc_packages, version = "3.12") #Different from the website's "3.9" since I'm using R 4.0.5
devtools::install_github("adw96/breakaway")
devtools::install_github(repo = "UVic-omics/selbal")

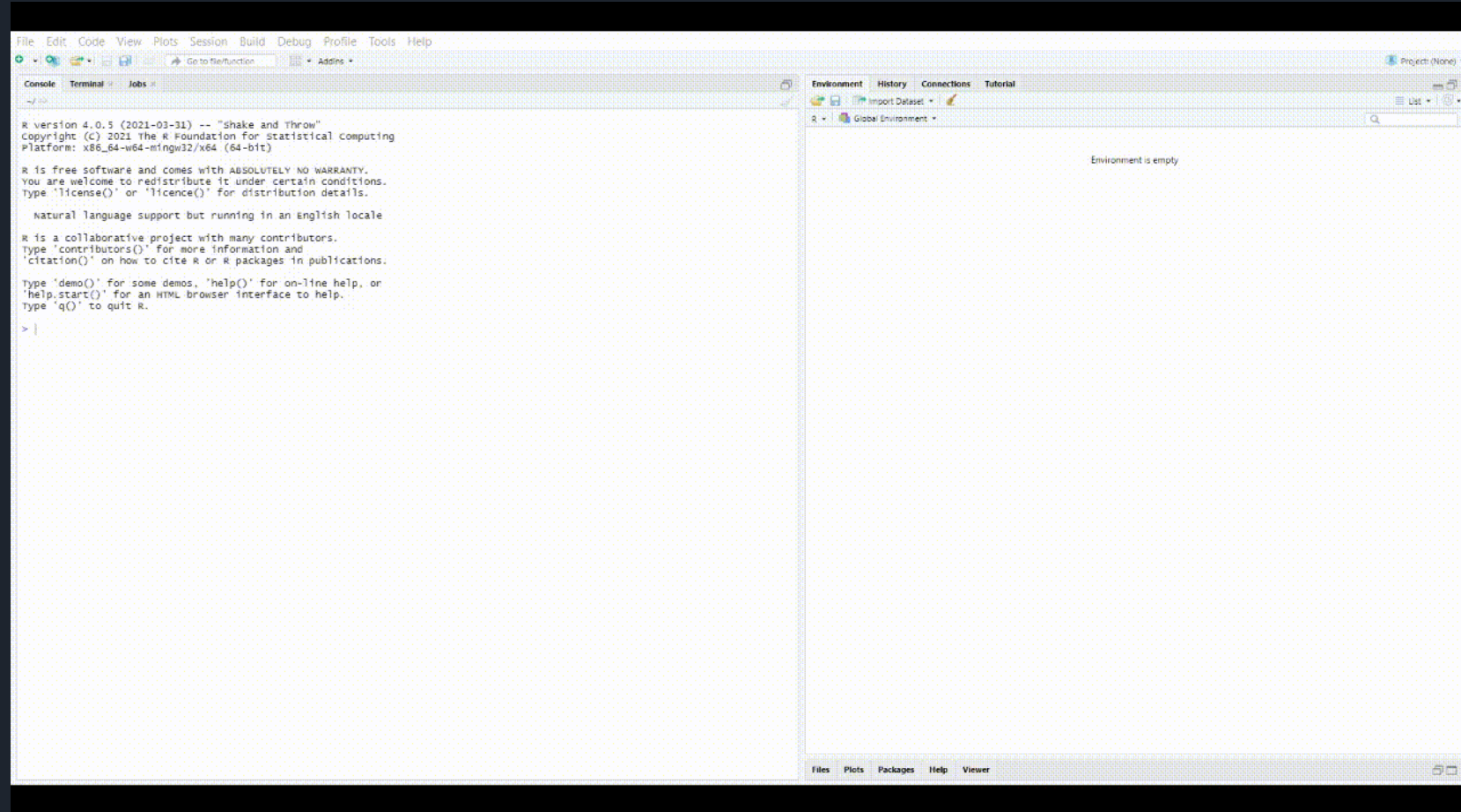
## Loading Libraries
library(tidyverse); packageVersion("tidyverse")
library(phyloseq); packageVersion("phyloseq")
library(DESeq2); packageVersion("DESeq2")
library(microbiome); packageVersion("microbiome")
library(vegan); packageVersion("vegan")
library(picante); packageVersion("picante")
library(ALDEx2); packageVersion("ALDEx2")
library(metagenomeSeq); packageVersion("metagenomeSeq")
library(HMP); packageVersion("HMP")
library(dendextend); packageVersion("dendextend")
library(selbal); packageVersion("selbal")
library(rms); packageVersion("rms")
library(breakaway); packageVersion("breakaway")
```



<https://www.nicholas-ollberding.com/post/introduction-to-the-statistical-analysis-of-microbiome-data-in-r/>

Packages for R 4.0.5

If using R Studio, to run
scripts click "Source" to
run the whole script

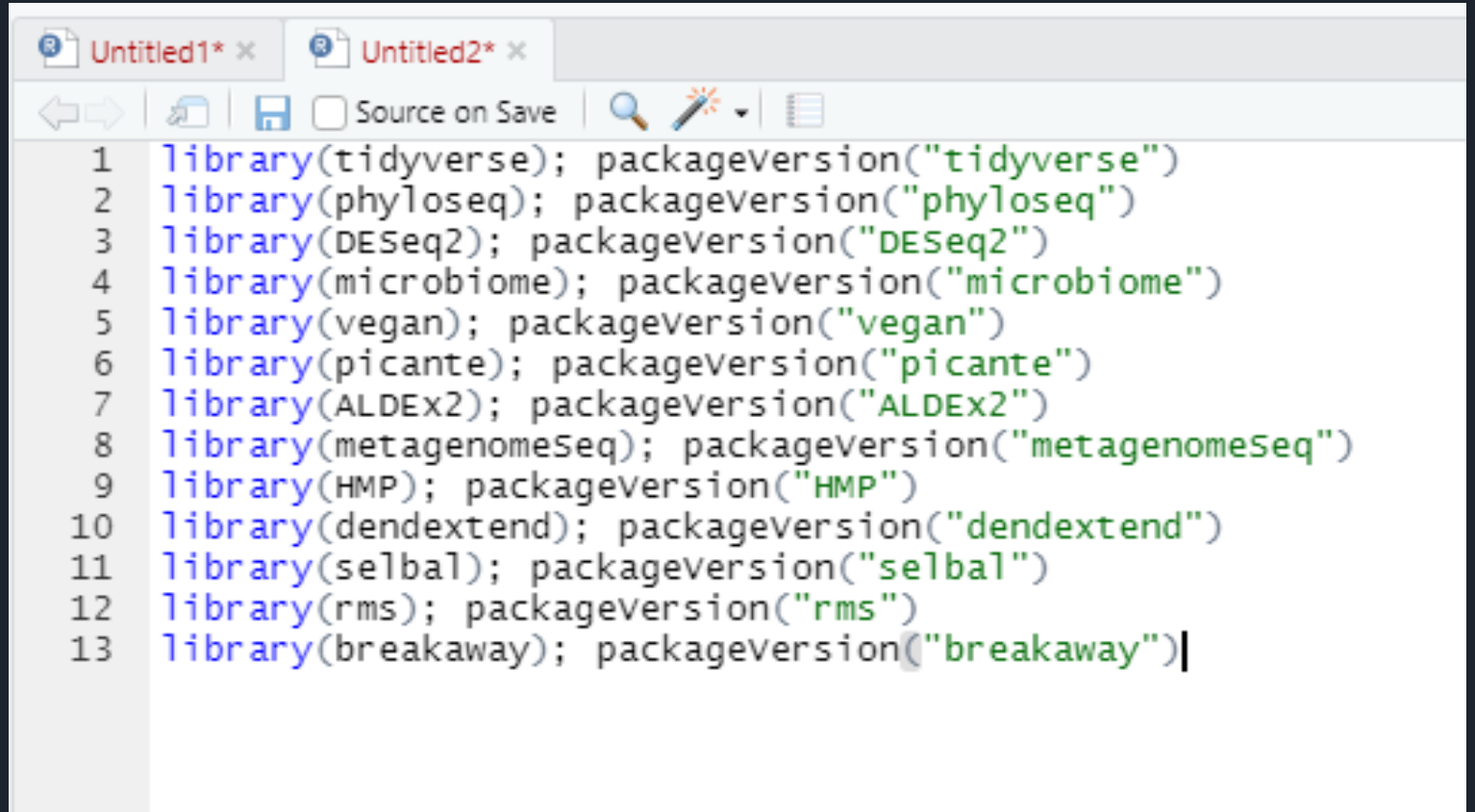


Loading & Checking Packages

It is important to check to make sure all packages were installed correctly

If a package was installed incorrectly use this command:

`install.packages("name")`



```
1 library(tidyverse); packageVersion("tidyverse")
2 library(phyloseq); packageVersion("phyloseq")
3 library(DESeq2); packageVersion("DESeq2")
4 library(microbiome); packageVersion("microbiome")
5 library(vegan); packageVersion("vegan")
6 library(picante); packageVersion("picante")
7 library(ALDEx2); packageVersion("ALDEx2")
8 library(metagenomeSeq); packageVersion("metagenomeSeq")
9 library(HMP); packageVersion("HMP")
10 library(dendextend); packageVersion("dendextend")
11 library(selbal); packageVersion("selbal")
12 library(rms); packageVersion("rms")
13 library(breakaway); packageVersion("breakaway")|
```


Obtaining Data Gut Microbial Data

- `ps_giloteaux_2016.rds` from https://github.com/Nick243/Create-Giloteaux-2016-Phyloseq-Object/blob/master/ps_giloteaux_2016.rds
 - Given as a `phyloseq` object
 - Place in same directory as the working environment

- Load data and sort by read count with:

```
(ps <- readRDS("ps_giloteaux_2016.rds"))  
sort(phyloseq::sample_sums(ps))
```

```
phyloseq-class experiment-level object  
otu_table() OTU Table: [ 138 taxa and 87 samples ]  
sample_data() Sample Data: [ 87 samples by 22 sample variables ]  
tax_table() Taxonomy Table: [ 138 taxa by 7 taxonomic ranks ]  
phy_tree() Phylogenetic Tree: [ 138 tips and 136 internal nodes ]  
refseq() DNASTringSet: [ 138 reference sequences ]  
ERR1331827 ERR1331852 ERR1331856 ERR1331869 ERR1331833 ERR1331797 ERR1331786 ERR1331818  
2707 3031 3117 5083 5245 5307 5696 5733  
ERR1331795 ERR1331846 ERR1331811 ERR1331845 ERR1331842 ERR1331838 ERR1331855 ERR1331824  
7314 7569 7665 7815 7911 8102 8115 8148  
ERR1331801 ERR1331841 ERR1331861 ERR1331820 ERR1331854 ERR1331863 ERR1331806 ERR1331787  
11173 11442 11826 12940 13029 13094 13095 13690  
ERR1331809 ERR1331828 ERR1331813 ERR1331798 ERR1331816 ERR1331830 ERR1331785 ERR1331823  
16162 16494 16749 16947 17015 17457 17557 18506  
ERR1331849 ERR1331860 ERR1331808 ERR1331872 ERR1331812 ERR1331850 ERR1331791 ERR1331788  
21540 21553 21713 22339 22518 22639 23246 23751  
ERR1331839 ERR1331794  
61206 65941
```

Phylogenetic Data in a *phyloseq* File

- Remove the samples with less than 5,000 total reads:

```
r$> (ps <- phyloseq::subset_samples(ps, phyloseq::sample_sums(ps) > 5000))
phyloseq-class experiment-level object
otu_table()   OTU Table:             [ 138 taxa and 84 samples ]
sample_data() Sample Data:          [ 84 samples by 23 sample variables ]
tax_table()   Taxonomy Table:        [ 138 taxa by 7 taxonomic ranks ]
phy_tree()    Phylogenetic Tree:     [ 138 tips and 136 internal nodes ]
refseq()      DNASTringSet:          [ 138 reference sequences ]
```

- Remove OTUs (Operational Taxonomic Units) within the remaining samples

```
r$> (ps <- phyloseq::prune_taxa(phyloseq::taxa_sums(ps) > 0, ps))
phyloseq-class experiment-level object
otu_table()   OTU Table:             [ 138 taxa and 84 samples ]
sample_data() Sample Data:          [ 84 samples by 23 sample variables ]
tax_table()   Taxonomy Table:        [ 138 taxa by 7 taxonomic ranks ]
phy_tree()    Phylogenetic Tree:     [ 138 tips and 136 internal nodes ]
refseq()      DNASTringSet:          [ 138 reference sequences ]
```

*OTUs - Similarly grouped samples not necessarily related via conventional taxonomy. Sometimes develop from sequencing errors.

Phylogenetic Data in a *phyloseq* File

- What's our data look like now after adding some metadata?

```
r$> phyloseq::sample_data(ps)$Status <- ifelse(phyloseq::sample_data(ps)$Subject == "Patient", "Chronic Fatigue", "Control")
phyloseq::sample_data(ps)$Status <- factor(phyloseq::sample_data(ps)$Status, levels = c("Control", "Chronic Fatigue"))
ps %>%
  sample_data %>%
  dplyr::count(Status)
```

```
$Status
[1] Control      Chronic Fatigue
Levels: Control Chronic Fatigue
```

```
$n
[1] 37 47
```

```
attr(,"row.names")
[1] 1 2
attr(,"class")
[1] "sample_data"
attr(,"class")attr(,"package")
[1] "phyloseq"
attr(,".S3Class")
[1] "data.frame"
```

phyloseq-class experiment-level object

```
otu_table() OTU Table: [ 138 taxa and 84 samples ]
sample_data() Sample Data: [ 84 samples by 23 sample variables ]
tax_table() Taxonomy Table: [ 138 taxa by 7 taxonomic ranks ]
phy_tree() Phylogenetic Tree: [ 138 tips and 136 internal nodes ]
refseq() DNASTringSet: [ 138 reference sequences ]
```

- 138 taxa
- 84 samples (37 controls, 47 patients)
 - Control: without chronic fatigue
 - Patient: with chronic fatigue

Visualizing Data

- Create a phylum-level table:

```
r$> #Visualization of the data
table(phyloseq::tax_table(ps)[, "Phylum"])
ps_rel_abund = phyloseq::transform_sample_counts(ps, function(x){x / sum(x)})
```

Actinobacteria	Bacteroidetes	Cyanobacteria	Euryarchaeota	Firmicutes	Fusobacteria	Proteobacteria	Tenericutes	Verrucomicrobia
7	11	2	1	105	1	7	2	1

- Then view as relative abundances:

```
r$> phyloseq::otu_table(ps)[1:5, 1:5]
phyloseq::otu_table(ps_rel_abund)[1:5, 1:5]
```

OTU Table: [5 taxa and 5 samples]
taxa are rows

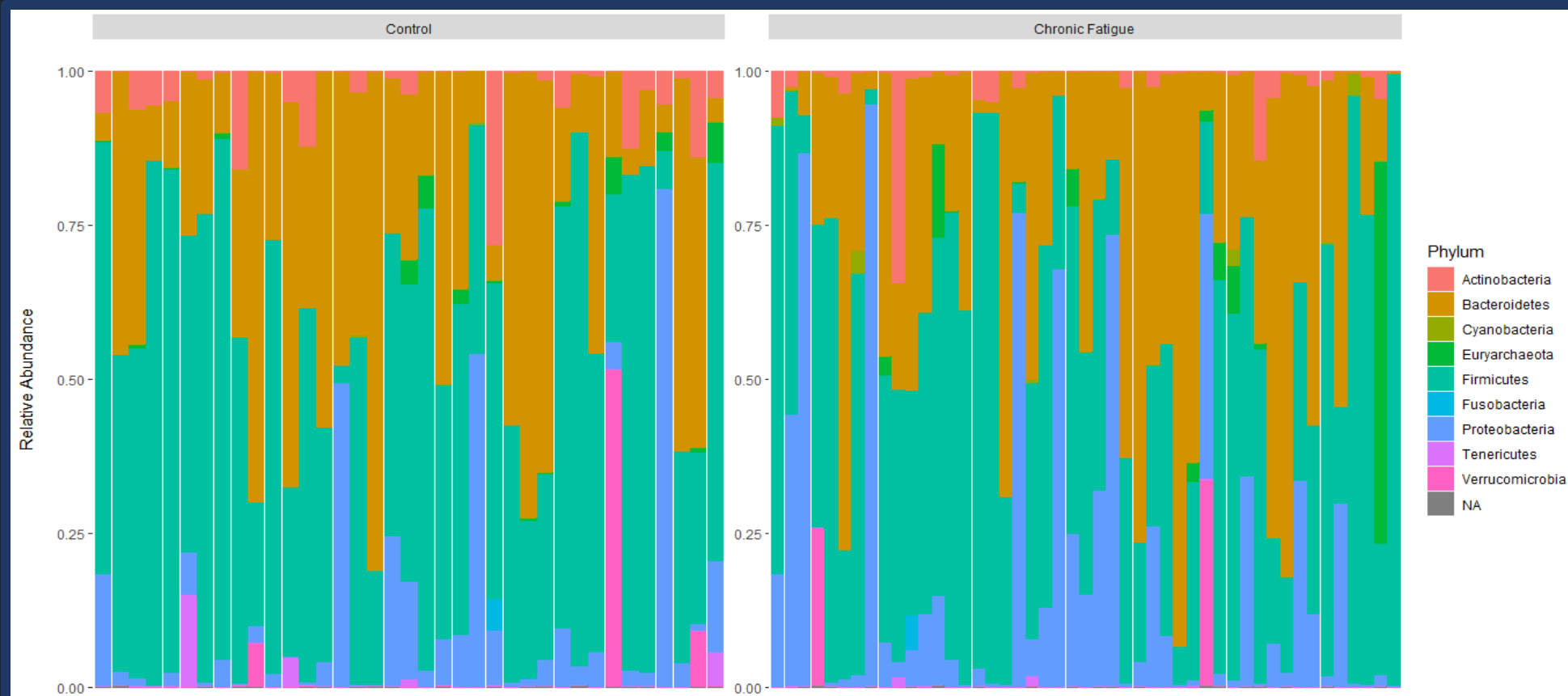
	ERR1331793	ERR1331872	ERR1331819	ERR1331794	ERR1331851
OTU1	2	581	347	916	10498
OTU2	371	46	0	233	301
OTU3	1189	81	637	199	0
OTU4	0	172	246	0	372
OTU5	308	44	143	155	221

OTU Table: [5 taxa and 5 samples]
taxa are rows

	ERR1331793	ERR1331872	ERR1331819	ERR1331794	ERR1331851
OTU1	0.0003020236	0.026008326	0.05028986	0.013891206	0.73080404
OTU2	0.0560253700	0.002059179	0.00000000	0.003533462	0.02095371
OTU3	0.1795530051	0.003625946	0.09231884	0.003017849	0.00000000
OTU4	0.0000000000	0.007699539	0.03565217	0.000000000	0.02589628
OTU5	0.0465116279	0.001969649	0.02072464	0.002350586	0.01538462

Visualizing Data (Now with Colors)

```
r$> phyloseq::plot_bar(ps_rel_abund, fill = "Phylum") +  
  geom_bar(aes(color = Phylum, fill = Phylum), stat = "identity", position = "stack") +  
  labs(x = "", y = "Relative Abundance\n") +  
  facet_wrap(~ Status, scales = "free") +  
  theme(panel.background = element_blank(),  
        axis.text.x=element_blank(),  
        axis.ticks.x=element_blank())
```



Visualizing Data (Now on the Phylum-level)

- Preparing to compare, phylum-by-phylum

```
r$> ps_phylum <- phyloseq::tax_glom(ps, "Phylum")
      phyloseq::taxa_names(ps_phylum) <- phyloseq::tax_table(ps_phylum)[, "Phylum"]
      phyloseq::otu_table(ps_phylum)[1:5, 1:5]
```

OTU Table: [5 taxa and 5 samples]

taxa are rows

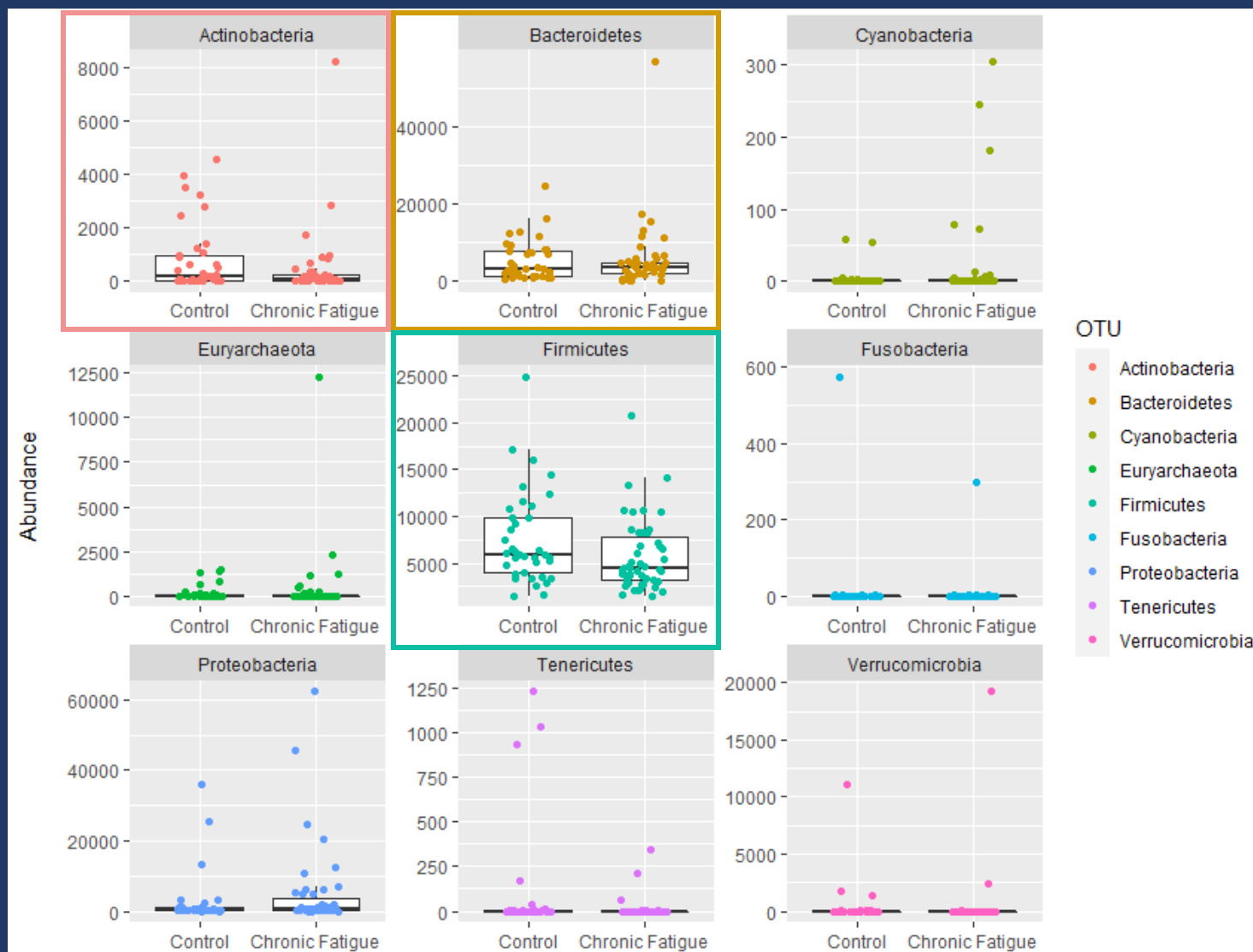
	ERR1331793	ERR1331872	ERR1331819	ERR1331794	ERR1331851
Bacteroidetes	1903	878	1837	1969	11776
Proteobacteria	119	3315	468	62358	319
Firmicutes	4319	14429	3548	1609	2207
Actinobacteria	30	976	17	0	58
Cyanobacteria	246	0	0	0	0

Visualizing Data (Phylum-to-Phylum)

```
r$> phyloseq::psmelt(ps_phylum) %>%  
  ggplot(data = ., aes(x = Status, y = Abundance)) +  
  geom_boxplot(outlier.shape = NA) +  
  geom_jitter(aes(color = OTU), height = 0, width = .2) +  
  labs(x = "", y = "Abundance\n") +  
  facet_wrap(~ OTU, scales = "free")
```



- Notice the high abundances:
 - Firmicutes
 - Bacteroidetes
 - Actinobacteria



Testing for Abundance Differences (Hypothesis Testing)

- Assuming a Dirichlet-Multinomial distribution, we essentially 't-test' between all phyla (or really any taxonomic level)

```
r$> #Subset groups
controls <- phyloseq::subset_samples(ps_phylum, Status == "Control")
cf <- phyloseq::subset_samples(ps_phylum, Status == "Chronic Fatigue")
#Output OTU tables
control_otu <- data.frame(phyloseq::otu_table(controls))
cf_otu <- data.frame(phyloseq::otu_table(cf))
#Group rare phyla
control_otu <- control_otu %>%
  t(.) %>%
  as.data.frame(.) %>%
  mutate(Other = Cyanobacteria + Euryarchaeota + Tenericutes + Verrucomicrobia + Fusobacteria) %>%
  dplyr::select(-Cyanobacteria, -Euryarchaeota, -Tenericutes, -Verrucomicrobia, -Fusobacteria)
cf_otu <- cf_otu %>%
  t(.) %>%
  as.data.frame(.) %>%
  mutate(Other = Cyanobacteria + Euryarchaeota + Tenericutes + Verrucomicrobia + Fusobacteria) %>%
  dplyr::select(-Cyanobacteria, -Euryarchaeota, -Tenericutes, -Verrucomicrobia, -Fusobacteria)
#HMP test
group_data <- list(control_otu, cf_otu)
(xdc <- HMP::Xdc.sevsample(group_data))
```

```
$`Xdc statistics`
[1] 0.2769004

$p value`
[1] 0.9980551
```

```
$`Xdc statistics`
[1] 0.2769004
```

```
$`p value`
[1] 0.9980551
```

- P-value is too high to reject the null hypothesis.

Clustering to Determine Similarity (Bray-Curtis Method)

- Bray-Curtis Dissimilarity: 0 for max similarity; 1 for no shared taxa (no similarity)

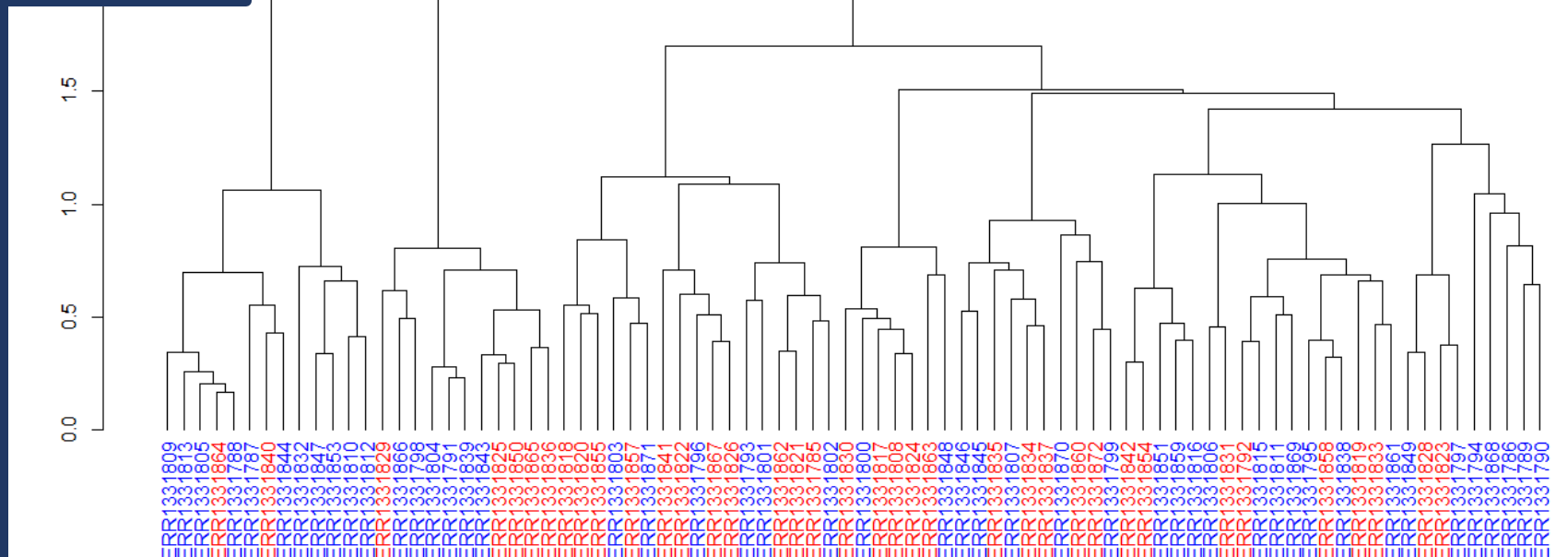
```
r$> ps_rel_otu <- data.frame(phyloseq::otu_table(ps_rel_abund))
ps_rel_otu <- t(ps_rel_otu)
bc_dist <- vegan::vegdist(ps_rel_otu, method = "bray")
as.matrix(bc_dist)[1:5, 1:5]
```

	ERR1331793	ERR1331872	ERR1331819	ERR1331794	ERR1331851
ERR1331793	0.0000000	0.8801040	0.5975550	0.9767218	0.8684629
ERR1331872	0.8801040	0.0000000	0.7590766	0.9596181	0.9206484
ERR1331819	0.5975550	0.7590766	0.0000000	0.9556656	0.7810736
ERR1331794	0.9767218	0.9596181	0.9556656	0.0000000	0.9693291
ERR1331851	0.8684629	0.9206484	0.7810736	0.9693291	0.0000000

Clustering to Determine Similarity (Visualized as a Dendrogram)

- Using the distances to make a phylo-tree

```
r$> #Save as dendrogram
ward <- as.dendrogram(hclust(bc_dist, method = "ward.D2"))
#Provide color codes
meta <- data.frame(phyloseq::sample_data(ps_rel_abund))
colorCode <- c(Control = "red", `Chronic Fatigue` = "blue")
labels_colors(ward) <- colorCode[meta$Status][order.dendrogram(ward)]
#Plot
plot(ward)
```

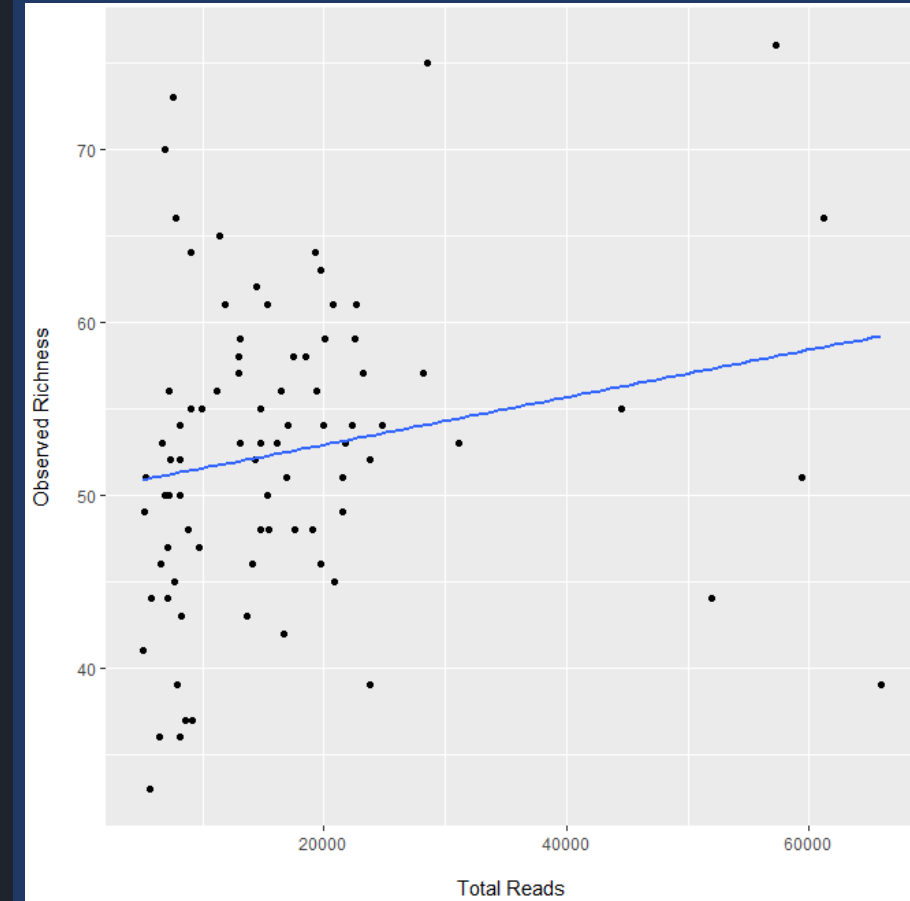


- Dissimilarity lower than 0.5
 - Most similar
- Dissimilarity greater than 0.5
 - Least similar

Alpha-diversity: Local Sample Diversity

- Testing for differences using such things as richness, Shannon diversity, and phylogenetic diversity

```
r$> #Alpha-Diversity
ggplot(data = data.frame("total_reads" = phyloseq::sample_sums(ps),
                          "observed" = phyloseq::estimate_richness(ps, measures = "Observed")[, 1]),
       aes(x = total_reads, y = observed)) +
  geom_point() +
  geom_smooth(method="lm", se = FALSE) +
  labs(x = "\nTotal Reads", y = "Observed Richness\n")
```



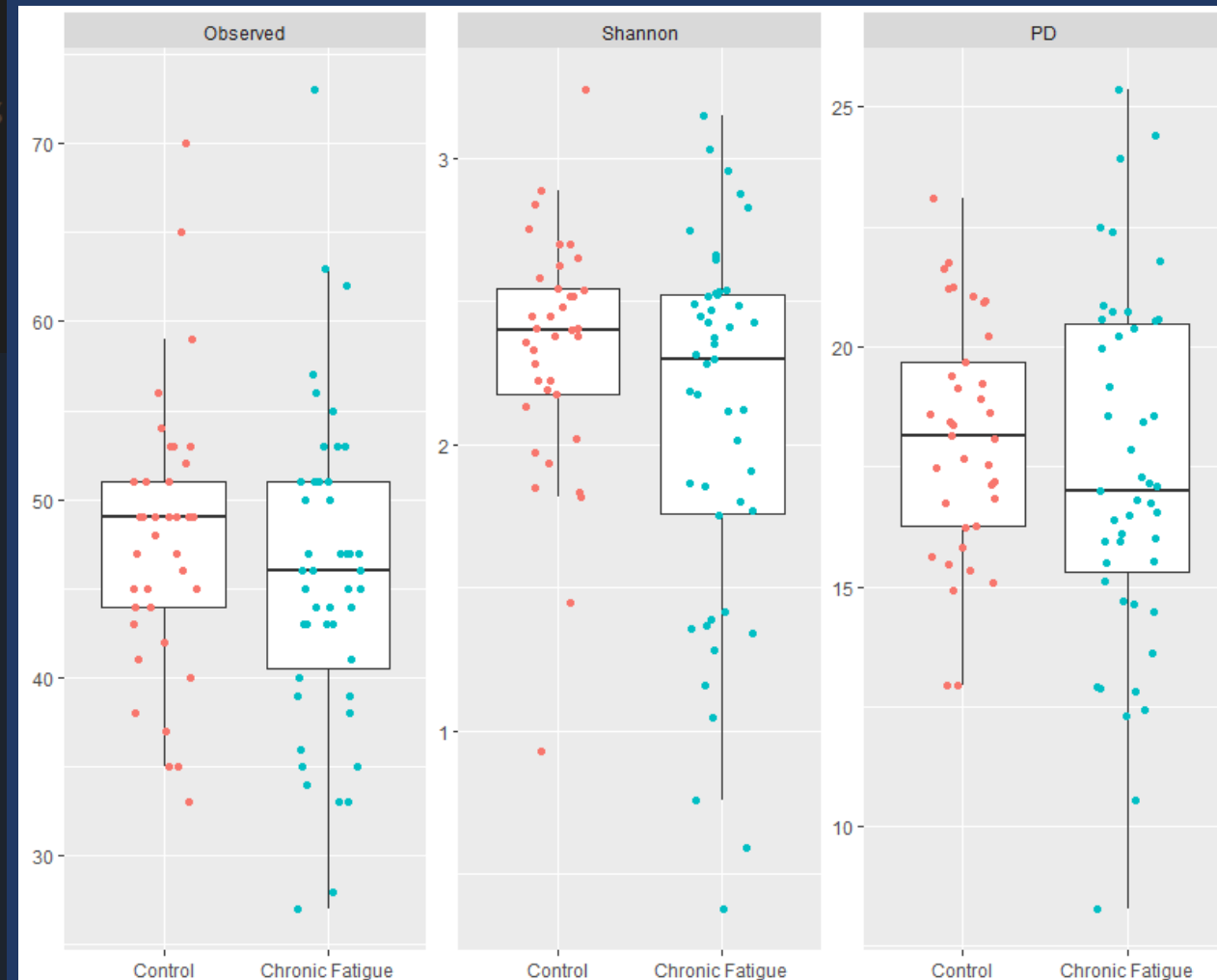
Alpha-diversity: Comparing Metrics (Plot Forme)

- Lower alpha-diversity for chronic patients

```
r$> #Plot adiv measures
adiv %>%
  gather(key = metric, value = value, c("Observed", "Shannon", "PD")) %>%
  mutate(metric = factor(metric, levels = c("Observed", "Shannon", "PD"))) %>%
  ggplot(aes(x = Status, y = value)) +
  geom_boxplot(outlier.color = NA) +
  geom_jitter(aes(color = Status), height = 0, width = .2) +
  labs(x = "", y = "") +
  facet_wrap(~ metric, scales = "free") +
  theme(legend.position="none")
```

```
r$> #Summarize
adiv %>%
  group_by(Status) %>%
  dplyr::summarise(median_observed = median(Observed),
                  median_shannon = median(Shannon),
                  median_pd = median(PD))

# A tibble: 2 x 4
  Status      median_observed median_shannon median_pd
<fct>          <dbl>          <dbl>      <dbl>
1 Control           49            2.40       18.1
2 Chronic Fatigue    46            2.30       17.0
```



Alpha-diversity: Comparing Metrics (Wilcoxon Rank Sum Tests)

```
r$> wilcox.test(Shannon ~ Status, data = adiv, conf.int = TRUE)
```

Wilcoxon rank sum exact test

data: Shannon by Status

W = 1037, p-value = 0.1329

alternative hypothesis: true location shift is not equal to 0

95 percent confidence interval:

-0.04346366 0.39218192

sample estimates:

difference in location

0.1421467

Fail to Reject

Shannon
Entropy

```
r$> wilcox.test(PD ~ Status, data = adiv, conf.int = TRUE)
```

Wilcoxon rank sum exact test

data: PD by Status

W = 998, p-value = 0.2503

alternative hypothesis: true location shift is not equal to 0

95 percent confidence interval:

-0.7340479 2.1737379

sample estimates:

difference in location

0.7063854

Fail to Reject

PD

```
r$> wilcox.test(Observed ~ Status, data = adiv, exact = FALSE, conf.int = TRUE)
```

Wilcoxon rank sum test with continuity correction

data: Observed by Status

W = 1007.5, p-value = 0.2146

alternative hypothesis: true location shift is not equal to 0

95 percent confidence interval:

-1.000059 5.000002

sample estimates:

difference in location

2.000087

Fail to Reject

Observed

Richness using the *breakaway* Package

- Form breakaway estimates:

```
r$> ba_adiv <- breakaway::breakaway(ps)
      ba_adiv[1]
      #Plot estimates
$ERR1331793
Estimate of richness from method breakaway:
Estimate is 53
with standard error 0.6
Confidence interval: (53, 55)
Cutoff: 10
```

- Summary of Estimates:

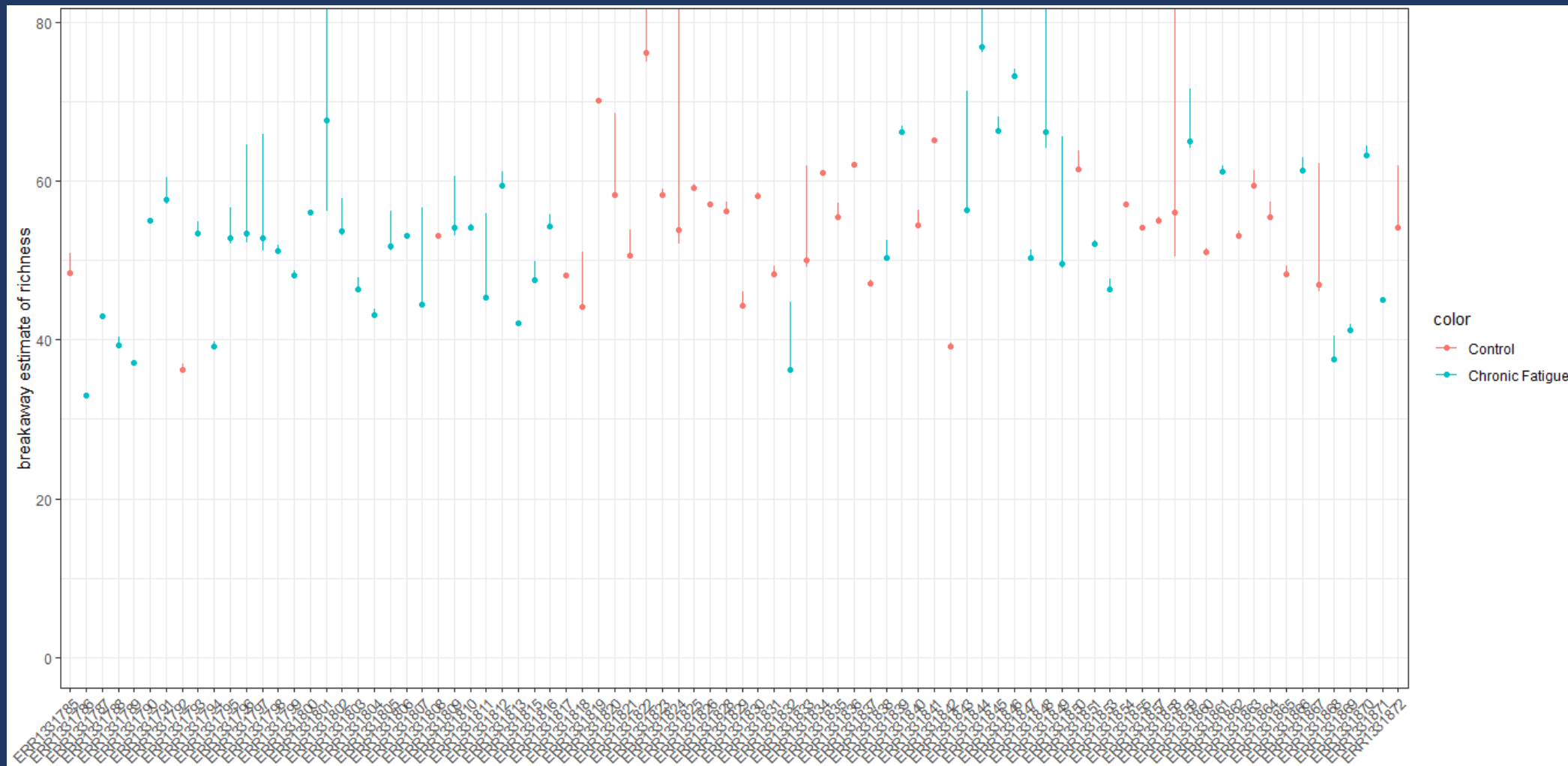
```
r$> summary(ba_adiv) %>%
  add_column("SampleNames" = ps %>% otu_table %>% sample_names)
#Test for group difference
bt <- breakaway::betta(summary(ba_adiv)$estimate,
  summary(ba_adiv)$error,
  make_design_matrix(ps, "Status"))

bt$table
# A tibble: 84 x 8
  estimate error lower upper sample_names name      model      SampleNames
  <dbl> <dbl> <dbl> <dbl> <chr>      <chr>      <chr>      <chr>
1 53.3 0.602 53.1 54.8 ERR1331793 breakaway Poisson ERR1331793
2 54.1 3.10 54.0 61.9 ERR1331872 breakaway Negative Binomial ERR1331872
3 70.1 0.296 70.0 70.4 ERR1331819 breakaway Kemp ERR1331819
4 39.1 0.381 39.0 39.7 ERR1331794 breakaway Poisson ERR1331794
5 52.1 0.326 52.0 52.5 ERR1331851 breakaway Kemp ERR1331851
6 61.1 0.280 61.0 61.4 ERR1331834 breakaway Kemp ERR1331834
7 54.1 0.346 54.0 54.6 ERR1331810 breakaway Kemp ERR1331810
8 48.1 0.243 48.0 48.3 ERR1331817 breakaway Kemp ERR1331817
9 56.0 5.17 50.5 127. ERR1331858 breakaway Kemp ERR1331858
10 50.0 2.14 49.1 61.9 ERR1331833 breakaway Kemp ERR1331833
# ... with 74 more rows

Estimates Standard Errors p-values
(Intercept) 54.088528 0.9817616 0.000
predictorsChronic Fatigue -2.186958 1.3085094 0.095
```

Richness using the *breakaway* Package (Plots)

```
r$> #Plot estimates  
plot(ba_adiv, ps, color = "Status")
```



Beta-diversity: Regional to Local Diversity Ratios

- Create the CLRs (centered log-ratios)

```
(ps_clr <- microbiome::transform(ps, "clr"))
```

- Untransformed values (counts):
- Transformed values (log dominance):

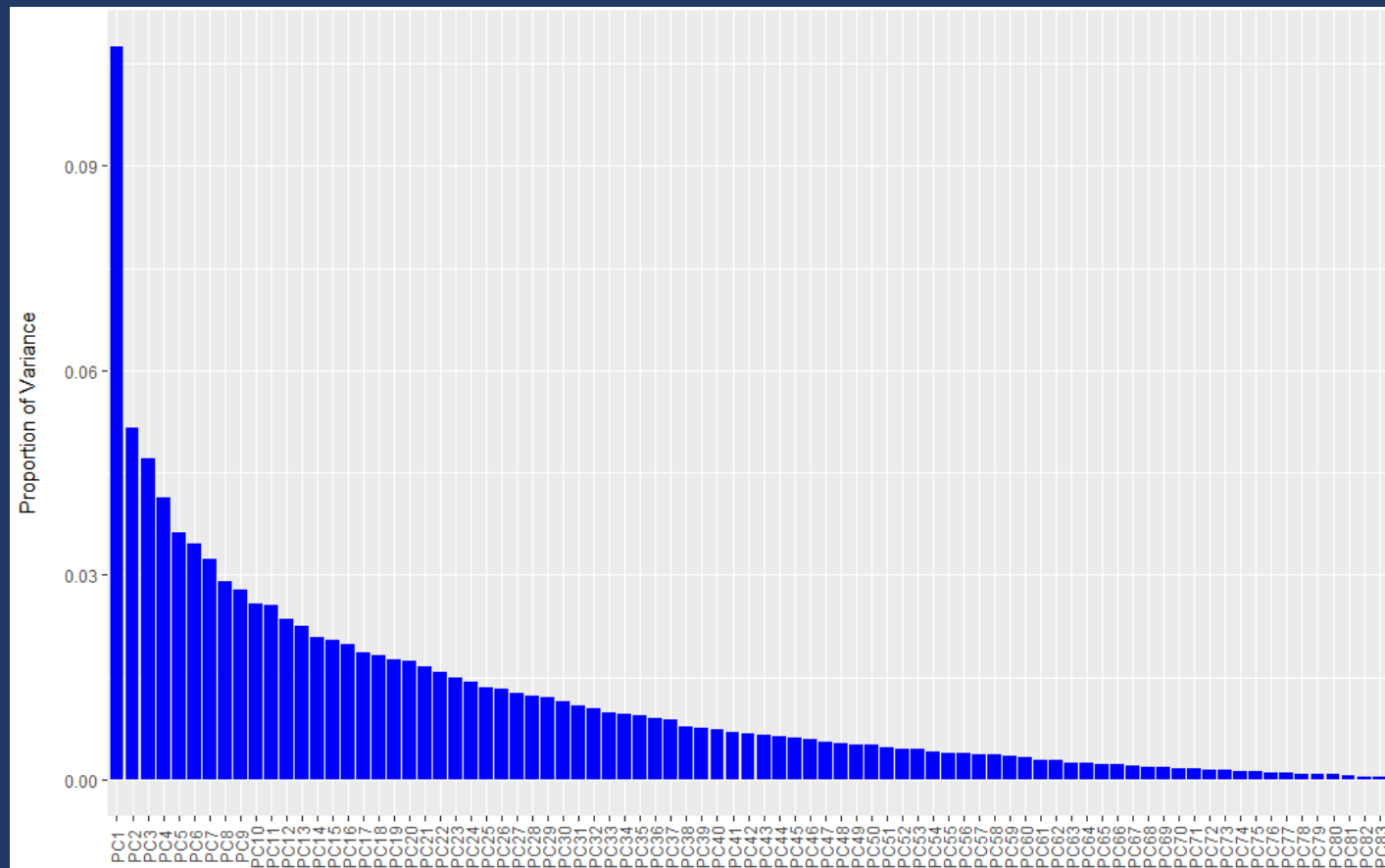
```
r$> phyloseq::otu_table(ps)[1:5, 1:5]
OTU Table:           [5 taxa and 5 samples]
                    taxa are rows
      ERR1331793 ERR1331872 ERR1331819 ERR1331794 ERR1331851
OTU1           2         581         347         916        10498
OTU2          371          46           0         233         301
OTU3         1189          81         637         199           0
OTU4           0         172         246           0         372
OTU5          308          44         143         155         221
```

```
r$> phyloseq::otu_table(ps_clr)[1:5, 1:5]
OTU Table:           [5 taxa and 5 samples]
                    taxa are rows
      ERR1331793 ERR1331872 ERR1331819 ERR1331794 ERR1331851
OTU1    1.289544    5.812706    5.615063    6.230204    9.467837
OTU2    6.485240    3.280355   -3.079591    4.863001    5.916398
OTU3    7.649802    3.844401    6.222432    4.705673   -1.903003
OTU4   -2.317399    4.596219    5.271139   -1.178342    6.128105
OTU5    6.299168    3.236089    4.728822    4.456584    5.607596
```

Beta-diversity: Regional to Local Diversity Ratios (continued)

- Apply PCA (Principal Component Analysis)

```
r$> #PCA via phyloseq
ord_clr <- phyloseq::ordinate(ps_clr, "RDA")
#Plot scree plot
phyloseq::plot_scee(ord_clr) +
  geom_bar(stat="identity", fill = "blue") +
  labs(x = "\nAxis", y = "Proportion of Variance\n")
```



Beta-diversity: Eigenvalues and Principal Components

- Eigenvalues

```
r$> head(ord_clr$CA$eig)
      PC1      PC2      PC3      PC4      PC5      PC6
75.69204 36.27003 33.16649 29.08833 25.52986 24.32215
```

- Proportion of Variance explained by Principal Component

```
r$> sapply(ord_clr$CA$eig[1:5], function(x) x / sum(ord_clr$CA$eig))
      PC1      PC2      PC3      PC4      PC5
0.10744095 0.05148344 0.04707812 0.04128939 0.03623832
```

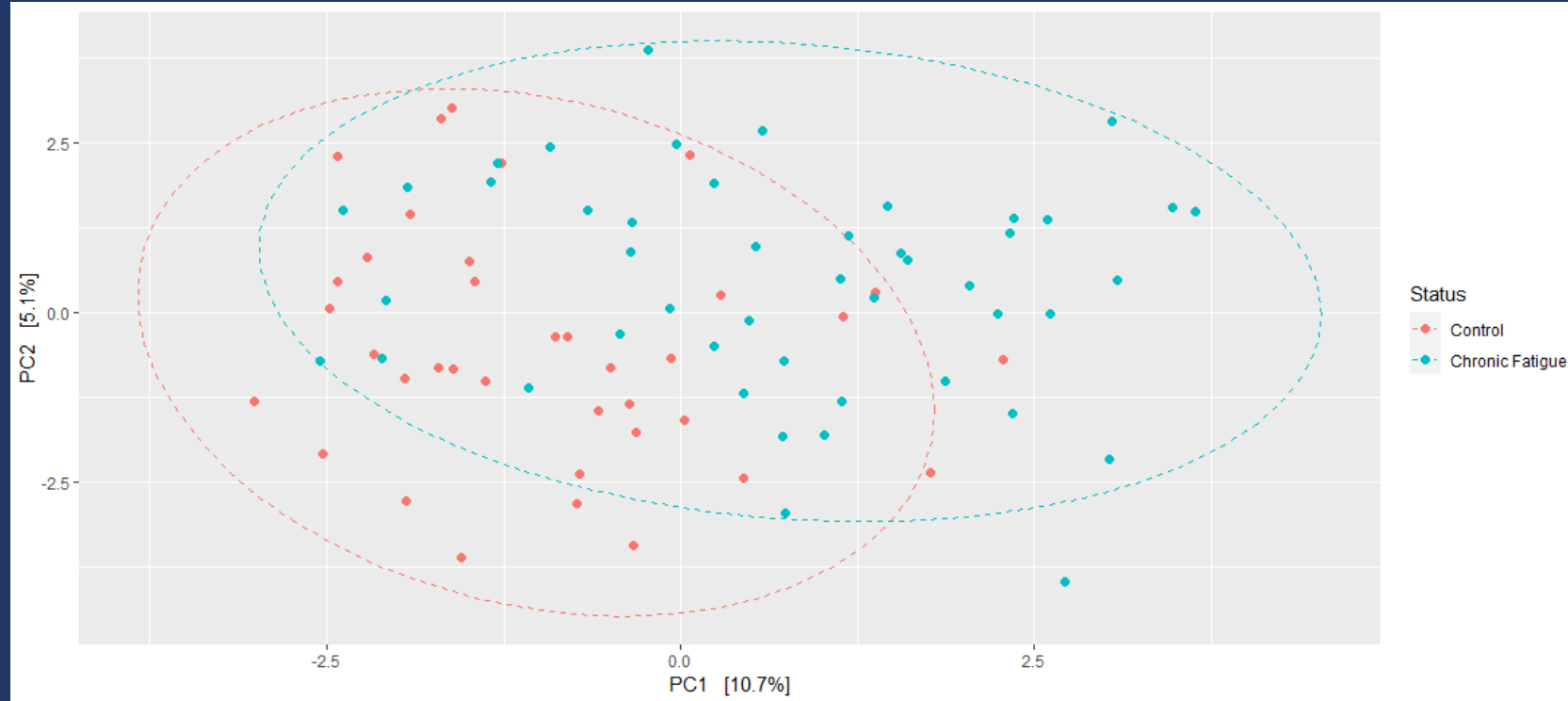


Beta-diversity: The First Two Principal Components

```
r$> #Scale axes and plot ordination
clr1 <- ord_clr$CA$eig[1] / sum(ord_clr$CA$eig)
clr2 <- ord_clr$CA$eig[2] / sum(ord_clr$CA$eig)
phyloseq::plot_ordination(ps, ord_clr, type="samples", color="Status") +
  geom_point(size = 2) +
  coord_fixed(clr2 / clr1) +
  stat_ellipse(aes(group = Status), linetype = 2)
```



- Notice the slight separation between the Control and the Fatigued



Beta-diversity: PERMANOVA and *adonis*

```
r$> #Generate distance matrix
clr_dist_matrix <- phyloseq::distance(ps_clr, method = "euclidean")
#ADONIS test
vegan::adonis(clr_dist_matrix ~ phyloseq::sample_data(ps_clr)$Status)
```

Call:
vegan::adonis(formula = clr_dist_matrix ~ phyloseq::sample_data(ps_clr)\$Status)

Permutation: free
Number of permutations: 999

Terms added sequentially (first to last)

	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)	
phyloseq::sample_data(ps_clr)\$Status	1	2240	2240.17	3.2666	0.03831	0.001	***
Residuals	82	56233	685.77		0.96169		
Total	83	58473			1.00000		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Beta-diversity: PERMANOVA and *adonis* (Plot and Dispersion Test)

- Dispersion Test

```
r$> #Dispersion test and plot
dispr <- vegan::betadisper(clr_dist_matrix, phyloseq::sample_data(ps_clr)$Status)
dispr

Homogeneity of multivariate dispersions

Call: vegan::betadisper(d = clr_dist_matrix, group = phyloseq::sample_data(ps_clr)$Status)

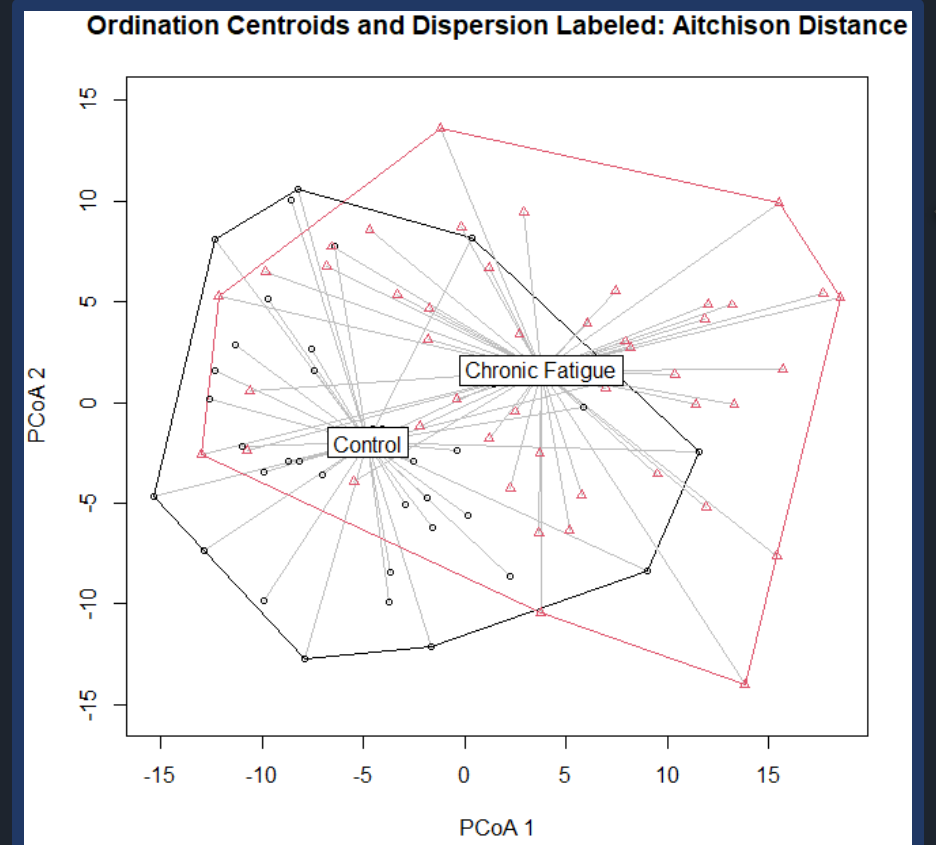
No. of Positive Eigenvalues: 83
No. of Negative Eigenvalues: 0

Average distance to median:
      Control Chronic Fatigue
      25.1      26.2

Eigenvalues for PCoA axes:
(Showing 8 of 83 eigenvalues)
PCoA1 PCoA2 PCoA3 PCoA4 PCoA5 PCoA6 PCoA7 PCoA8
 6282 3010 2753 2414 2119 2019 1895 1693
```

- PCoA Plotted

```
r$> plot(dispr, main = "Ordination Centroids and Dispersion Labeled: Aitchison Distance", sub = "")
```



Beta-diversity: PERMANOVA and *adonis* (Box and Permutation)

- PCoA Plot as Box

```
r$> boxplot(dispr, main = "", xlab = "")
```

- Permutation Test

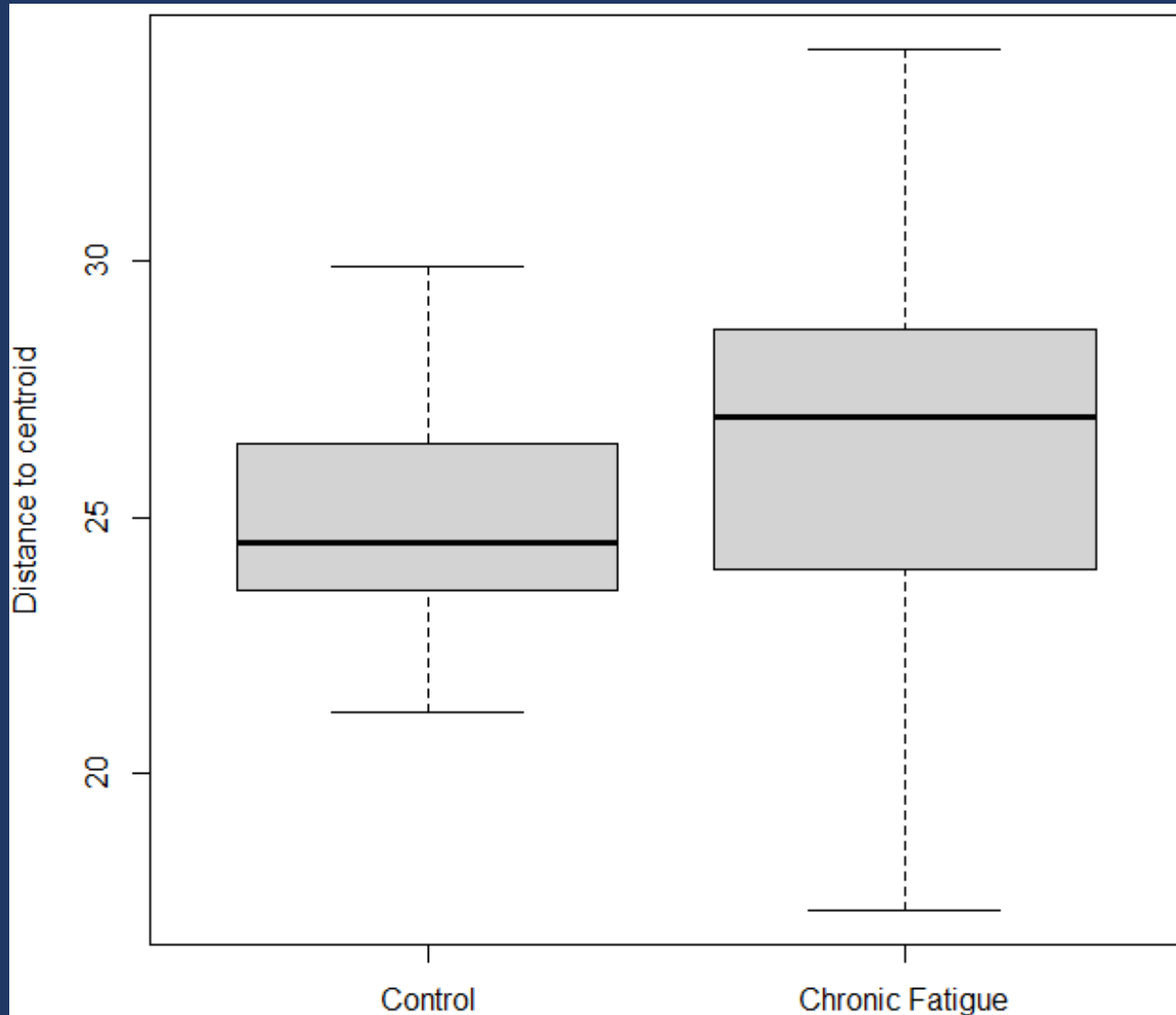
```
r$> permutest(dispr)
```

Permutation test for homogeneity of multivariate dispersions
Permutation: free
Number of permutations: 999

Response: Distances

	Df	Sum Sq	Mean Sq	F	N.Perm	Pr(>F)
Groups	1	24.95	24.9463	3.0491	999	0.077 .
Residuals	82	670.89	8.1816			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

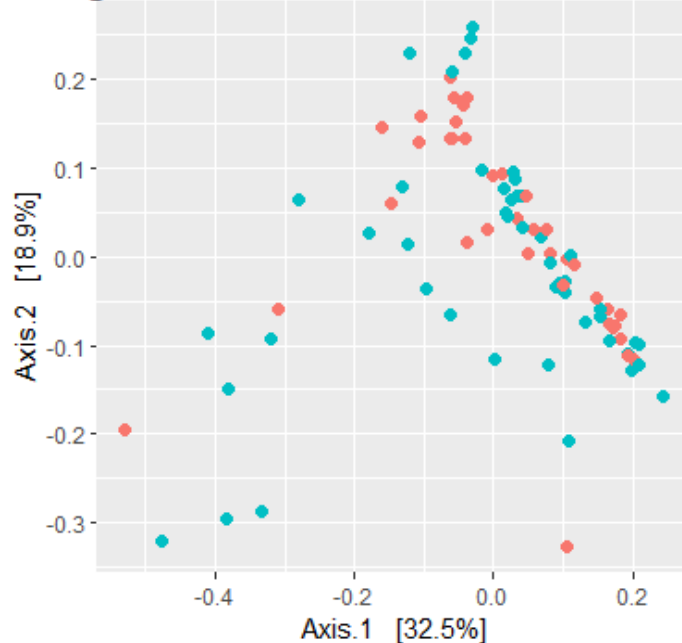


Principal Coordinate Analysis and the UniFrac Distance

- Using PCoA to computer similarity as the UniFrac distance

```
r$> #Generate distances
ord_unifrac <- ordinate(ps_rare, method = "PCoA", distance = "wunifrac")
ord_unifrac_un <- ordinate(ps_rare, method = "PCoA", distance = "unifrac")
#Plot ordinations
a <- plot_ordination(ps_rare, ord_unifrac, color = "Status") + geom_point(size = 2)
cowplot::plot_grid(a, b, nrow = 1, ncol = 2, scale = .9, labels = c("Weighted", "Unweighted"))
```

Weighted



Unweighted



Differential Abundance Testing

- Identifying taxa that respond the greatest to changes in some condition
 - Statistically difficult due to sampling issues
- Two Main Approaches:
 - Non-parametric Wilcoxon rank-sum test
 - Modified Wilcoxon test for NGS data
 - NGS (Next Generation Sequencing)

```
r$> ps_wilcox <- data.frame(t(data.frame(phyloseq::otu_table(ps_clr))))
ps_wilcox$Status <- phyloseq::sample_data(ps_clr)$Status

r$> wilcox_model <- function(df){
  wilcox.test(abund ~ Status, data = df)
}
wilcox_pval <- function(df){
  wilcox.test(abund ~ Status, data = df)$p.value
}

r$> wilcox_results <- ps_wilcox %>%
  gather(key = OTU, value = abund, -Status) %>%
  group_by(OTU) %>%
  nest() %>%
  mutate(wilcox_test = map(data, wilcox_model),
         p_value = map(data, wilcox_pval))
```

Differential Abundance Testing (Unmodified Wilcoxon)

- Test results including:

- Abundance fields
- Status fields
- P-Values

```
> head(wilcox_results)
# A tibble: 6 x 2
# Groups:   OTU [6]
  OTU      p_value
  <chr>    <dbl>
1 OTU1  0.00607
2 OTU2  0.0686
3 OTU3  0.830
4 OTU4  0.0130
5 OTU5  0.419
6 OTU6  0.258
```

```
> #Show results
> head(wilcox_results)
# A tibble: 6 x 4
# Groups:   OTU [6]
  OTU      data      wilcox_test p_value
  <chr> <list>      <list>      <list>
1 OTU1 <tibble[,2] [84 x 2]> <htest>    <dbl [1]>
2 OTU2 <tibble[,2] [84 x 2]> <htest>    <dbl [1]>
3 OTU3 <tibble[,2] [84 x 2]> <htest>    <dbl [1]>
4 OTU4 <tibble[,2] [84 x 2]> <htest>    <dbl [1]>
5 OTU5 <tibble[,2] [84 x 2]> <htest>    <dbl [1]>
6 OTU6 <tibble[,2] [84 x 2]> <htest>    <dbl [1]>
> head(wilcox_results$data[[1]])
# A tibble: 6 x 2
```

Status	abund
<fct>	<dbl>
1 Chronic Fatigue	1.29
2 Control	5.81
3 Control	5.62
4 Chronic Fatigue	6.23
5 Chronic Fatigue	9.47
6 Control	7.43

```
> wilcox_results$wilcox_test[[1]]
```

wilcoxon rank sum exact test

data: abund by Status

w = 1172, p-value = 0.006066

alternative hypothesis: true location shift is not equal to 0

```
> wilcox_results$p_value[[1]]
```

```
[1] 0.006066387
```

Differential Abundance Testing (Modified Wilcoxon)

- Unpack all test values and make some more transforms
- Modified test shows many Clostridiales organisms as being 'differentially abundant'
 - In the 'Order' column of the output

```
tutorial_3.R
304 dplyr::select(otu, p_value) %>%
305   unnest()
306
307
308 head(wilcox_results)
309
310
311 #adding taxonomic labels
312 taxa_info <- data.frame(tax_table(ps_clr))
313 taxa_info <- taxa_info %>% rownames_to_column(var = "otu")
314 #computing FDR corrected p-values
315 wilcox_results <- wilcox_results %>%
316   full_join(taxa_info) %>%
317   arrange(p_value) %>%
318   mutate(BH_FDR = p.adjust(p_value, "BH")) %>%
319   filter(BH_FDR < 0.05) %>%
320   dplyr::select(otu, p_value, BH_FDR, everything())
321
322
323 #printing results
324 print.data.frame(wilcox_results)
325
326
327 #run ALDEx2 (runs for a little bit)
328 aldex2_da <- ALDEx2::aldex(data.frame(phyloseq::otu_table(ps)), phyloseq::sample_data(ps)$status, test="t", effect = "t")
329
330
331
```

Joining, by = "otu"

```
> #printing results
> print.data.frame(wilcox_results)
```

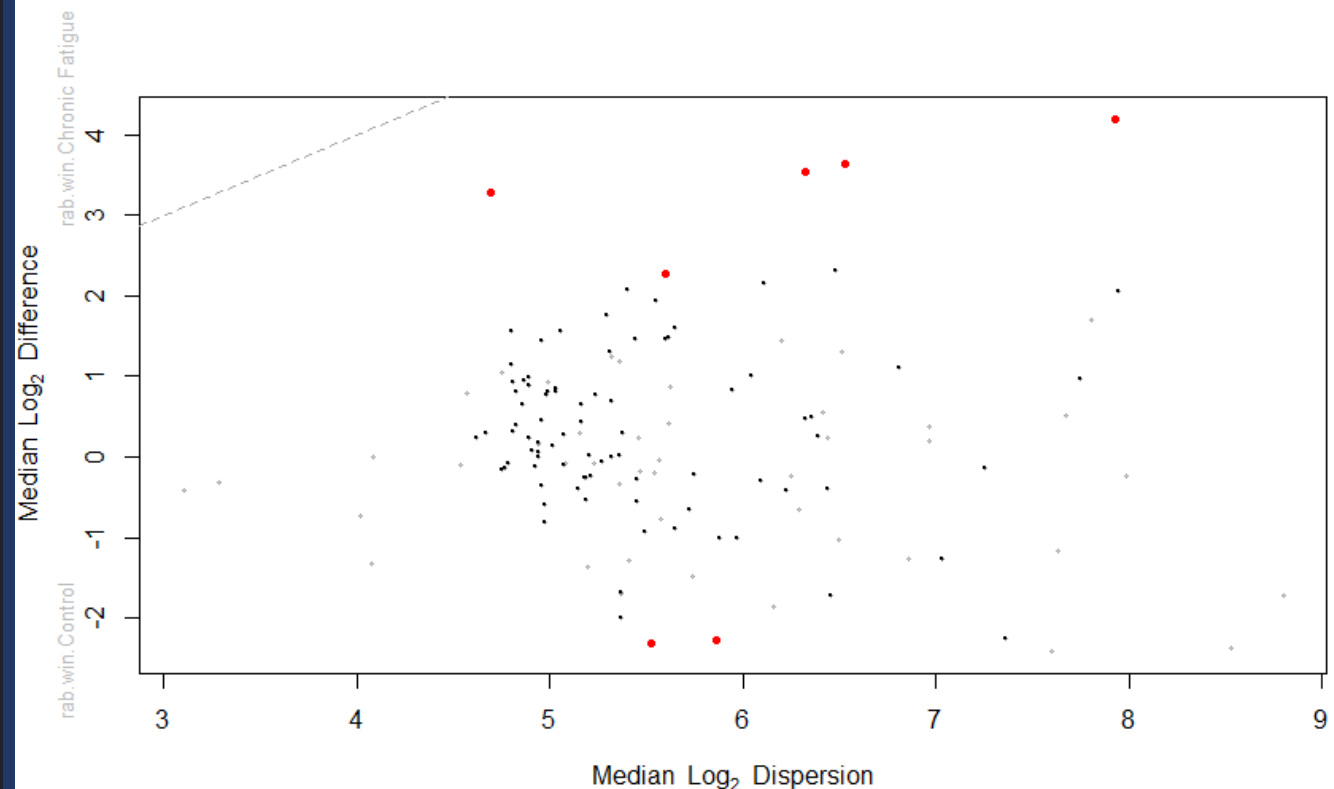
	otu	p_value	BH_FDR	Kingdom	Phylum	Class	order	family
1	OTU48	1.893126e-05	1.893126e-05	Bacteria	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae
2	OTU38	4.168412e-05	4.168412e-05	Bacteria	Firmicutes	Clostridia	Clostridiales	Ruminococcaceae
3	OTU44	2.750125e-04	2.750125e-04	Bacteria	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae
4	OTU61	1.217944e-03	1.217944e-03	Bacteria	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae
5	OTU104	1.390580e-03	1.390580e-03	Bacteria	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae
6	OTU115	1.804359e-03	1.804359e-03	Bacteria	Firmicutes	Clostridia	Clostridiales	[Mogibacteriaceae]
7	OTU83	2.050647e-03	2.050647e-03	Bacteria	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae
8	OTU8	2.719699e-03	2.719699e-03	Bacteria	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae
9	OTU123	2.719699e-03	2.719699e-03	Bacteria	Firmicutes	Erysipelotrichi	Erysipelotrichales	Erysipelotrichaceae
10	OTU117	3.801488e-03	3.801488e-03	Bacteria	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae
11	OTU1	6.066387e-03	6.066387e-03	Bacteria	Bacteroidetes	Bacteroidia	Bacteroidales	Porphyromonadaceae
12	OTU133	6.420958e-03	6.420958e-03	Bacteria	Actinobacteria	Coriobacteria	Coriobacteriales	Coriobacteriaceae
13	OTU26	8.720398e-03	8.720398e-03	Bacteria	Actinobacteria	Actinobacteria	Bifidobacteriales	Bifidobacteriaceae
14	OTU116	9.462190e-03	9.462190e-03	Bacteria	Firmicutes	Clostridia	Clostridiales	Ruminococcaceae
15	OTU43	9.721528e-03	9.721528e-03	Bacteria	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae
16	OTU51	9.721528e-03	9.721528e-03	Bacteria	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae
17	OTU4	1.301459e-02	1.301459e-02	Bacteria	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae
18	OTU21	1.519295e-02	1.519295e-02	Bacteria	Firmicutes	Clostridia	Clostridiales	Ruminococcaceae
19	OTU42	1.639651e-02	1.639651e-02	Bacteria	Firmicutes	Erysipelotrichi	Erysipelotrichales	Erysipelotrichaceae
20	OTU39	1.953122e-02	1.953122e-02	Bacteria	Cyanobacteria	Chloroplast	Streptophyta	<NA>
21	OTU17	2.051686e-02	2.051686e-02	Bacteria	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae
22	OTU113	2.102556e-02	2.102556e-02	Bacteria	Firmicutes	Clostridia	Clostridiales	Subbacteriaceae
23	OTU12	2.261734e-02	2.261734e-02	Bacteria	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae

Differential Abundance Testing (ALDEx2)

- ALDEx2 (ANOVA-like differential expression) does quite a few things:

- Generate 128 Monte-Carlo posterior probabilities for all 138 taxa
- Transform via centered log-ratio
- Unmodified Wilcoxon for each taxa per probability
- Determine effect size
- Determine average p-value per taxa
- Determine expected p-values (all instances)
- Avoid false positives with BH-FDR (Benjamin-Hochberg False Discovery Rate)

```
r$> aldex2_da <- ALDEx2::aldex(data.frame(phyloseq::otu_table(ps)), phyloseq::sample_data(ps)$Status, test="t", effect = TRUE, denom="iqlr")  
#Plot effect sizes  
ALDEx2::aldex.plot(aldex2_da, type="MW", test="wilcox", called.cex = 1, cutoff = 0.05)
```



Differential Abundance Testing (ALDEx2)

- Which is most abundant according to this test?
 - The winner is: Clostridiales again!

```
r$> #Clean up presentation
sig_aldex2 <- aldex2_da %>%
  rownames_to_column(var = "OTU") %>%
  filter(wi.eBH < 0.05) %>%
  arrange(effect, wi.eBH) %>%
  dplyr::select(OTU, diff.btw, diff.win, effect, wi.ep, wi.eBH)
sig_aldex2 <- left_join(sig_aldex2, taxa_info)
```

sig_aldex2

Joining, by = "OTU"

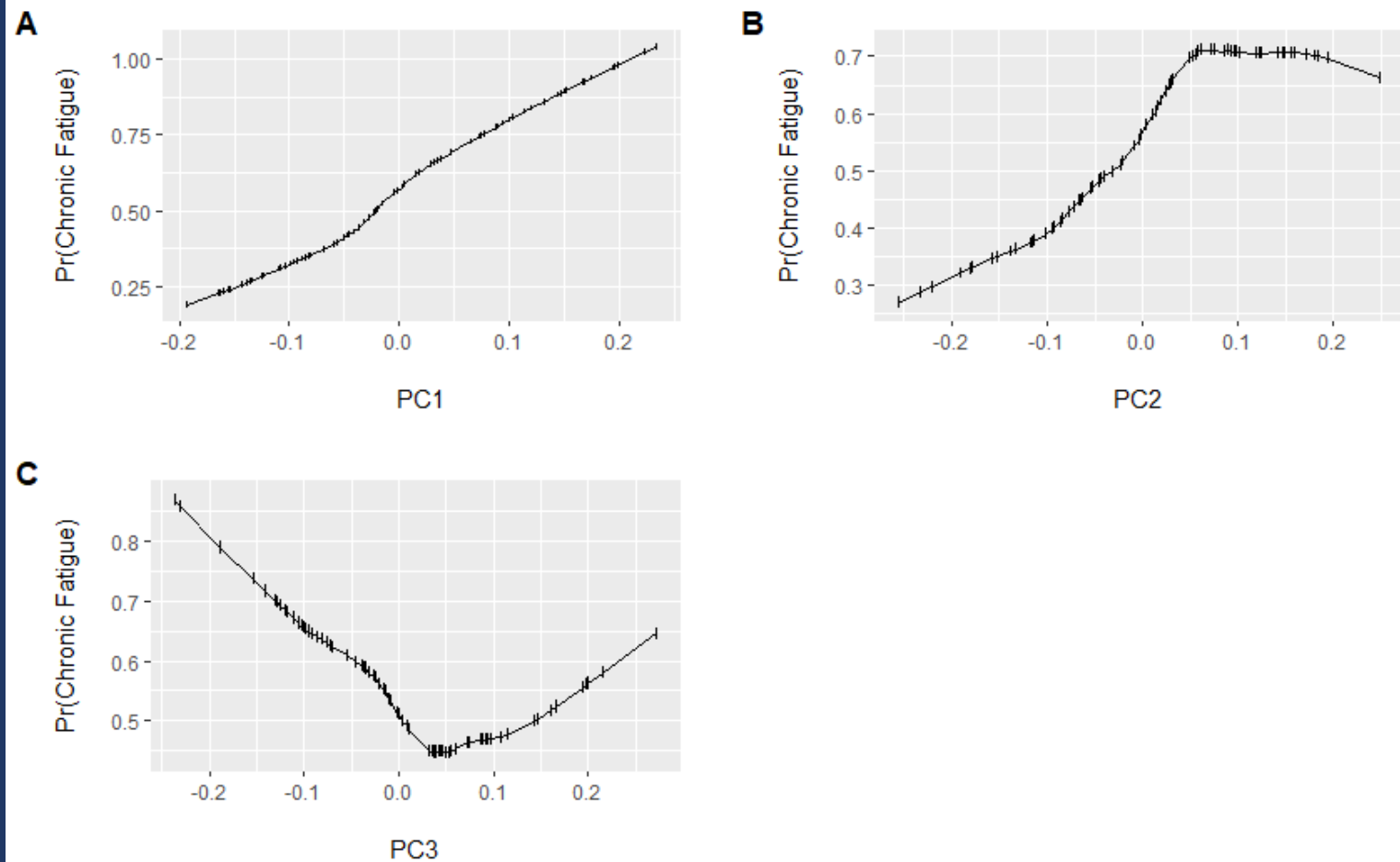
	OTU	diff.btw	diff.win	effect	wi.ep	wi.eBH	Kingdom	Phylum	Class	Order	Family	Genus	Species
1	OTU8	-2.307035	5.522587	-0.3839105	0.0015096450	0.039837292	Bacteria	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae	[Ruminococcus]	<NA>
2	OTU48	3.639216	6.528139	0.5283635	0.0025403967	0.042007768	Bacteria	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae	Coproccoccus	<NA>
3	OTU44	3.541267	6.324375	0.5296702	0.0013725146	0.031600961	Bacteria	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae	[Ruminococcus]	<NA>
4	OTU38	3.277257	4.696329	0.6206553	0.0000348666	0.004124241	Bacteria	Firmicutes	Clostridia	Clostridiales	Ruminococcaceae	Oscillospira	<NA>

Prediction

- Plotting the Principal Component with the outcome of Chronic Fatigue verse control
- From the graph there is potential for non-linear association
- Better to fit a non-linear on a linear than vice versa as you will have much less penalties
- This specifically can be modeled with restricted cubic splines

```
> head(c1r_pcs)
```

	pc1	pc2	pc3	Status	Status_num
ERR1331793	0.02850343	-0.07709724	0.0938970408	Chronic Fatigue	1
ERR1331872	-0.08156129	0.14193568	0.1155088427	Control	0
ERR1331819	-0.19356039	-0.08436341	-0.1048722096	Control	0
ERR1331794	-0.04193714	0.09705602	0.0110912849	Chronic Fatigue	1
ERR1331851	0.09994410	0.05534786	-0.0005008101	Chronic Fatigue	1
ERR1331834	-0.15577774	0.02921040	-0.0204667015	Control	0



Prediction (Restricted Cubic Splines)

- Fit Restricted Cubic Splines to the model then find the optimum value for the penalty
- Can also penalty to differ for simple and complex if we want to allow complexity but down weight the impact

```
> #Fit full model with splines (3 knots each)
> m1 <- rms::lrm(Status_num ~ rcs(pc1, 3) + rcs(pc2, 3) + rcs(pc3, 3), data = clr_pcs, x = TRUE, y = TRUE)
> #Grid search for penalties
> pentrace(m1, list(simple = c(0, 1, 2), nonlinear = c(0, 100, 200)))
```

Best penalty:

simple	nonlinear	df
1	200	2.783027

simple	nonlinear	df	aic	bic	aic.c
0	0	6.000000	23.10845	8.523552	22.01754
0	100	3.049209	28.21043	20.798359	27.90157
1	100	2.810152	28.38363	21.552668	28.11659
2	100	2.641219	28.11811	21.697792	27.87875
0	200	3.024831	28.24577	20.892958	27.94131
1	200	2.783027	28.42060	21.655570	28.15810
2	200	2.611196	28.15166	21.804324	27.91706

```
> pen_m1 <- update(m1, penalty = list(simple = 1, nonlinear = 200))
> pen_m1
Logistic Regression Model

rms::lrm(formula = Status_num ~ rcs(pc1, 3) + rcs(pc2, 3) + rcs(pc3,
3), data = clr_pcs, x = TRUE, y = TRUE, penalty = list(simple = 1,
nonlinear = 200))
```

Penalty factors

	simple	nonlinear	interaction	nonlinear.interaction
	1	200	200	200

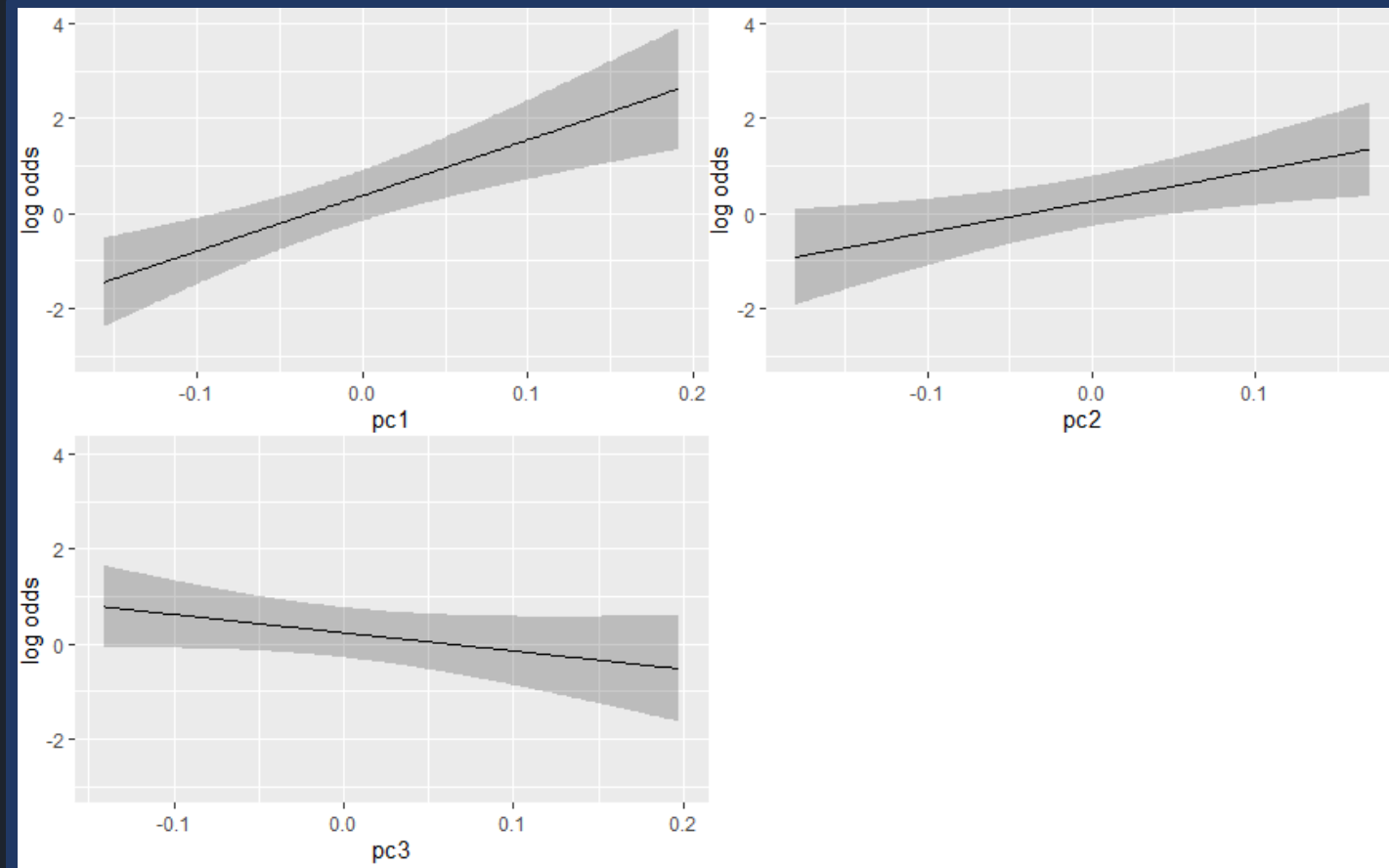
	Model Likelihood	Discrimination	Rank Discrim.
	Ratio Test	Indexes	Indexes
obs	LR chi2 33.99	R2 0.421	C 0.848
0	d.f. 2.783	g 1.759	Dxy 0.695
1	Pr(> chi2)<0.0001	gr 5.807	gamma 0.695
max	Penalty 2.34	gp 0.322	tau-a 0.347
deriv		Brier 0.159	

	Coef	S.E.	wald z	Pr(> z)	Penalty scale
Intercept	0.3458	0.2852	1.21	0.2254	0.0000
pc1	11.6489	2.8714	4.06	<0.0001	0.1098
pc1'	0.1202	0.9287	0.13	0.8970	1.0715
pc2	6.4946	2.5132	2.58	0.0098	0.1098
pc2'	-0.0015	0.7643	0.00	0.9984	1.2987
pc3	-3.8538	2.5659	-1.50	0.1331	0.1098
pc3'	0.0259	1.0080	0.03	0.9795	0.9856

Prediction (Restricted Cubic Splines Plot)

- Plot of the penalised log odds
- It can be seen that the conditional associations are quite linear
- The optimal penalties were 1 for the simple and 200 for the non-linear terms
- Effective degrees of freedom shrunk to 2.78

```
r$> #Plot log odds  
ggplot(Predict(pen_m1))
```



Prediction (Bootstrap Resampling)

- Bootstrap resampling is done to find an out-of-sample estimate of model performance

```
> #Obtain optimism corrected estimates
> (val <- rms::validate(pen_m1))
```

	index.orig	training	test	optimism	index.corrected	n
Dxy	0.6952	0.7248	0.6817	0.0431	0.6521	40
R2	0.4206	0.4536	0.4295	0.0240	0.3965	40
Intercept	0.0000	0.0000	-0.0250	0.0250	-0.0250	40
Slope	1.0000	1.0000	1.0265	-0.0265	1.0265	40
E _{max}	0.0000	0.0000	0.0100	0.0100	0.0100	40
D	0.3927	0.4045	0.3749	0.0297	0.3630	40
U	-0.0238	-0.0238	-0.0073	-0.0165	-0.0073	40
Q	0.4165	0.4283	0.3822	0.0461	0.3704	40
B	0.1589	0.1481	0.1647	-0.0166	0.1755	40
g	1.7591	1.9083	1.9158	-0.0075	1.7666	40
gp	0.3218	0.3287	0.3363	-0.0076	0.3293	40

```
> #Compute corrected c-statistic
> (c_opt_corr <- 0.5 * (val[1, 5] + 1))
[1] 0.8260443
```

More on describing this in greater detail: <https://thestatsgeek.com/2014/10/04/adjusting-for-optimismoverfitting-in-measures-of-predictive-ability-using-bootstrapping/>

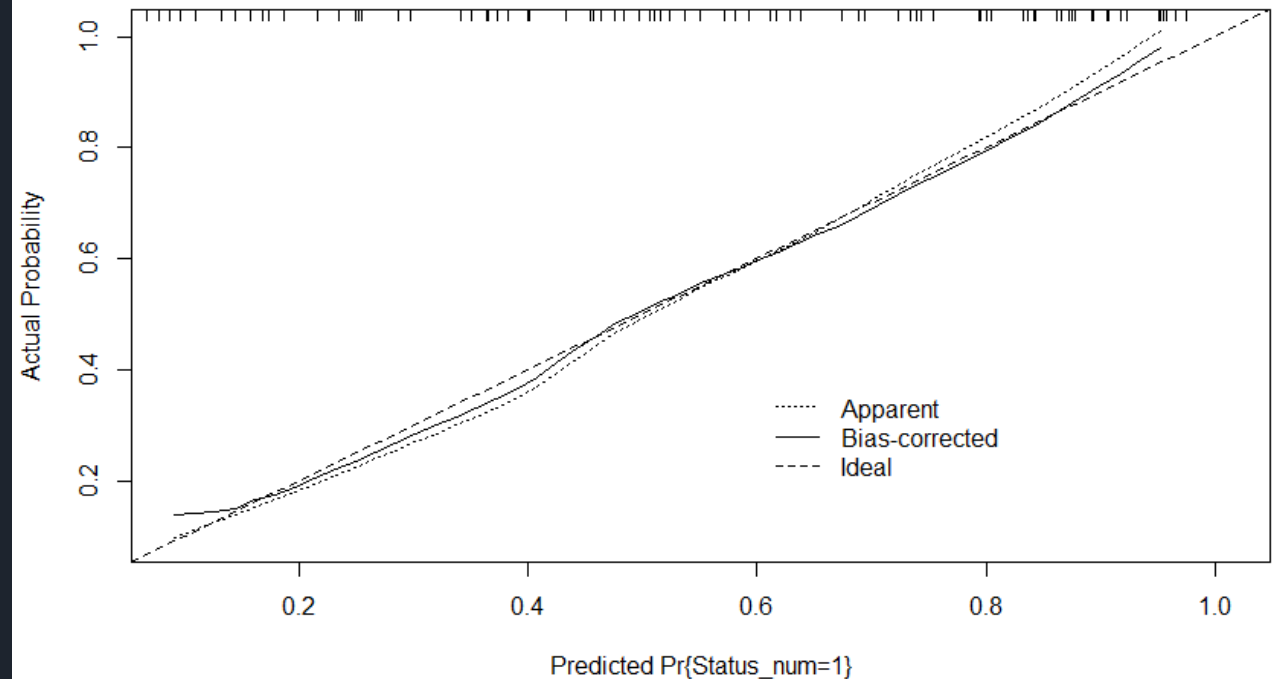
Prediction (Resampling Plot)

- The Brier score mildly increased but the c-statistic mildly decreased with repeated resampling
- The calibration curve shows that the predictions are near the ideal across the range of predicted values
- This suggests we can expect to predict patients with chronic fatigue from healthy controls with reasonable accuracy
- Must be from similar population using top three principal components

```
> plot(cal)

n=84  Mean absolute error=0.011  Mean squared error=0.00021
0.9 quantile of absolute error=0.022

> #Output pred. probs
> head(predict(pen_m1, type = "fitted"))
[1] 0.4560689 0.4689260 0.1137757 0.6098891 0.8683044 0.2314747
```



B= 200 repetitions, boot

Prediction (Sebal)

- "sebal implements a forward-selection method for the identification of two groups of taxa whose relative abundance, or balance, is associated with the response variable of interest."

```
> #Run selbal
> cv_selbal <- selbal::selbal.cv(x = data.frame(t(data.frame(phyloseq::otu_table(ps_family)))),
+                               y = phyloseq::sample_data(ps_family)$Status,
+                               n.fold = 5, n.iter = 1)

#####
STARTING selbal.cv FUNCTION
#####

#-----#
# ZERO REPLACEMENT . . .

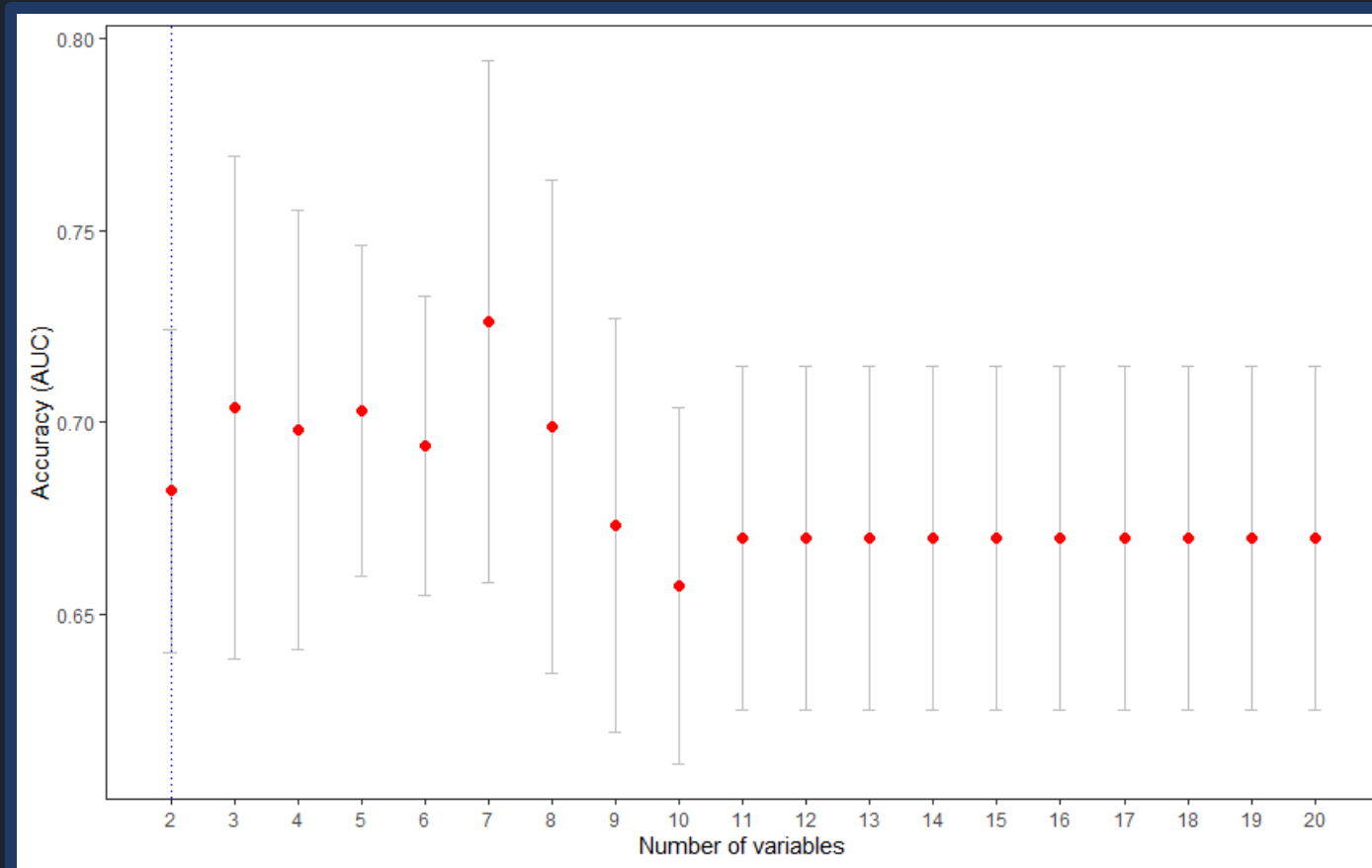
, . . . FINISHED.
#-----#

#-----#
# Starting the cross - validation procedure . . .
. . . finished.
#-----#
#####

The optimal number of variables is: 2
```

Prediction (Sebal Accuracy per Variable)

- Shows the accuracy of the system per variable
- It was already decided that two variables would be used so we focus on that for the balance graph



Prediction (Sebal Balance Graph)

- cross-validation shows two balance objects as having the relatively best rank-discrimination
- erysipelotrichaceae in the numerator and bifidobacteriaceae in the denominator
- The AUC was 0.77, but as low as AUC = 0.68 with 1 repeat of 5 fold cross-validation

