16s analysis-Visualization

Tsionkis Georgios

```
library(qiime2R)
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
library(extrafont)
library("phyloseq")
library("ggplot2")      # graphics
library(ggdendro)
```

Reading Artifacts

First, we will read the table of sequence variants (SVs). After that we will print the first 9 data (features and samples) just to be sure that everything is OK.

```
SVs <- read_qza("table.qza")
SVs$data[1:9,1:9]
```

```
##                                   A0C   A0S  A1C  A1S  A2C  A2S  A3C  A3S  A4C
## e8b7de4c07f308c714246ef5c937f9c9 5927     0 1402 1344 4157 6012 4189 4338 4094
## c0975f7ca5988524fffc584723730284 5367     0  625  488 2298 3065 1847 2279 3967
## 0b49ac64303dab0742b3623157119c9f    0 18591    0 2558    0 1610    0 4380    0
## bc7d0b055b853323f68f94738587dd31   31     0 5673 6515 1695 2175 1653 1455  547
## e926e73cb9002d9d0f8c18864bc04755   34     0 2870 3374  903 1102  847  683  289
## 55d80dfd205030315dba35be88e413b3    0     0  730 1319  206  428  189  290   63
## 8cb24777cb48dde0aac60dfeca125d10    0     0  401 2375  119  408   84  147   21
## 565b4f421328126f98e9f2aff31630f1  165     0  196  143   79   91   66   44   31
## 244210708ca539305df720d858624615  164     0    0    0   97  163   85  126  109
```

Reading Metadata

Then, we have to read the metadata file. This file is just a biological insight from our data. It contains some columns as : sampleID, Tissue (from were collect the sample), Treatment group (Treatment or control), Time point. Before we continue we have to change a little bit the format of our metadata file. In particular, the second line must have the #q2:types, that is the type of each of our data. It can be either numeric or categorical type. In our example, we have only categorical type.

```
metadata <- read_q2metadata("GRBR_New16S_Metadata.tsv")
head(metadata) #show top lines of metadata
```

```
##   SampleID        Tissue Treatment_Group Timepoint
## 1      A0C         Flour         Control        T0
## 2      A1C     Sourdough         Control        T1
## 3      A2C         Dough         Control        T2
## 4      A3C Proofed dough         Control        T3
## 5      A4C         Bread         Control        T4
## 6      B0C         Flour         Control        T0
```

Reading taxonomy

```
taxonomy <- read_qza("classification.qza")
head(taxonomy$data)
```

```
##                         Feature.ID
## 1 e8b7de4c07f308c714246ef5c937f9c9
## 2 c0975f7ca5988524fffc584723730284
## 3 0b49ac64303dab0742b3623157119c9f
## 4 bc7d0b055b853323f68f94738587dd31
## 5 e926e73cb9002d9d0f8c18864bc04755
## 6 55d80dfd205030315dba35be88e413b3
##
## 1       d__Bacteria; p__Proteobacteria; c__Alphaproteobacteria; o__Rickettsiales; f__Mitochondria; g
## 2                                       d__Bacteria; p__Cyanobacteria; c__Cyanobacteriia; o__Chloropl
## 3             d__Bacteria; p__Firmicutes; c__Bacilli; o__Lactobacillales; f__Lactobacillaceae; g__Lac
## 4                       d__Bacteria; p__Proteobacteria; c__Gammaproteobacteria; o__Enterobacterales;
## 5                       d__Bacteria; p__Proteobacteria; c__Gammaproteobacteria; o__Enterobacterales;
## 6 d__Bacteria; p__Proteobacteria; c__Gammaproteobacteria; o__Enterobacterales; f__Enterobacteriaceae
##   Confidence
## 1  0.8456021
## 2  1.0000000
## 3  0.9958289
## 4  0.9768360
## 5  0.9575058
## 6  0.7288714
```

Due to the fact that a single string is returned we want to break up this string

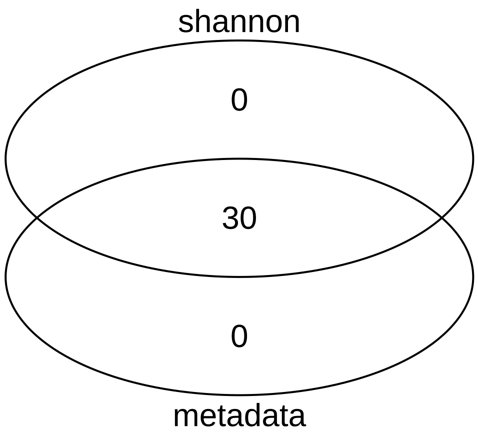
```
taxonomy <- parse_taxonomy(taxonomy$data)
head(taxonomy)
```

```
##                                      Kingdom         Phylum               Class
## e8b7de4c07f308c714246ef5c937f9c9 d__Bacteria Proteobacteria Alphaproteobacteria
## c0975f7ca5988524fffc584723730284 d__Bacteria  Cyanobacteria      Cyanobacteriia
## 0b49ac64303dab0742b3623157119c9f d__Bacteria     Firmicutes             Bacilli
## bc7d0b055b853323f68f94738587dd31 d__Bacteria Proteobacteria Gammaproteobacteria
## e926e73cb9002d9d0f8c18864bc04755 d__Bacteria Proteobacteria Gammaproteobacteria
## 55d80dfd205030315dba35be88e413b3 d__Bacteria Proteobacteria Gammaproteobacteria
##                                             Order             Family
## e8b7de4c07f308c714246ef5c937f9c9    Rickettsiales       Mitochondria
## c0975f7ca5988524fffc584723730284      Chloroplast        Chloroplast
## 0b49ac64303dab0742b3623157119c9f  Lactobacillales   Lactobacillaceae
## bc7d0b055b853323f68f94738587dd31 Enterobacterales Enterobacteriaceae
## e926e73cb9002d9d0f8c18864bc04755 Enterobacterales Enterobacteriaceae
## 55d80dfd205030315dba35be88e413b3 Enterobacterales Enterobacteriaceae
##                                           Genus              Species
## e8b7de4c07f308c714246ef5c937f9c9  Mitochondria    Triticum_aestivum
## c0975f7ca5988524fffc584723730284   Chloroplast                 <NA>
## 0b49ac64303dab0742b3623157119c9f Lactobacillus Lactobacillus_brevis
## bc7d0b055b853323f68f94738587dd31     Kosakonia                 <NA>
## e926e73cb9002d9d0f8c18864bc04755     Kosakonia                 <NA>
## 55d80dfd205030315dba35be88e413b3     Kosakonia    Kosakonia_cowanii
```

We will check if all the samples have an assigned Shannon diversity value. Shannon diversity index tells you how diverse the species in a given community are. It rises with the number of species and the evenness of their abundance.

```
shannon <- read_qza("shannon_vector.qza")

shannon<-shannon$data %>% rownames_to_column("SampleID") # this moves the sample names to a new column
gplots::venn(list(metadata=metadata$SampleID, shannon=shannon$SampleID))
```



We will add to the metadata a column with shannon index

```
metadata<-metadata %>%
          left_join(shannon)
```

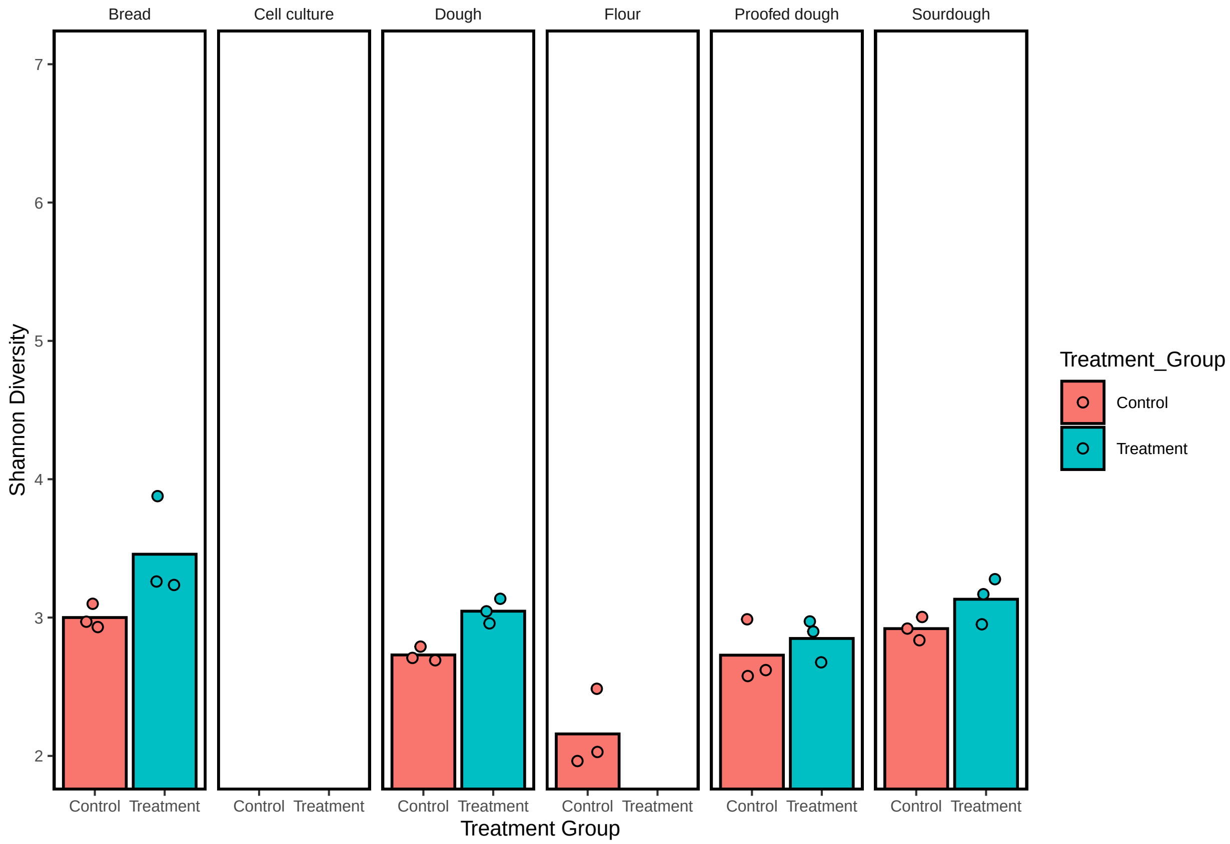
```
## Joining, by = "SampleID"
```

```
head(metadata)
```

```
##   SampleID        Tissue Treatment_Group Timepoint shannon_entropy
## 1      A0C         Flour         Control        T0        2.485254
## 2      A1C     Sourdough         Control        T1        2.836062
## 3      A2C         Dough         Control        T2        2.708242
## 4      A3C Proofed dough         Control        T3        2.620157
## 5      A4C         Bread         Control        T4        2.931268
## 6      B0C         Flour         Control        T0        1.962765
```

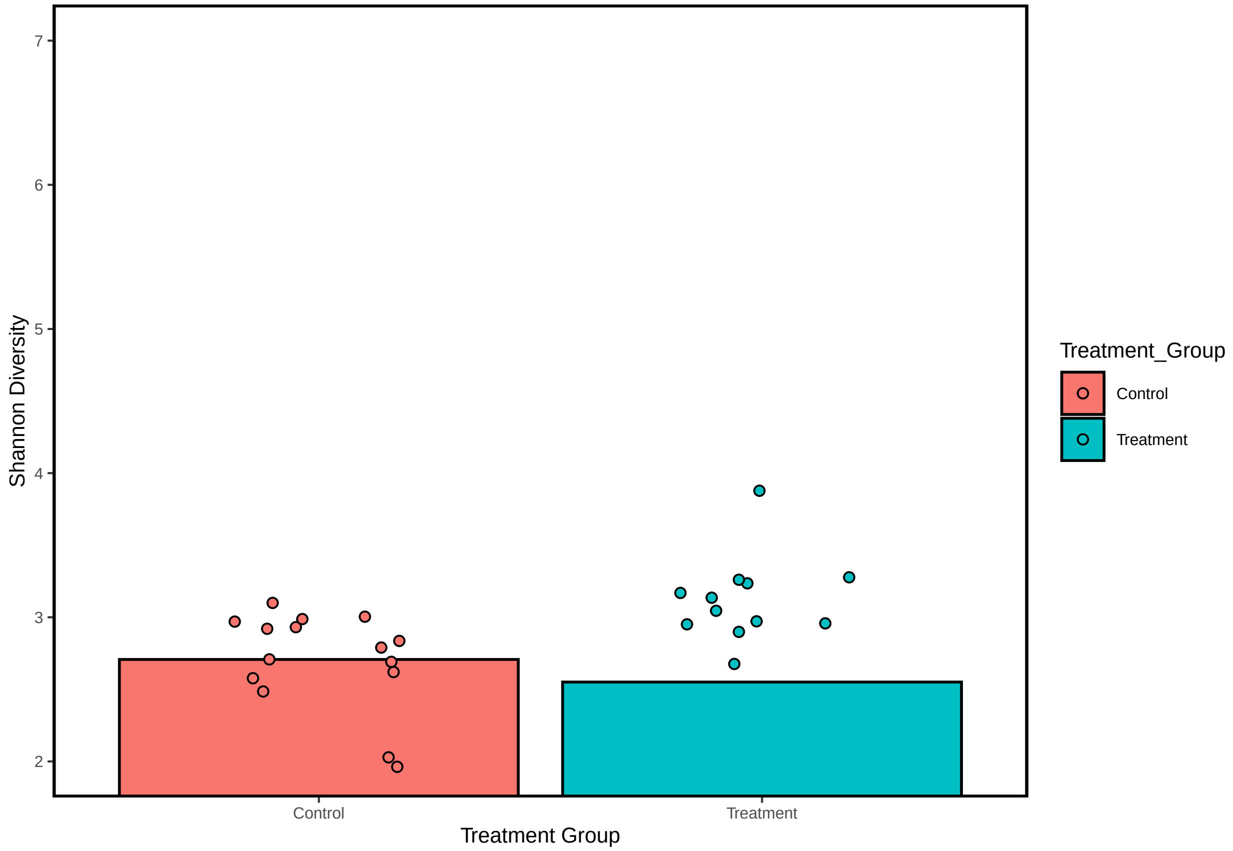
In the next step, we want to find if the treatment of our samples has an effect on our diversity through Shannon Diversity index. Shannon index is an alpha diversity metric which takes into account both richness and abundance across samples. Species richness refers to the number of species in a community. Species abundance refers to the number of individuals per species. As we can see from the plot below the biggest difference found in bread tissue between control and treatment. In cell culture, the Shannon index is <0.3 (which makes a lot of sense due to the fact that as we saw from barchart2.qzv in qiime, the samples from cell culture differ a lot from all the other samples) treatment samples thats why we are not able to see the bar.

```
metadata %>%
  filter(!is.na(shannon_entropy)) %>%
  ggplot(aes(x=`Treatment_Group`, y=shannon_entropy, fill=`Treatment_Group`)) +
  stat_summary(geom="bar", fun.data=mean_se, color="black") + #here black is the outline for the bars
  geom_jitter(shape=21, width=0.2, height=0) +
  coord_cartesian(ylim=c(2,7)) + # adjust y-axis
  facet_grid(~`Tissue`) + # create a panel for each Tissue
  xlab("Treatment Group") +
  ylab("Shannon Diversity") +
  theme_q2r()
```



Also, we would like to see the overall diversity between control and treatment group. As we can see there is not such a big difference between these two groups

```
metadata %>%
  filter(!is.na(shannon_entropy)) %>%
  ggplot(aes(x=`Treatment_Group`, y=shannon_entropy, fill=`Treatment_Group`)) +
  stat_summary(geom="bar", fun.data=mean_se, color="black") + #here black is the outline for the bars
  geom_jitter(shape=21, width=0.2, height=0) +
  coord_cartesian(ylim=c(2,7)) + # adjust y-axis
  xlab("Treatment Group") +
  ylab("Shannon Diversity") +
  theme_q2r()
```

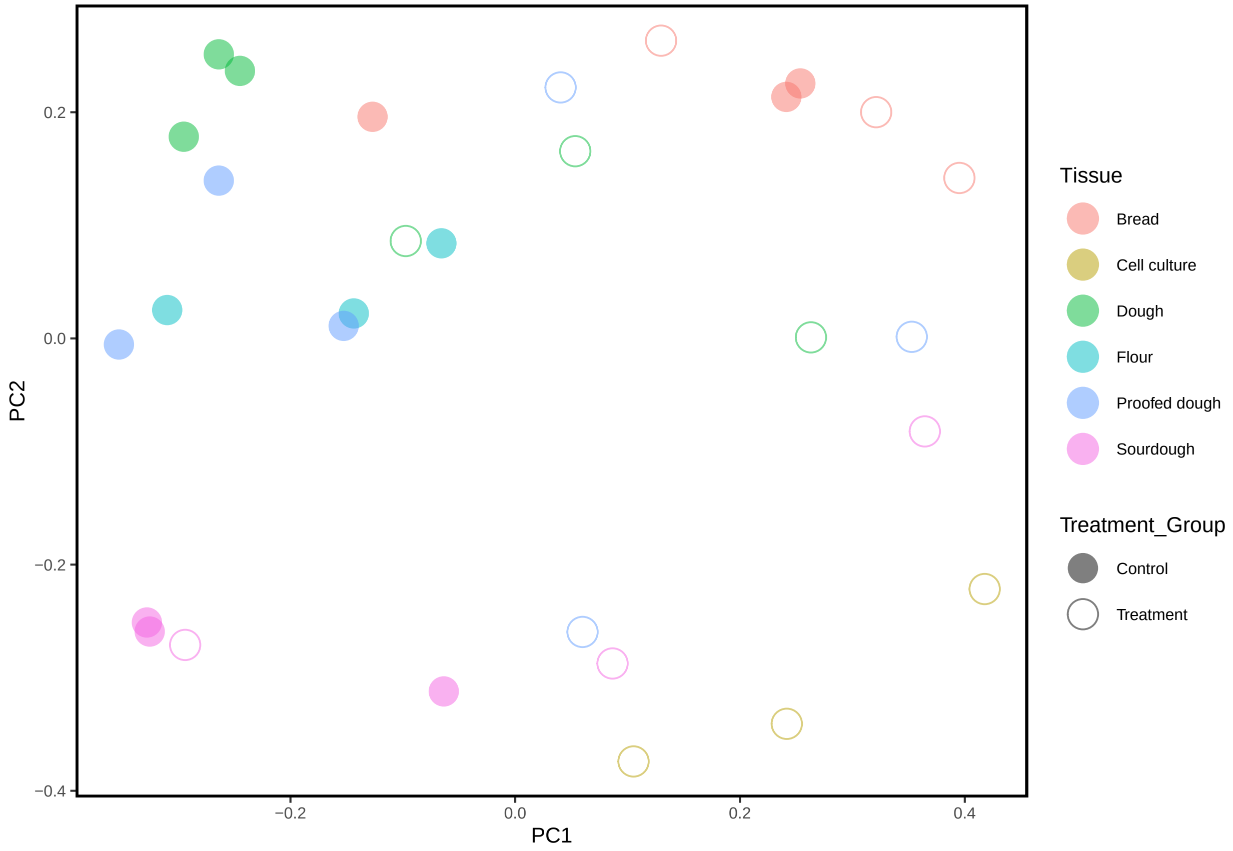


PCoA plot (Unweighted)

PCoA (Principal Coordinates analysis) is a method to explore and to visualize similarities or dissimilarities of data. It starts with a similarity matrix or dissimilarity matrix (= distance matrix) and assigns for each item a location in a low-dimensional space. Objects ordinated closer to one another are more similar than those ordinated further away. The main difference with PCA is that PCA based on the Euclidean distance, and PCoA is based on distances other than the Euclidean distance, and finds the potential principal components that affect the difference in the composition of the sample community through dimensionality reduction. We will use unweighted_unifrac_pcoa_results.qza from the diversity metrics that we calculate with qiime2.

```
uwunifrac<-read_qza("unweighted_unifrac_pcoa_results.qza")
uwunifrac$data$Vectors %>%
  select(SampleID, PC1, PC2) %>%
  left_join(metadata) %>%
  ggplot(aes(x=PC1, y=PC2, color=`Tissue`, shape=`Treatment_Group`)) +
  geom_point(alpha=0.5, size = 5 ) + #alpha controls transparency and helps when points are overlapping
  theme_q2r() +
  scale_shape_manual(values=c(16,1), name="Treatment_Group") +
  scale_color_discrete(name="Tissue")
```

```
## Joining, by = "SampleID"
```

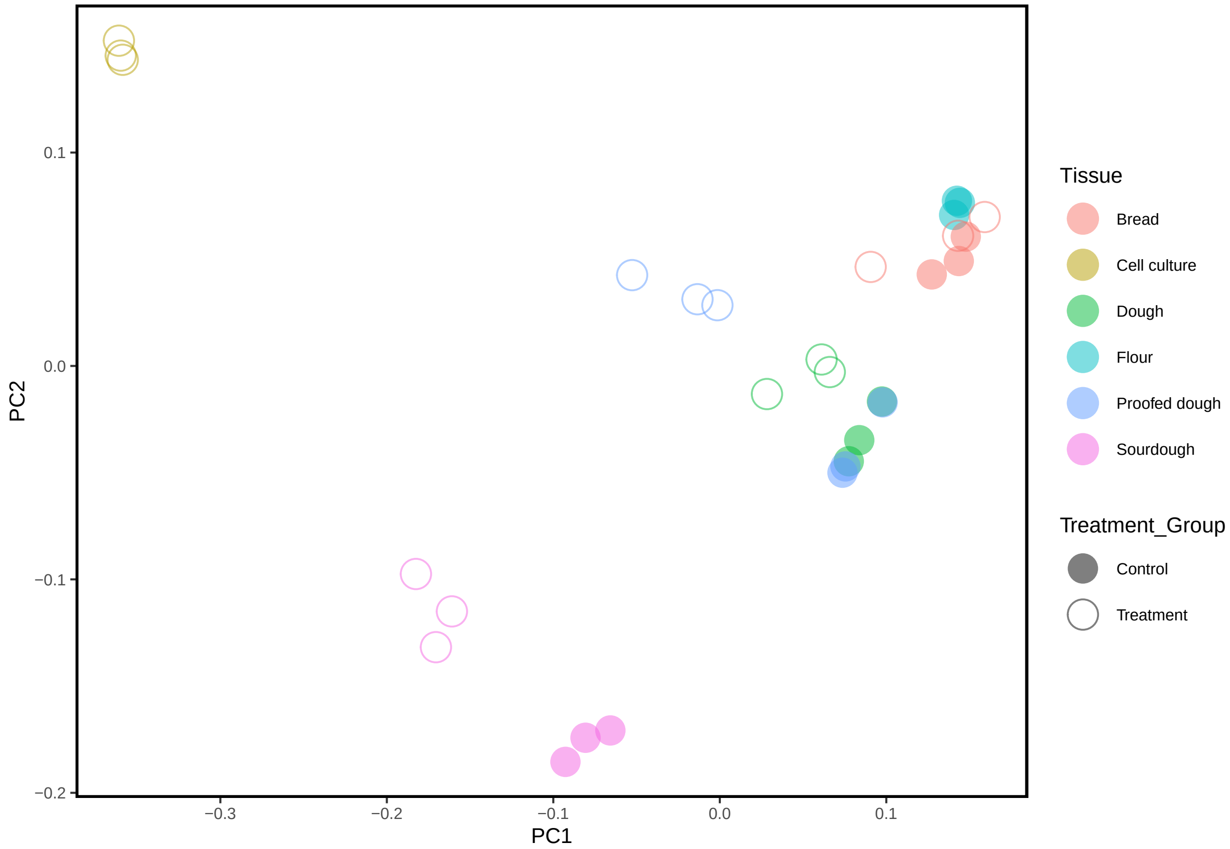


PCoA plot (Weighted)

Here we will use weighted_unifrac_pcoa_results.qza from qiime2. Weighted (quantitative) accounts for abundance of observed organisms and unweighted (qualitative) is based on their presence or absence. In practice, this means that Weighted UniFrac is useful for examining differences in community structure, Unweighted UniFrac is more sensitive to differences in low-abundance features. Hence, I think that in our analysis Weighted PCoA is more useful.

```
uwunifrac<-read_qza("weighted_unifrac_pcoa_results.qza")
uwunifrac$data$Vectors %>%
  select(SampleID, PC1, PC2) %>%
  left_join(metadata) %>%
  ggplot(aes(x=PC1, y=PC2, color=`Tissue`, shape=`Treatment_Group`)) +
  geom_point(alpha=0.5, size = 5) + #alpha controls transparency and helps when points are overlapping
  theme_q2r() +
  scale_shape_manual(values=c(16,1), name="Treatment_Group") +
  scale_color_discrete(name="Tissue")
```

```
## Joining, by = "SampleID"
```



We are able to see that there is a clear difference between cell culture tissue and all the other samples. Moreover, there is a visible difference in each tissue among control and treatment Treatment group, except from Bread where there is a quite good similarity between two groups.

Creating a Phyloseq Object and abundance bar plot

Kingdom

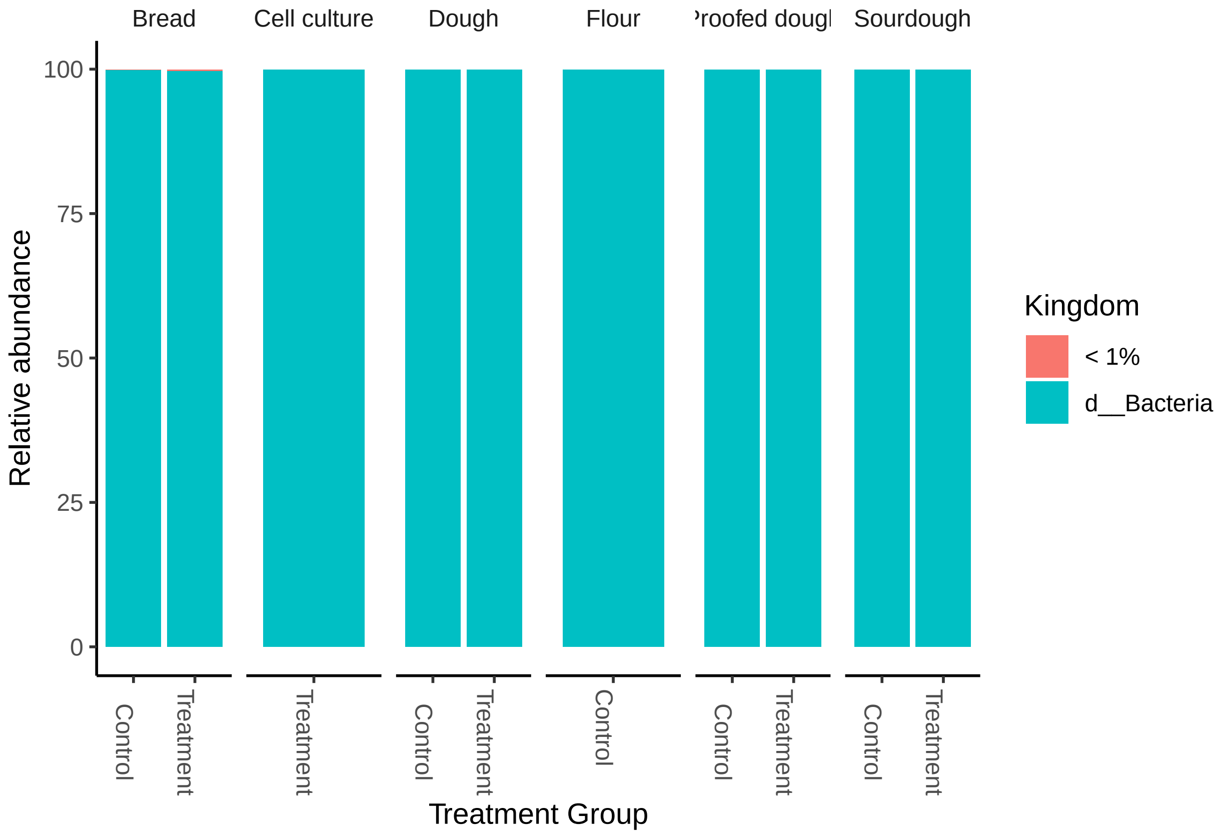
In the graphs below, we see the presence of some of the most abundant bacteria at each taxonomic level divided into categories according to tissue for two treatments group (Control-Treatment)

```
physeq <- qza_to_phyloseq(features = "NoMitoNoChloroNoUnass_table.qza",
                          tree = "16S_rooted_tree.qza", "classification.qza",
                          metadata = "GRBR_New16S_Metadata.tsv")


#Normalization, the normalization function was x/sum(x)*100, however I want to visualize my graph group
physeq.rel = transform_sample_counts(physeq, function(x) x/sum(x)*33.3)
# agglomerate taxa
glom <- tax_glom(physeq.rel, taxrank = 'Kingdom', NArm = FALSE)
physeq.melt <- psmelt(glom)
# change to character for easy-adjusted level
physeq.melt$Kingdom <- as.character(physeq.melt$Kingdom)

physeq.melt <- physeq.melt %>%
  group_by(Tissue, Kingdom, Treatment_Group) %>%
  mutate(median=median(Abundance))
# select group median > 1
keep <- unique(physeq.melt$Kingdom[physeq.melt$median > 1])
physeq.melt$Kingdom[!(physeq.melt$Kingdom %in% keep)] <- "< 1%"
#to get the same rows together
physeq.melt_sum <- physeq.melt %>%
  group_by(Sample,Tissue, Treatment_Group, Kingdom) %>%
  summarise(Abundance=sum(Abundance))

ggplot(physeq.melt_sum, aes(x = Treatment_Group, y = Abundance, fill = Kingdom)) +
  geom_bar(stat = "identity", aes(fill=Kingdom)) +
  labs(x="Treatment Group", y="Relative abundance") +
  facet_wrap(~Tissue, scales= "free_x", nrow=1) +
  theme_classic() +
  theme(strip.background = element_blank(),
        axis.text.x.bottom = element_text(angle = -90))
```



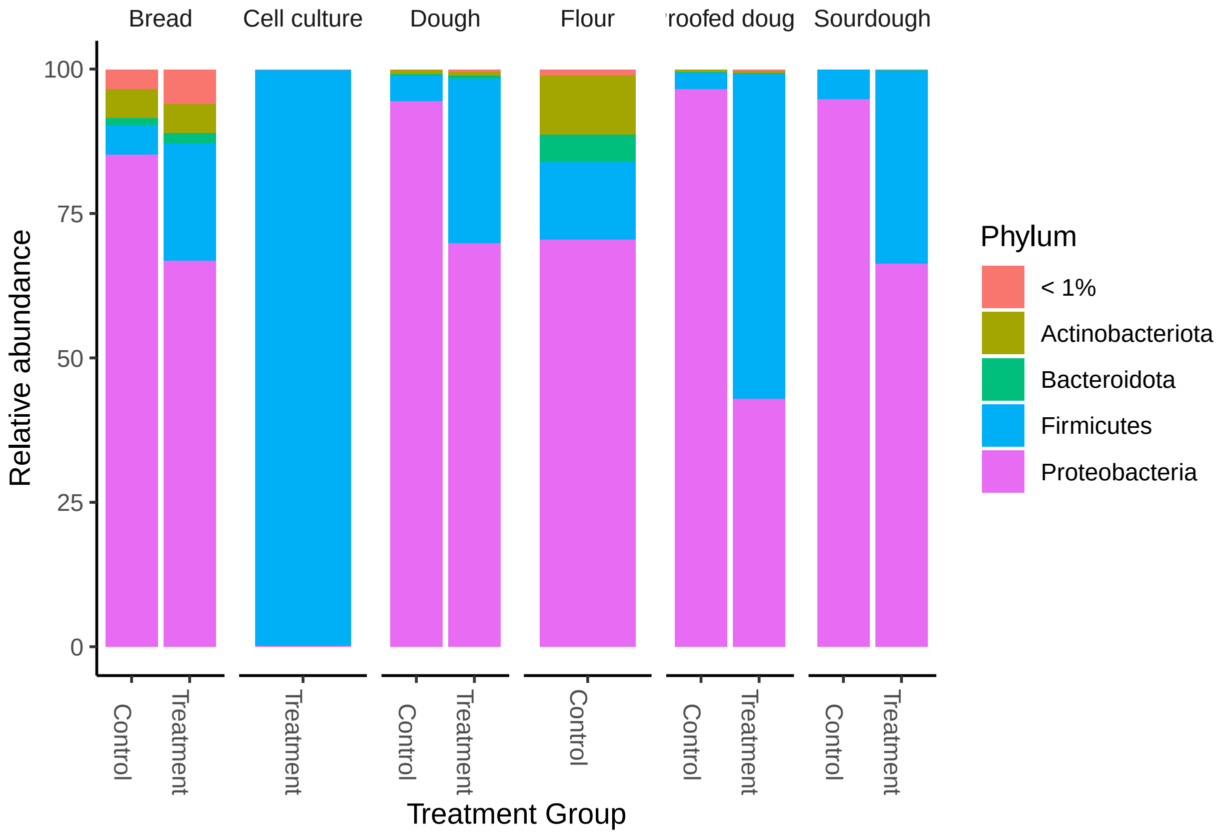
```

#Normalization, the normalization function was x/sum(x)*100, however I want to visualize my graph group
physeq.rel = transform_sample_counts(physeq, function(x) x/sum(x)*33.3)
# agglomerate taxa
glom <- tax_glom(physeq.rel, taxrank = 'Phylum', NArm = FALSE)
```

```
physeq.melt <- psmelt(glom)
# change to character for easy-adjusted level
physeq.melt$Phylum <- as.character(physeq.melt$Phylum)

physeq.melt <- physeq.melt %>%
  group_by(Tissue, Phylum, Treatment_Group) %>%
  mutate(median=median(Abundance))
# select group median > 1
keep <- unique(physeq.melt$Phylum[physeq.melt$median > 1])
physeq.melt$Phylum[!(physeq.melt$Phylum %in% keep)] <- "< 1%"
#to get the same rows together
physeq.melt_sum <- physeq.melt %>%
  group_by(Sample,Tissue, Treatment_Group, Phylum) %>%
  summarise(Abundance=sum(Abundance))

ggplot(physeq.melt_sum, aes(x = Treatment_Group, y = Abundance, fill = Phylum)) +
  geom_bar(stat = "identity", aes(fill=Phylum)) +
  labs(x="Treatment Group", y="Relative abundance") +
  facet_wrap(~Tissue, scales= "free_x", nrow=1) +
  theme_classic() +
  theme(strip.background = element_blank(),
        axis.text.x.bottom = element_text(angle = -90))
```



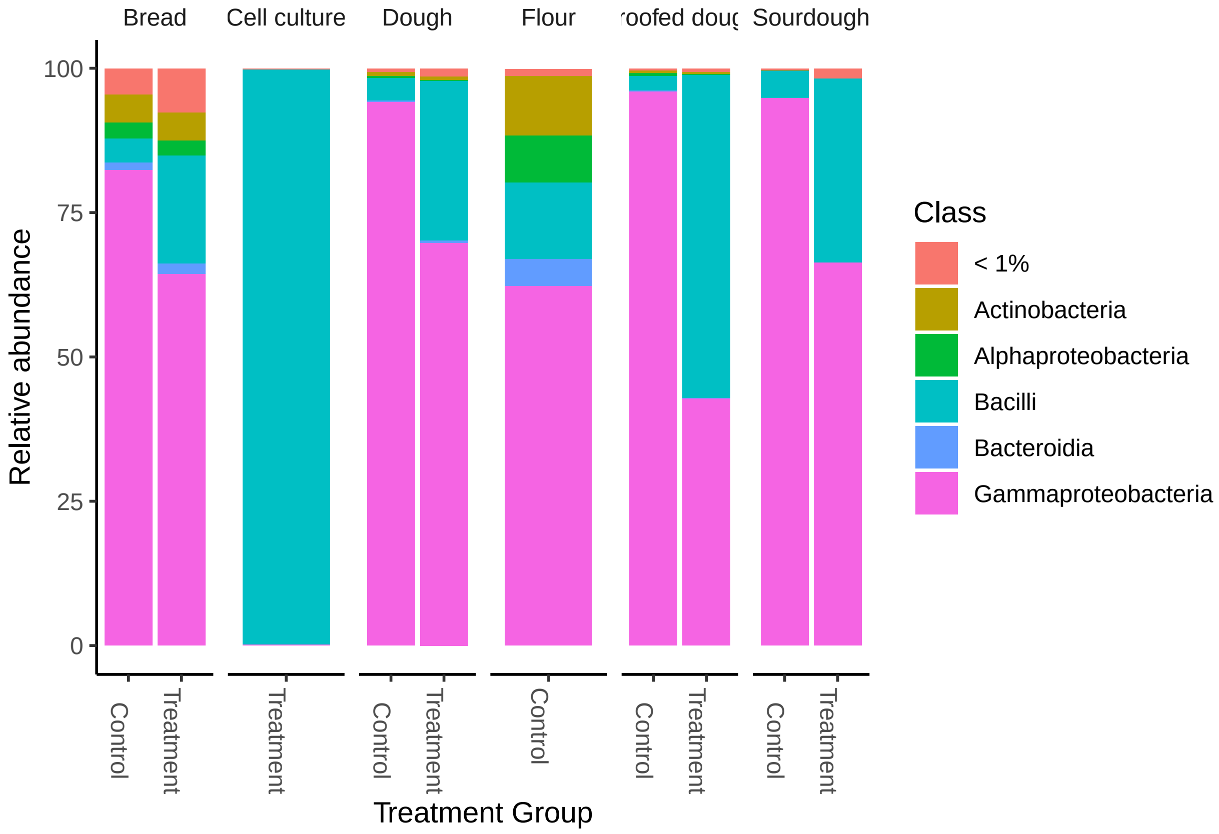
Phylum

Class

```r
#Normalization, the normalization function was x/sum(x)*100, however I want to visualize my graph group
physeq.rel = transform_sample_counts(physeq, function(x) x/sum(x)*33.3)
# agglomerate taxa
glom <- tax_glom(physeq.rel, taxrank = 'Class', NArm = FALSE)
physeq.melt <- psmelt(glom)
# change to character for easy-adjusted level
physeq.melt$Class <- as.character(physeq.melt$Class)

physeq.melt <- physeq.melt %>%
  group_by(Tissue, Class, Treatment_Group) %>%
  mutate(median=median(Abundance))
# select group median > 1
keep <- unique(physeq.melt$Class[physeq.melt$median > 1])
physeq.melt$Class[!(physeq.melt$Class %in% keep)] <- "< 1%"
#to get the same rows together
physeq.melt_sum <- physeq.melt %>%
  group_by(Sample,Tissue, Treatment_Group, Class) %>%
  summarise(Abundance=sum(Abundance))

ggplot(physeq.melt_sum, aes(x = Treatment_Group, y = Abundance, fill = Class)) +
  geom_bar(stat = "identity", aes(fill=Class)) +
  labs(x="Treatment Group", y="Relative abundance") +
  facet_wrap(~Tissue, scales= "free_x", nrow=1) +
  theme_classic() +
  theme(strip.background = element_blank(),
        axis.text.x.bottom = element_text(angle = -90))
```

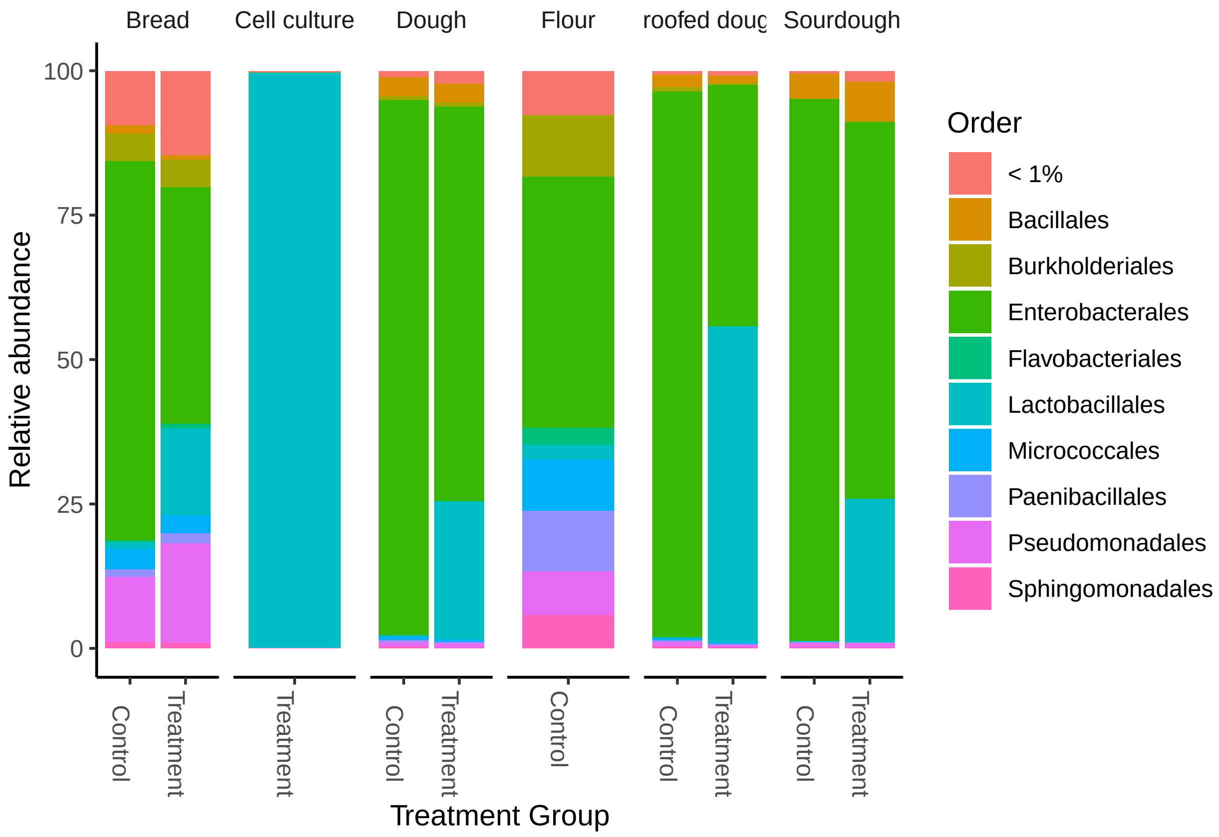


Order

```
#Normalization, the normalization function was x/sum(x)*100, however I want to visualize my graph group
physeq.rel = transform_sample_counts(physeq, function(x) x/sum(x)*33.3)
# agglomerate taxa
glom <- tax_glom(physeq.rel, taxrank = 'Order', NArm = FALSE)
physeq.melt <- psmelt(glom)
# change to character for easy-adjusted level
physeq.melt$Order <- as.character(physeq.melt$Order)

physeq.melt <- physeq.melt %>%
  group_by(Tissue, Order, Treatment_Group) %>%
  mutate(median=median(Abundance))
# select group median > 1
keep <- unique(physeq.melt$Order[physeq.melt$median > 1])
physeq.melt$Order[!(physeq.melt$Order %in% keep)] <- "< 1%"
#to get the same rows together
physeq.melt_sum <- physeq.melt %>%
  group_by(Sample,Tissue, Treatment_Group, Order) %>%
  summarise(Abundance=sum(Abundance))

ggplot(physeq.melt_sum, aes(x = Treatment_Group, y = Abundance, fill = Order)) +
  geom_bar(stat = "identity", aes(fill=Order)) +
  labs(x="Treatment Group", y="Relative abundance") +
  facet_wrap(~Tissue, scales= "free_x", nrow=1) +
  theme_classic() +
  theme(strip.background = element_blank(),
        axis.text.x.bottom = element_text(angle = -90))
```

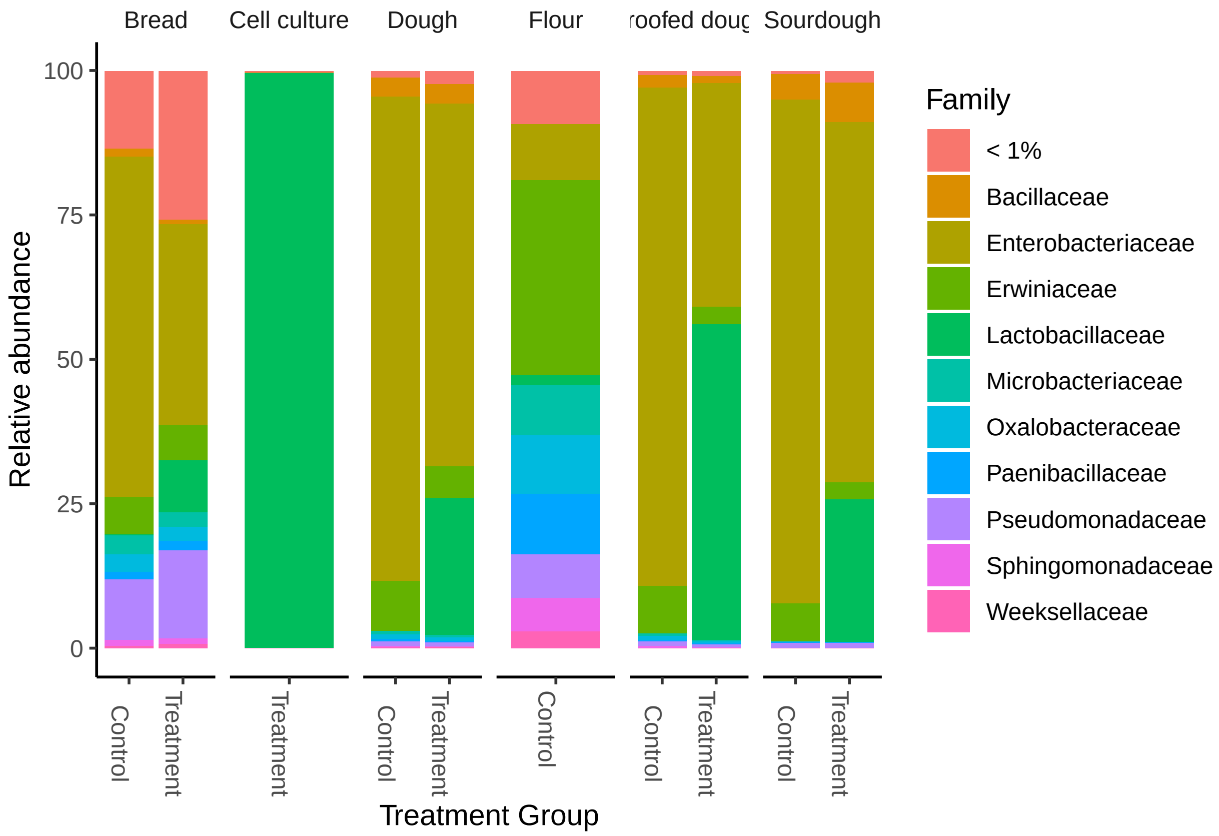


Family

```
#Normalization, the normalization function was x/sum(x)*100, however I want to visualize my graph group
physeq.rel = transform_sample_counts(physeq, function(x) x/sum(x)*33.3)
# agglomerate taxa
glom <- tax_glom(physeq.rel, taxrank = 'Family', NArm = FALSE)
physeq.melt <- psmelt(glom)
# change to character for easy-adjusted level
physeq.melt$Family <- as.character(physeq.melt$Family)

physeq.melt <- physeq.melt %>%
  group_by(Tissue, Family, Treatment_Group) %>%
  mutate(median=median(Abundance))
# select group median > 1
keep <- unique(physeq.melt$Family[physeq.melt$median > 1])
physeq.melt$Family[!(physeq.melt$Family %in% keep)] <- "< 1%"
#to get the same rows together
physeq.melt_sum <- physeq.melt %>%
  group_by(Sample,Tissue, Treatment_Group, Family) %>%
  summarise(Abundance=sum(Abundance))

ggplot(physeq.melt_sum, aes(x = Treatment_Group, y = Abundance, fill = Family)) +
  geom_bar(stat = "identity", aes(fill=Family)) +
  labs(x="Treatment Group", y="Relative abundance") +
  facet_wrap(~Tissue, scales= "free_x", nrow=1) +
  theme_classic() +
  theme(strip.background = element_blank(),
        axis.text.x.bottom = element_text(angle = -90))
```

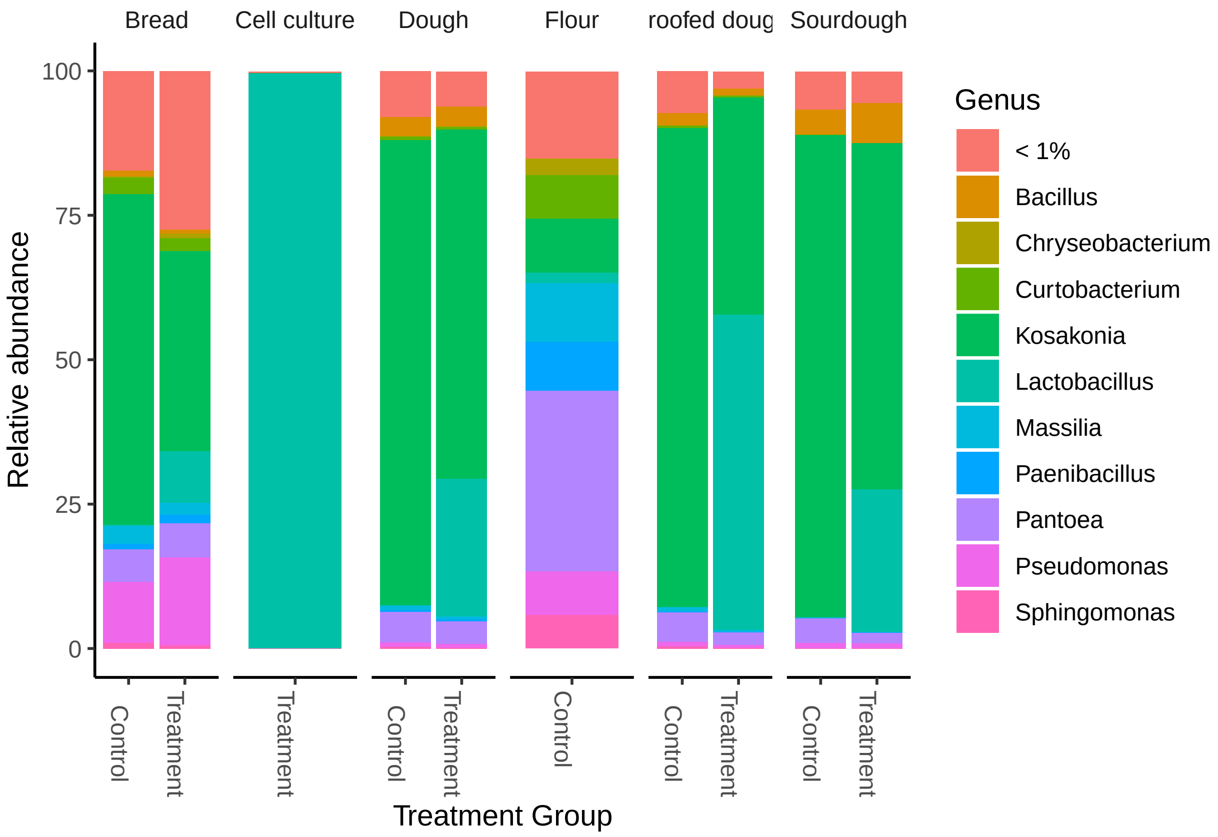


```
#Normalization, the normalization function was x/sum(x)*100, however I want to visualize my graph group
physeq.rel = transform_sample_counts(physeq, function(x) x/sum(x)*33.3)
# agglomerate taxa
glom <- tax_glom(physeq.rel, taxrank = 'Genus', NArm = FALSE)
physeq.melt <- psmelt(glom)
# change to character for easy-adjusted level
physeq.melt$Genus <- as.character(physeq.melt$Genus)

physeq.melt <- physeq.melt %>%
  group_by(Tissue, Genus, Treatment_Group) %>%
  mutate(median=median(Abundance))
```

```r
# select group median > 1
keep <- unique(physeq.melt$Genus[physeq.melt$median > 1])
physeq.melt$Genus[!(physeq.melt$Genus %in% keep)] <- "< 1%"
#to get the same rows together
physeq.melt_sum <- physeq.melt %>%
  group_by(Sample,Tissue, Treatment_Group, Genus) %>%
  summarise(Abundance=sum(Abundance))

ggplot(physeq.melt_sum, aes(x = Treatment_Group, y = Abundance, fill = Genus)) +
  geom_bar(stat = "identity", aes(fill=Genus)) +
  labs(x="Treatment Group", y="Relative abundance") +
  facet_wrap(~Tissue, scales= "free_x", nrow=1) +
  theme_classic() +
  theme(strip.background = element_blank(),
        axis.text.x.bottom = element_text(angle = -90))
```



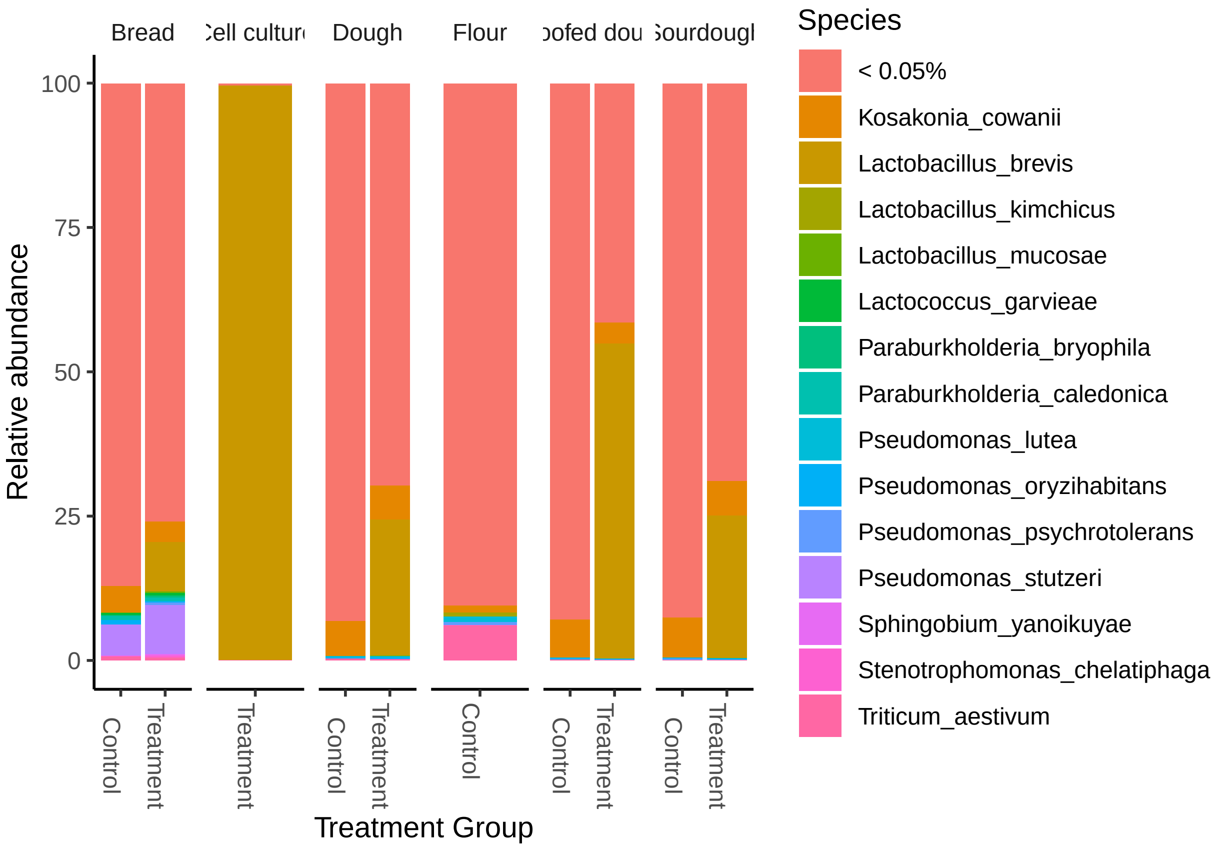
Genus

```r
#Normalization, the normalization function was x/sum(x)*100, however I want to visualize my graph group
physeq.rel = transform_sample_counts(physeq, function(x) x/sum(x)*33.3)
# agglomerate taxa
glom <- tax_glom(physeq.rel, taxrank = 'Species', NArm = FALSE)
physeq.melt <- psmelt(glom)
# change to character for easy-adjusted level
```

```
physeq.melt$Species <- as.character(physeq.melt$Species)

physeq.melt <- physeq.melt %>%
  group_by(Tissue, Species, Treatment_Group) %>%
  mutate(median=median(Abundance))
# select group median > 1
keep <- unique(physeq.melt$Species[physeq.melt$median > 0.05])
physeq.melt$Species[!(physeq.melt$Species %in% keep)] <- "< 0.05%"
#to get the same rows together
physeq.melt_sum <- physeq.melt %>%
  group_by(Sample,Tissue, Treatment_Group, Species) %>%
  summarise(Abundance=sum(Abundance))

ggplot(physeq.melt_sum, aes(x = Treatment_Group, y = Abundance, fill = Species)) +
  geom_bar(stat = "identity", aes(fill=Species)) +
  labs(x="Treatment Group", y="Relative abundance") +
  facet_wrap(~Tissue, scales= "free_x", nrow=1) +
  theme_classic() +
  theme(strip.background = element_blank(),
        axis.text.x.bottom = element_text(angle = -90))
```



Species

Heat Map

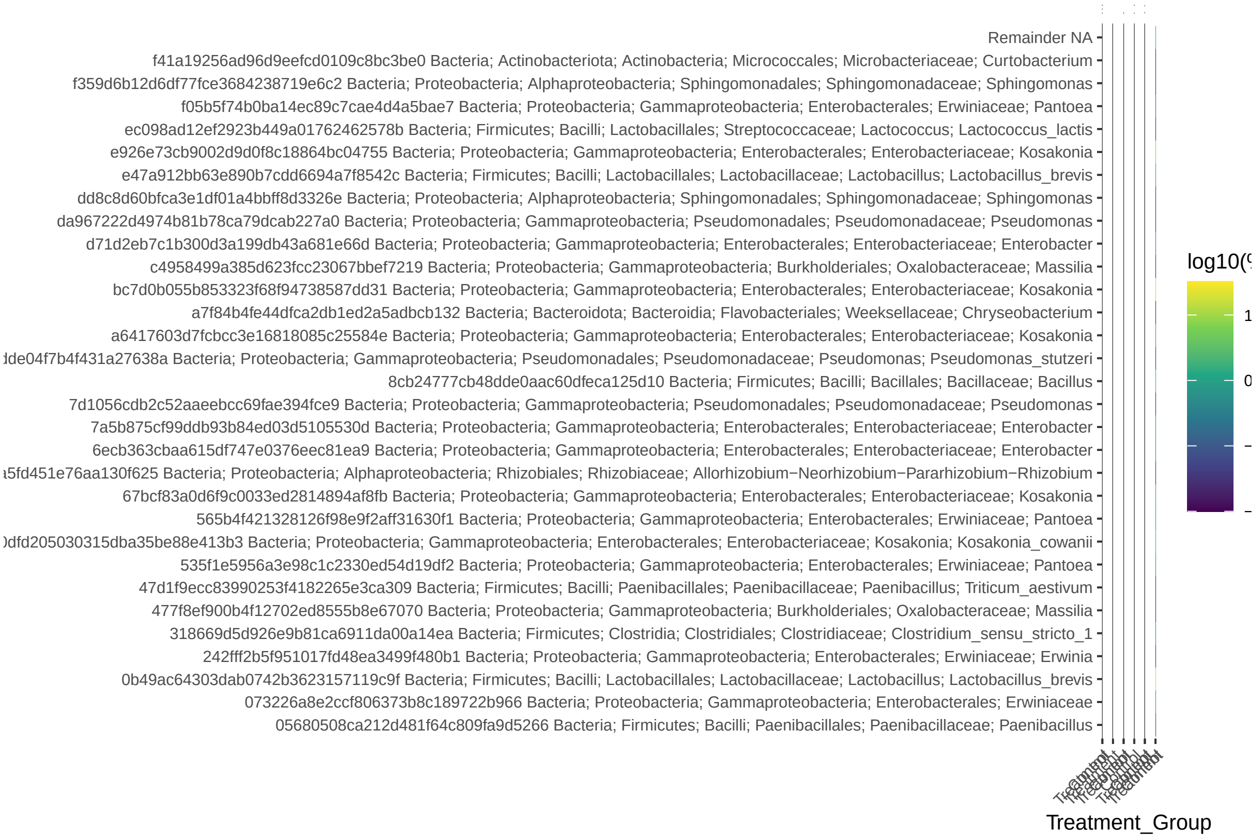
In heat maps the data is displayed in a grid where each row represents a bacterium and each column represents a sample. The colour and intensity of the boxes is used to represent changes of bacterium abundance. Check heatmap.pdf

```
metadata <- read_q2metadata("GRBR_New16S_Metadata.tsv")
SVs <- read_qza("NoMitoNoChloroNoUnass_table.qza")$data
taxonomy <- read_qza("classification.qza")$data

SVs<-apply(SVs, 2, function(x) x/sum(x)*33.3) #convert to percent

SVsToPlot<-
  data.frame(MeanAbundance=rowMeans(SVs)) %>% #find the average abundance of a SV
  rownames_to_column("Feature.ID") %>%
  arrange(desc(MeanAbundance)) %>%
  top_n(30, MeanAbundance) %>%
  pull(Feature.ID) #extract only the names from the table

SVs %>%
  as.data.frame() %>%
  rownames_to_column("Feature.ID") %>%
  gather(-Feature.ID, key="SampleID", value="Abundance") %>%
  mutate(Feature.ID=if_else(Feature.ID %in% SVsToPlot,  Feature.ID, "Remainder")) %>% #flag features to
  group_by(SampleID, Feature.ID) %>%
  summarize(Abundance=sum(Abundance)) %>%
  left_join(metadata) %>%
  mutate(NormAbundance=log10(Abundance+0.01)) %>% # do a log10 transformation after adding a 0.01% pseu
  left_join(taxonomy) %>%
  mutate(Feature=paste(Feature.ID, Taxon)) %>%
  mutate(Feature=gsub("[dkpcofgs]__", "", Feature)) %>% # trim out leading text from taxonomy string
  ggplot(aes(x=Treatment_Group, y=Feature, fill=NormAbundance)) +
  geom_tile() +
  facet_grid(~`Tissue`, scales="free_x") +
  theme_q2r() +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  scale_fill_viridis_c(name="log10(% Abundance)")
```



```
ggsave("heatmap.pdf", height=4, width=11, device="pdf") # save a PDF 3 inches by 4 inches
```