Nanopore\_16S\_rRNA\_analysis

## Analysing nanopore 16S rRNA sequences

Data source is from a article “RESCUE: a validated Nanopore pipeline to classify bacteria through long-read, 16S-ITS-23S rRNA sequencing.” by Petrone et. al (2023). Sequences were downloaded from SRA and grouped into three separate sets.

Link to the article: https://doi.org/10.3389/fmicb.2023.1201064

Zymo research mock community sample and gut mock community each contain 4 replicate samples sequenced on separate flow-cells. Oral microbiome set contain 20 saliva samples.

### Preprosessing

All reads were trimmed with cutadapt using the 1492R primer sequence

* cutadapt -g CGGTTACCTTGTTACGACTT –rc -o file input\_file

*Note, that sequences are reverse complement. Thus, 16S rRNA sequence starts from 3’ end*

*~85-90 % of total reads contained correct primer sequence on cutadapt default error rate 0.1.*

### Qiime 2 vsearch otu clustering workflow

Workflow: Importing sequences, dereplication, clustering, chimera check, chimera and singleton removal and taxa classification.

*Manifest file contains all sequence files with absolute path*

1. qiime tools import –type ‘SampleData[SequencesWithQuality]’  –input-path seq/manifest  –output-path output.qza –input-format SingleEndFastqManifestPhred33V2
2. qiime vsearch dereplicate-sequences –i-sequences seqs.qza –o-dereplicated-table dereplicated\_table.qza –o-dereplicated-sequences dereplicated\_sequences.qza
3. qiime vsearch cluster-features-de-novo –i-sequences dereplicated\_sequences.qza –i-table dereplicated\_table.qza –p-perc-identity 0.97 –o-clustered-sequences clustered\_otu\_sequences.qza –o-clustered-table clustered\_otu\_table.qza
4. qiime vsearch uchime-denovo –i-sequences clustered\_otu\_sequences.qza –output-dir chimeras –i-table clustered\_otu\_table.qza
5. qiime feature-table filter-features –i-table clustered\_otu\_table.qza –m-metadata-file chimeras/nonchimeras.qza —p-min-frequency 2  –o-filtered-table  final\_otu\_table.qza
6. qiime feature-table filter-seqs   –i-data clustered\_otu\_sequences.qza –i-table final\_otu\_table.qza  –o-filtered-data representative\_sequences.qza
7. qiime feature-classifier classify-sklearn –i-classifier silva-138-99-nb-classifier.qza –i-reads representative\_sequences.qza –p-read-orientation reverse –o-classification taxonomy.qza

### Qiime 2 dada denoising workflow

Workflow: Importing sequences, denoising and feature classification.

*Note. Using values 1-1.5 errors per 100 bp sequence produced best compromise.*

1. qiime tools import –type ‘SampleData[SequencesWithQuality]’  –input-path seq/manifest  –output-path output.qza –input-format SingleEndFastqManifestPhred33V2
2. qiime dada2 denoise-single –p-max-ee 14 —p-trunc-len 0 —i-demultiplexed-sequences —output-dir dada
3. qiime feature-classifier classify-sklearn —i-classifier silva-138-99-classifier.qza —i-reads dada/representative\_sequences.qza —p-read-orientation reverse —o-classification dada/taxonomy.qza

### Loading R libraries

#libraries  
library(mia)  
library(miaViz)  
library(tidyverse)  
library(knitr)  
library(kableExtra)  
library(dplyr)  
library(reshape2)  
library(ggplot2)  
library(ggthemes)

### Zymo Research mock community samples

Importing data from Qiime2 to a TreeSummarizedExperiment object. Dimension results describe size of each datatable (number of variants and samples).

#dada2\_results  
zymo\_mock\_dada <- loadFromQIIME2(  
 featureTableFile= "Qiime2\_files/mock\_community/mock\_dada\_asv\_table.qza",  
 taxonomyTableFile= "Qiime2\_files/mock\_community/mock\_dada\_taxa.qza",  
 sampleMetaFile= "Qiime2\_files/mock\_community/meta\_dada.tsv")  
#clustering\_results  
zymo\_mock\_vsearch <- loadFromQIIME2(  
 featureTableFile = "Qiime2\_files/mock\_community/mock\_vsearch\_otu\_table.qza",  
 taxonomyTableFile = "Qiime2\_files/mock\_community/mock\_vsearch\_taxa.qza",  
 sampleMetaFile = "Qiime2\_files/mock\_community/meta\_clust.tsv")  
#tse\_dimensions  
dim(zymo\_mock\_dada)

[1] 17 4

dim(zymo\_mock\_vsearch)

[1] 57 4

Creating a dataframe containing true composition of the mock sample

#create alphabetically sorted dataframe containing theoretical composition  
rowname <- c("Bacillus", "Enterococcus", "Escherichia-Shigella",  
 "Lactobacillus", "Listeria", "Pseudomonas", "Salmonella",  
 "Staphyloccus")  
abundance <- c(0.174, 0.10, 0.101, 0.184, 0.141, 0.042, 0.104, 0.155)  
mock <- data.frame(rowname, abundance)

Creating dataframe from dada2 results

#agglomerate to genus level and calculate relabundance  
zymo\_mock\_dada <- agglomerateByRank(zymo\_mock\_dada, rank="Genus",  
 onRankOnly=TRUE)  
zymo\_mock\_dada <- transformAssay(zymo\_mock\_dada, assay.type="counts",  
 method="relabundance")  
#create dataframe and sorting alphabetically  
zymo\_df\_dada <- data.frame(assays(zymo\_mock\_dada)$relabundance)  
zymo\_df\_dada <- zymo\_df\_dada[order(row.names(zymo\_df\_dada)),]  
rownames(zymo\_df\_dada) <- rowname   
zymo\_df\_dada <- tibble::rownames\_to\_column(zymo\_df\_dada)  
kable(zymo\_df\_dada, digits=2)

| rowname | Mock01 | Mock02 | Mock03 | Mock04 |
| --- | --- | --- | --- | --- |
| Bacillus | 0.23 | 0.23 | 0.28 | 0.27 |
| Enterococcus | 0.00 | 0.00 | 0.00 | 0.06 |
| Escherichia-Shigella | 0.00 | 0.06 | 0.00 | 0.01 |
| Lactobacillus | 0.19 | 0.17 | 0.19 | 0.17 |
| Listeria | 0.12 | 0.12 | 0.15 | 0.16 |
| Pseudomonas | 0.00 | 0.02 | 0.00 | 0.00 |
| Salmonella | 0.03 | 0.02 | 0.00 | 0.00 |
| Staphyloccus | 0.43 | 0.38 | 0.38 | 0.33 |

Creating dataframe from vsearch results

#agglomerate to genus level and calculate relabundance  
zymo\_mock\_vsearch <- agglomerateByRank(zymo\_mock\_vsearch, rank="Genus",  
 onRankOnly=TRUE)  
zymo\_mock\_vsearch <- transformAssay(zymo\_mock\_vsearch, assay.type="counts",  
 method="relabundance")  
#create dataframe and sort alphabetically  
zymo\_df\_vsearch <- data.frame(assays(zymo\_mock\_vsearch)$relabundance)  
zymo\_df\_vsearch <- zymo\_df\_vsearch[1:8,]  
zymo\_df\_vsearch <- zymo\_df\_vsearch[order(row.names(zymo\_df\_vsearch)),]  
rownames(zymo\_df\_vsearch) <- rowname  
zymo\_df\_vsearch <- tibble::rownames\_to\_column(zymo\_df\_vsearch)  
#table  
kable(zymo\_df\_vsearch, digits=2)

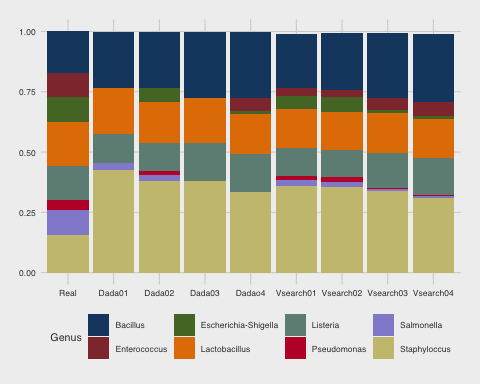
| rowname | Mock01 | Mock02 | Mock03 | Mock04 |
| --- | --- | --- | --- | --- |
| Bacillus | 0.23 | 0.24 | 0.27 | 0.28 |
| Enterococcus | 0.03 | 0.03 | 0.05 | 0.06 |
| Escherichia-Shigella | 0.05 | 0.06 | 0.01 | 0.01 |
| Lactobacillus | 0.16 | 0.16 | 0.17 | 0.16 |
| Listeria | 0.11 | 0.11 | 0.14 | 0.15 |
| Pseudomonas | 0.02 | 0.02 | 0.01 | 0.01 |
| Salmonella | 0.02 | 0.02 | 0.01 | 0.01 |
| Staphyloccus | 0.36 | 0.35 | 0.34 | 0.31 |

Merging all mock community dataframes

#merge dataframes and rename columns  
summary\_df <- left\_join(mock, zymo\_df\_dada, by="rowname")  
summary\_df <- left\_join(summary\_df, zymo\_df\_vsearch, by="rowname")  
colnames(summary\_df) <- c("Genus", "Real", "Dada01", "Dada02", "Dada03",  
 "Dadao4", "Vsearch01", "Vsearch02", "Vsearch03",  
 "Vsearch04")

Plotting mock community results

#melt summary  
molten\_summary <- melt(summary\_df, id.vars = 1)  
#plot object  
plot <- ggplot(molten\_summary, aes(x=variable,y=value, fill=Genus)) +   
 geom\_bar(position="stack", stat = "identity") +  
 theme\_fivethirtyeight(base\_size=8) + scale\_fill\_stata()  
plot



In this set, differences between dada and vsearch are minor. However, dada does miss taxa in some samples.

When compared to real composition, some bacteria are underpresented (*Salmonella, E. coli*) and some overpresented (*Bacillus, Streptococcus*).

Note also, that results compared to original article using Silva reference data are widely different suggesting there is some problem using Rescue and Silva.

### Zymo Gut Mock Community Samples

Importing data from Qiime2 to a TreeSummarizedExperiment object. Dimension results describe size of each datatable (number of variants and samples).

#dada2\_results  
gut\_dada <- loadFromQIIME2(  
 featureTableFile= "Qiime2\_files/gut\_mock\_community/gut\_dada\_asv\_table.qza",  
 taxonomyTableFile = "Qiime2\_files/gut\_mock\_community/gut\_dada\_taxa.qza",  
 sampleMetaFile = "Qiime2\_files/gut\_mock\_community/gut\_dada.tsv")  
#clustering\_results  
gut\_vsearch <- loadFromQIIME2(  
 featureTableFile = "Qiime2\_files/gut\_mock\_community/gut\_vsearch\_otu\_table.qza",  
 taxonomyTableFile = "Qiime2\_files/gut\_mock\_community/gut\_vsearch\_taxa.qza",  
 sampleMetaFile = "Qiime2\_files/gut\_mock\_community/gut\_clust.tsv")  
dim(gut\_dada)

[1] 14 4

dim(gut\_vsearch)

[1] 174 4

Creating dataframe containing true gut community composition

#create vectors and alphabetically sorted df containing theoretical composition  
Genus <- c("Akkermansia", "Bacteroides", "Bifidobacterium", "Clostridioides",  
 "Clostridium", "E. coli", "Enterococcus", "Faecalibacterium",  
 "Fusobacterium", "Lactobacillus", "Methanobrevibacter", "Prevotella",  
 "Roseburia", "Salmonella", "Veillonella")  
gut\_abundance <- c(0.01, 0.099, 0.089, 0.026, 0.000002, 0.121, 0.00001, 0.176,  
 0.075, 0.096, 0.001, 0.05, 0.099, 0.0001, 0.159)  
gut\_community <- data.frame(Genus, gut\_abundance)

Creating dataframe from dada results

#agglomerate to genus level and calculate relabundance  
gut\_dada <- agglomerateByRank(gut\_dada, rank="Genus",  
 onRankOnly=TRUE)  
gut\_dada <- transformAssay(gut\_dada, assay.type="counts",  
 method="relabundance")  
#create dataframe and sort alphbetically  
gdada <- data.frame(assays(gut\_dada)$relabundance)  
gdada <- gdada[order(row.names(gdada)),]  
gdada <- tibble::rownames\_to\_column(gdada)  
#modify genus names to match kit data  
gdada$rowname <- c("Bacteroides","Clostridioides","E. coli",  
 "Faecalibacterium","Fusobacterium", "Prevotella",  
 "Veillonella")  
colnames(gdada) <- c("Genus", "Gut01", "Gut02", "Gut03", "Gut04")  
#make table  
kable(gdada, digits=2)

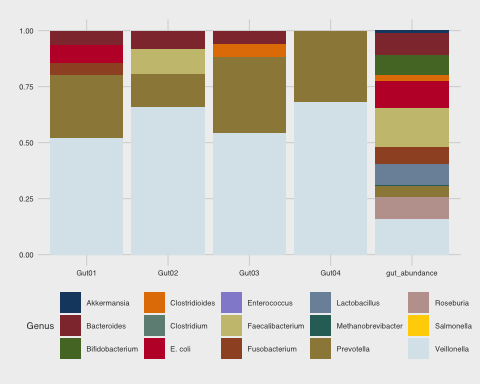
| Genus | Gut01 | Gut02 | Gut03 | Gut04 |
| --- | --- | --- | --- | --- |
| Bacteroides | 0.06 | 0.08 | 0.06 | 0.00 |
| Clostridioides | 0.00 | 0.00 | 0.06 | 0.00 |
| E. coli | 0.08 | 0.00 | 0.00 | 0.00 |
| Faecalibacterium | 0.00 | 0.11 | 0.00 | 0.00 |
| Fusobacterium | 0.05 | 0.00 | 0.00 | 0.00 |
| Prevotella | 0.28 | 0.15 | 0.34 | 0.32 |
| Veillonella | 0.52 | 0.66 | 0.54 | 0.68 |

Melting dataframe and creating plot object

#merge data and replace NA values with 0  
dada\_kit <- merge(gdada, gut\_community, all = TRUE)  
dada\_kit[is.na(dada\_kit)] <- 0  
#melt and create barplot object  
molten\_gutdada <- melt(dada\_kit, id.vars = 1)  
plot\_dgut <- ggplot(molten\_gutdada, aes(x=variable,y=value, fill=Genus)) +   
 geom\_bar(position="stack", stat = "identity") +  
 theme\_fivethirtyeight(base\_size=7) + scale\_fill\_stata()

Plotting dada results

plot\_dgut



In more complex mock community, denoising performance gets clearly worse. *Veillonella* and *Prevotella* are overpresented.

Creating dataframe from vsearch results. Some taxa is filtered in order to simplify barplot.

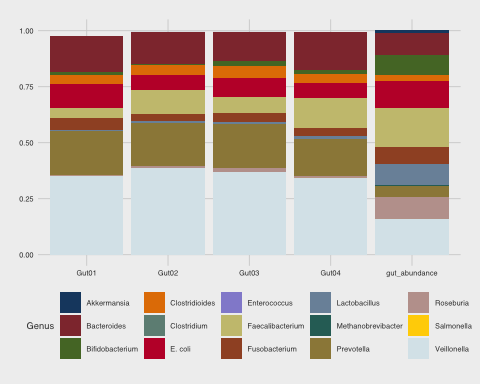
#agglomerate to genus level and calculate relabundance  
gut\_vsearch <- agglomerateByRank(gut\_vsearch, rank="Genus",  
 onRankOnly=TRUE)  
gut\_vsearch <- transformAssay(gut\_vsearch, assay.type="counts",  
 method="relabundance")  
#create dataframe and sort alphabetically  
g\_vsearch <- data.frame(assays(gut\_vsearch)$relabundance)  
g\_vsearch <- g\_vsearch[order(row.names(g\_vsearch)),]  
g\_vsearch <- tibble::rownames\_to\_column(g\_vsearch)  
#modify genus names to match kit  
vsearch\_vector <- g\_vsearch$rowname  
modified\_vector <- sub("^g\_\_", "", vsearch\_vector)  
g\_vsearch$rowname <- modified\_vector  
colnames(g\_vsearch) <- c("Genus", "Gut01", "Gut02", "Gut03", "Gut04")  
g\_vsearch[11,1] <- "E. coli"  
#remove "zero" results  
g\_vsearch <- g\_vsearch[-c(2,3,6:8,10,14:15,17,20,22),]  
#print table  
kable(g\_vsearch, digits=2)

|  | Genus | Gut01 | Gut02 | Gut03 | Gut04 |
| --- | --- | --- | --- | --- | --- |
| 1 | Akkermansia | 0.00 | 0.00 | 0.00 | 0.00 |
| 4 | Bacteroides | 0.16 | 0.14 | 0.13 | 0.17 |
| 5 | Bifidobacterium | 0.01 | 0.01 | 0.02 | 0.02 |
| 9 | Clostridioides | 0.04 | 0.04 | 0.06 | 0.04 |
| 11 | E. coli | 0.11 | 0.07 | 0.08 | 0.06 |
| 12 | Faecalibacterium | 0.05 | 0.11 | 0.07 | 0.13 |
| 13 | Fusobacterium | 0.05 | 0.03 | 0.04 | 0.04 |
| 16 | Lactobacillus | 0.00 | 0.01 | 0.01 | 0.01 |
| 18 | Prevotella | 0.20 | 0.19 | 0.20 | 0.17 |
| 19 | Roseburia | 0.01 | 0.01 | 0.02 | 0.01 |
| 21 | Salmonella | 0.00 | 0.00 | 0.00 | 0.00 |
| 23 | Veillonella | 0.35 | 0.39 | 0.37 | 0.34 |

#merge and replace NA values with 0  
vsearch\_kit <- merge(g\_vsearch, gut\_community, all = TRUE)  
vsearch\_kit[is.na(vsearch\_kit)] <- 0  
#melt dataframe  
molten\_gutvsearch <- melt(vsearch\_kit, id.vars = 1)  
#create plot object  
plot\_vgut <- ggplot(molten\_gutvsearch, aes(x=variable,  
 y=value, fill=Genus)) + geom\_bar(position="stack",  
 stat = "identity") + theme\_fivethirtyeight(base\_size=7) +  
 scale\_fill\_stata()

Plotting vsearch results

plot\_vgut



Using otu clustering by vsearch, results from gut mock community are much better. *Enterococcus, Clostridium, Methanobrevibacter and Salmonella* are present in very low proportion in the community and number of sequences is probably too low to recognize them reliably.

### Oral microbiome samples

Importing Qiime 2 files

#dada\_results  
saliva\_dada <- loadFromQIIME2(  
 featureTableFile = "Qiime2\_files/oral\_microbiome/oral\_dada\_asv\_table.qza",  
 taxonomyTableFile = "Qiime2\_files/oral\_microbiome/oral\_dada\_taxa.qza",  
 sampleMetaFile = "Qiime2\_files/oral\_microbiome/oral\_dada.tsv")  
#clustering\_results  
saliva\_vsearch <- loadFromQIIME2(  
 featureTableFile = "Qiime2\_files/oral\_microbiome/oral\_vsearch\_otu\_table.qza",  
 taxonomyTableFile = "Qiime2\_files/oral\_microbiome/oral\_vsearch\_taxa.qza",  
 sampleMetaFile = "Qiime2\_files/oral\_microbiome/oral\_clust.tsv")  
#calculate data dimentsions  
dim(saliva\_dada)

[1] 161 20

dim(saliva\_vsearch)

[1] 2838 20

Dada has only 161 variants compared to 2838 vsearch otus

Creating dataframe from dada results

#agglomerate to genus and calculate relabundance  
saliva\_dada <- agglomerateByRank(saliva\_dada, rank="Genus",  
 onRankOnly=TRUE)  
saliva\_dada <- transformAssay(saliva\_dada, assay.type="counts",  
 method="relabundance")  
#create dataframe and sort alphabetically  
saliva\_dada\_df <- data.frame(assays(saliva\_dada)$relabundance)  
saliva\_dada\_df <- saliva\_dada\_df[order(row.names(saliva\_dada\_df)),]  
saliva\_dada\_df <- tibble::rownames\_to\_column(saliva\_dada\_df)

Creating dataframe from vsearch results

#agglomerate to genus and calculate relabundance  
saliva\_vsearch <- agglomerateByRank(saliva\_vsearch, rank="Genus", onRankOnly=TRUE)  
saliva\_vsearch <- transformAssay(saliva\_vsearch, assay.type="counts",  
 method="relabundance")  
#create dataframe and sort alphabetically  
saliva\_vsearch\_df <- data.frame(assays(saliva\_vsearch)$relabundance)  
saliva\_vsearch\_df <- saliva\_vsearch\_df[order(row.names(saliva\_vsearch\_df)),]  
saliva\_vsearch\_df <- tibble::rownames\_to\_column(saliva\_vsearch\_df)

Create top10 dataframe for dada taxonomy, example abundance table and plot object

#get top10 dada taxa  
top10\_saliva\_dada <- getTopTaxa(saliva\_dada, top=10, method="sum",  
 assay\_name="relabundance")  
#filter dataframe containing only top10  
saliva\_dada\_df <- saliva\_dada\_df %>% filter(rowname %in% top10\_saliva\_dada)  
molten\_saliva\_dada\_df <- melt(saliva\_dada\_df, id.vars=1)  
#abundance table as an example  
kable(saliva\_dada\_df[,1:6],digits=2)

| rowname | saliva01 | saliva02 | saliva03 | saliva04 | saliva05 |
| --- | --- | --- | --- | --- | --- |
| g\_\_Bergeyella | 0.00 | 0 | 0 | 0.1 | 0.02 |
| g\_\_Campylobacter | 0.00 | 0 | 0 | 0.0 | 0.00 |
| g\_\_Fusobacterium | 0.00 | 0 | 0 | 0.0 | 0.06 |
| g\_\_Granulicatella | 0.00 | 0 | 0 | 0.0 | 0.04 |
| g\_\_Neisseria | 0.00 | 0 | 0 | 0.0 | 0.00 |
| g\_\_Prevotella | 0.28 | 1 | 1 | 0.0 | 0.23 |
| g\_\_Serratia | 0.00 | 0 | 0 | 0.0 | 0.00 |
| g\_\_Solobacterium | 0.00 | 0 | 0 | 0.0 | 0.03 |
| g\_\_Streptococcus | 0.62 | 0 | 0 | 0.9 | 0.45 |
| g\_\_Veillonella | 0.04 | 0 | 0 | 0.0 | 0.09 |

#create plot object  
plot\_saliva\_dada <- ggplot(molten\_saliva\_dada\_df, aes(x=variable,y=value,  
 fill=rowname)) + geom\_bar(position="stack",  
 stat = "identity") +  
 theme\_fivethirtyeight(base\_size=7) +  
 scale\_fill\_stata()

Create top10 dataframe for vsearch taxonomy, example abundance table and plot object

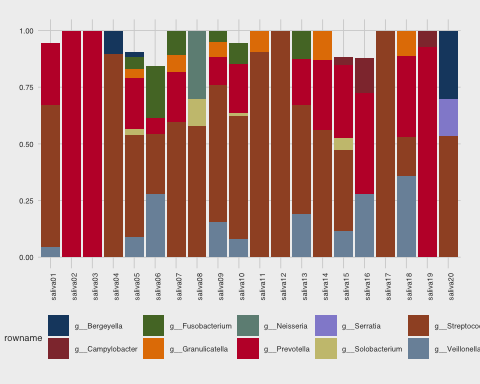
#get top10 dada taxa  
top10\_saliva\_vsearch <- getTopTaxa(saliva\_vsearch, top=10, method="sum",  
 assay\_name="relabundance")  
#filter dataframe containing only top10  
saliva\_vsearch\_df <- saliva\_vsearch\_df %>% filter(rowname %in% top10\_saliva\_vsearch)  
molten\_saliva\_vsearch\_df <- melt(saliva\_vsearch\_df, id.vars=1)  
#abundance table as an example  
kable(saliva\_vsearch\_df[,1:6],digits=2)

| rowname | saliva01 | saliva02 | saliva03 | saliva04 | saliva05 |
| --- | --- | --- | --- | --- | --- |
| g\_\_Alloprevotella | 0.07 | 0.01 | 0.03 | 0.07 | 0.01 |
| g\_\_Bergeyella | 0.01 | 0.00 | 0.01 | 0.10 | 0.01 |
| g\_\_Fusobacterium | 0.06 | 0.05 | 0.08 | 0.06 | 0.06 |
| g\_\_Granulicatella | 0.03 | 0.03 | 0.01 | 0.07 | 0.04 |
| g\_\_Haemophilus | 0.00 | 0.01 | 0.02 | 0.03 | 0.01 |
| g\_\_Oribacterium | 0.01 | 0.05 | 0.04 | 0.01 | 0.02 |
| g\_\_Prevotella | 0.21 | 0.21 | 0.41 | 0.09 | 0.24 |
| g\_\_Rothia | 0.01 | 0.04 | 0.01 | 0.01 | 0.03 |
| g\_\_Streptococcus | 0.38 | 0.37 | 0.09 | 0.36 | 0.33 |
| g\_\_Veillonella | 0.06 | 0.08 | 0.07 | 0.06 | 0.09 |

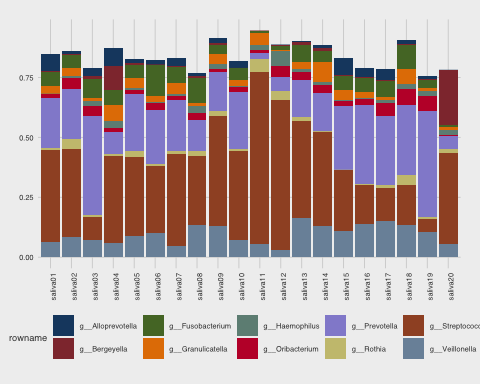
#create plot object  
plot\_saliva\_vsearch <- ggplot(molten\_saliva\_vsearch\_df, aes(x=variable,y=value,  
 fill=rowname)) + geom\_bar(position="stack",  
 stat = "identity") +  
 theme\_fivethirtyeight(base\_size=7) +  
 scale\_fill\_stata()

Plotting both sets

plot\_saliva\_dada + theme(axis.text.x = element\_text(angle=90))



plot\_saliva\_vsearch + theme(axis.text.x = element\_text(angle=90))



Results from oral microbiome samples suggest that using dada for nanopore reads does result in loss of diversity even for most abundant bacterial taxa.

*Note. Analysed dataset is fairly small, only ~110 000 reads in total.*