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) %>% kable_styling()
```

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) %>% kable_styling()
```

rowname	Mock01	Mock02	Mock03	Mock04

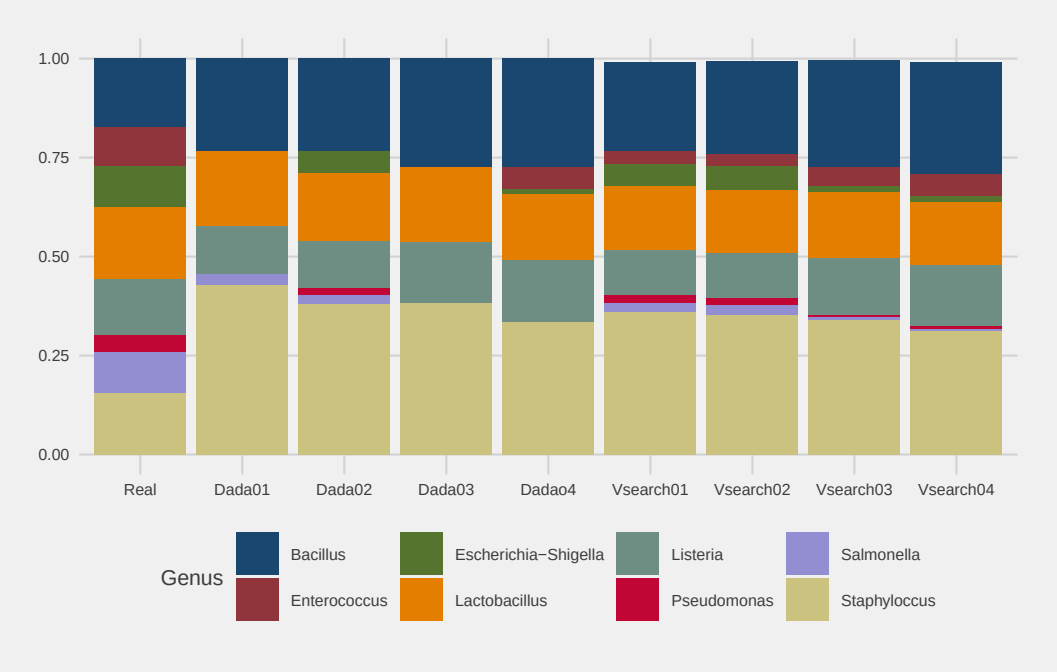
Bacillus	0.23	0.24	0.27	0.28
Enterococcus	0.03	0.03	0.05	0.06
Escherichia-Shigella	0.05	0.06	0.01	0.01
Lactobacillus	0.16	0.16	0.17	0.16
Listeria	0.11	0.11	0.14	0.15
Pseudomonas	0.02	0.02	0.01	0.01
Salmonella	0.02	0.02	0.01	0.01
Staphyloccus	0.36	0.35	0.34	0.31

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) %>% kable_styling()
```

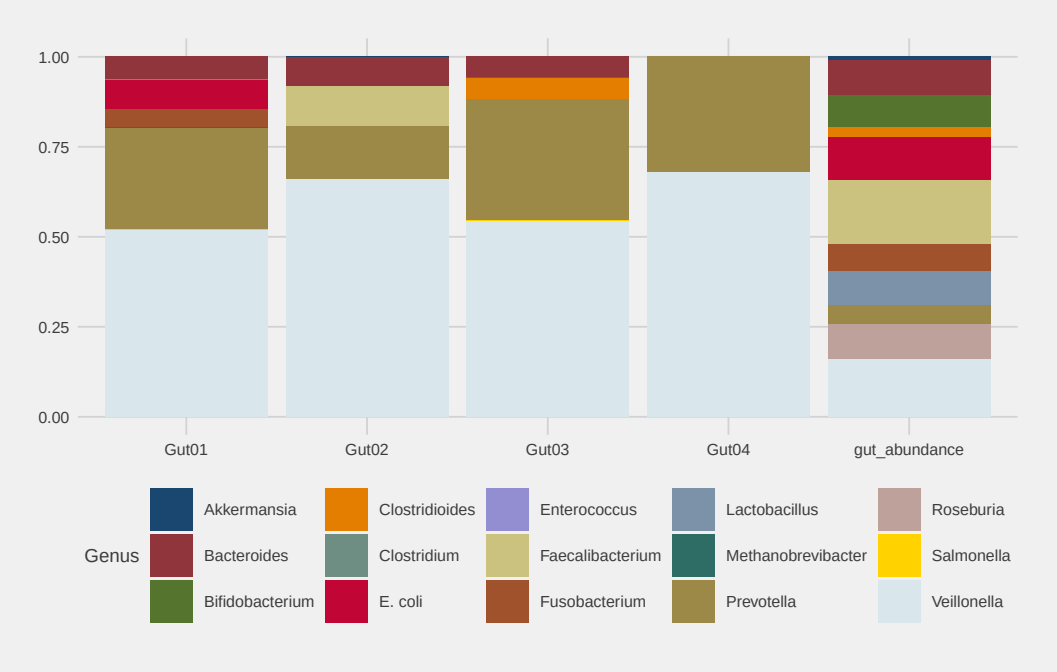
Genus	Gut01	Gut02	Gut03	Gut04
Bacteroides	0.06	0.08	0.06	0.00
Clostridioides	0.00	0.00	0.06	0.00
E. coli	0.08	0.00	0.00	0.00
Faecalibacterium	0.00	0.11	0.00	0.00
Fusobacterium	0.05	0.00	0.00	0.00
Prevotella	0.28	0.15	0.34	0.32
Veillonella	0.52	0.66	0.54	0.68

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),]
```

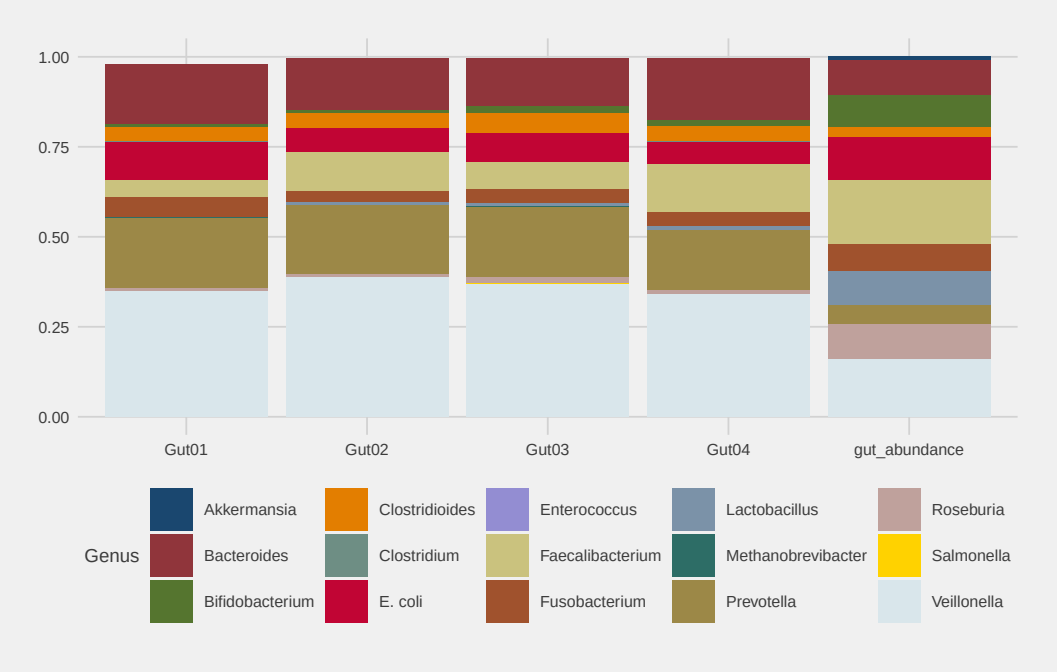
```
#print table
kable(g_vsearch, digits=2) %>% kable_styling()
```

	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")
#copying unmodified object
tse_saliva <- saliva_vsearch
#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) %>% kable_styling()
```

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) %>% kable_styling()
```

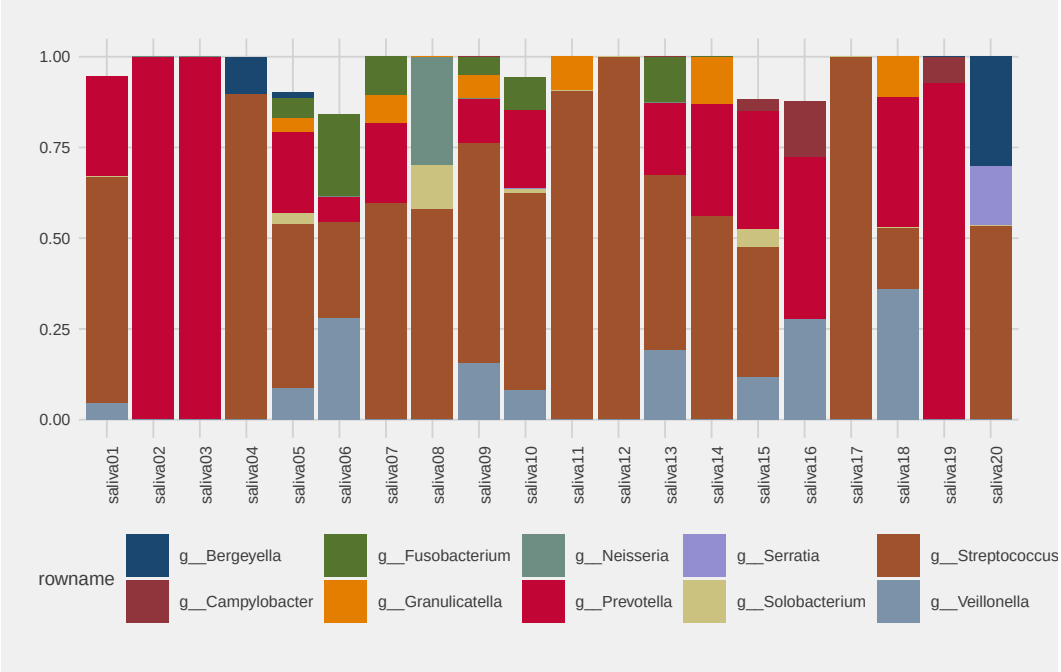
rowname	saliva01	saliva02	saliva03	saliva04	saliva05
g___Alloprevotella	0.07	0.01	0.03	0.07	0.01
g___Bergeyella	0.01	0.00	0.01	0.10	0.01
g___Fusobacterium	0.06	0.05	0.08	0.06	0.06
g___Granulicatella	0.03	0.03	0.01	0.07	0.04

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

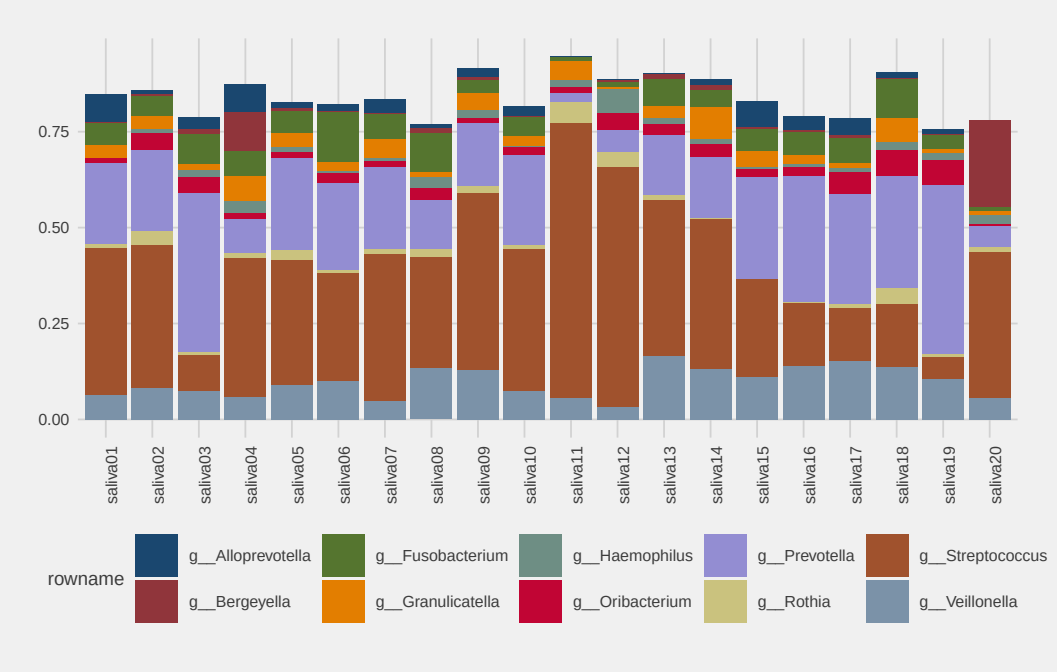
```
#create plot object
plot_saliva_vsearch <- ggplot(molten_saliva_vsearch_df, aes(x=variable,y=value,
                              fill=rowname)) + geom_bar(position="stack",
                              stat = "identity") +
                              theme_fivethirtyeight(base_size=7) +
                              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.

Taxonomic resolution of full length 16S rRNA sequencing

```
#tse object dimensions
total <- dim(tse_saliva)
#calculate total number of empty results at Genus level
l6 <-  1- (sum(taxonomyRankEmpty(tse_saliva, rank="Genus"))/total[1])
#percent of recognized
#calculate total number of empty results at Species level
l7 <- 1- (sum(taxonomyRankEmpty(tse_saliva, rank="Species"))/total[1])
#calculate total number of empty results at Phylum level
l2 <- 1 - (sum(taxonomyRankEmpty(tse_saliva, rank="Phylum"))/total[1])
#table
tax_level <- c("Phylum", "Genus", "Species")
recognised <- c(l2,l6,l7)
kable(data.frame(tax_level,recognised), digits=3) %>% kable_styling()
```

tax_level	recognised
Phylum	0.805
Genus	0.802
Species	0.476

80 % of otus are recognized at genus level and 47,6 % at species level. It also seems that 20 % of results are either non-bacterial or unknown origin.