MB668_Project

Mary English

Spring 2020

Loading libraries

```r
library(readr)
library(ggplot2)
```

Part I. Metagenomic sequence filtering
trim_galore

- Manual: https://github.com/FelixKrueger/TrimGalore/blob/master/Docs/Trim_Galore_User_Guide.md
- Wiki: http://www.metagenomics.wiki/tools/short-read/quality-control/trim-galore

Done on my personal computer.

```
conda install -c bioconda trim-galore
mv lane1-s007-indexN705-S508-GGACTCCT-CTAAGCCT-1B-Meta_S7_L001_R1_001.fastq 1b_meta_r1.fastq
mv lane1-s007-indexN705-S508-GGACTCCT-CTAAGCCT-1B-Meta_S7_L001_R2_001.fastq 1b_meta_r2.fastq
trim_galore --paired 1b_meta_r1.fastq 1b_meta_r2.fastq
```

AUTO-DETECTING ADAPTER TYPE

Attempting to auto-detect adapter type from the first 1 million sequences of the first file (>> 1b_meta_r1.fastq <<)

Found perfect matches for the following adapter sequences:

- Adapter type, Count, Sequence, Sequences analysed, Percentage
- Nextera 375246, CTGTCTCTTATA, 1000000, 37.52
- Illumina, 2, AGATCGGAAGAGC, 1000000, 0.00
- smallRNA, 2, TGGAATTCTCGG, 1000000, 0.00

Using Nextera adapter for trimming (count: 375246). Second best hit was Illumina (count: 2)

Writing report to '1b_meta_r1.fastq_trimming_report.txt'

SUMMARISING RUN PARAMETERS

- Input filename: 1b_meta_r1.fastq
- Trimming mode: paired-end
- Trim Galore version: 0.6.4_dev
- Cutadapt version: 1.18
- Number of cores used for trimming: 1
- Quality Phred score cutoff: 20
- Quality encoding type selected: ASCII+33
- Adapter sequence: 'CTGTCTCTTATA' (Nextera Transposase sequence; auto-detected)
- Maximum trimming error rate: 0.1 (default)
- Minimum required adapter overlap (stringency): 1 bp
- Minimum required sequence length for both reads before a sequence pair gets removed: 20 bp

Cutadapt seems to be reasonably up-to-date. Setting -j 1
Writing final adapter and quality trimmed output to 1b_meta_r1_trimmed.fq

Now performing quality (cutoff '-q 20') and adapter trimming in a single pass for the adapter sequence: 'CTGTCTCTTATA' from file 1b_meta_r1.fastq

- 10000000 sequences processed
- This is cutadapt 1.18 with Python 3.7.4
- Command line parameters: -j 1 -e 0.1 -q 20 -O 1 -a CTGTCTCTTATA 1b_meta_r1.fastq
  Processing reads on 1 core in single-end mode …
  Finished in 137.66 s (13 us/read; 4.61 M reads/minute).

Summary

- Total reads processed: 10,579,668
- Reads with adapters: 5,885,278 (55.6%)
- Reads written (passing filters): 10,579,668 (100.0%)

- Total basepairs processed: 2,655,496,668 bp
- Quality-trimmed: 134,695,996 bp (5.1%)
- Total written (filtered): 2,174,547,432 bp (81.9%)

Adapter 1

Sequence: CTGTCTCTTATA; Type: regular 3'; Length: 12; Trimmed: 5885278 times.

No. of allowed errors:

- 0-9 bp: 0; 10-12 bp: 1

Bases preceding removed adapters:

- A: 19.3%
- C: 31.4%
- G: 20.2%
- T: 29.1%
- none/other: 0.0%

RUN STATISTICS FOR INPUT FILE: 1b_meta_r1.fastq

10579668 sequences processed in total

SUMMARISING RUN PARAMETERS

- Input filename: 1b_meta_r2.fastq
- Trimming mode: paired-end
- Trim Galore version: 0.6.4_dev
- Cutadapt version: 1.18
- Number of cores used for trimming: 1
- Quality Phred score cutoff: 20
- Quality encoding type selected: ASCII+33
- Adapter sequence: 'CTGTCTCTTATA' (Nextera Transposase sequence; auto-detected)
- Maximum trimming error rate: 0.1 (default)
- Minimum required adapter overlap (stringency): 1 bp
- Minimum required sequence length for both reads before a sequence pair gets removed: 20 bp

Cutadapt seems to be reasonably up-to-date. Setting -j -j 1
Writing final adapter and quality trimmed output to 1b_meta_r2_trimmed.fq

Now performing quality (cutoff '-q 20') and adapter trimming in a single pass for the adapter sequence: 'CTGTCTCTTATA' from file 1b_meta_r2.fastq

- 10000000 sequences processed
- This is cutadapt 1.18 with Python 3.7.4
- Command line parameters: -j 1 -e 0.1 -q 20 -O 1 -a CTGTCTCTTATA 1b_meta_r2.fastq
  Processing reads on 1 core in single-end mode …
  Finished in 139.84 s (13 us/read; 4.54 M reads/minute).

Summary

- Total reads processed: 10,579,668
- Reads with adapters: 5,890,390 (55.7%)
- Reads written (passing filters): 10,579,668 (100.0%)

- Total basepairs processed: 2,655,496,668 bp
- Quality-trimmed: 200,507,982 bp (7.6%)
- Total written (filtered): 2,159,836,467 bp (81.3%)

Adapter 1

Sequence: CTGTCTCTTATA; Type: regular 3'; Length: 12; Trimmed: 5890390 times.

No. of allowed errors:

- 0-9 bp: 0; 10-12 bp: 1

Bases preceding removed adapters:

- A: 19.1%
- C: 31.7%
- G: 20.4%
- T: 28.8%
- none/other: 0.0%

RUN STATISTICS FOR INPUT FILE: 1b_meta_r2.fastq

10579668 sequences processed in total

Total number of sequences analysed: 10579668

Number of sequence pairs removed because at least one read was shorter than the length cutoff (20 bp): 5239 (0.05%)

Part II. Assembly and annotation

Megahit

```
megahit -1 1b_meta_r1_val_1.fq -2 1b_meta_r2_val_2.fq -o megahit_assembly --min-count 2 --k-list 21,41,61,81 --me
rge-level 20,0.95 --prune-depth 2 --min-contig-len 500
get_assembly_stats.py final.contigs.fa
```

Assembly stats: Scaffolds > 500nt

- Number 293963
- Total Length 261746631
- N50 880
- Avg 890.4067212540353
- Median 708.0150
- Max 108542
- Min 500

Kaiju

```
/dfs/MICRO/Mueller_Lab/mk/kaiju/bin/kaiju-makedb -s refseq

# Kaiju on READS - microbe
kaiju -t /dfs/MICRO/Mueller_Lab/mk/kaiju/nodes.dmp -f /dfs/MICRO/Mueller_Lab/mk/kaiju/refseq/kaiju_db_refseq.fmi
 -i /dfs/MICRO/Mueller_Lab/mk/1b_meta_r1_val_1.fq -j /dfs/MICRO/Mueller_Lab/mk/1b_meta_r2_val_2.fq -o /dfs/MICRO/
Mueller_Lab/mk/kaiju/kaiju_reads.out

bin/kaiju2table -t nodes.dmp -n names.dmp -r genus -o kaiju_reads_summary.tsv kaiju_reads.out

bin/kaiju-addTaxonNames -t /dfs/MICRO/Mueller_Lab/mk/kaiju/nodes.dmp -n /dfs/MICRO/Mueller_Lab/mk/kaiju/names.dmp
-i /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_reads/microbe/kaiju_reads.out -o /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_r
eads.names.out

# Kaiju on READS - oyster
bin/kaiju-mkbwt -a DNA -o kaiju_oyster c_gigas_genome.fasta

bin/kaiju-mkfmi kaiju_oyster

bin/kaiju -t /dfs/MICRO/Mueller_Lab/mk/kaiju/nodes.dmp -f /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_oyster.fmi -i /df
s/MICRO/Mueller_Lab/mk/1b_meta_r1_val_1.fq -j /dfs/MICRO/Mueller_Lab/mk/1b_meta_r2_val_2.fq -o /dfs/MICRO/Mueller
_Lab/mk/kaiju/kaiju_reads/oyster/kaiju_oyster_reads.out

bin/kaiju2table -t nodes.dmp -n names.dmp -r genus -o kaiju_oyster_reads_summary.tsv kaiju_reads/oyster/kaiju_oys
ter_reads.out

bin/kaiju-addTaxonNames -t /dfs/MICRO/Mueller_Lab/mk/kaiju/nodes.dmp -n /dfs/MICRO/Mueller_Lab/mk/kaiju/names.dmp
-i /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_reads/oyster/kaiju_oyster_reads.out -o /dfs/MICRO/Mueller_Lab/mk/kaiju/k
aiju_reads/oyster/kaiju_oyster_reads.names.out
# Kaiju on MEGAHIT ASSEMBLY - oyster
bin/kaiju -t /dfs/MICRO/Mueller_Lab/mk/kaiju/nodes.dmp -f /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_oyster.fmi -i /df
s/MICRO/Mueller_Lab/mk/megahit_assembly/final.contigs.fa -o /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_oyster_assembl
y.out

bin/kaiju2table -t nodes.dmp -n names.dmp -r genus -o kaiju_assembly_summary.tsv kaiju_oyster_assembly.out

bin/kaiju-addTaxonNames -t /dfs/MICRO/Mueller_Lab/mk/kaiju/nodes.dmp -n /dfs/MICRO/Mueller_Lab/mk/kaiju/names.dmp
-i /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_assembly/oyster/kaiju_oyster_assembly.out -o /dfs/MICRO/Mueller_Lab/mk/k
aiju/kaiju_assembly/oyster/kaiju_oyster_assembly.names.out

# Kaiju on MEGAHIT ASSEMBLY - microbes
kaiju -t /dfs/MICRO/Mueller_Lab/mk/kaiju/nodes.dmp -f /dfs/MICRO/Mueller_Lab/mk/kaiju/refseq/kaiju_db_refseq.fmi
 -i /dfs/MICRO/Mueller_Lab/mk/megahit_assembly/final.contigs.fa -o /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_assembl
y/microbe/kaiju_assembly.out

bin/kaiju2table -t nodes.dmp -n names.dmp -r genus -o kaiju_assembly_summary.tsv kaiju_assembly.out

bin/kaiju-addTaxonNames -t /dfs/MICRO/Mueller_Lab/mk/kaiju/nodes.dmp -n /dfs/MICRO/Mueller_Lab/mk/kaiju/names.dmp
-i /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_assembly/microbe/kaiju_assembly.out -o /dfs/MICRO/Mueller_Lab/mk/kaiju/k
aiju_assembly/microbe/kaiju_assembly.names.out
```

Krona

```
# microbe, reads
bin/kaiju2krona -t nodes.dmp -n names.dmp -i /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_reads/microbe/kaiju_reads.out
 -o /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_reads/microbe/kaiju_reads.out.krona

bin/ktImportText -o /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_reads/microbe/kaiju_reads.html /dfs/MICRO/Mueller_Lab/m
k/kaiju/kaiju_reads/microbe/kaiju_reads.out.krona

# oyster, reads
bin/kaiju2krona -t nodes.dmp -n names.dmp -i /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_reads/oyster/kaiju_oyster_read
s.out -o /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_reads/oyster/kaiju_oyster_reads.out.krona

bin/ktImportText -o /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_reads/oyster/kaiju_oyster_reads.html /dfs/MICRO/Mueller
_Lab/mk/kaiju/kaiju_reads/oyster/kaiju_oyster_reads.out.krona

# microbe, assembly
bin/kaiju2krona -t nodes.dmp -n names.dmp -i /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_assembly/microbe/kaiju_assembl
y.out -o /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_assembly/microbe/kaiju_assembly.out.krona

bin/ktImportText -o /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_assembly/microbe/kaiju_assembly.html /dfs/MICRO/Mueller
_Lab/mk/kaiju/kaiju_assembly/microbe/kaiju_assembly.out.krona

# oyster, assembly
bin/kaiju2krona -t nodes.dmp -n names.dmp -i /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_assembly/oyster/kaiju_oyster_a
ssembly.out -o /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_assembly/oyster/kaiju_oyster_assembly.out.krona

bin/ktImportText -o /dfs/MICRO/Mueller_Lab/mk/kaiju/kaiju_assembly/oyster/kaiju_oyster_assembly.html /dfs/MICRO/M
ueller_Lab/mk/kaiju/kaiju_assembly/oyster/kaiju_oyster_assembly.out.krona
```

Testing hanna's vibrio genome to see if things are working well

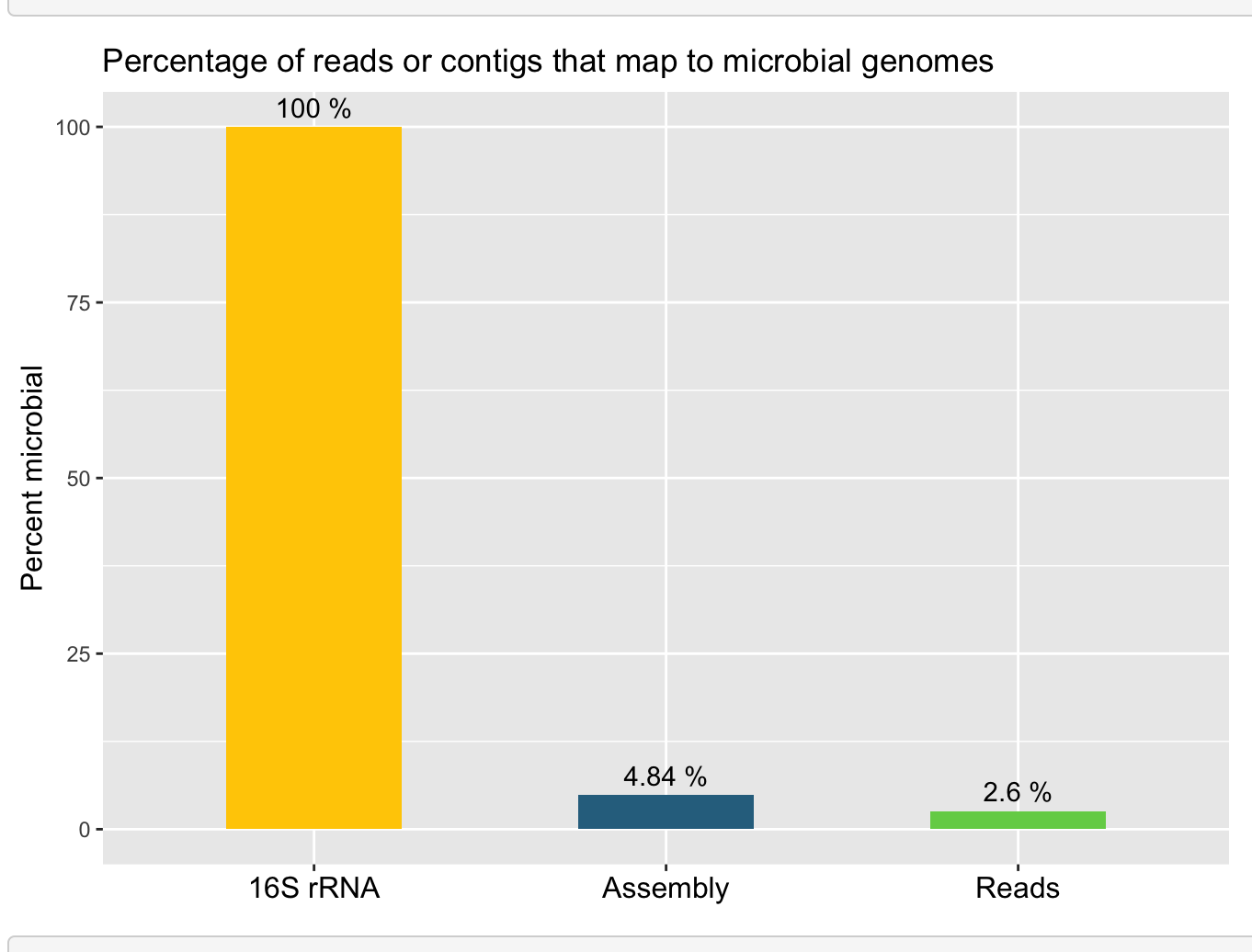
```
bin/kaiju -t /dfs/MICRO/Mueller_Lab/mk/kaiju/nodes.dmp -f /dfs/MICRO/Mueller_Lab/mk/kaiju/refseq/kaiju_db_refseq.
fmi -i /dfs/MICRO/Mueller_Lab/mk/kaiju/C154.scaffold.long.fna -o /dfs/MICRO/Mueller_Lab/mk/kaiju/C154_reads.out

bin/kaiju2krona -t nodes.dmp -n names.dmp -i /dfs/MICRO/Mueller_Lab/mk/kaiju/C154_reads.out -o /dfs/MICRO/Mueller
_Lab/mk/kaiju/C154_reads.out.krona

bin/ktImportText -o /dfs/MICRO/Mueller_Lab/mk/kaiju/C154_reads.html /dfs/MICRO/Mueller_Lab/mk/kaiju/C154_reads.ou
t.krona

bin/ktImportText -o /dfs/MICRO/Mueller_Lab/mk/exp18_1b_phylo.html /dfs/MICRO/Mueller_Lab/mk/krona_table.txt
```

Summary statistics

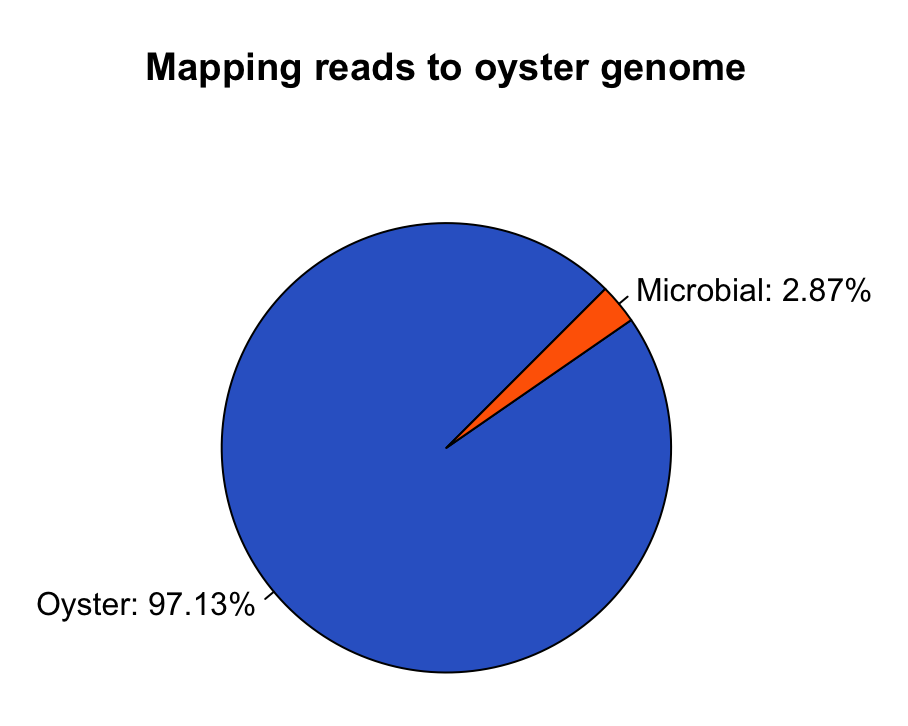
- Microbial reads:
  - 2.60% are microbial
  - ~97% eukaryotic
  - ~0.14% viruses
  - Number of microbial contigs: 275,107
- Microbial assembly:
  - 4.84% are microbial
  - ~95% eukaryotic
  - ~0.14% viruses
  - Number of microbial reads: 13,664
- Oyster reads:
  - 2.86% are microbial
  - 97.13% Crassostrea
  - 0 viruses
  - Number of oyster reads: 10,271,762
  - Number of microbial reads: 302,667
- Oyster assembly:
  - 0.005% microbial
  - 99.99% are Crassostrea
  - 0 viruses
  - Number of oyster contigs: 293,947
  - Number of microbial contigs: 16

Chart of these summary statistics

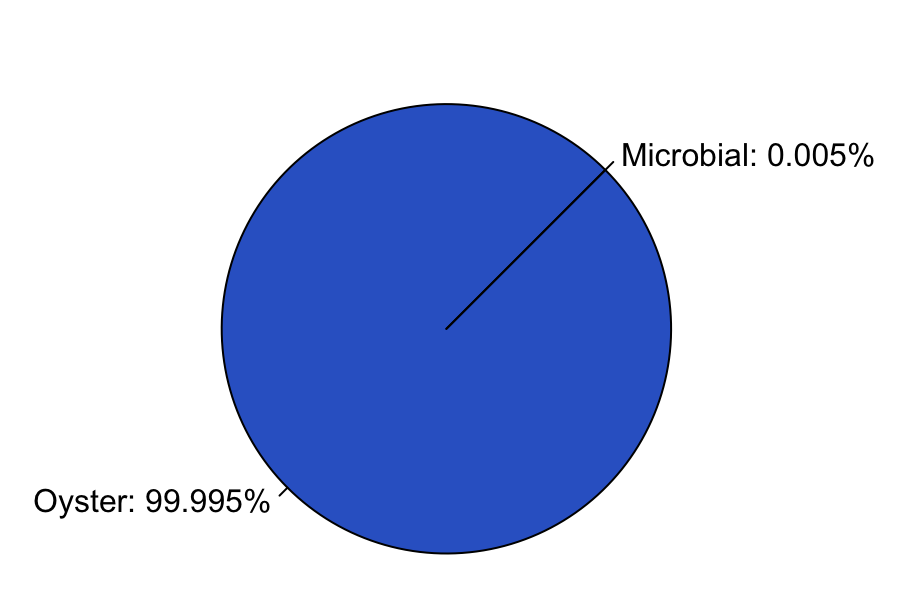
```r
# a summary bar chart
summarystats <- read_csv("~/Documents/2020Spring/mb668/metagenome/summarystats.csv")

summarystats$lab <- paste(summarystats$Percent_microbial, "%")

ggplot(summarystats) +
  geom_bar(aes(x = Method, y = Percent_microbial, fill = Method),
    stat = "identity",
          width = 0.5,
          position = "dodge", show.legend = FALSE) +
  geom_text(aes(x=Method, y = Percent_microbial, label = lab), vjust=-0.5)+
  theme(axis.text.x = element_text(colour = "black", size = 12),
        axis.title.y = element_text(colour = "black", size = 12))+
  labs(title = "Percentage of reads or contigs that map to microbial genomes", y = "Percent microbial", x = NULL)
+
  scale_fill_manual(values = c("#FFCC00","#2D708EFF", "#73D055FF"))
```




```r
# reads
oyster_reads_stats <- read_csv("~/Documents/2020Spring/mb668/metagenome/oyster_reads_stats.csv")

lbls <- paste(oyster_reads_stats$Type, ": ", oyster_reads_stats$Percentage, "%", sep = "")

pie(oyster_reads_stats$Percentage,
    labels = lbls,
    col = c("#3366CC", "#FF6600"),
    init.angle=45,
    main = "Mapping reads to oyster genome")
```

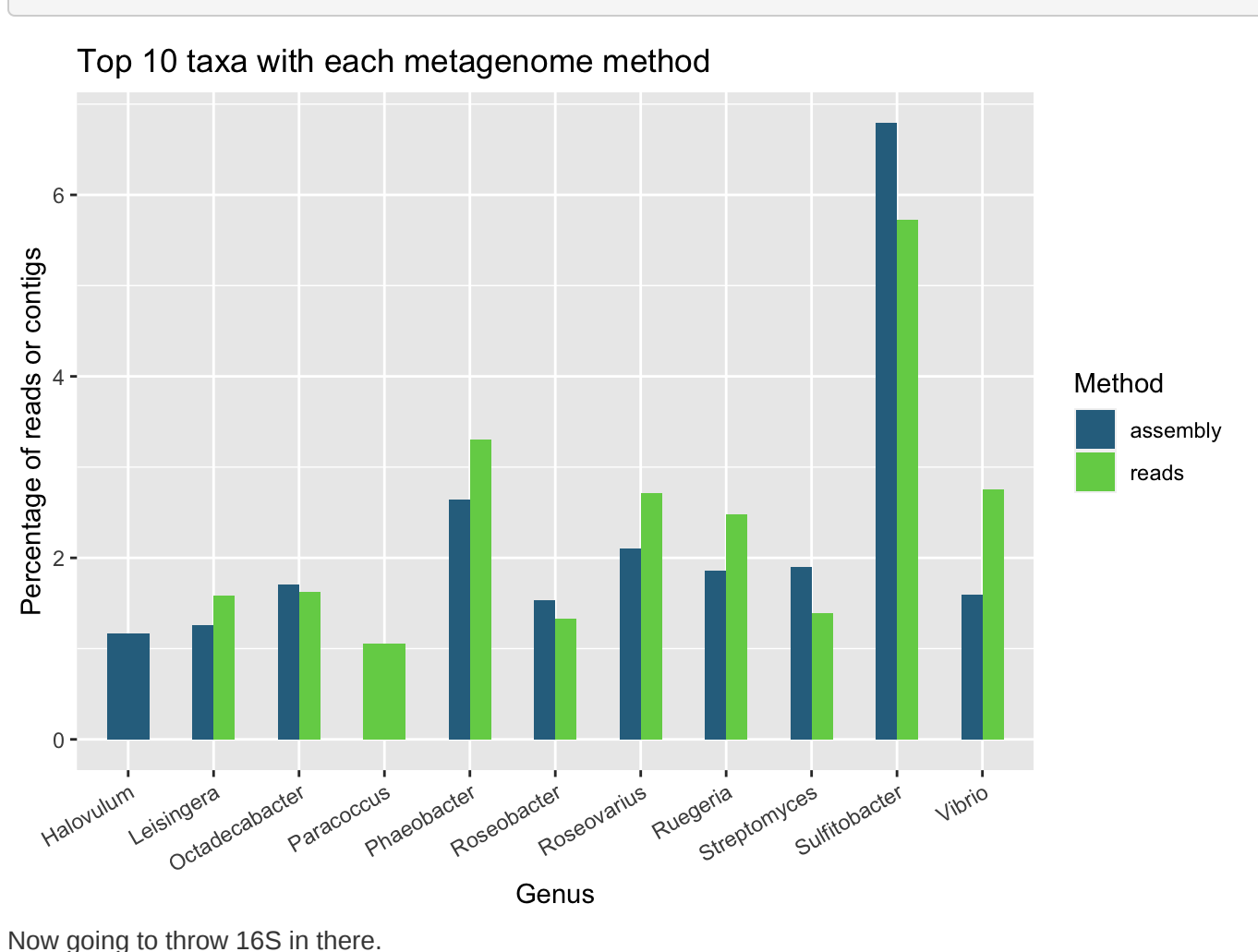



```r
# assembly
oyster_assem_stats <- read_csv("~/Documents/2020Spring/mb668/metagenome/oyster_assem_stats.csv")

lbls2 <- paste(oyster_assem_stats$Type, ": ", oyster_assem_stats$Percentage, "%", sep = "")

pie(oyster_assem_stats$Percentage,
    labels = lbls2,
    col = c("#3366CC", "#FF6600"),
    init.angle=45,
    main = "Mapping contigs to oyster genome")
```



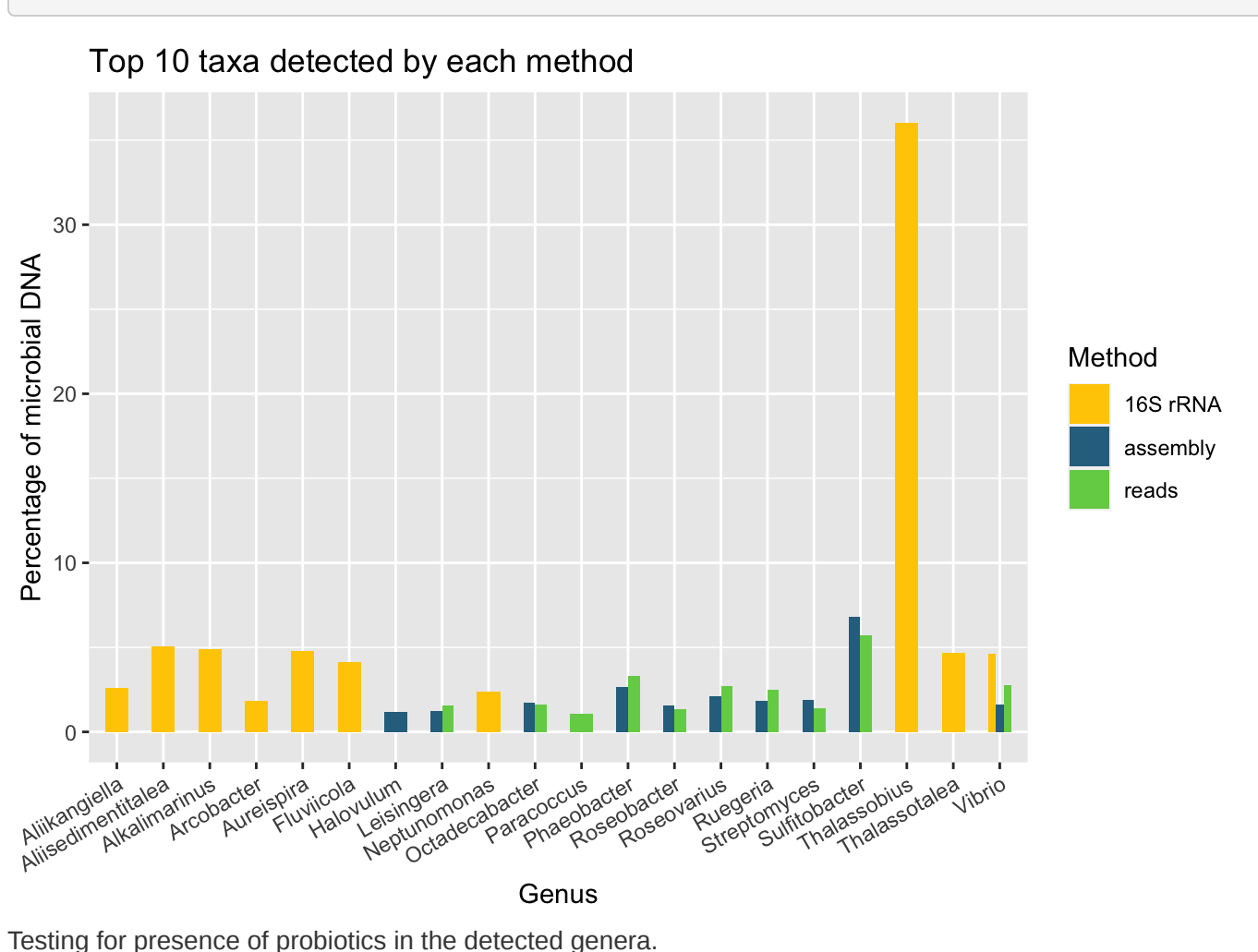
Comparison between the metagenome methods.

```r
mic_reads <- read_csv("~/Documents/2020Spring/mb668/metagenome/microbe_reads.csv")
mic_assem <- read_csv("~/Documents/2020Spring/mb668/metagenome/microbe_assembly.csv")
mic_comb <- rbind(mic_reads, mic_assem)

# top 10
mic_reads_10 <- mic_reads[1:10,]
mic_assem_10 <- mic_assem[1:10,]
mic_10 <- rbind(mic_reads_10, mic_assem_10)

# double bar plot of top 10 taxa in each
ggplot(data = mic_10, aes(x = Genus, y = Percent_Microbial, fill = Method))+
  geom_bar(stat = "identity",
          width = 0.5,
          position = "dodge")+
  theme(axis.text.x = element_text(angle = 30,
                                   hjust = 1))+
  labs(title = "Top 10 taxa with each metagenome method",
       x = "Genus",
       y = "Percentage of reads or contigs")+
  scale_fill_manual(values = c("#2D708EFF", "#73D055FF"))
```



Now going to throw 16S in there.

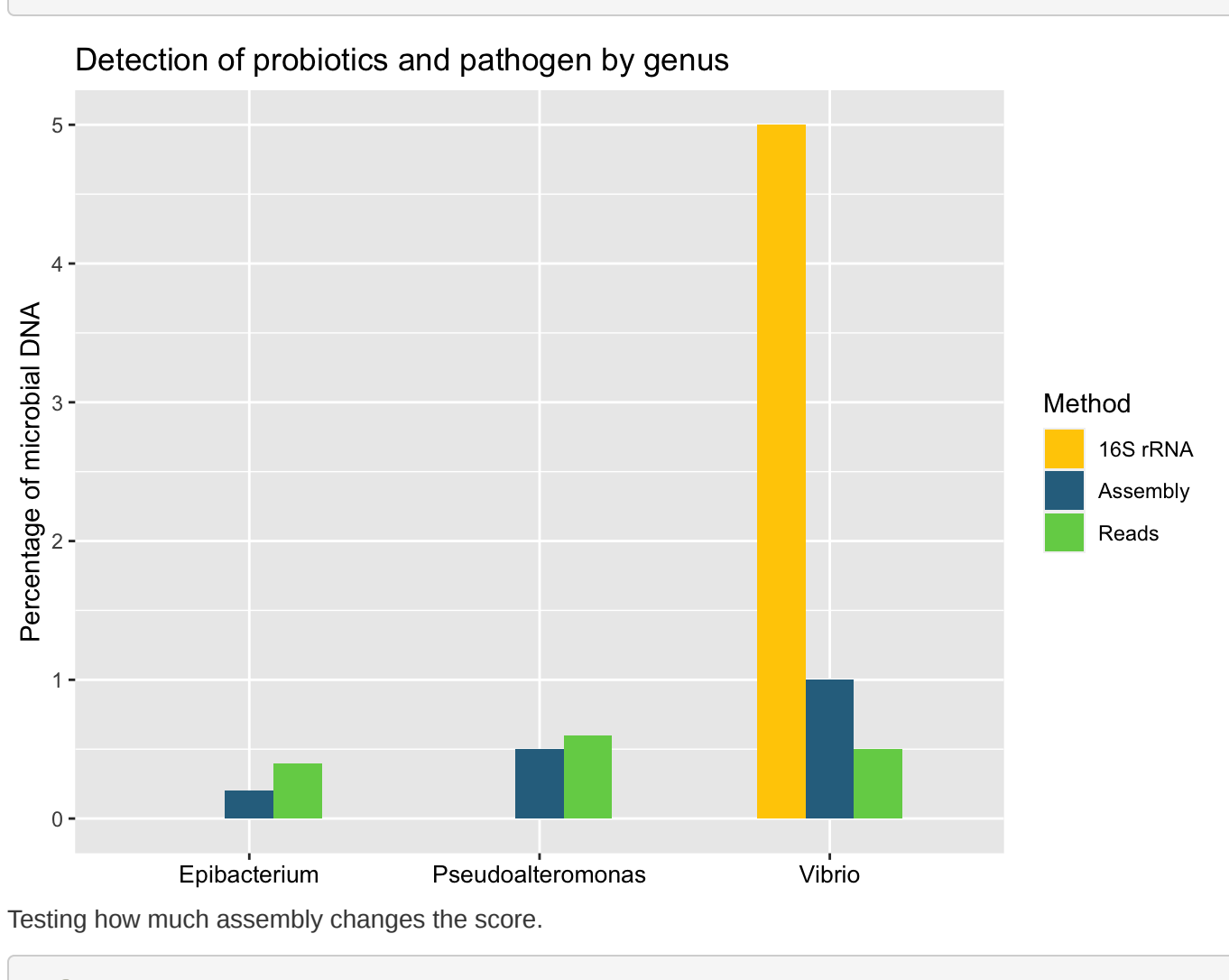
```r
rrna_table <- read_csv("~/Documents/2020Spring/mb668/metagenome/krona_table.csv")
rrna_noNA <- rrna_table[-c(4,12,21,22,25:28),]

# top 10
mic_reads_10_filt <- mic_reads_10[,-c(2,5)]
mic_assem_10_filt <- mic_assem_10[,-c(2,5)]
rrna_10 <- rrna_noNA[1:10,]

all_table_10 <- rbind(mic_reads_10_filt, mic_assem_10_filt, rrna_10)

ggplot(data=all_table_10, aes(x = Genus, y = Percent_Microbial, fill = Method))+
  geom_bar(stat = "identity",
          width = 0.5,
          position = "dodge")+
  theme(axis.text.x = element_text(angle = 30,
                                   hjust = 1))+
  labs(title = "Top 10 taxa detected by each method",
       x = "Genus",
       y = "Percentage of microbial DNA")+
  scale_fill_manual(values = c("#FFCC00","#2D708EFF", "#73D055FF"))
```



Testing for presence of probiotics in the detected genera.

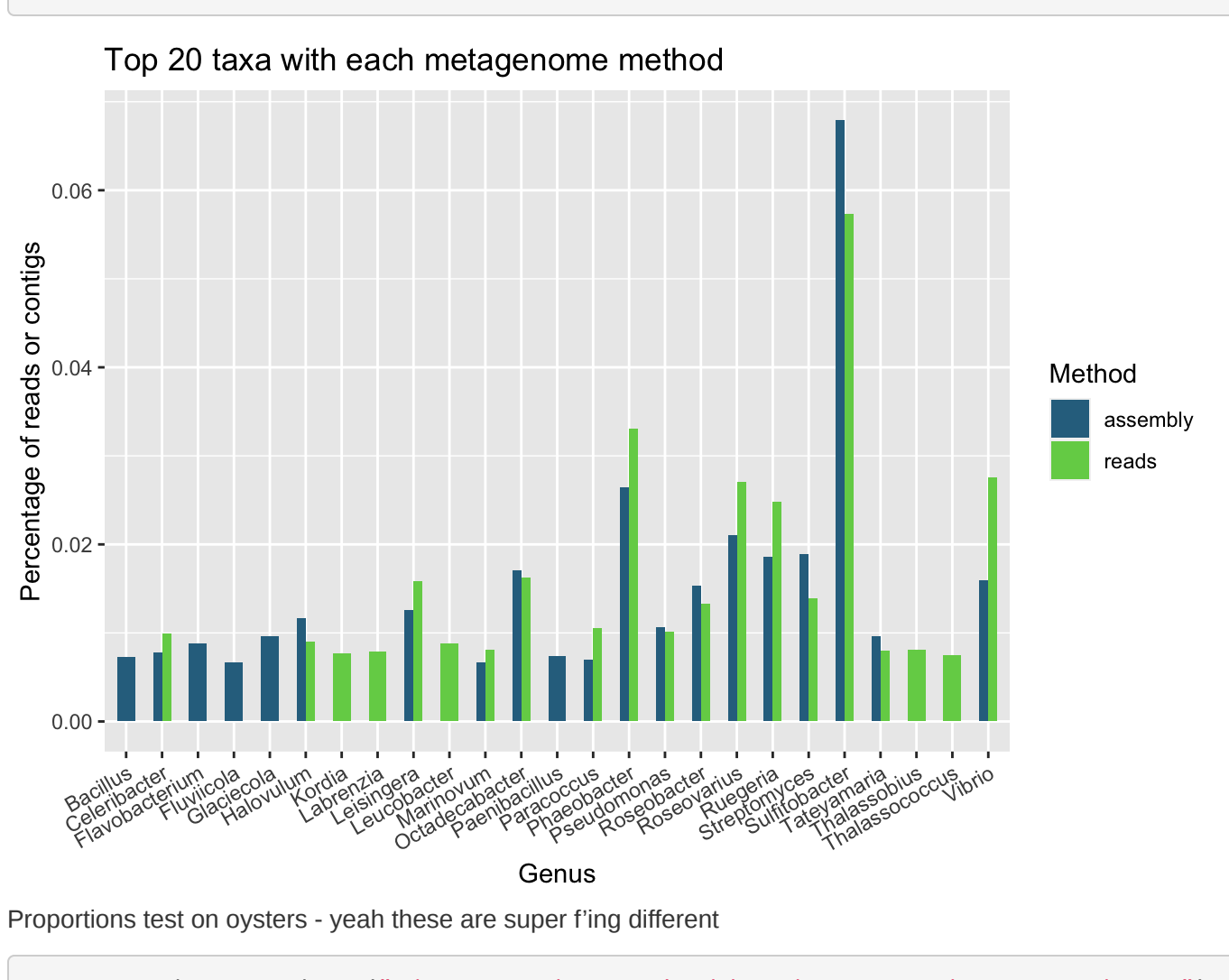
- B11: Epibacterium mobile (species) make up 0.4% of microbial reads, 0% of 16S, 0.2% of assembly
- DM14, D16: Pseudoalteromonas (genus): 0.6% of microbial reads, 0% of 16S, 0.5% of assembly
- Vibrio: Vibrio genus make up 5% of 16S but that doesn't tell us much because not all vibs are pathogenic, 0.5% of microbial reads, 1% of assembly.

```r
pbx_abund <- read_csv("~/Documents/2020Spring/mb668/metagenome/pbx_abund.csv")
ggplot(data = pbx_abund, aes(x = Genus, y = Percent_of_total, fill = Method))+
  geom_bar(stat = "identity",
          width = 0.5,
          position = "dodge")+
  theme(axis.text.x = element_text(color = "black", size = 10))+
  labs(title = "Detection of probiotics and pathogen by genus",
       x = NULL,
       y = "Percentage of microbial DNA")+
  scale_fill_manual(values = c("#FFCC00","#2D708EFF", "#73D055FF"))
```



Testing how much assembly changes the score.

```r
# for top 20 taxa
mic_reads_20 <- mic_reads[1:20,]
mic_assem_20 <- mic_assem[1:20,]
mic_reads_20$percentage <- (mic_reads_20$Percent_Microbial)/100
mic_assem_20$percentage <- (mic_assem_20$Percent_Microbial)/100
top20 <- rbind(mic_reads_20, mic_assem_20)
mic_reads_20$reads_percentage <- mic_reads_20$percentage
mic_assem_20$assem_percentage <- mic_assem_20$percentage

#plot
ggplot(data = top20, aes(x = Genus, y = percentage, fill = Method))+
  geom_bar(stat = "identity",
          width = 0.5,
          position = "dodge")+
  theme(axis.text.x = element_text(angle = 30,
                                   hjust = 1))+
  labs(title = "Top 20 taxa with each metagenome method",
       x = "Genus",
       y = "Percentage of reads or contigs")+
  scale_fill_manual(values = c("#2D708EFF", "#73D055FF"))
```



Proportions test on oysters - yeah these are super f'ing different

```r
oyster_reads <- read_csv("~/Documents/2020Spring/mb668/metagenome/oyster_reads.csv")
oyster_assem <- read_csv("~/Documents/2020Spring/mb668/metagenome/oyster_assembly.csv")
prop_reads <- oyster_reads$Percent_All[1]
x_reads <- oyster_reads$Count[1]
reads_total <- sum(oyster_reads$Count)
prop_assem <- oyster_assem$Percent_All[1]
x_assem <- oyster_assem$Count[1]
assem_total <- sum(oyster_assem$Count)
prop.test(x = c(x_reads, x_assem), n = c(reads_total, assem_total), alternative = "two.sided", correct = FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(x_reads, x_assem) out of c(reads_total, assem_total)
## X-squared = 8621.7, df = 1, p-value < 2.2e-16
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.02867209 -0.02846413
## sample estimates:
##    prop 1    prop 2
## 0.9713775 0.9999456
```

Proportions test on probiotic - Epibacterium

```r
microbe_reads <- read_csv("~/Documents/2020Spring/mb668/metagenome/microbe_reads.csv")
microbe_assembly <- read_csv("~/Documents/2020Spring/mb668/metagenome/microbe_assembly.csv")
reads_epi <- subset(microbe_reads, Genus == "Epibacterium")
assembly_epi <- subset(microbe_assembly, Genus == "Epibacterium")
prop.test(x = c(reads_epi$Count, assembly_epi$Count), n = c(reads_total, assem_total), alternative = "two.sided",
correct = FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(reads_epi$Count, assembly_epi$Count) out of c(reads_total, assem_total)
## X-squared = 0.75153, df = 1, p-value = 0.386
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -6.034501e-05  2.499958e-05
## sample estimates:
##       prop 1       prop 2
## 0.0001183988 0.0001360715
```

No, epibacterium is not significant.

Proportions test on probiotic = Pseudoalteromonas

```r
reads_pseudo <- subset(microbe_reads, Genus == "Pseudoalteromonas")
assembly_pseudo <- subset(microbe_assembly, Genus == "Pseudoalteromonas")
prop.test(x = c(reads_pseudo$Count, assembly_pseudo$Count), n = c(reads_total, assem_total), alternative = "two.s
ided", correct = FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(reads_pseudo$Count, assembly_pseudo$Count) out of c(reads_total, assem_total)
## X-squared = 13.031, df = 1, p-value = 0.0003064
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -1.557274e-04 -3.239544e-05
## sample estimates:
##       prop 1       prop 2
## 0.0001916888 0.0002857502
```

Yes, pseudoalteromonas is significant.
Proportions test on pathogen (Vibrio)

```r
reads_vibrio <- subset(microbe_reads, Genus == "Vibrio")
assembly_vibrio <- subset(microbe_assembly, Genus == "Vibrio")
prop.test(x = c(reads_vibrio$Count, assembly_vibrio$Count), n = c(reads_total, assem_total), alternative = "two.s
ided", correct = FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  c(reads_vibrio$Count, assembly_vibrio$Count) out of c(reads_total, assem_total)
## X-squared = 0.25622, df = 1, p-value = 0.6127
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -1.250524e-04  7.438484e-05
## sample estimates:
##       prop 1       prop 2
## 0.0007162562 0.0007415899
```

No, vibrio is not significant.