Comparison of sequence processing methods

In this document, we evaluate how different clustering methods and denoising approach for long amplicon sequences influence microbial abundance at the genus level. Note that in the heatmaps, comparisons are made across rows, not columns. Though, genera is sorted in descending order, colors do not reflect abundance differences.

Libraries

```
library(mia)
library(pheatmap)
library(viridis)
library(kableExtra)
```

Import data

Data is reloaded from tse files

```
closedref <- readRDS("results/closedref/tse.rds")
openref <- readRDS("results/openref/tse.rds")
denovo <- readRDS("results/denovo/tse.rds")
denoise <- readRDS("results/denoised/tse.rds")
```

Rarefaction and transformation

Each object is rarefied to 114,000 counts to minimize the effect of varying sample sizes on comparison between methods.

```
set.seed(3)
closedref <- subsampleCounts(closedref, assay.type="counts",
                             min_size=114000, name="subsampled")
openref <- subsampleCounts(openref, assay.type="counts",
                           min_size=114000, name="subsampled")
denovo <- subsampleCounts(denovo, assay.type="counts",
                          min_size=114000, name="subsampled")
denoise <- subsampleCounts(denoise, assay.type="counts",
                           min_size=114000, name="subsampled")
```

Taxonomy is aggregated at the genus level, and abundance data is converted to relative abundance.

```
closedref <- agglomerateByRank(closedref, rank="Genus", na.rm=T)
closedref <- transformAssay(closedref, assay.type="subsampled",
                            method="relabundance")
openref <- agglomerateByRank(openref, rank="Genus", na.rm=T)
openref <- transformAssay(openref, assay.type="subsampled",
                          method="relabundance")
denovo <- agglomerateByRank(denovo, rank="Genus", na.rm=T)
denovo <- transformAssay(denovo, assay.type="subsampled",
                         method="relabundance")
denoise <- agglomerateByRank(denoise, rank="Genus", na.rm=T)
denoise <- transformAssay(denoise, assay.type="subsampled",
                         method="relabundance")
```

Closed vs Open reference

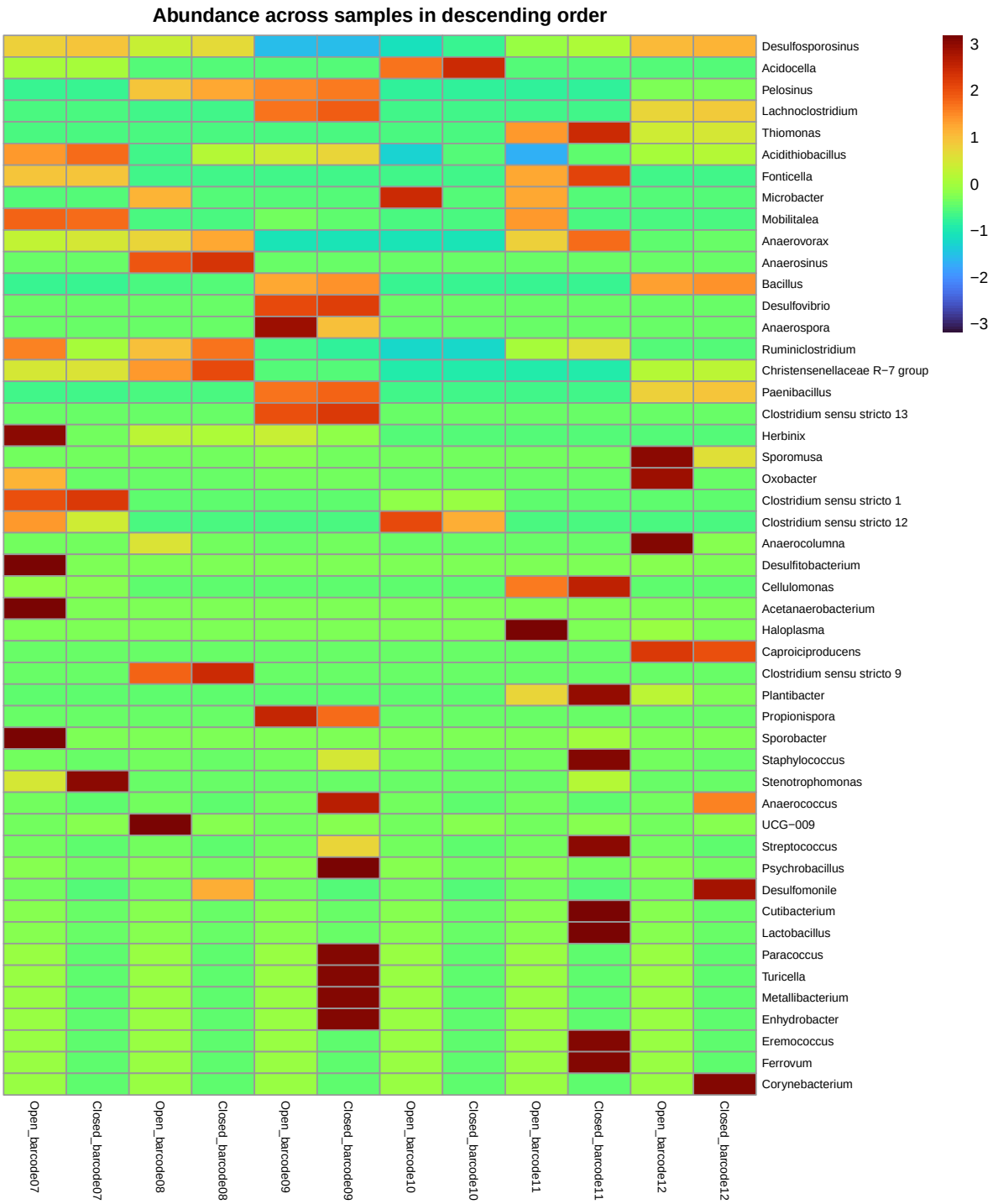
The first comparison pair is closed-reference and open-reference OTU picking. Since not all genera are found in both datasets, a value of 1e-6 is added to the empty columns.

```
# Extract the assay data
assay_open <- assay(openref, "relabundance")
assay_closed <- assay(closedref, "relabundance")
# Unify genera
unified <- union(rownames(assay_open), rownames(assay_closed))
# Create empty matrices with the full list of genera and existing samples
# Fill missing genera with pseudocounts (1e-6)
assay_open_filled <- matrix(1e-6, nrow = length(unified), ncol = ncol(assay_open),
                            dimnames = list(unified, colnames(assay_open)))
assay_closed_filled <- matrix(1e-6, nrow = length(unified), ncol = ncol(assay_closed),
                              dimnames = list(unified, colnames(assay_closed)))
# Fill in the original data for genera that are present
assay_open_filled[rownames(assay_open), ] <- assay_open
assay_closed_filled[rownames(assay_closed), ] <- assay_closed
# Combine the data matrices
merged <- cbind(assay_open_filled, assay_closed_filled)
# Ensure column names are unique for clarity
colnames(merged) <- c(
  paste0("Open_", colnames(assay_open)),
  paste0("Closed_", colnames(assay_closed)))
# Reorder the columns so that each open-closed pair is next to each other
sample_names <- colnames(assay_open)

ordered_columns <- unlist(lapply(sample_names, function(x) c(
  paste0("Open_", x),
  paste0("Closed_", x)
)))
merged <- merged[, ordered_columns]
# Calculate the sum of abundances for each genus across all samples
abundance_sums <- rowSums(merged)
# Order the rows based on descending abundance
merged_sorted <- merged[order(abundance_sums, decreasing = TRUE), ]
```

Heatmap comparison

```
# Create the heatmap with sorted genera
pheatmap(
  merged_sorted,
  cluster_rows = FALSE,  # Disable clustering to keep the sorted order
  cluster_cols = FALSE,  # Keep open-closed pairs together
  main = "Abundance across samples in descending order",
  scale = "row",  # Scale the data across rows (genera)
  color = turbo(100),
  fontsize = 8,          # Adjust the overall font size
  fontsize_row = 6,       # Adjust the font size for row labels
  fontsize_col = 6        # Adjust the font size for column labels
)
```


Abundance across samples in descending order
Desulfosporosinus
Acidocella
Pelosinus
Lachnoclostridium
Thiomonas
Acidithiobacillus
Fonticella
Microbacter
Mobilitalea
Anaerovorax
Anaerosinus
Bacillus
Desulfovibrio
Anaerospora
Ruminiclostridium
Christensenellaceae R−7 group
Paenibacillus
Clostridium sensu stricto 13
Herbinix
Sporomusa
Oxobacter
Clostridium sensu stricto 1
Clostridium sensu stricto 12
Anaerocolumna
Desulfitobacterium
Cellulomonas
Acetanaerobacterium
Haloplasma
Caproiciproducens
Clostridium sensu stricto 9
Plantibacter
Propionispora
Sporobacter
Staphylococcus
Stenotrophomonas
Anaerococcus
UCG−009
Streptococcus
Psychrobacillus
Desulfomonile
Cutibacterium
Lactobacillus
Paracoccus
Turicella
Metallibacterium
Enhydrobacter
Eremococcus
Ferrovum
Corynebacterium
3
2
1
0
−1
−2
−3
Open_barcode07
Closed_barcode07
Open_barcode08
Closed_barcode08
Open_barcode09
Closed_barcode09
Open_barcode10
Closed_barcode10
Open_barcode11
Closed_barcode11
Open_barcode12
Closed_barcode12

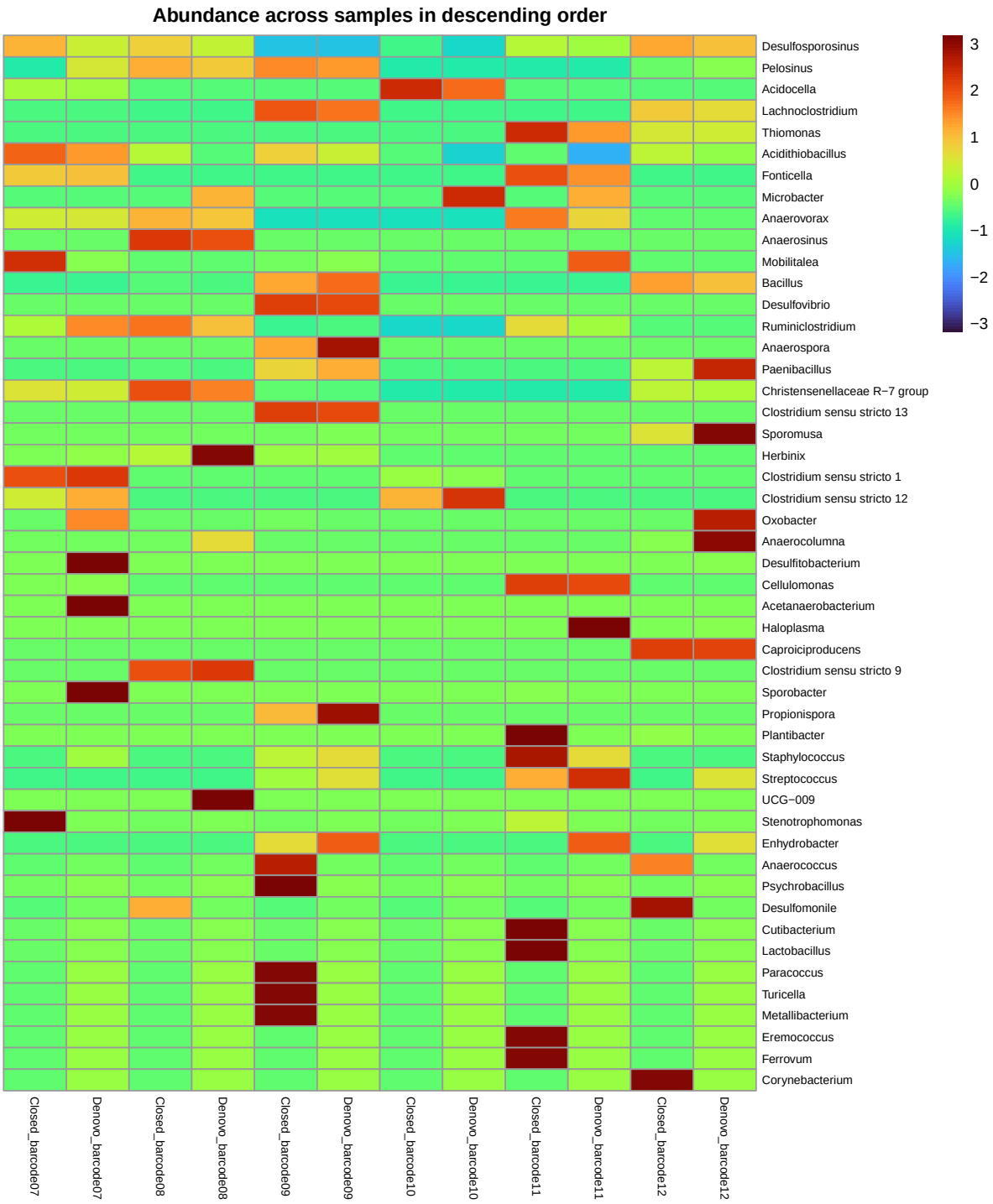
Closed reference vs de-novo

```
# Extract the assay data
assay_closed <- assay(closedref, "relabundance")
assay_denovo <- assay(denovo, "relabundance")
# Unify all genera
unified <- union(rownames(assay_closed), rownames(assay_denovo))
# Create empty matrices with the full list of genera and existing samples
# Fill missing genera with pseudocounts (1e-6)
assay_closed_filled <- matrix(1e-6, nrow = length(unified), ncol = ncol(assay_closed),
                              dimnames = list(unified, colnames(assay_closed)))
assay_denovo_filled <- matrix(1e-6, nrow = length(unified), ncol = ncol(assay_denovo),
                               dimnames = list(unified, colnames(assay_denovo)))
# Fill in the original data for genera that are present
assay_closed_filled[rownames(assay_closed), ] <- assay_closed
assay_denovo_filled[rownames(assay_denovo), ] <- assay_denovo
# Combine the data matrices
merged <- cbind(assay_closed_filled, assay_denovo_filled)
# Ensure column names are unique for clarity
colnames(merged) <- c(
  paste0("Closed_", colnames(assay_closed)),
  paste0("Denovo_", colnames(assay_denovo))
)
# Reorder the columns so that each open-closed pair is next to each other
sample_names <- colnames(assay_closed)
ordered_columns <- unlist(lapply(sample_names, function(x) c(
  paste0("Closed_", x),
  paste0("Denovo_", x)
)))
merged <- merged[, ordered_columns]
# Calculate the sum of abundances for each genus across all samples
abundance_sums <- rowSums(merged)

# Order the rows based on descending abundance
merged_sorted <- merged[order(abundance_sums, decreasing = TRUE), ]
```

Heatmap comparison

```
# Create the heatmap with sorted genera
pheatmap(
  merged_sorted,
  cluster_rows = FALSE,  # Disable clustering to keep the sorted order
  cluster_cols = FALSE,  # Keep open-closed pairs together
  main = "Abundance across samples in descending order",
  scale = "row",  # Scale the data across rows (genera)
  color = turbo(100),
  fontsize = 8,          # Adjust the overall font size
  fontsize_row = 6,       # Adjust the font size for row labels
  fontsize_col = 6        # Adjust the font size for column labels
)
```


Abundance across samples in descending order
Desulfosporosinus
Pelosinus
Acidocella
Lachnoclostridium
Thiomonas
Acidithiobacillus
Fonticella
Microbacter
Anaerovorax
Anaerosinus
Mobilitalea
Bacillus
Desulfovibrio
Ruminiclostridium
Anaerospora
Paenibacillus
Christensenellaceae R−7 group
Clostridium sensu stricto 13
Sporomusa
Herbinix
Clostridium sensu stricto 1
Clostridium sensu stricto 12
Oxobacter
Anaerocolumna
Desulfitobacterium
Cellulomonas
Acetanaerobacterium
Haloplasma
Caproiciproducens
Clostridium sensu stricto 9
Sporobacter
Propionispora
Plantibacter
Staphylococcus
Streptococcus
UCG−009
Stenotrophomonas
Enhydrobacter
Anaerococcus
Psychrobacillus
Desulfomonile
Cutibacterium
Lactobacillus
Paracoccus
Turicella
Metallibacterium
Eremococcus
Ferrovum
Corynebacterium
3
2
1
0
−1
−2
−3
Closed_barcode07
Denovo_barcode07
Closed_barcode08
Denovo_barcode08
Closed_barcode09
Denovo_barcode09
Closed_barcode10
Denovo_barcode10
Closed_barcode11
Denovo_barcode11
Closed_barcode12
Denovo_barcode12

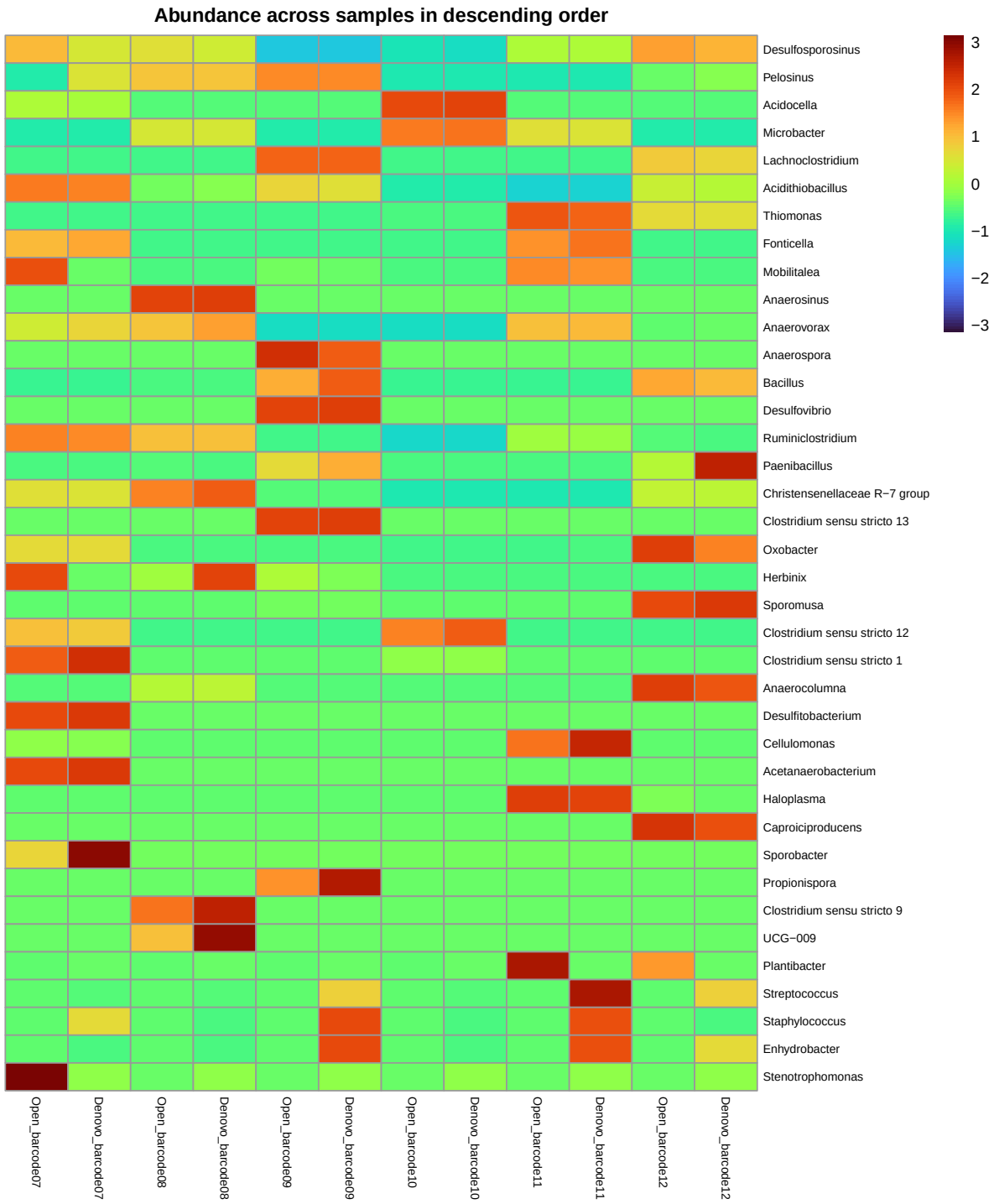
Open reference vs de-novo

```
# Extract the assay data
assay_open <- assay(openref, "relabundance")
assay_denovo <- assay(denovo, "relabundance")
# Unify all genera
unified <- union(rownames(assay_open), rownames(assay_denovo))
# Create empty matrices with the full list of genera and existing samples
# Fill missing genera with pseudocounts (1e-6)
assay_open_filled <- matrix(1e-6, nrow = length(unified), ncol = ncol(assay_open),
                            dimnames = list(unified, colnames(assay_open)))
assay_denovo_filled <- matrix(1e-6, nrow = length(unified), ncol = ncol(assay_denovo),
                              dimnames = list(unified, colnames(assay_denovo)))
# Fill in the original data for genera that are present
assay_open_filled[rownames(assay_open), ] <- assay_open
assay_denovo_filled[rownames(assay_denovo), ] <- assay_denovo
# Combine the data matrices
merged <- cbind(assay_open_filled, assay_denovo_filled)
# Ensure column names are unique for clarity
colnames(merged) <- c(
  paste0("Open_", colnames(assay_open)),
  paste0("Denovo_", colnames(assay_denovo))
)
# Reorder the columns so that each open-closed pair is next to each other
sample_names <- colnames(assay_closed)
ordered_columns <- unlist(lapply(sample_names, function(x) c(
  paste0("Open_", x),
  paste0("Denovo_", x)
)))
merged <- merged[, ordered_columns]
# Calculate the sum of abundances for each genus across all samples
abundance_sums <- rowSums(merged)

# Order the rows based on descending abundance
merged_sorted <- merged[order(abundance_sums, decreasing = TRUE), ]
```

Heatmap comparison

```
# Create the heatmap with sorted genera
pheatmap(
  merged_sorted,
  cluster_rows = FALSE,  # Disable clustering to keep the sorted order
  cluster_cols = FALSE,  # Keep open-closed pairs together
  main = "Abundance across samples in descending order",
  scale = "row",  # Scale the data across rows (genera)
  color = turbo(100),
  fontsize = 8,          # Adjust the overall font size
  fontsize_row = 6,       # Adjust the font size for row labels
  fontsize_col = 6        # Adjust the font size for column labels
)
```


Abundance across samples in descending order
Desulfosporosinus
Pelosinus
Acidocella
Microbacter
Lachnoclostridium
Acidithiobacillus
Thiomonas
Fonticella
Mobilitalea
Anaerosinus
Anaerovorax
Anaerospora
Bacillus
Desulfovibrio
Ruminiclostridium
Paenibacillus
Christensenellaceae R−7 group
Clostridium sensu stricto 13
Oxobacter
Herbinix
Sporomusa
Clostridium sensu stricto 12
Clostridium sensu stricto 1
Anaerocolumna
Desulfitobacterium
Cellulomonas
Acetanaerobacterium
Haloplasma
Caproiciproducens
Sporobacter
Propionispora
Clostridium sensu stricto 9
UCG−009
Plantibacter
Streptococcus
Staphylococcus
Enhydrobacter
Stenotrophomonas
3
2
1
0
−1
−2
−3
Open_barcode07
Denovo_barcode07
Open_barcode08
Denovo_barcode08
Open_barcode09
Denovo_barcode09
Open_barcode10
Denovo_barcode10
Open_barcode11
Denovo_barcode11
Open_barcode12
Denovo_barcode12

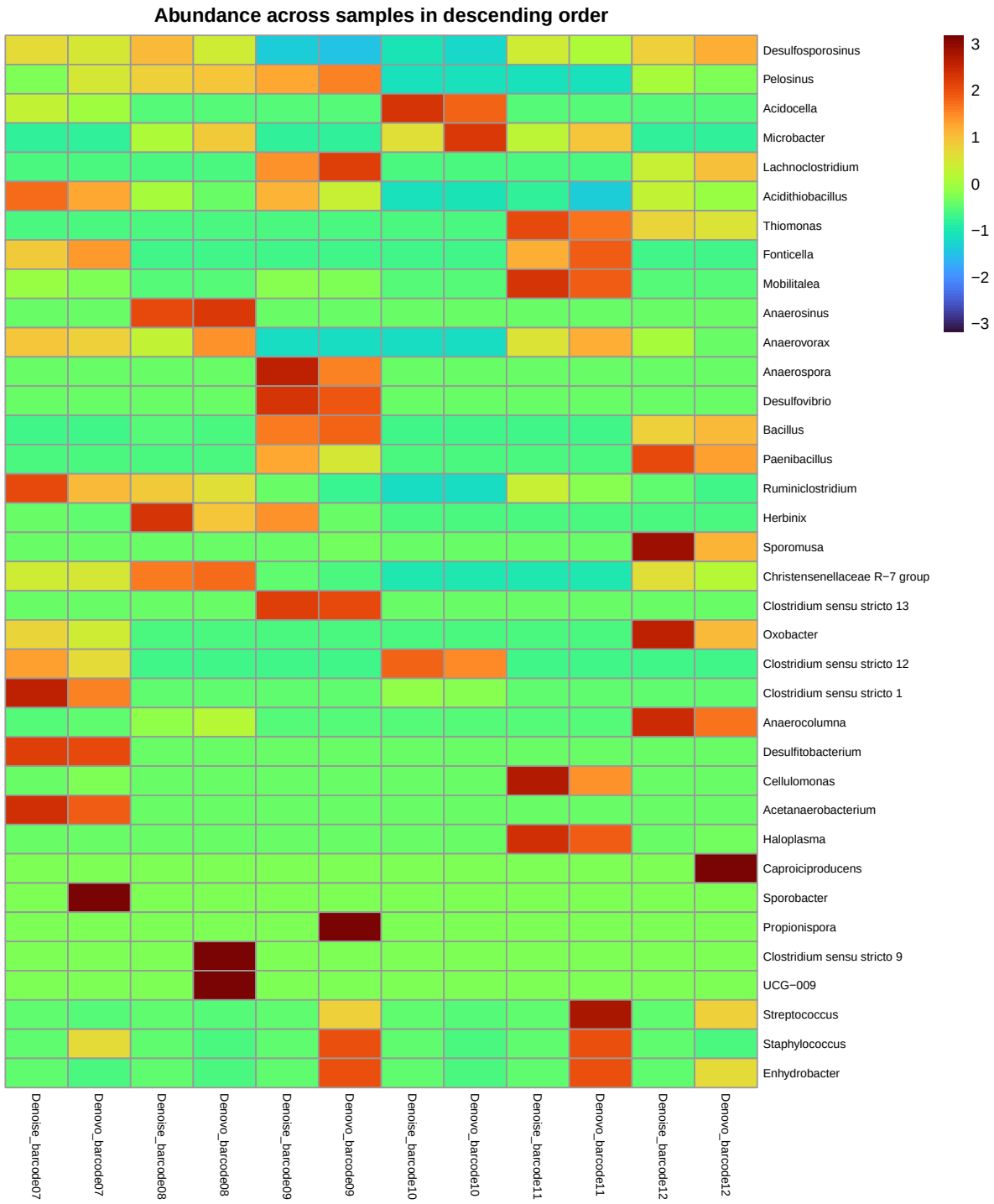
Denoise vs de-novo

```
# Extract the assay data
assay_denoise <- assay(denoise, "relabundance")
# Fix names discrepance
colnames(assay_denoise) <- c("barcode07", "barcode08", "barcode09", "barcode10", "barcode11", "barcode12")
assay_denovo <- assay(denovo, "relabundance")
# Unify all genera
unified <- union(rownames(assay_denoise), rownames(assay_denovo))
# Create empty matrices with the full list of genera and existing samples
# Fill missing genera with pseudocounts (1e-6)
assay_denoise_filled <- matrix(1e-6, nrow = length(unified), ncol = ncol(assay_denoise),
                               dimnames = list(unified, colnames(assay_denoise)))
assay_denovo_filled <- matrix(1e-6, nrow = length(unified), ncol = ncol(assay_denovo),
                              dimnames = list(unified, colnames(assay_denovo)))
# Fill in the original data for genera that are present
assay_denoise_filled[rownames(assay_denoise), ] <- assay_denoise
assay_denovo_filled[rownames(assay_denovo), ] <- assay_denovo
# Combine the data matrices
merged <- cbind(assay_denoise_filled, assay_denovo_filled)
# Ensure column names are unique for clarity
colnames(merged) <- c(
  paste0("Denoise_", colnames(assay_denoise)),
  paste0("Denovo_", colnames(assay_denovo))
)
# Reorder the columns so that each open-closed pair is next to each other
sample_names <- colnames(assay_denoise)
colnames(assay_denoise) <- colnames(assay_denovo)
ordered_columns <- unlist(lapply(sample_names, function(x) c(
  paste0("Denoise_", x),
  paste0("Denovo_", x)
)))
merged <- merged[, ordered_columns]
# Calculate the sum of abundances for each genus across all samples
abundance_sums <- rowSums(merged)

# Order the rows based on descending abundance
merged_sorted <- merged[order(abundance_sums, decreasing = TRUE), ]
```

Heatmap comparison

```
# Create the heatmap with sorted genera
pheatmap(
  merged_sorted,
  cluster_rows = FALSE,  # Disable clustering to keep the sorted order
  cluster_cols = FALSE,  # Keep open-closed pairs together
  main = "Abundance across samples in descending order",
  scale = "row",  # Scale the data across rows (genera)
  color = turbo(100),
  fontsize = 8,          # Adjust the overall font size
  fontsize_row = 6,       # Adjust the font size for row labels
  fontsize_col = 6        # Adjust the font size for column labels
)
```


Abundance across samples in descending order
Desulfosporosinus
Pelosinus
Acidocella
Microbacter
Lachnoclostridium
Acidithiobacillus
Thiomonas
Fonticella
Mobilitalea
Anaerosinus
Anaerovorax
Anaerospora
Desulfovibrio
Bacillus
Paenibacillus
Ruminiclostridium
Herbinix
Sporomusa
Christensenellaceae R−7 group
Clostridium sensu stricto 13
Oxobacter
Clostridium sensu stricto 12
Clostridium sensu stricto 1
Anaerocolumna
Desulfitobacterium
Cellulomonas
Acetanaerobacterium
Haloplasma
Caproiciproducens
Sporobacter
Propionispora
Clostridium sensu stricto 9
UCG−009
Streptococcus
Staphylococcus
Enhydrobacter
3
2
1
0
−1
−2
−3
Denoise_barcode07
Denovo_barcode07
Denoise_barcode08
Denovo_barcode08
Denoise_barcode09
Denovo_barcode09
Denoise_barcode10
Denovo_barcode10
Denoise_barcode11
Denovo_barcode11
Denoise_barcode12
Denovo_barcode12

Comparing richness

Reloading original data

```
closedref <- readRDS("results/closedref/tse.rds")
openref <- readRDS("results/openref/tse.rds")
denovo <- readRDS("results/denovo/tse.rds")
denoise <- readRDS("results/denoised/tse.rds")
set.seed(3)
closedref <- subsampleCounts(closedref, min_size=114000,
                             name="subsampled", verbose=F)
openref <- subsampleCounts(openref, min_size=114000,
                           name="subsampled", verbose=F)
denovo <- subsampleCounts(denovo, min_size=114000,
                          name="subsampled", verbose=F)
denoise <- subsampleCounts(denoise, min_size=114000,
                           name="subsampled", verbose=F)
```

Calculating shannon values

```
closedref <- estimateDiversity(closedref, assay.type="subsampled",
                               index="shannon")
openref <- estimateDiversity(openref, assay.type="subsampled",
                             index="shannon" )
denovo <- estimateDiversity(denovo, assay.type="subsampled",
                            index="shannon")
denoise <- estimateDiversity(denoise, assay.type="subsampled",
                             index="shannon")
```

Combine data to the table

```
richness <- data.frame(Closed_Ref=colData(closedref)$shannon,
                       Open_Ref=colData(openref)$shannon,
                       Denovo=colData(denovo)$shannon,
                       Denoise=colData(denoise)$shannon)
kable(richness, caption="Alpha diversity", booktabs=T, digits=2) %>%
kable_styling(latex_options=c("striped", "HOLD_position", "repeat_header"),
font_size = 11) %>%
row_spec(0,background = "teal", color = "white")
```

Table 1: Alpha diversity

	Closed_Ref	Open_Ref	Denovo	Denoise
barcode07	2.48	2.85	3.46	5.08
barcode08	2.89	3.28	3.45	5.07
barcode09	2.23	2.59	2.41	5.22
barcode10	0.99	1.39	1.71	3.85
barcode11	2.38	2.78	2.80	4.87
barcode12	2.40	2.67	2.57	5.10

Analysis

Analysis of the complete genus information from the samples did not reveal significant differences between the various sequence processing methods, particularly given that the ten most abundant genera account for a vast majority of the communities. The most notable difference

observed is that the closed-reference approach fails to detect *Microbacter* due to its identity being lower than 97% compared to the reference database. Its frequent occurrence in this dataset makes it highly unlikely that this is an error.

The richness values suggest that the *de-novo* method is most effective for preserving taxonomic information. Conversely, the Shannon index indicates that denoising may struggle to correct sequencing errors in long amplicon products. But erroneous variants could cluster within the same genera, leading to taxonomic abundance patterns close to results from other methods. Denoising could be more effective with shorter amplicons, where a higher proportion of error-free sequences are present.

This also would explain why denoising yields nearly perfect results for mock community samples, which consist of a few microbes with known abundances.