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")

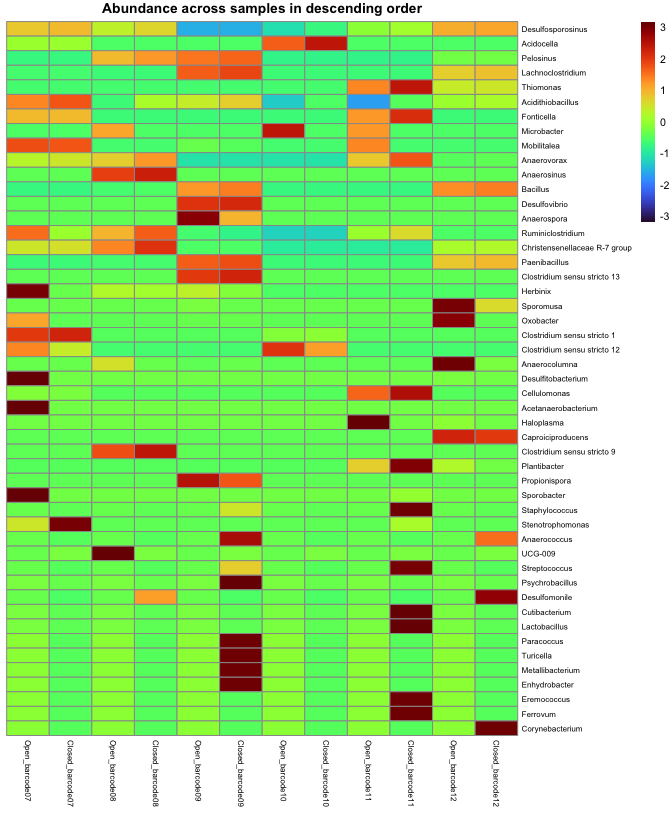
#### Closed vs Open reference

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  
)

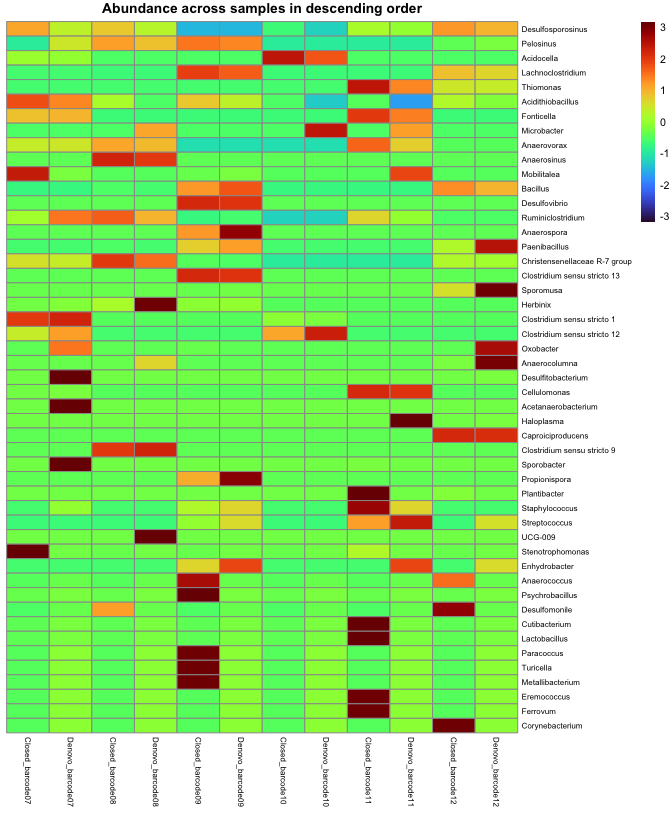


#### Closed reference vs de-novo

# 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  
)

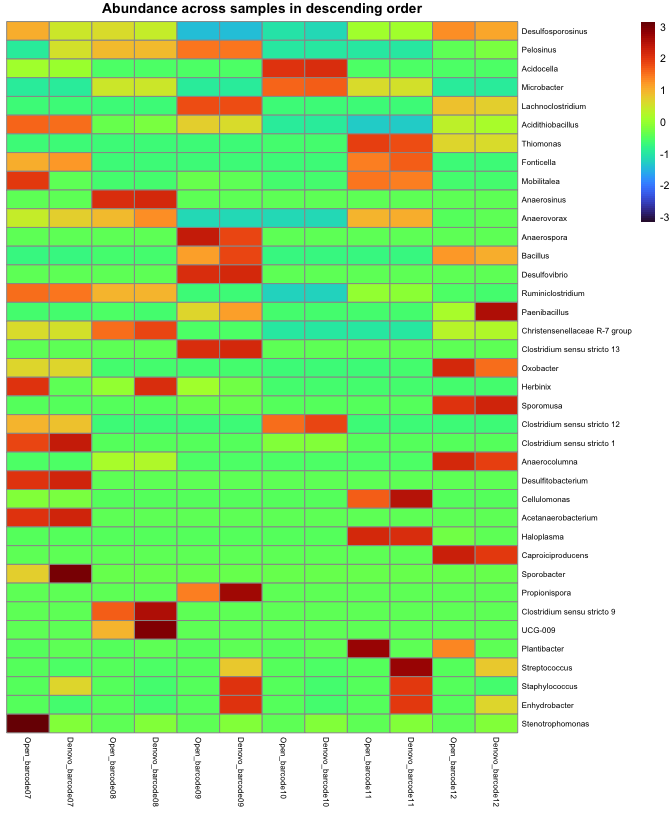


#### Open reference vs de-novo

# 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  
)

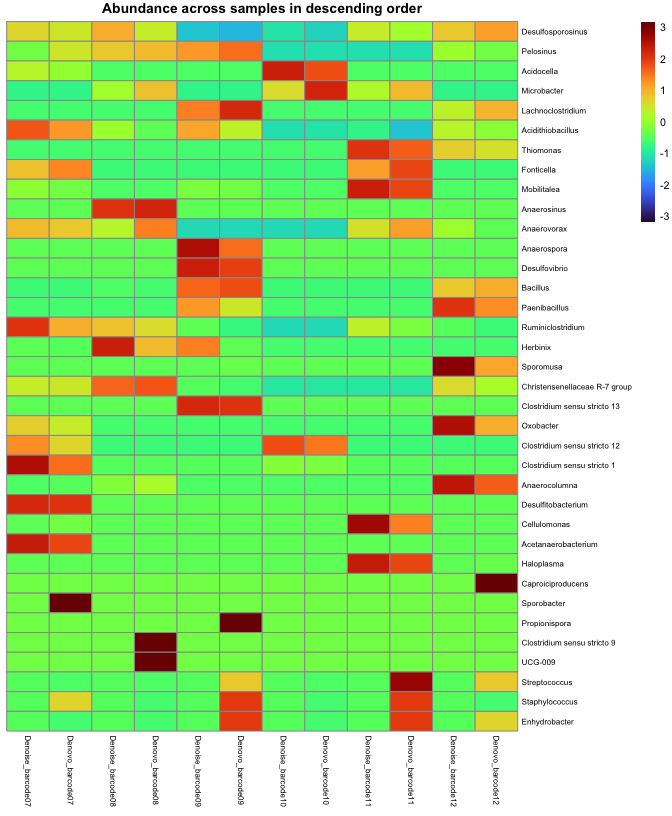


#### Denoise vs de-novo

# 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  
)



#### 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")

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.