Cluster for microbiome count data via K-means

Weijia Xiong

6/25/2020

```
load("data/DiGiulio.RData")
otu_data = as.data.frame(DiGiulio$OTU)  # 927 samples, 1271 OTU
taxonomy = DiGiulio$Taxonomy  # 1271
sampledata = DiGiulio$SampleData  #  927 samples, other covariates
```

```
otu_data_all=
  cbind(sampledata, otu_data) %>%
  mutate(
    Preg = as.factor(Preg),
    Subject = as.factor(Subject)
  ) %>%
  na.omit()
rownames(otu_data_all) = sampledata$SampleID
```

```
term =
  otu_data_all %>%
  filter(preterm == "Term")

preterm =
  otu_data_all %>%
  filter(preterm != "Term")
```

Term

```
term_count =
  term %>%
  select(-SampleID,-Subject,-weeks,-Race,-NumReads,-Preg,-preterm,-CST)
ncol(term_count)
```

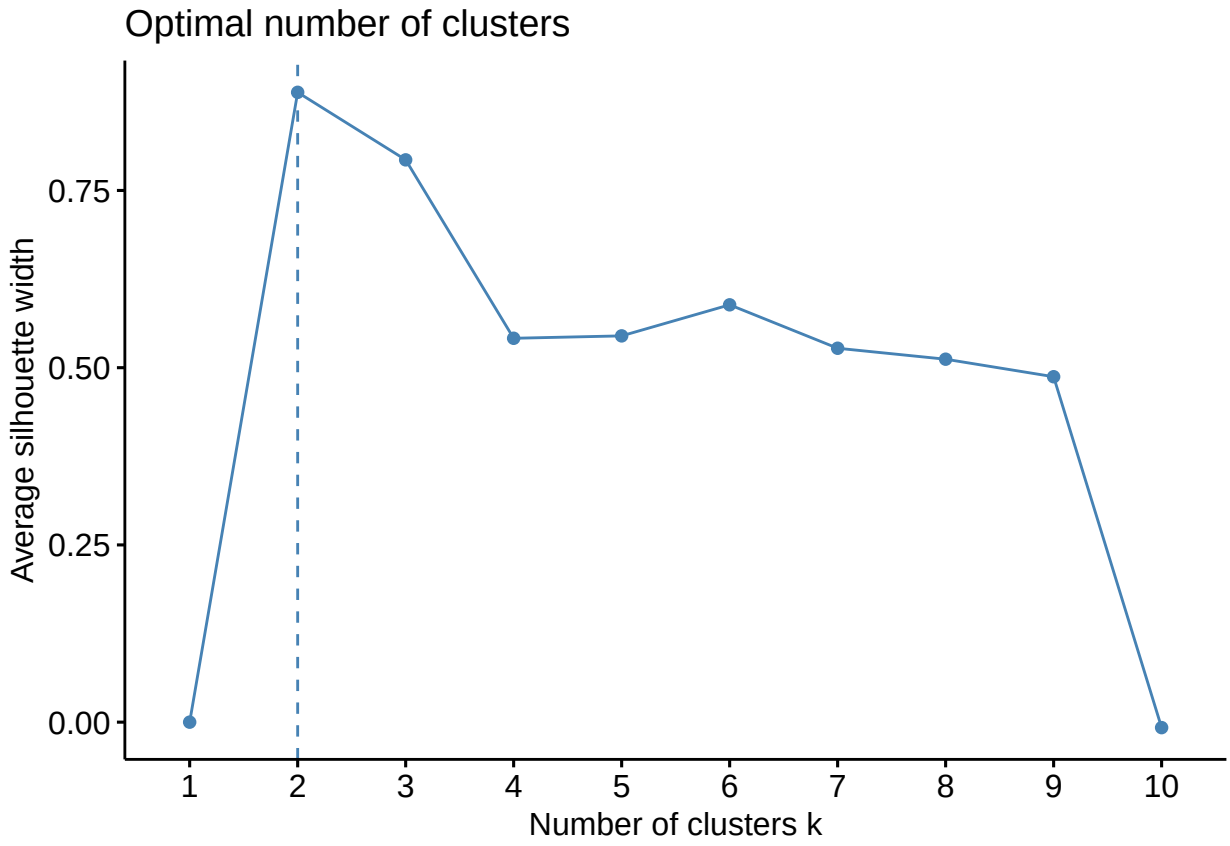
```
## [1] 1271
```

```
term_filter = term_count[,colSums(term_count) > 0] %>% scale()
ncol(term_filter)
```

```
## [1] 623
```

K-means cluster

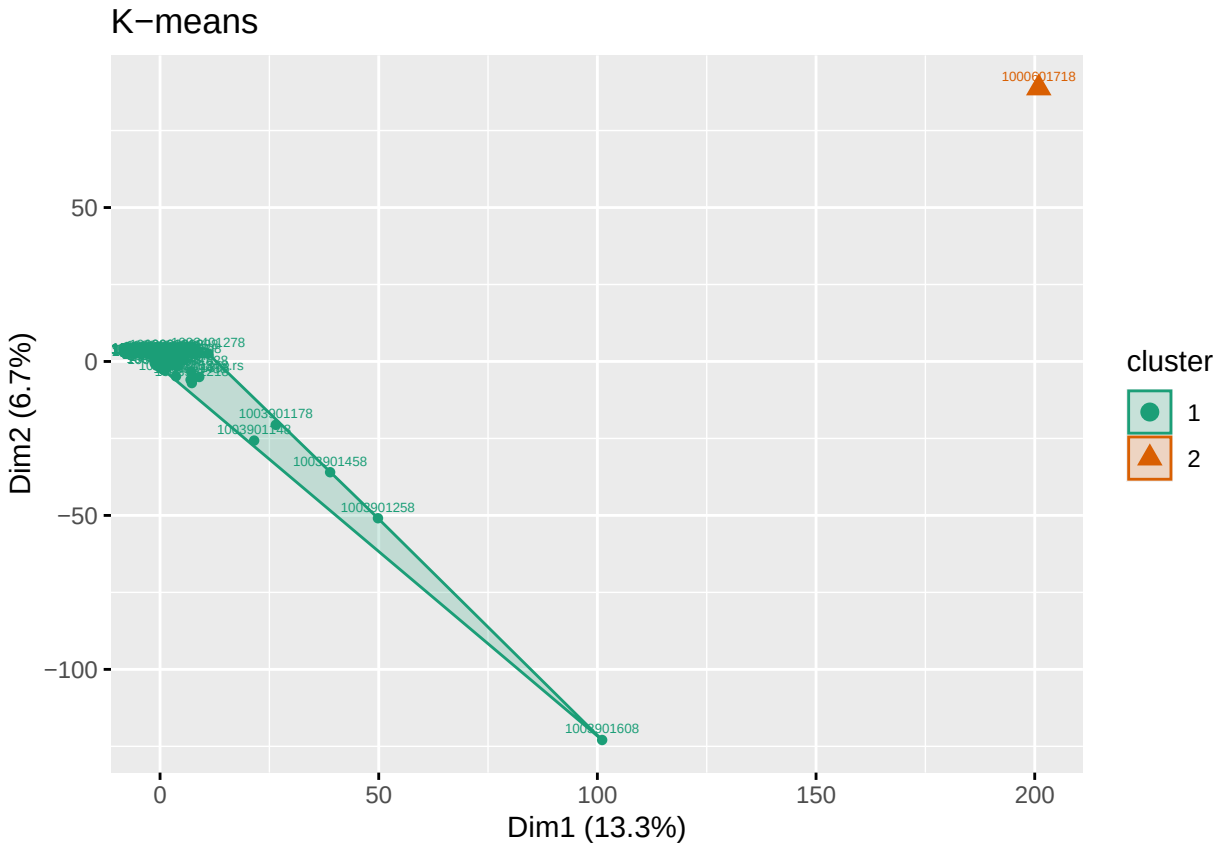
```
fviz_nbclust(term_filter,
             FUNcluster = kmeans,
             method = "silhouette")
```




```
set.seed(1)
km_term <- kmeans(term_filter, centers = 2, nstart = 20)
```

```
km_vis_term <- fviz_cluster(list(data = term_filter, cluster = km_term$cluster),
                            ellipse.type = "convex",
                            geom = c("point","text"),
                            labelsize = 5,
                            palette = "Dark2") + labs(title = "K-means")

km_vis_term
```




```
term[km_term$cluster == 2, 1:10]
```

```
##              SampleID Subject weeks  Race NumReads  Preg preterm CST 4330849
## 1000601718 1000601718   10006    71 White    10165 FALSE    Term   0       0
##            4400869
## 1000601718       0
```

Hierarchical clustering

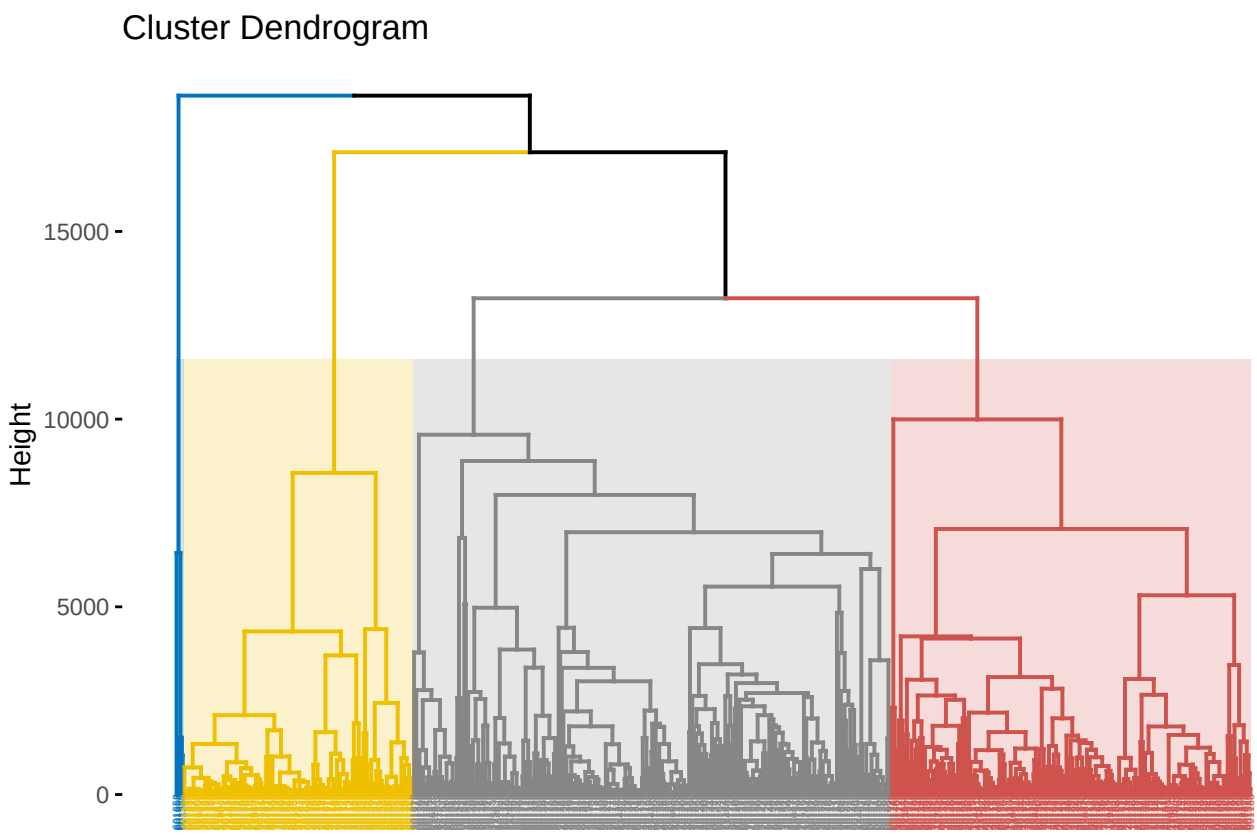
We can also apply hierarchical clustering on this data. Here we use the Euclidean distance and different types of linkage.

```
dat1 = term_count
hc.complete <- hclust(dist(dat1), method = "complete")

# distance.bray<-vegdist(dat1,method="bray",na.rm=TRUE)
# hc.bray<- hclust(distance.bray,method="complete")
```

The function `fviz_dend()` can be applied to visualize the dendrogram.

```
fviz_dend(hc.complete, k = 4,
          cex = 0.3,
          palette = "jco",
          color_labels_by_k = TRUE,
          rect = TRUE, rect_fill = TRUE, rect_border = "jco",
          labels_track_height = 2.5)
```

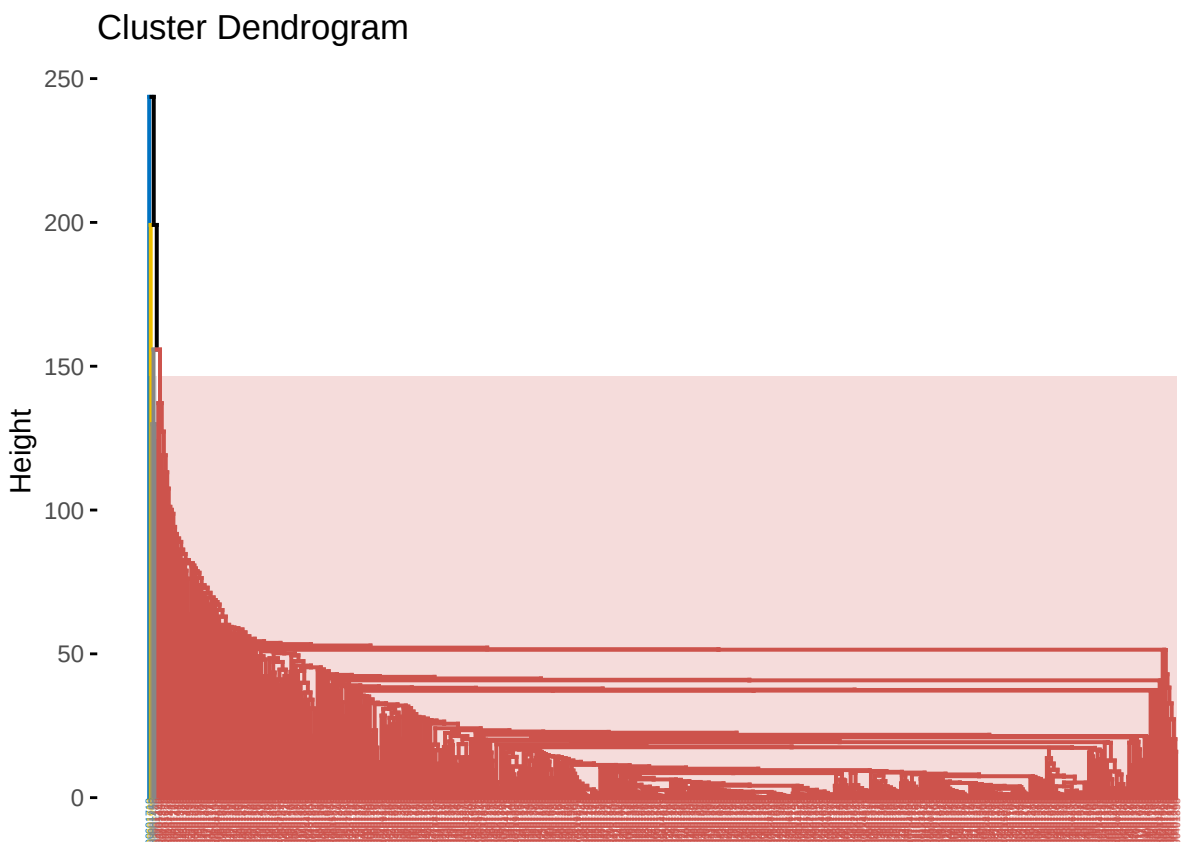



```
# Who are in the fourth cluster?
complete <- cutree(hc.complete, 4)
term[complete == 4,1:10]
```

```
##              SampleID Subject weeks  Race NumReads  Preg preterm CST 4330849
## 1000601528 1000601528   10006    53 White     7981 FALSE    Term   0       0
## 1000601608 1000601608   10006    60 White     8472 FALSE    Term   0       0
## 1000601658 1000601658   10006    66 White    13533 FALSE    Term   0       0
## 1000601718 1000601718   10006    71 White    10165 FALSE    Term   0       0
## 1004501308 1004501308   10045    31 White     7152  TRUE    Term   0       0
##            4400869
## 1000601528       0
## 1000601608       0
## 1000601658       0
## 1000601718       0
## 1004501308       0
```

After scaling and filtering

```
dat1 = term_filter
hc.complete <- hclust(dist(dat1), method = "complete")
fviz_dend(hc.complete, k = 4,
          cex = 0.3,
          palette = "jco",
          color_labels_by_k = TRUE,
          rect = TRUE, rect_fill = TRUE, rect_border = "jco",
          labels_track_height = 2.5)
```




```
complete <- cutree(hc.complete, 4)
preterm[complete == 4,1:10]
```

```
##    SampleID Subject weeks Race NumReads Preg preterm CST 4330849 4400869
## NA     <NA>    <NA>    NA <NA>       NA <NA>    <NA>  NA       NA       NA
```

Preterm

```
preterm_count =
  preterm %>%
  select(-SampleID,-Subject,-weeks,-Race,-NumReads,-Preg,-preterm,-CST)
ncol(preterm_count)
```

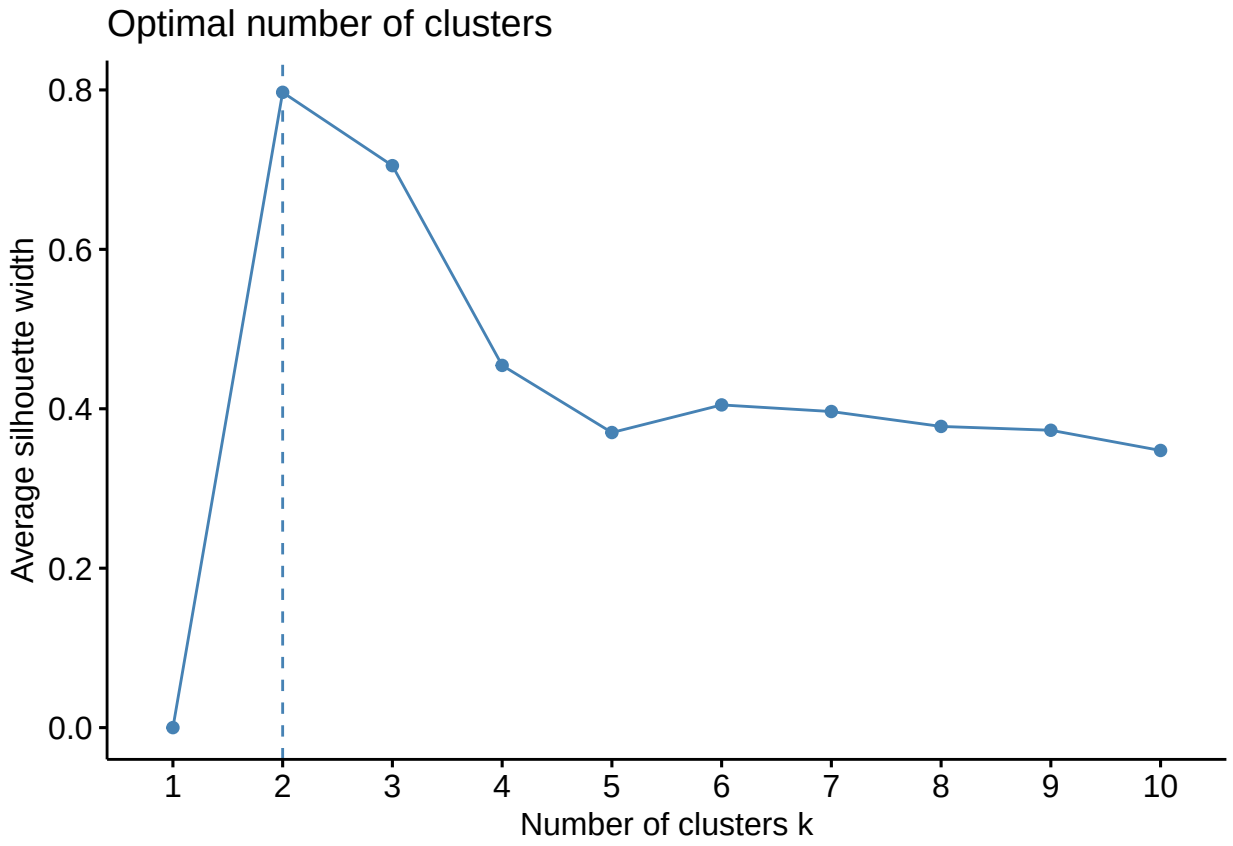
```
## [1] 1271
```

```
preterm_filter = preterm_count[,colSums(preterm_count) > 0] %>% scale()
ncol(preterm_filter)
```

```
## [1] 514
```

K-means cluster

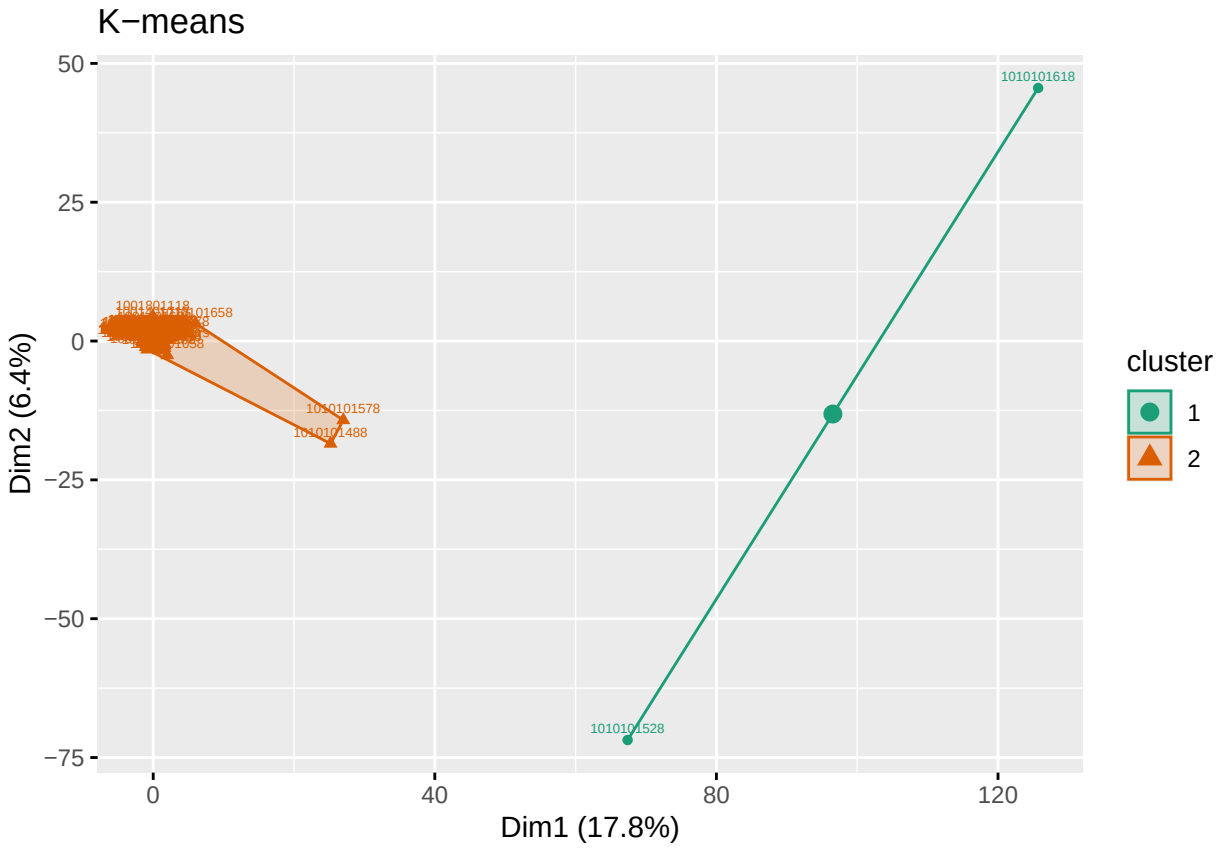
```
fviz_nbclust(preterm_filter,
             FUNcluster = kmeans,
             method = "silhouette")
```




```
set.seed(1)
km_preterm <- kmeans(preterm_filter, centers = 2, nstart = 20)
```

```
km_vis_preterm <- fviz_cluster(list(data = preterm_filter, cluster = km_preterm$cluster),
                               ellipse.type = "convex",
                               geom = c("point","text"),
                               labelsize = 5,
                               palette = "Dark2") + labs(title = "K-means")

km_vis_preterm
```




```
preterm[km_preterm$cluster == 1, 1:10]
```

```
##              SampleID Subject weeks  Race NumReads  Preg  preterm CST 4330849
## 1010101528 1010101528   10101    48 White     4078 FALSE Marginal   0       0
## 1010101618 1010101618   10101    58 White     9103 FALSE Marginal   0       0
##            4400869
## 1010101528       0
## 1010101618       0
```

Hierarchical clustering

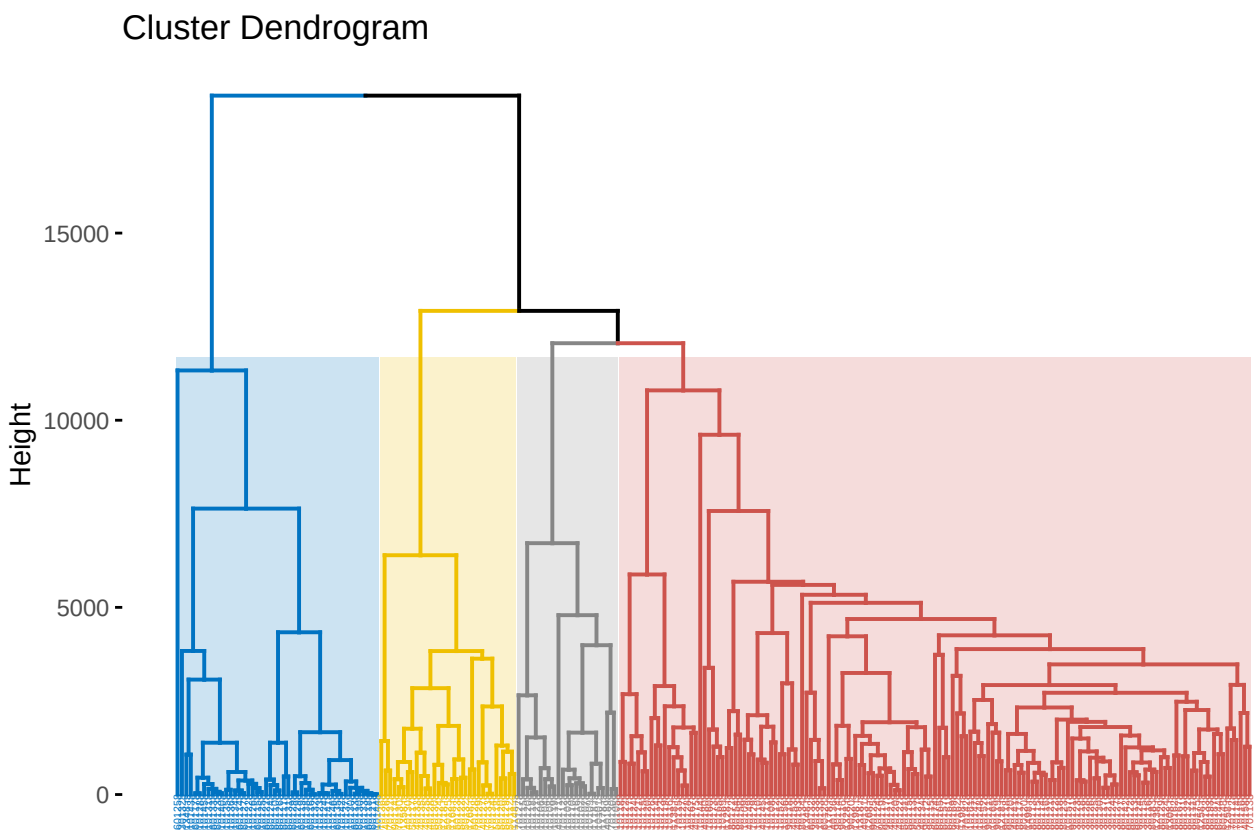
We can also apply hierarchical clustering on this data. Here we use the Euclidean distance and different types of linkage.

```
dat1 = preterm_count
hc.complete <- hclust(dist(dat1), method = "complete")

# distance.bray<-vegdist(dat1,method="bray",na.rm=TRUE)
# hc.bray<- hclust(distance.bray,method="complete")
```

The function `fviz_dend()` can be applied to visualize the dendrogram.

```
fviz_dend(hc.complete, k = 4,
          cex = 0.3,
          palette = "jco",
          color_labels_by_k = TRUE,
          rect = TRUE, rect_fill = TRUE, rect_border = "jco",
          labels_track_height = 2.5)
```



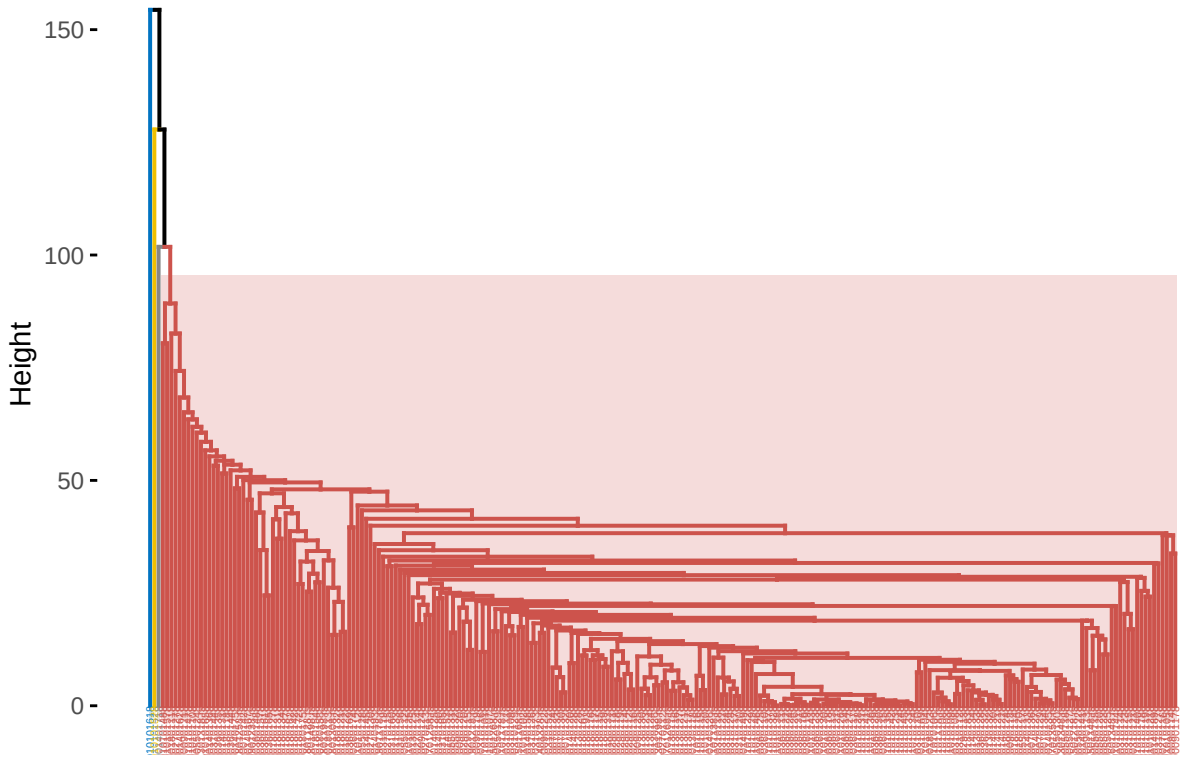

```
# Who are in the fourth cluster?
complete <- cutree(hc.complete, 4)
preterm[complete == 4,1:10]
```

```
##                   SampleID Subject weeks                  Race NumReads  Preg
## 1001401718       1001401718   10014    68     Asian-Unspecified    10311 FALSE
## 1001401898       1001401898   10014    86     Asian-Unspecified     5369 FALSE
## 1002701278       1002701278   10027    28 Other (Specify below)    11515  TRUE
## 1002701308       1002701308   10027    30 Other (Specify below)     3218  TRUE
## 1010101018       1010101018   10101    -7                 White     4580 FALSE
## 1010101028       1010101028   10101    -7                 White     6229 FALSE
## 1010101038       1010101038   10101    -5                 White     5697 FALSE
## 1010101048       1010101048   10101    -5                 White     6938 FALSE
## 1010101058       1010101058   10101    -3                 White     3182 FALSE
## 1010101068       1010101068   10101    -2                 White     7660 FALSE
## 1010101078       1010101078   10101    -2                 White     6293 FALSE
## 1010101088       1010101088   10101    -1                 White     7165 FALSE
## 1010101098       1010101098   10101     0                 White     5386 FALSE
## 1010101108       1010101108   10101     1                 White     6356  TRUE
## 1010101118.rs 1010101118.rs   10101     1                 White     5389  TRUE
## 1010101128       1010101128   10101     2                 White     6854  TRUE
## 1010101158       1010101158   10101     6                 White     8581  TRUE
## 1010101168       1010101168   10101     7                 White     6283  TRUE
## 1010101178.rs 1010101178.rs   10101     7                 White     4826  TRUE
## 1010101188       1010101188   10101     8                 White     8354  TRUE
## 1010101198       1010101198   10101    10                 White     7579  TRUE
## 1010101208       1010101208   10101    11                 White     8262  TRUE
## 1010101218       1010101218   10101    11                 White     8370  TRUE
```

```
##                preterm CST 4330849 4400869
## 1001401718    Marginal   0       0       0
## 1001401898    Marginal   0       0       0
## 1002701278    Marginal   0       0       0
## 1002701308    Marginal   0       0       0
## 1010101018    Marginal   0       0       0
## 1010101028    Marginal   0       0       0
## 1010101038    Marginal   0       0       0
## 1010101048    Marginal   0       0       0
## 1010101058    Marginal   0       0       0
## 1010101068    Marginal   0       0       0
## 1010101078    Marginal   0       0       0
## 1010101088    Marginal   0       0       0
## 1010101098    Marginal   0       0       0
## 1010101108    Marginal   0       0       0
## 1010101118.rs Marginal   0       0       0
## 1010101128    Marginal   0       0       0
## 1010101158    Marginal   0       0       0
## 1010101168    Marginal   0       0       0
## 1010101178.rs Marginal   0       0       0
## 1010101188    Marginal   0       0       0
## 1010101198    Marginal   0       0       0
## 1010101208    Marginal   0       0       0
## 1010101218    Marginal   0       0       0
```

After scaling and filtering

```
dat1 = preterm_filter
hc.complete <- hclust(dist(dat1), method = "complete")
fviz_dend(hc.complete, k = 4,
          cex = 0.3,
          palette = "jco",
          color_labels_by_k = TRUE,
          rect = TRUE, rect_fill = TRUE, rect_border = "jco",
          labels_track_height = 2.5)
```




```
complete <- cutree(hc.complete, 4)
preterm[complete == 4,1:10]
```

```
##              SampleID Subject weeks  Race NumReads  Preg  preterm CST 4330849
## 1010101618 1010101618   10101    58 White     9103 FALSE Marginal   0       0
##            4400869
## 1010101618       0
```

cluster using phyloseq

Weijia Xiong

6/30/2020

Load data

```
otu_file <- "data/PregnancyClosed15.RData"
load(otu_file)
```

Transform the data (proportions):

```
site <- "Vaginal_Swab"
ps <- PSPreg[[site]]
tt <- data.frame(tax_table(ps))
ps <- transform_sample_counts(ps, function(OTU) OTU/sum(OTU))
```

We are not doing differential abundance analysis here, so the proportion transformation is used for exploratory analyses only.

```
summary(sample_data(ps)$Outcome)
```

```
##    Marginal     Preterm        Term VeryPreterm
##          83          64         571          43
```

```
ps_preterm = subset_samples(ps, Outcome %in% c("Preterm","VeryPreterm"))
ps_term = subset_samples(ps, Outcome %in% c("Term","Marginal"))
```

Term data cluster

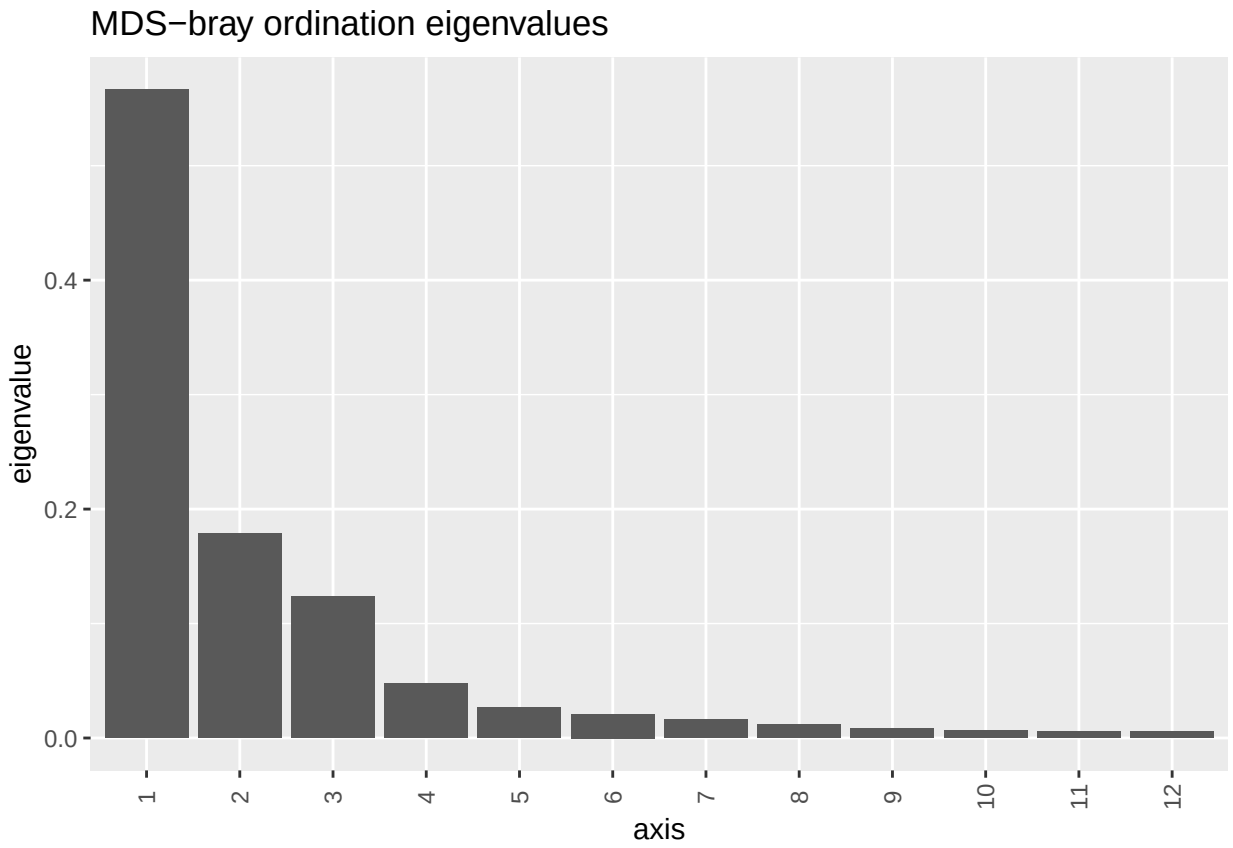
The vaginal community is dominated by closely related, but functionally distinct, Lactobacillus species. Therefore it is better to use a non-phylogenetically aware distance measure so as to be able to separate these species. Start with an MDS (or PCoA) ordination:

```
braydist <- phyloseq::distance(ps_term, method="bray")
ord = ordinate(ps, method = "MDS", distance = braydist)
## based in some fashion on the abundance table ultimately stored as a contingency matrix (otu_table-cl

# MDS: Performs principal coordinate analysis (also called principle coordinate decomposition, multidim

# Need a distance matrix, here use bray-curtis disctance

plot_scree(ord) + xlim(as.character(seq(1,12))) + ggtitle("MDS-bray ordination eigenvalues")
```




```
# p1 = plot_ordination(ps, ord, type="taxa", color="Phylum", title="taxa")
# print(p1)

evs <- ord$value$Eigenvalues
print(evs[1:20])
```

```
##  [1] 116.6689774  36.8329781  25.4839268   9.8136771   5.4647095   4.3200964
##  [7]   3.3399353   2.4345698   1.6683111   1.3444952   1.2280786   1.2082681
## [13]   0.8565684   0.7421970   0.7047971   0.6730503   0.6214064   0.5451675
## [19]   0.5306053   0.5036866
```

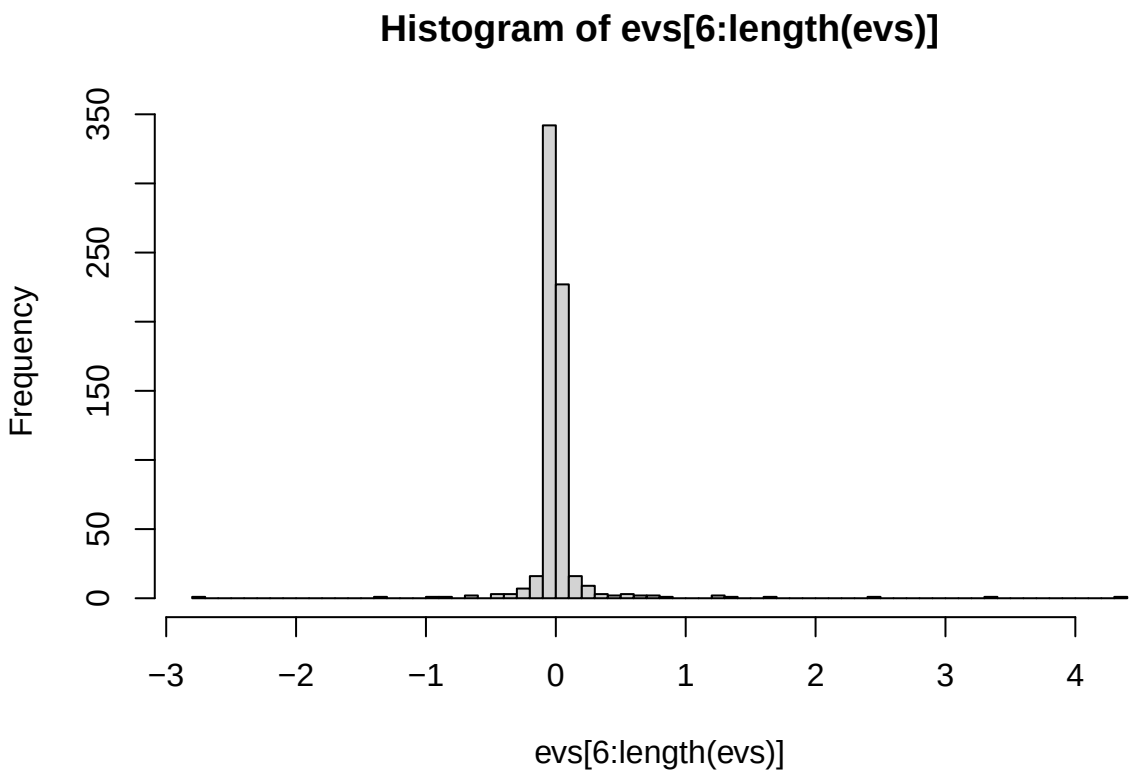
```
print(tail(evs))
```

```
## [1] -0.6061663 -0.6389676 -0.8712937 -0.9785011 -1.3789373 -2.7454736
```

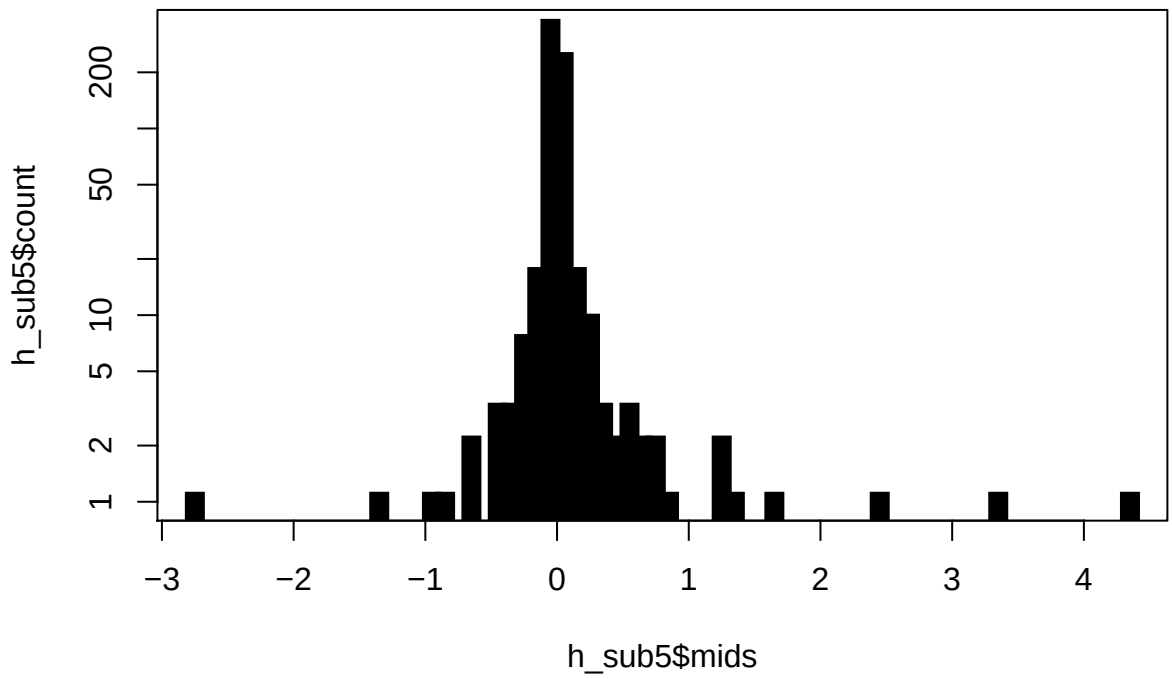
Denoise distance matrix

We would like to clean some of the noise from the data by restricting this to the truly significant dimensions. The top 5 eigenvalues are clearly very significant, but let's keep all the positive eigenvalues that clearly exceed the magnitude of the smallest negative eigenvalues:

```
h_sub5 <- hist(evs[6:length(evs)], 100)
```




```
plot(h_sub5$mids, h_sub5$count, log="y", type='h', lwd=10, lend=2)
```

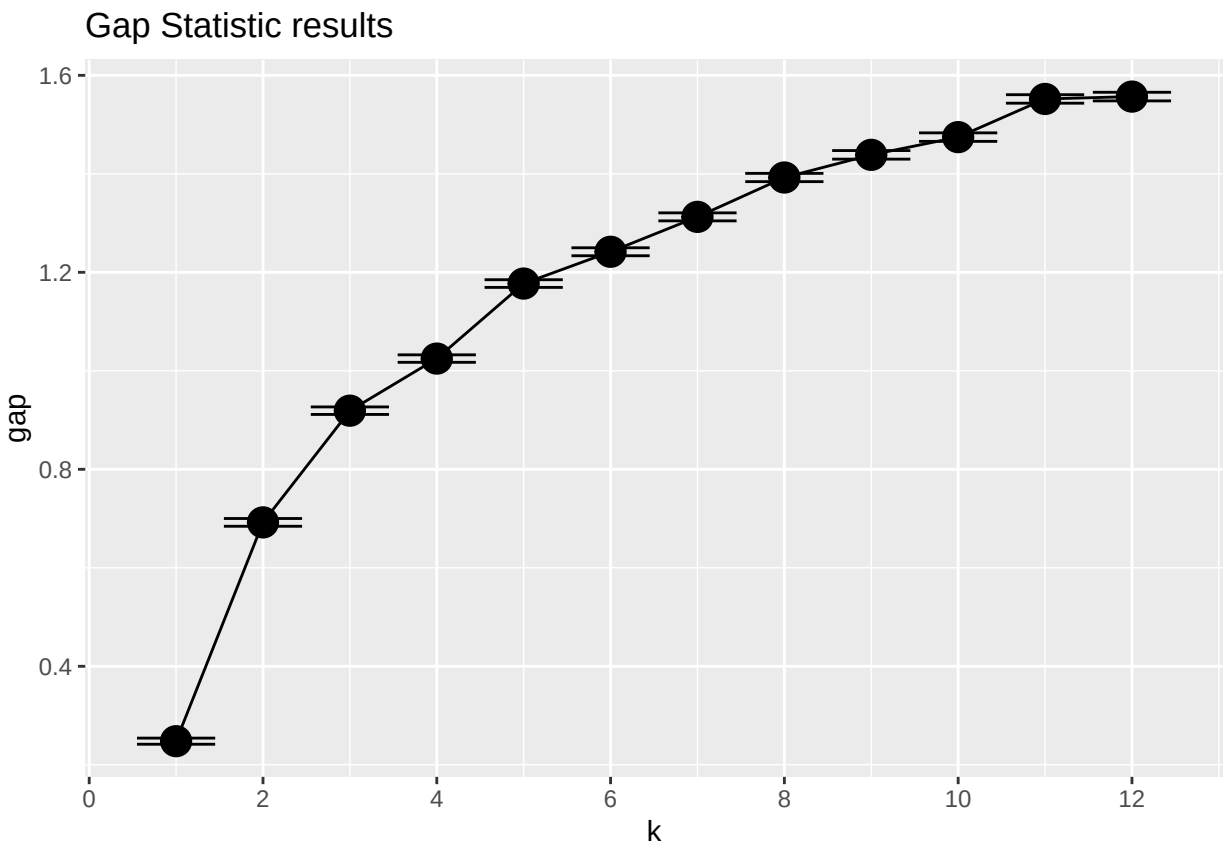


Looks like eigenvalues 6 and 7 still stand out, so we'll go with 7 MDS dimensions.

Determine number of clusters

We will use the gap statistic to indicate the number of clusters in this data:

```
NDIM <- 7
x <- ord$vectors[,1:NDIM]  # rows=sample, cols=MDS axes, entries = value
pamPCoA = function(x, k) {
    list(cluster = pam(x[,1:2], k, cluster.only = TRUE))
}
gs = clusGap(x, FUN = pamPCoA, K.max = 12, B = 50)
plot_clusgap(gs) + scale_x_continuous(breaks=c(seq(0, 12, 2)))
```



The gap statistic strongly suggests at least three clusters, but makes another big jump at K=5 before the slope gets a lot smaller. So, K=5 it is.

Cluster into CSTs

Perform PAM 5-fold clusters:

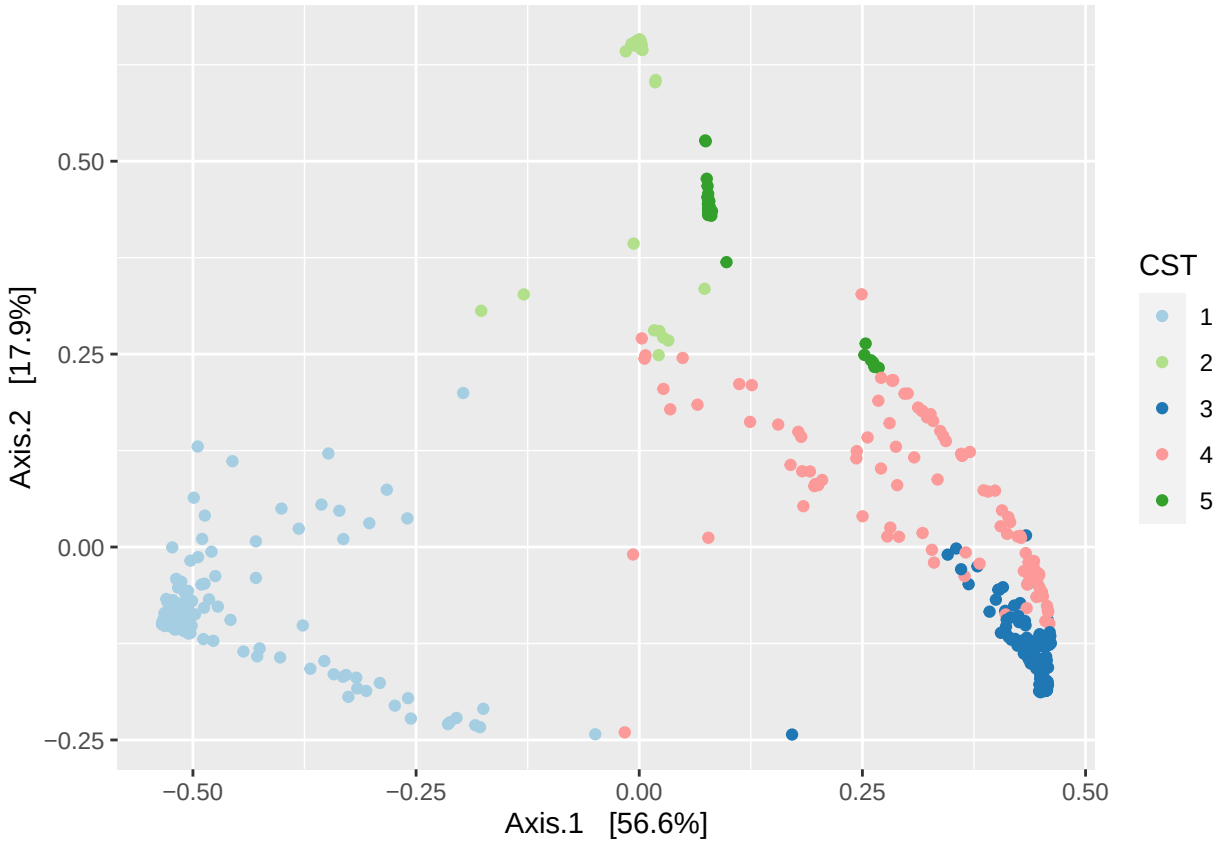
```
K <- 5
x <- ord$vectors[,1:NDIM]
clust <- as.factor(pam(x, k=K, cluster.only=T))
# SWAPPING THE ASSIGNMENT OF 2 AND 3 TO MATCH RAVEL CST ENUMERATION
clust[clust==2] <- NA
clust[clust==3] <- 2
clust[is.na(clust)] <- 3
sample_data(ps_term)$CST <- clust
CSTs <- as.character(seq(K))
```

Evaluate clustering

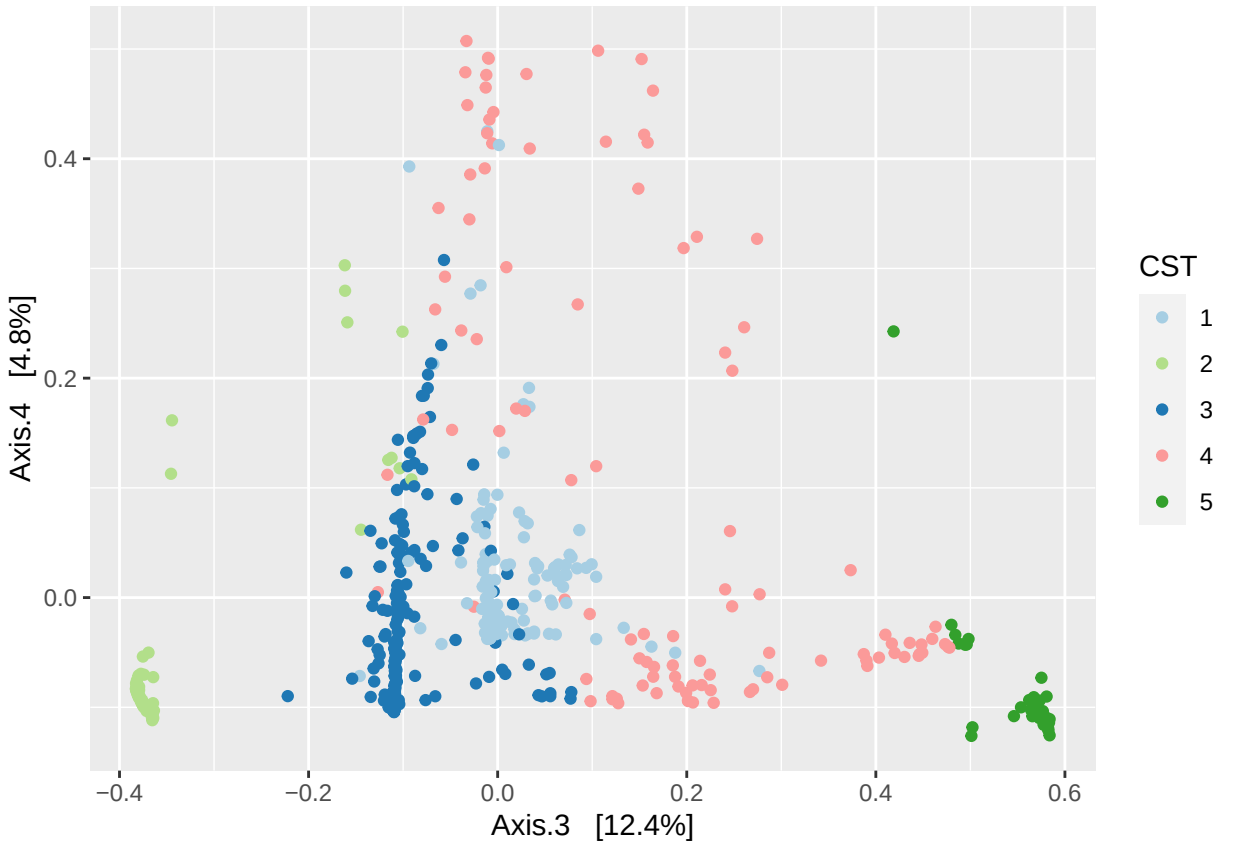
Inspect the results in MDS and NMDS ordinations:

```
CSTColors <- brewer.pal(6,"Paired")[c(1,3,2,5,4,6)] # Length 6 for consistency with pre-revision CST+ c
names(CSTColors) <- CSTs
CSTColorScale <- scale_colour_manual(name = "CST", values = CSTColors[1:5])
CSTFillScale <- scale_fill_manual(name = "CST", values = CSTColors[1:5])
# grid.arrange(plot_ordination(ps, ord, color="CST") + CSTColorScale,
#              plot_ordination(ps, ord, axes=c(3,4), color="CST") + CSTColorScale, main="Ordination by

plot_ordination(ps_term, ord, color="CST") + CSTColorScale
```




```
plot_ordination(ps_term, ord, axes=c(3,4), color="CST")  + CSTColorScale
```




```
nmds = ordinate(ps_term, method="NMDS", distance=braydist)
```

```
## Run 0 stress 0.1430354
## Run 1 stress 0.1919882
## Run 2 stress 0.1922407
## Run 3 stress 0.1946097
## Run 4 stress 0.1865393
## Run 5 stress 0.1918076
## Run 6 stress 0.1857592
## Run 7 stress 0.193173
## Run 8 stress 0.1759438
## Run 9 stress 0.1932515
## Run 10 stress 0.1952003
## Run 11 stress 0.1758556
## Run 12 stress 0.1933778
## Run 13 stress 0.1902203
## Run 14 stress 0.1808392
## Run 15 stress 0.1846196
## Run 16 stress 0.1630381
## Run 17 stress 0.1927165
## Run 18 stress 0.189357
## Run 19 stress 0.1940379
## Run 20 stress 0.1837754
## *** No convergence -- monoMDS stopping criteria:
##     19: stress ratio > sratmax
##      1: scale factor of the gradient < sfgrmin
```

```
plot_NMDS_bray_by_cluster = plot_ordination(ps,nmds, color="CST") + CSTColorScale + ggtitle("NMDS -- bra
```

```
sample_data(ps_term)$clust <- clust
samdf <- data.frame(sample_data(ps_term))
table(samdf$clust)
```

```
##
##   1   2   3   4   5
## 256  57 202 105  34
```

Cluster for whole data via Gower distance

Weijia Xiong

6/30/2020

```
load("data/DiGiulio.RData")
otu_data = as.data.frame(DiGiulio$OTU)  # 927 samples, 1271 OTU
taxonomy = DiGiulio$Taxonomy  # 1271
sampledata = DiGiulio$SampleData  #  927 samples, other covariates
```

```
otu_data_all=
  cbind(sampledata, otu_data) %>%
  mutate(
    Preg = as.factor(Preg),
    Subject = as.factor(Subject)
  ) %>%
  na.omit()
```

Term data

```
term =
  otu_data_all %>%
  filter(preterm == "Term")

term_data =
  term %>%
  dplyr::select(-SampleID,-Subject)
```

Gower distance for mixed variables

```
gower_dist <- daisy(term_data, metric = "gower")
gower_mat <- as.matrix(gower_dist)
```

```
#' Print most similar
term[which(gower_mat == min(gower_mat[gower_mat != min(gower_mat)]), arr.ind = TRUE)[1, ], 1:10]
```

```
##      SampleID Subject weeks  Race NumReads Preg preterm CST 4330849 4400869
## 27 1000601208   10006    20 White     2193 TRUE    Term   0       0       0
## 26 1000601198   10006    19 White     2385 TRUE    Term   0       0       0
```

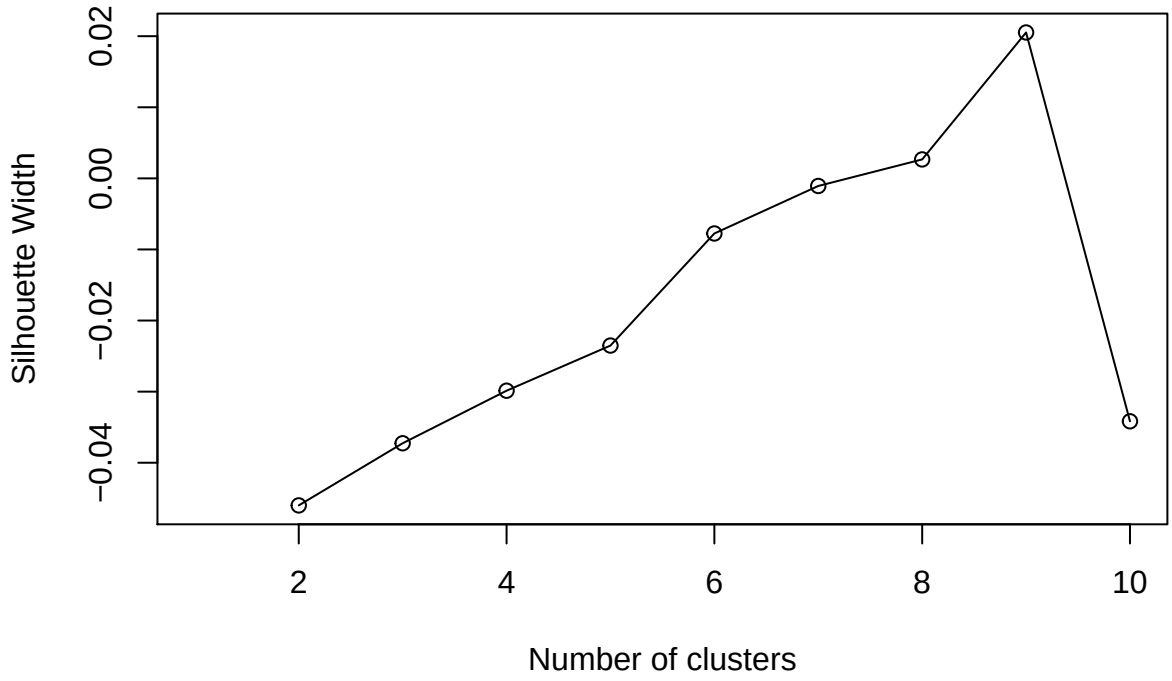
```
#' Print most dissimilar
term[which(gower_mat == max(gower_mat[gower_mat != max(gower_mat)]), arr.ind = TRUE)[1, ], 1:10]
```

```
##          SampleID Subject weeks  Race NumReads  Preg preterm CST 4330849
## 458 1004301328.rs   10043    32 White     5708  TRUE    Term   1       0
## 51     1000601718   10006    71 White    10165 FALSE    Term   0       0
##     4400869
```

```
## 458       0
## 51        0
```

Calculate silhouette width for many k using PAM

```
## Cluster
sil_width <- c(NA)
for(i in 2:10){
  pam_fit <- pam(gower_dist, diss = TRUE, k = i)
  sil_width[i] <- pam_fit$silinfo$avg.width
}
plot(1:10, sil_width,
     xlab = "Number of clusters",
     ylab = "Silhouette Width")
lines(1:10, sil_width)
```




```
k <- 9
pam_fit <- pam(gower_dist, diss = TRUE, k)
pam_results <- term_data %>%
  mutate(cluster = pam_fit$clustering) %>%
  group_by(cluster) %>%
  do(the_summary = summary(.))
result = pam_results$the_summary
term[pam_fit$medoids, 1:10]
```

```
##       SampleID Subject weeks                  Race NumReads  Preg preterm CST
## 137 1002101308   10021    30                 White     3408  TRUE    Term   0
## 159 1002201268   10022    27                 White     5668  TRUE    Term   0
## 534 1004501618   10045    61                 White     3820 FALSE    Term   0
## 51  1000601718   10006    71                 White    10165 FALSE    Term   0
## 424 1004001338   10040    33                Indian     4335  TRUE    Term   0
## 630 1900501178   19005    18 Other (Specify below)     6134  TRUE    Term   0
## 389 1003901258   10039    26                 White     8045  TRUE    Term   0
```

```
## 404 1003901458   10039    46                 White     2218 FALSE    Term   0
## 408 1003901608   10039    61                 White     5415 FALSE    Term   0
##     4330849 4400869
## 137       0       0
## 159       0       0
## 534       0       0
## 51        0       0
## 424       0       0
## 630       0       0
## 389       0       0
## 404       0       0
## 408       0       0
```

```
tsne_obj <- Rtsne(gower_dist, is_distance = TRUE)
tsne_data <- tsne_obj$Y %>%
  data.frame() %>%
  setNames(c("X", "Y")) %>%
  mutate(cluster = factor(pam_fit$clustering))
ggplot(aes(x = X, y = Y), data = tsne_data) +
  geom_point(aes(color = cluster))
```

Preterm data

```
preterm =
  otu_data_all %>%
  filter(preterm != "Term")
```

```r
preterm_data =
  preterm %>%
  dplyr::select(-SampleID,-Subject)
```

Gower distance for mixed variables

```r
gower_dist <- daisy(preterm_data, metric = "gower")
gower_mat <- as.matrix(gower_dist)
```

```r
#' Print most similar
preterm[which(gower_mat == min(gower_mat[gower_mat != min(gower_mat)]), arr.ind = TRUE)[1, ], 1:10]
```

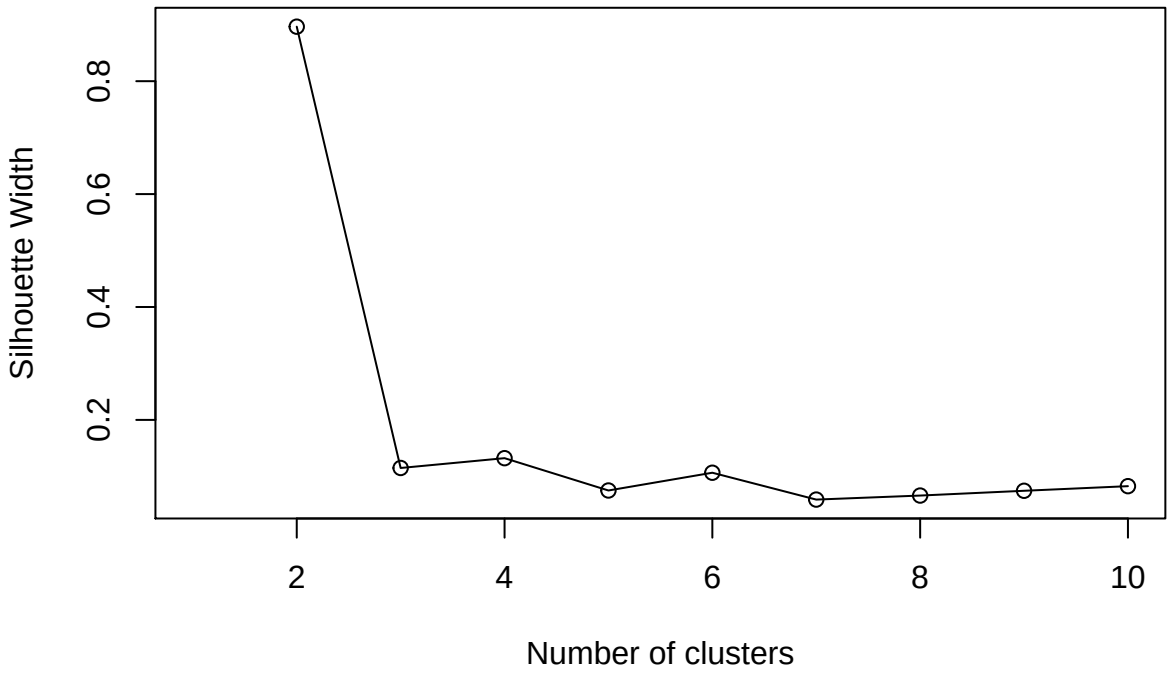
```
##       SampleID Subject weeks  Race NumReads Preg  preterm CST 4330849 4400869
## 195 1010101248   10101    14 White     8382 TRUE Marginal   0       0       0
## 194 1010101238   10101    14 White     8348 TRUE Marginal   0       0       0
```

```r
#' Print most dissimilar
preterm[which(gower_mat == max(gower_mat[gower_mat != max(gower_mat)]), arr.ind = TRUE)[1, ], 1:10]
```

```
##       SampleID Subject weeks            Race NumReads  Preg  preterm CST
## 220 1010101618   10101    58           White     9103 FALSE Marginal   0
## 45  1001801118   10018    12 American Indian     3599  TRUE  Preterm   1
##     4330849 4400869
## 220       0       0
## 45        0       0
```

Calculate silhouette width for many k using PAM

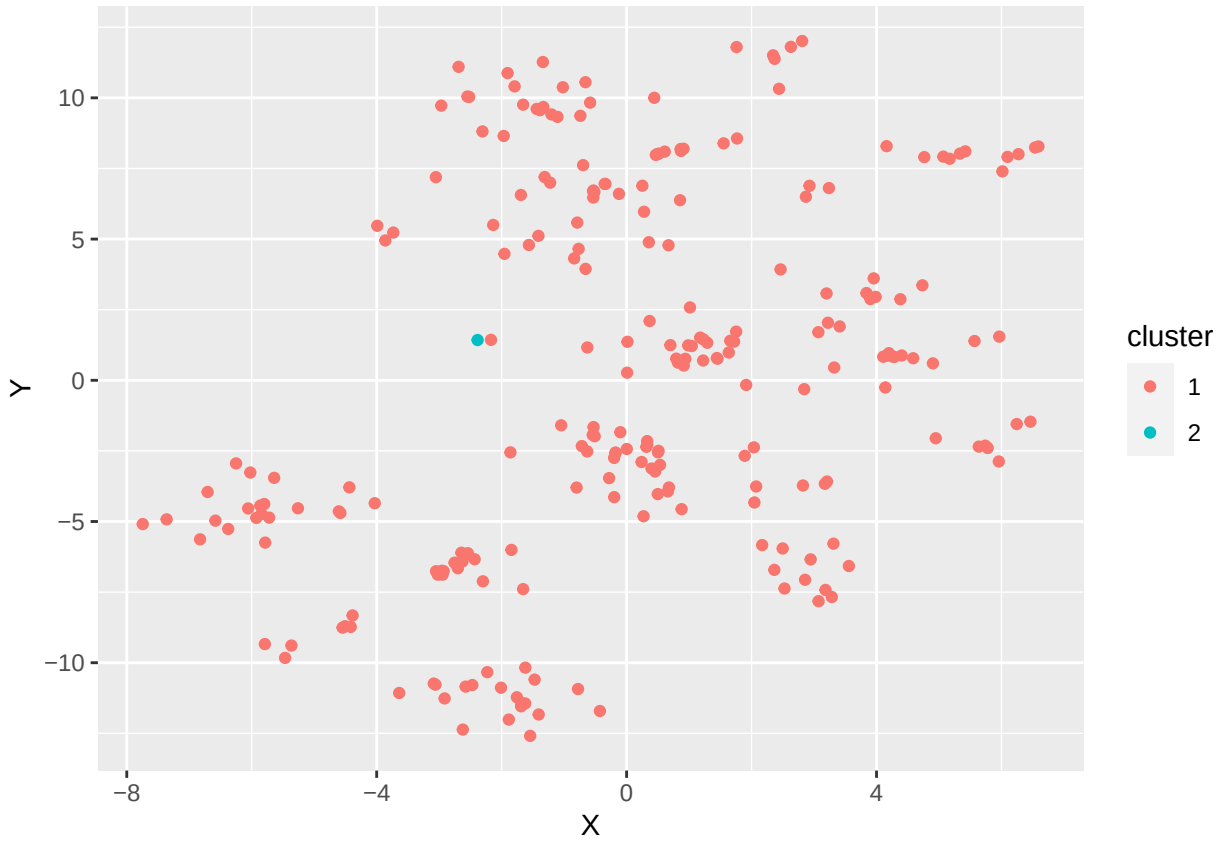
```r
## Cluster
sil_width <- c(NA)
for(i in 2:10){
  pam_fit <- pam(gower_dist, diss = TRUE, k = i)
  sil_width[i] <- pam_fit$silinfo$avg.width
}
plot(1:10, sil_width,
     xlab = "Number of clusters",
     ylab = "Silhouette Width")
lines(1:10, sil_width)
```




```
k <- 2
pam_fit <- pam(gower_dist, diss = TRUE, k)
pam_results <- preterm_data %>%
  mutate(cluster = pam_fit$clustering) %>%
  group_by(cluster) %>%
  do(the_summary = summary(.))
result = pam_results$the_summary
term[pam_fit$medoids, 1:10]
```

```
##       SampleID Subject weeks  Race NumReads  Preg preterm CST 4330849 4400869
## 212 1002301618   10023    62 White     7341 FALSE    Term   0       0       0
## 220 1002401138   10024    14 White     5934  TRUE    Term   0       0       0
```

```
tsne_obj <- Rtsne(gower_dist, is_distance = TRUE)
tsne_data <- tsne_obj$Y %>%
  data.frame() %>%
  setNames(c("X", "Y")) %>%
  mutate(cluster = factor(pam_fit$clustering))
ggplot(aes(x = X, y = Y), data = tsne_data) +
  geom_point(aes(color = cluster))
```


10
5
0
−5
−10
Y
−8
−4
0
4
X
cluster
1
2