LDA_flu_test

LDA

https://tgmstat.wordpress.com/2014/01/15/computing-and-visualizing-lda-in-r/ https://gist.github.com/thigm85/8424654

```
library(MASS)
library(ggplot2)
library(scales)
```

```
cpal <- c('#e41a1c','#377eb8','#4daf4a','#984ea3','#ff7f00','#e6e600',
          '#a65628','#f781bf','#4d4d4d')
```

flu virus all test set 1

```
# label and seq data
uniflna_test1 <-
  read.csv("/home/brian/Documents/data_mining/flu_project/uniflna_test1.csv", na.strings="")

# comparisons from sourmash
uniflna_test1_cmp <-
  read.csv("/home/brian/Documents/data_mining/flu_project/uniflna_test1_cmp.csv")

# Label the rows
rownames(uniflna_test1_cmp) <- colnames(uniflna_test1_cmp)

# add gene column
uniflna_test1_wgs <- uniflna_test1_cmp
uniflna_test1_wgs$gene <- uniflna_test1$gene
uniflna_test1_wgs$segment <- factor(uniflna_test1$segment)

# Transform for plotting
uniflna_test1_cmp_mat <- as.matrix(uniflna_test1_cmp)
```

```
lda <- lda(gene ~ .,
           uniflna_test1_wgs[,-1002])
```

```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

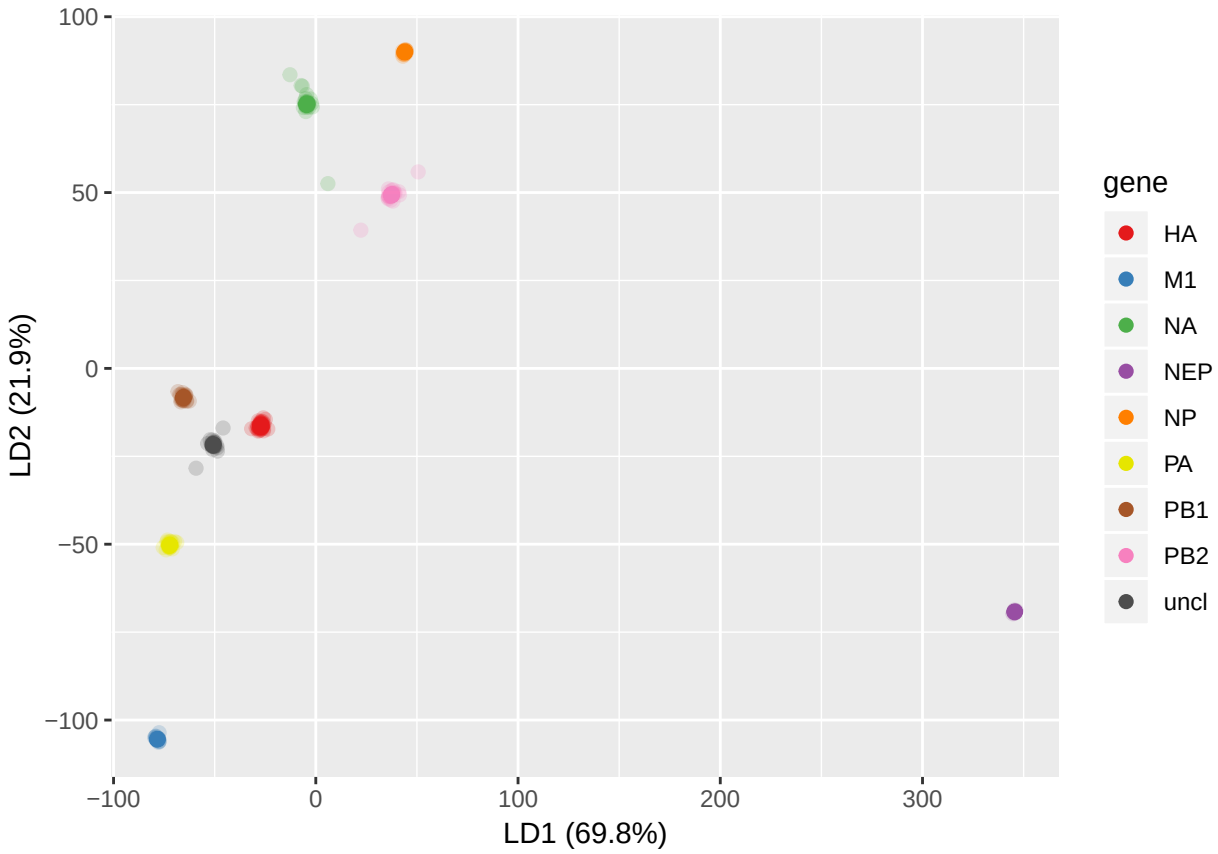
```
prop.lda = lda$svd^2/sum(lda$svd^2)

plda <- predict(object = lda,
                newdata = uniflna_test1_wgs)
```

```
dataset = data.frame(gene = uniflna_test1_wgs[,1001],
                     lda = plda$x)
```

```
p1 <- ggplot(dataset) + geom_point(aes(lda.LD1, lda.LD2, colour = gene), size=2, alpha=0.2) +
  #theme_minimal() +
  labs(x = paste("LD1 (", percent(prop.lda[1]), ")", sep=""),
       y = paste("LD2 (", percent(prop.lda[2]), ")", sep="")) +
  scale_color_manual(values=cpal) +
  guides(colour = guide_legend(override.aes = list(alpha = 1)))
# https://stackoverflow.com/questions/5290003/how-to-set-legend-alpha-with-ggplot2
```

```
# ggsave("gene_LDA_plot.png", plot = p1, width = 7, height = 4, dpi = 200)
```

```
p1
```




```
lda2 <- lda(segment ~ .,
            uniflna_test1_wgs[,-1001])
```

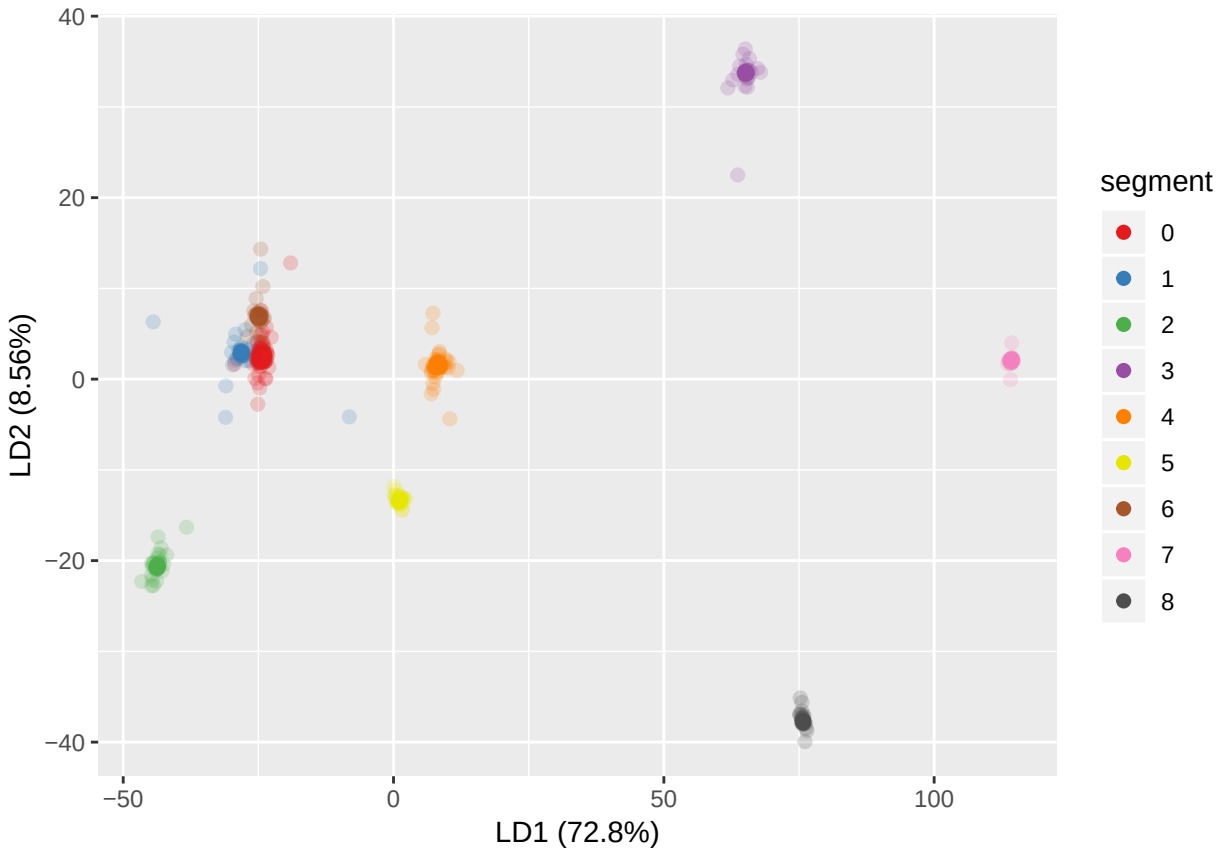
```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

```
prop.lda2 = lda2$svd^2/sum(lda2$svd^2)

plda2 <- predict(object = lda2,
                 newdata = uniflna_test1_wgs)
```

segment labels

```
dataset2 = data.frame(segment = uniflna_test1_wgs[,1002],
                      lda2 = plda2$x)
```

```
p2 <- ggplot(dataset2) + geom_point(aes(lda2.LD1, lda2.LD2, colour = segment), size=2, alpha=0.2) +
  #theme_minimal() +
  labs(x = paste("LD1 (", percent(prop.lda2[1]), ")", sep=""),
       y = paste("LD2 (", percent(prop.lda2[2]), ")", sep="")) +
  scale_color_manual(values=cpal) +
  guides(colour = guide_legend(override.aes = list(alpha = 1)))
p2
```



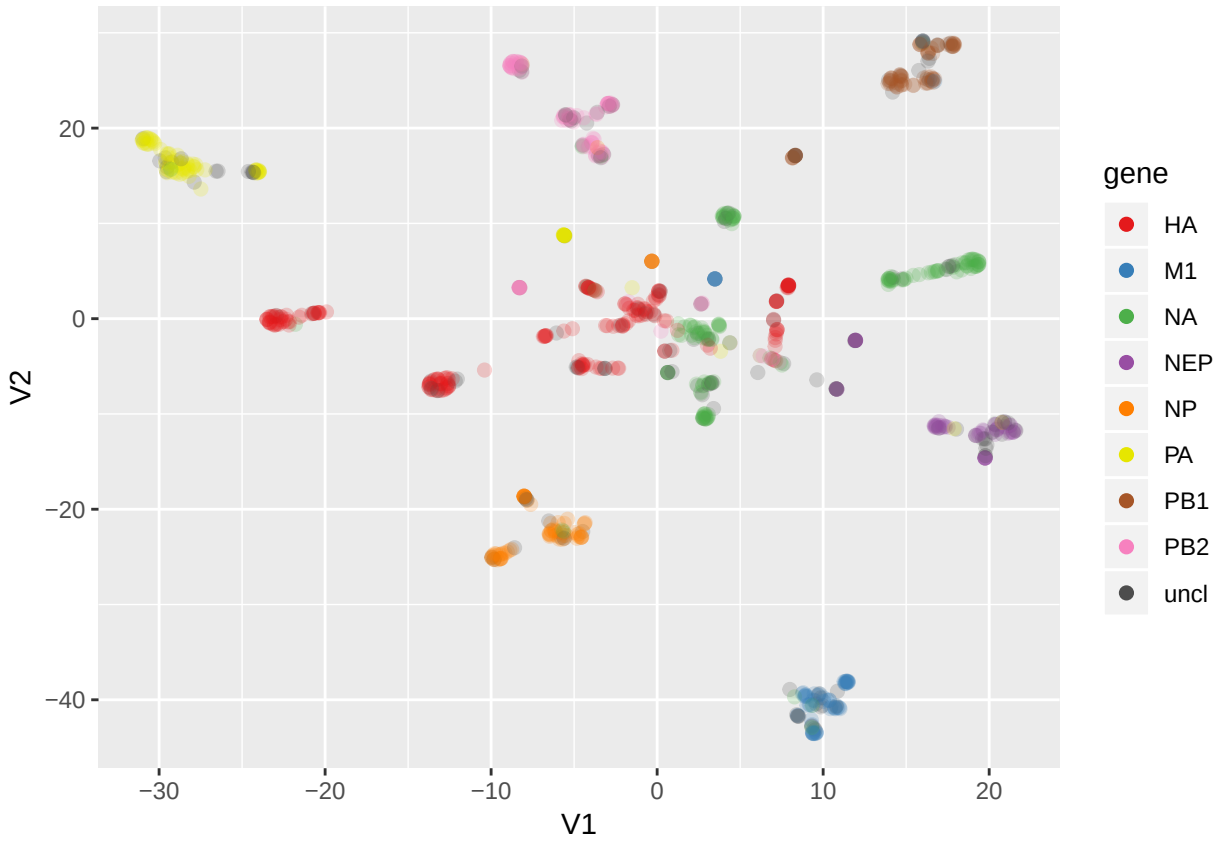
segment labels not as reliable.

Tsne for fun

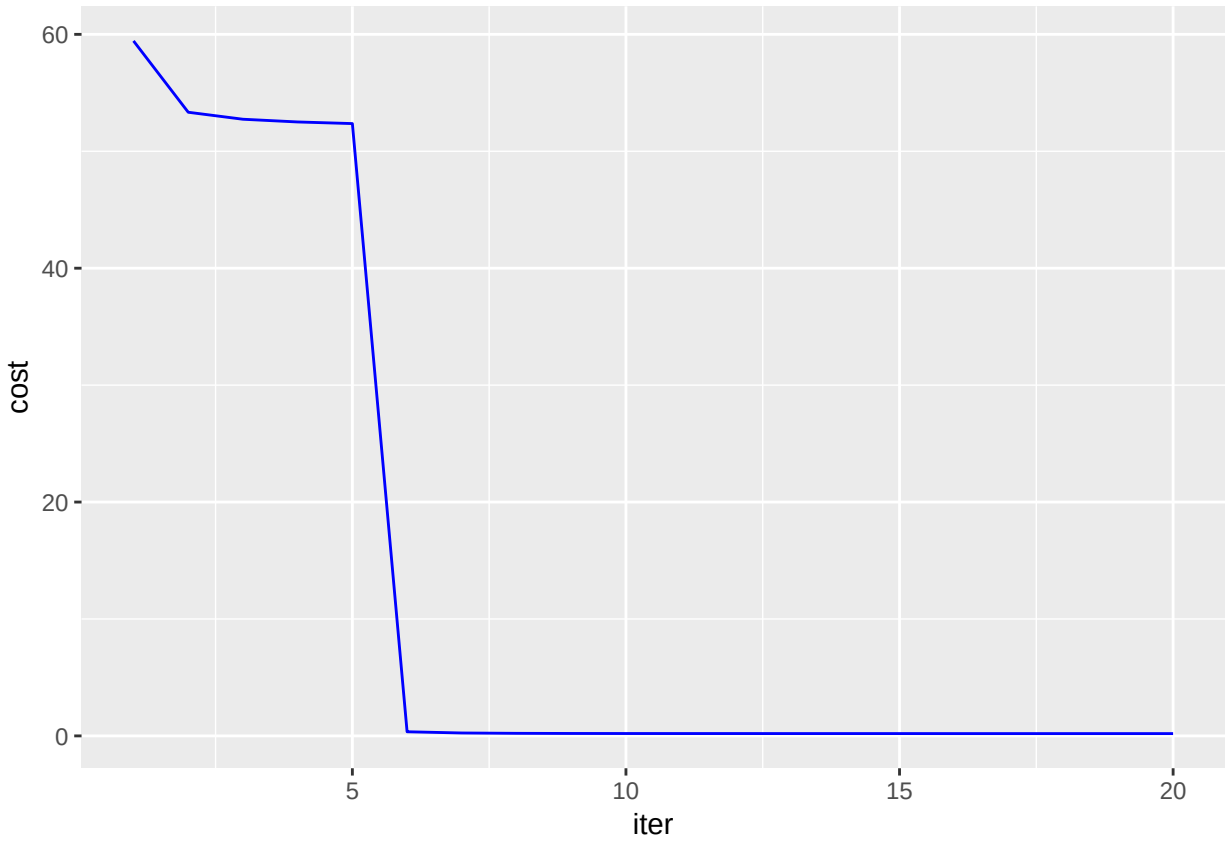
```
library(Rtsne)
```

```
tsne_model <- Rtsne(uniflna_test1_cmp_mat, check_duplicates=FALSE, pca=TRUE, perplexity=50, theta=0.25,
d_tsne = as.data.frame(tsne_model$Y)
d_tsne$gene <- uniflna_test1_wgs[,1001]
#plot(d_tsne$V1, d_tsne$V2)
```

```
ggplot(d_tsne, aes(V1, V2, colour = gene)) + geom_point(size=2, alpha=0.2) +
  scale_color_manual(values=cpal) +
  guides(colour = guide_legend(override.aes = list(alpha = 1)))
```




```
itercosts <- data.frame(iter = 1:20, cost = tsne_model$itercosts)

ggplot(itercosts, aes(iter,cost)) + geom_line(color='blue')
```



second sample with 2500 sequences, k=7 uniflna_test2_cmp.csv

```
# label and seq data
uniflna_test2 <-
  read.csv("/home/brian/Documents/data_mining/flu_project/uniflna_test2.csv", na.strings="")

# comparisons from sourmash
uniflna_test2_cmp <-
  read.csv("/home/brian/Documents/data_mining/flu_project/uniflna_test2_cmp.csv")

# Label the rows
rownames(uniflna_test2_cmp) <- colnames(uniflna_test2_cmp)

# add gene column
uniflna_test2_wgs <- uniflna_test2_cmp
uniflna_test2_wgs$gene <- uniflna_test2$gene
uniflna_test2_wgs$type <- uniflna_test2$type
uniflna_test2_wgs$HNtype <- uniflna_test2$HNtype
#uniflna_test2_wgs$segment <- factor(uniflna_test1$segment)

# Transform for plotting
uniflna_test2_cmp_mat <- as.matrix(uniflna_test2_cmp)
```

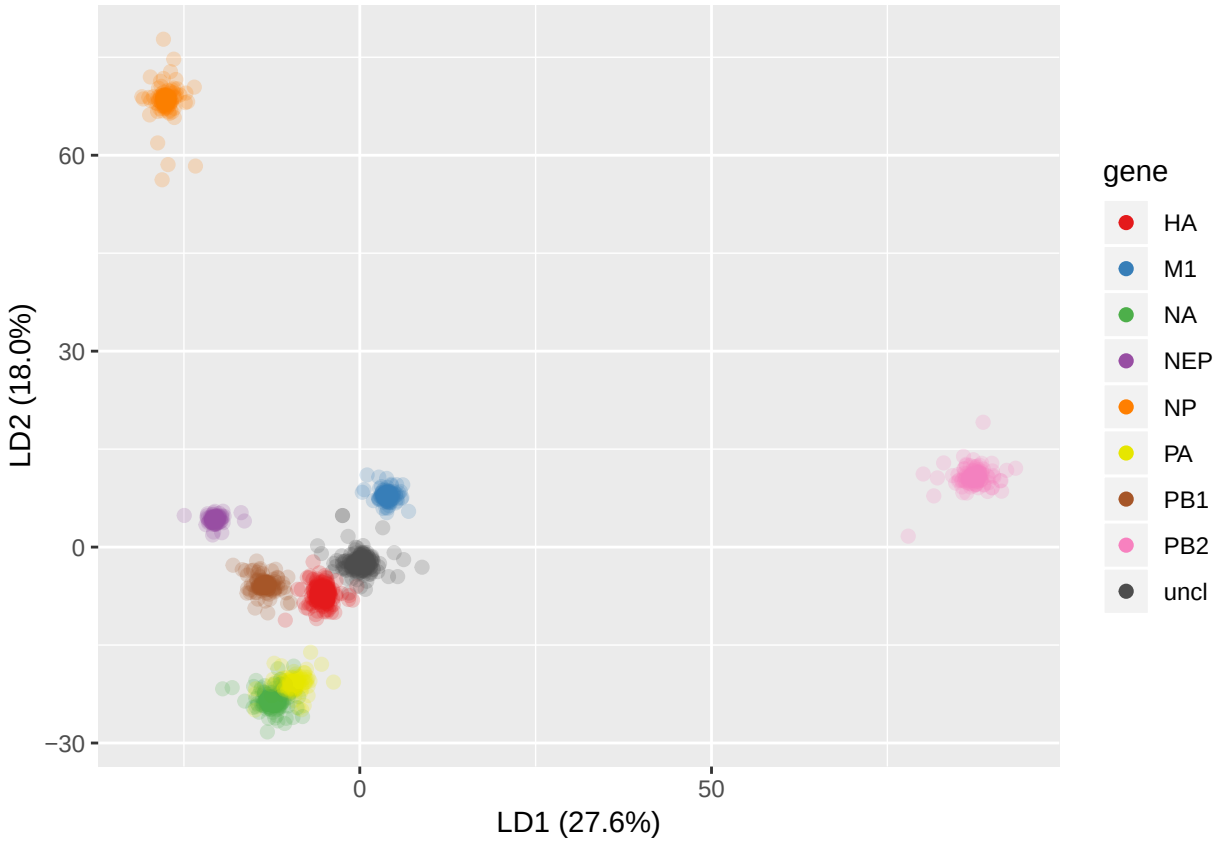
```
lda3 <- lda(gene ~ .,
            uniflna_test2_wgs)
```

```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

```
prop.lda3 = lda3$svd^2/sum(lda3$svd^2)

plda3 <- predict(object = lda3,
                 newdata = uniflna_test2_wgs)
```

```
dataset3 = data.frame(gene = uniflna_test2_wgs[,2501],
                      lda3 = plda3$x)
```

```
p3 <- ggplot(dataset3) + geom_point(aes(lda3.LD1, lda3.LD2, colour = gene), size=2, alpha=0.2) +
  #theme_minimal() +3
  labs(x = paste("LD1 (", percent(prop.lda3[1]), ")", sep=""),
       y = paste("LD2 (", percent(prop.lda3[2]), ")", sep="")) +
  scale_color_manual(values=cpal) +
  guides(colour = guide_legend(override.aes = list(alpha = 1)))
# https://stackoverflow.com/questions/5290003/how-to-set-legend-alpha-with-ggplot2

# ggsave("gene2500_LDA_plot.png", plot = p3, width = 7, height = 4, dpi = 200)

p3
```



ABC type labels

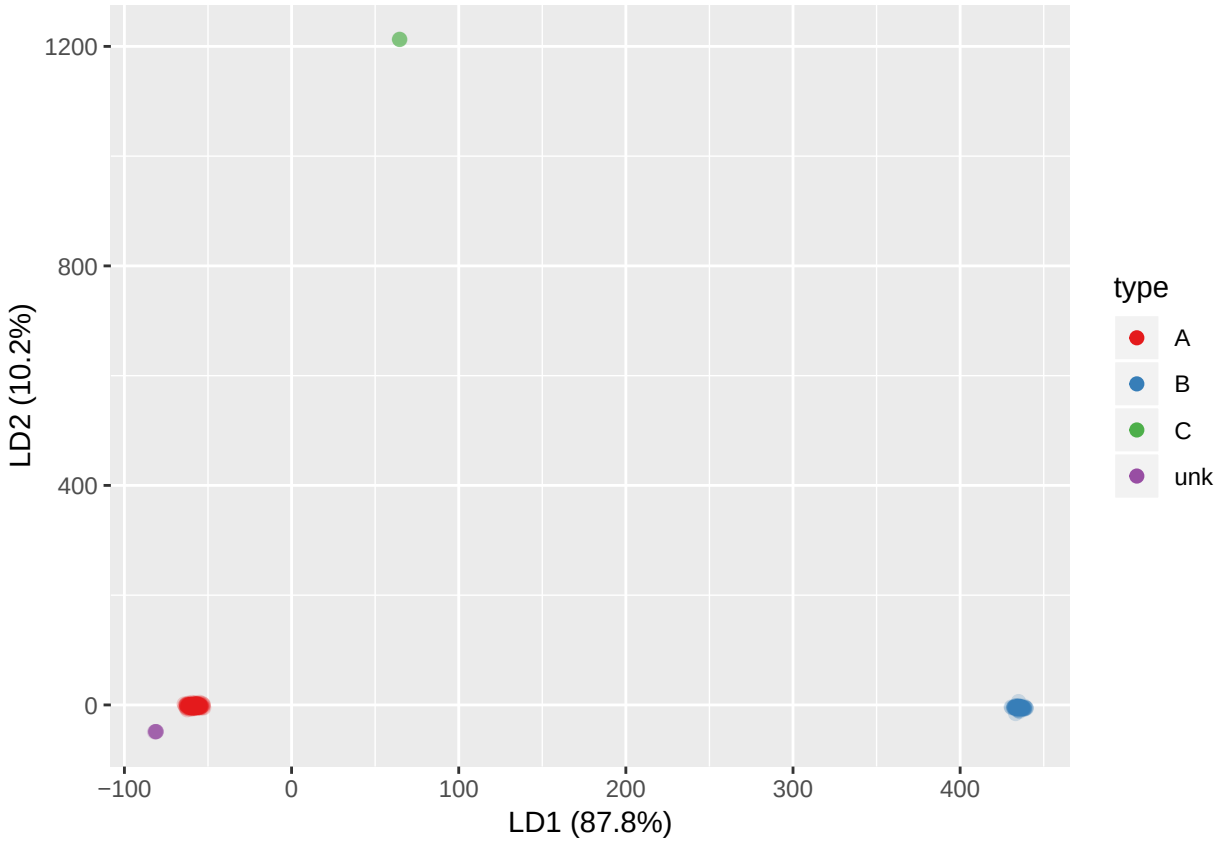
```
lda4 <- lda(type ~ .,
            uniflna_test2_wgs)
```

```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

```
prop.lda4 = lda4$svd^2/sum(lda4$svd^2)

plda4 <- predict(object = lda4,
                 newdata = uniflna_test2_wgs)
```

```
dataset4 = data.frame(type = uniflna_test2_wgs[,2502],
                      lda4 = plda4$x)
```

```
p4 <- ggplot(dataset4) + geom_point(aes(lda4.LD1, lda4.LD2, colour = type), size=2, alpha=0.2) +
  #theme_minimal() +3
  labs(x = paste("LD1 (", percent(prop.lda4[1]), ")", sep=""),
       y = paste("LD2 (", percent(prop.lda4[2]), ")", sep="")) +
  scale_color_manual(values=cpal) +
  guides(colour = guide_legend(override.aes = list(alpha = 1)))
# https://stackoverflow.com/questions/5290003/how-to-set-legend-alpha-with-ggplot2

#ggsave("ABCtype_LDA_plot.png", plot = p4, width = 7, height = 4, dpi = 200)

p4
```




```
table(uniflna_test2_wgs$type)
```

```
##
##    A    B    C  unk
## 2191  295    5    9
```

HN type labels

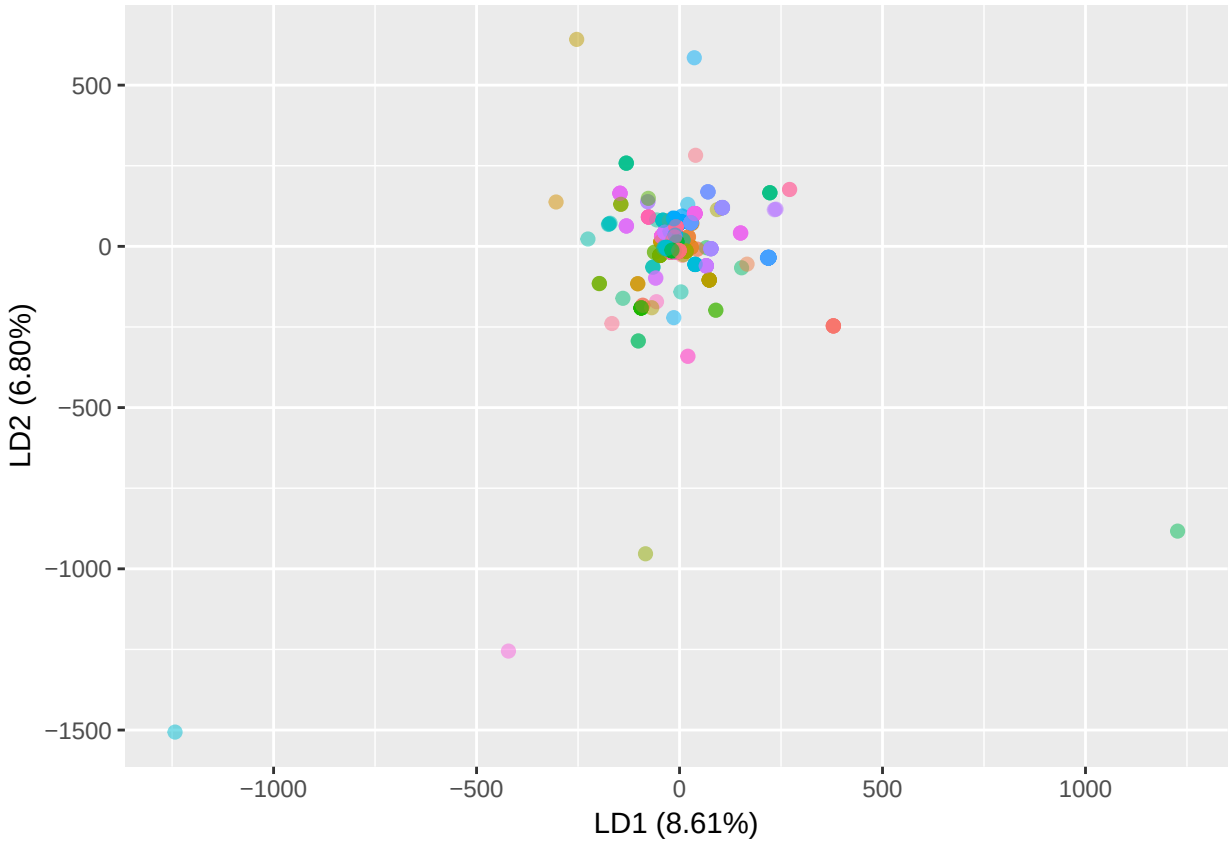
```
lda5 <- lda(HNtype ~ .,
            uniflna_test2_wgs)
```

```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

```
prop.lda5 = lda5$svd^2/sum(lda5$svd^2)

plda5 <- predict(object = lda5,
                 newdata = uniflna_test2_wgs)
```

```
dataset5 = data.frame(HNtype = uniflna_test2_wgs[,2503],
                      lda5 = plda5$x)
```

```
p5 <- ggplot(dataset5) + geom_point(aes(lda5.LD1, lda5.LD2, colour = HNtype), size=2, alpha=0.5) +
  #theme_minimal() +3
  labs(x = paste("LD1 (", percent(prop.lda5[1]), ")", sep=""),
       y = paste("LD2 (", percent(prop.lda5[2]), ")", sep="")) + theme(legend.position="none")
  #scale_color_manual(values=cpal) +
  #guides(colour = guide_legend(override.aes = list(alpha = 1)))


p5
```

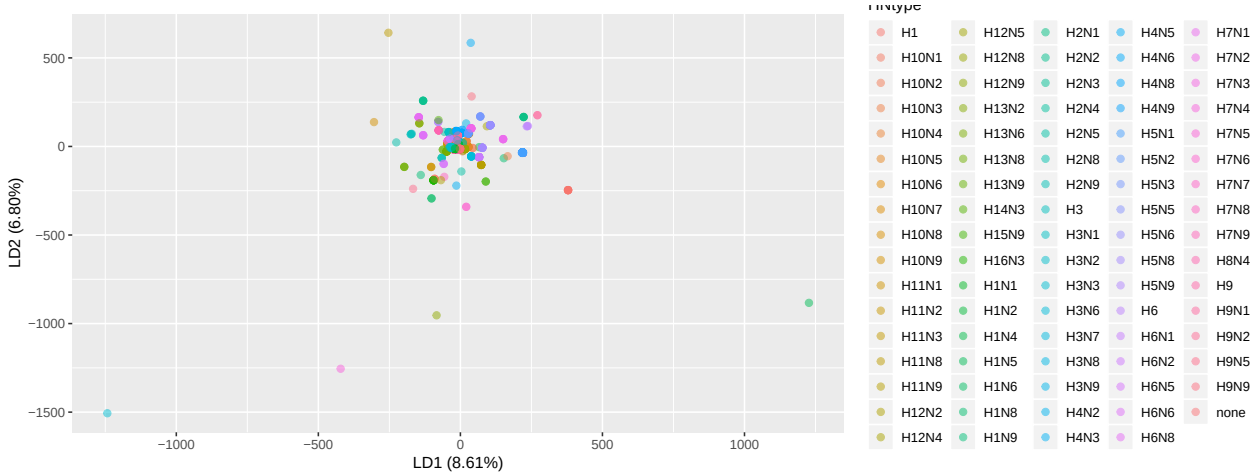



```
table(uniflna_test2_wgs$HNtype)
```

```
##
##    H1 H10N1 H10N2 H10N3 H10N4 H10N5 H10N6 H10N7 H10N8 H10N9 H11N1 H11N2
##     7     3     2     1     4     4     2    23     5     1     3     6
```

```
## H11N3 H11N8 H11N9 H12N2 H12N4 H12N5 H12N8 H12N9 H13N2 H13N6 H13N8 H13N9
##     2     1     9     1     1    13     1     1     3    10     3     2
## H14N3 H15N9 H16N3  H1N1  H1N2  H1N4  H1N5  H1N6  H1N8  H1N9  H2N1  H2N2
##     1     2    11   572    90     1     1     2     1     1     4     4
##  H2N3  H2N4  H2N5  H2N8  H2N9    H3  H3N1  H3N2  H3N3  H3N6  H3N7  H3N8
##     8     1     1     1     3     4     5   628     1    10     1    91
##  H3N9  H4N2  H4N3  H4N5  H4N6  H4N8  H4N9  H5N1  H5N2  H5N3  H5N5  H5N6
##     1     8     1     1    33    14     2   149    59     5     1    16
##  H5N8  H5N9    H6  H6N1  H6N2  H6N5  H6N6  H6N8  H7N1  H7N2  H7N3  H7N4
##    10     3     3    14    34     3     5     7     5    21    20     1
##  H7N5  H7N6  H7N7  H7N8  H7N9  H8N4    H9  H9N1  H9N2  H9N5  H9N9  none
##     1     2    17     1    27     8     8     2    98     1     1   366
```

```
ggplot(dataset5) + geom_point(aes(lda5.LD1, lda5.LD2, colour = HNtype), size=2, alpha=0.5) +
  #theme_minimal() +3
  labs(x = paste("LD1 (", percent(prop.lda5[1]), ")", sep=""),
       y = paste("LD2 (", percent(prop.lda5[2]), ")", sep=""))
```




```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

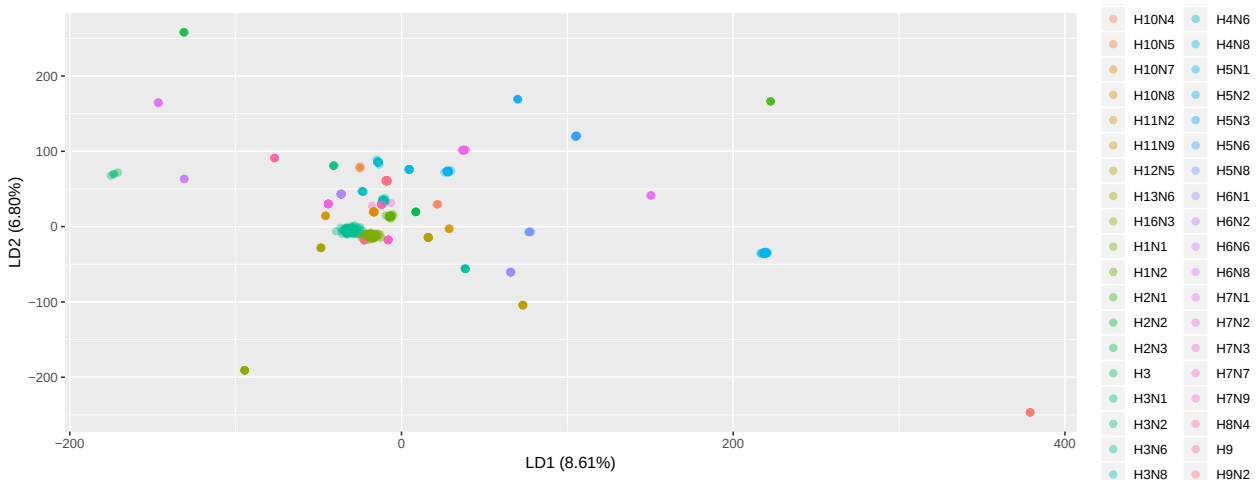
```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

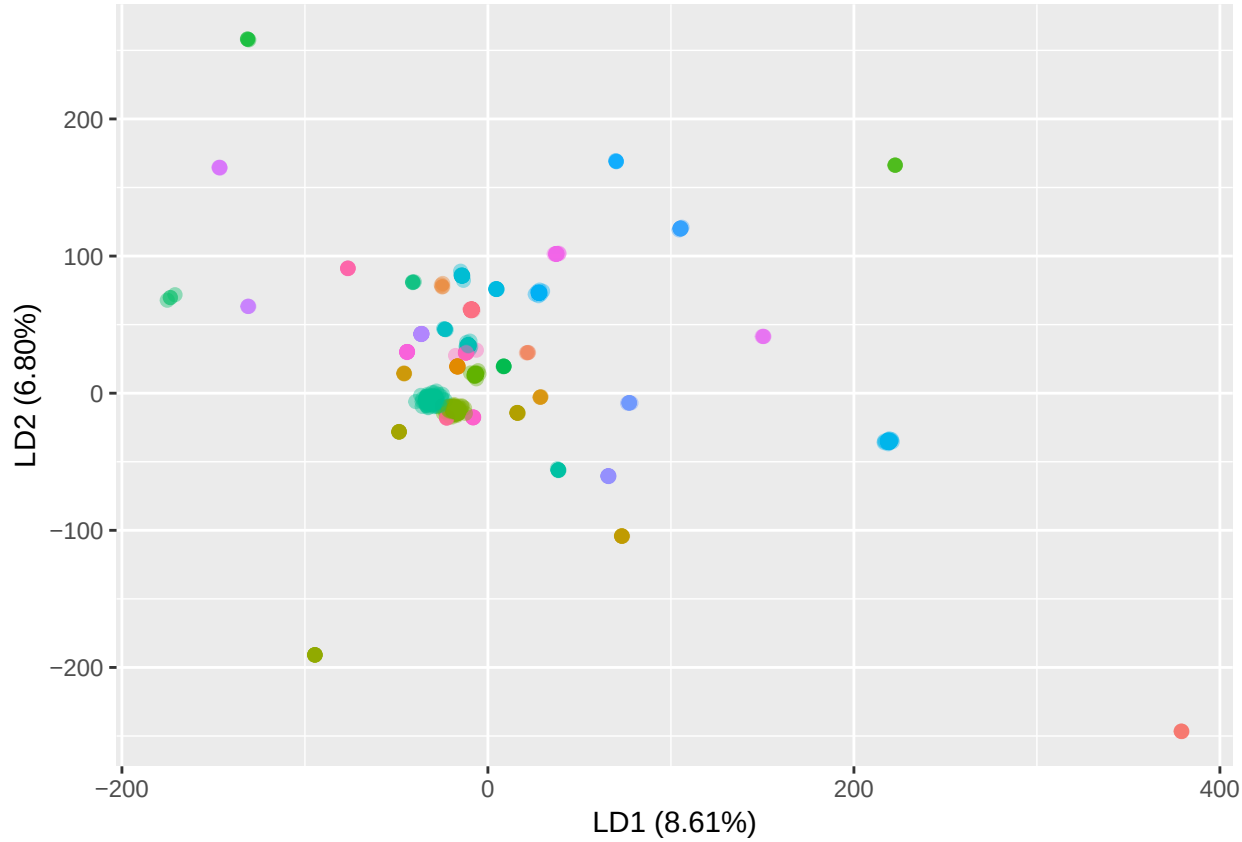
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
fds5 <- dataset5 %>% group_by(HNtype) %>% filter(n() > 3) %>% filter(HNtype != 'none')
```

```
ggplot(fds5) + geom_point(aes(lda5.LD1, lda5.LD2, colour = HNtype), size=2, alpha=0.4) +
  #theme_minimal() +
  labs(x = paste("LD1 (", percent(prop.lda5[1]), ")", sep=""),
       y = paste("LD2 (", percent(prop.lda5[2]), ")", sep=""))
```



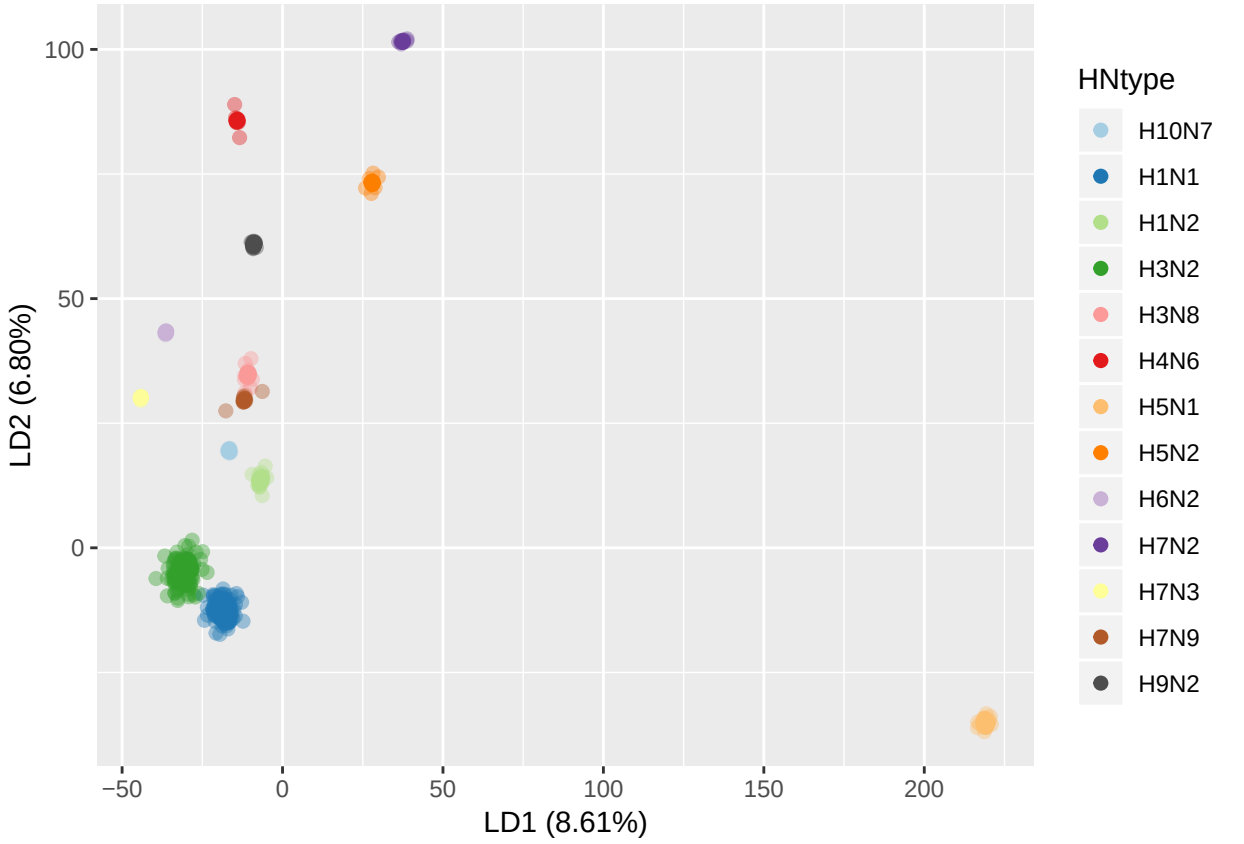

```
ggplot(fds5) + geom_point(aes(lda5.LD1, lda5.LD2, colour = HNtype), size=2, alpha=0.4) +
  labs(x = paste("LD1 (", percent(prop.lda5[1]), ")", sep=""),
       y = paste("LD2 (", percent(prop.lda5[2]), ")", sep="")) +
  theme(legend.position="none")
```




```
  #scale_color_manual(values=cpal) +
  #guides(colour = guide_legend(override.aes = list(alpha = 1)))
# https://stackoverflow.com/questions/5290003/how-to-set-legend-alpha-with-ggplot2

#ggsave("HNType_LDA_plot.png", width = 7, height = 4, dpi = 200)
```

```
fds5p20 <- dataset5 %>% group_by(HNtype) %>% filter(n() >= 20) %>% filter(HNtype != 'none')
```

```
tempcolors = c('#a6cee3','#1f78b4','#b2df8a','#33a02c','#fb9a99','#e31a1c','#fdbf6f',
               '#ff7f00','#cab2d6','#6a3d9a','#ffff99','#b15928','#4d4d4d')
ggplot(fds5p20) + geom_point(aes(lda5.LD1, lda5.LD2, colour = HNtype), size=2, alpha=0.4) +
  labs(x = paste("LD1 (", percent(prop.lda5[1]), ")", sep=""),
       y = paste("LD2 (", percent(prop.lda5[2]), ")", sep="")) +
  scale_color_manual(values=tempcolors) +
  guides(colour = guide_legend(override.aes = list(alpha = 1)))
```




```
# ggsave("HNType_thirty_LDA_plot.png", width = 7, height = 4, dpi = 200)
```