BIOS 664 HW4

Meng-Ni Ho, Chun-Hui Lin

4/22/2019

Background and Introduction

The primary goal of this project is to apply unsupervised learning algorithms to detect the population structure using the DNA information from samples. In this case, each sample is represented by 2,540 genetic markers (more precisely, SNPs) from the human genetic diversity panel (HGDP) data. Although each marker may provide very limited population structure information, the combination of all the markers can potentially improved the performance of population clustering dramatically.

Problems

Load data:

V2: country, V3: continent, V4 ~ V927: genetics

```
df = read.table("hgdp.dat")
dim(df)
```

```
## [1]  927 2543
```

Recode:

AA = 0, AG = 1, GA = 2, GG = 3, CC = 4, CT = 5, TT = 6, TC = 7, CG = 8, GC = 9, GT = 10,TG = 11

```
# recode genotype to integer
recode2 = function(col){
  return(recode(col, '"AA" = 0; "AG" = 1;  "GA" = 2;  "GG" = 3; "CC" = 4; "CT" = 5; "TT" = 6; "TC" = 7;
}
df2 = data.frame(lapply(df, as.character), stringsAsFactors=FALSE)
df3 = as.data.frame(sapply(df2, recode2))
name = df3 %>% select(V1, V2, V3) %>% mutate(V1 = as.numeric(V1))
df4 = data.frame(lapply(df3[,c(1,4:2543)], as.numeric), stringsAsFactors=FALSE)
df5 = merge(name, df4, by = 'V1')
```

1. Use PCA to explore the population structure using the only the genotype data (i.e., ignore sampling location and continent information for now).

```
pca = prcomp(df5[,c(4:2543)], center = TRUE,scale. = TRUE)
result = summary(pca)
variance = as.data.frame(result$importance)
var_ex = variance[which(variance[2,] > 0.01)]
dim(var_ex)
```

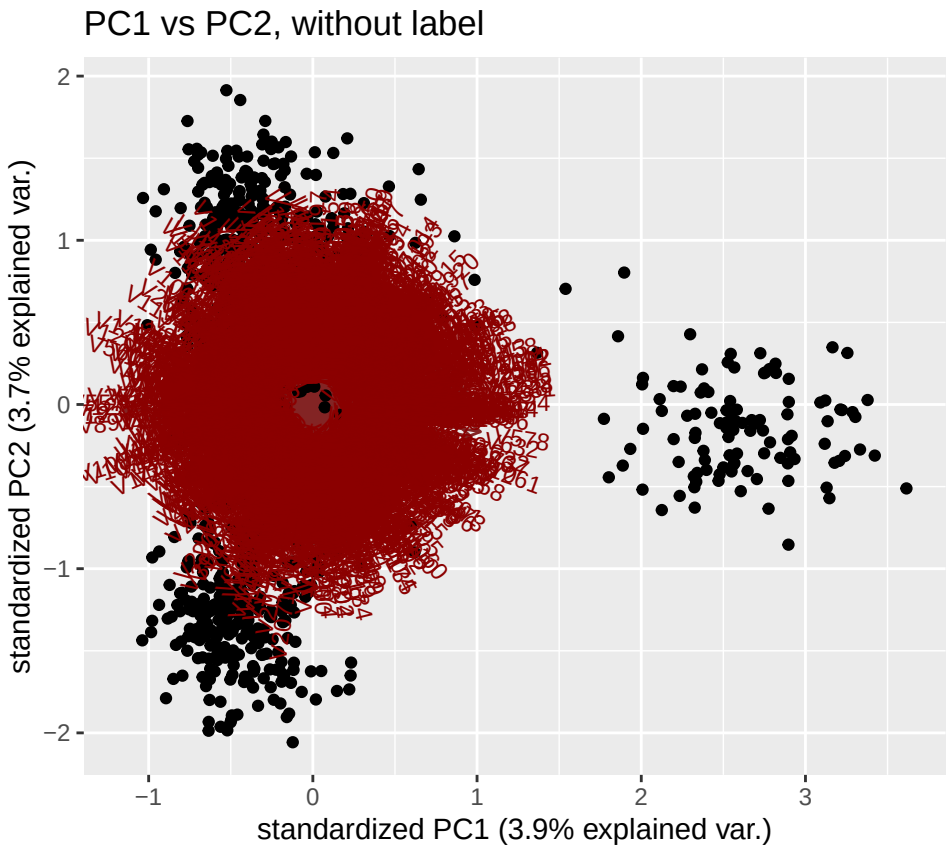
```
## [1] 3 6
```

```
kable(var_ex)
```

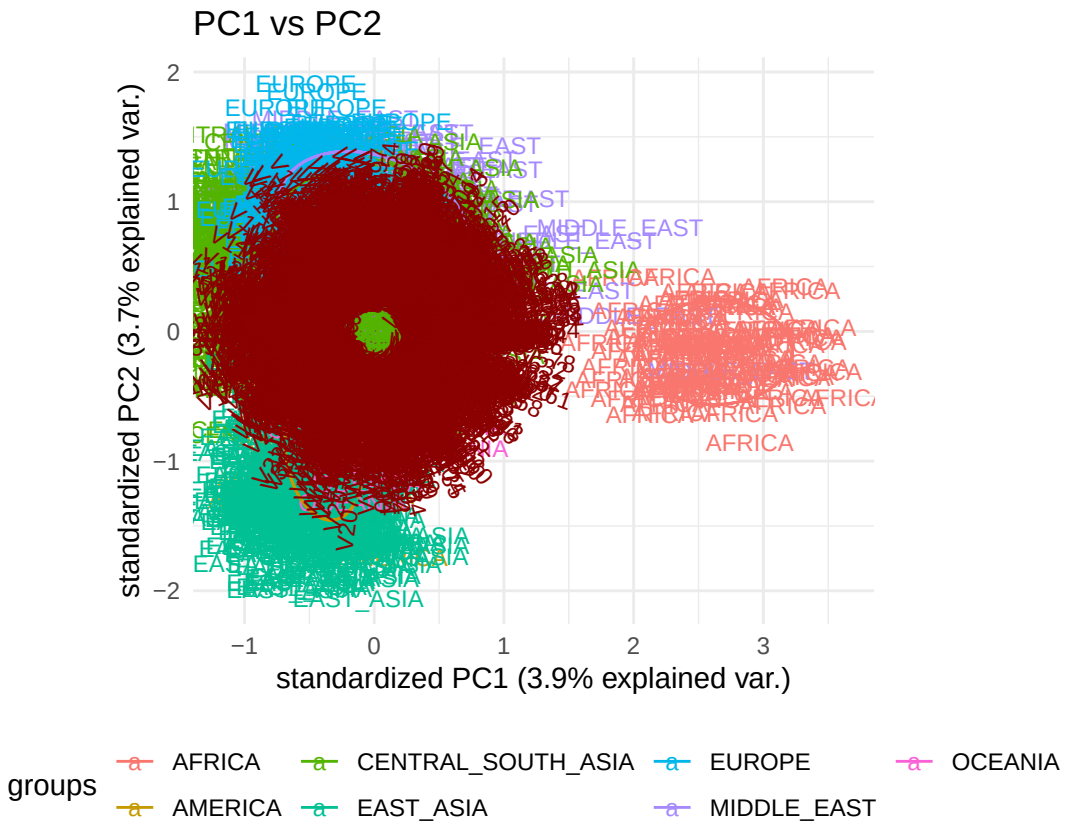
	PC1	PC2	PC3	PC4	PC5	PC6
Standard deviation	9.938323	9.650193	6.376878	5.335457	5.195621	5.075733
Proportion of Variance	0.038890	0.036660	0.016010	0.011210	0.010630	0.010140
Cumulative Proportion	0.038890	0.075550	0.091560	0.102770	0.113390	0.123540

2. Visualize the cluster structures identified from PCA, color each sample point using its continental information. (you may wish to plot multiple pairs of PCs)

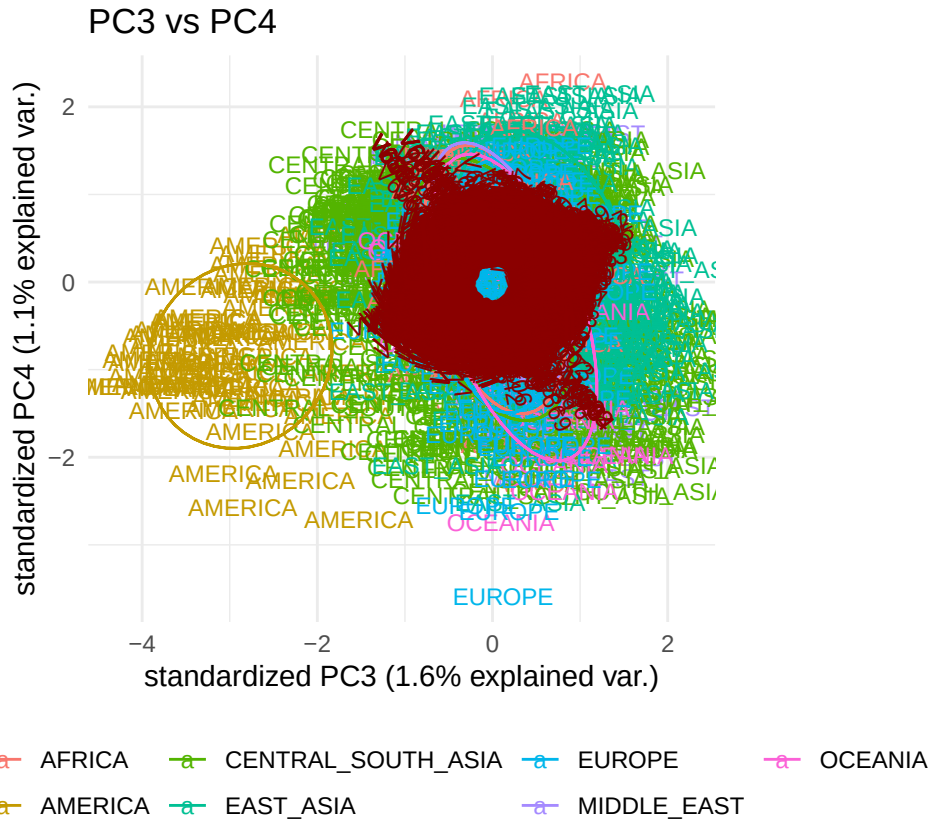
```
ggbiplot(pca) + ggtitle("PC1 vs PC2, without label")
```



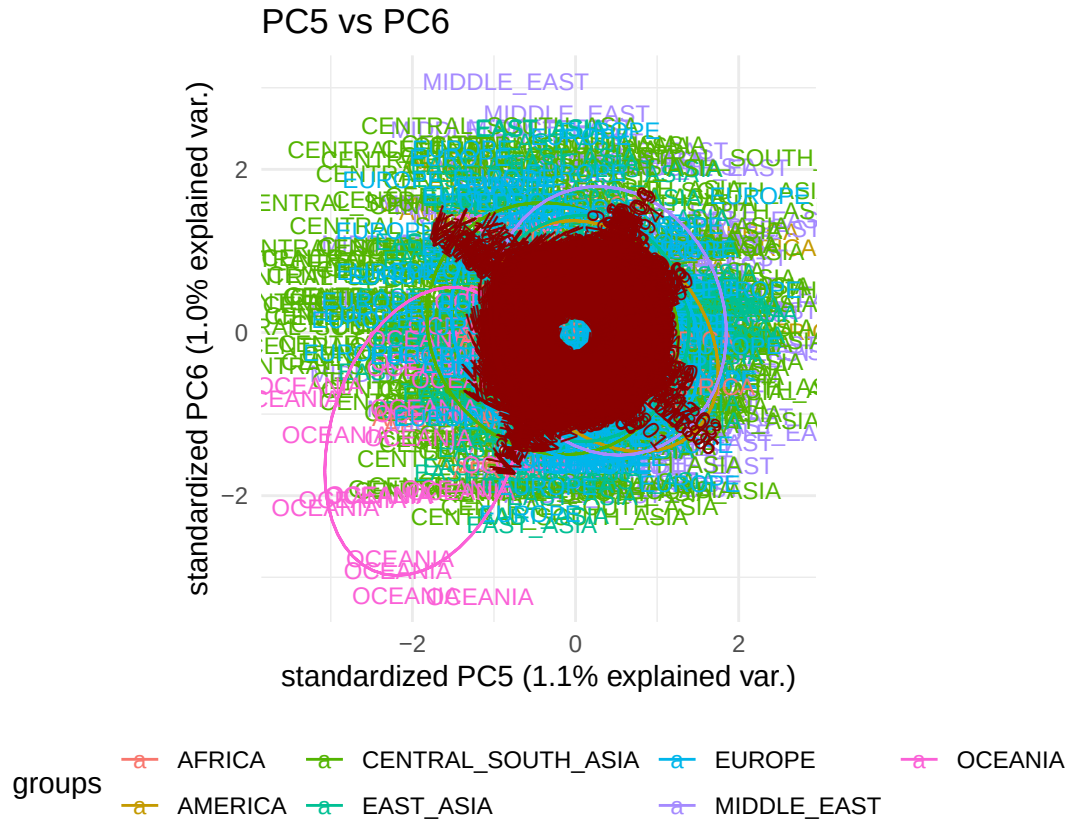

```
# "PC1 vs PC2"
ggbiplot(pca, ellipse = TRUE,  labels = df5$V3, groups=df5$V3) +
  ggtitle("PC1 vs PC2") + theme_minimal() + theme(legend.position = "bottom")
```




```
# "PC3 vs PC4"
ggbiplot(pca, ellipse=TRUE, choices=c(3,4), labels=df5$V3, groups=df5$V3) +
  ggtitle("PC3 vs PC4") + theme_minimal() + theme(legend.position = "bottom")
```




```
# "PC5 vs PC6"
ggbiplot(pca, ellipse=TRUE, choices=c(5,6), labels=df5$V3, groups=df5$V3) +
  ggtitle("PC5 vs PC6") + theme_minimal() + theme(legend.position = "bottom")
```



3. Comment on the cluster structures identified from the PCA analysis.

By looking at the plot, PC1 (explained 3.9% variation) vs PC2 (explained 3.7% variation) appear to have a better cluster when grouping by continent, with central South Asia, Europe closer to the center, and Africa being the most apart from center. However, when looking at PC3 (explained 1.6% variation) vs PC4 (explained 1.1% variation), American seems to be the most apart from center. PC5 vs PC6 did not show much clustering, since all the groups were aggregated together.

4. Do research on the emerging technique known as "t-distributed stochastic neighbor embedding", or, t-SNE, apply it to the data set. Summarize its connection and difference with PCA. Comment on the t-SNE result in comparison to the PCA result of the HGDP data.

- t-Distributed Stochastic Neighbor Embedding (t-SNE) is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets.

```
labels = df3$V3
df3$V3 = as.numeric(df3$V3)
colors = rainbow(length(unique(df3$V3)))
names(colors) = unique(df3$V3)
# reduce to 2-dim
tsne = Rtsne(df3[,-c(1:3)], dims = 2, perplexity=30, verbose=TRUE, max_iter = 500)
```

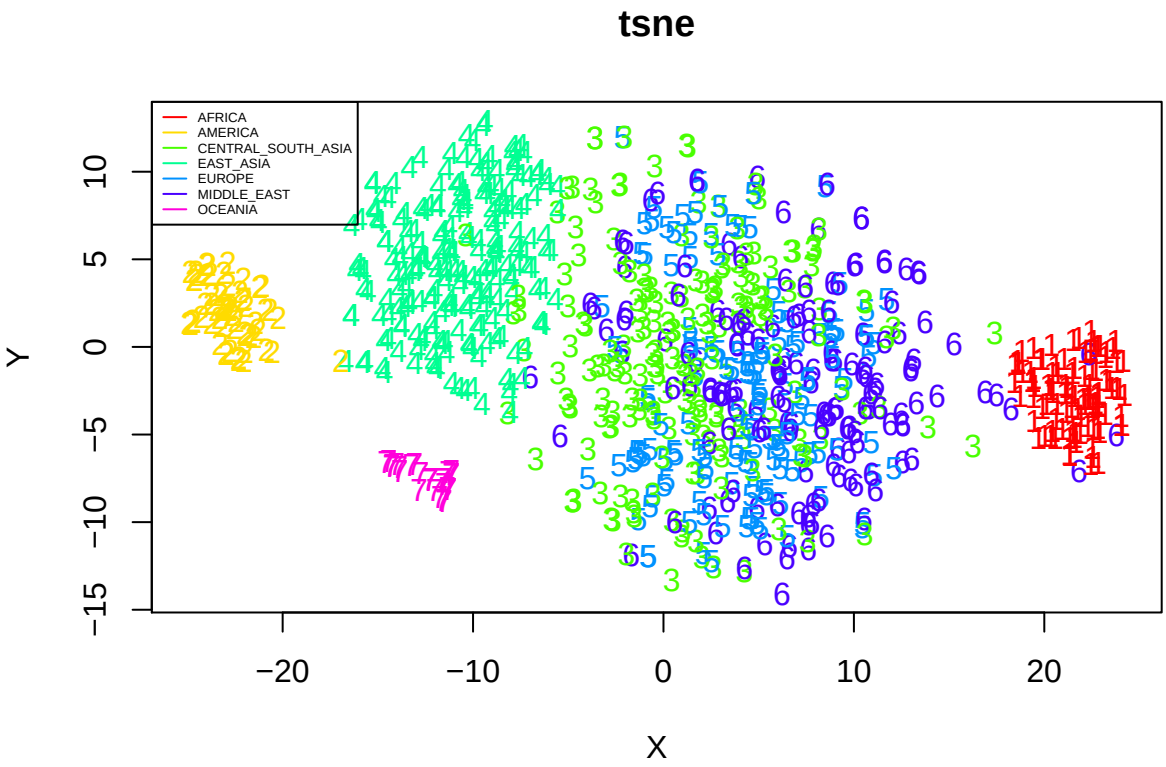
```
## Performing PCA
## Read the 927 x 50 data matrix successfully!
## OpenMP is working. 1 threads.
## Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
## Computing input similarities...
```

```
## Building tree...
## Done in 0.14 seconds (sparsity = 0.129927)!
## Learning embedding...
## Iteration 50: error is 68.696910 (50 iterations in 0.15 seconds)
## Iteration 100: error is 67.452921 (50 iterations in 0.13 seconds)
## Iteration 150: error is 67.433295 (50 iterations in 0.11 seconds)
## Iteration 200: error is 67.434867 (50 iterations in 0.11 seconds)
## Iteration 250: error is 67.434718 (50 iterations in 0.11 seconds)
## Iteration 300: error is 1.868020 (50 iterations in 0.11 seconds)
## Iteration 350: error is 1.748670 (50 iterations in 0.11 seconds)
## Iteration 400: error is 1.707172 (50 iterations in 0.11 seconds)
## Iteration 450: error is 1.691271 (50 iterations in 0.11 seconds)
## Iteration 500: error is 1.684187 (50 iterations in 0.11 seconds)
## Fitting performed in 1.16 seconds.
```

```
summary(tsne)
```

```
##                      Length Class  Mode
## N                       1   -none- numeric
## Y                    1854   -none- numeric
## costs                 927   -none- numeric
## itercosts              10   -none- numeric
## origD                   1   -none- numeric
## perplexity              1   -none- numeric
## theta                   1   -none- numeric
## max_iter                1   -none- numeric
## stop_lying_iter         1   -none- numeric
## mom_switch_iter         1   -none- numeric
## momentum                1   -none- numeric
## final_momentum          1   -none- numeric
## eta                     1   -none- numeric
## exaggeration_factor     1   -none- numeric
```

```
# display the results of t-SNE
# tsne$Y: Matrix containing the new representations for the objects
plot(tsne$Y, t='n', main="tsne", xlab = "X", ylab = "Y")
text(tsne$Y, labels=df3$V3, col=colors[df3$V3])
legend("topleft", legend=c("AFRICA", "AMERICA", "CENTRAL_SOUTH_ASIA", "EAST_ASIA", "EUROPE", "MIDDLE_EA
```



By looking at the tsne plot, Africa (1) and America (2) seems to be apart from the other continents, this correspond to the results in PC1 vs PC2 (Africa being the most apart), and PC3 vs PC4 (America being the most apart).