Homework #8 **20180047 김경수**

0. 이론적 배경

(1) Clustering - K-means, Hierarchical clustering

Clustering이란 unsupervised learning에 속하며, 가장 기본적이며 인지도 있는 방법은 hierarchical clustering과 K-means clustering이다.

Hierarchical clustering은 single object를 각각의 cluster로 한 n개의 cluster 상태에서 시작하며, 하나의 cluster가 남을 때까지 두 개의 가장 similar한 cluster들을 합치는 방식으로 진행된다. Linkage, distance measurement에 의해 cluster들이 순차적으로 합쳐진다. Linkage 방식은 single, complete, centroid, average 등등이 존재한다. Distance measurement 역시, 일반적인 euclidian distance 뿐만 아니라, correlation based distance 등 여러 선택이 가능하다. 일반적으로 알려진 최적의 조건은 없으며, data set의 특성, clustering의 목적 등에 따라 유동적으로 조정하여야 한다. 특징으로는 높은 computational complexity를 가지며, 계층적 관계를 가진다는 전제조건하에 clustering이 진행되므로, 실제 data set이 이러한 가정과 유사할 경우 해석력이 높다.

K-means clustering은 최종 cluster의 개수를 k개로 미리 지정한 후 진행하는 clustering 방식이다. Within cluster variation(WCV)의 합을 최소화 시키는 cluster를 찾는 것이 목적이다. 먼저 K-cluster의 center를 랜덤하게 설정한 후, expectation & maximization을 진행한다. cluster들이 converge할 때 까지 expectation & maximization을 적절하게 수행한다. 작은 computational complexity를 가지며, reproducible한 result를 도출하지 않으며, 서로 다른 density나 size, non-convex한 shape를 가지는 data set에 대해 신뢰성 있는 결과를 도출해내지 못한다. 이를 해결하기 위해 kernel function을 이용하거나, mean-shift, density based clustering 등 여러 방법들이 존재한다.

(2) Cluster assessment

unsupervised 이기 때문에 이전의 방법들처럼 cross-validation을 진행하지 못한다. 차선적인 internal, external assessment이 존재한다. 본 보고서에서 사용한 것은 silhouette coefficient이며, ratio of average intra-cluster to the inter-clusters distances를 수치화한 것이다. (a(i)는 same cluster의 구성원들간의 평균 거리이며, b(i)는 가장 가까운 이웃 cluster의 구성원들간의 평균 거리이다.)

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

또한 jaccard index를 사용하였다. 공통의 원소가 있으면 0, 두 집합이 동일하면 1의 값을 가진다.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

HW 8

1. Build hierarchical clustering model using whole genes. According to silhouette index, what is optimal cluster number? Then, assess model by calculating jaccard index between cluster membership and tumor stage. (Set linkage & distance measure according to your rationale)

2. Repeat upper question using PCA trasformed data. How many prinicipal components should be used? And discuss clustering model's similarity with tumor stage. If similarity is low, what would be the cause of it.

1. Procedure & R code

(1) clinical, expresson rds파일을 implement한 후, subtype이 NA인 정보를 제거한다.

R code :

```
gex <- readRDS("expression.rds")
clin <- readRDS("clinical.rds")
DATA <-na.omit(t(gex))
```

(2) Average linkage와, complete linkage 각각에 대해 hierarchical clustering을 진행하고, 그 결과를 확인한다. (결과 확인 후 complete linkage 방식을 채택하였고 그 이유를 후술함. Distance는 euclidian distance를 이용하였다.)

R code :

```
hc_average <- hclust(dist(DATA), method="average")
{plot(hc_average, main="Average Linkage", xlab="", sub="", cex=.9)
  rect.hclust(hc_average, k=4)}
hc_complete <- hclust(dist(DATA), method="complete")
{plot(hc_complete, main="Complete Linkage", xlab="", sub="", cex=.9)
  rect.hclust(hc_complete, k=4)}
```

(3) Silhouette coefficient를 이용하여 최적의 cluster number를 확인하고, 그 모델에 대한 jaccard index를 측정하여 cluster 간의 similarity를 확인한다..

R code :

```
fviz_nbclust(DATA, hcut, method="silhouette", hc_method="complete")
DATA_sam <- data.frame(as.character(rownames(DATA)), DATA)
colnames(DATA_sam)[1] <- "sample_id"
stage <- as.vector(clin$stage)
stage_p <- rep(0,length(stage))
for (i in 1:length(stage)){
  if(stage[i]=='stage i'){stage_p[i]=1}
  else if(stage[i]=='stage ii'){stage_p[i]=2}
  else if(stage[i]=='stage iii'){stage_p[i]=3}
```

```
  else if(stage[i]=='stage iv'){stage_p[i]=4}
  else if(stage[i]=='stage v'){stage_p[i]=5}
}
clin$stage <- stage_p
DATA_clin <- merge(clin[,c("sample_id","stage")], DATA_sam, by = "sample_id", all = FALSE)
data <- DATA_clin
data$stage <- NULL
data$sample_id <- NULL
data_hc <- cutree(hclust(dist(data), method="complete"), 3)
print(cluster_similarity(DATA_clin$stage, data_hc, similarity="jaccard"))
```

(4) PCA를 이용하여 차원축소를 진행한다.

R code :

```
PCA_DATA <- t(gex)
PCA <- prcomp(na.omit(PCA_DATA), scale=T, center=T)
```

(5) PCA를 이용하여 차원축소를 진행한다. 주성분이 2~5개 일 때의 4가지 case의 최적모델을 silhouette coefficient를 이용하여 구한 후 이에 대해서 jaccard index를 측정한 후 비교한다. 이를 통해 최적의 주성분의 수를 확인할 수 있다.

R code :

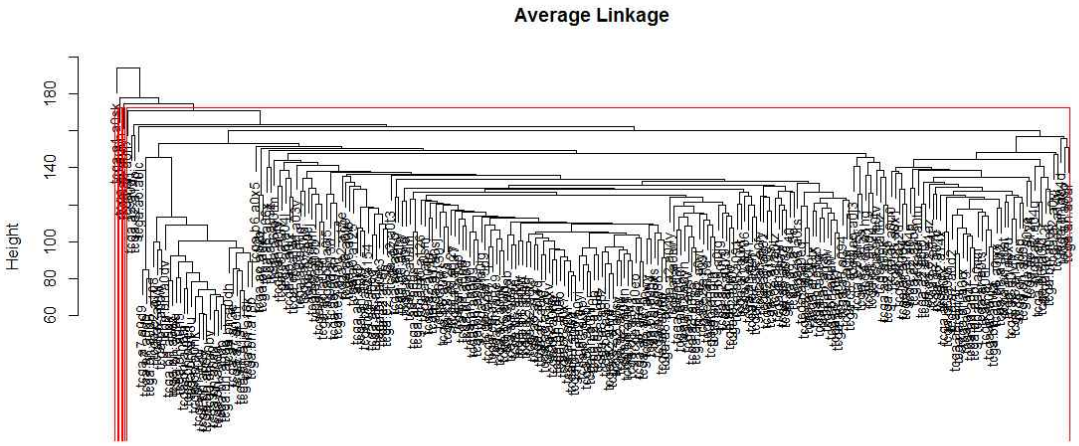
```
DATA <-PCA$x[,1:2]
fviz_nbclust(DATA, hcut, method="silhouette", hc_method="complete")
DATA_sam <- data.frame(as.character(rownames(DATA)), DATA)
colnames(DATA_sam)[1] <- "sample_id"
stage <- as.vector(clin$stage)
stage_p <- rep(0,length(stage))
for (i in 1:length(stage)){
  if(stage[i]=='stage i'){stage_p[i]=1}
  else if(stage[i]=='stage ii'){stage_p[i]=2}
  else if(stage[i]=='stage iii'){stage_p[i]=3}
  else if(stage[i]=='stage iv'){stage_p[i]=4}
  else if(stage[i]=='stage v'){stage_p[i]=5}
}
clin$stage <- stage_p
DATA_clin <- merge(clin[,c("sample_id","stage")], DATA_sam, by = "sample_id", all = FALSE)
data <- DATA_clin
data$stage <- NULL
data$sample_id <- NULL
data_hc <- cutree(hclust(dist(data), method="complete"), 2)
print(cluster_similarity(DATA_clin$stage, data_hc, similarity="jaccard"))
```

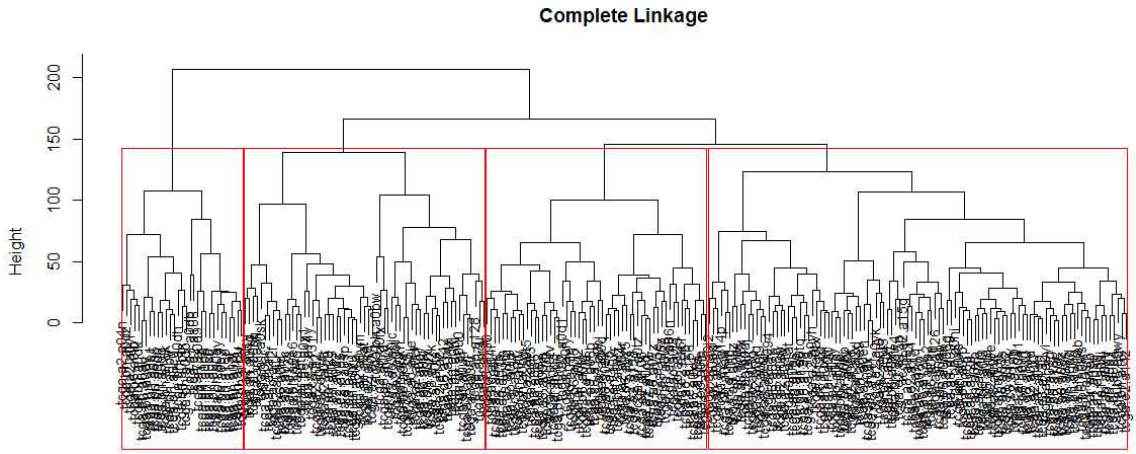
```
DATA <-PCA$x[,1:3]
fviz_nbclust(DATA, hcut, method="silhouette", hc_method="complete")
DATA_sam <- data.frame(as.character(rownames(DATA)), DATA)
colnames(DATA_sam)[1] <- "sample_id"
stage <- as.vector(clin$stage)
stage_p <- rep(0,length(stage))
for (i in 1:length(stage)){
  if(stage[i]=='stage i'){stage_p[i]=1}
  else if(stage[i]=='stage ii'){stage_p[i]=2}
  else if(stage[i]=='stage iii'){stage_p[i]=3}
  else if(stage[i]=='stage iv'){stage_p[i]=4}
  else if(stage[i]=='stage v'){stage_p[i]=5}
}
clin$stage <- stage_p
DATA_clin <- merge(clin[,c("sample_id","stage")], DATA_sam, by = "sample_id", all =
FALSE)
data <- DATA_clin
data$stage <- NULL
data$sample_id <- NULL
data_hc <- cutree(hclust(dist(data), method="complete"), 3)
print(cluster_similarity(DATA_clin$stage, data_hc, similarity="jaccard"))

DATA <-PCA$x[,1:4]
fviz_nbclust(DATA, hcut, method="silhouette", hc_method="complete")
DATA_sam <- data.frame(as.character(rownames(DATA)), DATA)
colnames(DATA_sam)[1] <- "sample_id"
stage <- as.vector(clin$stage)
stage_p <- rep(0,length(stage))
for (i in 1:length(stage)){
  if(stage[i]=='stage i'){stage_p[i]=1}
  else if(stage[i]=='stage ii'){stage_p[i]=2}
  else if(stage[i]=='stage iii'){stage_p[i]=3}
  else if(stage[i]=='stage iv'){stage_p[i]=4}
  else if(stage[i]=='stage v'){stage_p[i]=5}
}
clin$stage <- stage_p
DATA_clin <- merge(clin[,c("sample_id","stage")], DATA_sam, by = "sample_id", all =
FALSE)
data <- DATA_clin
data$stage <- NULL
data$sample_id <- NULL
```

```
data_hc <- cutree(hclust(dist(data), method="complete"), 3)
print(cluster_similarity(DATA_clin$stage, data_hc, similarity="jaccard"))

DATA <-PCA$x[,1:5]
fviz_nbclust(DATA, hcut, method="silhouette", hc_method="complete")
DATA_sam <- data.frame(as.character(rownames(DATA)), DATA)
colnames(DATA_sam)[1] <- "sample_id"
stage <- as.vector(clin$stage)
stage_p <- rep(0,length(stage))
for (i in 1:length(stage)){
  if(stage[i]=='stage i'){stage_p[i]=1}
  else if(stage[i]=='stage ii'){stage_p[i]=2}
  else if(stage[i]=='stage iii'){stage_p[i]=3}
  else if(stage[i]=='stage iv'){stage_p[i]=4}
  else if(stage[i]=='stage v'){stage_p[i]=5}
}
clin$stage <- stage_p
DATA_clin <- merge(clin[,c("sample_id","stage")], DATA_sam, by = "sample_id", all =
FALSE)
data <- DATA_clin
data$stage <- NULL
data$sample_id <- NULL
data_hc <- cutree(hclust(dist(data), method="complete"), 4)
print(cluster_similarity(DATA_clin$stage, data_hc, similarity="jaccard"))
```

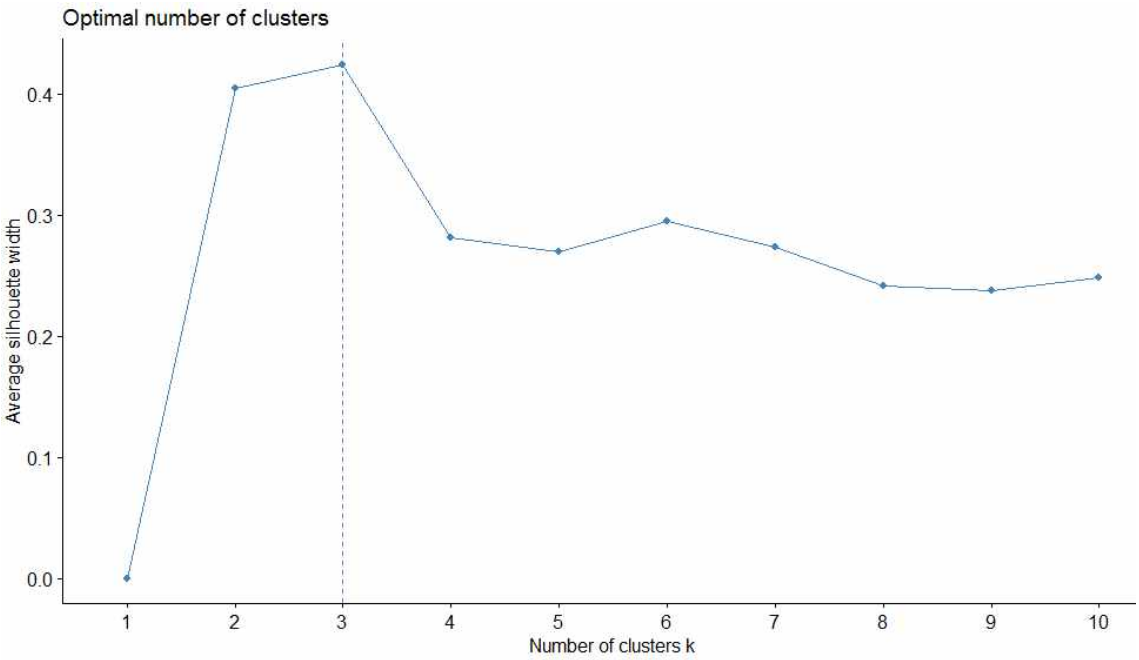
2. Results

(1) Average linkage와, complete linkage를 사용한 clustering의 결과이다. Average linkage를 사용한 경우 하나씩 cluster에 추가되는 결과가 관측되어, complete linkage를 사용하였다. 또한 distance measurement의 경우, expression.rds의 data는 gene의 발현정도를 수치화한 것이고, scale을 보았을 때 일반적인 euclidian distance를 사용해도 무방하다 판단하여, 이를 사용하였다.





다음으로 선택한 모델에 대해서 silhouette 상수를 이용하여 optimal number of cluster를 확인하였으며 그 결과는 3이었다. 그래프는 아래와 같다. 우리가 고른 최종적인 모델에 대해 jaccard index를 측정하였으며, 이는 0.2833826로 측정되었다.



주성분이 2개, 3개, 4개, 5개일 때 각각의 최적모델의 cluster의 수와 jaccard index는 다음과 같다.

Number of PC's	optimal number of clusters	jaccard index
2	2	0.7601319
3	3	0.4726448
4	2	0.5865333
5	4	0.4762259

jaccard index를 바탕으로 추론하였을 때, 주성분의 수가 3개이고, cluster의 개수가 3개일 때가 가장 최적모델이라 파악하였다.

PCA를 이용하여 dimension reduction을 진행하였을 때, jaccard index가 0.4726448로, 이를 수행하지 않았을 때의 jaccard index인 0.2833826보다 증가하여 나타났다. 즉 PCA를 이용하였을 때 similarity가 증가하였다. 이는 dimsenion reduction을 통해 noise response varaible가 미치는 영향을 효과적으로 줄여주었기 때문으로 추측된다.