

# 2023 SISBID Clustering Demo

Genevera I. Allen & Yufeng Liu

Load packages

```
library(ggplot2)
library(ISLR)
library(kknn)
```

## K-means Clustering

### 1. Data set 1 - Simulated Data

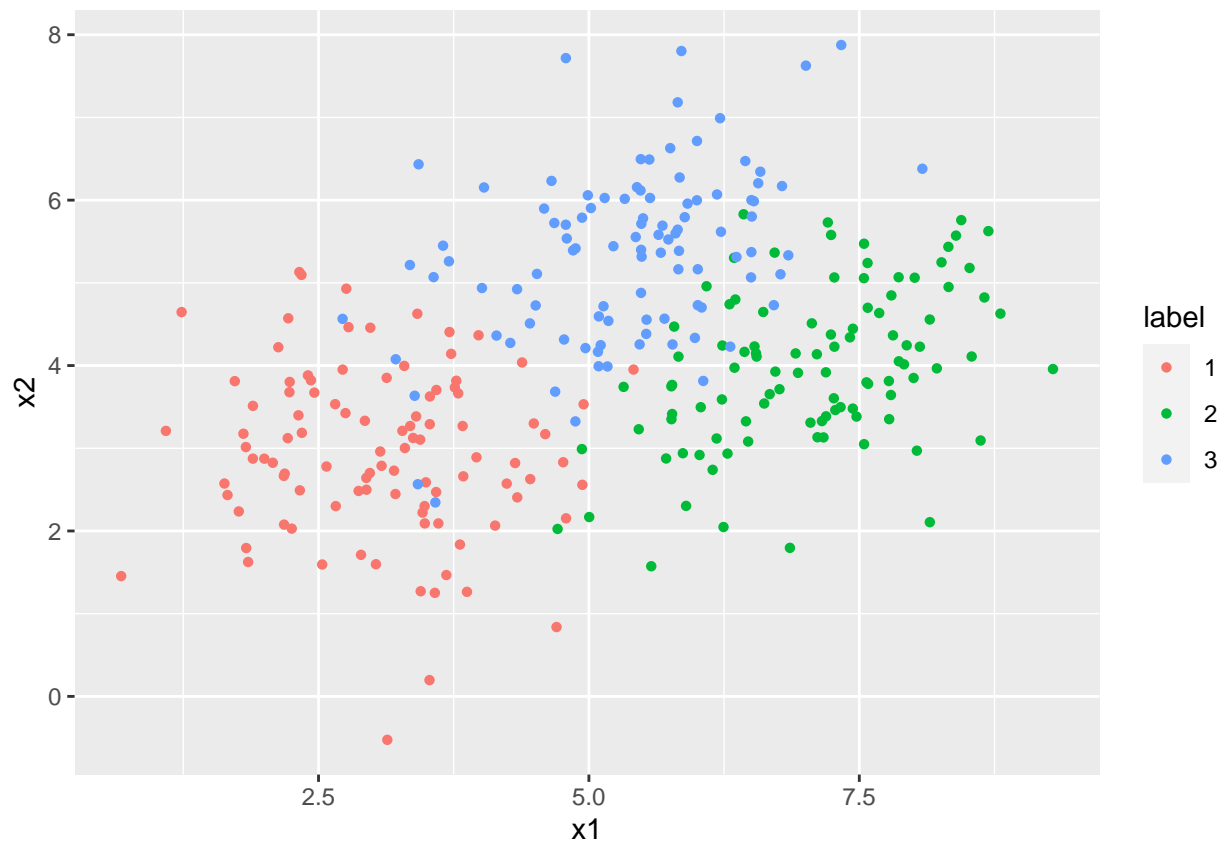
- small simulated data set to demonstrate concepts with k-means clustering

Simulate data: generate data from a mixture of three normal distribution

```
n = 300
mu1 = c(3,3)
mu2 = c(7,4)
mu3 = c(5.5,5.5)
Sig = matrix(c(1,.5,.5,1),2,2)
x1 = t(matrix(mu1,2,n/3)) + matrix(rnorm(n*2/3),n/3,2)
xx = matrix(rnorm(n*2/3),n/3,2)
x2 = t(matrix(mu2,2,n/3)) + xx%*%chol(Sig)
xx = matrix(rnorm(n*2/3),n/3,2)
x3 = t(matrix(mu3,2,n/3)) + xx%*%chol(Sig)
X = rbind(x1,x2,x3)
Y = c(rep(1,n/3),rep(2,n/3),rep(3,n/3))
Data = cbind(X,Y)
Data = data.frame(Data)
colnames(Data) = c("x1","x2","label")
Data$label = factor(Data$label)
```

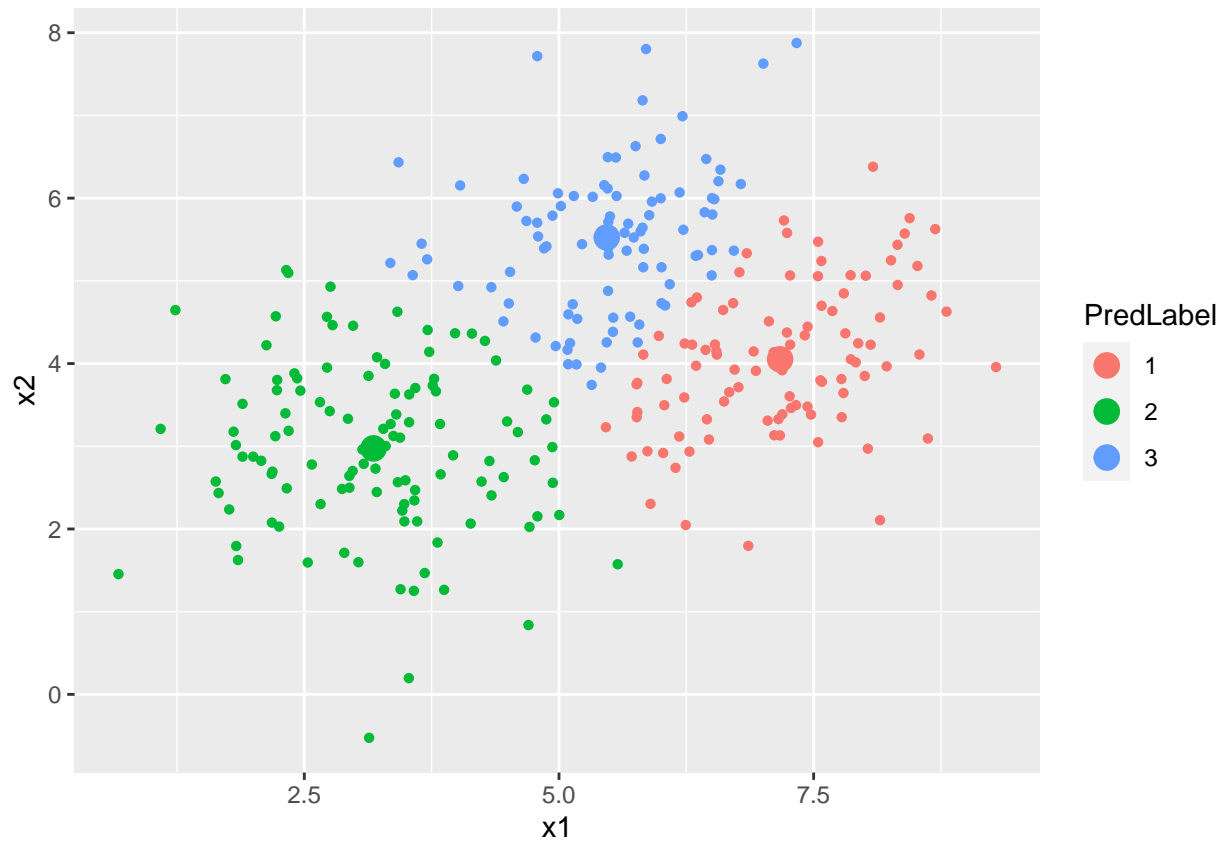
Plot with true labels

```
ggplot(data = Data) +
  geom_point(mapping = aes(x = x1,y = x2,color = label),pch = 16)
```



Apply k-means

```
k = 3
km = kmeans(X,centers=k)
gd = data.frame(km$centers)
gd$label = rownames(gd)
colnames(gd) = c("x1","x2","label")
Data$PredLabel = factor(km$cluster)
ggplot() +
  geom_point(data = Data,mapping = aes(x = x1,y = x2,color = PredLabel), pch = 16) +
  geom_point(gd,mapping = aes(x = x1,y = x2,color= factor(label)),size = 4)
```



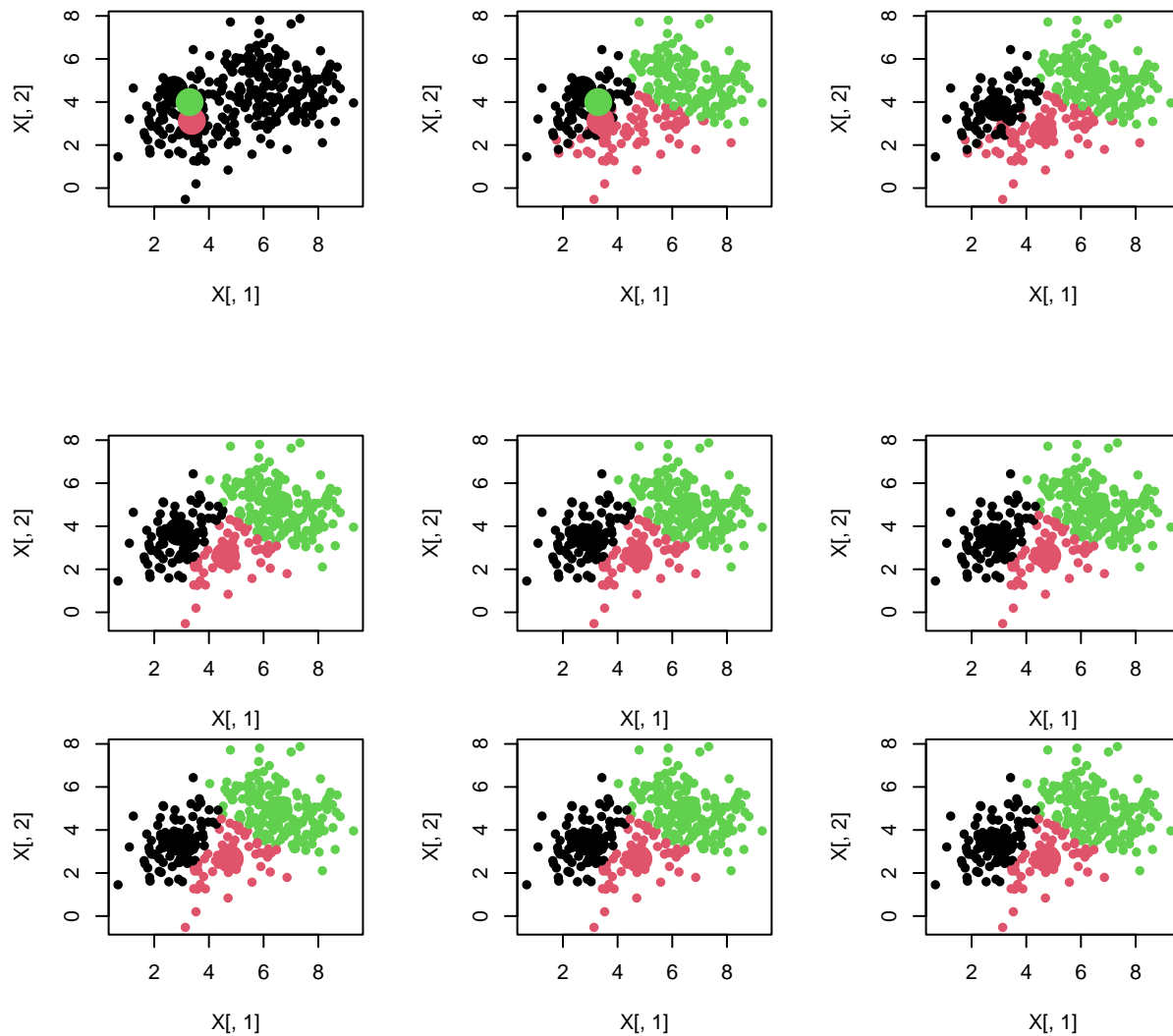
Code to understand K-means algorithm: raw code for k-means

```
par(mfrow=c(2,3))
k = 3
n = nrow(X)
cens = X[sample(1:n,k),]
plot(X[,1],X[,2],pch=16)
points(cens[,1],cens[,2],col=1:k,pch=16,cex=3)
for(i in 1:4){
  oldcen = cens
  km = kmeans(X,centers=cens,iter.max=1,nstart=1,algorithm="MacQueen")
  plot(X[,1],X[,2],col=km$cluster,pch=16)
  points(cens[,1],cens[,2],col=1:k,pch=16,cex=3)
  cens = km$centers
  plot(X[,1],X[,2],col=km$cluster,pch=16)
  points(cens[,1],cens[,2],col=1:k,pch=16,cex=3)
  ind = sum(diag((oldcen-cens)%*%t(oldcen-cens)))
}
```

```
## Warning: did not converge in 1 iteration
```

```
## Warning: did not converge in 1 iteration
```

```
## Warning: did not converge in 1 iteration
```



2. Data set 2 - NCI Microarray data: The data contains expression levels on 6830 genes from 64 cancer cell lines. Cancer type is also recorded.

- Apply K-means to cluster a high-dimensional data set.
- Apply hierarchical clustering & try out different linkages.
- Apply biclustering (Cluster heatmap) to visualize data.

```
ncidat = NCI60$data
rownames(ncidat) = NCI60$labs # cancer type
dim(ncidat)
```

```
## [1] 64 6830
```

```
table(NCI60$labs)
```

```
##
##      BREAST      CNS      COLON K562A-repro K562B-repro      LEUKEMIA
##          7        5          7          1          1          6
## MCF7A-repro MCF7D-repro      MELANOMA      NSCLC      OVARIAN      PROSTATE
##          1          1          8          9          6          2
```

```
##          RENAL          UNKNOWN
##          9             1
```

Apply K-means

```
K = 9
km = kmeans(ncidat,centers=K)
```

How do we visualize K-means results?

PCA - take SVD to get solution

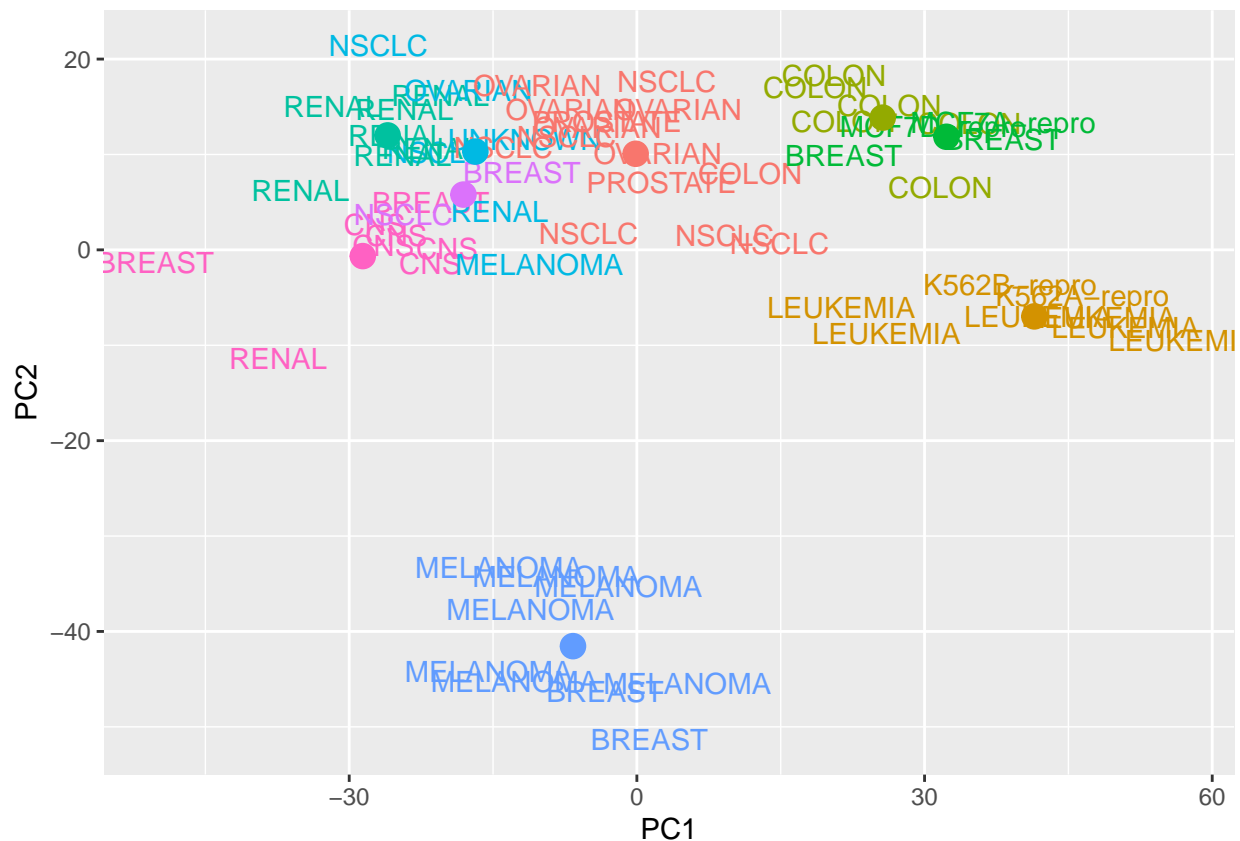
Center genes, but don't scale

```
X = scale(ncidat,center=TRUE,scale=FALSE)
sv = svd(X)
U = sv$u
V = sv$v
D = sv$d
Z = X%*%V
```

Visualization

```
# projected data
PCData = data.frame(cbind(Z[,1],Z[,2],km$cluster,NCI60$labs),stringsAsFactors = FALSE)
colnames(PCData) = c("PC1","PC2","PredLabel","CancerType")
PCData$PC1 = as.numeric(PCData$PC1)
PCData$PC2 = as.numeric(PCData$PC2)
# projected k-means centers
GroupData = data.frame(km$centers%*%V[,1:2])
GroupData$label = rownames(GroupData)
colnames(GroupData) = c("PC1","PC2","PredLabel")

ggplot(PCData,mapping=aes(x = PC1,y= PC2,color = PredLabel)) +
  geom_text(mapping=aes(label = CancerType)) +
  geom_point(data = GroupData,size = 4) +
  theme(legend.position="none")
```

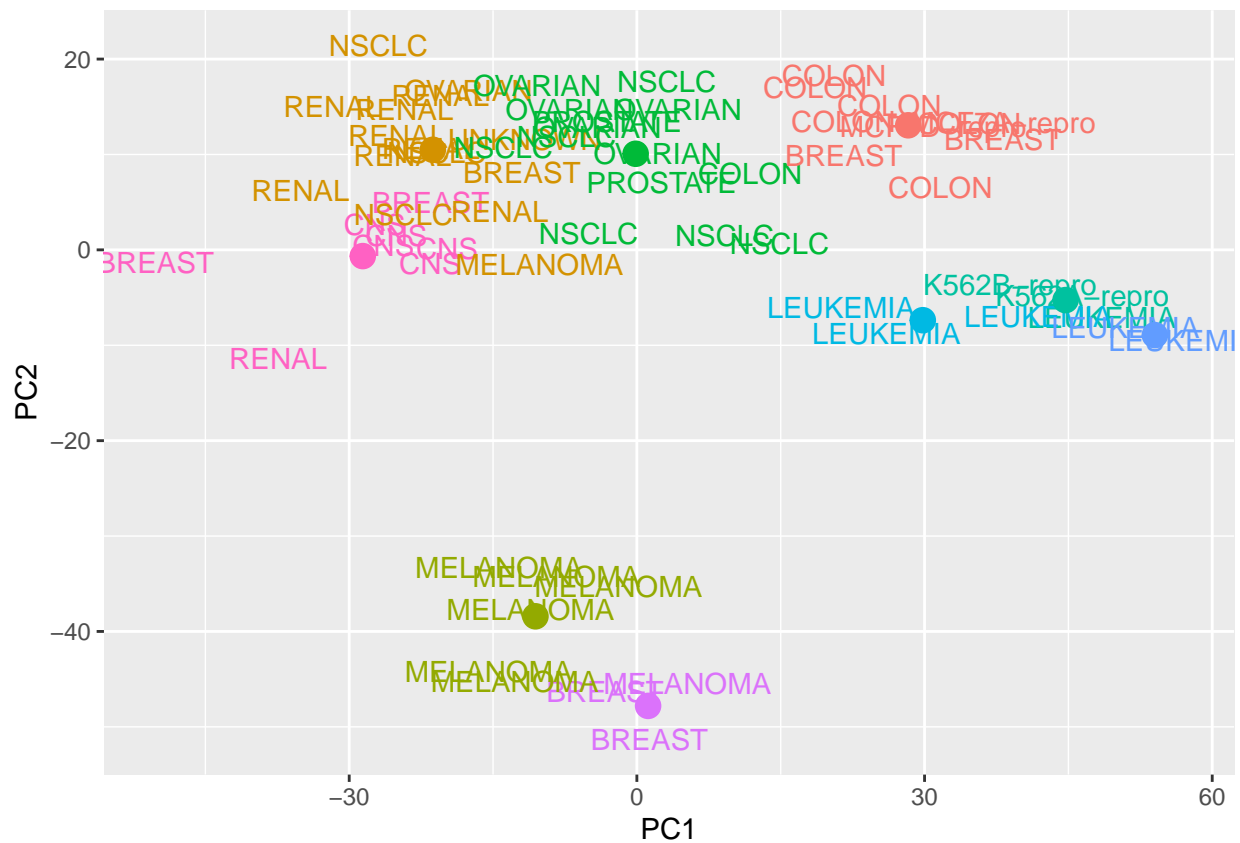


Re-run and see if solution changes

```
K = 9
km = kmeans(ncidat,centers=K)
PCData$PredLabel = as.factor(km$cluster)

# projected k-means centers
GroupData = data.frame(km$centers%*%V[,1:2])
GroupData$label = rownames(GroupData)
colnames(GroupData) = c("PC1", "PC2", "PredLabel")

# plot
ggplot(PCData, mapping=aes(x = PC1, y = PC2, color = PredLabel)) +
  geom_text(mapping=aes(label = CancerType)) +
  geom_point(data = GroupData, size = 4) +
  theme(legend.position="none")
```

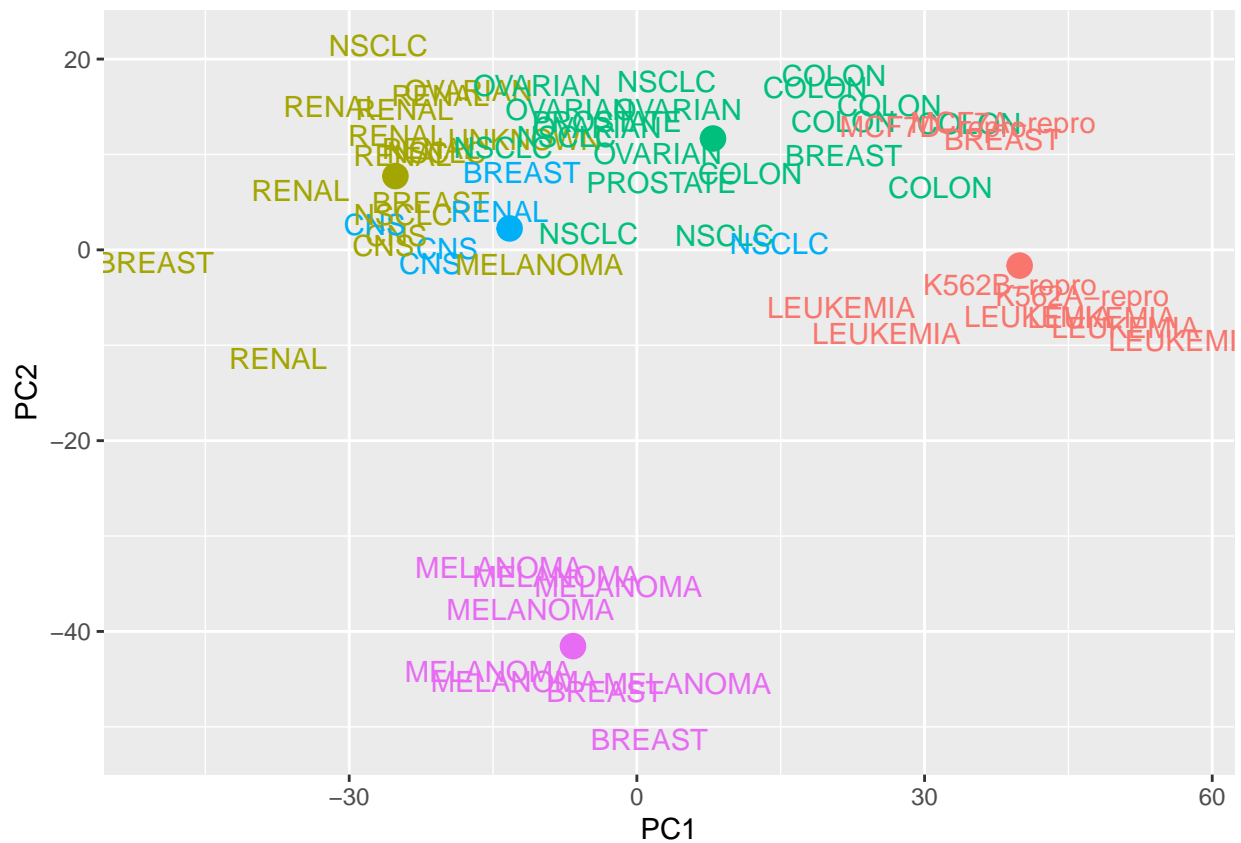


Try different K

```
K = 5
km = kmeans(ncidat,centers=K)
PCData$PredLabel = as.factor(km$cluster)

# projected k-means centers
GroupData = data.frame(km$centers%*%V[,1:2])
GroupData$label = rownames(GroupData)
colnames(GroupData) = c("PC1", "PC2", "PredLabel")

# plot
ggplot(PCData,mapping=aes(x = PC1,y= PC2,color = PredLabel)) +
  geom_text(mapping=aes(label = CancerType)) +
  geom_point(data = GroupData,size = 4) +
  theme(legend.position="none")
```



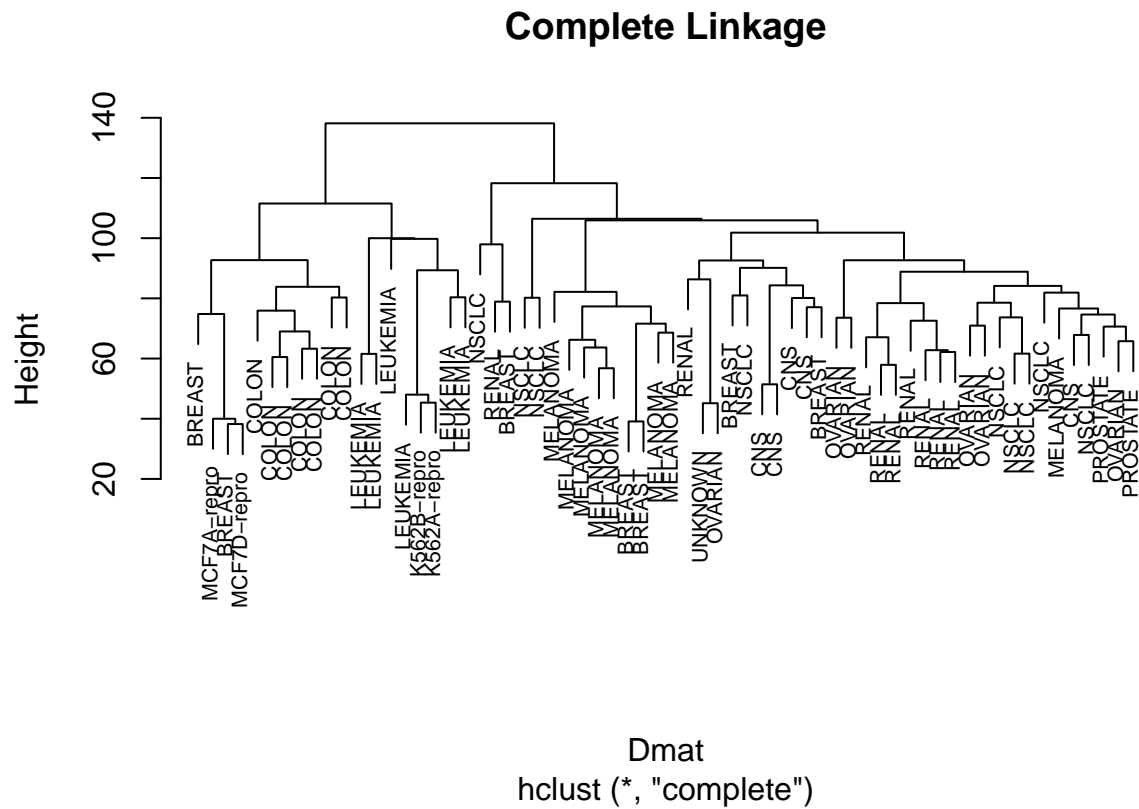
### Hierarchical clustering

Real Data: NCI 60 data in ISLR package

Complete linkage - Euclidean distance

```
cols = as.numeric(as.factor(rownames(ncidat)))
Dmat = dist(ncidat)
com.hclust = hclust(Dmat,method="complete")
plot(com.hclust,cex=.7,main="Complete Linkage")
```

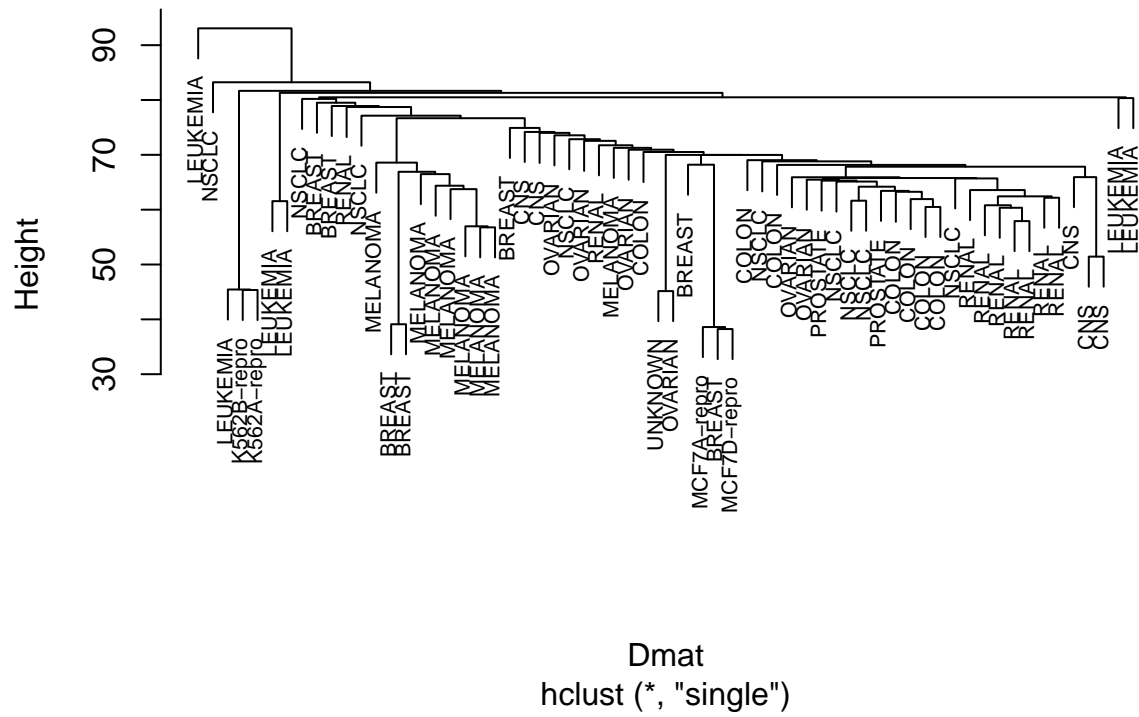




Single linkage

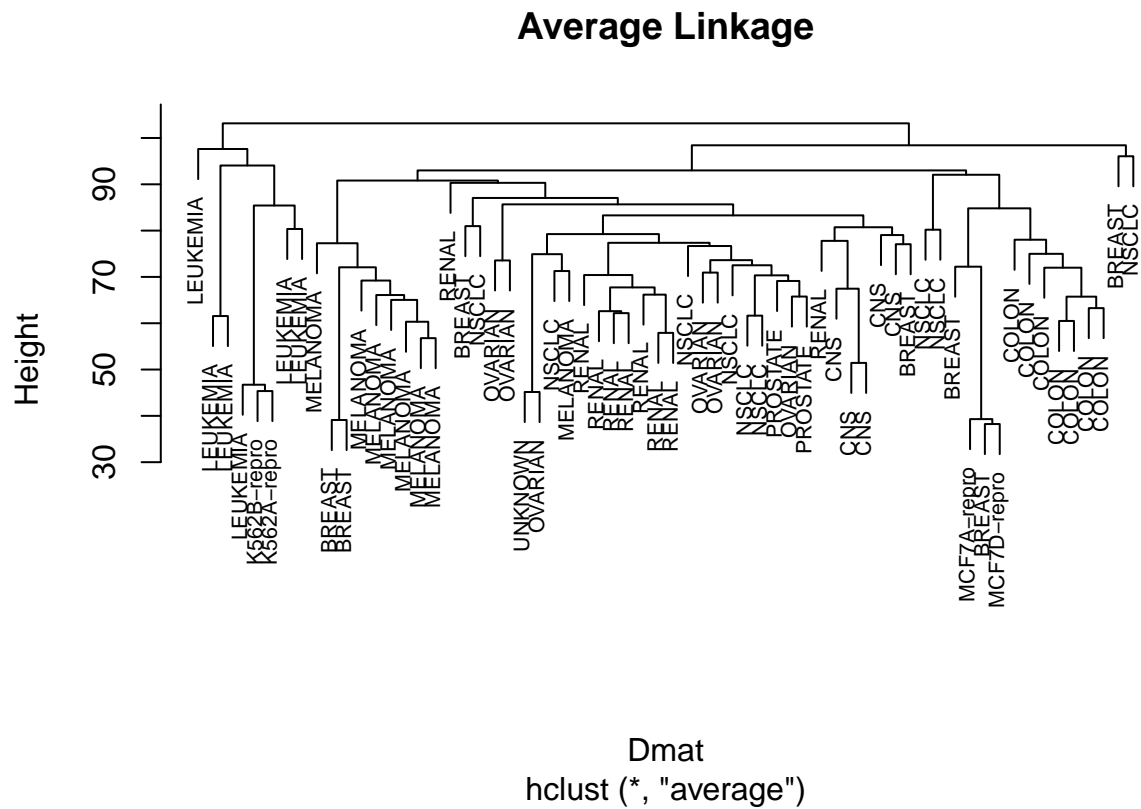
```
sing.hclust = hclust(Dmat,method="single")
plot(sing.hclust,cex=.7,main="Single Linkage")
```

## Single Linkage



Average linkage

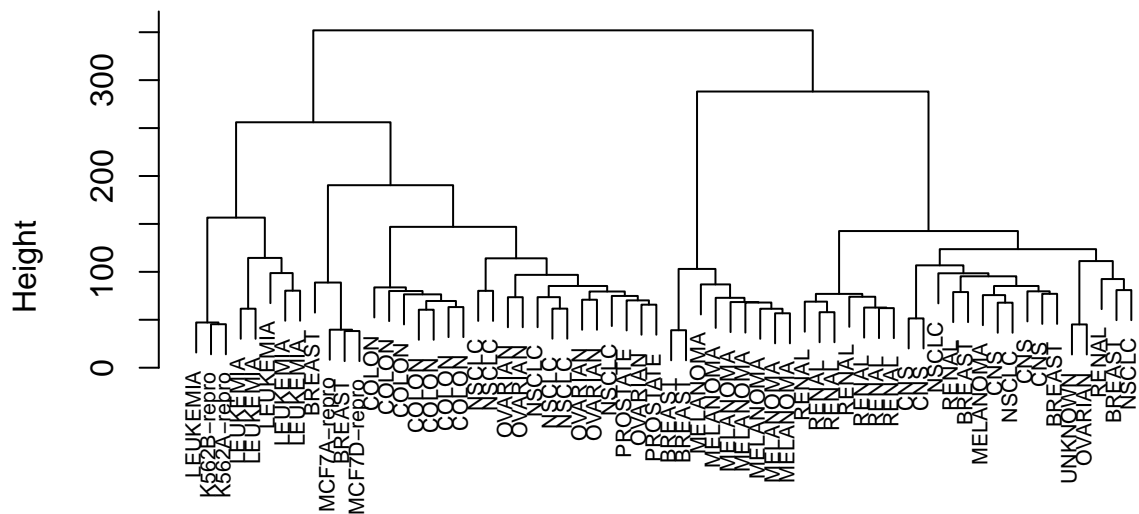
```
ave.hclust = hclust(Dmat,method="average")
plot(ave.hclust,cex=.7,main="Average Linkage")
```



Ward's linkage

```
ward.hclust = hclust(Dmat,method="ward.D")
plot(ward.hclust,cex=.7,main="Ward's Linkage")
```

## Ward's Linkage

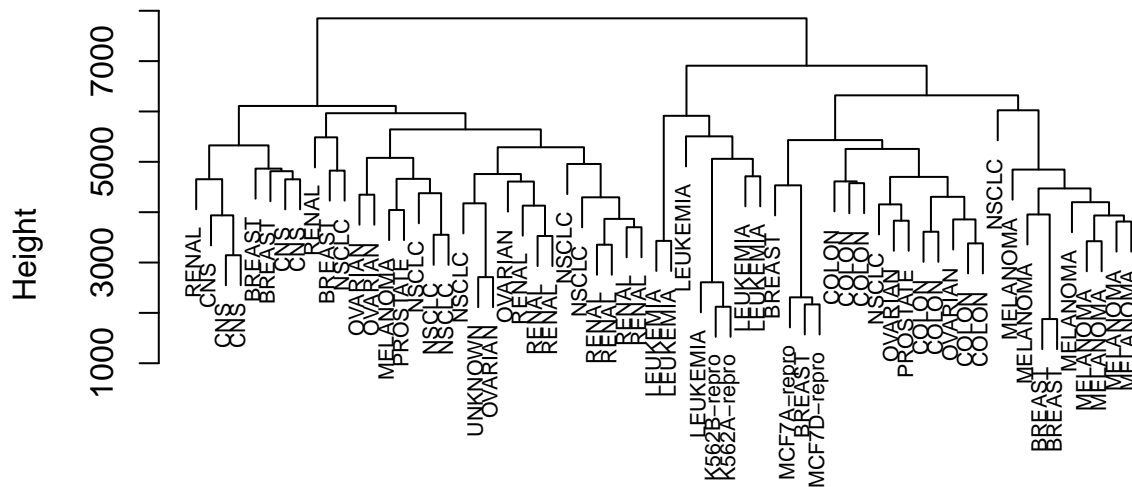


```
Dmat
hclust (*, "ward.D")
```

Complete linkage with different distances - L1 distance

```
Dmat = dist(ncidat,method="manhattan") #L1 distance
com.hclust = hclust(Dmat,method="complete")
plot(com.hclust,cex=.7,main="Complete Linkage - L1 Dist")
```

## Complete Linkage - L1 Dist



Dmat  
hclust (\*, "complete")

## Biclustering - Cluster Heatmap

Filter genes using PCA

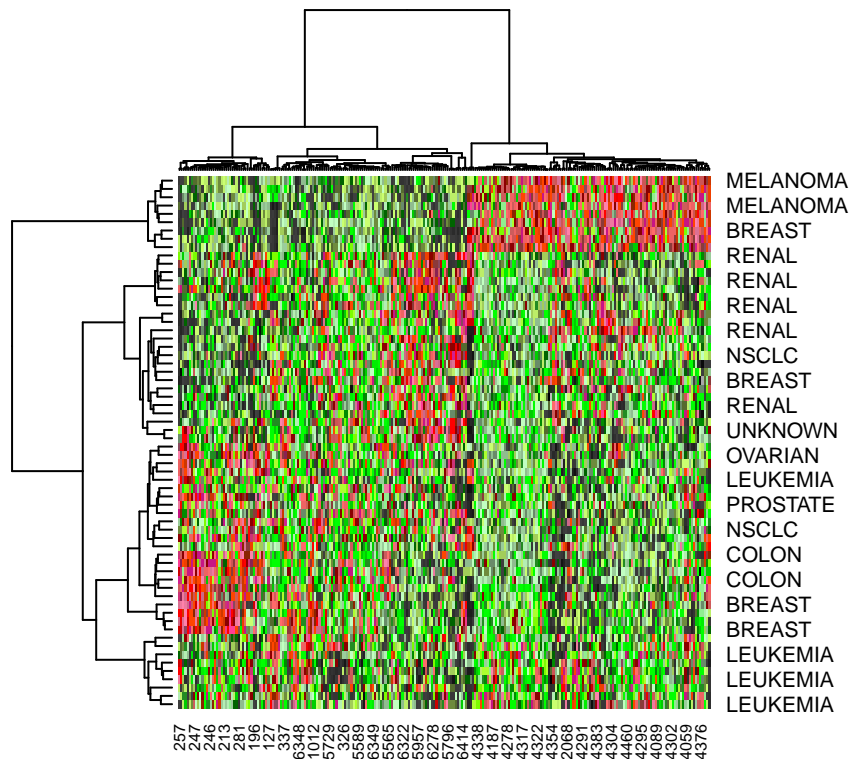
PC loadings - visualize data by limiting to top genes in magnitude in the PC loadings

```
aa = grep("grey",colors())
bb = grep("green",colors())
cc = grep("red",colors())
gcol2 = colors()[c(aa[1:30],bb[1:20],rep(cc,2))]
```

```
j = 2
ord = order(abs(V[,j]),decreasing=TRUE)
x = as.matrix(X[,ord[1:250]])
```

#cluster heatmap - uses Ward's linkage (complete is default)

```
heatmap(x,col=gcol2,hclustfun=function(x) hclust(x,method="ward.D"))
```



## Spectral clustering

```
K = 9
SC_NCI = specClust(ncidat, centers=K, nn = 7, method = "symmetric", gmax=NULL)
```

### Visualization

```
X = scale(ncidat,center=TRUE,scale=FALSE)
sv = svd(X)
U = sv$u
V = sv$v
D = sv$d
Z = X%*%V

# projected data
SCData = data.frame(cbind(Z[,1],Z[,2],SC_NCI$cluster,NCI60$labs),stringsAsFactors = FALSE)
colnames(SCData) = c("PC1","PC2","PredLabel","CancerType")
SCData$PC1 = as.numeric(SCData$PC1)
SCData$PC2 = as.numeric(SCData$PC2)

# plot
ggplot(SCData,mapping=aes(x = PC1,y= PC2,color = PredLabel)) +
  geom_text(mapping=aes(label = CancerType)) +
  theme(legend.position="none")
```

