Lab 15

quan le

2022-10-13

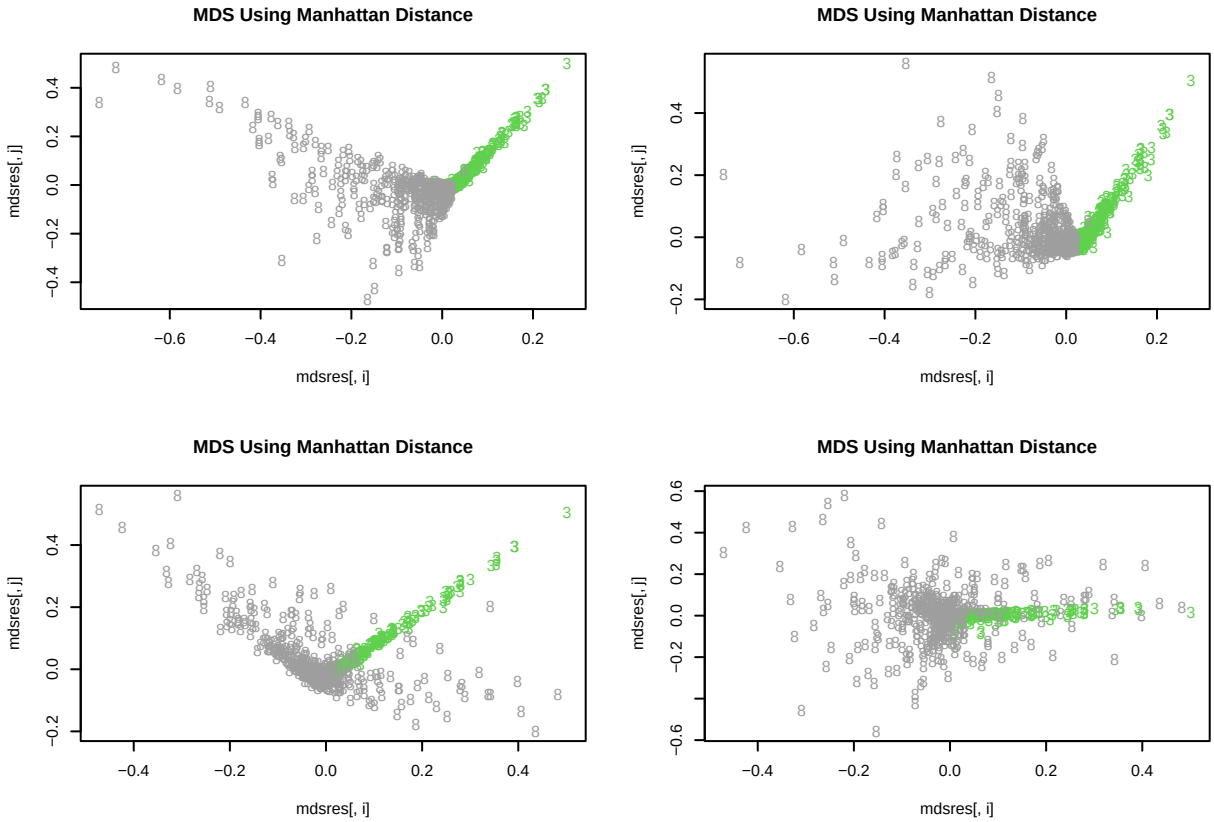
```
load("data/chu.rdata")
hiseq = readRDS("data/hiseq.rds")

chu.x = t(chu$sc_cnt)
chu.y = chu$sc_label
chu.colors = as.integer(factor(chu.y))
```

Multidimensional Scaling (MDS)

classical MDS (Note, this may take some time - try only on 3's and 8's)

```
#MDS
Dmat = dist(dat38,method="maximum") #Manhattan (L1) Distance
mdsres = cmdscale(Dmat,k=10)
```

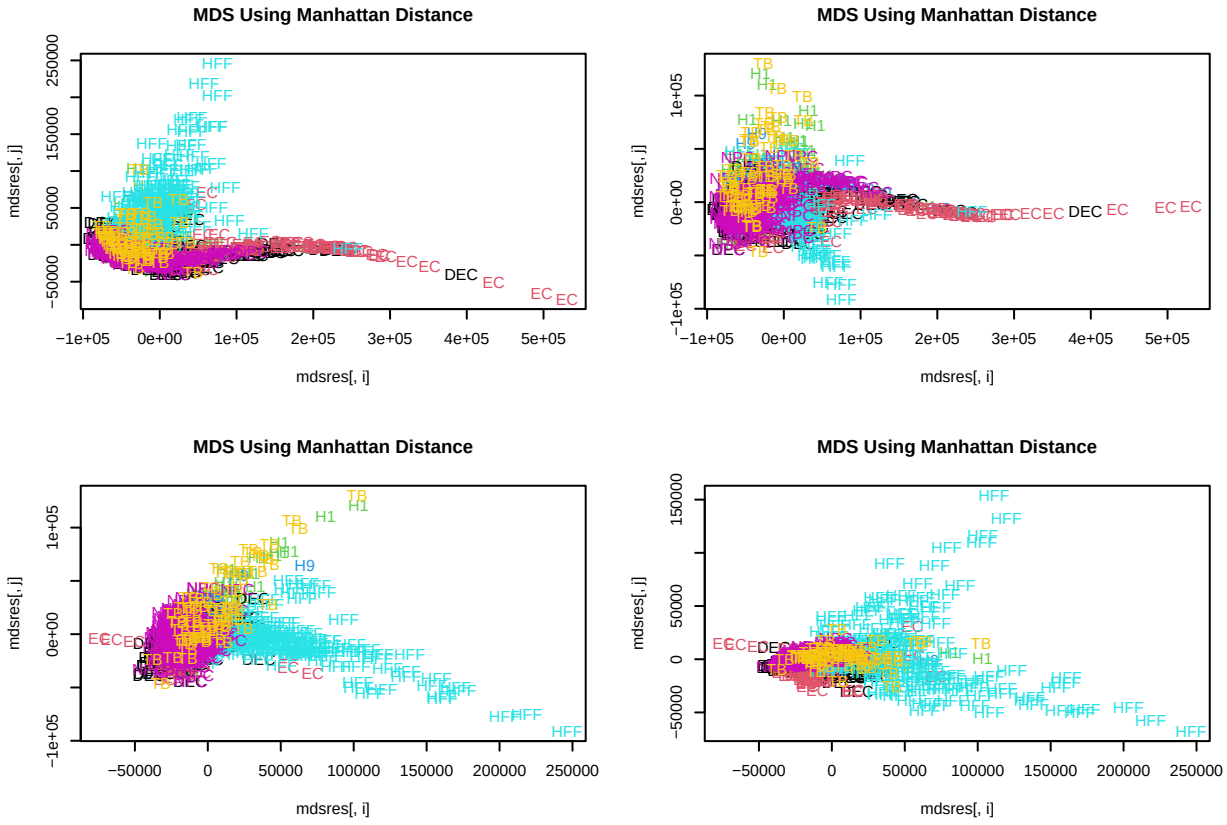
```
par(mfrow=c(2,2), cex=0.5)

i = 1; j = 2;
plot(mdsres[,i],mdsres[,j],type="n", main = "MDS Using Manhattan Distance")
text(mdsres[,i],mdsres[,j],rownames(dat38),col=rownames(dat38))

i = 1; j = 3;
plot(mdsres[,i],mdsres[,j],type="n", main = "MDS Using Manhattan Distance")
text(mdsres[,i],mdsres[,j],rownames(dat38),col=rownames(dat38))

i = 2; j = 3;
plot(mdsres[,i],mdsres[,j],type="n", main = "MDS Using Manhattan Distance")
text(mdsres[,i],mdsres[,j],rownames(dat38),col=rownames(dat38))

i = 2; j = 4;
plot(mdsres[,i],mdsres[,j],type="n", main = "MDS Using Manhattan Distance")
text(mdsres[,i],mdsres[,j],rownames(dat38),col=rownames(dat38))
```



cell data

```
#MDS
Dmat = dist(chu.x,method="maximum") #Manhattan (L1) Distance
mdsres = cmdscale(Dmat,k=5)
```

```
par(mfrow=c(2,2), cex=0.5)

i = 1; j = 2;
plot(mdsres[,i],mdsres[,j],type="n", main = "MDS Using Manhattan Distance")
text(mdsres[,i],mdsres[,j],chu.y,col=chu.colors)

i = 1; j = 3;
plot(mdsres[,i],mdsres[,j],type="n", main = "MDS Using Manhattan Distance")
text(mdsres[,i],mdsres[,j],chu.y,col=chu.colors)

i = 2; j = 3;
plot(mdsres[,i],mdsres[,j],type="n", main = "MDS Using Manhattan Distance")
text(mdsres[,i],mdsres[,j],chu.y,col=chu.colors)

i = 2; j = 4;
plot(mdsres[,i],mdsres[,j],type="n", main = "MDS Using Manhattan Distance")
text(mdsres[,i],mdsres[,j],chu.y,col=chu.colors)
```



Uniform Manifold Approximation and Projection (UMAP)

Load Packages

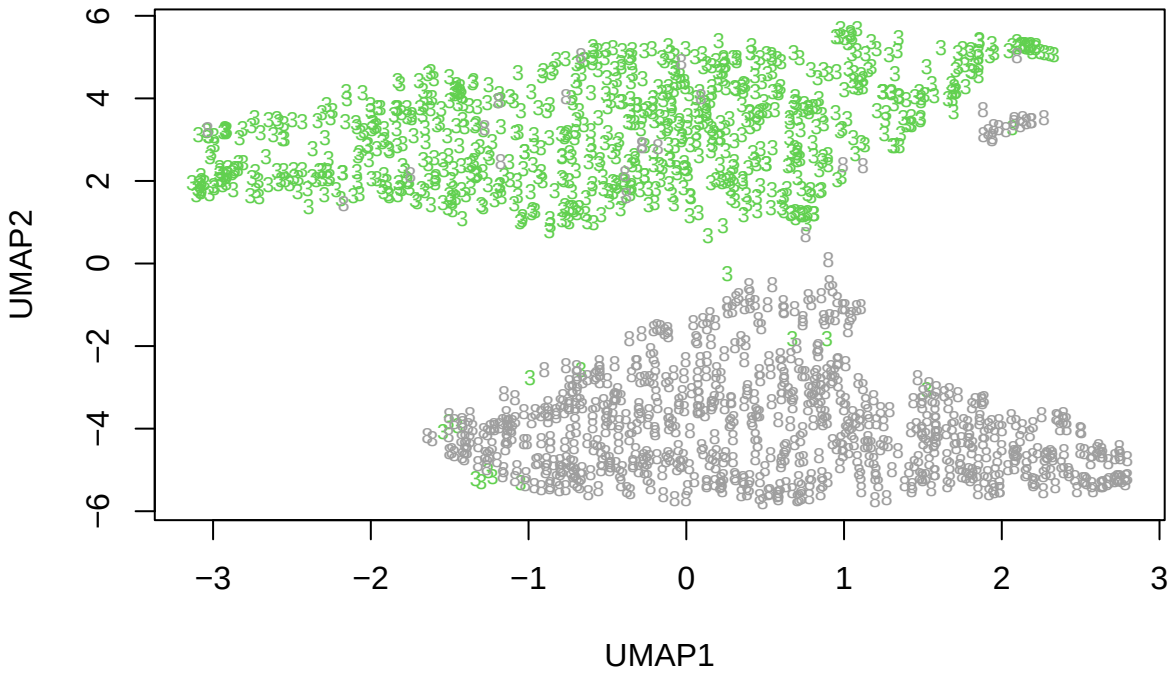
```
library(umap)
library(Rtsne)
```

Run UMAP

```
digits.umap = umap(dat38)
```

Plot UMAP

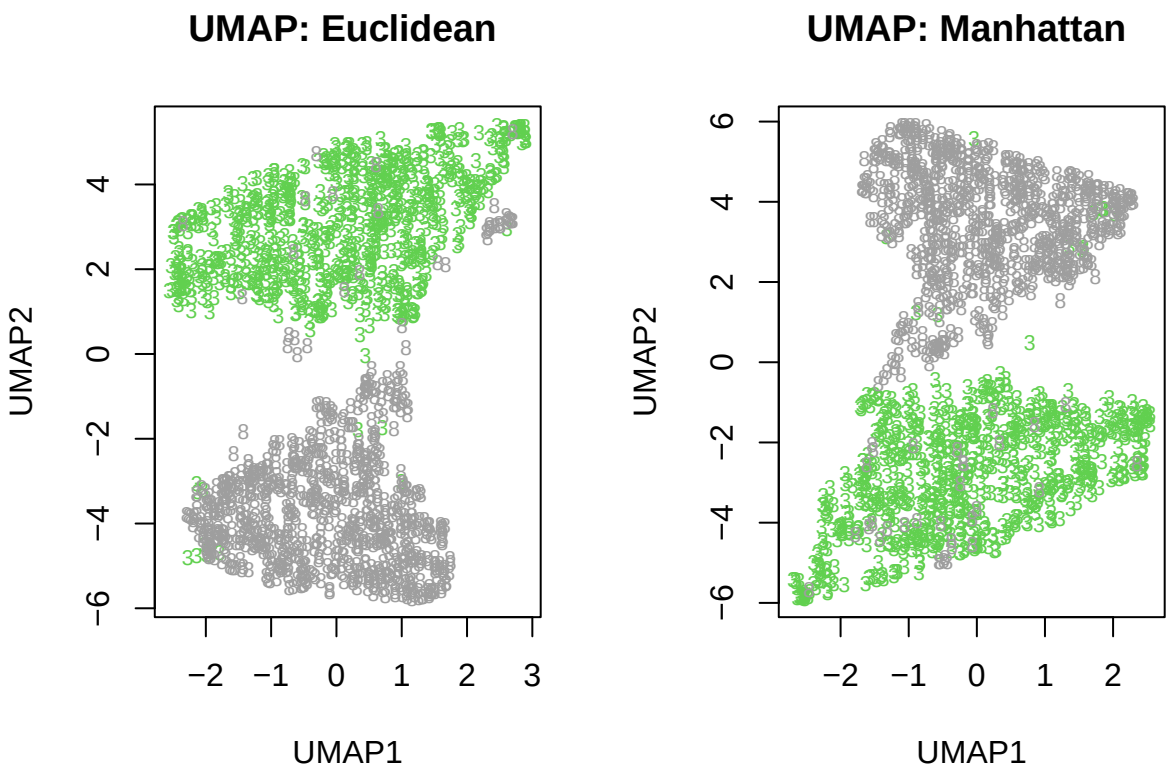
```
plot(digits.umap$layout[,1],y=digits.umap$layout[,2], type ='n', main = "UMAP on Digits 3,8 ", xlab = "U
text(digits.umap$layout[,1],y=digits.umap$layout[,2],rownames(dat38),col=rownames(dat38),cex=.7)
```



Try

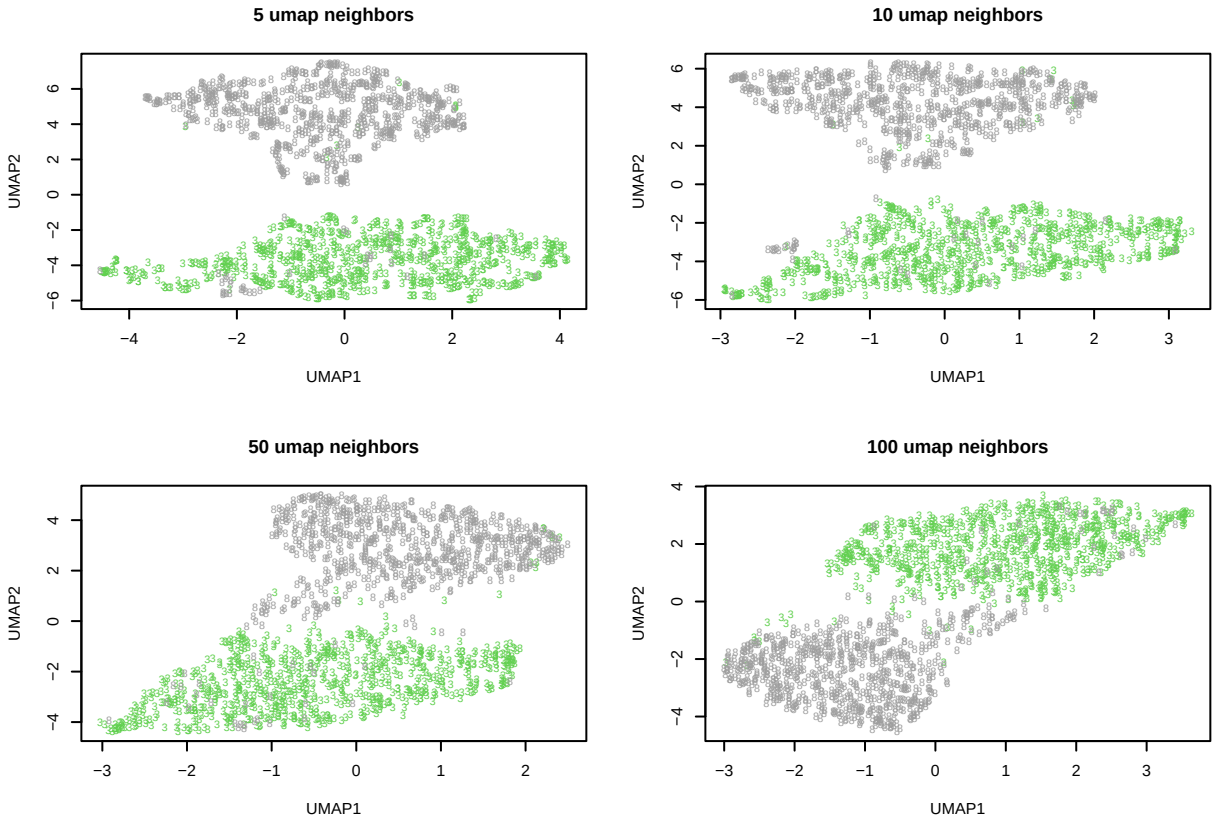
Changing Hyperparameters

```
digits.umap1 = umap(dat38,metric="euclidean")
digits.umap2 = umap(dat38,metric="manhattan")


par(mfrow=c(1,2))
plot(digits.umap1$layout[,1],y=digits.umap1$layout[,2], type ='n', main = "UMAP: Euclidean ", xlab = "U
text(digits.umap1$layout[,1],y=digits.umap1$layout[,2],rownames(dat38),col=rownames(dat38),cex=.7)

plot(digits.umap2$layout[,1],y=digits.umap2$layout[,2], type ='n', main = "UMAP: Manhattan ", xlab = "U
text(digits.umap2$layout[,1],y=digits.umap2$layout[,2],rownames(dat38),col=rownames(dat38),cex=.7)
```




```
nneighs = c(5,10,50,100)
digits.umap = list()
for(i in nneighs){
  digits.umap[[i]] = umap(dat38,n_neighbors=i)
}
```

```
par(mfrow=c(2,2), cex=0.5)
for(i in nneighs){
  plot(digits.umap[[i]]$layout[,1],y=digits.umap[[i]]$layout[,2], type ='n', xlab = "UMAP1", ylab = "UMA
  text(digits.umap[[i]]$layout[,1],y=digits.umap[[i]]$layout[,2],rownames(dat38),col=rownames(dat38),ce
}
```

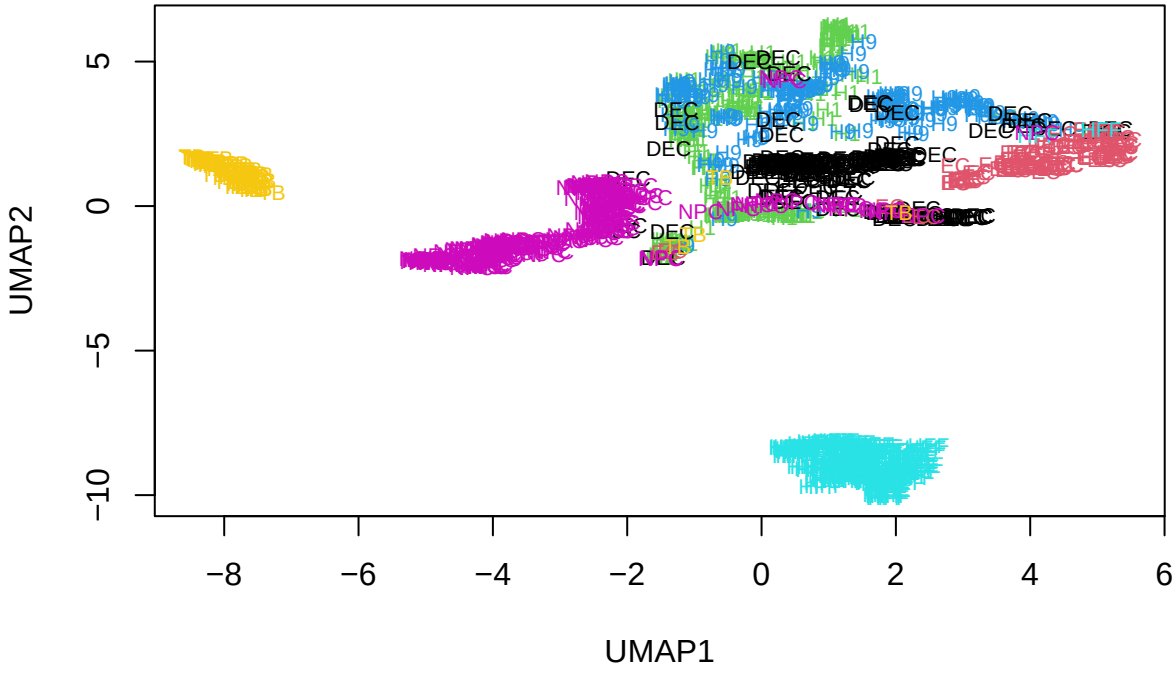


cell data

```
chu.umap = umap(chu.x)
```

Plot UMAP

```
plot(chu.umap$layout[,1],y=chu.umap$layout[,2], type ='n', main = "UMAP on cells", xlab = "UMAP1", ylab
text(chu.umap$layout[,1],y=chu.umap$layout[,2],chu.y,col=chu.colors,cex=.7)
```

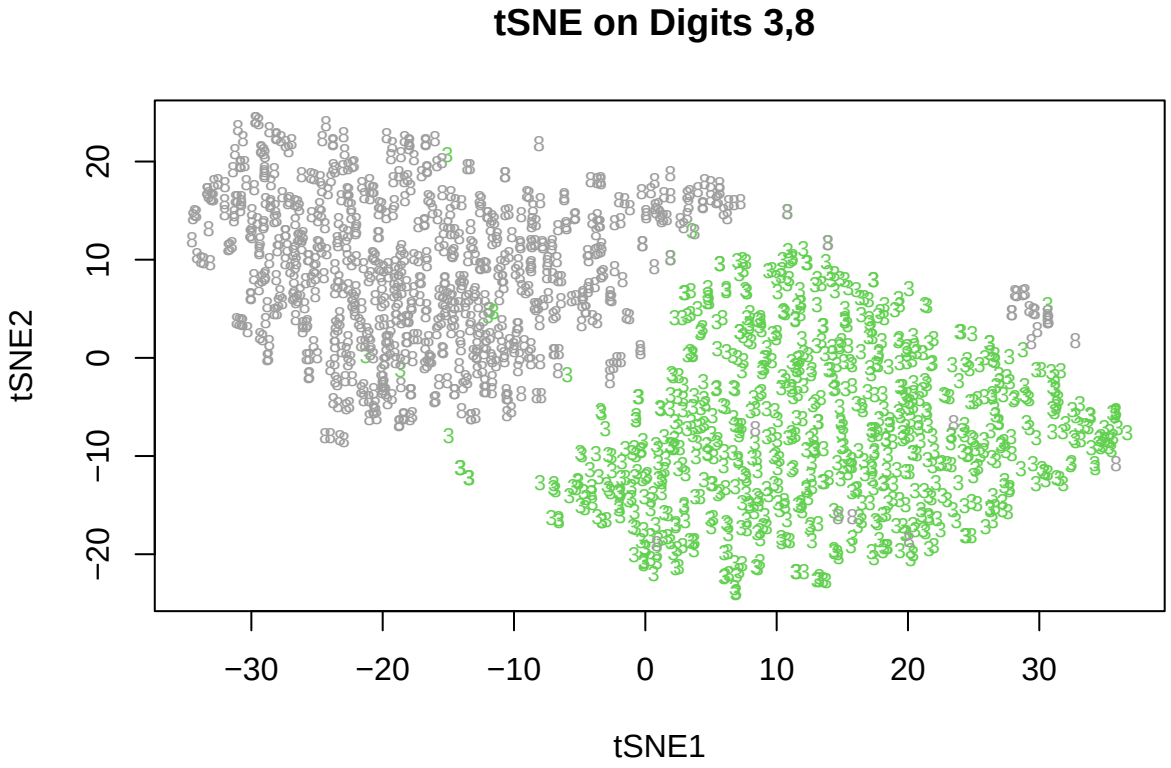


t-distributed stochastic neighbor embedding (tSNE)

Run tSNE

```
tsne_digit <- Rtsne(as.matrix(dat38))
```

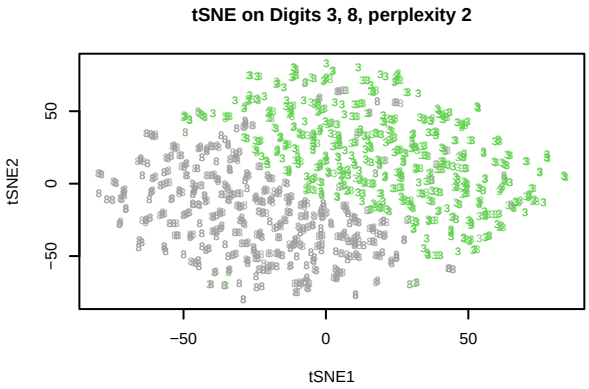
```
plot(tsne_digit$Y[,1],y=tsne_digit$Y[,2], type ='n', main = "tSNE on Digits 3,8 ", xlab = "tSNE1", ylab
text(tsne_digit$Y[,1],y=tsne_digit$Y[,2],rownames(dat38),col=rownames(dat38),cex=.7)
```

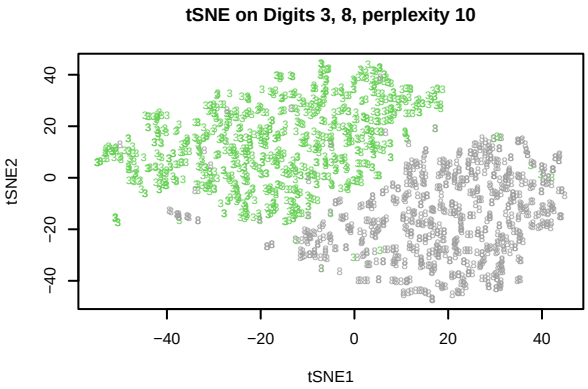


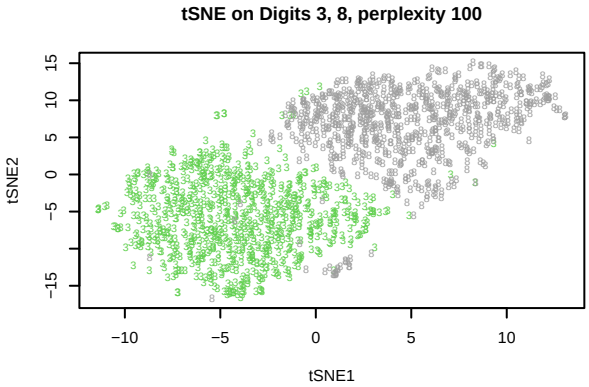
Try Changing Hyperparameters (Perplexity)

```
perps = c(2,10,100)
tsne_digit = list()
for(i in perps){
  tsne_digit[[i]] <- Rtsne(as.matrix(dat38),perplexity=i)
}
```

```
par(mfrow=c(2,2), cex=0.5)
for(i in perps){
  plot(tsne_digit[[i]]$Y[,1],y=tsne_digit[[i]]$Y[,2], type ='n', main = paste("tSNE on Digits 3, 8, perp
  text(tsne_digit[[i]]$Y[,1],y=tsne_digit[[i]]$Y[,2],rownames(dat38),col=rownames(dat38),cex=.7)
}
```



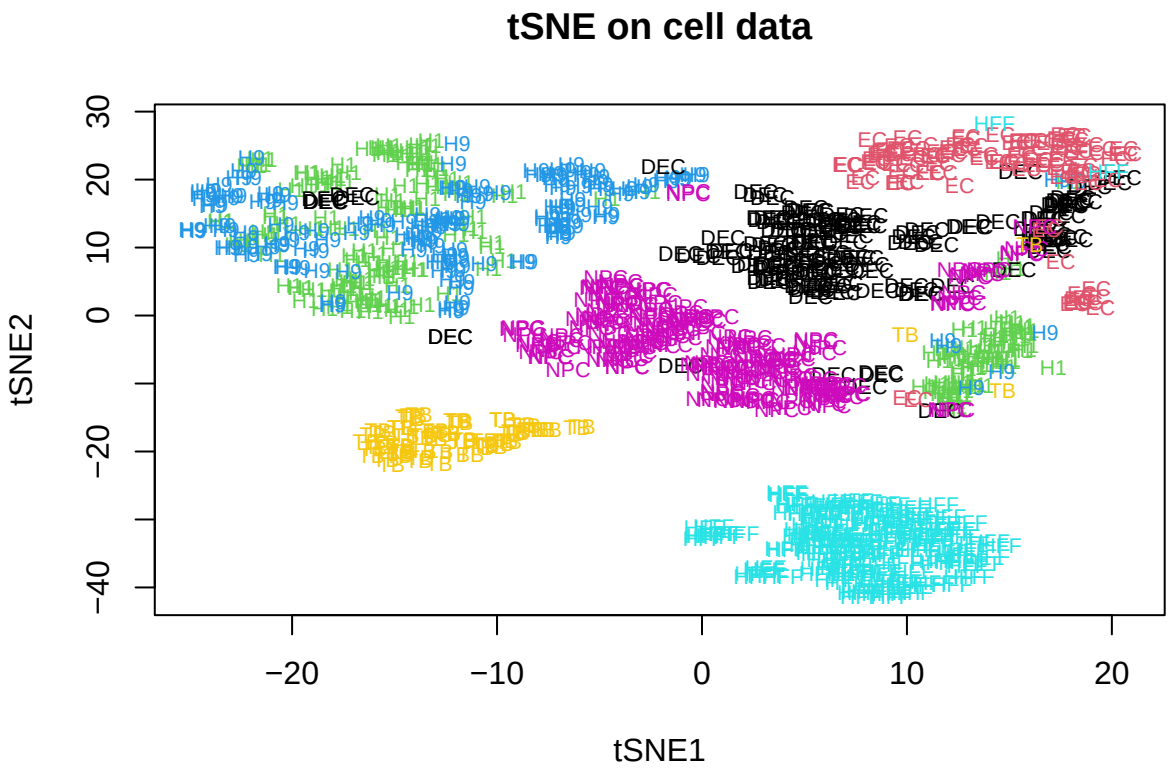




cell data

```
chu.tsne <- Rtsne(as.matrix(chu.x))
```

```
plot(chu.tsne$Y[,1],y=chu.tsne$Y[,2], type ='n', main = "tSNE on cell data", xlab = "tSNE1", ylab = "tS
text(chu.tsne$Y[,1],y=chu.tsne$Y[,2],chu.y,col=chu.colors,cex=.7)
```


tSNE on cell data
tSNE2
tSNE1
30
20
10
0
−20
−40
−20
−10
0
10
20
DEC
NPC
EC
HEF
TB
H1
H9