CCA on Halper & Yang

Nick Wawee

August 27, 2018

Loading

```
#Loading
loadmaty<-read.table("Yang_ReadCount_ERCC_Converted.txt", fill = TRUE, stringsAsFactors = FALSE)
loadmaty2<-read.table("pseudotimemat.txt", stringsAsFactors = FALSE)
zonatedmat<-read.table("OrderedPtMatrix_Filtered_1.txt", stringsAsFactors = FALSE)
zonatedmat<-as.data.frame(zonatedmat)

#Feature Data
zgenes<-rownames(zonatedmat)
zonatedmat<-cbind(zgenes,zonatedmat)
zonatedmat$zgenes<-as.character(zonatedmat$zgenes)
zonatedmat2<-zonatedmat %>%
  separate_rows(zgenes, sep=";")
row.names(zonatedmat2)<-zonatedmat2[,1]

# Testing separte_rows ---------------------------------
testexp<-zonatedmat2[c(which(rownames(zonatedmat2)=="Apr3"),which(rownames(zonatedmat2)=="0610007C21Rik")),]
testexp2<-zonatedmat[which(rownames(zonatedmat)=="0610007C21Rik;Apr3"),]
#View(rbind(testexp,testexp2))
#---------------------------------------------------------------------------------------------------------------------
-----------------------

loadmaty[,2]<-as.character(loadmaty[,2])
matchind<-match(rownames(zonatedmat2),loadmaty[,2])
matchind<-matchind[-which(is.na(matchind))]
fdmat<-as.matrix(loadmaty[matchind,1:2])
row.names(fdmat)<-fdmat[,1]
fdmat<-as.matrix(fdmat[,-1])
colnames(fdmat)<-"gene_short_name"
fd<-as.data.frame(fdmat)
fd<-new("AnnotatedDataFrame", data=fd)

#Pheno Data
#Yang
cellnamesy<-as.matrix(colnames(loadmaty2)[which(loadmaty2[2,]=="Hepatoblast"|loadmaty2[2,]=="Hepatocyte")])
colnames(cellnamesy)<-cellnamesy[1,]
cellnamesy<-cbind(cellnamesy,rep("Embryonic",length(cellnamesy[,1])))
colnames(cellnamesy)<-c("Cell_Name", "Cell_Type")
row.names(cellnamesy)<-cellnamesy[,1]
pt<-loadmaty2[1,match(rownames(cellnamesy),colnames(loadmaty2))]
pt<-as.numeric(pt)
pt<-scalefun(pt, "0and1")
statey<-loadmaty2[2,match(rownames(cellnamesy),colnames(loadmaty2))]
statey<-t(statey)
colnames(statey)<-"State0"
statey2<-statey
colnames(statey2)<-"State2"
cellnamesy<-cbind(cellnamesy,pt,statey,statey2)

#Zonation
cellnamesz<-as.matrix(colnames(zonatedmat2)[2:length(zonatedmat2)])
row.names(cellnamesz)<-cellnamesz
cellnamesz<-cbind(cellnamesz, rep("Zonation",length(cellnamesz)))
colnames(cellnamesz)<-c("Cell_Name", "Cell_Type")
#cellnamesz<-cellnamesz[-length(cellnamesz[,1]),]
cellnamesz<-cbind(cellnamesz,t(zonatedmat2[c(1,4,5),-1]))
cellnamesz[,3]<-scalefun(as.numeric((cellnamesz[,3])),"0and1")
colnames(cellnamesz)<-colnames(cellnamesy)
#Combination
pdmat<-as.matrix(rbind(cellnamesy,cellnamesz))
pddf<-as.data.frame(pdmat)
pddf$pt<-as.numeric(as.character(pddf$pt))
pd<-new("AnnotatedDataFrame",data=pddf)

#Numeric Matrices

#Yang

row.names(loadmaty)<-loadmaty[,1]
maty<-loadmaty[2:length(loadmaty[,1]),match(rownames(cellnamesy),loadmaty[1,])]
colnames(maty)<-rownames(cellnamesy)
maty<-as.matrix(maty)
matchind<-match(rownames(fdmat),rownames(maty))
maty<-maty[matchind,]
nummaty<-mat2numericmat(maty)
#idandname<-function(idnamemat, geneid, genename)
row.names(nummaty)<-idandname(fdmat, rownames(nummaty),"NA")
```

```
## Warning in if (geneid == "NA") {: the condition has length > 1 and only the
## first element will be used
```

```
#Zonation
tidymatz<-zonatedmat2[6:length(zonatedmat2[,1]),]
matchind<-match(rownames(tidymatz),fdmat[,1])
matchind<-matchind[-which(is.na(matchind))]
tidymatz<-tidymatz[matchind,]
row.names(tidymatz)<-rownames(fdmat)[matchind]
matz<-as.matrix(tidymatz[,-1])
nummatz<-mat2numericmat(matz)
row.names(nummatz)<-idandname(fdmat, rownames(nummatz),"NA")
```

```
## Warning in if (geneid == "NA") {: the condition has length > 1 and only the
## first element will be used
```

Seurat Implementation

```
#Yang cell dataset
cdsy <- CreateSeuratObject(raw.data = nummaty)
cdsy <- NormalizeData(object = cdsy)
cdsy <- ScaleData(object = cdsy)
cdsy <- FindVariableGenes(object = cdsy, do.plot = FALSE)
#Zonation cell dataset
cdsz <- CreateSeuratObject(raw.data = nummatz)
cdsz <- NormalizeData(object = cdsz)
cdsz <- ScaleData(object = cdsz)
cdsz <- FindVariableGenes(object = cdsz, do.plot = FALSE)

# we will take the union of the top 2k variable genes in each dataset for
# alignment note that we use 1k genes in the manuscript examples, you can
# try this here with negligible changes to the overall results
hvg.cdsy <- rownames(x = head(x = cdsy@hvg.info, n = 4437))
hvg.cdsz <- rownames(x = head(x = cdsz@hvg.info, n = 4437))
hvg.union <- union(x = hvg.cdsy, y = hvg.cdsz)

cdsz@meta.data[, "protocol"] <- "Zonation"
cdsy@meta.data[, "protocol"] <- "Yang"
```

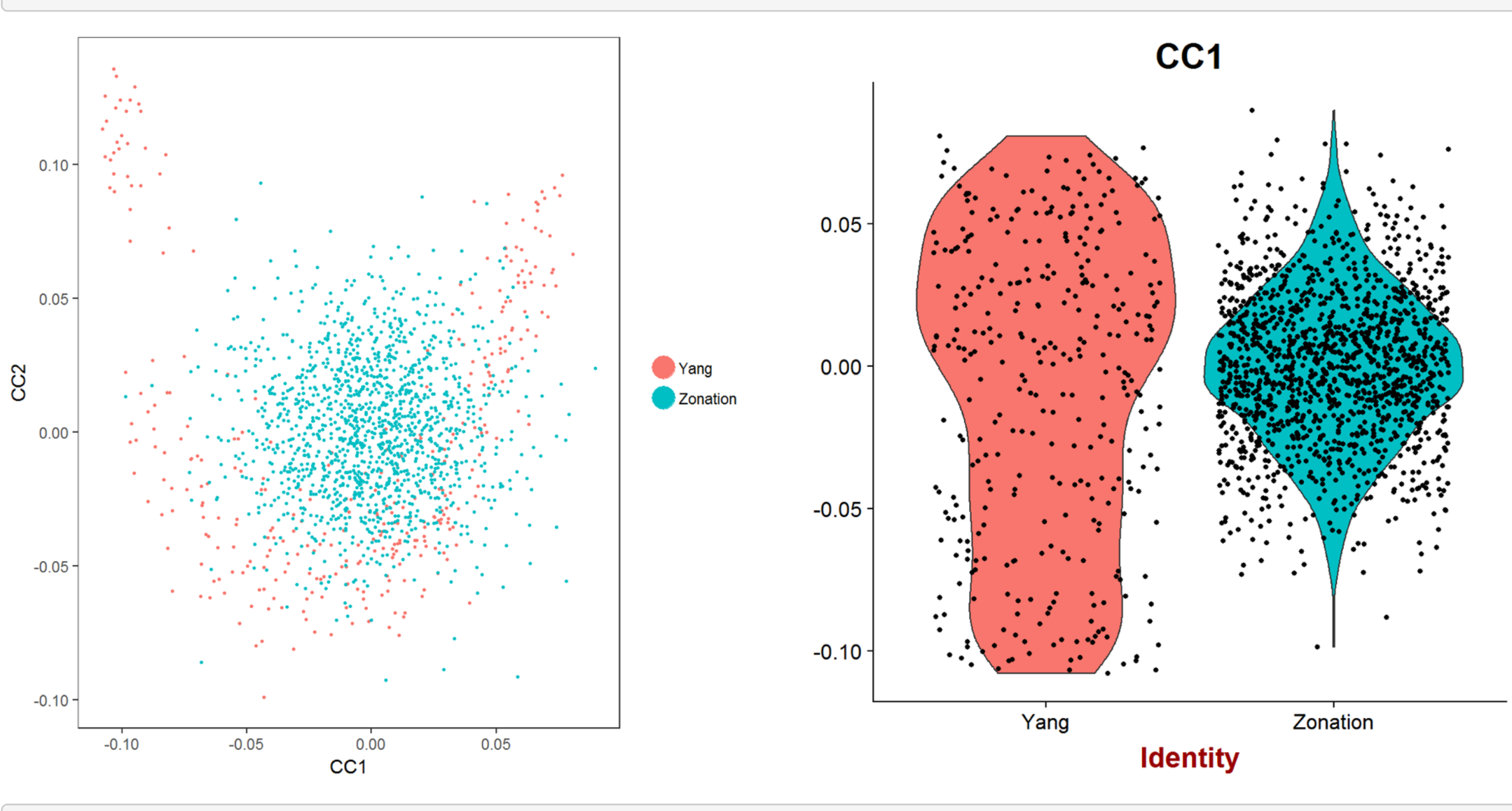
Running CCA

```
cdscomb<- RunCCA(object = cdsy, object2 = cdsz, genes.use = hvg.union)
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
## Scaling data matrix
```

```
p1 <- DimPlot(object = cdscomb, reduction.use = "cca", group.by = "protocol", pt.size = 0.5,
    do.return = TRUE)
p2 <- VlnPlot(object = cdscomb, features.plot = "CC1", group.by = "protocol", do.return = TRUE)
plot_grid(p1, p2)
```

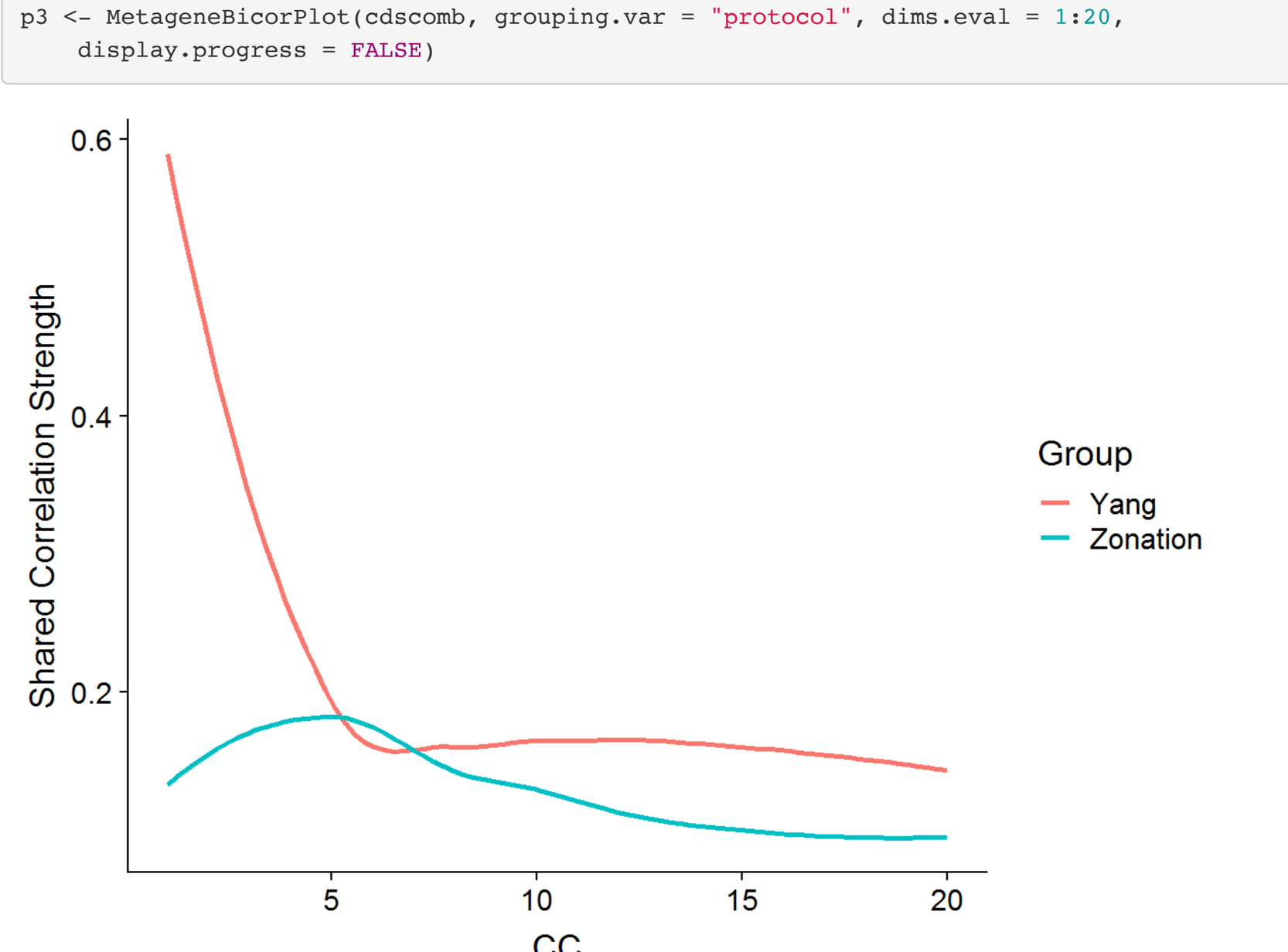



```
PrintDim(object =cdscomb, reduction.type = "cca", dims.print = 1:2, genes.print = 10)
```

```
## [1] "CC1"
##  [1] "Apoh"       "Ahsg"       "Fga"        "Itih3"      "Serpina1d"
##  [6] "Trf"        "Mat1a"      "Agt"        "Kng1"       "Pzp"
## [1] ""
##  [1] "Cnbp"      "Rpl14"    "Ncl"       "Hsp90aa1" "Serbp1"    "Naa50"
##  [7] "Cbx3"      "Fkbp3"    "Nudc"      "Snrpg"
## [1] ""
## [1] ""
## [1] "CC2"
##  [1] "Serpina6" "Rtp3"     "Fdps"      "Gstt2"     "Tm7sf2"    "Gcgr"
##  [7] "Apoc2"     "Ugt2b34"  "Cpt2"      "Aldh3a2"
## [1] ""
##  [1] "Fbp1"       "Rdh7"       "Serpina1e" "Aqp8"       "Hsd11b1"
##  [6] "Cyp2c40"    "Sult1a1"    "Serpina1c" "Qsox1"      "Hmgn1"
## [1] ""
## [1] ""
```

Choosing which CCs to use

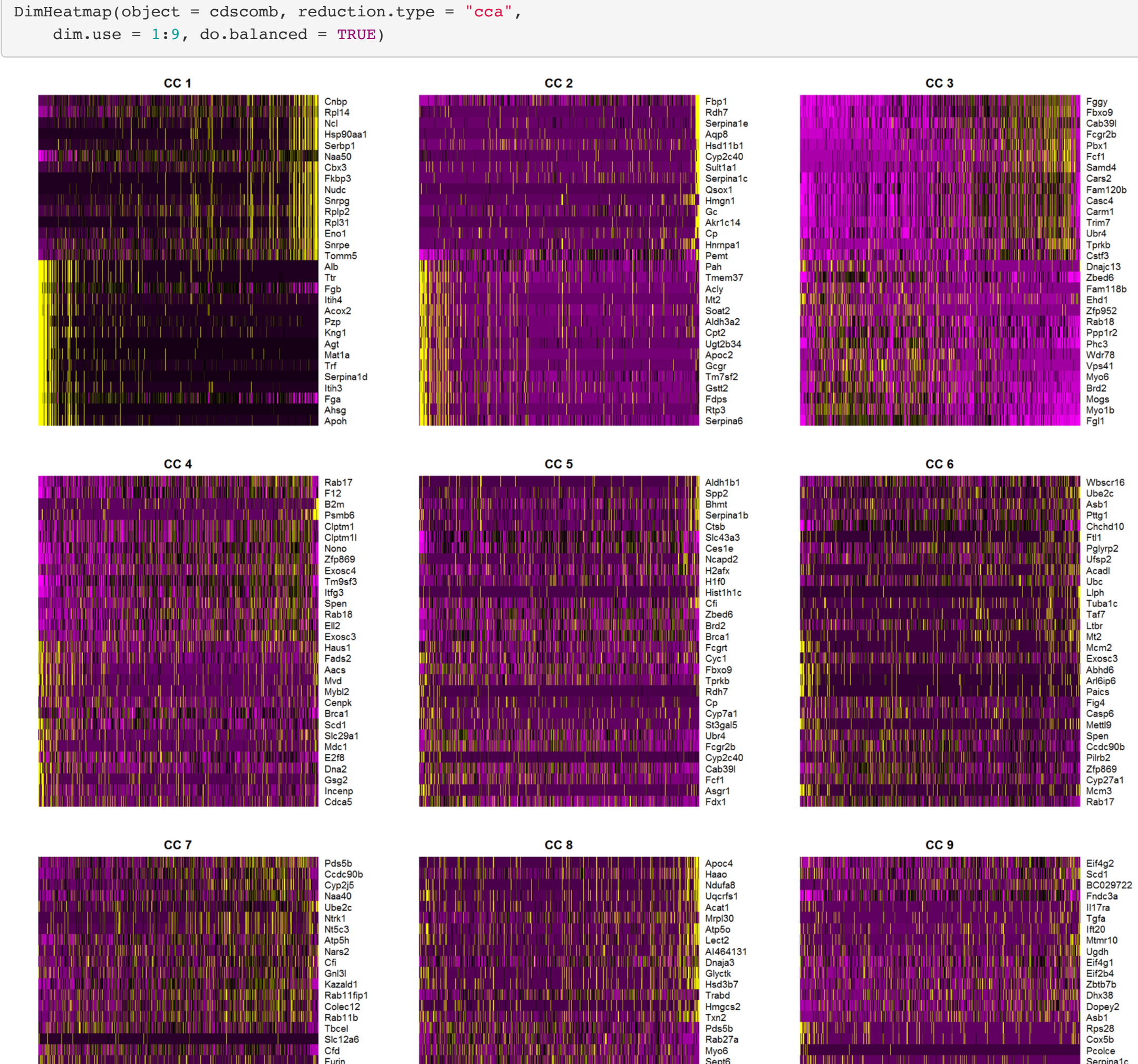
```
p3 <- MetageneBicorPlot(cdscomb, grouping.var = "protocol", dims.eval = 1:20,
    display.progress = FALSE)
```



20 CCs are appropriate to use

Lets look at heatmaps for first 9 CCs

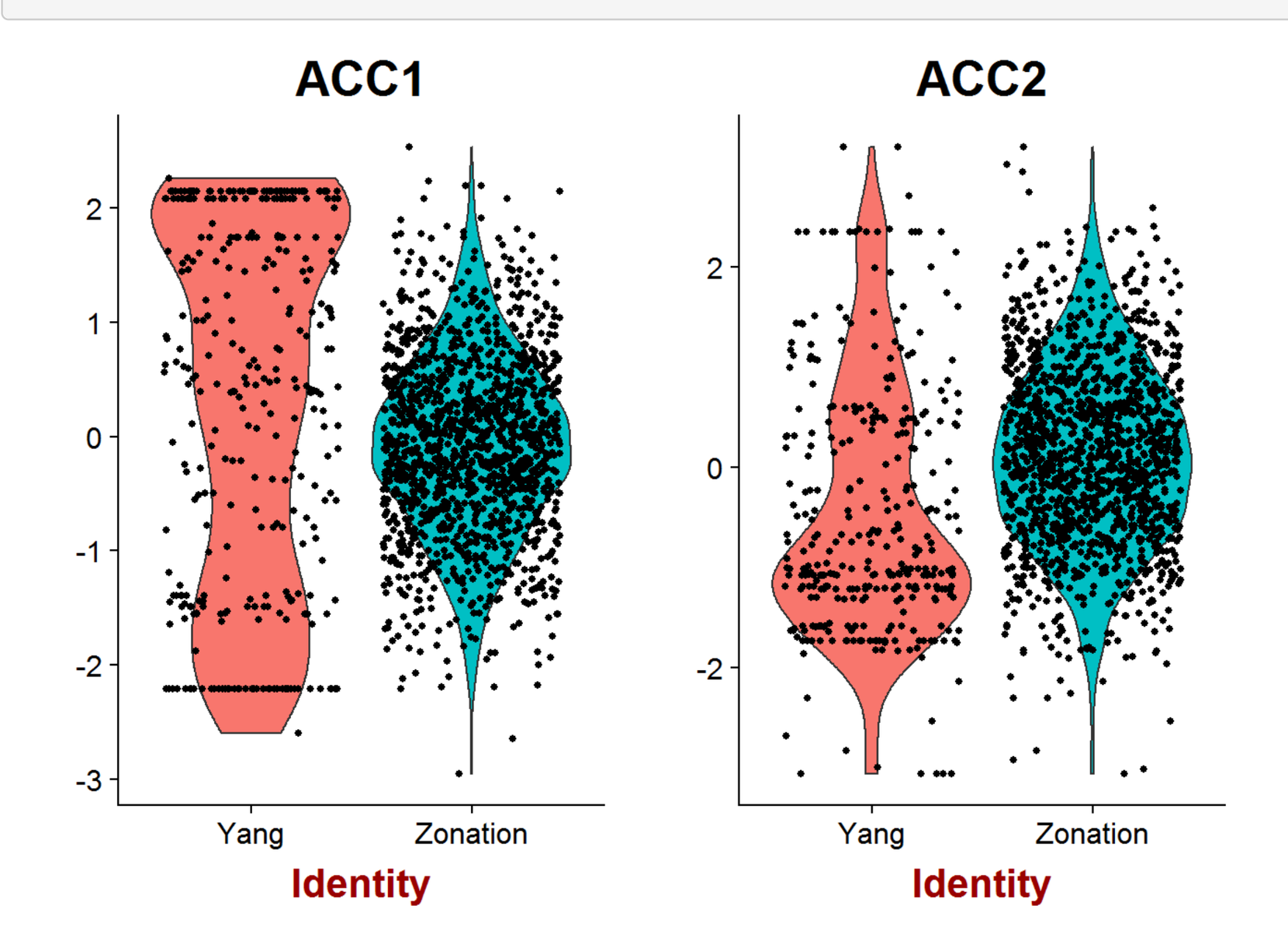
```
DimHeatmap(object = cdscomb, reduction.type = "cca",
    dim.use = 1:9, do.balanced = TRUE)
```




```
cdscomb <- AlignSubspace(cdscomb, reduction.type = "cca", grouping.var = "protocol",
    dims.align = 1:20)
```

Violin plots to verify alignment:

```
p1 <- VlnPlot(object = cdscomb, features.plot = "ACC1", group.by = "protocol",
    do.return = TRUE)
p2 <- VlnPlot(object = cdscomb, features.plot = "ACC2", group.by = "protocol",
    do.return = TRUE)
plot_grid(p1, p2)
```



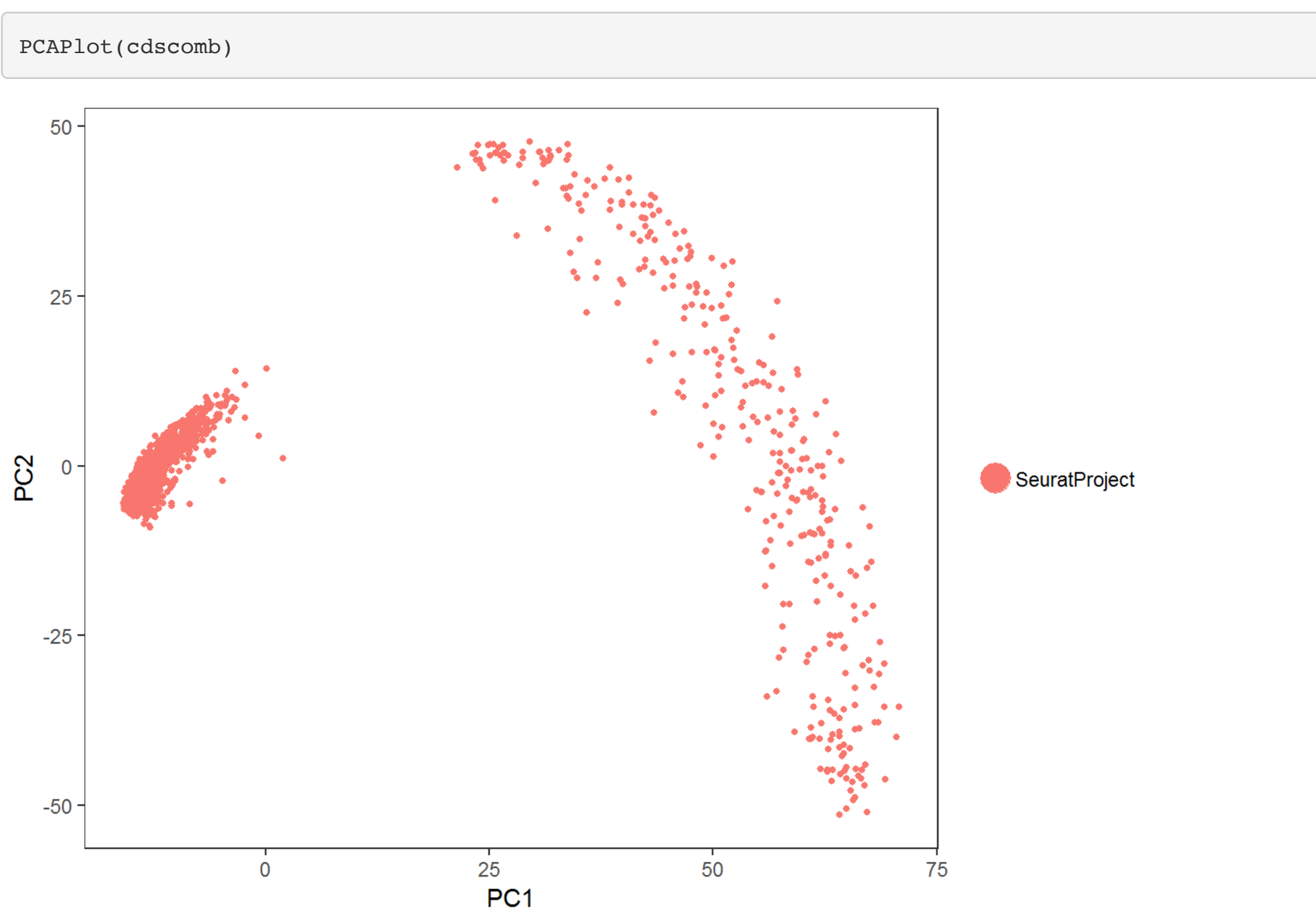
The Yang data still seems a little skewed, but appears to be better aligned than the 1st.

Now a tSNE is ran on the combined dataset

```
cdscomb<-RunPCA(cdscomb)
```

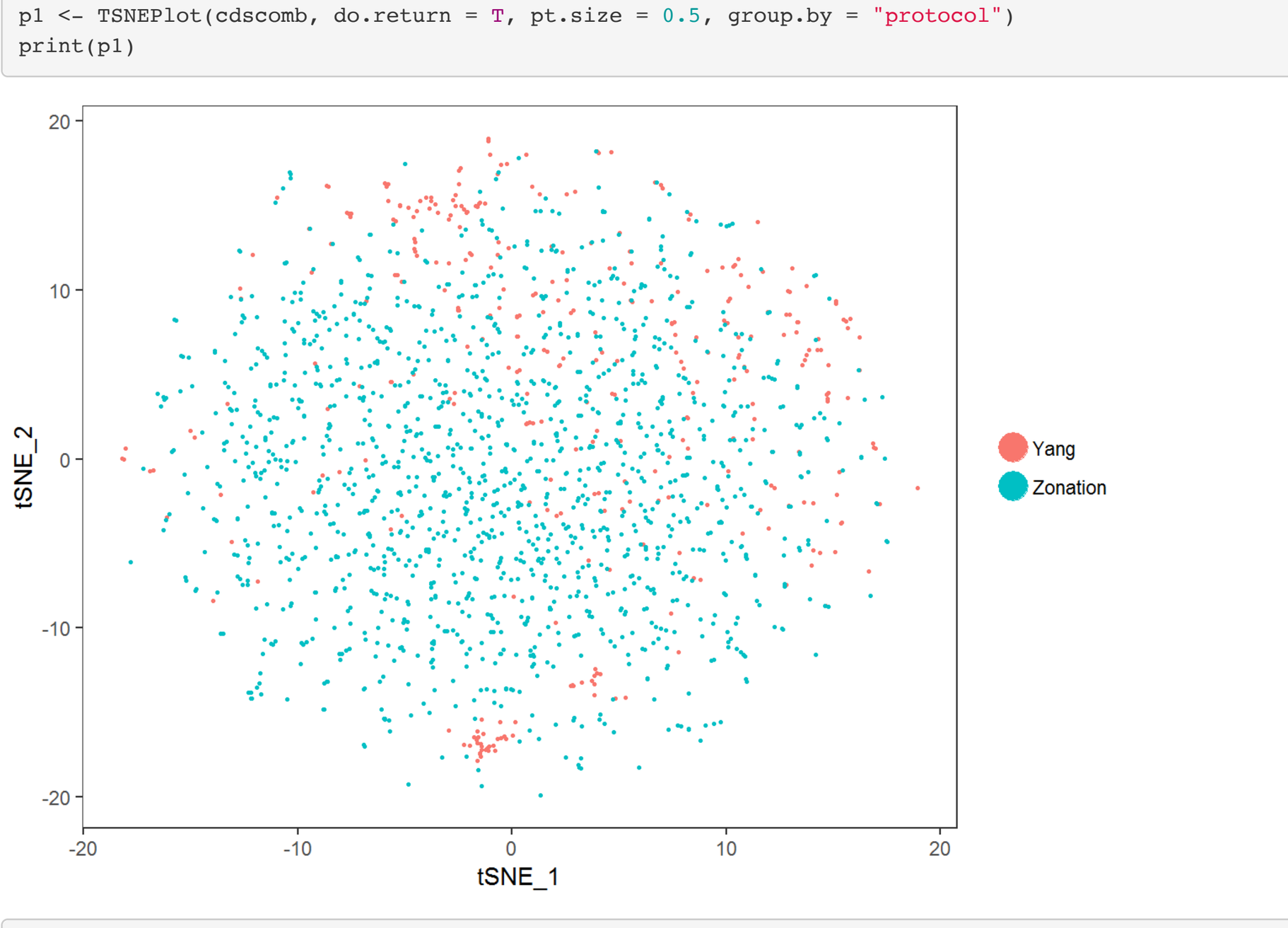
```
## [1] "PC1"
##  [1] "Ccl6"      "Ccdc55"    "Plekho2"   "C6"        "Wdr91"     "Ccl17"
##  [7] "Hps6"      "Wdtc1"     "Calcoco1"  "Wdr83"     "Irf7"      "Mup21"
## [13] "Ccl24"     "Wdr92"     "Wdr82"     "Ccdc91"    "Mtus1"     "Ccl25"
## [19] "P4ha2"     "Dusp14"    "Mup3"      "Wee1"      "Mvp"       "Ccdc58"
## [25] "Ccdc9"     "Klf3"      "Camta2"    "Wipi1"     "Wdr93"     "Gm12166"
## [1] ""
##  [1] "Fabp5"     "Sdhd"      "Ldha"      "Glud1"     "Serpind1"  "Serpinf2"
##  [7] "Hspe1"     "Itih2"     "Uqcr11"    "Serpinf1"  "Phb2"      "Aass"
## [13] "Pdia6"     "Pla2g12b"  "Fh1"       "Ppa1"      "Lpl"       "Agmat"
## [19] "Atp5b"     "Pcbd1"     "Rogdi"     "Mmd2"      "Apom"      "Reln"
## [25] "Surf4"     "Atp5f1"    "Calr"      "Sec61b"    "Got2"      "Sec61a1"
## [1] ""
## [1] ""
## [1] "PC2"
##  [1] "Snrpe"     "Sumo2"     "Naa50"     "Snrpd1"    "Cnbp"      "Hmgn1"
##  [7] "Nap1l1"    "Nudc"      "G3bp1"     "Snrnp40"   "Srsf7"     "Hsp90ab1"
## [13] "Hsph1"     "Tubb5"     "Hsp90aa1"  "Hnrnpa1"   "Snrpd2"    "Serbp1"
## [19] "Cct2"      "Rpl14"     "Nasp"      "Cbx3"      "Psma7"     "Nhp2"
## [25] "Fkbp3"     "Emb"       "Hspa8"     "Rps9"      "Ilf2"      "Ncl"
## [1] ""
##  [1] "Apoh"      "Ahsg"      "Kng1"      "Itih3"     "Alb"
##  [6] "Ttr"       "Acox2"     "Fga"       "Agt"       "Mat1a"
## [11] "Serpina1d" "Trf"       "Aldob"     "Angptl3"   "F2"
## [16] "Itih4"     "Apoa1"     "Gjb1"      "Pzp"       "Cps1"
## [21] "Gsta3"     "Hpd"       "Hgfac"     "Nrn1"      "Adh1"
## [26] "Acat1"     "Hpx"       "Cyp2d26"   "Fgb"       "C3"
## [1] ""
## [1] ""
## [1] "PC3"
##  [1] "Fgl1"     "Nap1l1"   "Hmgn1"    "Snrnp40"  "Ilf2"     "Tspan6"   "Wdr78"
##  [8] "Myo1b"    "Pfdn2"    "Hsph1"    "Hnrnpa1"  "Tubb5"    "Rdh7"     "Slc7a6"
## [15] "Phc3"     "Mogs"     "Zfp952"   "Nudc"     "Srsf7"    "Llph"     "Car4"
## [22] "Eif4a1"   "Nucks1"   "Mat2a"    "Naca"     "Unc119b"  "Vps41"    "Nkd1"
## [29] "Rangap1"  "Cpxm1"
## [1] ""
##  [1] "Fbxo9"     "Fcgrt"     "Mgrn1"     "Fcgr2b"    "Fcf1"      "C8g"
##  [7] "Cars2"     "Ubr4"      "Trim7"     "Fggy"      "Tecr"      "Casc4"
## [13] "Serpina6"  "Carm1"     "Fech"      "Pbx1"      "Cab39l"    "Fdx1"
## [19] "Mt1"       "Cd59a"     "Zfp385b"   "Samd4"     "Atp5g1"    "Fam120b"
## [25] "Sepp1"     "Mlxipl"    "Naaladl1"  "Pex7"      "P2rx4"     "Atp5h"
## [1] ""
## [1] ""
## [1] "PC4"
##  [1] "Rdh7"      "Serpina1e" "Sult1a1"   "Aqp8"      "Pzp"
##  [6] "Fbp1"      "Adh1"      "Akr1c14"   "Qsox1"     "Trf"
## [11] "Serpina1d" "Cfhr2"     "Hsd17b13"  "Mat1a"     "C3"
## [16] "Ahsg"      "Fdx1"      "Hsd11b1"   "Selenbp2"  "Cyp2c40"
## [21] "Fcf1"      "Agt"       "Serpina1c" "Apoh"      "Fmo1"
## [26] "Tff3"      "Fggy"      "Fbxo9"     "Fcgr2b"    "Cps1"
## [1] ""
##  [1] "Nsdhl"     "Fgl1"      "Myo1b"     "Ndufv2"    "Mogs"      "Ehd1"
##  [7] "Serpina6"  "Myo6"      "Scp2"      "Snx15"     "Phc3"      "Wdr78"
## [13] "Acly"      "Brd2"      "Pah"       "Mrpl2"     "Mvd"       "Slc25a4"
## [19] "Gstm3"     "Soat2"     "Vps41"     "Rtp3"      "Il4ra"     "Fam114a1"
## [25] "Zbed6"     "Tm7sf2"    "Cxcl12"    "Cpt2"      "Slc25a1"   "Fads2"
## [1] ""
## [1] ""
## [1] "PC5"
##  [1] "Cab39l"        "Fggy"          "Samd4"         "Pbx1"
##  [5] "Fbxo9"         "Fcf1"          "Fam120b"       "Fcgr2b"
##  [9] "Cars2"         "Casc4"         "Cstf3"         "Carm1"
## [13] "Tprkb"         "Cab39"         "Elmo3"         "Mgrn1"
## [17] "Naaladl1"      "Zfp524"        "Tpst1"         "Zfp507"
## [21] "1110008F13Rik" "Cgref1"        "Id1"           "Rab43"
## [25] "Smg6"          "Pcolce2"       "Itih5"         "Mrps28"
## [29] "Cdk15"         "Ucp2"
## [1] ""
##  [1] "Myo1b"     "Rab18"     "Ccdc90b"   "Mogs"      "Krr1"
##  [6] "Fgl1"      "Brd2"      "F12"       "Myo6"      "Naa40"
## [11] "Pipox"     "Rab17"     "Hadh"      "Zmym3"     "Ccnd2"
## [16] "Tm9sf3"    "Vezf1"     "Exosc3"    "Leprot"    "Zbed6"
## [21] "Kazald1"   "Fam107a"   "Hist1h2bj" "Lmo4"      "Hadhb"
## [26] "Fam103a1"  "Myl4"      "Nono"      "Spop"      "Vps41"
## [1] ""
## [1] ""
```

```
PCAPlot(cdscomb)
```

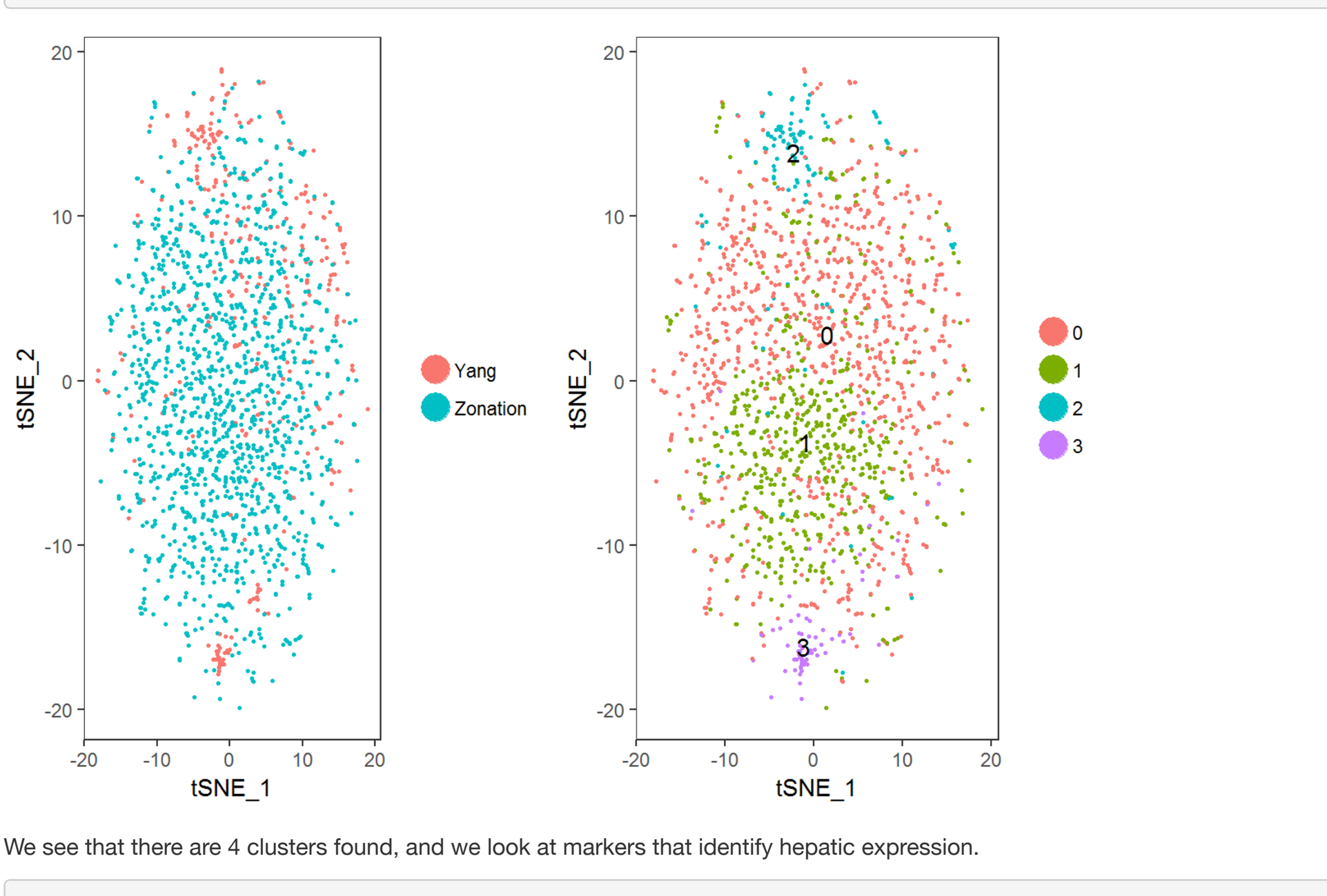



```
cdscomb <- RunTSNE(cdscomb, reduction.use = "cca.aligned", dims.use = 1:20,
    do.fast = T)
```

```
p1 <- TSNEPlot(cdscomb, do.return = T, pt.size = 0.5, group.by = "protocol")
print(p1)
```



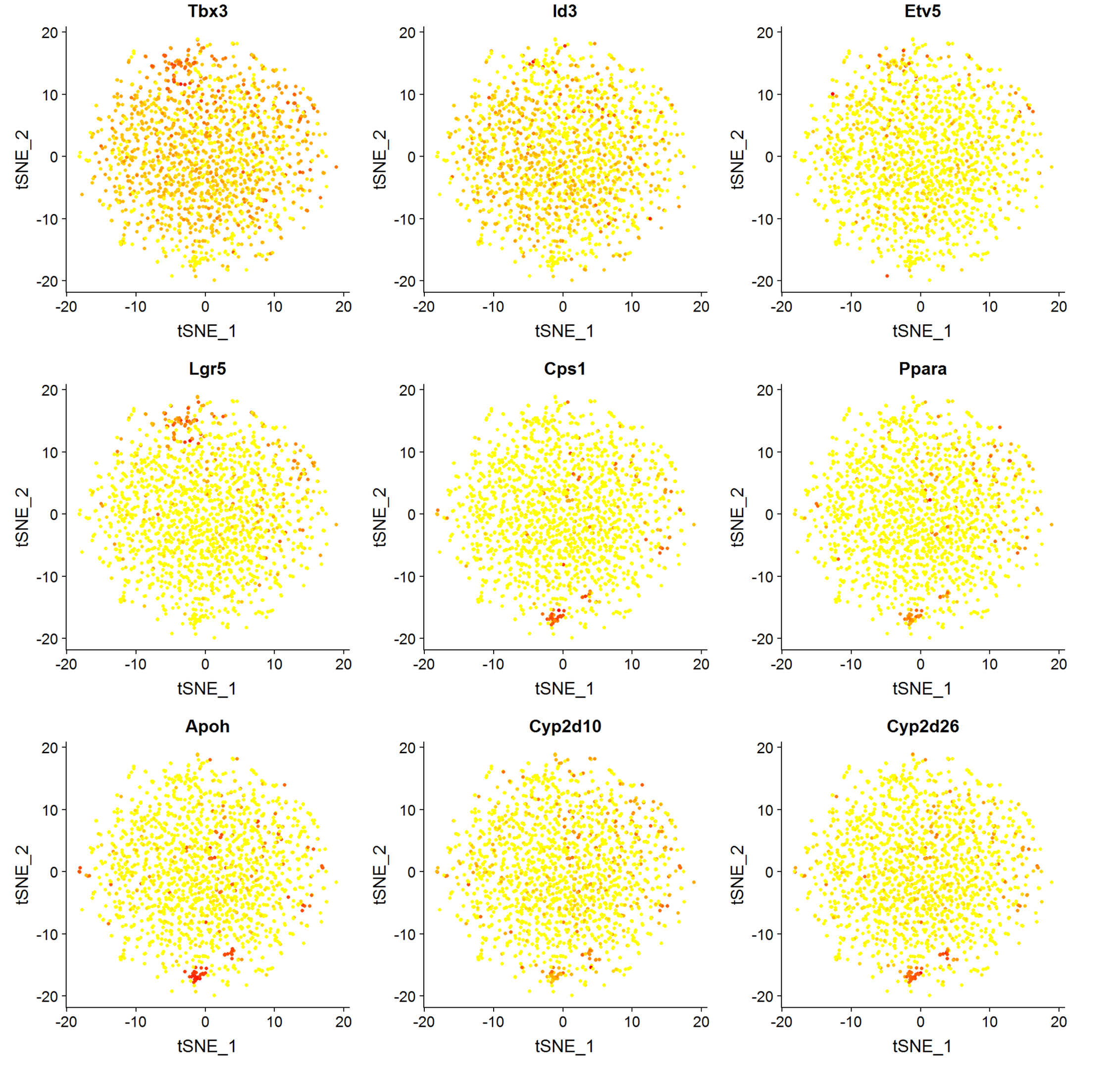

```
cdscomb <- FindClusters(cdscomb, reduction.type = "cca.aligned",
    resolution = .7, dims.use = 1:20)
p2 <- TSNEPlot(cdscomb, do.label = T, do.return = T, pt.size = 0.5)
plot_grid(p1, p2)
```



We see that there are 4 clusters found, and we look at markers that identify hepatic expression.

```
clusterb<-c("Tbx3", "Id3",  "Etv5", "Lgr5")#hepatoblast genes
clusterc<-c("Cps1", "Ppara", "Apoh", "Cyp2d10", "Cyp2d26")#cluster c genes

FeaturePlot(object = cdscomb, features.plot=c(clusterb,clusterc) )
```



We see that cluster 3 can be identified as hepatocytes

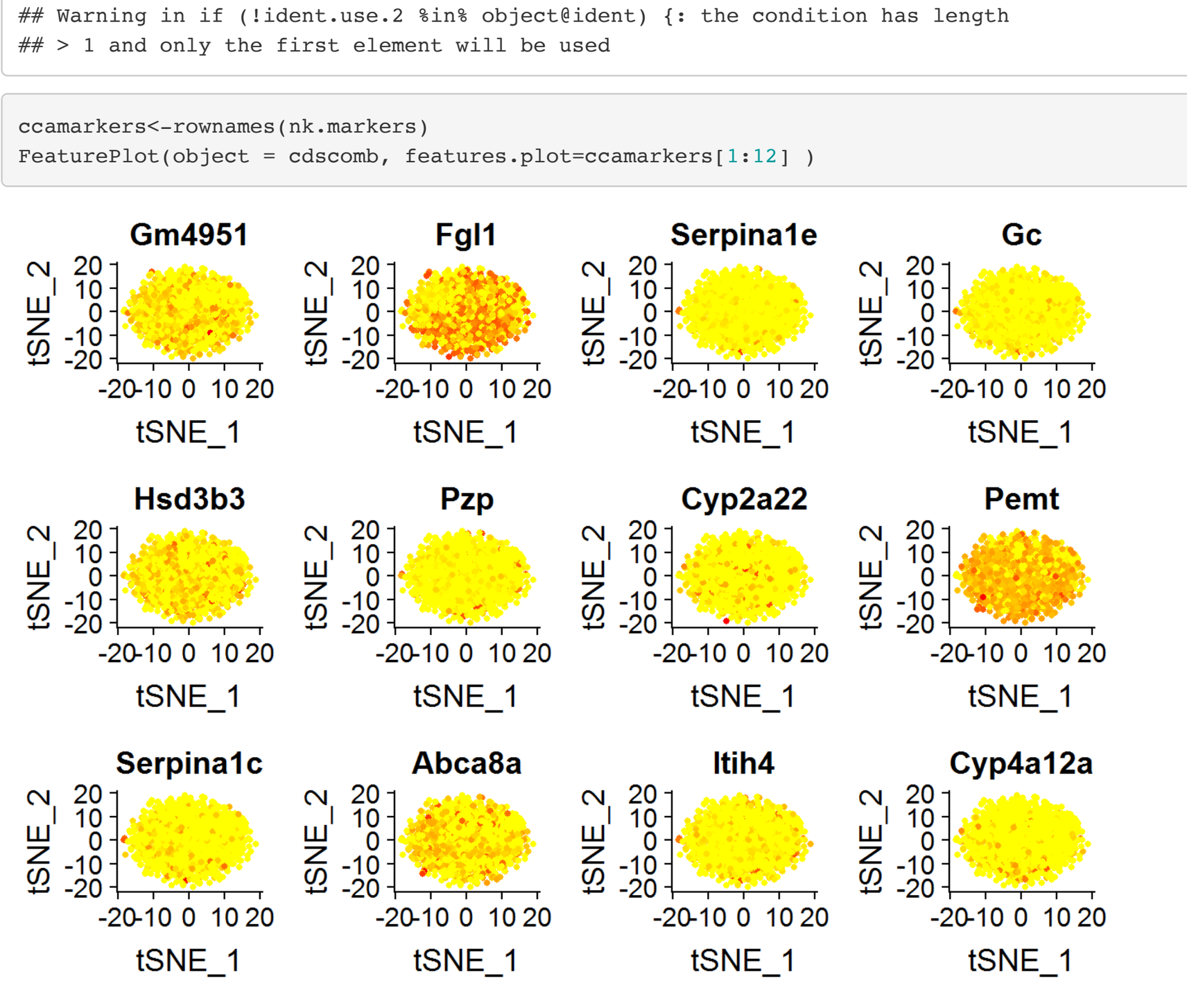
Conserved markers are identified and plotted in a similar fashion, we use cluster 3 as an example.

```
nk.markers <- FindConservedMarkers(cdscomb, ident.1=3, grouping.var = "protocol",
    print.bar = FALSE)
```

```
## Warning in if (!ident.use.2 %in% object@ident) {: the condition has length
## > 1 and only the first element will be used

## Warning in if (!ident.use.2 %in% object@ident) {: the condition has length
## > 1 and only the first element will be used
```

```
ccamarkers<-rownames(nk.markers)
FeaturePlot(object = cdscomb, features.plot=ccamarkers[1:12] )
```




```
write.table(ccamarkers, "ccamarkers.txt")
tsnevals<-as.matrix(cdscomb@dr$tsne@cell.embeddings)
write.xlsx(tsnevals, "tSNEvals2.xlsx")
ccavals<-as.matrix(cdscomb@dr$cca@cell.embeddings)
ccavals<-ccavals[,1:2]
write.xlsx(ccavals,"ccavals.xlsx")
ccamarkers<-as.matrix(cdscomb@dr$cca@gene.loadings.full)
ccamarkers<-ccamarkers[,1:2]
write.xlsx(ccamarkers, "ccamarkers.xlsx")
accamarkers<-as.matrix(cdscomb@dr$cca.aligned@cell.embeddings)
accamarkers<-accamarkers[,1:2]
write.xlsx(accamarkers, "accamarkers.xlsx")
```