**CCA** on Halper & Yang Nick Wawee August 27, 2018 Loading #Loading loadmaty<-read.table("Yang\_ReadCount\_ERCC\_Converted.txt", fill = TRUE, stringsAsFactors = FALSE)</pre> loadmaty2<-read.table("pseudotimemat.txt", stringsAsFactors = FALSE)</pre> zonatedmat<-read.table("OrderedPtMatrix Filtered 1.txt", stringsAsFactors = FALSE)</pre> zonatedmat<-as.data.frame(zonatedmat)</pre> #Feature Data zgenes<-rownames(zonatedmat)</pre> zonatedmat<-cbind(zgenes,zonatedmat)</pre> zonatedmat\$zgenes<-as.character(zonatedmat\$zgenes)</pre> zonatedmat2<-zonatedmat %>% separate rows(zgenes, sep=";") row.names(zonatedmat2)<-zonatedmat2[,1]</pre> # 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])</pre> matchind<-match(rownames(zonatedmat2),loadmaty[,2])</pre> matchind<-matchind[-which(is.na(matchind))]</pre> fdmat<-as.matrix(loadmaty[matchind,1:2])</pre> row.names(fdmat)<-fdmat[,1]</pre> fdmat<-as.matrix(fdmat[,-1])</pre> colnames(fdmat)<-"gene\_short\_name"</pre> fd<-as.data.frame(fdmat)</pre> fd<-new("AnnotatedDataFrame", data=fd)</pre> #Pheno Data cellnamesy<-as.matrix(colnames(loadmaty2)[which(loadmaty2[2,]=="Hepatoblast"|loadmaty2[2,]=="Hepatocyte")]) colnames(cellnamesy)<-cellnamesy[1,]</pre> cellnamesy<-cbind(cellnamesy,rep("Embryonic",length(cellnamesy[,1])))</pre> colnames(cellnamesy)<-c("Cell Name", "Cell Type")</pre> row.names(cellnamesy)<-cellnamesy[,1]</pre> pt<-loadmaty2[1,match(rownames(cellnamesy),colnames(loadmaty2))]</pre> pt<-as.numeric(pt)</pre> pt<-scalefun(pt, "0and1")</pre> statey<-loadmaty2[2,match(rownames(cellnamesy),colnames(loadmaty2))]</pre> statey<-t(statey)</pre> colnames(statey)<-"State0"</pre> statey2<-statey colnames(statey2)<-"State2"</pre> cellnamesy<-cbind(cellnamesy,pt,statey,statey2)</pre> #Zonation cellnamesz<-as.matrix(colnames(zonatedmat2)[2:length(zonatedmat2)])</pre> row.names(cellnamesz)<-cellnamesz</pre> cellnamesz<-cbind(cellnamesz, rep("Zonation",length(cellnamesz)))</pre> colnames(cellnamesz)<-c("Cell\_Name", "Cell\_Type")</pre> #cellnamesz<-cellnamesz[-length(cellnamesz[,1]),]</pre> cellnamesz<-cbind(cellnamesz,t(zonatedmat2[c(1,4,5),-1])) cellnamesz[,3]<-scalefun(as.numeric((cellnamesz[,3])), "Oand1")</pre> colnames(cellnamesz)<-colnames(cellnamesy)</pre> #Combination pdmat<-as.matrix(rbind(cellnamesy,cellnamesz))</pre> pddf<-as.data.frame(pdmat)</pre> pddf\$pt<-as.numeric(as.character(pddf\$pt))</pre> pd<-new("AnnotatedDataFrame",data=pddf)</pre> #Numeric Matrices #Yang row.names(loadmaty)<-loadmaty[,1]</pre> maty<-loadmaty[2:length(loadmaty[,1]), match(rownames(cellnamesy), loadmaty[1,])]</pre> colnames(maty)<-rownames(cellnamesy)</pre> maty<-as.matrix(maty)</pre> matchind<-match(rownames(fdmat),rownames(maty))</pre> maty<-maty[matchind,]</pre> nummaty<-mat2numericmat(maty)</pre> #idandname<-function(idnamemat, geneid, genename)</pre> row.names(nummaty)<-idandname(fdmat, rownames(nummaty), "NA")</pre> ## 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]),]</pre> matchind<-match(rownames(tidymatz),fdmat[,1])</pre> matchind<-matchind[-which(is.na(matchind))]</pre> tidymatz<-tidymatz[matchind,]</pre> row.names(tidymatz)<-rownames(fdmat)[matchind]</pre> matz<-as.matrix(tidymatz[,-1])</pre> nummatz<-mat2numericmat(matz)</pre> row.names(nummatz)<-idandname(fdmat, rownames(nummatz), "NA")</pre> ## 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)</pre> cdsy <- NormalizeData(object = cdsy)</pre> cdsy <- ScaleData(object = cdsy)</pre> cdsy <- FindVariableGenes(object = cdsy, do.plot = FALSE)</pre> #Zonation cell dataset cdsz <- CreateSeuratObject(raw.data = nummatz)</pre> cdsz <- NormalizeData(object = cdsz)</pre> cdsz <- ScaleData(object = cdsz)</pre> cdsz <- FindVariableGenes(object = cdsz, do.plot = FALSE)</pre> # 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))  $hvg.cdsz \leftarrow rownames(x = head(x = cdsz@hvg.info, n = 4437))$ hvg.union <- union(x = hvg.cdsy, y = hvg.cdsz) cdsz@meta.data[, "protocol"] <- "Zonation"</pre> cdsy@meta.data[, "protocol"] <- "Yang"</pre> Running CCA cdscomb<- RunCCA(object = cdsy, object2 = cdsz, genes.use = hvg.union)</pre> ## 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) CC1 0.05 0.05 0.00 Yang Zonation 0.00 -0.05 -0.05 -0.10 -0.10 · Yang Zonation -0.05 0.00 0.05 -0.10 Identity CC1 PrintDim(object =cdscomb, reduction.type = "cca", dims.print = 1:2, genes.print = 10) ## [1] "CC1" "Itih3" "Ahsg" "Fga" [1] "Apoh" "Serpinald" [6] "Trf" "Mat1a" "Kng1" "Agt" "Pzp" ## [1] "" "Rpl14" "Ncl" "Hsp90aa1" "Serbp1" [1] "Cnbp" "Naa50" "Fkbp3" "Nudc" "Snrpg" [7] "Cbx3" ## [1] [1] "CC2" "Rtp3" "Gstt2" "Tm7sf2" [1] "Serpina6" "Fdps" "Gcgr" "Cpt2" "Aldh3a2" "Ugt2b34" [7] "Apoc2" ## [1] "" [1] "Fbp1" "Rdh7" "Serpinale" "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) 0.6 Shared Correlation Strength Group Yang Zonation 0.2 15 10 20 5 CC 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) CC 1 CC 2 CC 3 Cnbp Rpl14 Ncl Serpina1e Fcgr2b Pbx1 Fcf1 Samd4 Hsp90aa1 Fkbp3 Nudc Cars2 Fam120b Snrpg Rplp2 Rpl31 Eno1 Casc4 Akr1c14 Tmem37 Fam118b Ehd1 Zfp952 Rab18 Acly Mt2 Soat2 Aldh3a2 Cpt2 Itih4
Acox2
Pzp
Kng1
Agt
Mat1a
Trf
Serpina1d
Itih3
Fga
Ahsg
Apoh Ugt2b34 Apoc2 Vyps41 Myo6 Brd2 Mogs Myo1b Fgl1 CC 4 CC 5 CC 6 Aldh1b1 Wbscr16 Spp2 Serpina1b Ctsb Chchd10 Slc43a3 Nono Zfp869 Ces1e Pglyrp2 Ncapd2 Ufsp2 Acadl Ubc H2afx Exosc4 Tm9sf3 H1f0 Itfg3 Spen Hist1h1c Llph Tuba1c Rab18 Zbed6 EII2 Brd2 Ltbr Brca1 Exosc3 Fcgrt Mcm2 Exosc3 Abhd6 Mybl2 Rdh7 Paics Cp Cyp7a1 St3gal5 Ubr4 Fig4 Casp6 Mettl9 Brca1 Scd1 Slc29a1 Spen Mdc1 Fcgr2b Ccdc90b E2f8 Cyp2c40 Pilrb2 Dna2 Cab39I Zfp869 Gsg2 Incenp Fcf1 Cyp27a1 Asgr1 CC 7 CC8 CC 9 Apoc4 Cyp2j5 BC029722 Tgfa Ift20 Mrpl30 Nt5c3 Atp5h Lect2 Mtmr10 Ugdh Eif4g1 Eif2b4 Zbtb7b Nars2 AI464131 Dnaja3 Gnl3l Glyctk Hsd3b7 Kazald1 Dhx38 Rab11fip1 Trabd Colec12 Hmgcs2 Txn2 Dopey2 Rab11b Asb1 Pds5b Rps28 Slc12a6 Pcolce Chd1 Rab21 Ndufa7 Wdr7 Usp2 Mknk2 Mxra8 Fam35a Ftcd Rab3gap1 Chchd7 Serpina1b Chchd5 Cox6a1 Rabep1 Ufm1 Rab43 Apoe Fam107a . Ndufa4 Pawr ltm2b Rhbdd1 Slc25a34 B020018G12Rik Paics cdscomb <- AlignSubspace(cdscomb, reduction.type = "cca", grouping.var = "protocol",</pre> 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) ACC2 ACC1 2 2 0 0 -1 -2 -2 -3 Zonation Zonation Yang Yang **Identity Identity** 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)</pre> ## [1] "PC1" "Ccl17" "Ccl6" "Ccdc55" "Plekho2" "C6" "Wdr91" "Wdtc1" "Irf7" "Hps6" "Calcoco1" "Wdr83" "Mup21" "Wdr92" "Wdr82" "Cc125" "Cc124" "Ccdc91" "Mtus1" "Mup3" "Wee1" "dvM" "Ccdc58" ## [19] "P4ha2" "Dusp14" ## [25] "Ccdc9" "Klf3" "Camta2" "Wipi1" "Wdr93" "Gm12166" "Ldha" "Serpinf2" "Sdhd" "Glud1" "Serpind1" "Fabp5" "Aass" "Hspe1" "Itih2" "Phb2" "Uqcr11" "Serpinf1" "Pdia6" "Fh1" "Lpl" "Agmat" "Pla2g12b" "Ppa1" "Reln" [19] "Atp5b" "Pcbd1" "Rogdi" "Mmd2" "Apom" ## [25] "Surf4" "Atp5f1" "Calr" "Got2" "Sec61a1" "Sec61b" ## [1] ## [1] "PC2" "Snrpe" "Sumo2" "Naa50" "Snrpd1" "Cnbp" "Hmgn1" "Nudc" "G3bp1" "Snrnp40" "Hsp90ab1" "Nap1l1" "Srsf7" ## [13] "Hsph1" "Serbp1" "Tubb5" "Hsp90aa1" "Hnrnpa1" "Snrpd2" ## [19] "Cct2" "Rpl14" "Cbx3" "Nhp2" "Nasp" "Psma7" ## [25] "Fkbp3" "Ilf2" "Ncl" "Rps9" "Hspa8" ## [1] "" "Alb" ## [1] "Apoh" "Ahsg" "Kng1" "Itih3" [6] "Ttr" "Acox2" "Fga" "Agt" "Matla" ## [11] "Serpinald" "Trf" "Aldob" "F2" "Angptl3" "Gjb1" ## [16] "Itih4" "Apoa1" "Pzp" "Cps1" ## [21] "Gsta3" "Hpd" "Hgfac" "Nrn1" "Adh1" "C3" "Cyp2d26" ## [26] "Acat1" "Hpx" "Fgb" ## [1] "" ## [1] "" ## [1] "PC3" [1] "Fgl1" "Nap1l1" "Hmgn1" "Snrnp40" "Ilf2" "Tspan6" "Wdr78" "Pfdn2" [8] "Myo1b" "Hsph1" "Hnrnpa1" "Tubb5" "Rdh7" "Slc7a6" "Car4" ## [15] "Phc3" "Mogs" "Zfp952" "Nudc" "Srsf7" "Llph" "Unc119b" "Vps41" ## [22] "Eif4a1" "Nkd1" "Nucks1" "Mat2a" "Naca" ## [29] "Rangap1" "Cpxm1" ## [1] "" "Mgrn1" ## [1] "Fbxo9" "Fcgrt" "Fcgr2b" "Fcf1" "C8g" "Trim7" ## [7] "Cars2" "Ubr4" "Fggy" "Tecr" "Casc4" "Cab391" "Fdx1" ## [13] "Serpina6" "Carm1" "Fech" "Pbx1" "Cd59a" "Zfp385b" "Samd4" "Fam120b" ## [19] "Mt1" "Atp5g1" "Naaladl1" "Pex7" ## [25] "Sepp1" "Mlxipl" "P2rx4" "Atp5h" ## [1] "" ## [1] "" ## [1] "PC4" "Pzp" [1] "Rdh7" "Serpinale" "Sultla1" "Aqp8" ## [6] "Fbp1" "Adh1" "Akr1c14" "Qsox1" "Trf" ## [11] "Serpinald" "Cfhr2" "Hsd17b13" "C3" "Mat1a" "Fdx1" "Hsd11b1" ## [16] "Ahsg" "Selenbp2" "Cyp2c40" "Fmo1" ## [21] "Fcf1" "Agt" "Serpinalc" "Apoh" "Fbxo9" "Fcgr2b" "Cps1" ## [26] "Tff3" "Fggy" ## [1] "" "Fgl1" "Ndufv2" "Ehd1" ## [1] "Nsdhl" "Myo1b" "Mogs" "Wdr78" [7] "Serpina6" "Myo6" "Scp2" "Snx15" "Phc3" "Brd2" "Mvd" "Slc25a4" ## [13] "Acly" "Pah" "Mrpl2" "Soat2" "Il4ra" "Fam114a1" ## [19] "Gstm3" "Vps41" "Rtp3" "Tm7sf2" ## [25] "Zbed6" "Cxcl12" "Cpt2" "Slc25a1" "Fads2" ## [1] "" ## [1] "" ## [1] "PC5" [1] "Cab391" "Samd4" "Pbx1" "Fggy" [5] "Fbxo9" "Fcf1" "Fam120b" "Fcgr2b" "Cstf3" [9] "Cars2" "Casc4" "Carm1" "Cab39" ## [13] "Tprkb" "Elmo3" "Mgrn1" "Zfp524" "Tpst1" ## [17] "Naaladl1" "Zfp507" ## [21] "1110008F13Rik" "Cgref1" "Id1" "Rab43" ## [25] "Smg6" "Pcolce2" "Itih5" "Mrps28" ## [29] "Cdk15" "Ucp2" ## [1] "" ## [1] "Myo1b" "Rab18" "Ccdc90b" "Mogs" "Krr1" "Brd2" "F12" [6] "Fgl1" "Myo6" "Naa40" ## [11] "Pipox" "Rab17" "Hadh" "Zmym3" "Ccnd2" "Leprot" "Zbed6" ## [16] "Tm9sf3" "Vezf1" "Exosc3" "Hist1h2bj" "Hadhb" ## [21] "Kazald1" "Fam107a" "Lmo4" "Nono" ## [26] "Fam103a1" "Myl4" "Vps41" "Spop" ## [1] "" ## [1] "" PCAPlot(cdscomb) 50 25 PC2 SeuratProject -25 -50 25 50 PC1 cdscomb <- RunTSNE(cdscomb, reduction.use = "cca.aligned", dims.use = 1:20,</pre> do.fast = T)p1 <- TSNEPlot(cdscomb, do.return = T, pt.size = 0.5, group.by = "protocol") print(p1) 20 10 tSNE\_2 Yang Zonation -10 -20 -10 0 10 20 tSNE\_1 cdscomb <- FindClusters(cdscomb, reduction.type = "cca.aligned",</pre> resolution = .7, dims.use = 1:20) p2 <- TSNEPlot(cdscomb, do.label = T, do.return = T, pt.size = 0.5) plot\_grid(p1, p2) 20 20 SNE 2 -10 -10 -20 -20 -10 -20 -10 10 20 10 20 -20 tSNE\_1 tSNE\_1 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) ) Tbx3 Etv5 ld3 20 20 20 10 10 10 tSNE\_2 tSNE -10 -10 -10 -20 -20 -20 20 -10 20 -10 10 -10 20 -20 10 -20 -20 10 tSNE\_1 tSNE\_1 tSNE\_1 Lgr5 Cps1 **Ppara** 20 -20 -20 10 10 10 tSNE\_2 -10 -10 -10 -20 -20 -20 20 -10 20 -10 10 20 -20 0 10 -20 -20 -10 10 tSNE\_1 tSNE\_1 tSNE\_1 Cyp2d10 Cyp2d26 **Apoh** 20 | 20 -20 -10 10 10 tSNE\_2 tSNE -10 -10 -10 -20 -20 -20 -20 -10 10 20 -20 -10 10 20 -20 -10 10 20 tSNE\_1 tSNE\_1 tSNE\_1 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",</pre> 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)</pre> FeaturePlot(object = cdscomb, features.plot=ccamarkers[1:12] ) Gm4951 FgI1 Serpina1e Gc 20 10 20 10 20 10  $\alpha$  $\alpha$ tSNE tSNE tSNE tSNE\_ -10 -20 -10 -20 -10 -20-10 0 10 20 -20-10 0 10 20 -20-10 0 10 20 -20-10 0 10 20 tSNE\_1 tSNE\_1 tSNE\_1 tSNE\_1 Hsd3b3 Cyp2a22 **Pemt** Pzp 20 10 2  $\alpha$ tSNE\_ tSNE tSNE tSNE -20-10 0 10 20 -20-10 0 10 20 -20-10 0 10 20 -20-10 0 10 20 tSNE\_1 tSNE\_1 tSNE\_1 tSNE\_1 Cyp4a12a Serpina1c Abca8a ltih4 20 10 0 20 10 0 20 10 0 20 10 0  $\alpha$ tSNE **tSNE** tSNE\_ tSNE -10 -20 -10 -10 -20-10 0 10 20 -20-10 0 10 20 -20-10 0 10 20 -20-10 0 10 20 tSNE\_1 tSNE\_1 tSNE\_1 tSNE\_1 write.table(ccamarkers, "ccamarkers.txt") tsnevals<-as.matrix(cdscomb@dr\$tsne@cell.embeddings)</pre> write.xlsx(tsnevals, "tSNEvals2.xlsx") ccavals<-as.matrix(cdscomb@dr\$cca@cell.embeddings)</pre> ccavals<-ccavals[,1:2]</pre> write.xlsx(ccavals, "ccavals.xlsx") ccamarkers<-as.matrix(cdscomb@dr\$cca@gene.loadings.full)</pre> ccamarkers<-ccamarkers[,1:2]</pre> write.xlsx(ccamarkers, "ccamarkers.xlsx") accamarkers<-as.matrix(cdscomb@dr\$cca.aligned@cell.embeddings)</pre> accamarkers<-accamarkers[,1:2]</pre>

write.xlsx(accamarkers, "accamarkers.xlsx")