GenerateMergedData&Figures.R

Alison G. Cole

2022-06-28

# set working directory and memory  
load ('GenesNVE.RData')   
memory.limit(2000000) #high memory requirements to run cytoTRACE

# load libraries:   
library(easypackages)  
libraries("Seurat", "Matrix", "readxl","RColorBrewer",'Rmagic',  
 'patchwork','dplyr','viridis','ggplot2','pals','SeuratWrappers')

set library palette

LibCP = c( "#053061", "#2166AC", "#4393C3","#025656", "#037272",  
 "#00A08A", "#66C6B8", "#F98400", "#F2AD00","#E2D200",   
 "#AD2323","#FF0000", "#9B51B4", "#530C6B","#DF6FA0", 'pink')

set other color palettes

gene.cp=c('lightgrey',rev(brewer.pal(11 , "Spectral" )))  
clust.cp.separate = unique (c(cols25(25),alphabet2(26),glasbey(32),alphabet(26)))  
clust.cp.graded = unique(c(stepped3(16),stepped(20),stepped2(20)))  
CLcp=clust.cp.graded[c(11,9,25,3,1,18,26,27,28,8,6,5)]  
clust.cp=CLcp  
  
run.save = F #if you want to re-save generated objects to start at different levels   
  
##Choose which sections to run  
setup = F #load the individual libraries  
PRE\_ANALYSIS=F #generate the AllData dataset  
generate.figure1=F  
Figure2.Cnidocyte=F  
Fig.3.lineages = F  
  
## load all libraries and generate merged object  
if (setup)   
{  
 individual.libraries = T  
 #loads the individual library objects  
 if (individual.libraries)  
 {  
 # load all individual datasets:   
 # 1) update gene annotations from models to human names  
 # 2) add library identifier to barcode  
 # 3) add library ID  
 # 4) plot the individual UMAP in library colour for Fig 1A  
   
 load (file = 'Z:/Personal\_Folders/Alison\_JuliaS/RObj\_SingleDatasets/gastrula18h.Robj')  
 gastrula18h@assays$RNA@counts@Dimnames[[1]] = genes$gene\_short\_name  
 gastrula18h@assays$RNA@data@Dimnames[[1]] = genes$gene\_short\_name  
 gastrula18h <- RenameCells(gastrula18h, add.cell.id = "gastrula18h")  
 gastrula18h@meta.data$orig.ident <- 'gastrula18h'  
 gastrula18h.dimplot = DimPlot(gastrula18h, label = F,  
 cols = rep(LibCP[2],  
 length(gastrula18h@active.ident)))+NoLegend()+NoAxes()  
   
 load(file = 'Z:/Personal\_Folders/Alison\_JuliaS/RObj\_SingleDatasets/updated\_mixedfilters/gast2\_250\_AGC.Robj')  
 Gastrula2@assays$RNA@counts@Dimnames[[1]] = genes$gene\_short\_name  
 Gastrula2@assays$RNA@data@Dimnames[[1]] = genes$gene\_short\_name  
 Gastrula2@meta.data$orig.ident <- 'gastrula25h'  
 Gastrula2.dimplot = DimPlot(Gastrula2, label = F,  
 cols = rep(LibCP[4],  
 length(Gastrula2@active.ident)))+NoLegend()+NoAxes()  
   
 load (file = 'Z:/Personal\_Folders/Alison\_JuliaS/RObj\_SingleDatasets/updated\_mixedfilters/gast3\_250.Robj')  
 gastrula3 = gast3  
 rm(gast3)  
 gastrula3@assays$RNA@counts@Dimnames[[1]] = genes$gene\_short\_name  
 gastrula3@assays$RNA@data@Dimnames[[1]] = genes$gene\_short\_name  
 gastrula3.dimplot = DimPlot(gastrula3, label = F,cols = rep(LibCP[3],  
 length(gastrula3@active.ident)))+NoLegend()+NoAxes()  
   
 gastrula3 <- RenameCells(gastrula3, add.cell.id = "gastrula3")  
 gastrula3@meta.data$orig.ident <- 'gastrula24h'  
   
 load (file = 'Z:/Personal\_Folders/Alison\_JuliaS/RObj\_SingleDatasets/planula2d.Robj')  
 planula2d@assays$RNA@counts@Dimnames[[1]] = genes$gene\_short\_name  
 planula2d@assays$RNA@data@Dimnames[[1]] = genes$gene\_short\_name  
 planula2d <- RenameCells(planula2d, add.cell.id = "planula2d")  
 planula2d@meta.data$orig.ident <- 'planula2d'  
 planula2d.dimplot = DimPlot(planula2d, label = F,cols = rep(LibCP[5],  
 length(planula2d@active.ident)))+NoLegend()+NoAxes()  
   
 load (file = 'Z:/Personal\_Folders/Alison\_JuliaS/RObj\_SingleDatasets/updated\_mixedfilters/midpla\_3dpf\_250.Robj') #missing!  
 planula3d <- midpla  
 rm(midpla)  
 planula3d@assays$RNA@counts@Dimnames[[1]] = genes$gene\_short\_name  
 planula3d@assays$RNA@data@Dimnames[[1]] = genes$gene\_short\_name  
 planula3d <- RenameCells(planula3d, add.cell.id = "planula3d")  
 planula3d@meta.data$orig.ident <- 'planula3d'  
 planula3d.dimplot = DimPlot(planula3d, label = F,cols = rep(LibCP[6],  
 length(planula3d@active.ident)))+NoLegend()+NoAxes()  
   
 load (file = 'Z:/Personal\_Folders/Alison\_JuliaS/RObj\_SingleDatasets/updated\_mixedfilters/plalive\_4dpf\_300.Robj')  
 planula4d <- plalive  
 rm(plalive)  
 planula4d@assays$RNA@counts@Dimnames[[1]] = genes$gene\_short\_name  
 planula4d@assays$RNA@data@Dimnames[[1]] = genes$gene\_short\_name  
 planula4d <- RenameCells(planula4d, add.cell.id = "planula4d")  
 planula4d.dimplot = DimPlot(planula4d, label = F,cols = rep(LibCP[7],  
 length(planula4d@active.ident)))+NoLegend()+NoAxes()  
   
 planula4d@meta.data$orig.ident <- 'planula4d'  
   
 load(file = 'Z:/Personal\_Folders/Alison\_JuliaS/RObj\_SingleDatasets/updated\_mixedfilters/placryo\_4dpf\_300.Robj')  
 planula4d2c = placryo  
 rm(placryo)  
 planula4d2c@assays$RNA@counts@Dimnames[[1]] = genes$gene\_short\_name  
 planula4d2c@assays$RNA@data@Dimnames[[1]] = genes$gene\_short\_name  
 planula4d2c <- RenameCells(planula4d2c, add.cell.id = "planula4d2c")  
 planula4d2c@meta.data$orig.ident <- 'planula4d2c'  
 planula4d2c.dimplot = DimPlot(planula4d2c, label = F,cols = rep(LibCP[8],  
 length(planula4d2c@active.ident)))+NoLegend()+NoAxes()  
   
   
 load (file = 'Z:/Personal\_Folders/Alison\_JuliaS/RObj\_SingleDatasets/updated\_mixedfilters/tentbud\_5dpf\_250.Robj')  
 planula5d <- tentbud  
 rm(tentbud)  
 planula5d@assays$RNA@counts@Dimnames[[1]] = genes$gene\_short\_name  
 planula5d@assays$RNA@data@Dimnames[[1]] = genes$gene\_short\_name  
   
 planula5d <- RenameCells(planula5d, add.cell.id = "planula5d")  
 planula5d.dimplot = DimPlot(planula5d, label = F,cols = rep(LibCP[9],  
 length(planula5d@active.ident)))+NoLegend()+NoAxes()  
   
 planula5d@meta.data$orig.ident <- 'tentaclebud5d'  
   
   
 load (file = 'Z:/Personal\_Folders/Alison\_JuliaS/RObj\_SingleDatasets/updated\_mixedfilters/pol12\_8dpf\_250.Robj')  
 polyp8d <-pol12  
 rm(pol12)  
 polyp8d@assays$RNA@counts@Dimnames[[1]] = genes$gene\_short\_name  
 polyp8d@assays$RNA@data@Dimnames[[1]] = genes$gene\_short\_name  
 polyp8d <- RenameCells(polyp8d, add.cell.id = "polyp8d")  
 polyp8d@meta.data$orig.ident <- 'polyp8d'  
 polyp8d.dimplot = DimPlot(polyp8d, label = F,cols = rep(LibCP[10],  
 length(polyp8d@active.ident)))+NoLegend()+NoAxes()  
   
   
 load (file = 'Z:/Personal\_Folders/Alison\_JuliaS/RObj\_SingleDatasets/updated\_mixedfilters/pol3\_16dpf\_250.Robj')  
 polyp16d <- pol3  
 rm(pol3)  
 polyp16d@assays$RNA@counts@Dimnames[[1]] = genes$gene\_short\_name  
 polyp16d@assays$RNA@data@Dimnames[[1]] = genes$gene\_short\_name  
 polyp16d <- RenameCells(polyp16d, add.cell.id = "polyp16d")  
 polyp16d@meta.data$orig.ident <- 'polyp16d'  
 polyp16d.dimplot = DimPlot(polyp16d, label = F,cols = rep(LibCP[11],  
 length(polyp16d@active.ident)))+NoLegend()+NoAxes()  
   
   
 load (file = 'Z:/Personal\_Folders/Alison\_JuliaS/RObj\_SingleDatasets/updated\_mixedfilters/bodywall\_250.Robj')  
 bw@assays$RNA@counts@Dimnames[[1]] = genes$gene\_short\_name  
 bw@assays$RNA@data@Dimnames[[1]] = genes$gene\_short\_name  
 bw <- RenameCells(bw, add.cell.id = "bodywall")  
 bw@meta.data$orig.ident <- 'bodywall'  
 bw.dimplot = DimPlot(bw, label = F,cols = rep(LibCP[14],  
 length(bw@active.ident)))+NoLegend()+NoAxes()  
   
   
 load (file = 'Z:/Personal\_Folders/Alison\_JuliaS/RObj\_SingleDatasets/updated\_mixedfilters/mesentery\_250.Robj')  
 mes@assays$RNA@counts@Dimnames[[1]] = genes$gene\_short\_name  
 mes@assays$RNA@data@Dimnames[[1]] = genes$gene\_short\_name  
 mes.dimplot = DimPlot(mes, label = F,cols = rep(LibCP[16],  
 length(mes@active.ident)))+NoLegend()+NoAxes()  
   
   
 load (file = 'Z:/Personal\_Folders/Alison\_JuliaS/RObj\_SingleDatasets/updated\_mixedfilters/pharynx\_250.Robj')  
 phar<- pha  
 rm(pha)  
 phar@assays$RNA@counts@Dimnames[[1]] = genes$gene\_short\_name  
 phar@assays$RNA@data@Dimnames[[1]] = genes$gene\_short\_name  
 phar <- RenameCells(phar, add.cell.id = "pharynx")  
 phar@meta.data$orig.ident <- 'pharynx'  
 phar.dimplot = DimPlot(phar, label = F,cols = rep(LibCP[13],  
 length(phar@active.ident)))+NoLegend()+NoAxes()  
   
   
 load (file = 'Z:/Personal\_Folders/Alison\_JuliaS/RObj\_SingleDatasets/updated\_mixedfilters/tentacle\_250.Robj')  
 tentacle=tent  
 rm(tent)  
 tentacle@assays$RNA@counts@Dimnames[[1]] = genes$gene\_short\_name  
 tentacle@assays$RNA@data@Dimnames[[1]] = genes$gene\_short\_name  
 tentacle <- RenameCells(tentacle, add.cell.id = "tentacle")  
 tentacle@meta.data$orig.ident <- 'tentacle'  
 tentacle.dimplot = DimPlot(tentacle, label = F,cols = rep(LibCP[15],  
 length(tentacle@active.ident)))+NoLegend()+NoAxes()  
   
 load (file = 'Z:/Personal\_Folders/Alison\_JuliaS/RObj\_SingleDatasets/pharbw.Robj')  
 phbw=pharbw  
 rm(pharbw)  
 phbw@assays$RNA@counts@Dimnames[[1]] = genes$gene\_short\_name  
 phbw@assays$RNA@data@Dimnames[[1]] = genes$gene\_short\_name  
 phbw@meta.data$orig.ident <- 'pharynx/bw'  
 phbw.dimplot = DimPlot(phbw, label = F,cols = rep(LibCP[12],  
 length(phbw@active.ident)))+NoLegend()+NoAxes()  
 load(file = 'FemaleMes.Robj')  
 mesF.dimplot = DimPlot(mesF, label = F,cols = rep(LibCP[17],  
 length(mesF@active.ident)))+NoLegend()+NoAxes()  
   
 }   
   
 #generate the merged object  
 data1 <- merge (x=gastrula18h,y= c(Gastrula2,gastrula3,planula2d,planula3d,planula4d2c,planula4d,planula5d,polyp8d,polyp16d,  
 phbw,bw,mes,phar,tentacle,mesF),   
 merge.data = F)  
 #make sure all genes were properly merged. Should be the same length as genes$NVE  
 length(data1@assays$RNA@counts@Dimnames[[1]])  
   
 #clean up the workspace  
 rm (gastrula3, Gastrula2,gastrula18h, planula2d,planula3d,planula4d2c,planula4d,planula5d,polyp8d,polyp16d,bw,mes,phar,tentacle,phbw,mesF)  
 rm (gastrula18h.dimplot, Gastrula2.dimplot,gastrula3.dimplot, planula2d.dimplot,planula3d.dimplot,planula4d.dimplot,planula5d.dimplot,polyp8d.dimplot,polyp16d.dimplot,bw.dimplot,mes.dimplot,phar.dimplot,phbw.dimplot,tentacle.dimplot,mesF.dimplot)  
  
 #names assigned above:   
 lib.order = c("gastrula18h", "gastrula24h", "gastrula25h",  
 "planula2d", "planula3d" ,'planula4d','planula4d2c',   
 'tentaclebud5d','polyp8d','polyp16d',  
 'pharynx/bw','pharynx','bodywall','tentacle','mesentery','mesF')   
   
 # new names for paper:  
 lib.names = c("D|18h gastrula","D|24h gastrula","D|25hr gastrula","D|2d planula",  
 "D|3d planula","D|4d planula","D|4d.c planula", "D|5d p.polyp","D|8d p.polyp","D|16d p.polyp",  
 "T|pharynx/bw","T|pharynx","T|bodywall","T|tentacle","T|mesentery",'T|fem.mesentery')  
   
 # set the library order according to time:  
 data1@meta.data$orig.ident<-as.factor(data1@meta.data$orig.ident)  
 levels(data1$orig.ident)  
 order.ind=match(lib.order,levels(data1$orig.ident))  
   
 data1@meta.data$orig.ident = factor(data1@meta.data$orig.ident,  
 levels(data1@meta.data$orig.ident)[order.ind])  
   
 #look at quality of the dataset:   
 VlnPlot(object = data1, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mt"),   
 ncol = 3, group.by = 'orig.ident',cols = LibCP)  
   
 #normalize the dataset (merged)   
 data1 <- NormalizeData(data1, scale.factor = 10000)  
   
}  
  
## analyze the full dataset - identify coarse clustering  
if (PRE\_ANALYSIS)   
{  
   
 #generate variable gene list from all libraries separately:  
 list= NULL  
 vargenelist <- SplitObject(data1, split.by = "orig.ident")  
 for (i in 1:length(vargenelist)) {  
 vargenelist[[i]] <- NormalizeData(vargenelist[[i]], verbose = FALSE)  
 vargenelist[[i]] <- FindVariableFeatures(vargenelist[[i]], selection.method = "vst",  
 nfeatures = 1000, verbose = FALSE)   
 }  
 #collate into a single list and import into the object:  
 for (i in 1:length(vargenelist)) {  
 x <- vargenelist[[i]]@assays$RNA@var.features  
 list=c(list,x)}  
 var.features.list=unique(list)  
 length(var.features.list)  
 data1@assays$RNA@var.features = var.features.list  
 data1@misc$var.genes = var.features.list  
  
 #Scale variable features in library-specific sets:  
 data1 <- ScaleData(data1, features = data1@assays$RNA@var.features,   
 split.by = 'orig.ident', do.scale = T, do.center = T)  
   
 #Run reductions:  
 data1 <- RunPCA(data1, pcs.compute = 100)  
 ElbowPlot(object = data1, ndims = 50)  
 DimPlot(data1, dims = c(1,2), reduction = 'pca', group.by = 'orig.ident')  
 d=30   
   
 #UMAPS  
 data1 <- RunUMAP(data1, n.neighbors = 30L,spread = 0.4,min.dist = 0.15,  
 seed.use = 42, reduction='pca',  
 metric = 'cosine',local.connectivity = 100,  
 dims = c(1:d), n.components = 2,   
 reduction.name = 'umap2d')  
  
 #cluster the data  
 data1 <- FindNeighbors(object = data1,reduction ="pca",dims = c(1:30),  
 nn.method = 'annoy', annoy.metric = 'cosine',  
 k.param = 60)  
  
 data1 <- FindClusters(object = data1,resolution = 0.2,random.seed = 0)#  
   
 data1 <- BuildClusterTree(object = data1, features= intersect(TF\_list$gene\_short\_name,data1@misc$var.genes),#dims = c(1:30) ,  
 reorder = T, reorder.numeric = T)  
  
   
 #also generate a 3D UMAP topology:  
 use.3d = T  
 if (use.3d)  
 {  
 data1 <- RunUMAP(data1, n.neighbors = 30L,spread = 0.4,min.dist = 0.15,  
 seed.use = 42, reduction='pca',  
 metric = 'cosine',local.connectivity = 10,  
 dims = c(1:30), n.components = 3,   
 reduction.name = 'umap3d')  
   
 UMAP\_1 <- data1@reductions$umap3d@cell.embeddings[,1]  
 UMAP\_2 <- data1@reductions$umap3d@cell.embeddings[,2]  
 UMAP\_3 <- data1@reductions$umap3d@cell.embeddings[,3]  
   
 library(scatterplot3d)  
 library(viridis)  
   
 # colour library  
 color.library = as.numeric(data1@meta.data$orig.ident)  
 cp=LibCP  
 for (i in 1:length(cp))  
 { color.library[color.library==i]<-cp[i]}  
   
 # colour clusters  
 color.clusters = as.numeric(data1@active.ident)  
 cp <- clust.cp.graded  
 for (i in 1:length(cp))  
 { color.clusters[color.clusters==i]<-cp[i]}  
   
 # colour gene  
 GOI ='NvSoxC'  
 color.gene = round(as.numeric(log2(1+data1@assays$RNA@counts[GOI,])))   
 cp <-c('lightgrey',rev(brewer.pal(11 , "Spectral" )))  
 for (i in 0:length(cp))   
 { color.gene[color.gene==i]<-cp[i+1]}  
   
 # choose which you want to use  
 color = color.gene   
 library(rgl)  
 par3d(windowRect = c(20, 30, 800, 800))  
 plot3d(x = UMAP\_1, y = UMAP\_2, z = UMAP\_3, setLab = F, xlab = NULL,ylab = NULL,zlab = NULL,  
 box = F, lwd = 0, axes=F, tick.marks = F,grid = F,  
 col = color,type = "p", size = 1)  
 M <- par3d("userMatrix")  
   
 movie3d(spin3d(axis = c(1,1,1)), duration = 10,  
 dir = "Robjects/")  
   
 }   
  
 # assign cluster names:  
 clusterNames<- read\_excel("SI2\_DEG.xlsx",  
 sheet = 'Fig1C\_fulldataset\_AUTO\_ANNO')  
 goi = clusterNames$Marker  
   
 data1 <- SetIdent(data1, value = 'tree.ident')  
 data1 <- BuildClusterTree(data1,reorder = T,reorder.numeric = T, dims= c(1:30))  
   
 # assign cluster ID to the individual libraries from mean cluster expression of marker genes  
  
 data1<-ScaleData(data1,features = goi, split.by = 'orig.ident')  
 cl <-length(levels(data1@active.ident))  
 C.suffix <-seq(1:cl)  
   
 g=length(goi)  
 clName = vector()  
 m=matrix(0L,g,cl)  
 for (j in 1:cl)  
 {  
 for (i in 1:g)  
 m[i,j]=mean(data1@assays$RNA@scale.data[goi[i],WhichCells(data1,idents = C.suffix[j])])  
 clName[j]=as.integer(which.max(m[,j]))  
 }  
   
 # check that this is what was expected:  
 DimPlot(data1,reduction = 'umap2d',cols=clust.cp.separate)+NoAxes()  
 sort(clName)  
 clusterNames$ID[clName]  
   
 #order clusters according to spreadsheet:  
 data1@active.ident = factor(data1@active.ident,  
 levels(data1@active.ident)[order(clName)])  
 #set the names to your IDs  
 levels(data1@active.ident) = clusterNames$ID[clName][order(clName)]  
   
 #save the IDs in metadata:  
 data1@meta.data$IDs = data1@active.ident  
   
 #check that everything worked:  
 DimPlot(data1,reduction = 'umap2d',cols=clust.cp.separate)+NoAxes()  
  
 #Save the dataset:   
 AllData = data1  
 save(AllData,file = 'Robjects/AllData.Robj')  
}   
  
## generate the figures for the paper for this part  
if (generate.figure1)  
{  
  
 # Fig1C.2   
 print(pie(table(Idents(AllData)), col = CLcp, labels = NULL ))  
   
 ids.cluster.library.AllData = as.data.frame(table(Idents(AllData), AllData@meta.data$orig.ident))  
 colnames(ids.cluster.library.AllData) = c('ID','Library','CellCount')  
   
 #generate your library and cluster UMAP plots:  
 Fig1B = DimPlot(AllData,group.by = 'orig.ident',  
 order = rev(levels(AllData$orig.ident)),  
 cols = (LibCP))+NoAxes()+  
 labs(title = 'Time | Library origin')+NoLegend()  
print(Fig1B)  
  
 Fig1Ca =DimPlot(AllData, group.by = 'IDs',reduction = 'umap2d',   
 cols = CLcp)+NoAxes()+  
 labs(title = 'Clusters | ID')+NoLegend()  
  
print(Fig1Ca)  
  
 Fig1C=  
 ggplot(ids.cluster.library.AllData, aes(fill=ID, y= CellCount,  
 x=Library)) +  
 geom\_bar(mapping =aes(fill=ID, y= (CellCount),  
 x=(Library)),  
 position="fill", stat="identity", width = 0.5)+  
 scale\_fill\_manual(values = clust.cp)+  
 theme(axis.text.x = element\_text(#face="bold", color="#993333",   
 size=8, angle=-45,hjust=0,vjust = 0.5))+  
 geom\_area(mapping =aes(fill=ID, y= (CellCount),  
 x=as.integer(Library)),  
 position="fill", stat="identity",alpha=0.2 , size=.5, colour="white") +  
 geom\_bar(mapping =aes(fill=ID, y= (CellCount),#this re-plots the bars over the area  
 x=(Library)),  
 position="fill", stat="identity", width = 0.5)+  
 ggtitle("Distribution of cell types in time and space")  
   
print(Fig1C)  
  
 #barplot of library identities in each cluster: not included in paper  
 dist.lib=ggplot(ids.cluster.library.AllData, aes(fill=Library, y=(CellCount), x=ID)) +   
 geom\_bar(position="fill", stat="identity")+scale\_fill\_manual(values = (LibCP))+  
 theme(axis.text.x = element\_text(#face="bold", color="#993333",   
 size=8, angle=-45,hjust=0,vjust = 0.5))  
 leg <- ggpubr::get\_legend(dist.lib)  
 # Convert to a ggplot and print  
 Fig1B.legend.lib=ggpubr::as\_ggplot(leg)  
   
print(Fig1B.legend.lib)  
  
 #make sure the variable genes are set:   
 AllData@assays$RNA@var.features = AllData@misc$var.genes  
   
 #generate gene list:  
 all.markers.Alldata <- FindAllMarkers(AllData,  
 logfc.threshold = 1,  
 features = AllData@assays$RNA@var.features,  
 return.thresh = 0.0001,min.pct = 0.2,max.cells.per.ident = 200,  
 only.pos = TRUE)  
   
 # add GO terms and NVEs associated with this list:  
 all.markers.Alldata$go.annotation <- 'NA'  
 all.markers.Alldata$NVE <- 'NA'  
 for (i in 1:length(levels(AllData@active.ident))) #   
 {  
 x=all.markers.Alldata[as.numeric(all.markers.Alldata$cluster)==i,][1:length(which(as.numeric(all.markers.Alldata$cluster)==i)),7]  
 anInd = match(genes[match(x,genes$gene\_short\_name),1],annotations$NVE)  
 print(unique(annotations$gene\_ontology\_pfam[anInd]))  
 all.markers.Alldata[as.numeric(all.markers.Alldata$cluster)==i,][1:length(which(as.numeric(all.markers.Alldata$cluster)==i)),8]<-annotations$gene\_ontology\_pfam[anInd]  
 all.markers.Alldata[as.numeric(all.markers.Alldata$cluster)==i,][1:length(which(as.numeric(all.markers.Alldata$cluster)==i)),9]<-annotations$NVE[anInd]  
 }   
  
 #also generate TFs only:   
 all.markers.TF <- FindAllMarkers(AllData,  
 features = TF\_list$gene\_short\_name,  
 return.thresh = 0.001,min.pct = 0.01,max.cells.per.ident = 200,  
 only.pos = TRUE)  
   
 # add GO terms and NVEs associated with this list:  
 all.markers.TF$go.annotation <- 'NA'  
 all.markers.TF$NVE <- 'NA'  
 for (i in 1:length(levels(AllData@active.ident))) #   
 {  
 x=all.markers.TF[as.numeric(all.markers.TF$cluster)==i,][1:length(which(as.numeric(all.markers.TF$cluster)==i)),7]  
 anInd = match(genes[match(x,genes$gene\_short\_name),1],annotations$NVE)  
 print(unique(annotations$gene\_ontology\_pfam[anInd]))  
 all.markers.TF[as.numeric(all.markers.TF$cluster)==i,][1:length(which(as.numeric(all.markers.TF$cluster)==i)),8]<-annotations$gene\_ontology\_pfam[anInd]  
 all.markers.TF[as.numeric(all.markers.TF$cluster)==i,][1:length(which(as.numeric(all.markers.TF$cluster)==i)),9]<-annotations$NVE[anInd]  
 }  
  
 list = NULL  
 all.markers\_variable = all.markers.Alldata  
 for (i in 1:length(levels(AllData@active.ident)))  
 {  
 x=all.markers\_variable[as.numeric(all.markers\_variable$cluster)==i,][1:min(5,length(which(as.numeric(all.markers\_variable$cluster)==i))),7]  
 if (is.na (x) ==F)  
 list=c(list,x)  
 }  
 list <- unique(c(list))  
   
 #Image the list  
 Fig1D = DotPlot(AllData, features = unique(c(list)),   
 scale.by='size' , col.min = 0, col.max = 3,   
 cols = c('lightgrey','darkred')) +   
 RotatedAxis() +FontSize(6,6) +  
 labs(title = 'Top 5 Markers',subtitle = 'p-val < 0.0001 | log.fc >1')  
print(Fig1D)   
  
 if(run.save)  
 {  
 write.csv(all.markers.Alldata, file = 'Robjects/AllData\_DEGenes.csv')  
 }  
   
 }  
  
## separate out the cnidocyte lineage and run analysis of the subset  
if (Figure2.Cnidocyte)  
{  
## generate the dataset:  
 {  
#pull out the nematocytes  
 levels(AllData)  
 coi=WhichCells(AllData,idents=levels(AllData)[c(11,12)])  
 coi.ind=match(coi,colnames(AllData@assays$RNA@counts))  
 nematocytes=CreateSeuratObject(AllData@assays$RNA@counts[,coi])  
 nematocytes@meta.data$orig.ident = AllData@meta.data$orig.ident[coi.ind]  
 nematocytes@active.ident = AllData@active.ident[coi.ind]  
 nematocytes <- NormalizeData(nematocytes, scale.factor = 10000)  
   
 #calculate variable genes  
 nematocytes <- FindVariableFeatures(nematocytes, nfeatures = 2000,selection.method = 'vst')#  
 #scale those genes in full dataset:  
 t=ScaleData(AllData,model.use = 'linear', use.umi = F,  
 split.by = 'orig.ident', features = nematocytes@assays$RNA@var.features)  
 #import scaling to subset:  
 nematocytes@assays$RNA@scale.data = t@assays$RNA@scale.data[,coi]  
 #run reductions  
 nematocytes <- RunPCA(nematocytes, pcs.compute = 50)  
 ElbowPlot(nematocytes,ndims = 50)  
 # set dimensions  
 d=20  
 #UMAPS  
 nematocytes <- RunUMAP(nematocytes, n.neighbors = 20L,spread = 0.12,  
 seed.use = 42, dims = 1:d,min.dist = 0.08,  
 metric = 'cosine', local.connectivity = 1)  
  
 #clustering:  
 nematocytes <- FindNeighbors(object = nematocytes,reduction ="pca",dims = 1:d,  
 nn.method = 'annoy', annoy.metric = 'cosine',  
 k.param = 10)  
   
 #celltypes:  
 nematocytes <- FindClusters(object = nematocytes,resolution = 0.1,random.seed = 0)  
 nematocytes <- BuildClusterTree(object = nematocytes, reorder = TRUE, reorder.numeric = T,  
 features = intersect(TF\_list$gene\_short\_name,  
 nematocytes@assays$RNA@var.features))  
 #assign cluster names:  
 nem.clusterNames<- read\_excel("SI2\_DEG.xlsx",  
 sheet = 'Fig2B\_cnido\_AUTO\_ANNO')  
 goi = nem.clusterNames$Marker  
 nematocytes<-ScaleData(nematocytes,features = goi)#, split.by = 'orig.ident')  
 cl <-length(levels(nematocytes@active.ident))  
 C.suffix <-seq(1:cl)  
   
 g=length(goi)  
 clName = vector()  
 m=matrix(0L,g,cl)  
 for (j in 1:cl)  
 {  
 for (i in 1:g)  
 m[i,j]=mean(nematocytes@assays$RNA@scale.data[goi[i],WhichCells(nematocytes,idents = C.suffix[j])])  
 clName[j]=as.integer(which.max(m[,j]))  
 }  
 #check that it worked as expected:  
 DimPlot(nematocytes,cols=clust.cp.separate,label=T)+NoAxes()  
 sort(clName)   
 nem.clusterNames$ID[clName]  
   
 #order the identities..  
 nematocytes@active.ident = factor(nematocytes@active.ident,  
 levels(nematocytes@active.ident)[order(clName)])  
 #set the names to your IDs  
 levels(nematocytes@active.ident) = nem.clusterNames$ID[clName][order(clName)]  
   
 #save the IDs in metadata:  
 nematocytes@meta.data$IDs = nematocytes@active.ident  
 DimPlot(nematocytes, label = T,label.size = 5,   
 repel = T,order=rev(levels(nematocytes@active.ident)),  
 cols = clust.cp.separate)+NoAxes()+  
 labs(title = 'Clusters | ID')+NoLegend()  
   
 #calculate differentially expressed genes:  
 nematocytes@active.assay='RNA'  
 nem.markers <- FindAllMarkers(nematocytes,logfc.threshold = 1,  
 return.thresh = 0.00001,min.pct = 0.3,  
 only.pos = TRUE)  
 #add annotations  
 {  
 # add GO terms associated with this list:  
 nem.markers$go.annotation <- 'NA'  
 nem.markers$NVE <- 'NA'  
 for (i in 1:length(levels(nematocytes@active.ident))) #   
 {  
 x=nem.markers[as.numeric(nem.markers$cluster)==i,][1:length(which(as.numeric(nem.markers$cluster)==i)),7]  
 anInd = match(genes[match(x,genes$gene\_short\_name),1],annotations$NVE)  
 print(unique(annotations$gene\_ontology\_pfam[anInd]))  
 nem.markers[as.numeric(nem.markers$cluster)==i,][1:length(which(as.numeric(nem.markers$cluster)==i)),8]<-annotations$NVE[anInd]  
 nem.markers[as.numeric(nem.markers$cluster)==i,][1:length(which(as.numeric(nem.markers$cluster)==i)),9]<-annotations$gene\_ontology\_pfam[anInd]  
 }   
 }  
   
 nem.markers.TF <- FindAllMarkers(nematocytes,logfc.threshold = 0.4,  
 features = TF\_list$gene\_short\_name,  
 return.thresh = 0.001,min.pct = 0.05,  
 only.pos = TRUE)  
 #add annotations  
 {  
 # add GO terms associated with this list:  
 nem.markers.TF$go.annotation <- 'NA'  
 nem.markers.TF$NVE <- 'NA'  
 for (i in 1:length(levels(nematocytes@active.ident))) #   
 {  
 x=nem.markers.TF[as.numeric(nem.markers.TF$cluster)==i,][1:length(which(as.numeric(nem.markers.TF$cluster)==i)),7]  
 anInd = match(genes[match(x,genes$gene\_short\_name),1],annotations$NVE)  
 print(unique(annotations$gene\_ontology\_pfam[anInd]))  
 nem.markers.TF[as.numeric(nem.markers.TF$cluster)==i,][1:length(which(as.numeric(nem.markers.TF$cluster)==i)),8]<-annotations$NVE[anInd]  
 nem.markers.TF[as.numeric(nem.markers.TF$cluster)==i,][1:length(which(as.numeric(nem.markers.TF$cluster)==i)),9]<-annotations$gene\_ontology\_pfam[anInd]  
 }   
 }  
   
 write.csv(nem.markers, file = 'Robjects/cnido\_DEGenes.csv')  
 write.csv(nem.markers.TF, file = 'Robjects/cnido\_DEGenesTF.csv')  
  
 # use MAGIC to infer gene expression for expression plots:  
 nemat.transg = c('NvSoxC','NvSox2','NvKlf-spotty','NvMMP1')  
 toxin.nem = c('TX60B-like3','TX60B-like5','NvNEP8','NvNEP3-like','ANTR2-like','NvNEP3')  
 nematocytes=magic(nematocytes,genes=unique(c(toxin.nem,nemat.transg,  
 nem.markers$gene,nem.markers.TF$gene)))  
   
 #calculate differentiation score as proxy for pseudotime:  
 library(CytoTRACE)  
 cyto<-CytoTRACE(as.matrix(nematocytes@assays$RNA@counts))  
 #import from cyto to nematocytes  
 nematocytes@meta.data$cytoTRACE = cyto$CytoTRACE  
 FeaturePlot(nematocytes,'cytoTRACE', label=F,label.size = 6,  
 pt.size = 1, reduction = 'umap',   
 cols = rev(magma(11)))+NoAxes()+labs(title = 'Pseuodtime | Cytotrace')  
   
 #calculate trajectories:  
 library(monocle3)  
 cds <- SeuratWrappers::as.cell\_data\_set(nematocytes)  
 cds <- cluster\_cells(cds = cds, reduction\_method = "UMAP",resolution = 0.01) #tissue/dev res=0.01; gast = 0.001  
 cds <- learn\_graph(cds, use\_partition = F,close\_loop=F)  
 cds.cnido = cds  
   
 if(run.save)  
 {  
 save(cds.cnido,file ='Robjects/monocle.cnido.RObj')  
 save(nematocytes,file = 'Robjects/cnido.RObj')  
 }  
 }  
  
## generate the figures:   
 {  
   
 ids.cluster.library.nem = as.data.frame(table(Idents(nematocytes), nematocytes@meta.data$orig.ident))  
 colnames(ids.cluster.library.nem) = c('ID','Library','CellCount')  
  
 #Fig.2B.1  
 print(pie(table(Idents(nematocytes)),   
 col = clust.cp.separate,labels= NULL, radius = 1 ))   
 Fig2A=DimPlot(AllData,cells.highlight = nematocytes@assays$RNA@counts@Dimnames[[2]],  
 cols.highlight = clust.cp[11])+NoLegend()+NoAxes()  
print(Fig2A)   
 #barplot of library identities in each cluster:  
 dist.lib=ggplot(ids.cluster.library.nem, aes(fill=Library, y=(CellCount), x=ID)) +   
 geom\_bar(position="fill", stat="identity")+scale\_fill\_manual(values = (LibCP))+  
 theme(axis.text.x = element\_text(#face="bold", color="#993333",   
 size=8, angle=-45,hjust=0,vjust = 0.5))  
 leg.lib <- ggpubr::get\_legend(dist.lib)  
 # Convert to a ggplot and print  
 Fig2.legend.lib=ggpubr::as\_ggplot(leg.lib)  
print(Fig2.legend.lib)   
 Fig2B.1 = DimPlot(nematocytes, label = F,label.size = 5, repel = T,order=rev(levels(nematocytes@active.ident)),  
 cols = clust.cp.separate)+NoLegend()+NoAxes()#+labs(title = 'Clusters | ID')  
print(Fig2B.1)   
 Fig.2Ea = plot\_cells(cds.cnido,   
 color\_cells\_by = 'cytoTRACE',   
 label\_cell\_groups=F,   
 label\_leaves=F,  
 label\_branch\_points=F,  
 label\_roots = F,  
 trajectory\_graph\_color = 'cyan',  
 trajectory\_graph\_segment\_size = 1,  
 cell\_size = 2,  
 cell\_stroke = 0,  
 )+scale\_color\_viridis(option = 'A',discrete = F,direction = -1)  
print(Fig.2Ea)   
 cyto.plot= FeaturePlot(nematocytes,'cytoTRACE', label=F,label.size = 6,  
 pt.size = 1, reduction = 'umap',   
 cols = rev(magma(11 )))+NoAxes()+  
 labs(title = 'Pseuodtime | Cytotrace')  
 pseudotime.scale <- ggpubr::get\_legend(cyto.plot)  
 # Convert to a ggplot and print  
 Fig2.legend.pseudotime=ggpubr::as\_ggplot(pseudotime.scale)  
 Fig2B=  
 ggplot(ids.cluster.library.nem, aes(fill=ID, y= CellCount,  
 x=Library)) +  
 geom\_bar(mapping =aes(fill=ID, y= (CellCount),  
 x=(Library)),  
 position="fill", stat="identity", width = 0.5)+  
 scale\_fill\_manual(values = clust.cp.separate)+  
 theme(axis.text.x = element\_text(  
 size=8, angle=-45,hjust=0,vjust = 0.5))+  
 geom\_area(mapping =aes(fill=ID, y= (CellCount),  
 x=as.integer(Library)),  
 position="fill", stat="identity",alpha=0.2 , size=.5, colour="white") +  
 geom\_bar(mapping =aes(fill=ID, y= (CellCount),  
 x=(Library)),  
 position="fill", stat="identity", width = 0.5)+  
 ggtitle("Distribution of cell types in time and space")   
 print(Fig2B)   
 leg.cl <- ggpubr::get\_legend(Fig2B)  
 # Convert to a ggplot and print  
 Fig2.legend.cl=ggpubr::as\_ggplot(leg.cl)  
   
 #generate a list of DEGs for plotting:  
 list = NULL  
 for (i in 1:length(levels(nematocytes@active.ident)))  
 {  
 x=nem.markers[as.numeric(nem.markers$cluster)==i,][1:min(5,length(which(as.numeric(nem.markers$cluster)==i))),7]  
 if (is.na (x) ==F)  
 list=c(list,x)  
 }  
 #Image the list  
 Fig2D=DotPlot(nematocytes, features = unique(c(list)),   
 scale.by='size' , col.min = 0, col.max = 3,   
 cols = c('lightgrey','darkred')) +   
 RotatedAxis() +FontSize(10,6) +  
 labs(title = 'Top 5 DEGs',subtitle = 'p-val < 0.0001')  
print(Fig2D)  
 #plot the toxins:  
 nematocytes@active.assay='MAGIC\_RNA'  
 Fig2C=FeaturePlot(nematocytes,toxin.nem,label=F,order = T,cols =gene.cp)&  
 NoLegend()&NoAxes()  
   
 #plot the differentiation state genes:  
 goi = c('NvSox2','ATF2-like1','GFI1B-like3','Nvmyc5',  
 'NvKlf-spotty','JUN-like','FOS-like','NvSoxA',  
 'NvMMP1', 'NvNcol3','NvNcol1','NvNcol')  
   
 Fig2F.1=FeaturePlot(nematocytes,goi,label=F,order = T,ncol = 4,cols = gene.cp)&NoLegend()&NoAxes()  
  
 #also single plots for integration into figure file:   
 Fig2F.2=FeaturePlot(nematocytes,'CALM3-like',label=F,order = T,cols =gene.cp)&  
 NoLegend()&NoAxes()  
 Fig2F.3=FeaturePlot(nematocytes,'NvNKx2.2D',label=F,order = T,cols =c('lightgrey',clust.cp.separate[7]))&  
 NoLegend()&NoAxes()  
 Fig2F.4=FeaturePlot(nematocytes,'NvFOXL2',label=F,order = T,cols =c('lightgrey',clust.cp.separate[9]))&  
 NoLegend()&NoAxes()  
 Fig2F.5=FeaturePlot(nematocytes,'NvSix1-2',label=F,order = T,cols =c('lightgrey',clust.cp.separate[5]))&  
 NoLegend()&NoAxes()  
 Fig2F.6=FeaturePlot(nematocytes,'NvFoxA',label=F,order = T,cols =c('lightgrey','black'))&  
 NoLegend()&NoAxes()  
   
 nematocytes@active.assay='RNA'  
 Fig2G.1=DotPlot(nematocytes,features= 'NvSox2', cols=c('lightgrey','darkorange'),  
 scale.by='size')&RotatedAxis()  
 Fig2H.1=DotPlot(nematocytes,features= c('NvKlf-spotty','NvMMP1'), cols=c('lightgrey','red'),  
 scale.by='size')&RotatedAxis()  
 }  
  
}  
  
if (Fig.3.lineages)  
{  
##generate the datasets:  
{  
 # pull out only the ecto. RM:  
 {  
 coi=WhichCells(AllData,idents='retractor muscle')  
 data1=CreateSeuratObject(AllData@assays$RNA@counts[,coi])  
 #standard workflow   
 data1 <- NormalizeData(data1, scale.factor = 5000)  
 #calculate variable genes  
 data1 <- FindVariableFeatures(data1, nfeatures = 2000,selection.method = 'vst')#  
 #use the full dataset scaling:  
 t=ScaleData(AllData,model.use = 'linear', use.umi = F,  
 split.by = 'orig.ident', features = data1@assays$RNA@var.features)  
 data1@assays$RNA@scale.data = t@assays$RNA@scale.data[,coi]  
 data1 <- RunPCA(data1, pcs.compute = 50)  
 data1 <- RunUMAP(data1, n.neighbors = 6L,spread = 0.2,seed.use = 1234, dims = 1:11,min.dist = 0.1,  
 metric = 'cosine')  
 data1 <- FindNeighbors(object = data1,reduction ="pca",dims = 1:11,  
 nn.method = 'annoy', annoy.metric = 'cosine')  
 data1 <- FindClusters(object = data1,resolution = 0.05,random.seed = 0)  
 FeaturePlot(data1, c('NvNem64','NvNem24'), label = T,label.size = 5, repel = T,  
 order=T,cols = gene.cp)&NoLegend()&NoAxes()  
 tRM=WhichCells(data1,idents= '0')   
 }  
   
 # add the rest of the neurosec lineage:  
 {  
 levels(AllData)  
 coi=WhichCells(AllData,idents=levels(AllData)[c(7:12)])  
 coi = (c(coi,tRM))  
  
 coi.ind=match(coi,colnames(AllData@assays$RNA@counts))  
 data1=CreateSeuratObject(AllData@assays$RNA@counts[,coi])  
 data1@meta.data$orig.ident = AllData@meta.data$orig.ident[coi.ind]  
 data1@active.ident = AllData@active.ident[coi.ind]  
  
 #separate the data into three subsets:  
 data1<-SetIdent(data1,value = 'orig.ident')  
 ectderiv.tissue<-subset(data1,cells = WhichCells(data1,idents = c(levels(data1)[11:16])))  
 ectderiv.dev<-subset(data1,cells = WhichCells(data1,idents = c(levels(data1)[4:10])))  
 ectderiv.gast<-subset(data1,cells = WhichCells(data1,idents = c(levels(data1)[1:3])))  
   
 ectderiv.gast$orig.ident = droplevels(as.factor(ectderiv.gast$orig.ident))  
 ectderiv.dev$orig.ident = droplevels(as.factor(ectderiv.dev$orig.ident))  
 ectderiv.tissue$orig.ident = droplevels(as.factor(ectderiv.tissue$orig.ident))  
   
 #standard workflow gast  
 {  
 {  
 coi = colnames(ectderiv.gast@assays$RNA@counts)  
 ectderiv.gast <- NormalizeData(ectderiv.gast, scale.factor = 10000)  
 list= NULL  
 vargenelist <- SplitObject(ectderiv.gast, split.by = "orig.ident")  
 for (i in 1:length(vargenelist)) {  
 vargenelist[[i]] <- FindVariableFeatures(vargenelist[[i]], selection.method = "vst",  
 nfeatures = 1000, verbose = FALSE)  
 }  
 for (i in 1:length(vargenelist)) {  
 x <- vargenelist[[i]]@assays$RNA@var.features  
 list=c(list,x)}  
 list=unique(list)  
 length(list)  
   
 ectderiv.gast@assays$RNA@var.features = list  
 #use the full dataset scaling:  
 t=ScaleData(AllData,model.use = 'linear', use.umi = F,  
 split.by = 'orig.ident', features = ectderiv.gast@assays$RNA@var.features)  
 ectderiv.gast@assays$RNA@scale.data = t@assays$RNA@scale.data[,coi]  
   
 ectderiv.gast <- RunPCA(ectderiv.gast, pcs.compute = 50)  
 ectderiv.gast <- RunUMAP(ectderiv.gast, n.neighbors = 6L,spread = 0.2,seed.use = 10,   
 dims = 1:30,min.dist = 0.1,#saved: d=11  
 metric = 'cosine', local.connectivity = 10)  
 DimPlot(ectderiv.gast,group.by = 'orig.ident',cols=LibCP[5:12]) + NoAxes()+  
 labs(title = 'Library | ID | 10 neighbours')  
 d=30  
   
 ectderiv.gast <- FindNeighbors(object = ectderiv.gast,reduction ="pca",dims = 1:30,  
 nn.method = 'annoy', annoy.metric = 'cosine',  
 k.param = 30)  
 ectderiv.gast <- FindClusters(object = ectderiv.gast,resolution = 1,random.seed = 0)#RM only 0.05 for end/ectosplit  
 ectderiv.gast <- BuildClusterTree(object = ectderiv.gast, reorder = T,   
 reorder.numeric = TRUE, dims = c(1:20))  
 DimPlot(ectderiv.gast, label = T,label.size = 5, repel = T,order=(levels(ectderiv.gast@active.ident)),  
 cols = clust.cp.separate)+NoAxes()+labs(title = 'Clusters | ID')+NoLegend()  
 }   
 #assign cluster names:  
 neursec.clusterNames<- read\_excel("SI2\_DEG.xlsx",  
 # neursec.clusterNames<- read\_excel("SI2\_DEG.xlsx",  
 sheet = 'Fig3\_AUTO\_ANNO')  
   
 goi = neursec.clusterNames$Marker  
 DimPlot(ectderiv.gast,label=T,cols=clust.cp.separate)+NoAxes()  
 #assign cluster ID to the individual libraries  
 ectderiv.gast<-ScaleData(ectderiv.gast,features = goi, split.by = 'orig.ident')  
 cl <-length(levels(ectderiv.gast@active.ident))  
 C.suffix <-seq(1:cl)  
   
 g=length(goi)  
 clName = vector()  
 m=matrix(0L,g,cl)  
 for (j in 1:cl)  
 {  
 for (i in 1:g)  
 m[i,j]=mean(ectderiv.gast@assays$RNA@scale.data[goi[i],WhichCells(ectderiv.gast,idents = C.suffix[j])])  
   
 clName[j]=as.integer(which.max(m[,j]))  
 }  
 sort(clName)  
 neursec.clusterNames$ID[clName]  
   
 ectderiv.gast@active.ident = factor(ectderiv.gast@active.ident,  
 levels(ectderiv.gast@active.ident)[order(clName)])  
 levels(ectderiv.gast@active.ident) = neursec.clusterNames$ID[clName][order(clName)]  
 #save the IDs in metadata:  
 ectderiv.gast@meta.data$IDs = ectderiv.gast@active.ident  
   
 library.plot = DimPlot(ectderiv.gast,group.by = 'orig.ident',pt.size = 1,  
 cols = LibCP[1:4]  
 )+NoAxes()+labs(title = 'Time | Library origin')  
 cluster.plot =  
 DimPlot(ectderiv.gast, label = T,label.size = 5, repel = T,order=(levels(ectderiv.gast@active.ident)),  
 cols = clust.cp.separate)+NoAxes()+labs(title = 'Clusters | ID')  
 cluster.plot  
   
 #generate DEG lists  
 {  
 #generate marker lists for each population (cluster)  
 ectderiv.gast@active.assay='RNA'  
 all.markers <- FindAllMarkers(ectderiv.gast,  
 logfc.threshold = 1,  
 features = ectderiv.gast@assays$RNA@var.features,  
 return.thresh = 0.001,  
 min.pct = 0.2,  
 only.pos = TRUE,   
 max.cells.per.ident = 200,  
 )  
 # add GO terms associated with this list:  
 all.markers$NVE <- 'NA'  
 all.markers$annotation\_notes <- 'NA'  
 all.markers$jgi <- 'NA'  
   
 for (i in 1:length(levels(ectderiv.gast@active.ident))) #   
 {  
 x=all.markers[as.numeric(all.markers$cluster)==i,][1:length(which(as.numeric(all.markers$cluster)==i)),7]  
 anInd = match(genes[match(x,genes$gene\_short\_name),1],annotations$NVE)  
 all.markers[as.numeric(all.markers$cluster)==i,][1:length(which(as.numeric(all.markers$cluster)==i)),8]<-annotations$NVE[anInd]  
 all.markers[as.numeric(all.markers$cluster)==i,][1:length(which(as.numeric(all.markers$cluster)==i)),9]<-annotations$annotation\_notes[anInd]  
 all.markers[as.numeric(all.markers$cluster)==i,][1:length(which(as.numeric(all.markers$cluster)==i)),10]<-annotations$HM\_ID[anInd]  
 }   
   
 #save your gene lists:  
 write.csv(all.markers, file='ectderiv.gast.DEGenes.csv')  
 }  
   
 library(CytoTRACE)  
 cyto<-CytoTRACE(as.matrix(ectderiv.gast@assays$RNA@counts),enableFast = T,subsamplesize = 500)  
 ectderiv.gast@meta.data$cytoTRACE = cyto$CytoTRACE  
  
 library(monocle3)   
 cds <- SeuratWrappers::as.cell\_data\_set(ectderiv.gast)  
 cds <- cluster\_cells(cds = cds, reduction\_method = "UMAP",resolution = 0.005) #tissue/dev res=0.01; gast = 0.001  
 cds <- learn\_graph(cds, use\_partition = F,close\_loop=T)  
 plot\_cells(cds, color\_cells\_by='cytoTRACE')  
 cds.gast = cds   
 }  
 cluster.plot  
 if(run.save)  
 {  
 save (ectderiv.gast, file='Robjects/ectderiv.gast.Robj')  
 save(cds.gast,file='Robjects/monocle.ectderiv.gast.RObj')  
 }  
   
 #standard workflow dev  
 {  
 coi = colnames(ectderiv.dev@assays$RNA@counts)  
 ectderiv.dev <- NormalizeData(ectderiv.dev, scale.factor = 10000)  
   
 list= NULL  
 vargenelist <- SplitObject(ectderiv.dev, split.by = "orig.ident")  
 for (i in 1:length(vargenelist)) {  
 vargenelist[[i]] <- FindVariableFeatures(vargenelist[[i]], selection.method = "vst",  
 nfeatures = 1000, verbose = FALSE)  
 }  
 for (i in 1:length(vargenelist)) {  
 x <- vargenelist[[i]]@assays$RNA@var.features  
 list=c(list,x)}  
 list=unique(list)  
 length(list)  
   
 ectderiv.dev@assays$RNA@var.features = list  
 #use the full dataset scaling:  
 t=ScaleData(AllData,model.use = 'linear', use.umi = F,  
 split.by = 'orig.ident', features = ectderiv.dev@assays$RNA@var.features)  
 ectderiv.dev@assays$RNA@scale.data = t@assays$RNA@scale.data[,coi]  
   
 ectderiv.dev <- RunPCA(ectderiv.dev, pcs.compute = 50)  
 #saved:  
 ectderiv.dev <- RunUMAP(ectderiv.dev, n.neighbors = 10L,spread = 0.1,seed.use = 123,  
 dims = 1:30,min.dist = 0.1,  
 metric = 'cosine', local.connectivity = 10)  
   
 DimPlot(ectderiv.dev, label = T,label.size = 4, repel = T,  
 cols = clust.cp)+NoAxes()+  
 labs(title = 'Clusters | ID')+NoLegend()  
   
 DimPlot(ectderiv.dev,group.by = 'orig.ident',cols=LibCP[4:11]) + NoAxes()+  
 labs(title = 'Library | ID | 10 neighbours')  
 d=30  
   
 ectderiv.dev <- FindNeighbors(object = ectderiv.dev,reduction ="pca",dims = 1:30,  
 nn.method = 'annoy', annoy.metric = 'cosine',  
 k.param = 30)  
 ectderiv.dev <- FindClusters(object = ectderiv.dev,resolution = 1,random.seed = 0)  
 ectderiv.dev <- BuildClusterTree(object = ectderiv.dev, reorder = T,   
 reorder.numeric = TRUE, dims = c(1:20))  
 DimPlot(ectderiv.dev, label = T,label.size = 5, repel = T,order=(levels(ectderiv.dev@active.ident)),  
 cols = clust.cp.separate)+NoAxes()+labs(title = 'Clusters | ID')+NoLegend()  
   
 #assign cluster names:  
 neursec.clusterNames<- read\_excel("SI2\_DEG.xlsx",  
 # neursec.clusterNames<- read\_excel("SI2\_DEG.xlsx",  
 sheet = 'Fig3\_AUTO\_ANNO')  
  
 goi = neursec.clusterNames$Marker  
 #if doover"  
 ectderiv.dev <- SetIdent(ectderiv.dev, value = 'tree.ident')  
 ectderiv.dev <- BuildClusterTree(ectderiv.dev,reorder = T,reorder.numeric = T,  
 dims= c(1:30))  
 DimPlot(ectderiv.dev,label=T,cols=clust.cp.separate)+NoAxes()  
 #assign cluster ID to the individual libraries  
 ectderiv.dev<-ScaleData(ectderiv.dev,features = goi, split.by = 'orig.ident')  
 cl <-length(levels(ectderiv.dev@active.ident))  
 C.suffix <-seq(1:cl)  
   
 g=length(goi)  
 clName = vector()  
 m=matrix(0L,g,cl)  
 for (j in 1:cl)  
 {  
 for (i in 1:g)  
 m[i,j]=mean(ectderiv.dev@assays$RNA@scale.data[goi[i],WhichCells(ectderiv.dev,idents = C.suffix[j])])  
 clName[j]=as.integer(which.max(m[,j]))  
 }  
 sort(clName)  
 neursec.clusterNames$ID[clName]  
   
 ectderiv.dev@active.ident = factor(ectderiv.dev@active.ident,  
 levels(ectderiv.dev@active.ident)[order(clName)])  
 levels(ectderiv.dev@active.ident) = neursec.clusterNames$ID[clName][order(clName)]  
 #save the IDs in metadata:  
 ectderiv.dev@meta.data$IDs = ectderiv.dev@active.ident  
   
 library.plot.dev = DimPlot(ectderiv.dev,group.by = 'orig.ident',pt.size = 1,# order = c('mesentery','pharynx','bodywall','tentacle','phbw'),  
 cols = LibCP[5:12]  
 )+NoAxes()+labs(title = 'Time | Library origin')  
 cluster.plot.dev =  
 DimPlot(ectderiv.dev, label = F,label.size = 5, repel = T,order=(levels(ectderiv.dev@active.ident)),  
 cols = clust.cp.separate)+NoAxes()+labs(title = 'Clusters | ID')  
 cluster.plot.dev #library.plot.dev+  
   
 #generate DEG lists  
 {  
 #generate marker lists for each population (cluster)  
 ectderiv.dev@active.assay='RNA'  
 all.markers <- FindAllMarkers(ectderiv.dev,  
 logfc.threshold = 1,  
 features = ectderiv.dev@assays$RNA@var.features,  
 return.thresh = 0.001,  
 min.pct = 0.2,  
 only.pos = TRUE,   
 max.cells.per.ident = 200,  
 )  
 # add GO terms associated with this list:  
 all.markers$NVE <- 'NA'  
 all.markers$annotation\_notes <- 'NA'  
 all.markers$jgi <- 'NA'  
   
 for (i in 1:length(levels(ectderiv.dev@active.ident))) #   
 {  
 x=all.markers[as.numeric(all.markers$cluster)==i,][1:length(which(as.numeric(all.markers$cluster)==i)),7]  
 anInd = match(genes[match(x,genes$gene\_short\_name),1],annotations$NVE)  
 all.markers[as.numeric(all.markers$cluster)==i,][1:length(which(as.numeric(all.markers$cluster)==i)),8]<-annotations$NVE[anInd]  
 all.markers[as.numeric(all.markers$cluster)==i,][1:length(which(as.numeric(all.markers$cluster)==i)),9]<-annotations$annotation\_notes[anInd]  
 all.markers[as.numeric(all.markers$cluster)==i,][1:length(which(as.numeric(all.markers$cluster)==i)),10]<-annotations$HM\_ID[anInd]  
 }   
   
   
 #save your gene lists:  
 write.csv(all.markers, file='ectderiv.dev.DEGenes.csv')  
 }  
   
 library(CytoTRACE)  
 #this is memory intensive to run locally  
 cyto<-CytoTRACE(as.matrix(ectderiv.dev@assays$RNA@counts),enableFast = T,subsamplesize = 500)  
 ectderiv.dev@meta.data$cytoTRACE = cyto$CytoTRACE  
   
 ectderiv.dev <- RunUMAP(ectderiv.dev, n.neighbors = 20L,spread = 0.12,seed.use = 23,#24 is OK  
 dims = 1:38,min.dist = 0.18,#saved: d=11  
 metric = 'cosine', local.connectivity = 20)  
 DimPlot(ectderiv.dev, label = F,label.size = 5, repel = T,order=(levels(ectderiv.dev@active.ident)),  
 cols = clust.cp.separate)+NoAxes()+labs(title = 'Clusters | ID')  
   
 library(monocle3)  
 cds <- SeuratWrappers::as.cell\_data\_set(ectderiv.dev)  
 cds <- cluster\_cells(cds = cds, reduction\_method = "UMAP",  
 resolution = 0.001,k=10)  
 p=plot\_cells(cds, color\_cells\_by='partition')  
 q=plot\_cells(cds)  
 p+q  
 cds <- learn\_graph(cds, use\_partition = T,close\_loop=T)  
 plot\_cells(cds, color\_cells\_by='IDs')  
 # cds = order\_cells(cds,reduction\_method = 'UMAP')  
 cds.dev = cds   
 }  
 cluster.plot.dev  
 if(run.save)  
 {  
 save(cds.dev,file ='Robjects/monocle.ectderiv.dev.RObj')  
 save (ectderiv.dev, file='Robjects/ectderiv.dev.Robj')  
 }  
   
 #standard workflow   
 {  
 coi = colnames(ectderiv.tissue@assays$RNA@counts)  
 ectderiv.tissue <- NormalizeData(ectderiv.tissue, scale.factor = 10000)  
   
 list= NULL  
 vargenelist <- SplitObject(ectderiv.tissue, split.by = "orig.ident")  
 for (i in 1:length(vargenelist)) {  
 vargenelist[[i]] <- FindVariableFeatures(vargenelist[[i]], selection.method = "vst",  
 nfeatures = 1000, verbose = T)  
 }  
 for (i in 1:length(vargenelist)) {  
 x <- vargenelist[[i]]@assays$RNA@var.features  
 list=c(list,x)}  
 list=unique(list)  
 length(list)  
   
 ectderiv.tissue@assays$RNA@var.features = list  
 #use the full dataset scaling:  
 t=ScaleData(AllData,model.use = 'linear', use.umi = F,  
 split.by = 'orig.ident', features = ectderiv.tissue@assays$RNA@var.features)  
 ectderiv.tissue@assays$RNA@scale.data = t@assays$RNA@scale.data[,coi]  
   
 ectderiv.tissue <- RunPCA(ectderiv.tissue, pcs.compute = 50)  
 ElbowPlot(object = ectderiv.tissue, ndims = 50)  
 d=as.integer(which(ectderiv.tissue@reductions$pca@stdev>2.25))  
 ectderiv.tissue <- RunUMAP(ectderiv.tissue, n.neighbors = 30L,spread = 0.12,seed.use = 42,  
 dims = d,min.dist = 0.2,#saved: d=11  
 metric = 'cosine', local.connectivity = 1)  
   
 DimPlot(ectderiv.tissue, label = T,label.size = 4, repel = T,  
 cols = clust.cp)+NoAxes()+  
 labs(title = 'Clusters | ID')+NoLegend()  
   
 DimPlot(ectderiv.tissue,cols = clust.cp.separate,label = T)+NoAxes()+NoLegend()  
  
 ectderiv.tissue <- FindNeighbors(object = ectderiv.tissue,reduction ="pca",dims = 1:40,  
 nn.method = 'annoy', annoy.metric = 'cosine',  
 k.param = 20)  
 ectderiv.tissue <- FindClusters(object = ectderiv.tissue,resolution = 1,random.seed = 0)  
 ectderiv.tissue <- BuildClusterTree(object = ectderiv.tissue, reorder = T,   
 reorder.numeric = T, dims = c(1:20))  
 DimPlot(ectderiv.tissue, label = T,label.size = 5, repel = T,order=(levels(ectderiv.tissue@active.ident)),  
 cols = clust.cp.separate)+NoAxes()+labs(title = 'Clusters | ID')+NoLegend()  
   
 # assign cluster names:  
 neursec.clusterNames<- read\_excel("SI2\_DEG.xlsx",  
 # neursec.clusterNames<- read\_excel("SI2\_DEG.xlsx",  
 sheet = 'Fig3\_AUTO\_ANNO')  
   
 neursec.clusterNames = neursec.clusterNames[1:49,]  
 goi = neursec.clusterNames$Marker  
 #assign cluster ID to the individual libraries  
 ectderiv.tissue<-ScaleData(ectderiv.tissue,features = goi, split.by = 'orig.ident')  
 ectderiv.tissue <- SetIdent(ectderiv.tissue, value = 'tree.ident')  
 ectderiv.tissue <- BuildClusterTree(ectderiv.tissue,reorder = T,reorder.numeric = T,  
 dims= c(1:30))  
 DimPlot(ectderiv.tissue, label = T,cols = clust.cp.separate)  
 cl <-length(levels(ectderiv.tissue@active.ident))  
 C.suffix <-seq(1:cl)  
   
 g=length(goi)  
 clName = vector()  
 m=matrix(0L,g,cl)  
 for (j in 1:cl)  
 {  
 for (i in 1:g)  
 m[i,j]=mean(ectderiv.tissue@assays$RNA@scale.data[goi[i],WhichCells(ectderiv.tissue,idents = C.suffix[j])])  
 clName[j]=as.integer(which.max(m[,j]))  
 }  
  
 sort(clName)  
 neursec.clusterNames$ID[clName]  
   
 ectderiv.tissue@active.ident = factor(ectderiv.tissue@active.ident,  
 levels(ectderiv.tissue@active.ident)[order(clName)])  
 levels(ectderiv.tissue@active.ident) = neursec.clusterNames$ID[clName][order(clName)]  
 #save the IDs in metadata:  
 ectderiv.tissue@meta.data$IDs = ectderiv.tissue@active.ident  
 DimPlot(ectderiv.tissue, label = F,label.size = 5, repel = T,order=(levels(ectderiv.tissue@active.ident)),  
 cols = clust.cp.separate)+NoAxes()+labs(title = 'Clusters | ID')  
   
 library.plot.tissue = DimPlot(ectderiv.tissue,group.by = 'orig.ident',pt.size = 1,  
 cols = LibCP[5:12]  
 )+NoAxes()+labs(title = 'Time | Library origin')  
 cluster.plot.tissue =  
 DimPlot(ectderiv.tissue, label = F,label.size = 5, repel = T,order=(levels(ectderiv.tissue@active.ident)),  
 cols = clust.cp.separate)+NoAxes()+labs(title = 'Tissues | ID')  
 cluster.plot.tissue  
   
 #generate DEG lists  
 {  
 #generate marker lists for each population (cluster)  
 ectderiv.tissue@active.assay='RNA'  
 all.markers <- FindAllMarkers(ectderiv.tissue,  
 logfc.threshold = 1,  
 features = ectderiv.tissue@assays$RNA@var.features,  
 return.thresh = 0.001,  
 min.pct = 0.2,  
 only.pos = TRUE,   
 max.cells.per.ident = 200,  
 )  
 # add GO terms associated with this list:  
 all.markers$NVE <- 'NA'  
 all.markers$annotation\_notes <- 'NA'  
 all.markers$jgi <- 'NA'  
   
 for (i in 1:length(levels(ectderiv.tissue@active.ident))) #   
 {  
 x=all.markers[as.numeric(all.markers$cluster)==i,][1:length(which(as.numeric(all.markers$cluster)==i)),7]  
 anInd = match(genes[match(x,genes$gene\_short\_name),1],annotations$NVE)  
 all.markers[as.numeric(all.markers$cluster)==i,][1:length(which(as.numeric(all.markers$cluster)==i)),8]<-annotations$NVE[anInd]  
 all.markers[as.numeric(all.markers$cluster)==i,][1:length(which(as.numeric(all.markers$cluster)==i)),9]<-annotations$annotation\_notes[anInd]  
 all.markers[as.numeric(all.markers$cluster)==i,][1:length(which(as.numeric(all.markers$cluster)==i)),10]<-annotations$HM\_ID[anInd]  
 }   
   
   
 #save your gene lists:  
 write.csv(all.markers, file='ectderiv.tissue.DEGenes.csv')  
 }  
   
 library(CytoTRACE)  
 cyto<-CytoTRACE(as.matrix(ectderiv.tissue@assays$RNA@counts),enableFast = T,subsamplesize = 500)  
 ectderiv.tissue@meta.data$cytoTRACE = cyto$CytoTRACE  
   
   
 ectderiv.tissue <- RunUMAP(ectderiv.tissue, n.neighbors = 30L,spread = 0.12,seed.use = 23,  
 dims = d,min.dist = 0.18,  
 metric = 'cosine', local.connectivity = 25)  
 DimPlot(ectderiv.tissue, label = F,label.size = 5, repel = T,order=(levels(ectderiv.tissue@active.ident)),  
 cols = clust.cp.separate)+NoAxes()+labs(title = 'Clusters | ID')  
   
 library(monocle3)  
 cds <- SeuratWrappers::as.cell\_data\_set(ectderiv.tissue)  
 cds <- cluster\_cells(cds = cds, reduction\_method = "UMAP",resolution = 0.05)  
 cds <- learn\_graph(cds, use\_partition = T,close\_loop=T)  
 plot\_cells(cds, color\_cells\_by='cytoTRACE')  
 cds.tissue = cds  
 }  
 cluster.plot.tissue  
 if(run.save)  
 {  
 save (ectderiv.tissue, file='Robjects/ectderiv.tissue.Robj')   
 save(cds.tissue,file='Robjects/ectderiv.tissue.monocle.RObj')  
 }  
 }  
  
}  
   
##generate the figures:   
{  
   
#inset plots: data subsets on full data:  
 coi=NULL  
 coi[[1]]=ectderiv.gast@assays$RNA@counts@Dimnames[[2]]  
 coi[[2]]=ectderiv.dev@assays$RNA@counts@Dimnames[[2]]  
 coi[[3]]=ectderiv.tissue@assays$RNA@counts@Dimnames[[2]]  
 Fig3A.1=DimPlot(AllData,cells.highlight = coi[[1]],cols.highlight = LibCP[2])+NoAxes()+NoLegend()  
 Fig3C.1=DimPlot(AllData,cells.highlight = coi[[2]],cols.highlight = CLcp[3])+NoAxes()+NoLegend()  
 Fig3E.1=DimPlot(AllData,cells.highlight = coi[[3]],cols.highlight = LibCP[15])+NoAxes()+NoLegend()  
   
 Fig3A.1+Fig3C.1+Fig3E.1+plot\_layout(ncol=3)  
  
 #monocle plots : Fig.3A,C,E  
 {  
 library(monocle3)  
 g=plot\_cells(cds.gast,   
 color\_cells\_by = 'cytoTRACE',  
 label\_cell\_groups=F,   
 label\_leaves=F,  
 label\_branch\_points=F,  
 label\_roots = F,  
 trajectory\_graph\_color = 'cyan',  
 trajectory\_graph\_segment\_size = 1,  
 cell\_size = 3,  
 cell\_stroke = 0,  
 )+scale\_color\_viridis(option = 'A',discrete = F,direction = -1)  
   
 d=plot\_cells(cds.dev,   
 color\_cells\_by = 'cytoTRACE',  
 label\_cell\_groups=F,   
 label\_leaves=F,  
 label\_branch\_points=F,  
 label\_roots = F,  
 trajectory\_graph\_color = 'cyan',  
 trajectory\_graph\_segment\_size = 1,  
 cell\_size = 2,  
 cell\_stroke = 0,  
 )+scale\_color\_viridis(option = 'A',discrete = F,direction = -1)  
   
 t=  
 plot\_cells(cds.tissue,   
 color\_cells\_by = 'cytoTRACE',  
 label\_cell\_groups=T,   
 label\_leaves=F,  
 label\_branch\_points=F,  
 label\_roots = F,  
 trajectory\_graph\_color = 'cyan',  
 trajectory\_graph\_segment\_size = 2,  
 cell\_size = 3,  
 cell\_stroke = 0)+scale\_color\_viridis(option = 'A',discrete = F,direction = -1)  
 g+d+t+plot\_layout(ncol=3)  
 }  
  
#generate various gene lists and Rmagic-impute the data for visualization:  
 markers = unique(c('INSM1-like','NvPOU4','ID4-like2',  
 "NvKlf-spotty","NvNanos1",'NvAshA',   
 'PRD14-like3','NvFoxQ2d',  
 'NvTLL-like','NvNnNot2-like',"NvAshC","NvNem64",  
 'NvNscl2-like'))  
   
 goi = c('PCNA-like','CDN1A-like',"NSE2-like",'NvHes3','Nvmyc1','NvSox3',  
 'NvMyc3','NvSoxC','NvNeurogenin1','NvSoxB.2',markers)  
 goi2= c('NOT2-like2',  
 'NvAshA','NvHoxA-Anthox6','GBGE-like','ZC4H2-like',  
 "CD151-like3","NvCellulase", "SEGN-like3",'PCNA-like','CDN1A-like')  
 g.mag=unique(c(markers,goi,goi2))  
 ectderiv.gast <- magic(ectderiv.gast, genes=g.mag)  
 ectderiv.dev <- magic(ectderiv.dev, genes=g.mag)  
 ectderiv.tissue <- magic(ectderiv.tissue, genes=g.mag)  
  
 #gastrula gene expression plots:  
 {  
 ectderiv.gast@active.assay='MAGIC\_RNA'  
 gene.plots.g = FeaturePlot(ectderiv.gast,c(goi[c(1:12)],'NOT2-like2',  
 'NvAshA','NvHoxA-Anthox6','GBGE-like','ZC4H2-like',  
 "CD151-like3","NvCellulase", "SEGN-like3"),#,'NvPrdl-d'[c(1:17,25,18:24)],#[c(2:4,6,8,15:20,)],  
 order = T,   
 ncol=5,raster = F,  
 cols = gene.cp)&FontSize(main=8)&NoLegend()&NoAxes()  
   
 gene.plots.g  
 }  
 #post-gast.dev gene expression plots:  
 {  
 ectderiv.dev@active.assay='MAGIC\_RNA'  
 gene.plots.d = FeaturePlot(ectderiv.dev,c('PCNA-like','CDN1A-like',goi[3:20]),#  
 order = T,   
 ncol=5,raster = F,  
 cols = gene.cp)&FontSize(main=8)&NoLegend()&NoAxes()  
 gene.plots.d  
   
 }  
 #tissue gene expression plots:  
 {  
 ectderiv.tissue@active.assay='MAGIC\_RNA'  
 gene.plots.t = FeaturePlot(ectderiv.tissue,  
 c(goi[c(1:20)]),  
 order = T,   
 ncol=5,raster = F,  
 cols = gene.cp)&FontSize(main=8)&NoLegend()&NoAxes()  
 gene.plots.t  
   
 }  
  
 }  
}