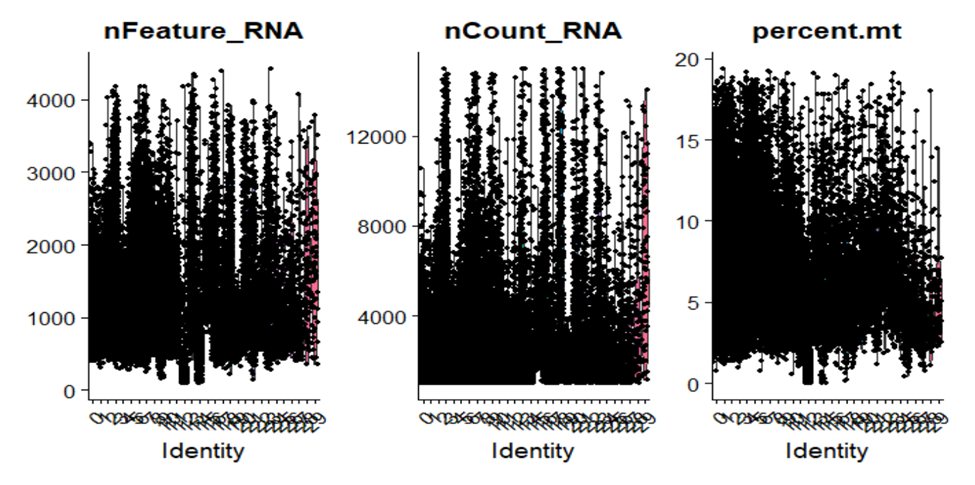
Pareto Archetype Analysis applied to single cell gene expression helps to observe extreme points in multi-dimensional data where archetypes may represent specific cell specialization or the optimal trade off within cell tasks. Performing a gene set enrichment analysis at different archetypes highlights the key biological process of that archetype, thus identifying an optimal biological task within similar cell types. Here, I apply a Pareto task inference on peripheral blood mononuclear cells using gene expression meta-data from the Covid-19 cell atlas of infected Covid-19 patients.I want to identify key biological processes and the primary cell type that indicates Sars-Cov-2’s ability to damage blood vessels, cause blood clotting, and travel in the blood stream. I use R’s ParetoTI package written by Vitalii Kleshchevnikov and perform a GSEA using Python’s gseapy package.

library(ParetoTI)

memory.limit(30000)  
library(Seurat)  
pbmcs <- readRDS("C:\\Users\\zulfi\\Desktop\\blish\_covid.seu.rds")  
DefaultAssay(pbmcs) <- "RNA"  
print(pbmcs)

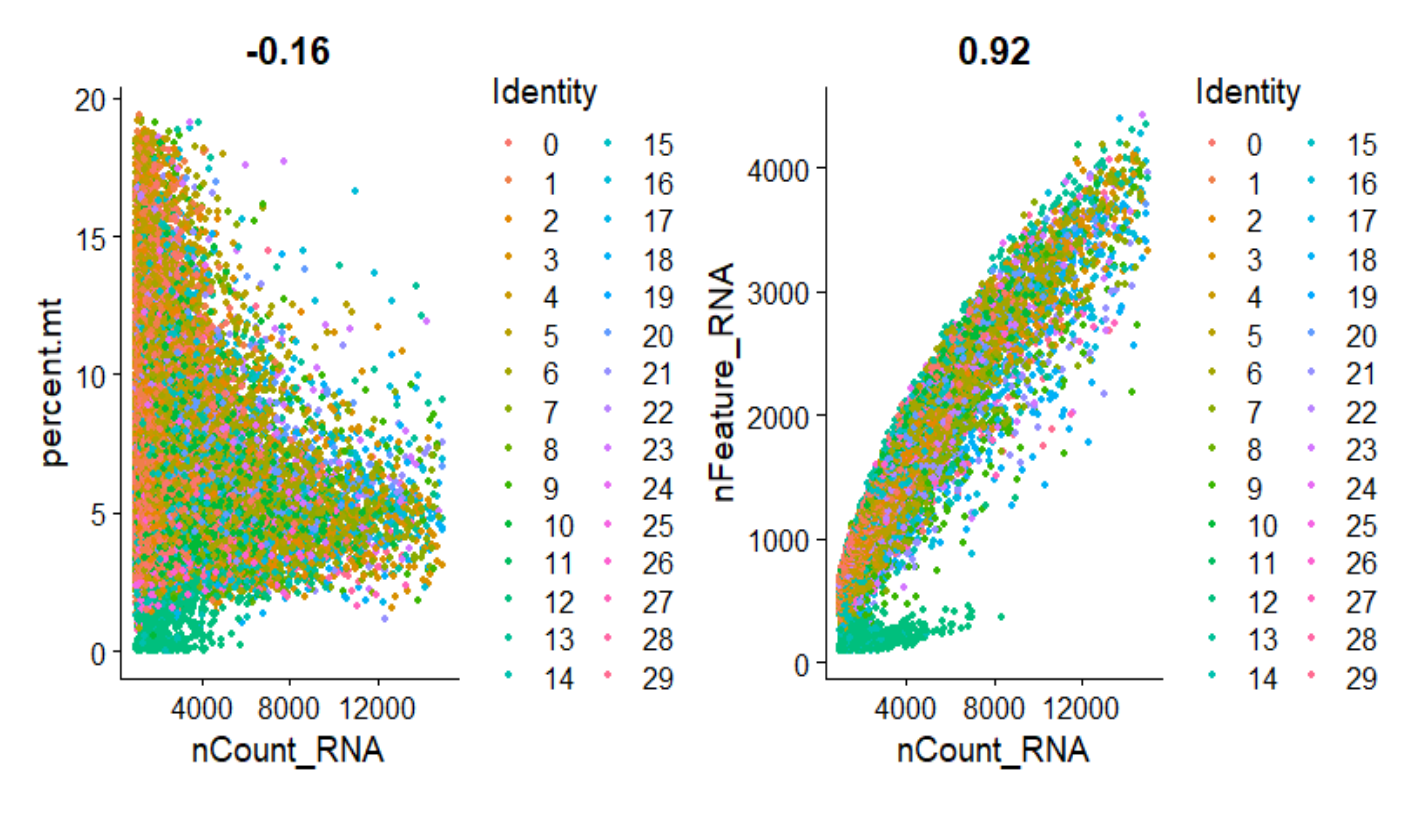
Violin Plots help show where most of the data is concentrated at. This plot was shown as a way to see how I could subset my data.

VlnPlot(pbmcs,features = c("nFeature\_RNA","nCount\_RNA","percent.mt"),ncol = 3)



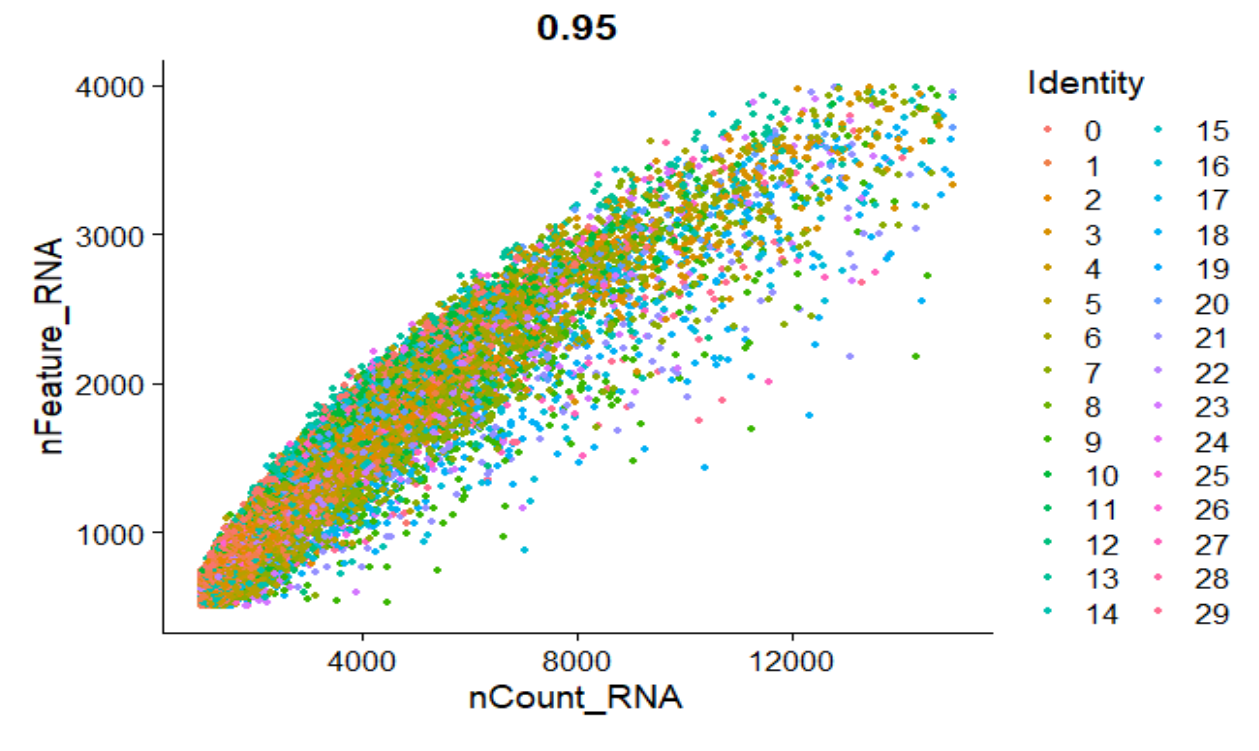
We can also show the relationship of the count of RNA molecules by the count of genes to subset for genes that are strongly correlated with RNA molecules. Mitochondria contamination is also observed with respect to the RNA molecule count.

plot1 <- FeatureScatter(pbmcs,feature1 = "nCount\_RNA",feature2 = "percent.mt")  
plot2 <- FeatureScatter(pbmcs, feature1 = "nCount\_RNA", feature2 = "nFeature\_RNA")  
plot1 + plot2



I maybe including a lot of mitochondrial contamination here, but the bulk of the cells are around 2.5 percent to 10 percent. We can get a 0.95 correlation of molecule count to gene counts if we filter with percent.mt < 10 and percent.mt > 2.5. But that would still be including a lot of contamination. However, if we remove all percent.mt or even stick with percent.mt below 2.5, then we won’t have much biological signal. To work around this, I will regress out the mitochondrial contamination when I scale my data in the later steps.

memory.limit(30000)  
# subset cells according to the above plots  
pbmcs <- subset(pbmcs, subset = nFeature\_RNA < 4000 & nFeature\_RNA > 500 & percent.mt < 10 & percent.mt > 2.5)  
plot3 <- FeatureScatter(pbmcs, feature1 = "nCount\_RNA", feature2 = "nFeature\_RNA")  
print(plot3)

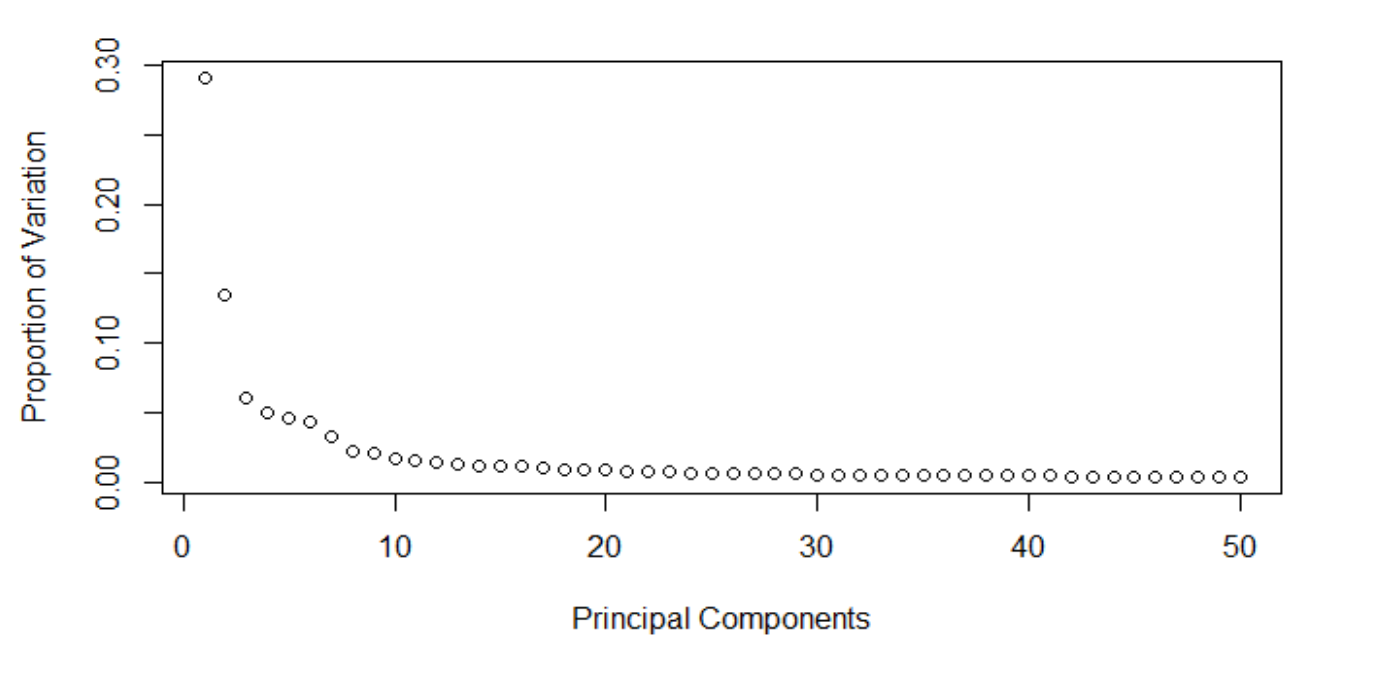


These steps correspond to standard single cell processing for clustering, and although I’m not clustering, I would still like to reduce dimensionality when observing the archetypes and will only use highly variable features for the rest of the analysis. This will also help in implementing the principal convex hull algorithm used for the Pareto inferencence later on.

memory.limit(30000)  
# use log normalization  
pbmcs <- NormalizeData(pbmcs,normalization.method = "LogNormalize",scale.factor = 10000, assay = "RNA")  
# use a variance stabilization transformation; transform skewness to make it more normal  
pbmcs <- FindVariableFeatures(pbmcs, selection.method = "vst", nfeatures = 2000,assay = "RNA")  
# regress out the mitochondrial contamination and use only variable features(default)  
pbmcs <- ScaleData(pbmcs,assay = "RNA",vars.to.regress = "percent.mt")  
pbmcs <- RunPCA(pbmcs, features = VariableFeatures(object = pbmcs),assay = "RNA",npcs = 50)  
print(pbmcs)

I will only use the principal components that contribute to the majority of variation. In this case, I use 10.

# Find how many PCs to use by observing drop in proportion of variation  
eigs <- pbmcs@reductions$pca@stdev\*\*2  
props <- eigs/sum(eigs)  
plot(props,ylab = "Proportion of Variation",xlab="Principal Components")

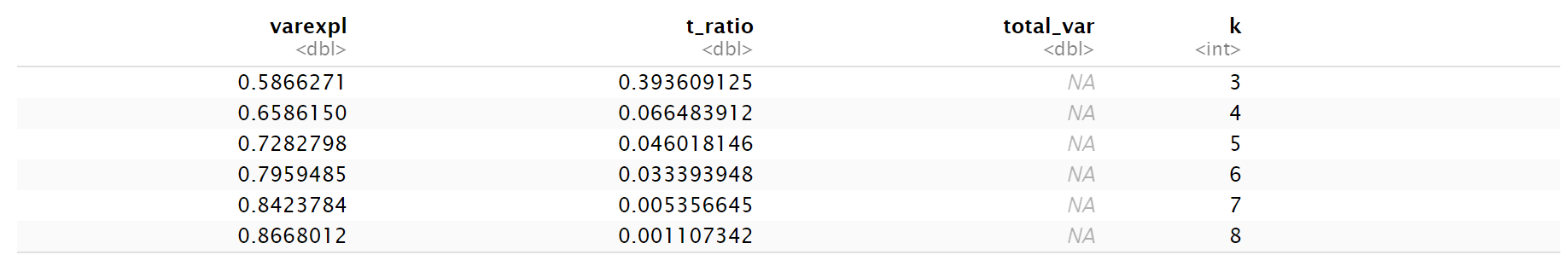


Here, I run a principal convex hull from 3 archetypes through 8.I want to see which archetype is the most statistically significant. Handled by k\_fit\_pch function.

memory.limit(30000)  
# get pca cell embeddings  
pca <- data.frame(pbmcs@reductions$pca@cell.embeddings)  
pca <- pca[c(1:10)]  
pca <- data.matrix(pca)  
pca <- t(pca)  
pcha <- k\_fit\_pch(pca,ks = 3:8,maxiter = 500,seed=345)

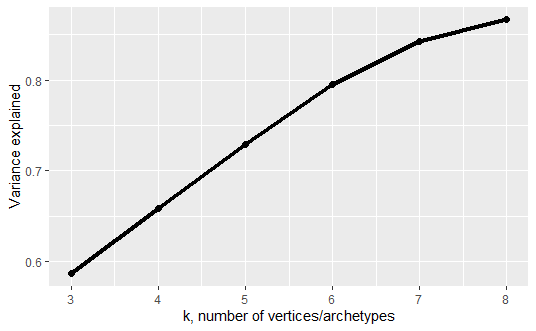
The summary statistics shows that 8 archetypes will display the most variance explained. Although 3 archetypes are more statistically significant than the rest of the archetypes, I find that using 3-5 archetypes will have the majority of biological processes relating to the innate and adaptive immune responses which makes sense as these processes would be more statistically significant in a viral disease. To find an archetype that may indicate blood vessel damage and invasive properties,I’ll use all 8 archetypes.

print(pcha$summary)



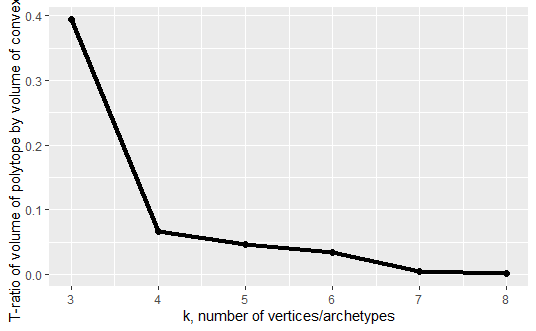
8 archetypes do contribute to the most variation and may allow me to observe genes and mechanisms behind blood vessel spread.

plot\_arc\_var(pcha, type = "varexpl", point\_size = 2)



A t-ratio plot shows me which archetype is the most statistically significant, however, I will still use 8 arcs to yield highly specific archetype tasks.

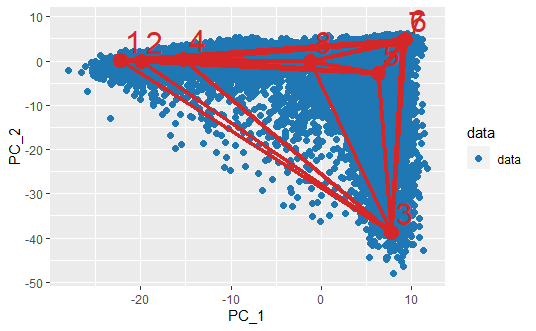
plot\_arc\_var(pcha, type = "t\_ratio", point\_size = 2)



Apply PCH with 8 archetypes. Although the plot is difficult to observe, we see a polytope with 8 archeytypes using the first two principal components.

memory.limit(30000)  
arc <- fit\_pch(pca,noc= 8,conv\_crit = 1e-06,maxiter = 500,volume\_ratio = "t\_ratio")

plt\_arc <- plot\_arc(arc\_data = arc, data = pca)  
print(plt\_arc)



I use the euclidean distance as the metric for the distance matrix from genes to arches, and find features that are a decreasing function of distance from archetype by finding features with the highest bin closest to the archetypes using a wilcoxon 1 vs all test(vitkl). Here, I use 10 percent of the closest features to the archetype. BioQC refers to the method(software tool/package) used to solve the wilcox test. I get an output with the closest, enriched genes at the archetypes, and I’ll perform a GSEA on those enriched genes using Python’s gseapy.

#function to merge data into a data object needed for finding enriched genes  
mergeDat <- merge\_arch\_dist(arc\_data = arc,data = pca,dist\_metric = c("euclidean", "arch\_weights")[1],feature\_data = pbmcs@assays$RNA@scale.data)  
  
  
# function to find features by using a decreasing function to get closest genes  
enrGenes <- find\_decreasing\_wilcox(mergeDat$data,mergeDat$arc\_col,features = mergeDat$features\_col,bin = 0.1,method = "BioQC")  
  
 #use the wilcoxon-p-val for a cutoff and adjust using Benjamin-Hochberg  
genesAtArcs <- get\_top\_decreasing(summary\_genes = enrGenes,summary\_sets = NULL, cutoff\_genes = 0.001,cutoff\_metric = "wilcoxon\_p\_val",p.adjust.method = "BH")

-- archetype\_3

IGJ, MZB1, ELL2, MKI67

ITM2C, IGHG4, IGHG3, IGHG1

IGHG2, IGLC2, HSP90B1, FAM46C

-- archetype\_5

MS4A1, BANK1, CD79A, PAX5

CD22, RALGPS2, FCRL1, FAM129C

IGHD, CD79B, LINC00926, IGHM

-- archetype\_1

PLBD1, CLU, S100A12, VCAN

S100A8, S100A9, CD14, MNDA

DYSF, GRN, FPR1, RNASE2

-- archetype\_4

CSF1R, MS4A7, AIF1, LST1

FGL2, SERPINA1, SAT1, PSAP

CDKN1C, LILRB2, CTSS, MAFB

-- archetype\_6

PRF1, GNLY, FGFBP2, SPON2

GZMB, FCGR3A, CD247, GPR56

NKG7, IL2RB, S1PR5, SYNE1

-- archetype\_2

FCN1, FOS, LYZ, VCAN

TNFAIP2, CD14, TGFBI, AIF1

KLF4, CPVL, MS4A6A, RGS2

-- archetype\_7

TCF7, LEF1, IL7R, CCR7

MAL, LTB, TRABD2A, ITGA6

CAMK4, ABLIM1, MYC, TRAT1

-- archetype\_8

SDPR, TUBB1, PPBP, ITGB3

PF4, SPARC, GNG11, ITGA2B

PRKAR2B, C2orf88, HIST1H2AC, NRGN

I want to observe the cell types at archetypes that I’m interested in. I find:

**Archetype 7 : CD4 M and CD4 N T Cells**

**Archetype 8 : Platelets**

**Archetype 6: Natural Killer Cells**

**Archetype 4: CD14 and CD16 Monocytes**

**Code below shows how we can find corresponding cell types.**

# Write data table to csv for later GSEA  
dt <- fwrite(x = genesAtArcs$enriched\_genes,file="enrGenes.csv")

#Archetype 7,8,6,and 4  
nearCells = bin\_cells\_by\_arch(mergeDat$data,mergeDat$arc\_col,bin\_prop = 0.1,return\_names = TRUE,dist\_cutoff = 0.01)

arc7Cells <- data.frame(nearCells)  
cellTypes <- data.frame(pbmcs$cell.type)  
cellTypes <- data.frame(cells = rownames(cellTypes),cellTypes,row.names = NULL)  
arc7CellTypes <- subset(cellTypes,cells %in% nearCells$archetype\_7)  
print(arc7CellTypes)

arc8Cells <- data.frame(nearCells)  
cellTypes <- data.frame(pbmcs$cell.type)  
cellTypes <- data.frame(cells = rownames(cellTypes),cellTypes,row.names = NULL)  
arc8CellTypes <- subset(cellTypes,cells %in% nearCells$archetype\_8)  
print(arc8CellTypes)

arc6Cells <- data.frame(nearCells)  
cellTypes <- data.frame(pbmcs$cell.type)  
cellTypes <- data.frame(cells = rownames(cellTypes),cellTypes,row.names = NULL)  
arc6CellTypes <- subset(cellTypes,cells %in% nearCells$archetype\_6)  
print(arc6CellTypes)

arc4Cells <- data.frame(nearCells)  
cellTypes <- data.frame(pbmcs$cell.type)  
cellTypes <- data.frame(cells = rownames(cellTypes),cellTypes,row.names = NULL)  
arc4CellTypes <- subset(cellTypes,cells %in% nearCells$archetype\_4)  
print(arc4CellTypes)

Idents(object = pbmcs) <- "cell.type"  
indCells <- subset(pbmcs,idents = c("Platelet"))

memory.limit(30000)  
indCellsPCS <- data.frame(indCells@reductions$pca@cell.embeddings)  
indCellsPCS <- indCellsPCS[c(1:10)]  
indCellsPCS <- data.matrix(indCellsPCS)  
indCellsPCS <- t(indCellsPCS)

indCellPcha <- k\_fit\_pch(indCellsPCS,ks = 3:8,maxiter = 500,seed=345)  
print(indCellPcha$summary)

Above, I only want to observe if there are any unique cell tasks among Platelets only(indCells). I will then perform a gene set enrichment analysis just with Platelet archetypes.

I will use 3 archetypes to see if I can distinguish tasks among Platelet cells.

plateletPCHA <- fit\_pch(indCellsPCS,noc = 3,conv\_crit= 1e-06,maxiter = 500,volume\_ratio = "t\_ratio")  
plt\_arc <- plot\_arc(arc\_data = plateletPCHA, data = indCellsPCS)  
print(plt\_arc)

obj <- merge\_arch\_dist(arc\_data = plateletPCHA,data = indCellsPCS,dist\_metric = c("euclidean", "arch\_weights")[1],feature\_data = indCells@assays$RNA@scale.data)  
  
plateletGenes <- find\_decreasing\_wilcox(obj$data,obj$arc\_col,features = obj$features\_col,  
 bin = 0.1,method = "BioQC")

plateletStats <- get\_top\_decreasing(summary\_genes = arcEnrGenes,summary\_sets = NULL, cutoff\_genes = 0.05,cutoff\_metric = "wilcoxon\_p\_val",p.adjust.method = "BH")  
print(CD4Stats$enriched\_genes)

dt2 <- fwrite(x = CD4Stats$enriched\_genes,file="Platelets.csv")

-- archetype\_1

NELL2, NUCB2, RP11-291B21.2, NME1

AMICA1, HNRNPAB, BACH1, HEMGN

CLPTM1L, HERC5, JUN

-- archetype\_3

MAF, LIMS1

-- archetype\_2

IL6ST, ITGA6, SP140