Gene graph with elife asexuals, all genes and feature slxn with scmap

```
setwd("/Users/vh3/Documents/PfMCA/ANALYSIS_2/")
library(scater)
library(pheatmap)
library(viridis)
library(scran)
library(Seurat)
library(M3Drop)
library(RColorBrewer)
library(dplyr)
library(plotly)
```

all atlas data with elife 2018 data added back in, remove gams from 2018 paper and 15 min fb spz

```
mca.qc <- readRDS("/Users/vh3/Documents/PfMCA/ANALYSIS_2/pf.mca.plus.elife_20200527.rds")

mca.qc$stage_yr <- paste(mca.qc$stage, mca.qc$year, sep = "_")

mca.qc <- mca.qc[, mca.qc$stage_yr != "gam_2016"]

mca.qc[, which(is.na(mca.qc$time))]$time <- "0min"
mca.qc <- mca.qc[, mca.qc$time != "15min"]

ookclusts <- read.csv("ook_suerat_clusters_20200625.csv", row.names = 1)
ookclusts$xfilename <- rownames(ookclusts)
clust2 <- ookclusts[ookclusts$seurat_clusters == 2, ]

mca.qc2 <- mca.qc[, !(mca.qc$xfilename %in% clust2$xfilename)]

mca.qc.counts <- as.data.frame(counts(mca.qc2))

# write.csv(mca.qc.counts, file='mca.qc.counts.elifeasex.subook.csv')
```

convert to seurat to cluster all cells and subset 100 cells from the large spz clusters

```
mca.qc.seurat <- as.Seurat(mca.qc2, counts = "counts", data = "logcounts")
```

```
## Warning: Feature names cannot have underscores ('_'), replacing with dashes
## ('-')
```

```
## Warning: Feature names cannot have underscores ('_'), replacing with dashes
## ('-')
```

```
mca.qc.seurat <- FindVariableFeatures(mca.qc.seurat, selection.method = "vst", nfeatures = 2000)
all.genes <- rownames(mca.qc.seurat)
mca.qc.seurat <- ScaleData(mca.qc.seurat, features = all.genes)
```

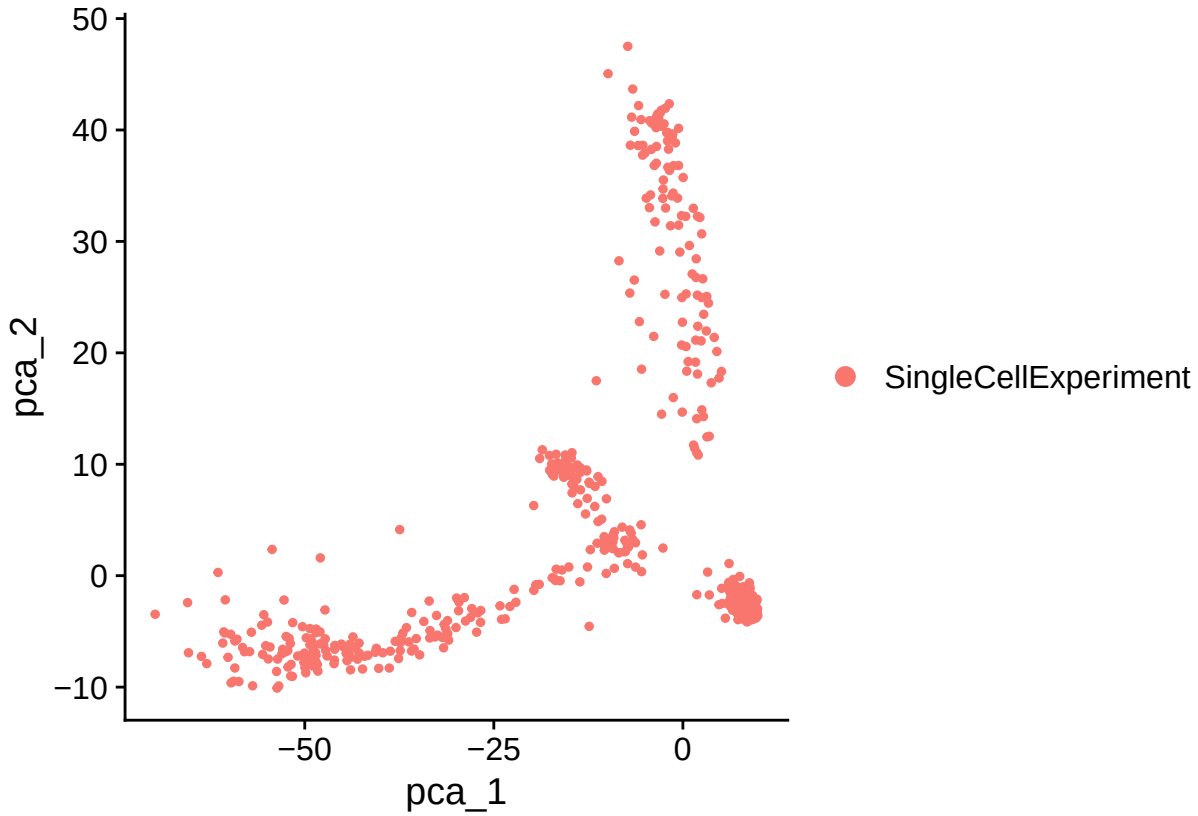
```
## Centering and scaling data matrix
```

```
mca.qc.seurat <- RunPCA(mca.qc.seurat, features = VariableFeatures(object = mca.qc.seurat))
```

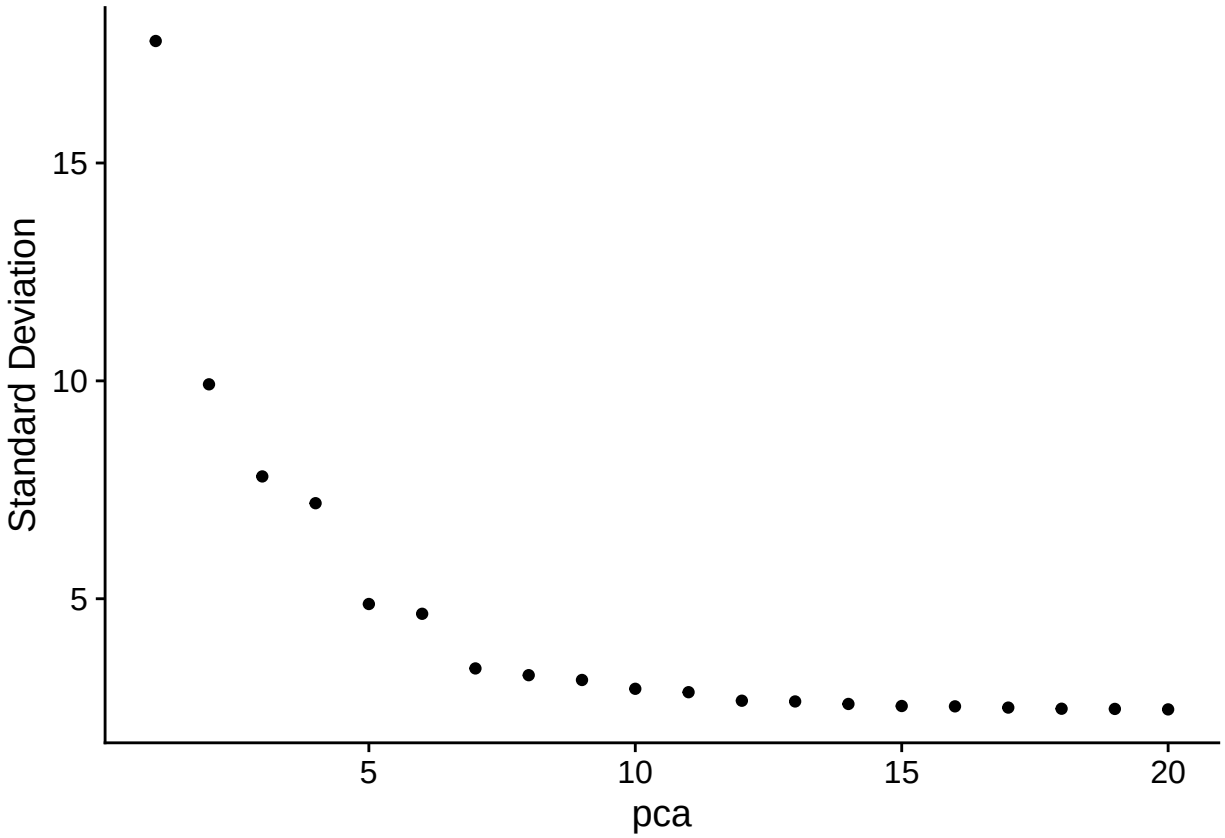
```
## PC_ 1
## Positive:  PF3D7-0304700.1, PF3D7-1201300.1, PF3D7-0207300.1, PF3D7-0718300.1, PF3D7-1442600.1, PF3D
##     PF3D7-0818100.1, PF3D7-1132100.1, PF3D7-1218100.1, PF3D7-0728700.1, PF3D7-0319100.1, PF3D7-07123
##     PF3D7-0624800.1, PF3D7-1352900.1, PF3D7-0610700.1, PF3D7-1213400.1, PF3D7-1350900.1, PF3D7-09188
## Negative:  PF3D7-1038000.1, PF3D7-0316300.1, PF3D7-1004200.1, PF3D7-1001600.1, PF3D7-1352500.1, PF3D
##     PF3D7-0619400.1, PF3D7-1466300.1, PF3D7-1222300.1, PF3D7-0529400.1, PF3D7-0918000.1, PF3D7-09261
##     PF3D7-1426000.1, PF3D7-0207700.1, PF3D7-1138400.1, PF3D7-0207500.1, PF3D7-1457000.1, PF3D7-11156
## PC_ 2
## Positive:  PF3D7-1404300.1, PF3D7-0508400.1, PF3D7-1312450.1, PF3D7-1145000.1, PF3D7-1434900.1, PF3D
##     PF3D7-1309800.1, PF3D7-0320400.1, PF3D7-1137200.1, PF3D7-0407200.1, PF3D7-1144800.1, PF3D7-09293
##     PF3D7-0819400.1, PF3D7-1248400.1, PF3D7-1334000.1, PF3D7-1106600.1, PF3D7-0607900.1, PF3D7-08235
## Negative:  PF3D7-0907200.1, PF3D7-0207600.1, PF3D7-1001600.1, PF3D7-0316300.1, PF3D7-0207700.1, PF3D
##     PF3D7-1441100.1, PF3D7-0207800.1, PF3D7-0202400.1, PF3D7-1343000.1, PF3D7-1426000.1, PF3D7-11365
##     PF3D7-0529400.1, PF3D7-0202000.1, PF3D7-1335100.1, PF3D7-1104400.1, PF3D7-0917900.1, PF3D7-07316
## PC_ 3
## Positive:  PF3D7-1016300.1, PF3D7-0532400.1, PF3D7-0220000.1, PF3D7-1347200.1, PF3D7-0532100.1, PF3D
##     PF3D7-0731300.1, PF3D7-0731600.1, PF3D7-0721100.1, PF3D7-1341200.1, PF3D7-0500800.1, PF3D7-07088
##     PF3D7-0210400.1, PF3D7-1460400.1, PF3D7-0220600.1, PF3D7-1305300.1, PF3D7-0813900.1, PF3D7-14412
## Negative:  PF3D7-0501500.1, PF3D7-0501600.1, PF3D7-1410400.1, PF3D7-1012200.1, PF3D7-1463900.1, PF3D
##     PF3D7-0404700.1, PF3D7-0722200.1, PF3D7-1476300.1, PF3D7-1401600.1, PF3D7-0817700.1, PF3D7-06211
##     PF3D7-0620400.1, PF3D7-1323700.1, PF3D7-1436200.1, PF3D7-1140400.1, PF3D7-1009700.1, PF3D7-06180
## PC_ 4
## Positive:  PF3D7-1330400.1, PF3D7-1338700.1, PF3D7-0912900.1, PF3D7-1124300.1, PF3D7-0308100.1, PF3D
##     PF3D7-1367800.1, PF3D7-1312450.1, PF3D7-0807900.1, PF3D7-0620000.1, PF3D7-1434200.1, PF3D7-07254
##     PF3D7-0934800.1, PF3D7-0625900.1, PF3D7-0207600.1, PF3D7-1115600.1, PF3D7-0613300.1, PF3D7-14137
## Negative:  PF3D7-1103500.1, PF3D7-1316700.1, PF3D7-1361300.1, PF3D7-0825700.1, PF3D7-1356000.1, PF3D
##     PF3D7-1454800.1, PF3D7-1455300.1, PF3D7-1201600.1, PF3D7-1207700.1, PF3D7-1471800.1, PF3D7-07229
##     PF3D7-1438500.1, PF3D7-0205100.1, PF3D7-1362700.1, PF3D7-0304100.1, PF3D7-1141900.1, PF3D7-13021
## PC_ 5
## Positive:  PF3D7-0905300.1, PF3D7-1311100.1, PF3D7-1112900.1, PF3D7-1207800.1, PF3D7-1440600.1, PF3D
##     PF3D7-0113100.1, PF3D7-1303900.1, PF3D7-1413000.1, PF3D7-0214300.1, PF3D7-0521100.1, PF3D7-13434
##     PF3D7-1475700.1, PF3D7-0520100.1, PF3D7-1473300.1, PF3D7-1014200.1, PF3D7-1361500.1, PF3D7-13556
## Negative:  PF3D7-1361300.1, PF3D7-0825800.1, PF3D7-1327100.1, PF3D7-1356000.1, PF3D7-0825700.1, PF3D
##     PF3D7-0722900.1, PF3D7-1471800.1, PF3D7-1034200.1, PF3D7-1207700.1, PF3D7-0621400.1, PF3D7-14518
##     PF3D7-1331600.1, PF3D7-0908300.1, PF3D7-1475500.1, PF3D7-1222300.1, PF3D7-1333500.1, PF3D7-07296
```

```
## Warning: Cannot add objects with duplicate keys (offending key: PC_), setting
## key to 'pca_'
```

```
DimPlot(mca.qc.seurat, reduction = "pca")
```




```
# JackStrawPlot(mca.qc.seurat, dims = 1:10)
ElbowPlot(mca.qc.seurat)
```




```
mca.qc.seurat <- FindNeighbors(mca.qc.seurat, dims = 1:10)
```

```
## Computing nearest neighbor graph
```

```
## Computing SNN
```

```
mca.qc.seurat <- FindClusters(mca.qc.seurat, resolution = 0.5)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 1353
## Number of edges: 43934
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8355
## Number of communities: 8
## Elapsed time: 0 seconds
```

```
mca.qc.seurat <- RunUMAP(mca.qc.seurat, dims = 1:10)
```

```
## Warning: The default method for RunUMAP has changed from calling Python UMAP via reticulate to the R
## To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'correlation'
## This message will be shown once per session
```

```
## 16:20:10 UMAP embedding parameters a = 0.9922 b = 1.112
```

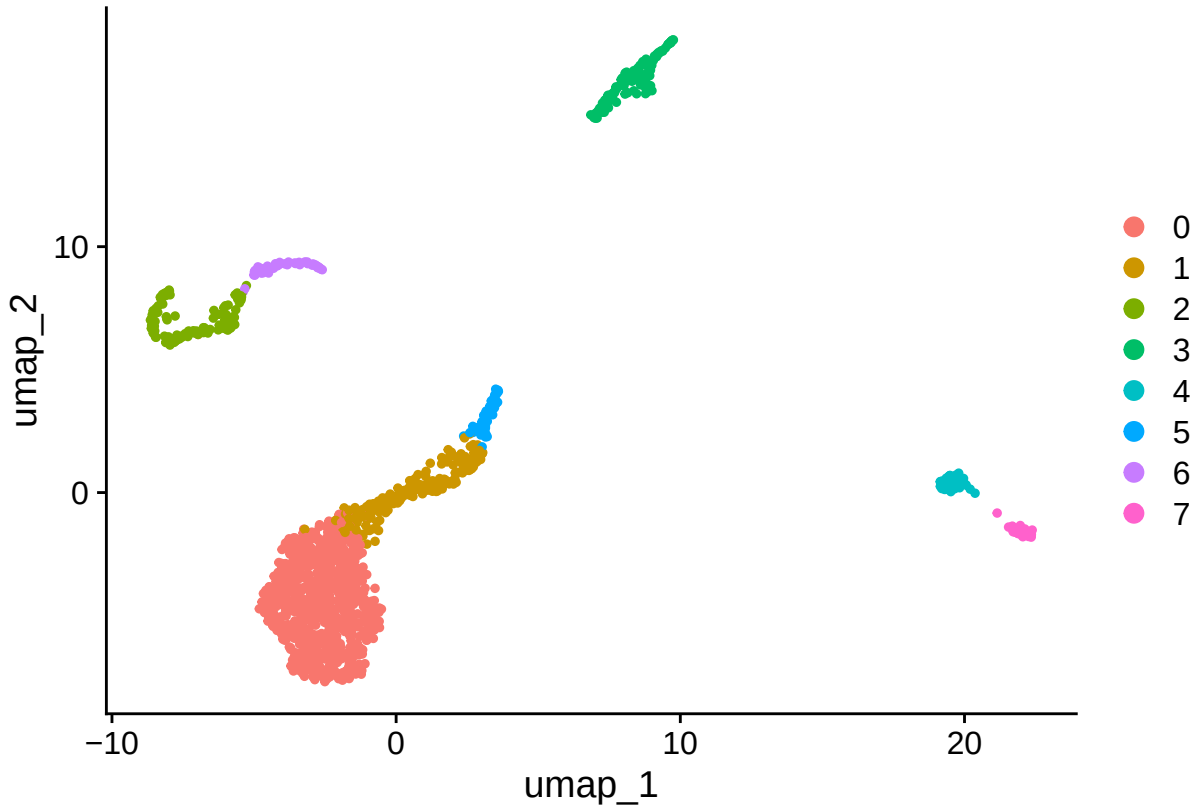
```
## 16:20:10 Read 1353 rows and found 10 numeric columns
```

```
## 16:20:10 Using Annoy for neighbor search, n_neighbors = 30
```

```
## 16:20:10 Building Annoy index with metric = cosine, n_trees = 50
```

```
## 0%   10   20   30   40   50   60   70   80   90   100%
## [----|----|----|----|----|----|----|----|----|----|
## **************************************************|
## 16:20:10 Writing NN index file to temp file /var/folders/jg/ylpkqzys38lfzggn01b_krqw000g9y/T//RtmpZ4
## 16:20:10 Searching Annoy index using 1 thread, search_k = 3000
## 16:20:10 Annoy recall = 100%
## 16:20:11 Commencing smooth kNN distance calibration using 1 thread
## 16:20:12 Initializing from normalized Laplacian + noise
## 16:20:12 Commencing optimization for 500 epochs, with 56188 positive edges
## 16:20:14 Optimization finished
## Warning: Cannot add objects with duplicate keys (offending key: UMAP_), setting
## key to 'umap_'
```

```
DimPlot(mca.qc.seurat, reduction = "umap")
```




```
table(Idents(mca.qc.seurat))
```

```
##
##   0   1   2   3   4   5   6   7
## 743 187 120 112  54  53  50  34
```

```
table(Idents(mca.qc.seurat), mca.qc.seurat@meta.data$stage)
```

```
##
##     asex bbSpz fbSpz gam hlSpz ook ooSpz sgSpz
##   0    1   161   223   0   110   0     5   243
##   1    0     2     3   0    67   0   101    14
##   2  120     0     0   0     0   0     0     0
```

```
##   3    0     0     0   0     0 112     0     0
##   4    0     0     0  54     0   0     0     0
##   5    0     0     0   0     1   0    52     0
##   6   50     0     0   0     0   0     0     0
##   7    0     0     0  34     0   0     0     0
```

```
table(mca.qc.seurat@meta.data$seurat_clusters)
```

```
##
##   0   1   2   3   4   5   6   7
## 743 187 120 112  54  53  50  34
```

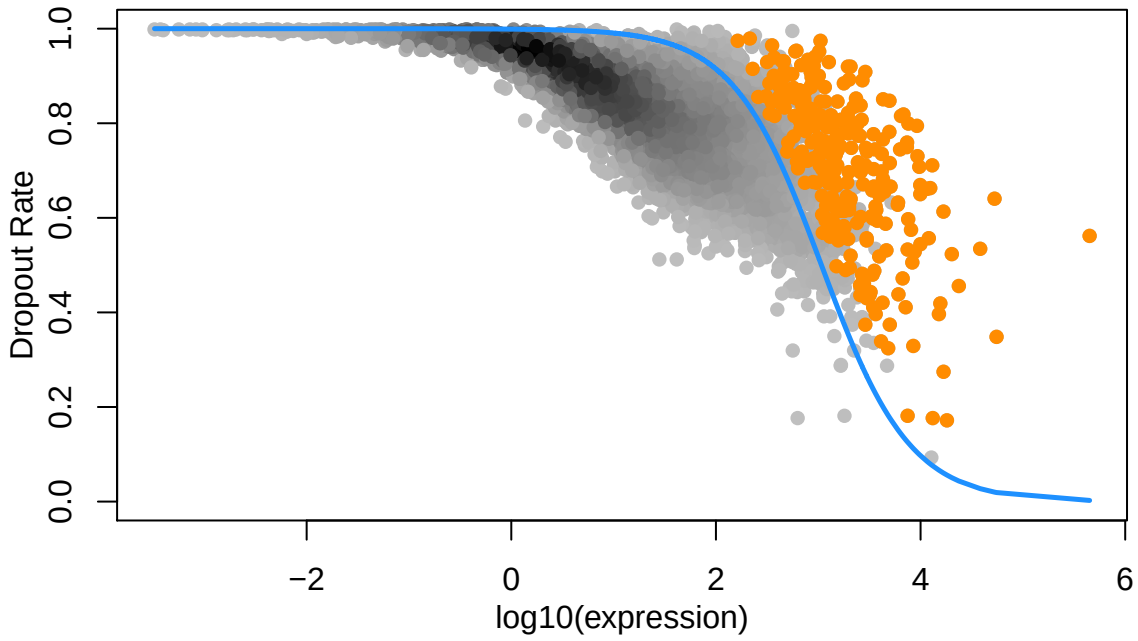
```
clusts <- mca.qc.seurat@meta.data["seurat_clusters"]
write.csv(clusts, file = "Seurat_clusters_forgenegraph.subook.elilfeasex_20200625.csv")

smd <- as.data.frame(mca.qc.seurat@meta.data)

spz <- c(0, 1)
subsmd <- smd[smd$seurat_clusters %in% spz, ]

subsub <- subsmd %>% group_by(seurat_clusters) %>% sample_n(size = 100)

keepers <- subsub$xfilename
# write.csv(keepers, file='subsampledspz_20200616.csv')
other <- c(2, 3, 4, 5, 6, 7)
othersub <- smd[smd$seurat_clusters %in% other, ]

keepers2 <- othersub$xfilename

all <- c(keepers, keepers2)

mca.qc2$seurat_clusters <- smd$seurat_clusters
# saveRDS(mca.qc2, file='pfmca.withelifeasex.subook_20200626.rds')
mca.qc.sub <- mca.qc2[, colnames(mca.qc2) %in% all]

# write.csv(as.data.frame(counts(mca.qc.sub)),
# file='submca.qc.counts.elifeasex.subook_forgenegraph.csv')
```

look into feature selection, stick with scmap top 2000 features

```
norm <- assay(mca.qc.sub, "normcounts")
M3Drop_genes <- M3DropFeatureSelection(norm, mt_method = "fdr", mt_threshold = 0.5)
```

```
## Warning in bg__calc_variables(expr_mat): Warning: Removing 24 undetected genes.
```

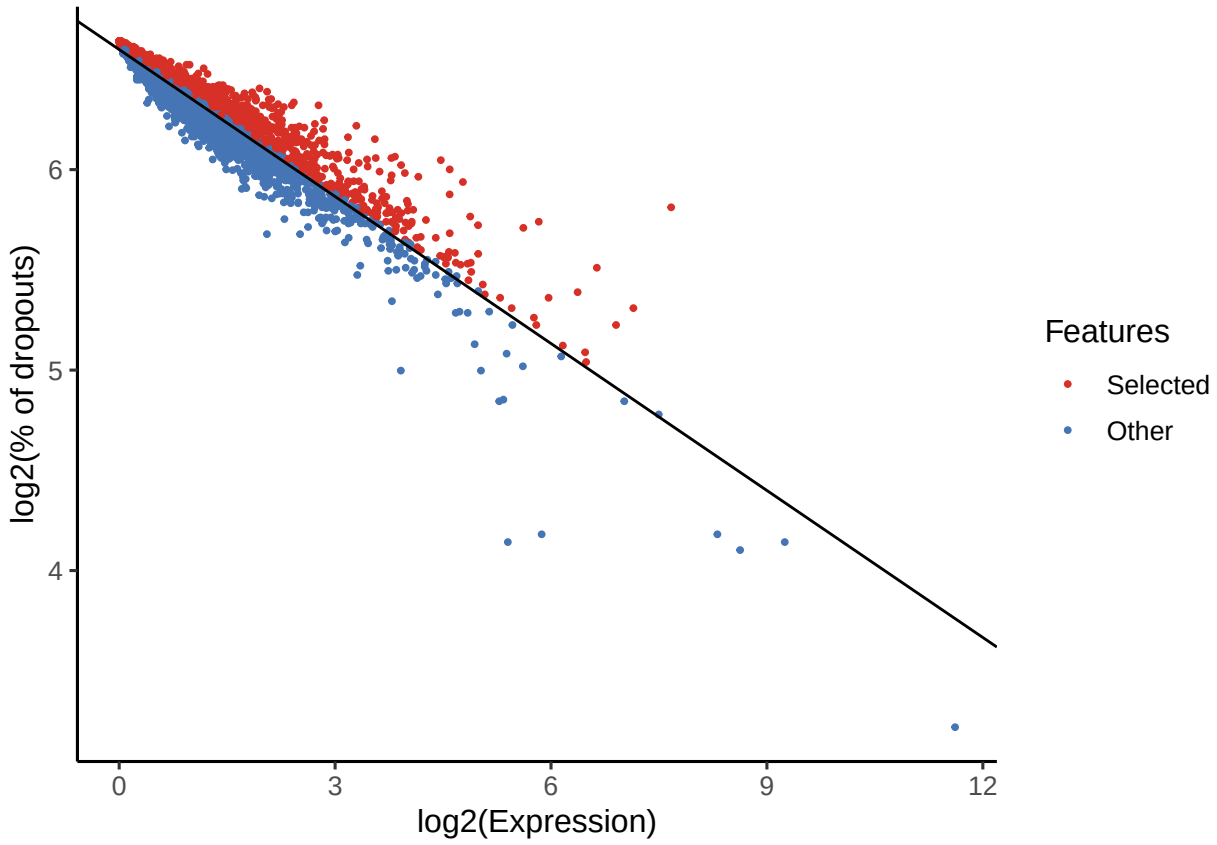
```
## Warning in mle2(LL, start = list(krt = 3, sigma = 0.25)): convergence failure:
## code=1 (iteration limit 'maxit' reached)
```




```
library(scmap)
```

```
## Creating a generic function for 'toJSON' from package 'jsonlite' in package 'googleVis'
```

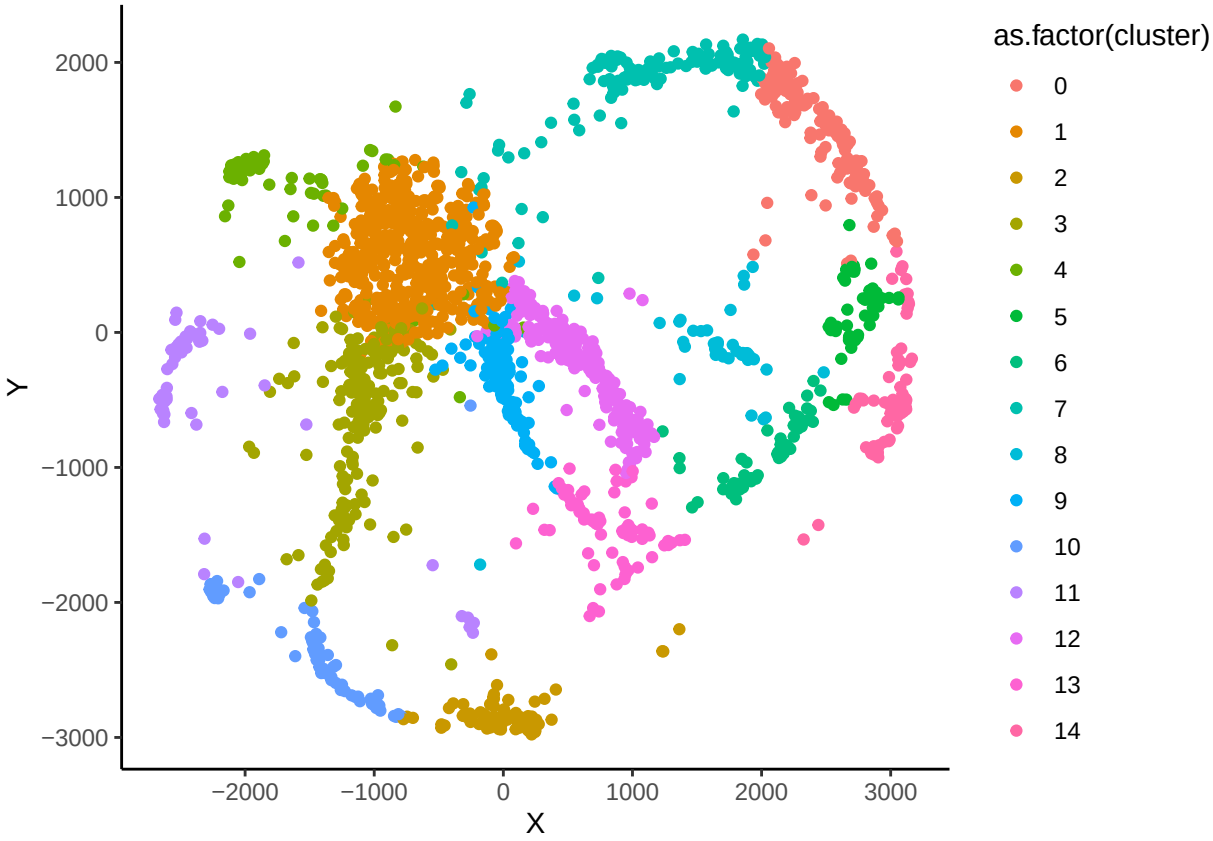
```
rowData(mca.qc.sub)$feature_symbol <- rownames(mca.qc.sub)
test <- selectFeatures(mca.qc.sub, suppress_plot = FALSE, n_features = 2000)
```



```
test2 <- test[rowData(test)$scmap_features == TRUE, ]

# write.csv(as.data.frame(counts(test2)),
# file='submca.qc.counts.elifeasex.subook.featslxn_forgenegraph.csv')
```

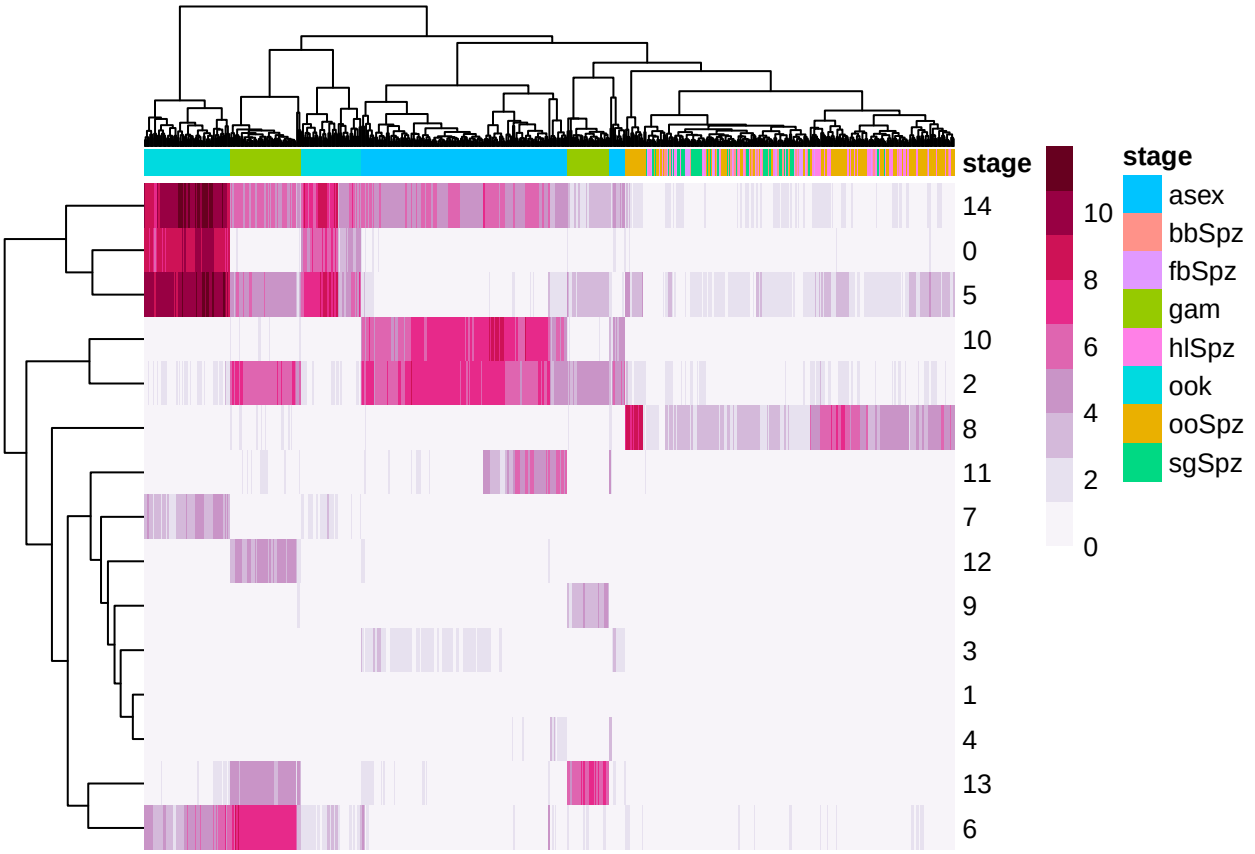
k15 with feature selection

```
coord <- read.csv("/Users/vh3/Documents/Ookinetes_2020/genegraph/subclusters.submca.subook.elifeasex.fea
    header = TRUE)
colnames(coord) <- c("gene_id", "X", "Y", "cluster")

ggplot(coord, aes(X, Y)) + geom_point(aes(colour = as.factor(cluster))) + theme_classic()
```




```
rowData(test2)$X_graph <- coord[match(rownames(test2), coord[, 1]), 2]
rowData(test2)$Y_graph <- coord[match(rownames(test2), coord[, 1]), 3]
rowData(test2)$Cluster_k15 <- coord[match(rownames(test2), coord[, 1]), 4]



lcpm_mat <- as.data.frame(assays(test2)[["logcounts"]])

lcpm_mat$Cluster_k15 <- rowData(test2)$Cluster_k15
lcpm_mat <- as.data.frame(lcpm_mat)
lcpm_mat_sub <- lcpm_mat[!is.na(lcpm_mat$Cluster_k15), ]

clustmean <- aggregate(lcpm_mat_sub[, 1:623], by = list(as.factor(as.character(lcpm_mat_sub$Cluster_k15
    mean)
rownames(clustmean) <- clustmean$Group.1
```

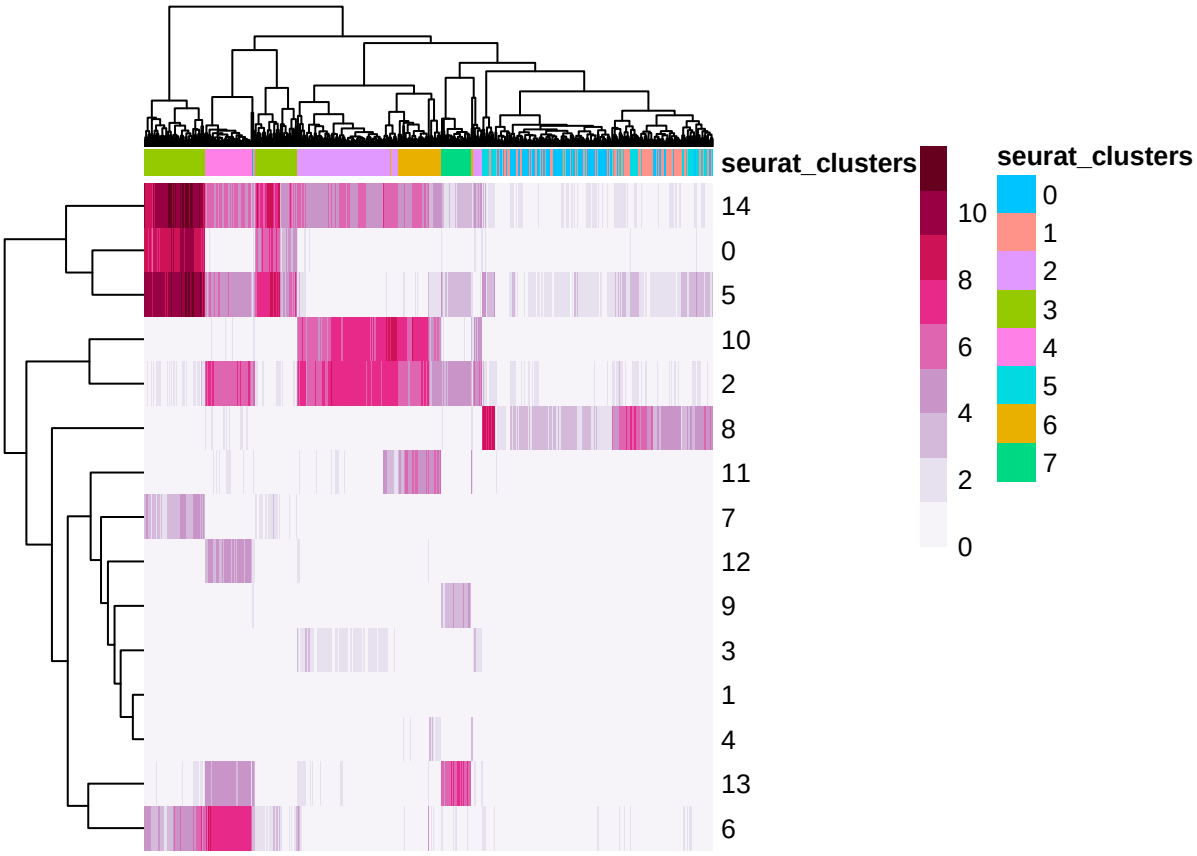
```
clustmean2 <- clustmean[, 2:624]


stage <- as.data.frame(colData(mca.qc.sub)["stage"])
cluster <- as.data.frame(colData(mca.qc.sub)["seurat_clusters"])


pheatmap(clustmean2, annotation_col = stage, show_colnames = FALSE, show_rownames = TRUE,
    color = brewer.pal(9, "PuRd"))
```




```
table(rowData(mca.qc.sub)$Cluster_k15)
```

```
## < table of extent 0 >
```

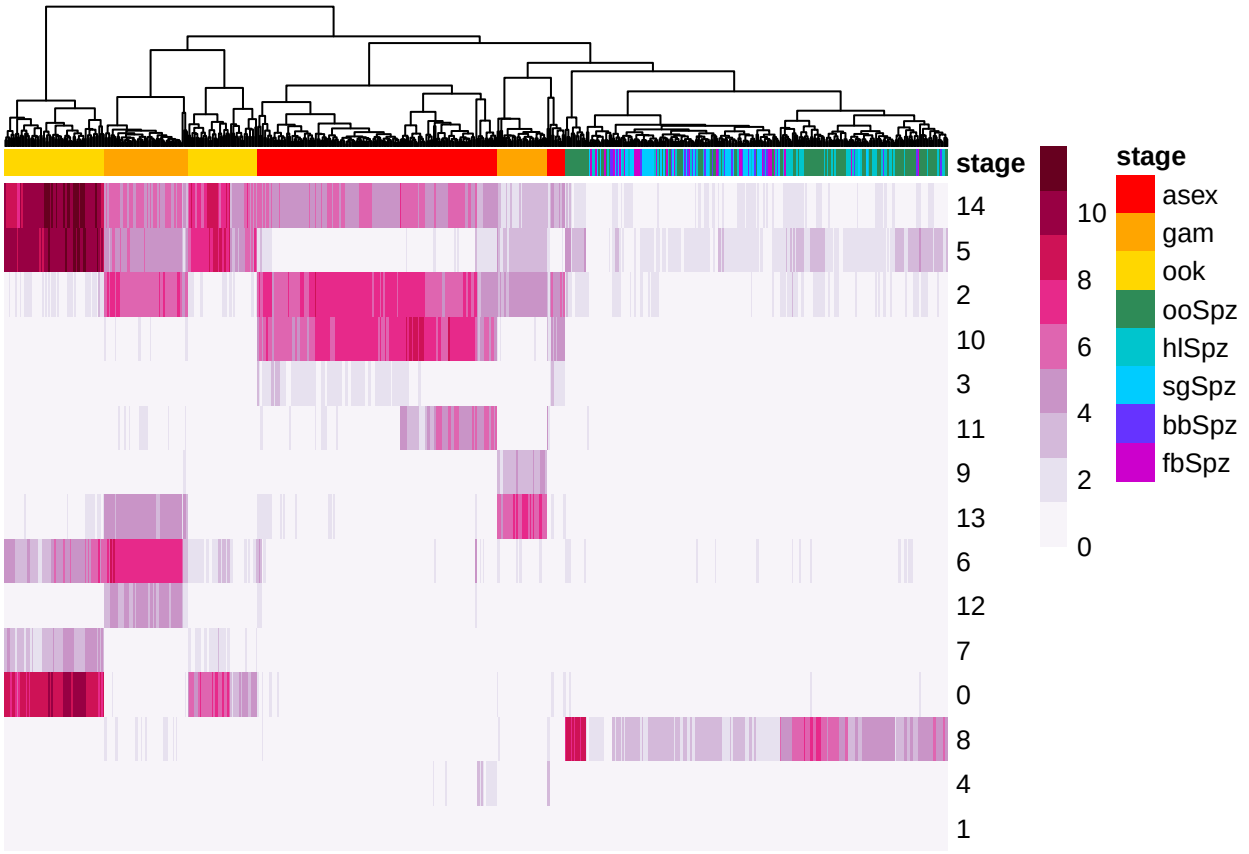
```
pheatmap(clustmean2, annotation_col = cluster, show_colnames = FALSE, show_rownames = TRUE,
    color = brewer.pal(9, "PuRd"))
```



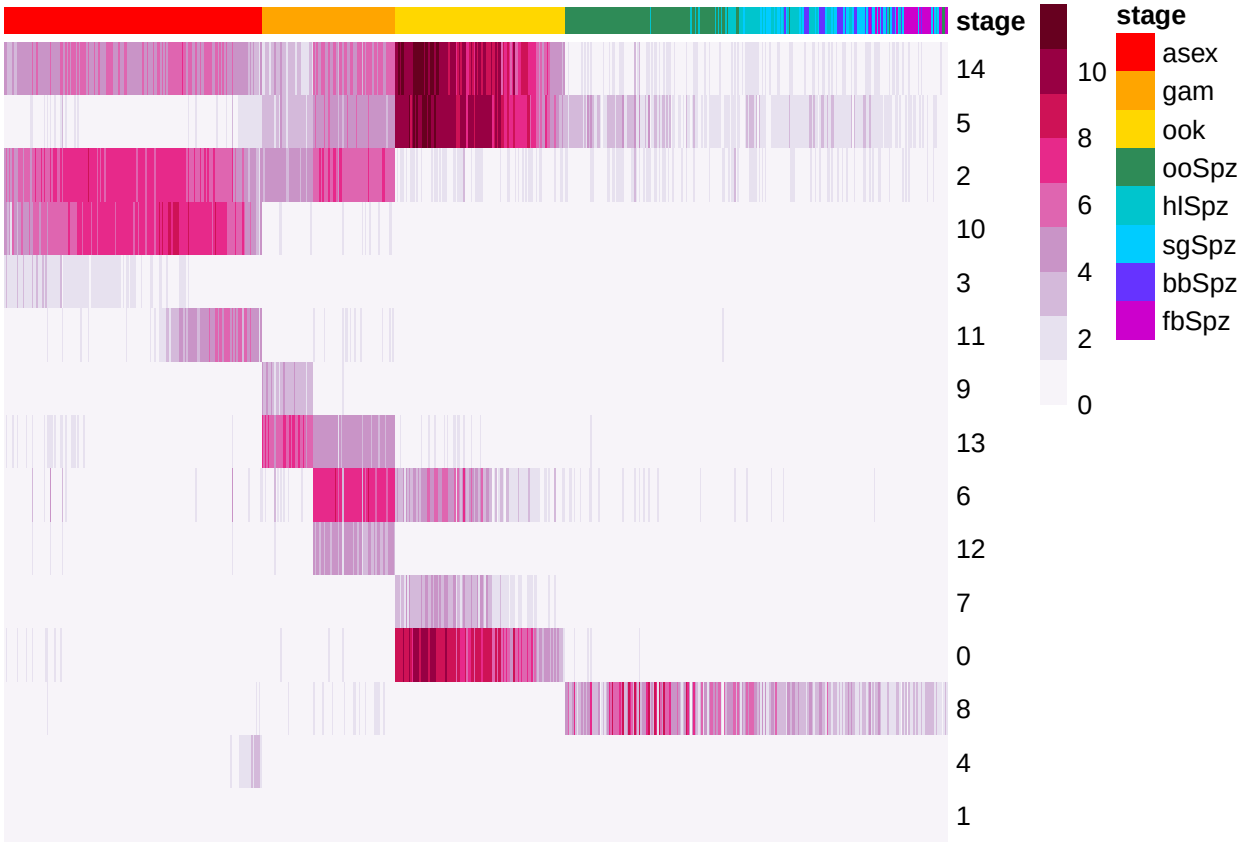

```
table(rowData(mca.qc.sub)$Cluster_k15)
```

```
## < table of extent 0 >
```

```
clustmean <- aggregate(lcpm_mat_sub[, 1:623], by = list(as.factor(as.character(lcpm_mat_sub$Cluster_k15)),
    mean)
rownames(clustmean) <- clustmean$Group.1

# create a vector with letters in the desired order x <- c(5, 10, 11, 2, 9, 6, 7,
# 13, 3, 14, 1, 4, 12, 0, 8)
x <- c(14, 5, 2, 10, 3, 11, 9, 13, 6, 12, 7, 0, 8, 4, 1)
clustmean3 <- clustmean %>% slice(match(x, clustmean$Group.1))

rownames(clustmean3) <- clustmean3$Group.1

clustmean3 <- clustmean3[, 2:624]


colors = c(asex = "red", gam = "orange", ook = "gold", ooSpz = "seagreen", hlSpz = "turquoise3",
    sgSpz = "#00CCFF", bbSpz = "#6633FF", fbSpz = "#CC00CC")

ann_c <- list(stage = colors)

pheatmap(clustmean3, annotation_col = stage, cluster_rows = FALSE, show_colnames = FALSE,
    show_rownames = TRUE, color = brewer.pal(9, "PuRd"), annotation_colors = ann_c)
```

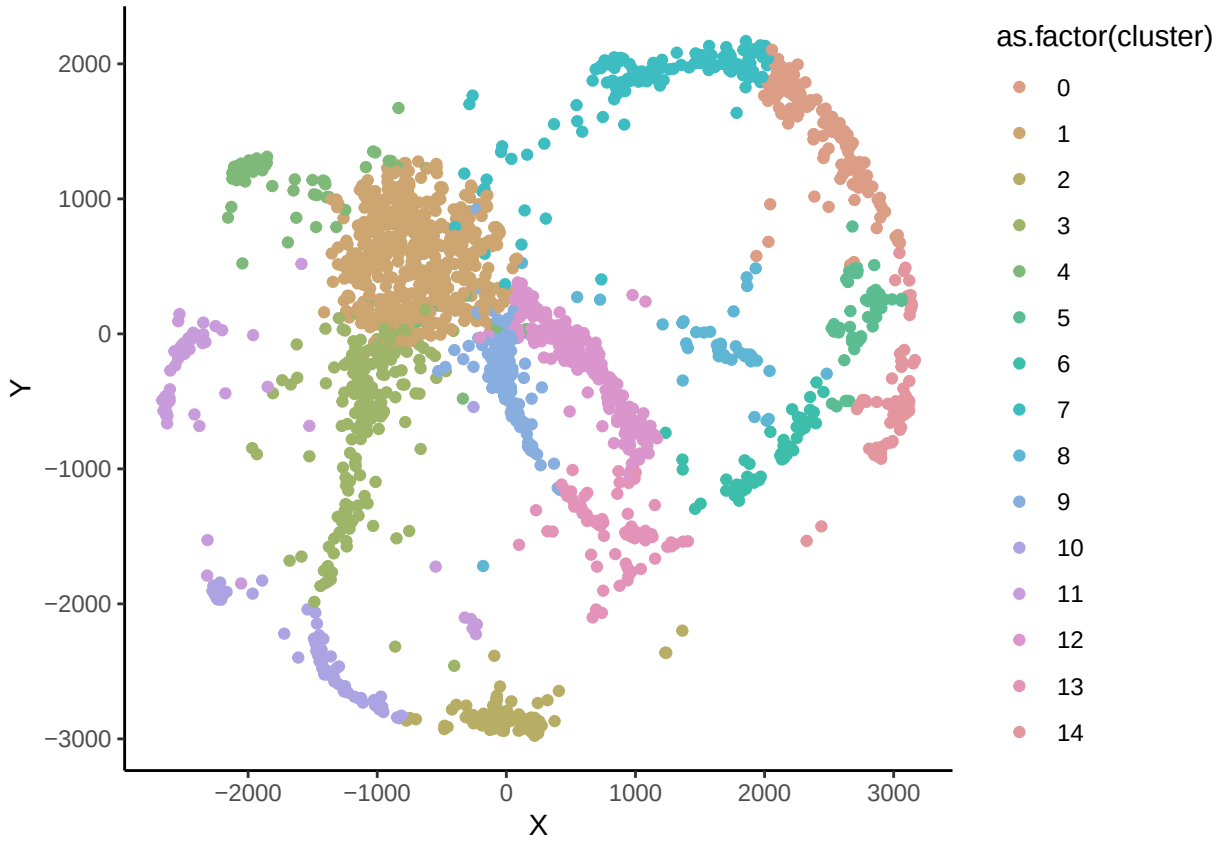



```
ppt <- read.csv("/Users/vh3/Documents/PfMCA/ANALYSIS_2/ppt_20200625.csv")
subppt <- ppt[ppt$ppt %in% colnames(clustmean3), ]
#x <- ppt$ppt

clustmean4 <- clustmean3[, subppt]

pheatmap(clustmean4, annotation_col = stage, cluster_rows = FALSE, cluster_cols=FALSE, show_colnames = F
```

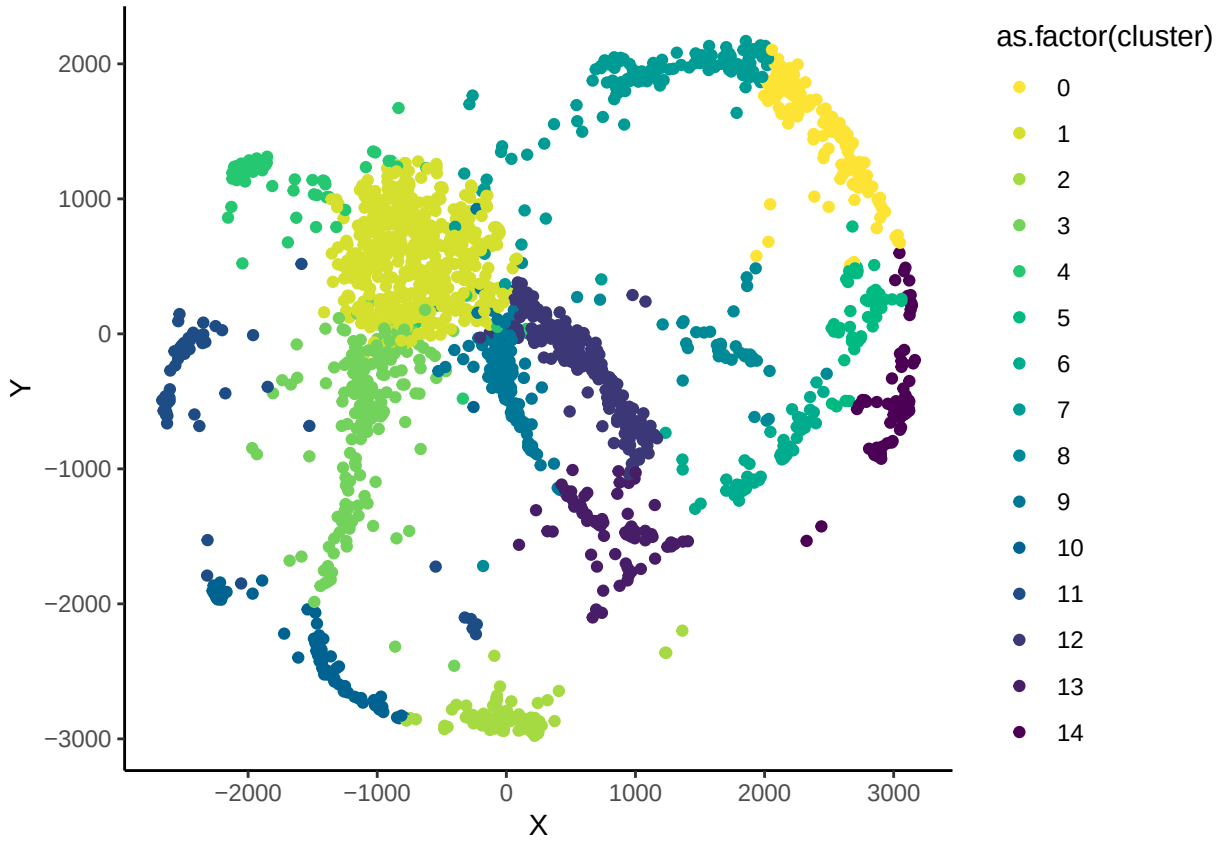



```
library(colorspace)
hcl_palettes(plot = TRUE)
```

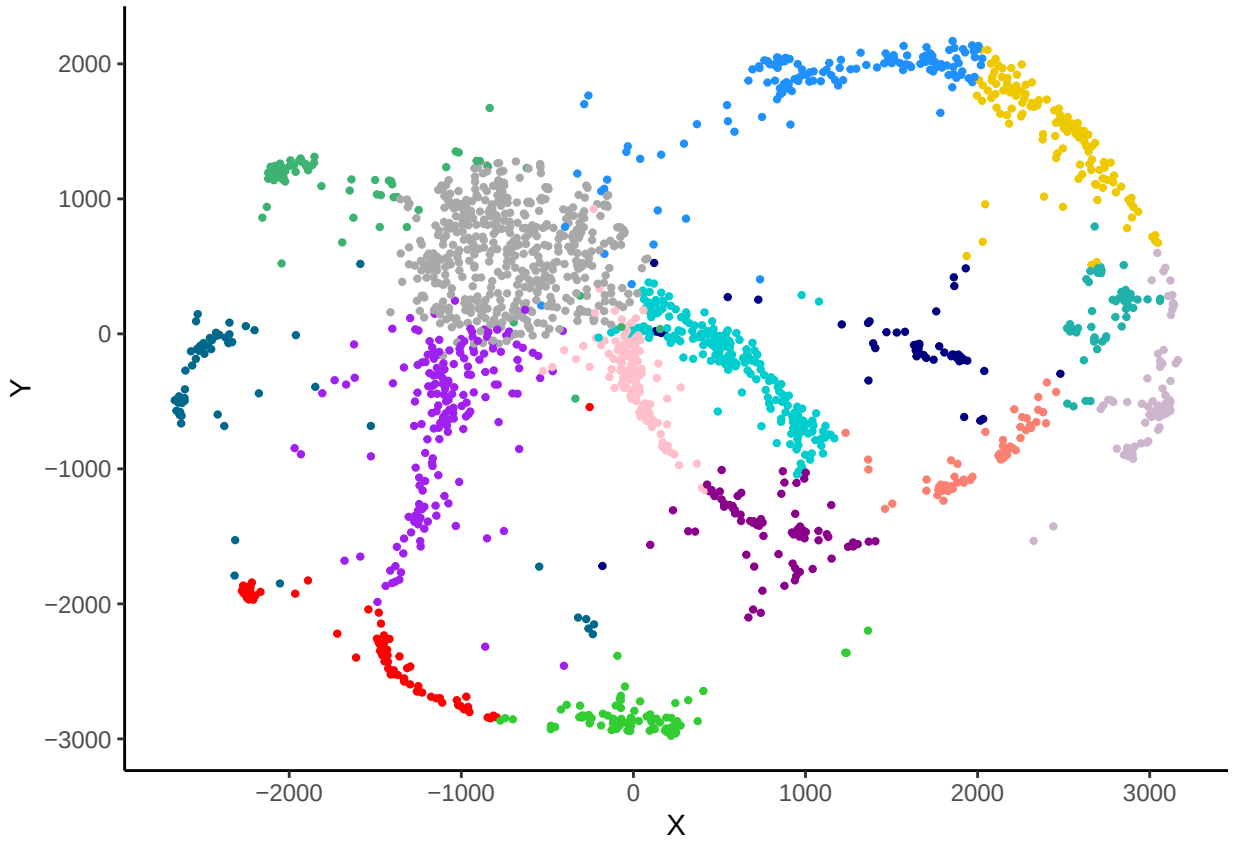

```
ggplot(coord, aes(X, Y)) + geom_point(aes(colour=as.factor(cluster))) +
  scale_color_discrete_qualitative(palette = "Dynamic") +
  theme_classic()
```



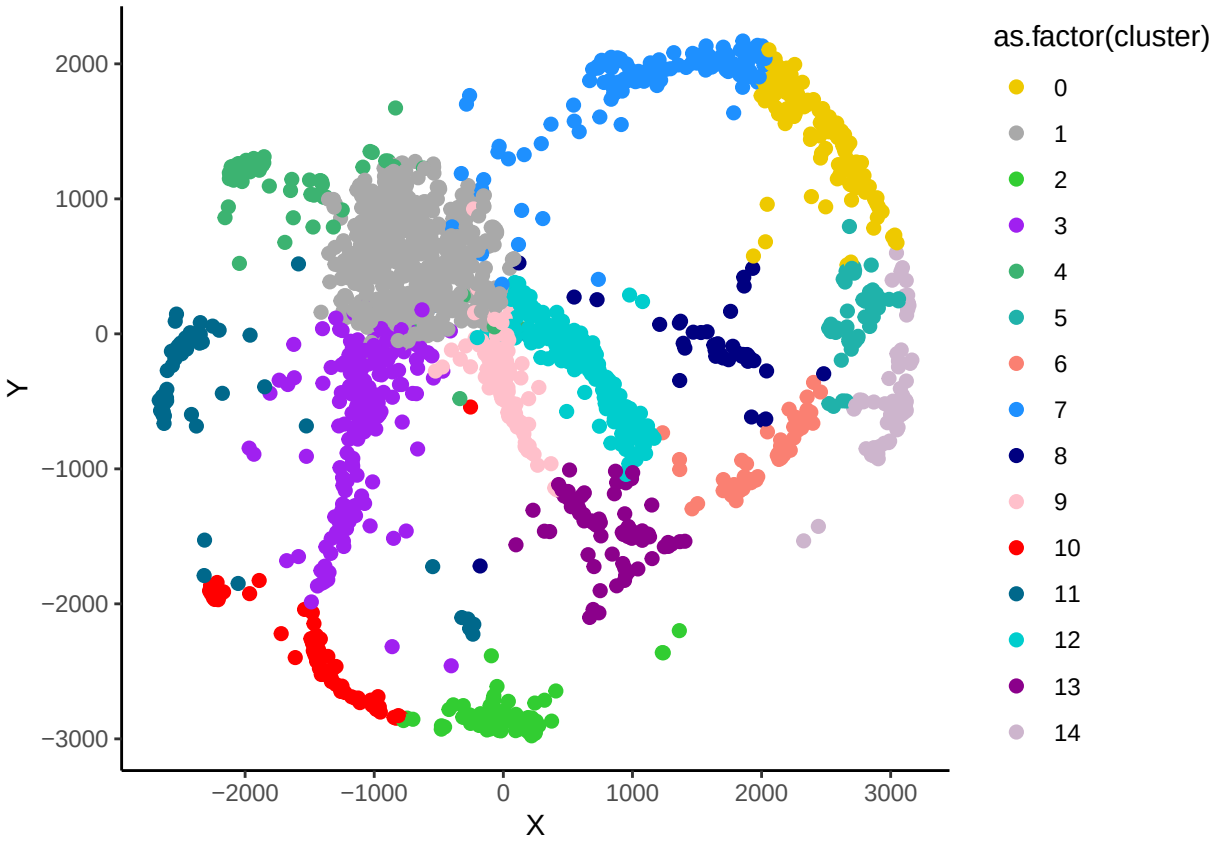

```
ggplot(coord, aes(X, Y)) + geom_point(aes(colour=as.factor(cluster))) +
  scale_color_discrete_sequential(palette = "Viridis") +
  theme_classic()
```



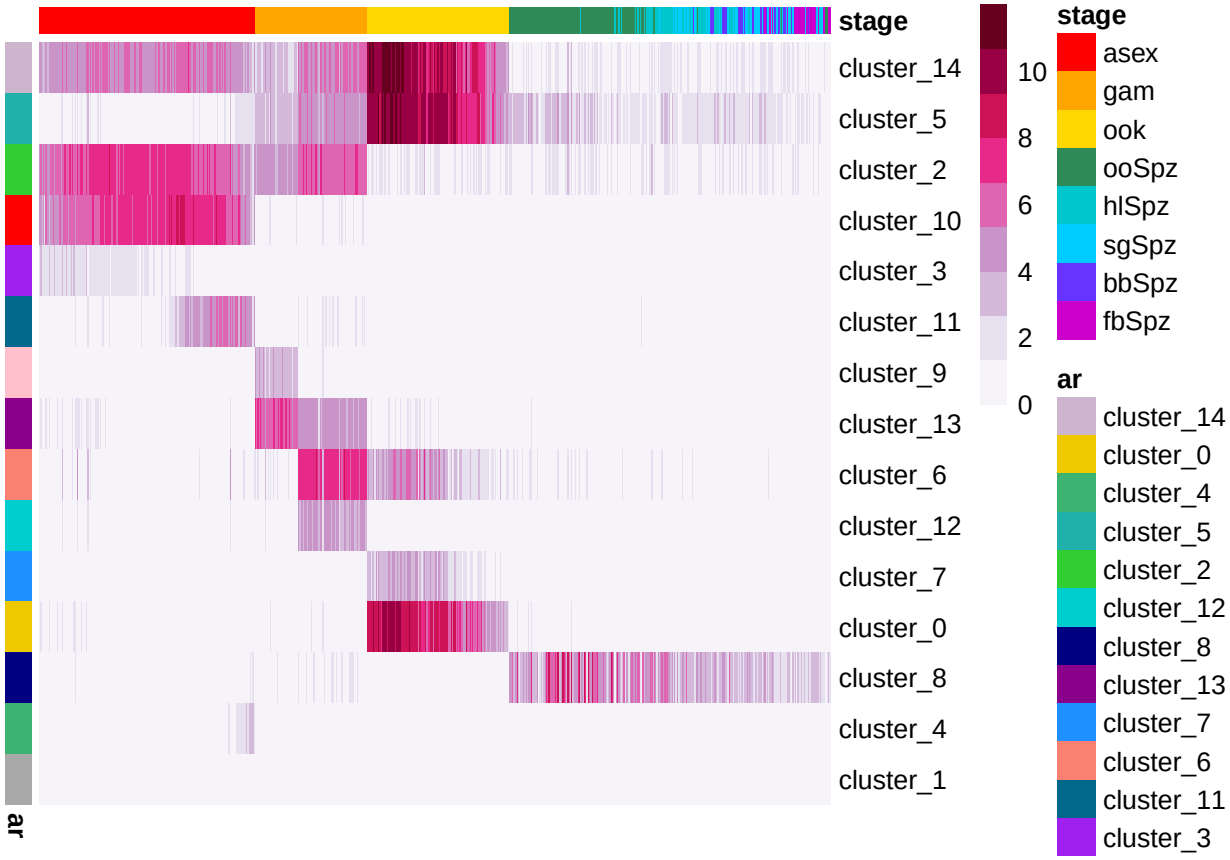

```
clust_col <- c("14" = "thistle3", "0" = "gold2", "4" = "mediumseagreen", "5" = "lightseagreen", "19" =


ggplot(coord, aes(X, Y)) + geom_point(aes(colour=as.factor(cluster)), size=0.8) +
  scale_color_manual(values = clust_col) +
  theme_classic() + theme(legend.position = "none")
```




```
ggplot(coord, aes(X, Y)) + geom_point(aes(colour=as.factor(cluster)), size=2) +
  scale_color_manual(values = clust_col) +
  theme_classic()
```




```
clust_col <- c("cluster_14" = "thistle3", "cluster_0" = "gold2", "cluster_4" = "mediumseagreen", "cluste

x <- c(14, 5, 2, 10, 3, 11, 9, 13, 6, 12, 7, 0, 8, 4, 1)
x2 <- as.data.frame(x)
x2$x <- as.character(x2$x)
x2$ar <- paste("cluster_", x2$x, sep="")
rownames(x2) <- x2$ar
ar <- x2["ar"]
clustmean5 <- clustmean4
rownames(clustmean5) <- ar$ar

ann_c <- list(
  stage = colors,
  ar = clust_col
)


pheatmap(clustmean5, annotation_col = stage, annotation_row = ar, cluster_rows = FALSE, cluster_cols=FA
```


stage
cluster_14
cluster_5
cluster_2
cluster_10
cluster_3
cluster_11
cluster_9
cluster_13
cluster_6
cluster_12
cluster_7
cluster_0
cluster_8
cluster_4
cluster_1
ar
10
8
6
4
2
0
stage
asex
gam
ook
ooSpz
hlSpz
sgSpz
bbSpz
fbSpz
ar
cluster_14
cluster_0
cluster_4
cluster_5
cluster_2
cluster_12
cluster_8
cluster_13
cluster_7
cluster_6
cluster_11
cluster_3