

MERFISH example

Jean Fan

11/17/2020

MERFISH

[add description of data]

Load preprocessed data

```
col = MERFISH$col
pcs = MERFISH$pcs
vel = MERFISH$vel
```

Load cell cycle genes - keep merfish genes that are cell cycle genes

```
cycle.genes = read.csv("/Users/lylaatta/Desktop/geneset_278.csv", header = FALSE)
# cycle.genes = cycle.genes$Gene
cycle.genes = cycle.genes$V1

curr = vel$current
proj = vel$projected

merfish.genes = rownames(curr)
merfish.cycle.genes = merfish.genes[which(merfish.genes %in% cycle.genes)]

curr.cycle = curr[merfish.cycle.genes,]
proj.cycle = proj[merfish.cycle.genes,]
```

Build VeloViz graph on all genes

```
# , fig.height = 11, fig.width = 7
curr = vel$current
proj = vel$projected

veloviz = buildVeloviz(
  curr = curr,
  proj = proj,
  normalize.depth = TRUE,
  use.ods.genes = FALSE,
  pca = TRUE,
  nPCs = 5,
  center = TRUE,
  scale = TRUE,
  k = 5,
  similarity.threshold = 0,
  distance.weight = 0.01,
  distance.threshold = 0.9,
  weighted = TRUE,
```

```

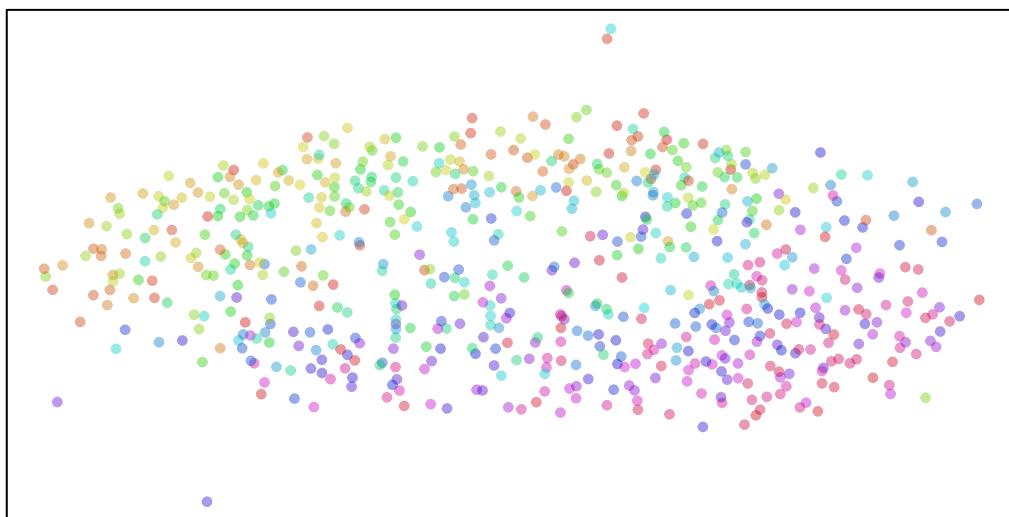
    seed = 0,
    verbose = FALSE
)

## Warning in if (!class(curr) %in% c("dgCMatrix", "dgTMatrix")) {: the condition
## has length > 1 and only the first element will be used
## Warning in if (!class(proj) %in% c("dgCMatrix", "dgTMatrix")) {: the condition
## has length > 1 and only the first element will be used
# par(mfrow=c(3,2))
emb.veloviz = veloviz$fdg_coords
plotEmbedding(emb.veloviz, groups=col[rownames(emb.veloviz)], main='veloviz')

```

veloviz

C2



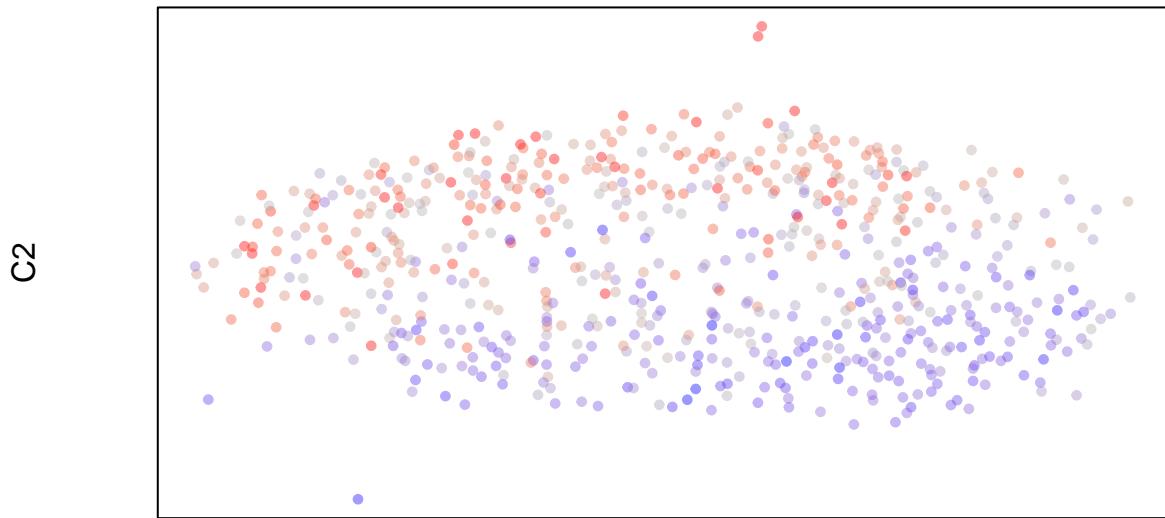
C1

```

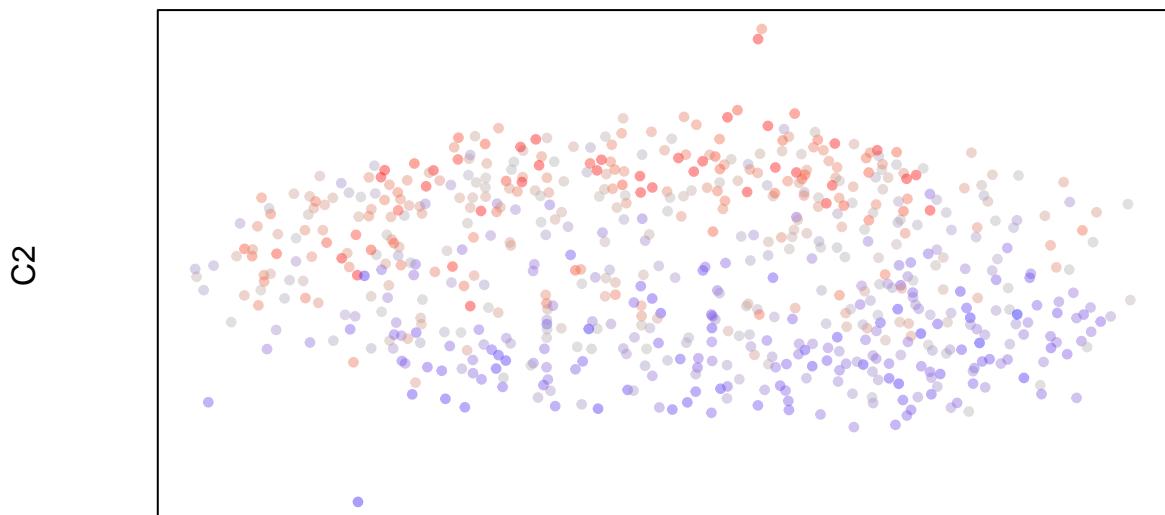
# g = plotVeloviz(veloviz, clusters=col[rownames(emb.veloviz)], seed=0, verbose=TRUE)
sapply(c('MCM6', 'CDC6', 'KIF2C', 'CCNF'), function(g) {
  gexp <- scale(curr[g,])[,1]
  plotEmbedding(emb.veloviz, col=gexp, main=g)
})

```

MCM6

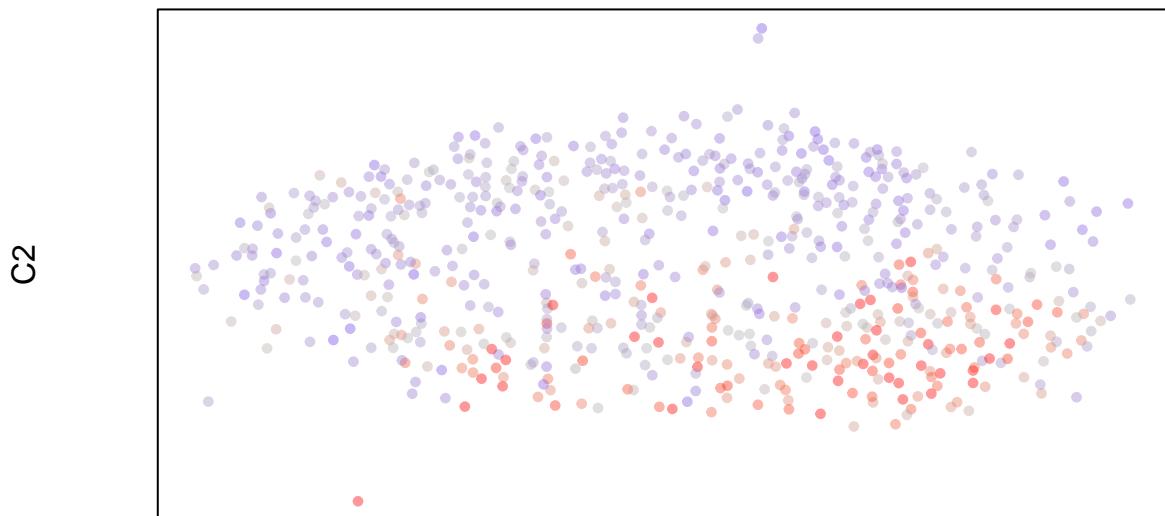


**C1
CDC6**

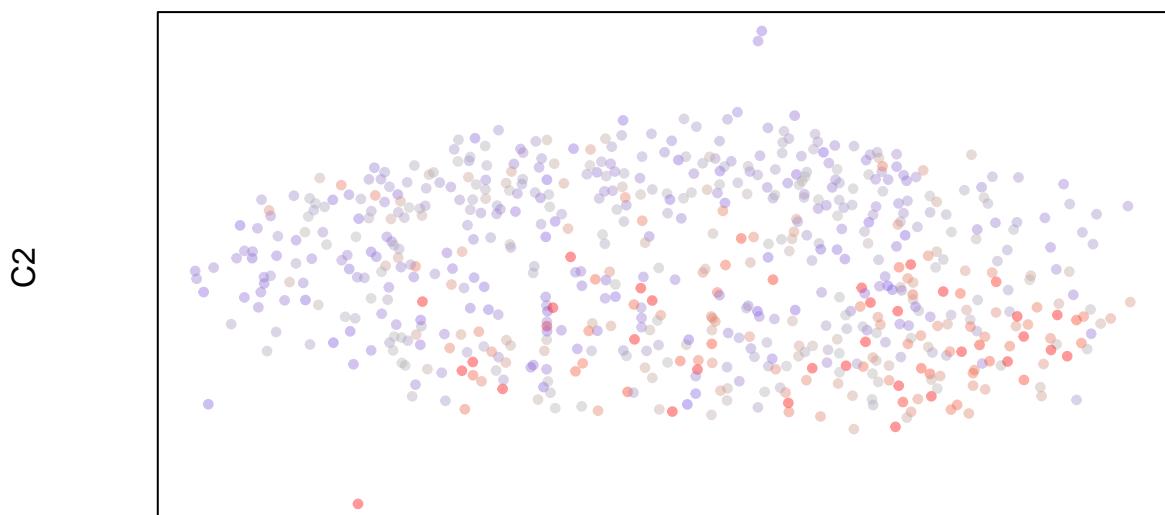


C1

KIF2C



C1 CCNF



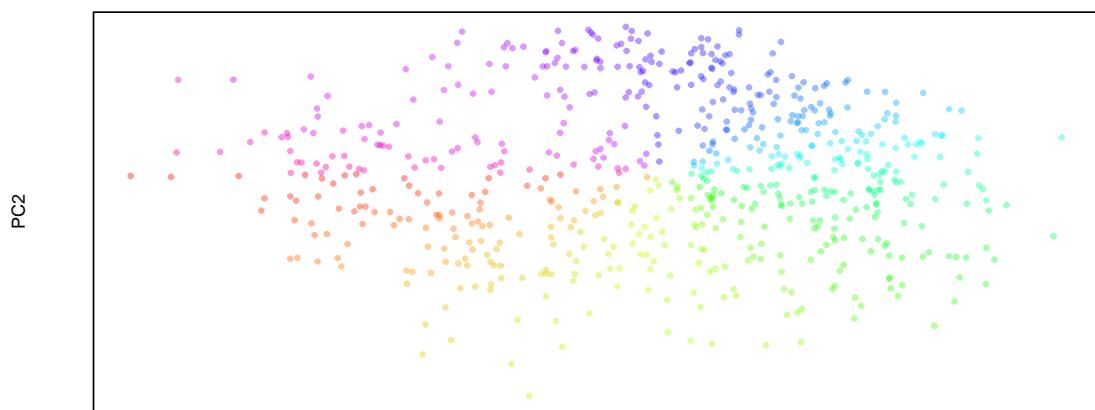
C1

Compare to other embeddings

```
#PCA
layout(matrix(c(1,1,2,3,4,5), 3, 2, byrow = TRUE))
emb.pcs = pcs[,1:2]
plotEmbedding(emb.pcs, col=col, main='PCA')
```

```
sapply(c('MCM6', 'CDC6', 'KIF2C', 'CCNF'), function(g) {  
  gexp <- scale(curr[g,])[,1]  
  plotEmbedding(emb.pcs, col=gexp, main=g)  
})
```

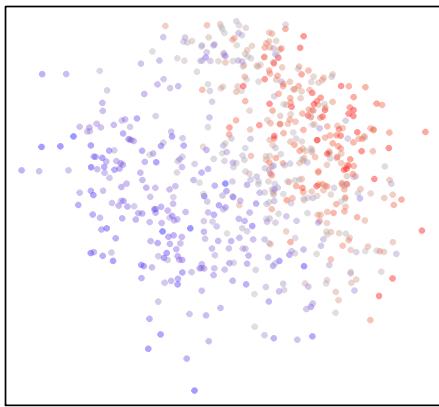
PCA



PC1

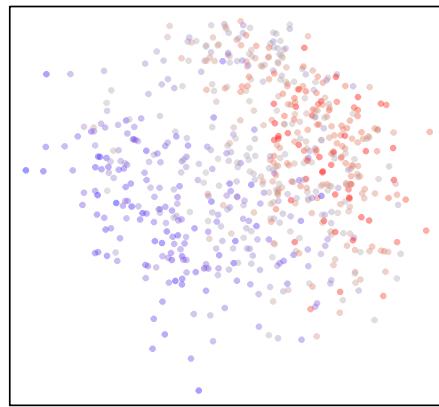
MCM6

PC2



CDC6

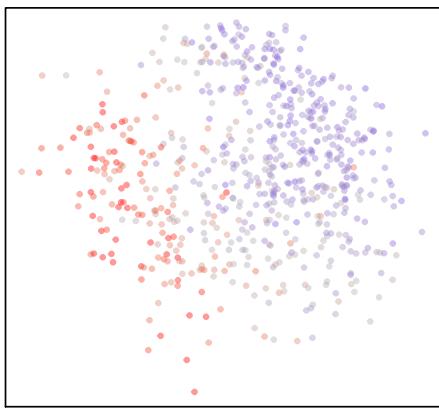
PC2



PC1

KIF2C

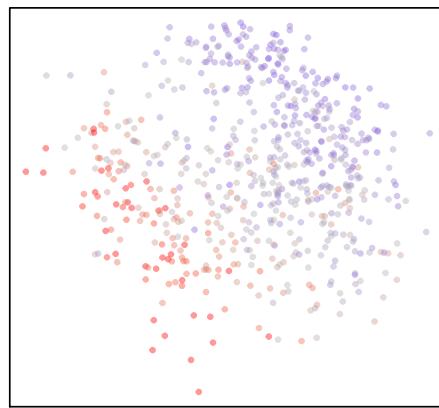
PC2



PC1

CCNF

PC2

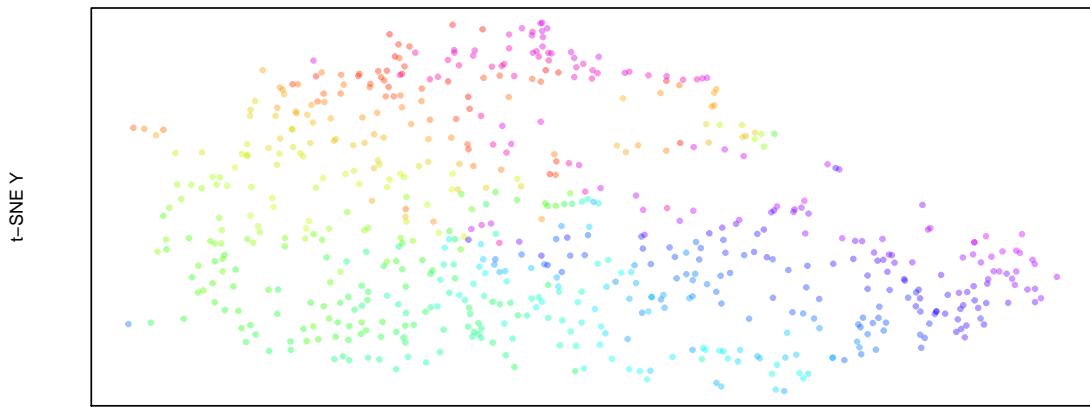


PC1

PC1

```
#tSNE
set.seed(1)
layout(matrix(c(1,1,2,3,4,5), 3, 2, byrow = TRUE))
emb.tsne = Rtsne::Rtsne(pcs[,1:5], perplexity = 100)$Y
rownames(emb.tsne) = rownames(pcs)
plotEmbedding(emb.tsne, col=col, main='tSNE',
              xlab = "t-SNE X", ylab = "t-SNE Y")
sapply(c('MCM6', 'CDC6', 'KIF2C', 'CCNF'), function(g) {
  gexp <- scale(curr[g,])[,1]
  plotEmbedding(emb.tsne, col=gexp, main=g)
})
```

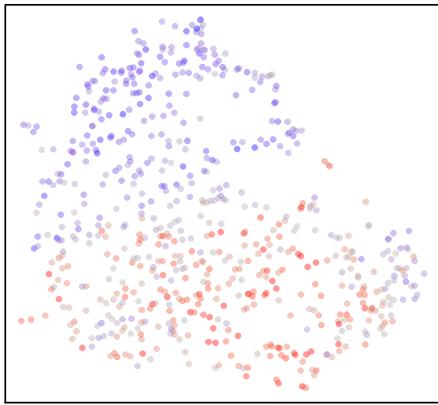
tSNE



t-SNE X

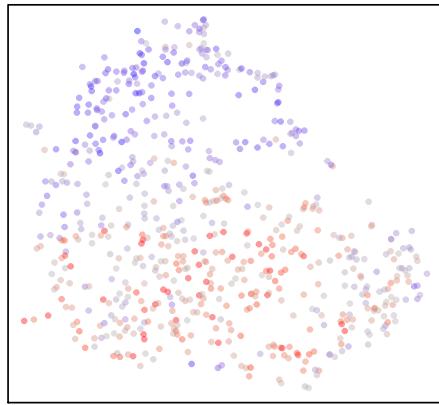
MCM6

emb[2]



CDC6

emb[2]

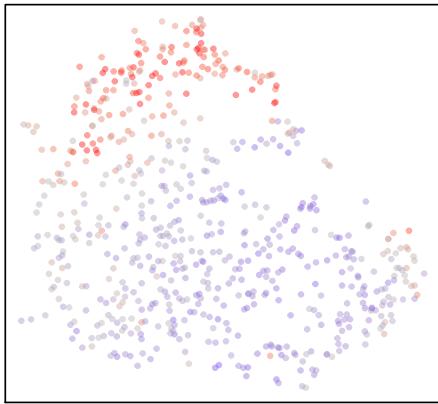


emb[1]

emb[1]

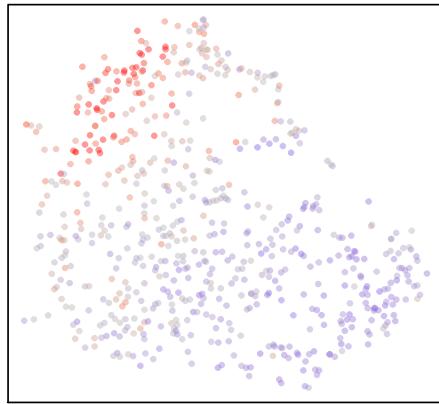
KIF2C

emb[2]



CCNF

emb[2]

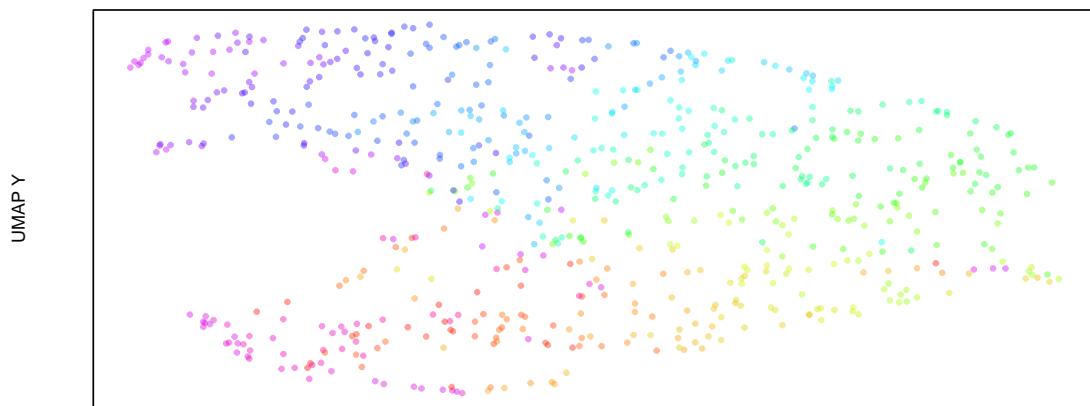


emb[1]

emb[1]

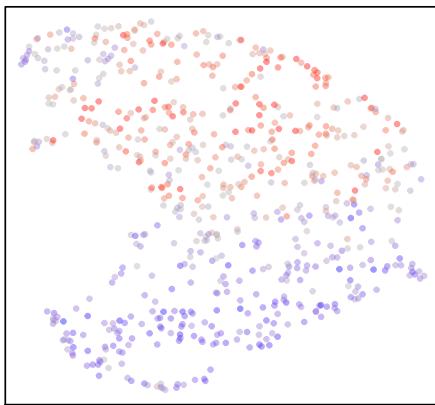
```
#UMAP
set.seed(1)
layout(matrix(c(1,1,2,3,4,5), 3, 2, byrow = TRUE))
emb.umap = umap::umap(pcs[,1:5], min_dist = 0.3)$layout
rownames(emb.umap) = rownames(pcs)
plotEmbedding(emb.umap, col=col, main='UMAP',
             xlab = "UMAP X", ylab = "UMAP Y")
sapply(c('MCM6', 'CDC6', 'KIF2C', 'CCNF'), function(g) {
  gexp <- scale(curr[g,])[,1]
  plotEmbedding(emb.umap, col=gexp, main=g)
})
```

UMAP



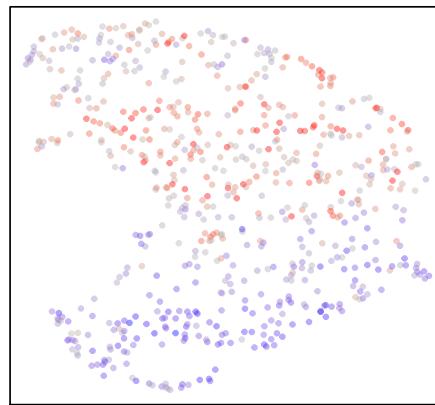
MCM6

emb[2]



CDC6

emb[2]

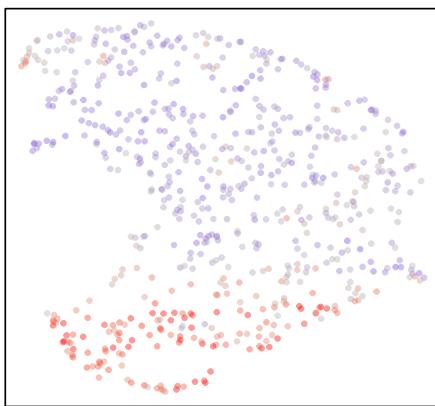


emb[1]

emb[1]

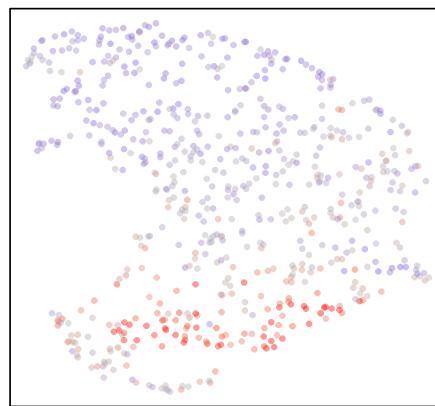
KIF2C

emb[2]



CCNF

emb[2]

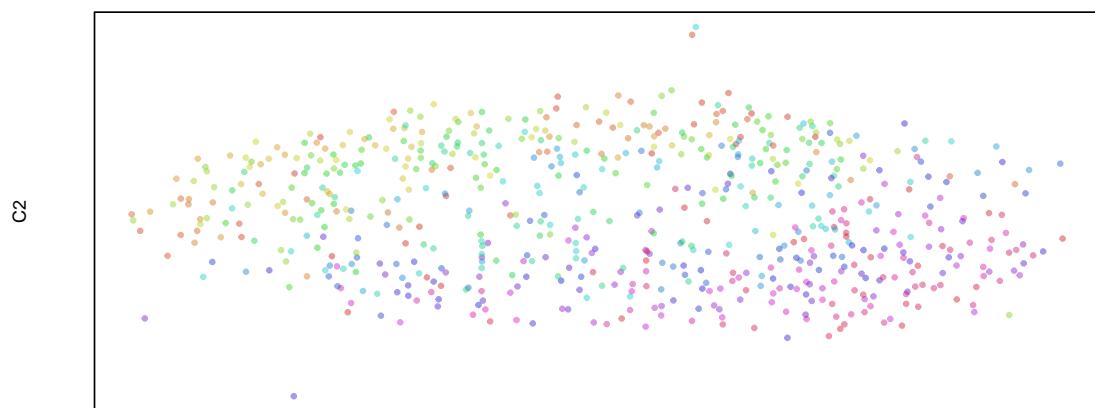


emb[1]

emb[1]

```
#veloviz
layout(matrix(c(1,1,2,3,4,5), 3, 2, byrow = TRUE))
plotEmbedding(emb.veloviz, groups=col[rownames(emb.veloviz)], main='veloviz')
sapply(c('MCM6', 'CDC6', 'KIF2C', 'CCNF'), function(g) {
  gexp <- scale(curr[g,])[,1]
  plotEmbedding(emb.veloviz, col=gexp, main=g)
})
```

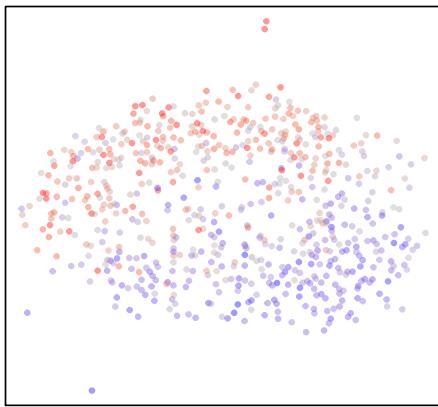
veloviz



C1

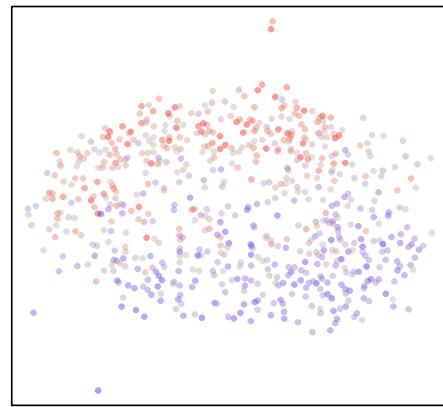
MCM6

C2



CDC6

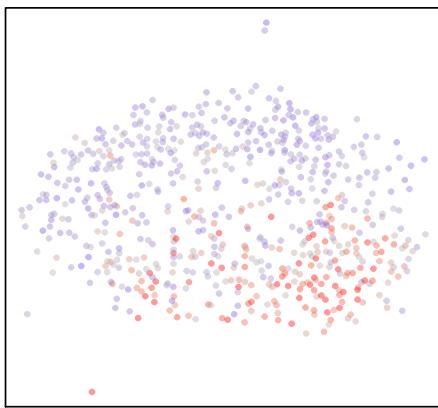
C2



C1

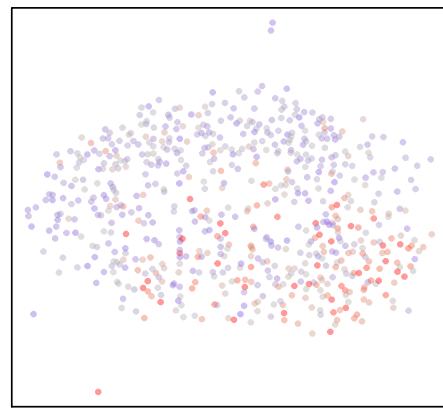
KIF2C

C2



CCNF

C2



C1

C1

Build VeloViz graph on cell cycle genes

```
veloviz.cycle = buildVeloviz(
  curr = curr.cycle,
  proj = proj.cycle,
  normalize.depth = TRUE,
  use.ods.genes = FALSE,
  pca = TRUE,
  nPCs = 4,
  center = TRUE,
  scale = TRUE,
  k = 100,
  similarity.threshold = 0.25,
  distance.weight = 0.5,
  distance.threshold = 1,
  weighted = TRUE,
  seed = 0,
  verbose = FALSE
)

## Warning in if (!class(curr) %in% c("dgCMatrix", "dgTMatrix")) {: the condition
## has length > 1 and only the first element will be used

## Warning in if (!class(proj) %in% c("dgCMatrix", "dgTMatrix")) {: the condition
## has length > 1 and only the first element will be used

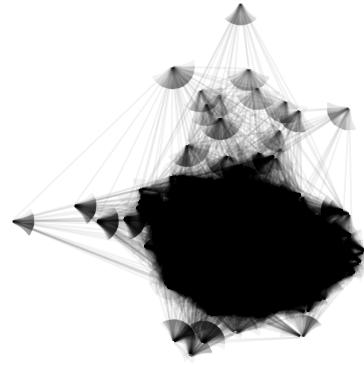
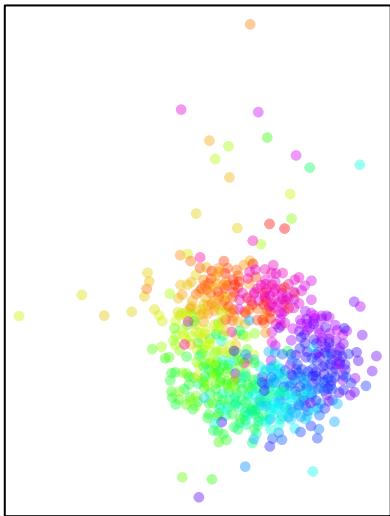
par(mfrow = c(1,2))
emb.veloviz.cycle = veloviz.cycle$fdg_coords
plotEmbedding(emb.veloviz.cycle, colors = col[rownames(emb.veloviz.cycle)], main='veloviz')
g = plotVeloviz(veloviz.cycle, clusters=col[rownames(emb.veloviz.cycle)], seed=0, verbose=TRUE)

## Warning in if (!is.na(clusters) & is.na(col)) {: the condition has length > 1
## and only the first element will be used

## Using provided clusters...
```

veloviz

C2

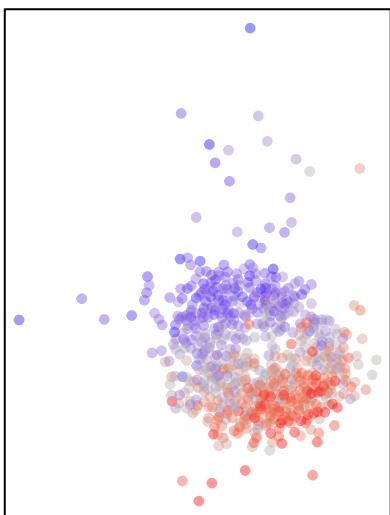


C1

```
sapply(c('MCM6', 'CDC6', 'KIF2C', 'CCNF'), function(g) {  
  gexp <- scale(curr[g,])[,1]  
  plotEmbedding(emb.veloviz.cycle, col=gexp, main=g)  
})
```

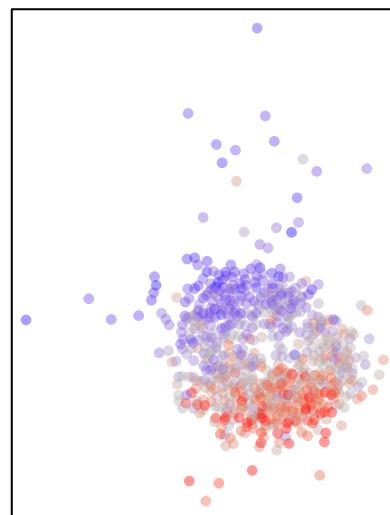
MCM6

C2



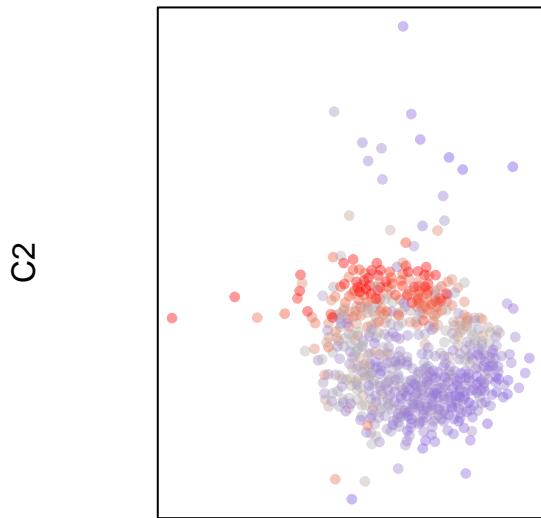
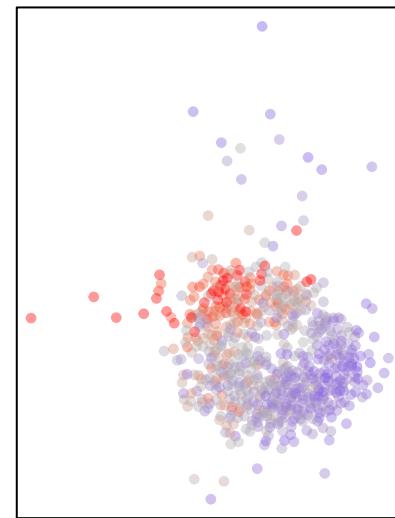
CDC6

C2



C1

C1

KIF2C**CCNF****C1****C1**

```
par(mfrow = c(1,1))
velocyto.R::show.velocity.on.embedding.cor(scale(emb.veloviz.cycle), vel,
                                             n = 50,
                                             scale='sqrt',
                                             cex=1, arrow.scale=1, show.grid.flow=TRUE,
                                             min.grid.cell.mass=10, grid.n=50, arrow.lwd=1,
                                             cell.colors=col[rownames(emb.veloviz.cycle)], main='VeloViz')
```

VeloViz

