

Simulation

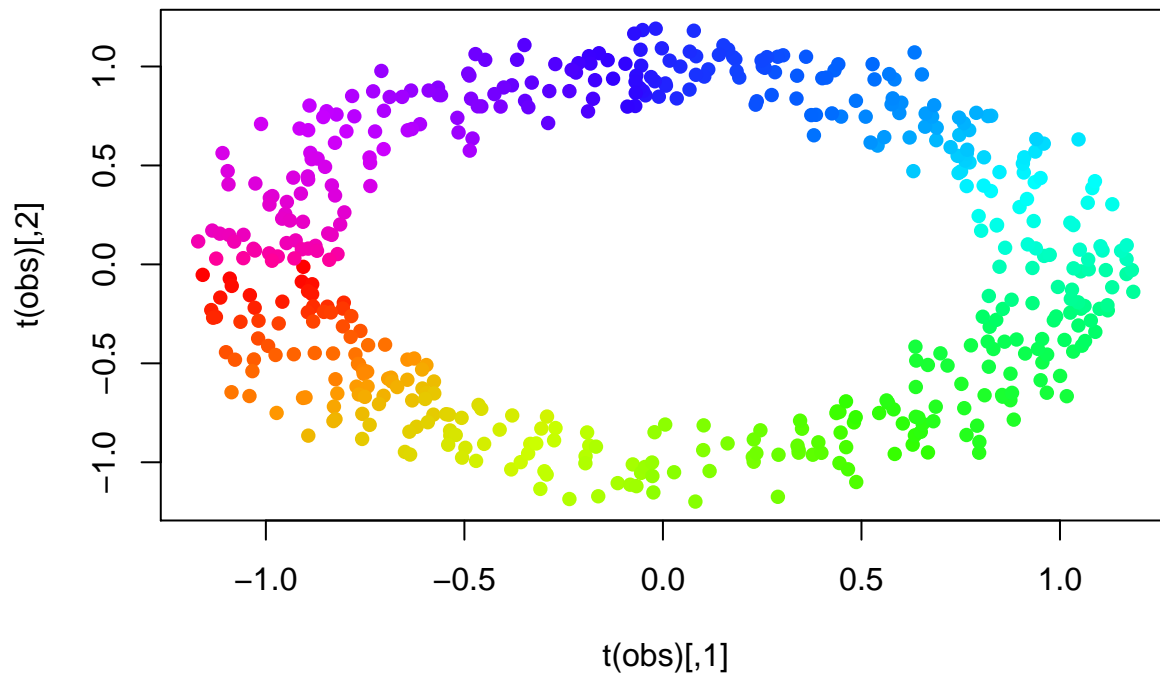
Jean Fan

11/13/2020

Simulation

```
## simulate data for current transcriptional state
## make circle like cell cycle
N <- 1000 ## number of cells
x <- matrix(rnorm(N),nc=2)
y <- x/sqrt(rowSums(x^2))
## add some jitter
obs <- t(y)
jitter <- 0.2
obs <- jitter(obs, amount = jitter)
## order points counterclockwise
angle <- atan2(obs[2,], obs[1,])
obs <- obs[, order(angle)]
labels <- paste0('cell', 1:ncol(obs))
colnames(obs) <- labels
## make into rainbow colors
col = colorRampPalette(c(rainbow(10)))(ncol(obs))
names(col) = labels

## plot
plot(t(obs),col=col, pch=16)
```

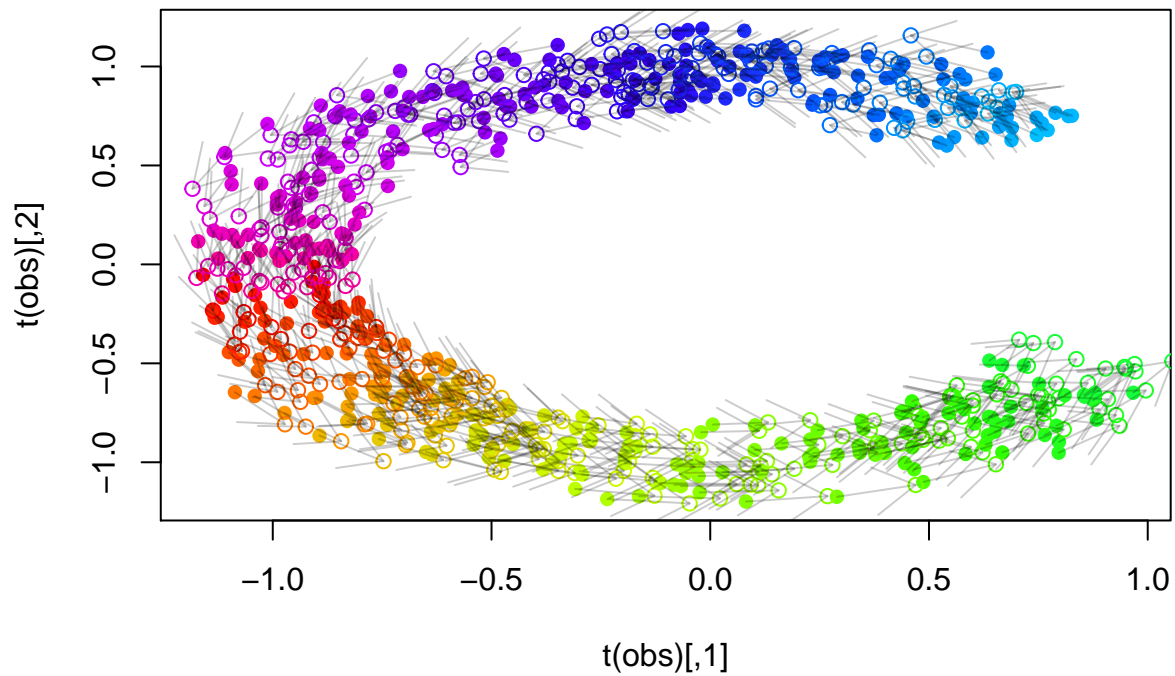


```
## remove some cells
cells.keep <- setdiff(labels, paste0('cell', 200:300))
labels <- labels[which(labels %in% cells.keep)]
obs <- obs[,cells.keep]
col <- col[cells.keep]

## plot
plot(t(obs),col=col, pch=16)

## rotate circle slightly to
## simulate future transcriptional state
rotate <- 0.05
f = pi*rotate # adjust as needed
#f = pi*0.2 # can't be too large relative to jitter noise
exp = t(obs)
exp[,1] = obs[,1]*cos(f) - obs[,2]*sin(f)
exp[,2] = obs[,2]*cos(f) + obs[,1]*sin(f)
exp = t(exp)
colnames(exp) <- labels

plot(t(obs),col=col, pch=16)
points(t(exp),col=col)
arrows(t(obs)[,1],t(obs)[,2],t(exp)[,1],t(exp)[,2], col=rgb(0,0,0,0.2))
```

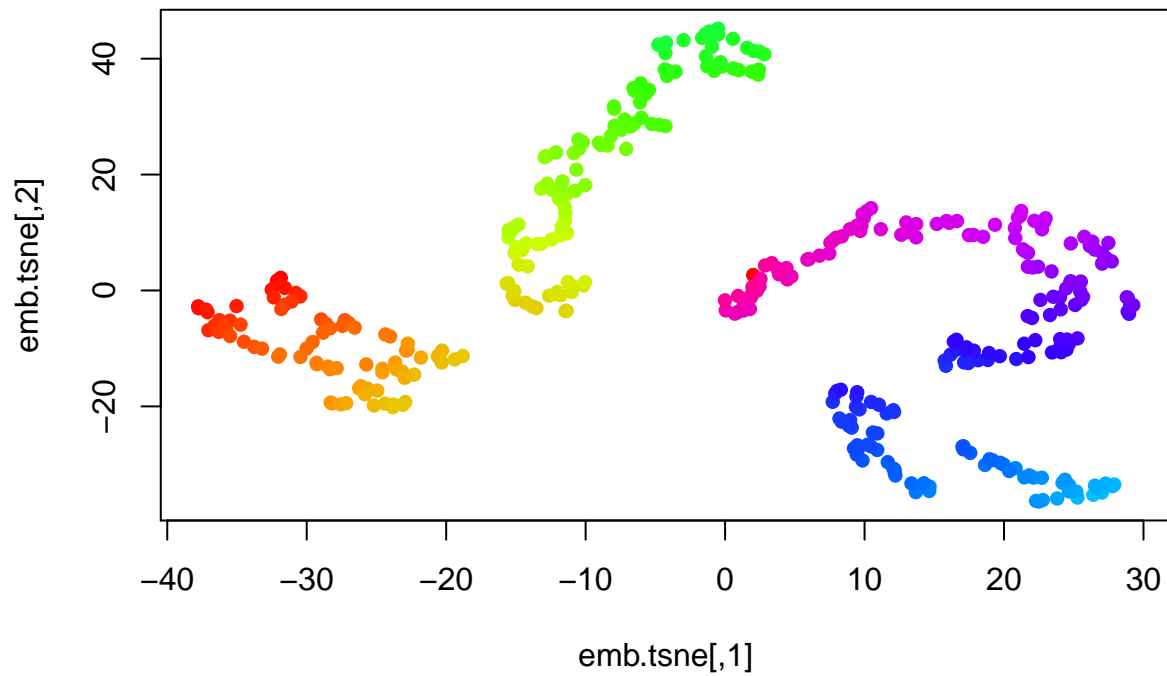


Compare embeddings

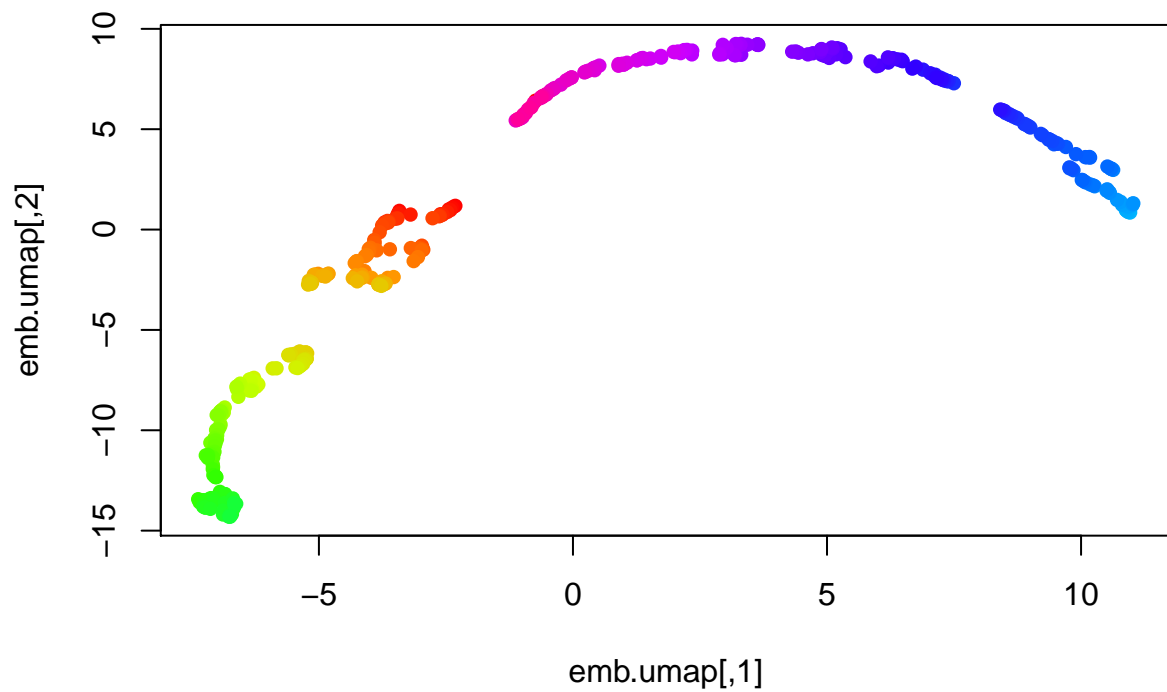
```
#library(veloviz)

## 2D embedding by tSNE
set.seed(0)
emb.tsne = Rtsne::Rtsne(t(obs), perplexity=10)$Y
plot(emb.tsne, col=col, pch=16)

## 2D embedding by UMAP
set.seed(0)
emb.umap = uwot::umap(t(obs), n_neighbors = 10)
```



```
plot(emb.umap, col=col, pch=16)
```



```
## 2D embedding by veloviz
vig = veloviz::buildVeloviz(
  curr = t(obs),
  proj = t(exp),
  normalize.depth = FALSE, ## no dimensionality reduction
  use.ods.genes = FALSE, ## no dimensionality reduction
  pca = FALSE, ## no dimensionality reduction
  k = 10,
  seed = 0,
```

```

    verbose = FALSE
  )

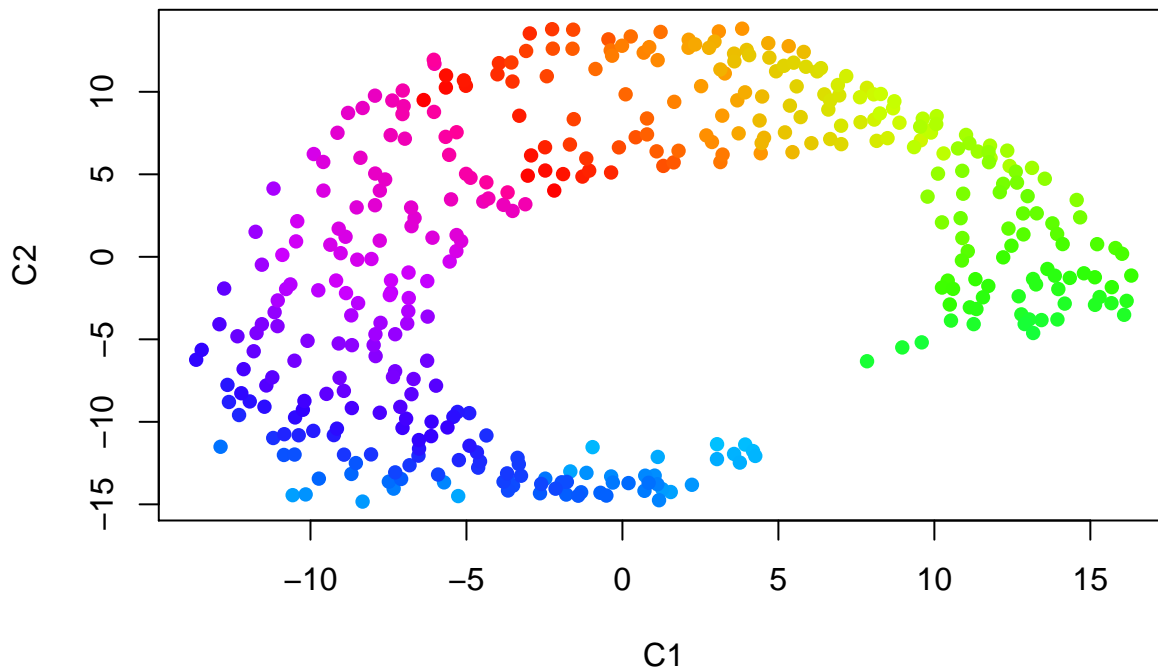
  ## Warning in if (!class(curr) %in% c("dgCMatrix", "dgTMatrix")) {: the condition
  ## has length > 1 and only the first element will be used

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

  ## [1] "Done finding neighbors"
  ## [1] "calculating weights"
  ## [1] "Done making graph"

  emb.veloviz = vig$fdg_coords
  plot(emb.veloviz, col=col, pch=16)

```



```

g = veloviz::plotVeloviz(vig, col=col, seed=0)

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

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

## Using provided colors...

```

