Cycle Simulation

Jean Fan

9/29/2020

Simulate cells along cycle

```
## make circle like cell cycle
par(mfrow=c(1,1))
x <- matrix(rnorm(200),nc=2)
y <- x/sqrt(rowSums(x^2))
## add some jitter
obs <- t(y)
obs <- jitter(obs, amount = 0.1)
## order points counterclockwise
angle <- atan2(obs[2,], obs[1,])
angle
```

```
##   [1] -2.13796320 -2.80041489 -1.19282811 -0.16523770  2.34229986 -0.91883201
##   [7]  3.00652607  2.68275322  0.68483664  0.71232286  0.07288291  0.99836205
##  [13]  1.74396379 -0.74312232  0.73919390  3.04758267 -1.51435036 -0.54398096
##  [19] -2.13760076 -0.62188364 -0.04107968  1.48658011 -0.69909399  0.06152424
##  [25]  0.68888219 -2.12477814  3.02378385 -3.05834134 -1.60684893  2.97913963
##  [31] -1.00707014  0.25634888 -2.93057386  2.43386539  0.75541036 -0.48440454
##  [37] -0.97633720  2.32271580  0.19559569 -0.57818092  2.54452358  0.41171069
##  [43] -0.47493732  0.98593502 -2.58255603 -2.00764832  0.02294714 -1.38079282
##  [49] -2.28733829 -2.89453834  1.09853281 -1.81604154 -1.17080622 -2.01339303
##  [55]  0.31974076  1.03416008 -0.63465009  1.73022107  2.57693305  1.97553185
##  [61]  3.11403357 -0.73850207  1.09607761 -0.47465082 -0.71245577 -0.38614148
##  [67] -2.53965970  1.37572580  2.80388590  0.30529487  2.54537896 -3.09745267
##  [73]  2.25192206 -1.14163112  1.94590690  0.10347722  1.24147828  2.80768089
##  [79]  0.24330921 -1.63527140 -0.95512245  2.71912945  1.19932423  1.66759753
##  [85] -1.02151789 -1.71881948  1.03718372 -2.89303395  2.31486206  0.06820375
##  [91]  1.90596185 -1.71247639  1.17983231  1.83498576 -2.27353189 -0.74338146
##  [97] -1.85290296  0.47352076  2.27775802 -0.73691813
```

```
## double check quandrants
angle[obs[2,]>0 & obs[1,]>0]
```

```
##  [1] 0.68483664 0.71232286 0.07288291 0.99836205 0.73919390 1.48658011
##  [7] 0.06152424 0.68888219 0.25634888 0.75541036 0.19559569 0.41171069
## [13] 0.98593502 0.02294714 1.09853281 0.31974076 1.03416008 1.09607761
## [19] 1.37572580 0.30529487 0.10347722 1.24147828 0.24330921 1.19932423
## [25] 1.03718372 0.06820375 1.17983231 0.47352076
```

```
angle[obs[2,]>0 & obs[1,]<0]
```

```
##  [1] 2.342300 3.006526 2.682753 1.743964 3.047583 3.023784 2.979140 2.433865
##  [9] 2.322716 2.544524 1.730221 2.576933 1.975532 3.114034 2.803886 2.545379
## [17] 2.251922 1.945907 2.807681 2.719129 1.667598 2.314862 1.905962 1.834986
```

```
## [25] 2.277758
```

```
angle[obs[2,]<0 & obs[1,]>0]
```

```
##  [1] -1.19282811 -0.16523770 -0.91883201 -0.74312232 -1.51435036 -0.54398096
##  [7] -0.62188364 -0.04107968 -0.69909399 -1.00707014 -0.48440454 -0.97633720
## [13] -0.57818092 -0.47493732 -1.38079282 -1.17080622 -0.63465009 -0.73850207
## [19] -0.47465082 -0.71245577 -0.38614148 -1.14163112 -0.95512245 -1.02151789
## [25] -0.74338146 -0.73691813
```

```
angle[obs[2,]<0 & obs[1,]<0]
```

```
##  [1] -2.137963 -2.800415 -2.137601 -2.124778 -3.058341 -1.606849 -2.930574
##  [8] -2.582556 -2.007648 -2.287338 -2.894538 -1.816042 -2.013393 -2.539660
## [15] -3.097453 -1.635271 -1.718819 -2.893034 -1.712476 -2.273532 -1.852903
```

```
angle[obs[2,]<0 & obs[1,]>0] = angle[obs[2,]<0 & obs[1,]>0] + 2*pi
angle[obs[2,]<0 & obs[1,]<0] = angle[obs[2,]<0 & obs[1,]<0] + 2*pi

angle[obs[2,]>0 & obs[1,]>0]
```

```
##  [1] 0.68483664 0.71232286 0.07288291 0.99836205 0.73919390 1.48658011
##  [7] 0.06152424 0.68888219 0.25634888 0.75541036 0.19559569 0.41171069
## [13] 0.98593502 0.02294714 1.09853281 0.31974076 1.03416008 1.09607761
## [19] 1.37572580 0.30529487 0.10347722 1.24147828 0.24330921 1.19932423
## [25] 1.03718372 0.06820375 1.17983231 0.47352076
```

```
angle[obs[2,]>0 & obs[1,]<0]
```

```
##  [1] 2.342300 3.006526 2.682753 1.743964 3.047583 3.023784 2.979140 2.433865
##  [9] 2.322716 2.544524 1.730221 2.576933 1.975532 3.114034 2.803886 2.545379
## [17] 2.251922 1.945907 2.807681 2.719129 1.667598 2.314862 1.905962 1.834986
## [25] 2.277758
```

```
angle[obs[2,]<0 & obs[1,]>0]
```

```
##  [1] 5.090357 6.117948 5.364353 5.540063 4.768835 5.739204 5.661302 6.242106
##  [9] 5.584091 5.276115 5.798781 5.306848 5.705004 5.808248 4.902392 5.112379
## [17] 5.648535 5.544683 5.808534 5.570730 5.897044 5.141554 5.328063 5.261667
## [25] 5.539804 5.546267
```

```
angle[obs[2,]<0 & obs[1,]<0]
```

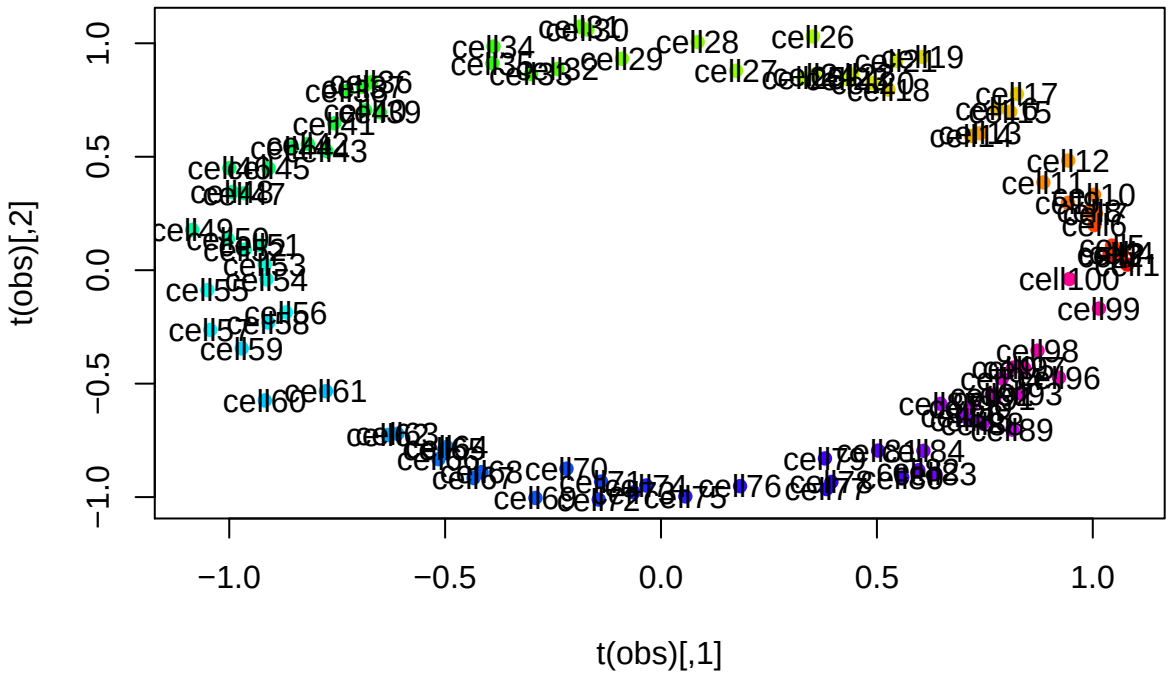
```
##  [1] 4.145222 3.482770 4.145585 4.158407 3.224844 4.676336 3.352611 3.700629
##  [9] 4.275537 3.995847 3.388647 4.467144 4.269792 3.743526 3.185733 4.647914
## [17] 4.564366 3.390151 4.570709 4.009653 4.430282
```

```
obs <- obs[, order(angle)]
obs
```

```
##            [,1]       [,2]      [,3]       [,4]      [,5]      [,6]      [,7]
## [1,] 1.07991549 1.04197258 1.0388451 1.06728989 1.044544 1.0038350 1.0070088
## [2,] 0.02478532 0.06418758 0.0709632 0.07792522 0.108474 0.1988886 0.2499667
##           [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
## [1,] 0.9922625 0.9423432 1.0029624 0.8855387 0.9431774 0.7391214 0.7185962
## [2,] 0.2600877 0.2969770 0.3320826 0.3866851 0.4832870 0.6036348 0.5917342
##          [,15]     [,16]     [,17]     [,18]     [,19]     [,20]     [,21]
## [1,] 0.8088058 0.7782054 0.8244426 0.5269550 0.6055551 0.4902049 0.5448508
## [2,] 0.6984651 0.7094217 0.7764218 0.7958378 0.9397071 0.8240589 0.9222565
```

```
##          [,22]     [,23]     [,24]    [,25]     [,26]     [,27]      [,28]
## [1,] 0.4432554 0.4301609 0.3550293 0.329671 0.3519277 0.1736581 0.08487828
## [2,] 0.8625047 0.8421042 0.8613405 0.846270 1.0297417 0.8789116 1.00547762
##           [,29]      [,30]      [,31]      [,32]      [,33]      [,34]
## [1,] -0.09049002 -0.1705762 -0.1877262 -0.2398816 -0.3009518 -0.3879847
## [2,]  0.93188096  1.0608679  1.0732156  0.8867671  0.8640424  0.9853472
##           [,35]      [,36]      [,37]      [,38]      [,39]      [,40]
## [1,] -0.3905551 -0.6711926 -0.6899634 -0.7278627 -0.6510634 -0.6851675
## [2,]  0.9116887  0.8280961  0.8076743  0.7906628  0.6961844  0.7044774
##           [,41]      [,42]      [,43]      [,44]      [,45]      [,46]
## [1,] -0.7567260 -0.8179983 -0.7752009 -0.8545050 -0.9090542 -0.9997088
## [2,]  0.6474431  0.5561102  0.5260457  0.5412945  0.4490765  0.4493993
##           [,47]      [,48]      [,49]      [,50]      [,51]       [,52]
## [1,] -0.9651253 -0.9933270 -1.0858473 -1.0033286 -0.9282462 -0.96388990
## [2,]  0.3389122  0.3445866  0.1779675  0.1363463  0.1098643  0.09088317
##           [,53]       [,54]       [,55]      [,56]      [,57]       [,58]
## [1,] -0.9178893 -0.91367801 -1.05061975 -0.8688079 -1.0438891 -0.9096332
## [2,]  0.0253026 -0.04035595 -0.08766811 -0.1861054 -0.2632756 -0.2308715
##           [,59]      [,60]      [,61]      [,62]      [,63]      [,64]
## [1,] -0.9710318 -0.9176869 -0.7762140 -0.6324866 -0.6106744 -0.4961960
## [2,] -0.3447771 -0.5741126 -0.5332421 -0.7261880 -0.7210060 -0.7789836
##           [,65]      [,66]      [,67]      [,68]      [,69]      [,70]
## [1,] -0.5020714 -0.5151993 -0.4341501 -0.4155903 -0.2909547 -0.2189669
## [2,] -0.7888383 -0.8328510 -0.9160123 -0.8900284 -1.0038581 -0.8748763
##           [,71]      [,72]      [,73]       [,74]       [,75]      [,76]
## [1,] -0.1389026 -0.1441964 -0.0632157 -0.03422589  0.05633096  0.1830697
## [2,] -0.9315209 -1.0109414 -0.9791082 -0.94892070 -0.99690252 -0.9518841
##           [,77]      [,78]      [,79]      [,80]      [,81]      [,82]
## [1,]  0.3829827  0.3945632  0.3796597  0.5588634  0.5026469  0.5959684
## [2,] -0.9645493 -0.9332555 -0.8296558 -0.9130072 -0.7951355 -0.8815668
##           [,83]      [,84]      [,85]      [,86]      [,87]      [,88]
## [1,]  0.6354267  0.6077231  0.7190471  0.6976975  0.6471024  0.7443876
## [2,] -0.8982590 -0.7961697 -0.6610268 -0.6410663 -0.5890874 -0.6754976
##           [,89]      [,90]      [,91]      [,92]      [,93]      [,94]
## [1,]  0.8136483  0.7166405  0.7718881  0.7660051  0.8343218  0.7895556
## [2,] -0.7028358 -0.6025089 -0.5683112 -0.5490391 -0.5444547 -0.4775658
##           [,95]      [,96]      [,97]      [,98]      [,99]      [,100]
## [1,]  0.8157181  0.9218923  0.8414201  0.8721174  1.0142970  0.94572053
## [2,] -0.4292489 -0.4740304 -0.4323475 -0.3545607 -0.1691423 -0.03887177
```

```r
## rainbow
col = colorRampPalette(c(rainbow(10)))(ncol(obs))
labels <- paste0('cell', 1:ncol(obs))
## plot
plot(t(obs),col=col, pch=16)
text(t(obs), labels)
```



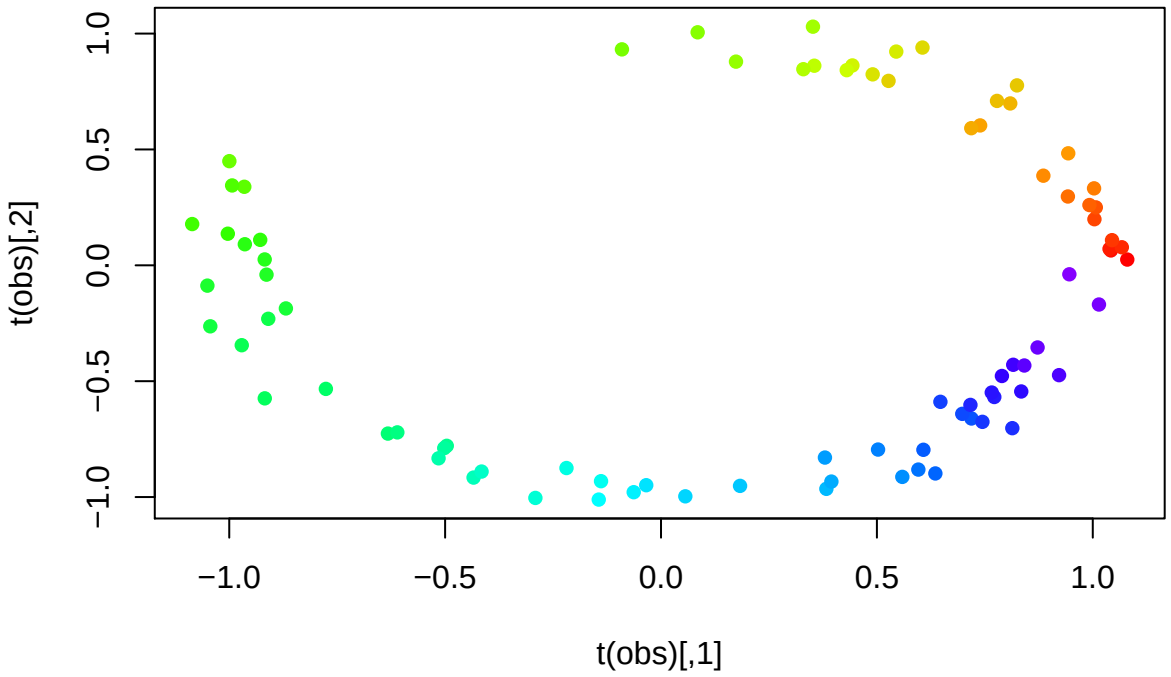

```
colnames(obs) <- labels
```

Remove some transient cells

```
cells.keep <- setdiff(labels, paste0('cell', 30:45))
cells.keep
```

```
##  [1] "cell1"   "cell2"   "cell3"   "cell4"   "cell5"   "cell6"   "cell7"
##  [8] "cell8"   "cell9"   "cell10"  "cell11"  "cell12"  "cell13"  "cell14"
## [15] "cell15"  "cell16"  "cell17"  "cell18"  "cell19"  "cell20"  "cell21"
## [22] "cell22"  "cell23"  "cell24"  "cell25"  "cell26"  "cell27"  "cell28"
## [29] "cell29"  "cell46"  "cell47"  "cell48"  "cell49"  "cell50"  "cell51"
## [36] "cell52"  "cell53"  "cell54"  "cell55"  "cell56"  "cell57"  "cell58"
## [43] "cell59"  "cell60"  "cell61"  "cell62"  "cell63"  "cell64"  "cell65"
## [50] "cell66"  "cell67"  "cell68"  "cell69"  "cell70"  "cell71"  "cell72"
## [57] "cell73"  "cell74"  "cell75"  "cell76"  "cell77"  "cell78"  "cell79"
## [64] "cell80"  "cell81"  "cell82"  "cell83"  "cell84"  "cell85"  "cell86"
## [71] "cell87"  "cell88"  "cell89"  "cell90"  "cell91"  "cell92"  "cell93"
## [78] "cell94"  "cell95"  "cell96"  "cell97"  "cell98"  "cell99"  "cell100"
```

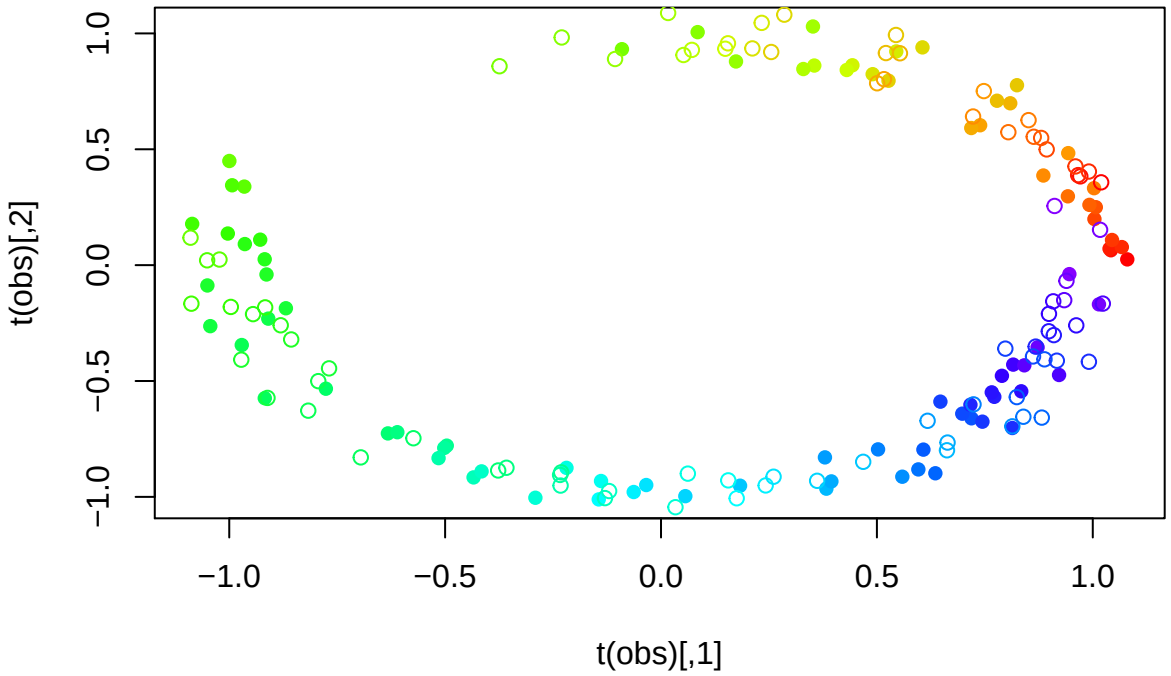
```
labels <- labels[cells.keep]
obs <- obs[,cells.keep]
plot(t(obs),col=col, pch=16)
text(t(obs), labels)
```



Simulate lower dimensional representation of future transcriptional state

```
## rotate circle slightly
f = pi*0.1 # adjust as needed
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)

plot(t(obs),col=col, pch=16)
points(t(exp),col=col)
```



```r
colnames(exp) <- labels
```

Try different embeddings

```r
k = 10

## Lyla's FDG
library(igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```r
library(matie)
```

```
## Registered S3 method overwritten by 'seriation':
##   method         from
##   reorder.hclust gclus
```

```
## Registered S3 methods overwritten by 'proxy':
##   method               from
##   print.registry_field registry
##   print.registry_entry registry
```

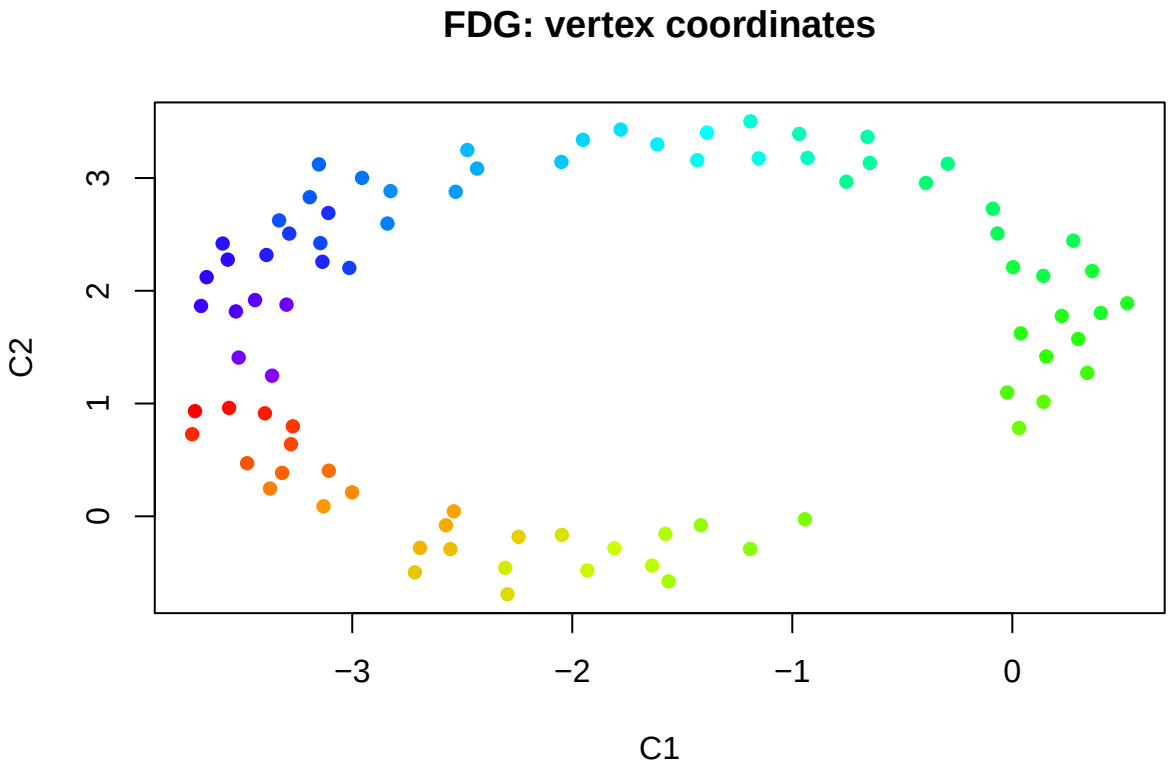
```r
library(RANN)
source('../graphViz/projectedNeighbors.R')
gsim = graphViz(obs, exp, k, cell.colors=col, weighted=TRUE, plot = FALSE, return_graph = TRUE)
```

```
## [1] "Done finding neighbors"
```

```
## Warning in vattrs[[name]][index] <- value: number of items to replace is not a
## multiple of replacement length
```

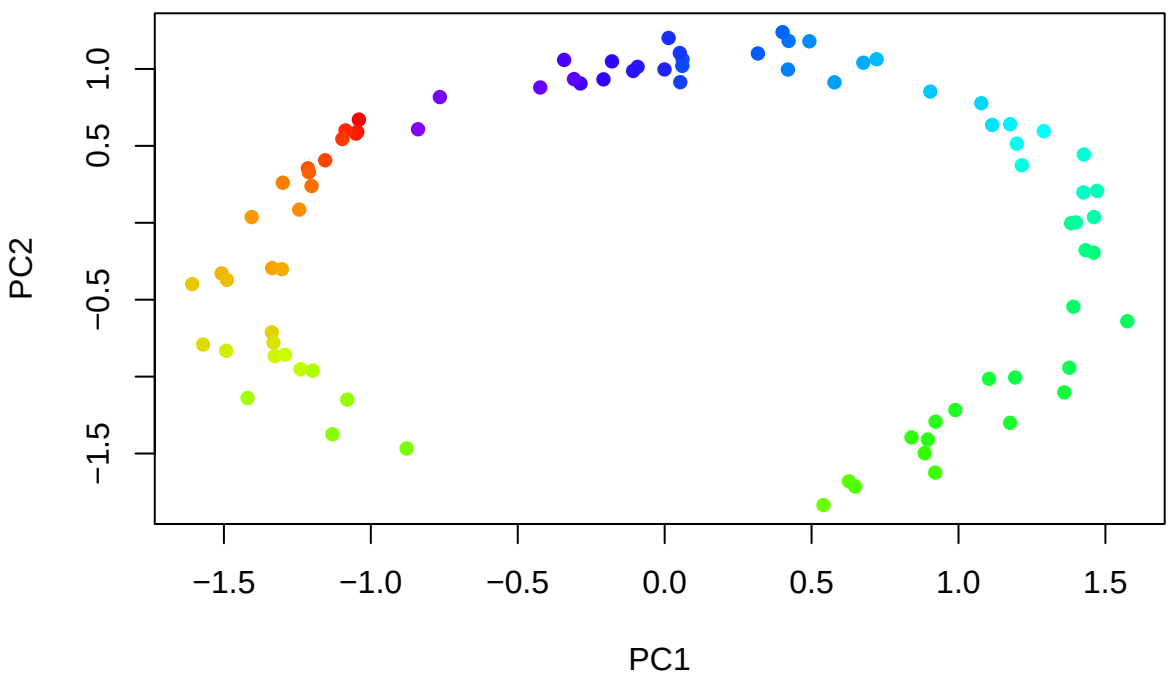
```
## [1] "Done making graph"
```

```r
plot(gsim$fdg_coords, main = "FDG: vertex coordinates", col=col, pch=16)
```



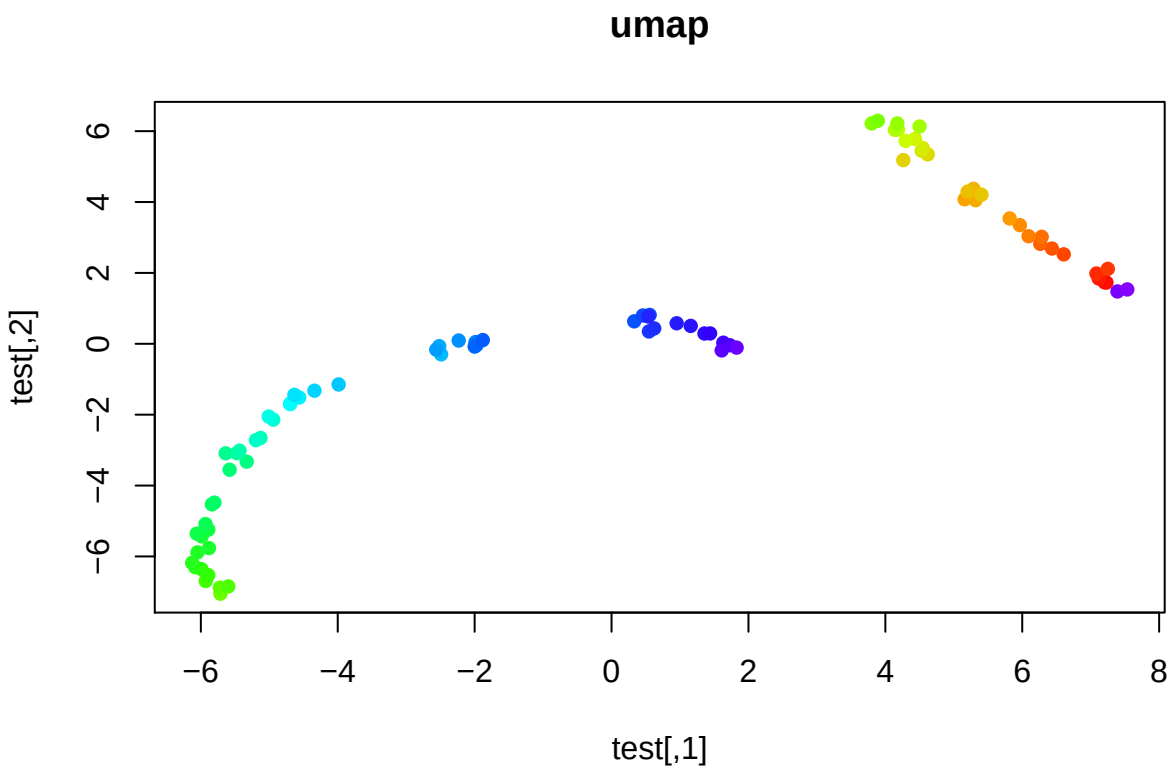

```
#text(gsim$fdg_coords+0.1, labels = labels)
```

```
## PCA
test <- prcomp(t(obs), scale=TRUE, center=TRUE)
test <- test$x
plot(test, main = "pca", col=col, pch=16)
```



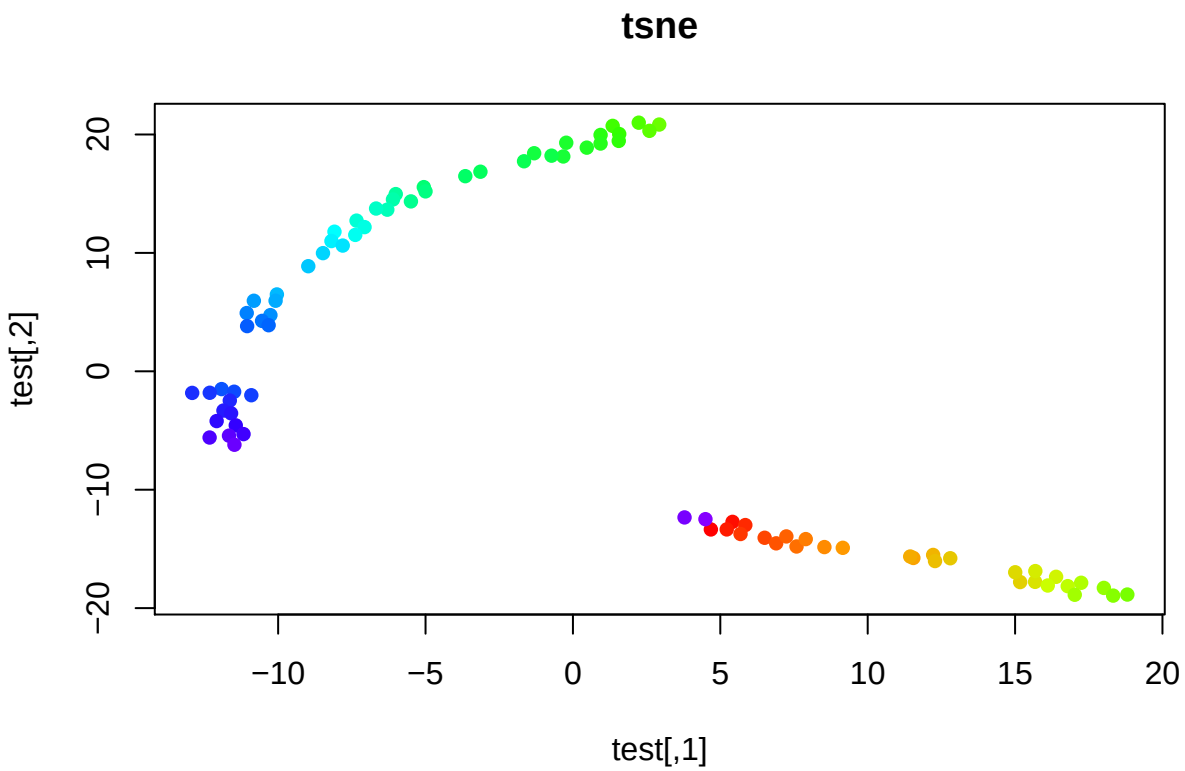
```
#text(test+0.1, labels=labels)
```

```
## Umap
test <- uwot::umap(t(obs), n_neighbors = k, min=0.1)
plot(test, main = "umap", col=col, pch=16)
```



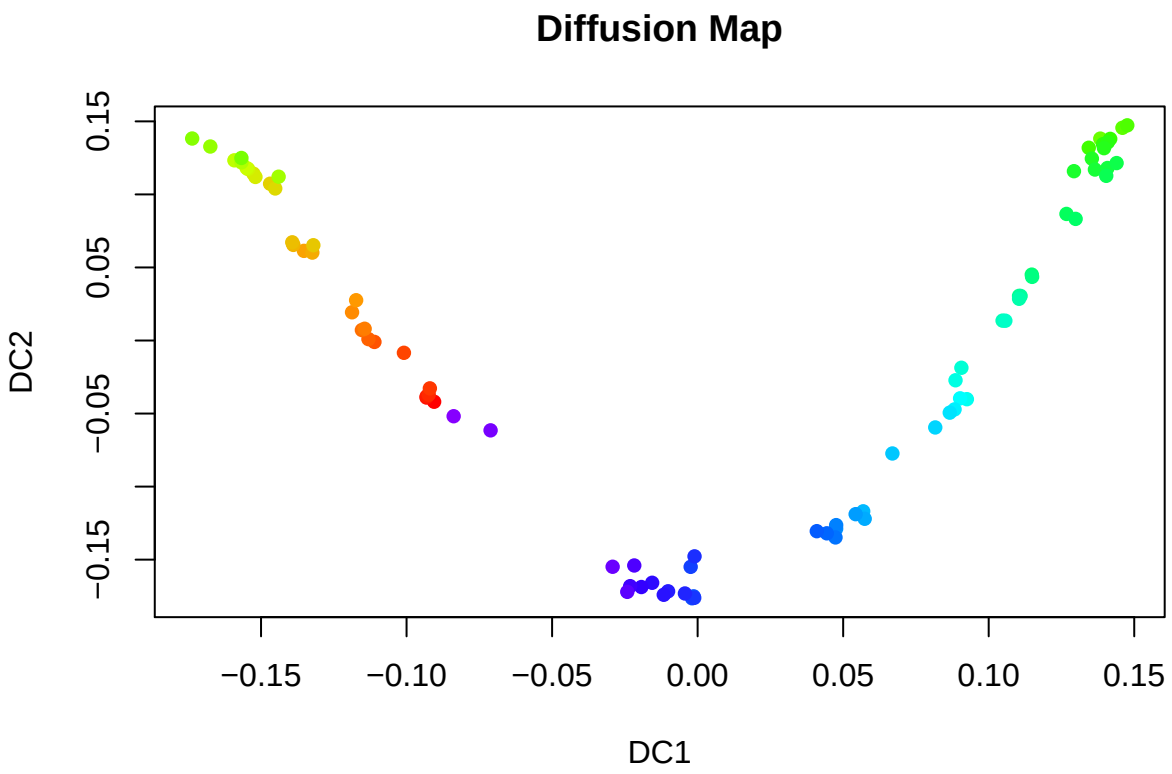

```
#text(test+0.1, labels=labels)
```

```
## tSNE
test <- Rtsne::Rtsne(t(obs), perplexity=k)$Y
plot(test, main = "tsne", col=col, pch=16)
```



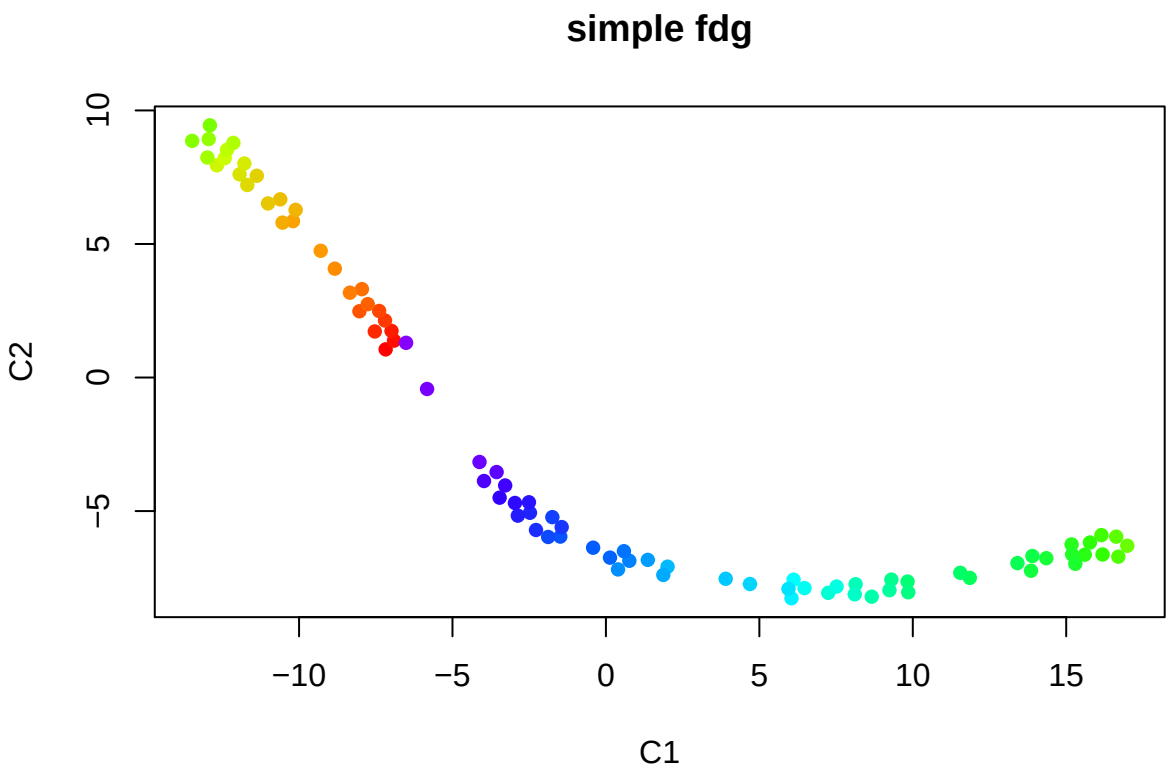

```
#text(test+0.1, labels=labels)
```

```
## diffusion map
test <- destiny::DiffusionMap(t(obs))
test <- test@eigenvectors[,1:2]
plot(test, main = "Diffusion Map", col=col, pch=16)
```




```
#text(test+0.1, labels=labels)
```

```
## Regular FDG on observed only
nn = RANN::nn2(t(obs), k = k) ## KNN
names(nn) <- c('idx', 'dists')
weight <- 1/(1+ as.vector(nn$dists))
nn.df = data.frame(from = rep(1:nrow(nn$idx), k),
                   to = as.vector(nn$idx),
                   weight = weight
)
g <- igraph::graph_from_data_frame(nn.df, directed = FALSE)
g <- igraph::simplify(g)
fdg = layout_with_fr(g,dim=2)
colnames(fdg) = c("C1","C2")
rownames(fdg) <- labels
plot(fdg, main = "simple fdg", col=col, pch=16)
```




```
#text(fdg+0.1, labels = labels)
```