Small RNA dataset

Luis Ángel Rodríguez García

25-05-2022

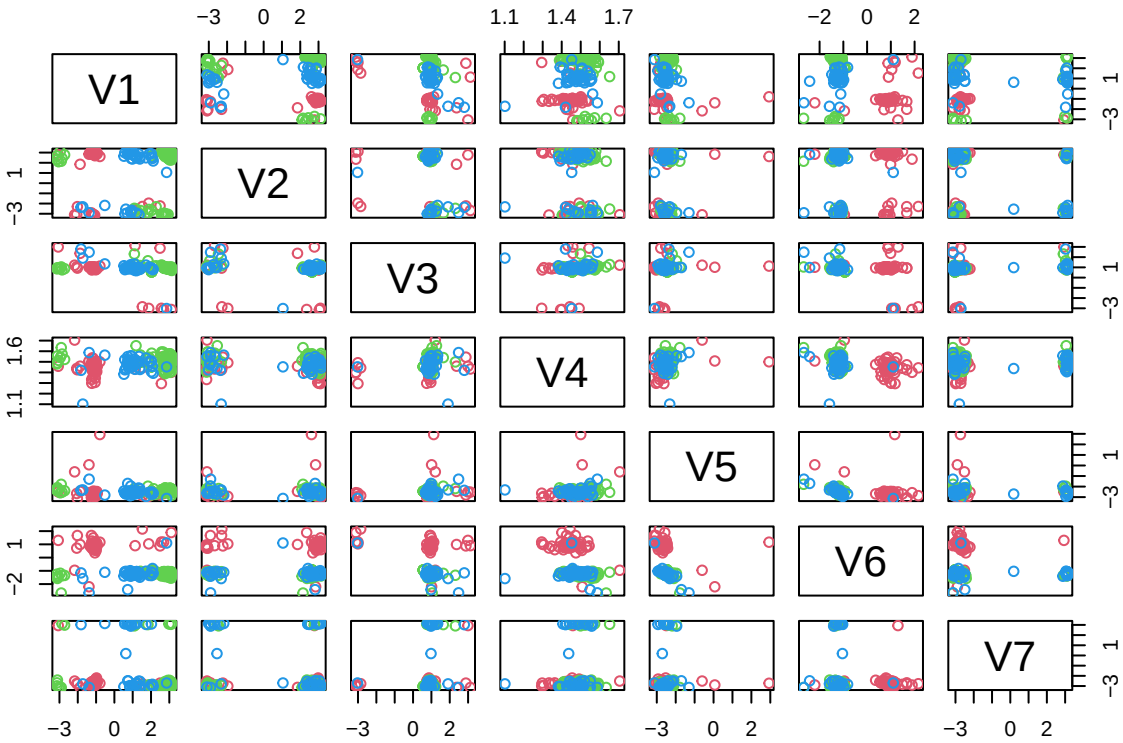
The objective in this case study is to recover clusters identified in `smallrna$clusters` using only the information on `smallrna$angles`, a 7-dimensional matrix of angles (i.e., data on $(\mathbb{S}^1)^7$)). The clusters have been constructed using the information in `smallrna$torsion`. If a dimension-reduction technique is able to successfully identify clusters, then it will be doing a good job in terms of identifying the underlying structure of the data. Section 5.2 in Zouboulouglou et al. (2021) describes the history of the "Small RNA" dataset and its construction.

Let's begin by importing the data.

```
# Load package
library(pscsne)
stopifnot(packageVersion("pscsne") >= "0.0.1.900001")

# Load dataset
data("smallrna")

# Visualize original data
pairs(smallrna$angles, col = smallrna$clusters + 1)
```



We can now run psc-SNE. First, we transform the data and obtain the ρ's giving the prescribed perplexity.

```
# Data to Cartesian coordinates
smallrna_X <- sphunif::Theta_to_X(Theta = smallrna$angles)

# Obtain rhos for given perplexity
rho_psc_list <- rho_optim_bst(x = smallrna_X, perp_fixed = 30)
```

```
## Time difference of 10.45816 secs
```

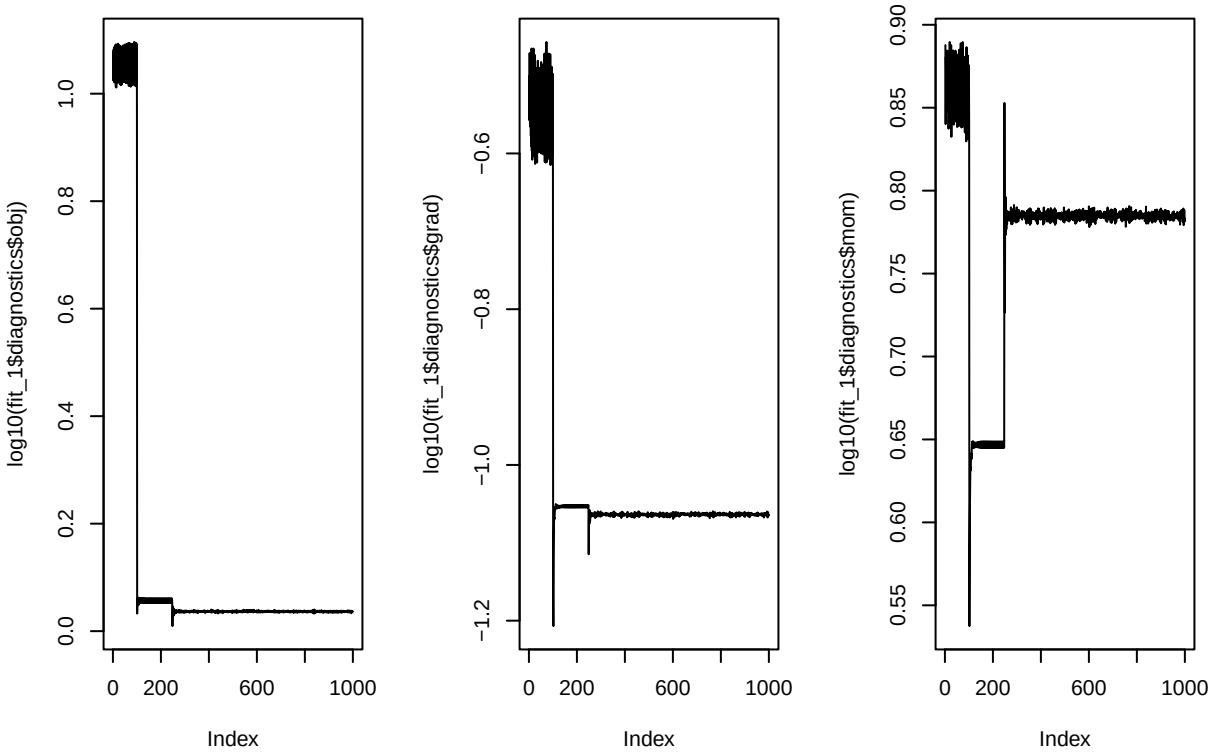
We run psc-SNE for $d = 1$ with its default η.

```
# Default
void_1 <- capture.output(
  fit_1 <- psc_sne(X = smallrna_X, d = 1, rho_psc_list = rho_psc_list,
                   eta = 200, num_iteration = 1e3, tol = 1e-6,
                   show_prog = FALSE, colors = smallrna$clusters))

# Does not converge
fit_1$convergence
```

```
## [1] FALSE
```

```
par(mfrow = c(1, 3))
plot(log10(fit_1$diagnostics$obj), type = "l")
plot(log10(fit_1$diagnostics$grad), type = "l")
plot(log10(fit_1$diagnostics$mom), type = "l")
```



The employed η seems to be too large. Let's reduce it.

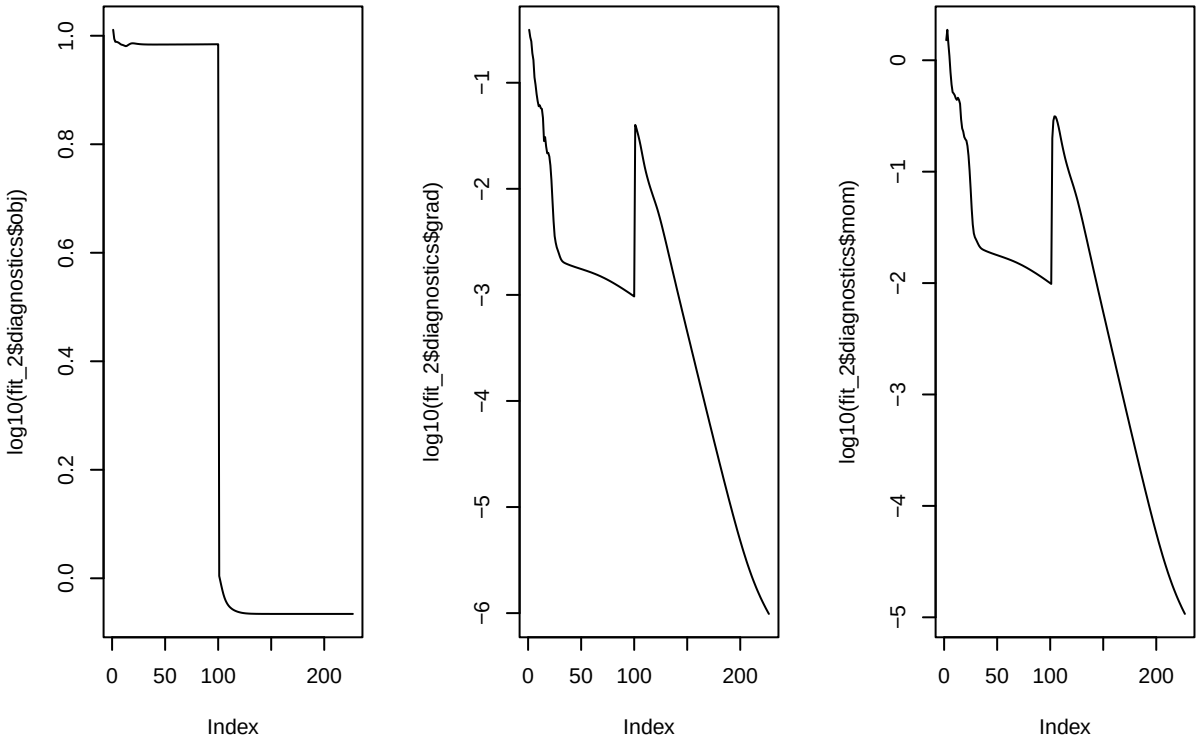
```
# Lower eta
void_2 <- capture.output(
  fit_2 <- psc_sne(X = smallrna_X, d = 1, rho_psc_list = rho_psc_list,
                   eta = 10, num_iteration = 1e3, tol = 1e-6,
```

```
                    show_prog = FALSE, colors = smallrna$clusters))

# Converges
fit_2$convergence
```

```
## [1] TRUE
```

```
par(mfrow = c(1, 3))
plot(log10(fit_2$diagnostics$obj), type = "l")
plot(log10(fit_2$diagnostics$grad), type = "l")
plot(log10(fit_2$diagnostics$mom), type = "l")
```



Convergence is attained in the second run, yet it is weird that the objective function takes exactly the zero value.

Let's see the recovery of the clusters.

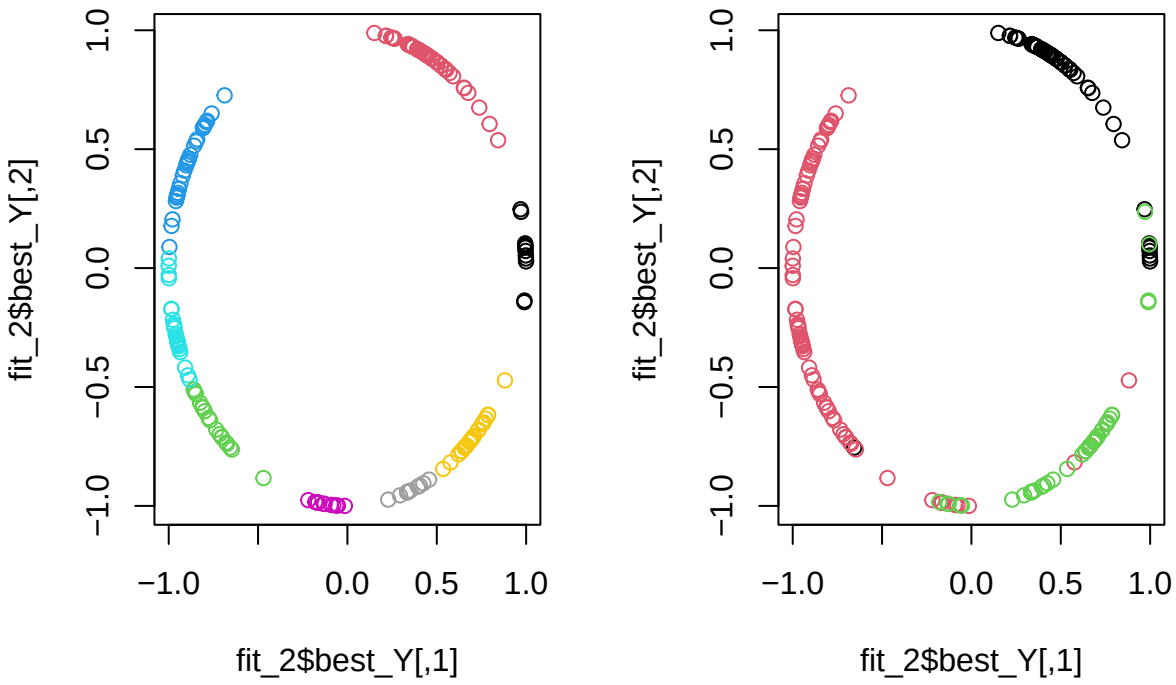
```
# Kernel mean shift clustering
n <- nrow(fit_2$best_Y)
d <- ncol(fit_2$best_Y) - 1
fit_mix <- DirStats::bic_vmf_mix(fit_2$best_Y, kappa_max = 1e3)
h <- DirStats::bw_dir_emi(data = fit_2$best_Y, fit_mix = fit_mix)$h_opt *
      n^(1 / (d + 4)) * n^(-1 / (d + 6))
kms <- kms_dir(x = fit_2$best_Y, data = fit_2$best_Y, h = h)
```

```
##   |                                                                      |
```

```
# Detects 8 clusters by splitting the 3 original clusters
length(unique(kms$cluster))
```

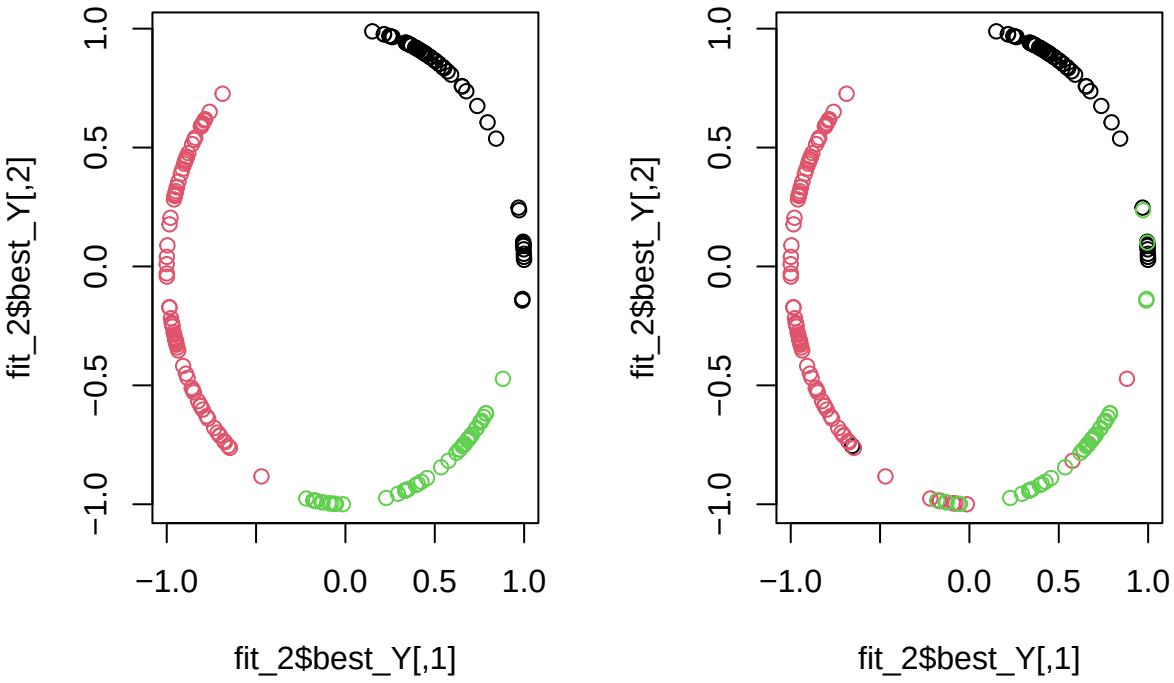
```
## [1] 8
```

```
# Fully-automatically recovered clusters vs. real clusters
par(mfrow = c(1, 2))
plot(fit_2$best_Y, col = kms$cluster)
plot(fit_2$best_Y, col = smallrna$clusters)
```



The original clusters are not fully recovered, in the sense that more clusters are obtained. However, the three-cluster structure is present, as the new clusters appear dividing the three main ones. This can be checked by cutting the hierarchical clustering tree behind kernel mean shift clustering exactly at three groups. Or, in other words, by merging the 8 groups into 3.

```
# Recovered clusters with three clusters vs. real clusters
par(mfrow = c(1, 2))
labels <- cutree(kms$tree, k = 3)
plot(fit_2$best_Y, col = labels)
plot(fit_2$best_Y, col = smallrna$clusters)
```




```
# Correct classification rate: 90%
mean(labels == smallrna$clusters)
```

```
## [1] 0.9
```

```
# 19 incorrectly classified observations
sum(labels != smallrna$clusters)
```

```
## [1] 19
```

The classification accuracy is on-par with Zouboulouglou et al. (2021), which misclassifies 16 points and has a classification rate of 0.916.