Gut Baby

Contents

Preliminary

```
library(tidyverse)
library(mixOmics)
walk(dir("../Rscript/", pattern = ".R$", full.names = TRUE),source)
```

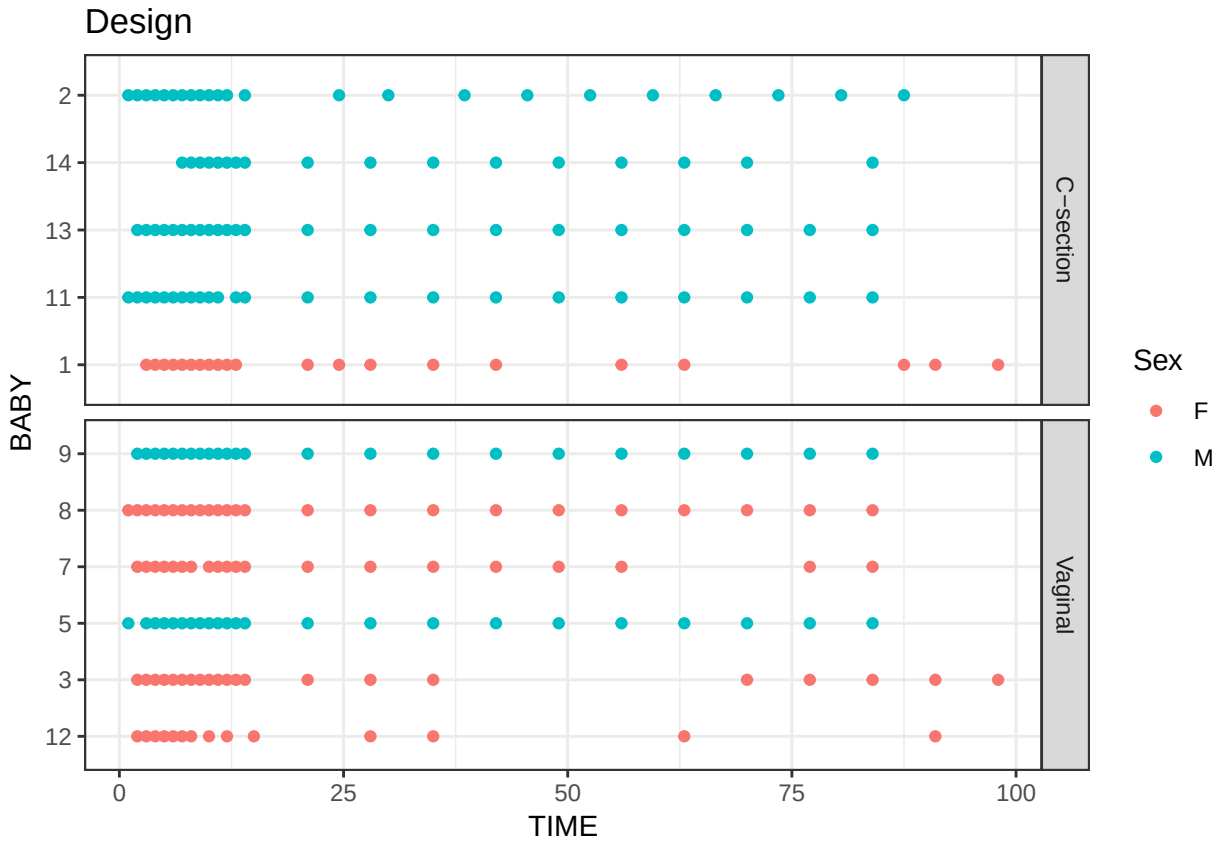
Data Description & Design

Original paper (Development of the Human Infant Intestinal Microbiota, Palmer et al. 2007) studied gastrointestinal microbiome of 14 babies during the first year of life. They collected an average of 26 stool samples from 14 healthy full-term human infants. They have also included vagina and milk microbiome composition from the mothers and stool samples from mothers and fathers.

For demonstration purposes and because babies' gut almost reach an "adult-like" composition, we have focused our attention on the first 100 days of life. We also excluded babies who recieved an antibiotic treatment during that period, because antibiotics can change drastically microbiome composition.

Our final design consists in an average of 21 timepoits for each of the 11 selected babies.

```
load("../Data/milk_data.RData")
ggplot(data= design %>% rename(Sex = Sexe), aes(x = TIME, y = BABY, color = Sex)) +
  geom_point() + facet_grid(Delivery~., scales = "free_y") + ggtitle("Design") +
  #scale_color_manual(values = color.mixo(1:2))  +
  theme_bw()
```




```
design %>% dplyr::select(BABY, TIME) %>% mutate(BABY = as.numeric(BABY)) %>%
  group_by(BABY) %>%summarise(n_timepoints = n()) %>% knitr::kable()
```

BABY	n_timepoints
1	21
2	23
3	21
5	23
7	20
8	24
9	23
11	23
12	14
13	23
14	17

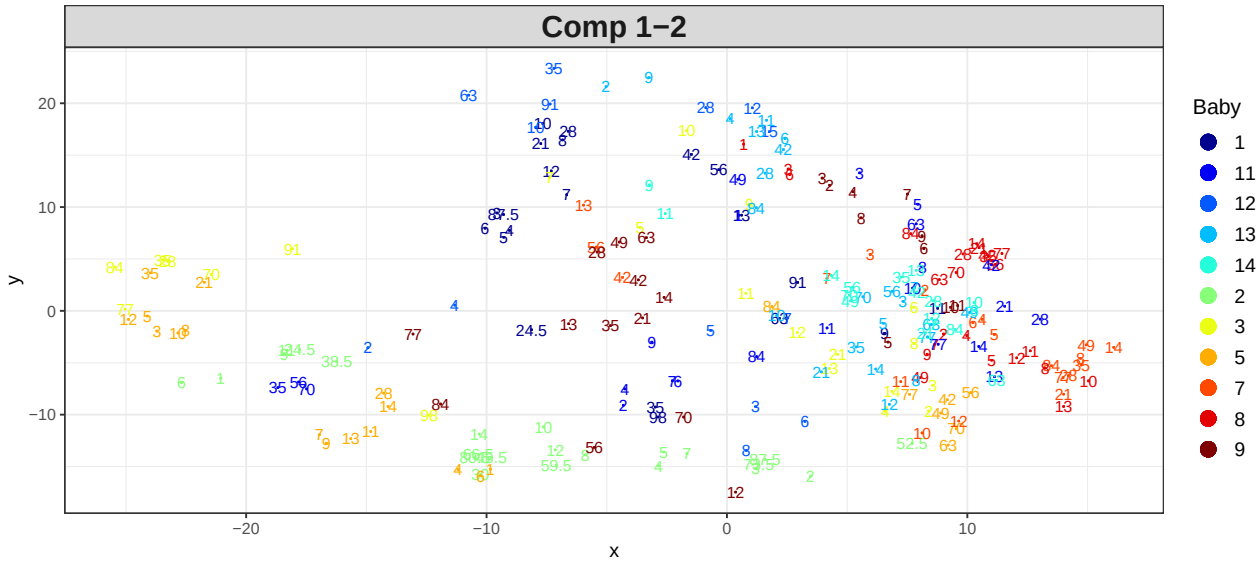
Analysis

Pre-processing

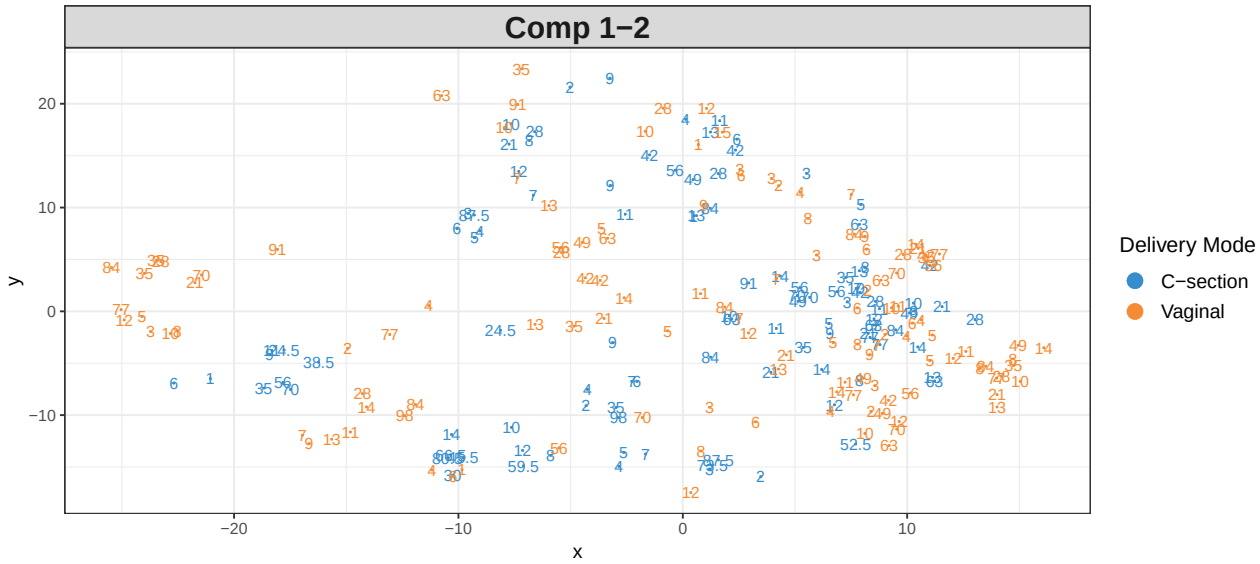
We perform standard pre-processing steps :

- Low Count Removal
- Total Sum Scalling
- Centered Log Ratio Transformation

```
OTU_norm <- norm_OTU(OTU, AR = T)
# option AR(Abondance Relative) = data allready in AR.
# need to add a "pouillème" in order to cumpute CLR.

#pca(OTU_norm, ncomp = 10) %>% plot
pca.res <- pca(OTU_norm, ncomp = 4)

plotIndiv(pca.res, group = design$BABY, ind.names = design$TIME,
          comp = c(1,2), legend.title = "Baby", legend = T, title = "Comp 1-2")
```

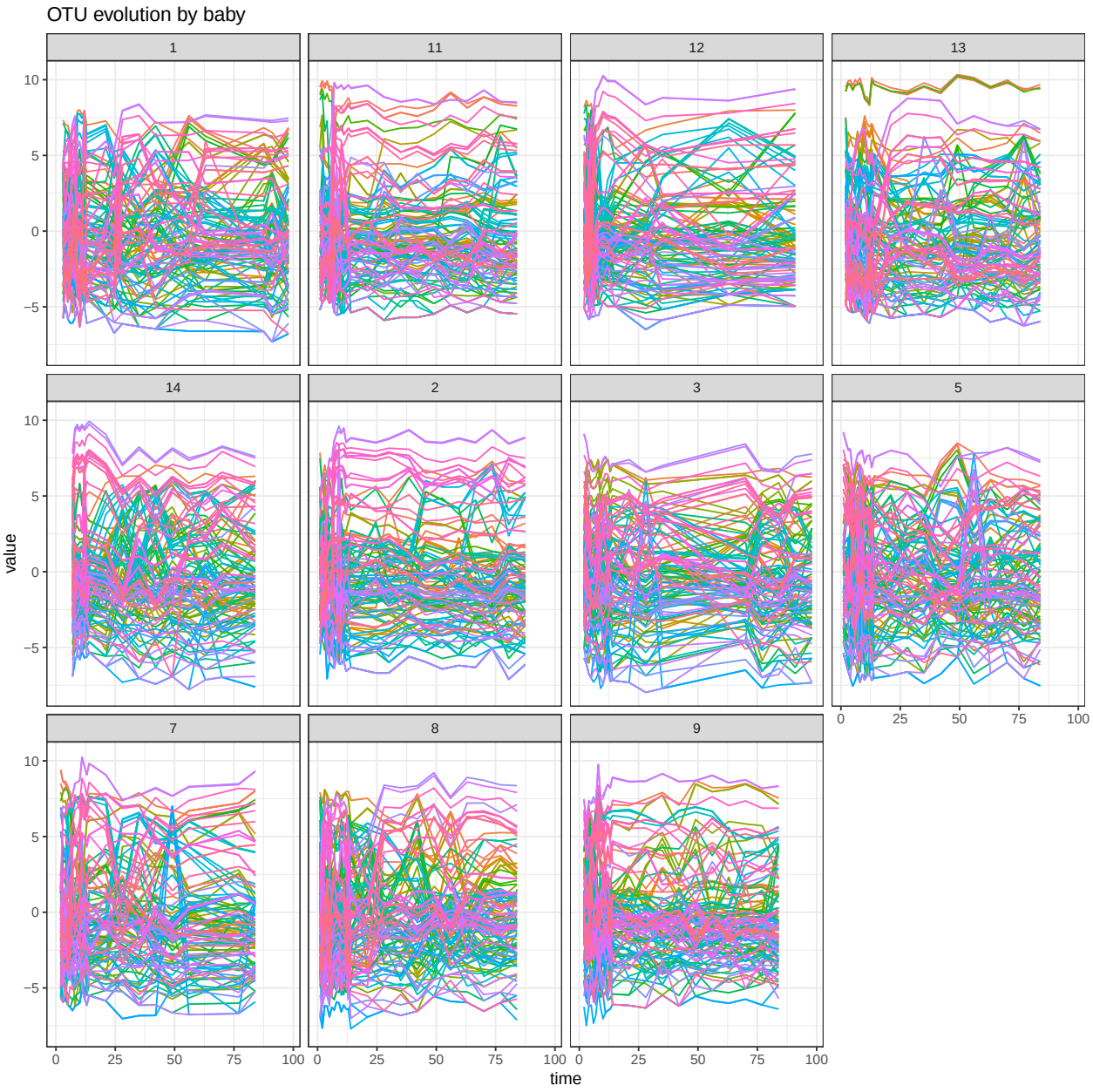



```
plotIndiv(pca.res, group = design$Delivery, ind.names = design$TIME,
          comp = c(1,2), legend.title = "Delivery Mode", legend = T, title = "Comp 1-2")
```




```
# per sample OTU evolution
OTU_norm %>% as.data.frame() %>% rownames_to_column("sample") %>%
  gather(OTU, value, -sample) %>%
```

```
  mutate(time = sample %>% str_split("_") %>% map_chr(~.x[2]) %>% as.numeric)%>%
  mutate(baby = sample %>% str_split("_") %>% map_chr(~.x[1])) %>%
  ggplot(aes(time, value, col=OTU)) + geom_line() + facet_wrap(~baby) + theme_bw() +
  theme(legend.position = "none") + ggtitle("OTU evolution by baby")
```



Split

```
delivery_mode <- design %>% dplyr::select(BABY, Delivery) %>% unique
index.C <- rownames(OTU) %>% str_split("_") %>% map_chr(~.x[1]) %in%
  (delivery_mode %>% filter(Delivery == "C-section") %>% pull(BABY))
OTU_norm.C <- OTU[index.C,] %>% norm_OTU(AR = T)
```

```
OTU_norm.V <- OTU[!index.C,] %>% norm_OTU(AR = T)
```

LMMS

C-section

```
time_lmms.C <- rownames(OTU_norm.C) %>% str_split("_") %>% map_chr(~.x[2]) %>% as.numeric
sample_id = rownames(OTU_norm.C)

# cubic p-spline
spline.MILK.C.cubicpspline = lmms::lmmSpline(data = OTU_norm.C, time = time_lmms.C,
                                        sampleID = sample_id,
                                        basis = 'cubic p-spline', keepModels = T,
                                        numCores = 2)

spline.MILK.C.pspline = lmms::lmmSpline(data = OTU_norm.C, time = time_lmms.C,
                                       sampleID = sample_id,
                                       basis = 'p-spline', keepModels = T,
                                       numCores = 2 )

spline.MILK.C.cubic = lmms::lmmSpline(data = OTU_norm.C, time = time_lmms.C,
                                     sampleID = sample_id,
                                     basis = 'cubic', keepModels = T,
                                     numCores = 2)
# summary
spline.MILK.C.cubicpspline@modelsUsed %>% table %>% as.data.frame() %>%
  set_names("ModelUsed", "Cubic P-spline") %>%
  left_join(spline.MILK.C.pspline@modelsUsed %>% table%>% as.data.frame() %>%
              set_names("ModelUsed", "P-spline")) %>%
  left_join(spline.MILK.C.cubic@modelsUsed %>% table%>% as.data.frame() %>%
              set_names("ModelUsed", "Cubic")) %>%
  knitr::kable()
```

ModelUsed	Cubic P-spline	P-spline	Cubic
0	95	78	82
1	12	29	25

Vaginal

```
time_lmms.V <- rownames(OTU_norm.V) %>% str_split("_") %>% map_chr(~.x[2]) %>% as.numeric
sample_id = rownames(OTU_norm.V)

# cubic p-spline
spline.MILK.V.cubicpspline = lmms::lmmSpline(data = OTU_norm.V, time = time_lmms.V,
                                        sampleID = sample_id,
                                        basis = 'cubic p-spline', keepModels = T,
                                        numCores = 2)

spline.MILK.V.pspline = lmms::lmmSpline(data = OTU_norm.V, time = time_lmms.V,
```

```
                                    sampleID = sample_id,
                                    basis = 'p-spline', keepModels = T,
                                    numCores = 2 )

spline.MILK.V.cubic = lmms::lmmSpline(data = OTU_norm.V, time = time_lmms.V,
                                    sampleID = sample_id,
                                    basis = 'cubic', keepModels = T,
                                    numCores = 2)
# summary
spline.MILK.V.cubicpspline@modelsUsed %>% table %>% as.data.frame() %>%
  set_names("ModelUsed", "Cubic P-spline") %>%
  left_join(spline.MILK.V.pspline@modelsUsed %>% table%>% as.data.frame() %>%
              set_names("ModelUsed", "P-spline")) %>%
  left_join(spline.MILK.V.cubic@modelsUsed %>% table%>% as.data.frame() %>%
              set_names("ModelUsed", "Cubic")) %>%
  knitr::kable()
```

ModelUsed	Cubic P-spline	P-spline	Cubic
0	107	95	98
1	10	22	19

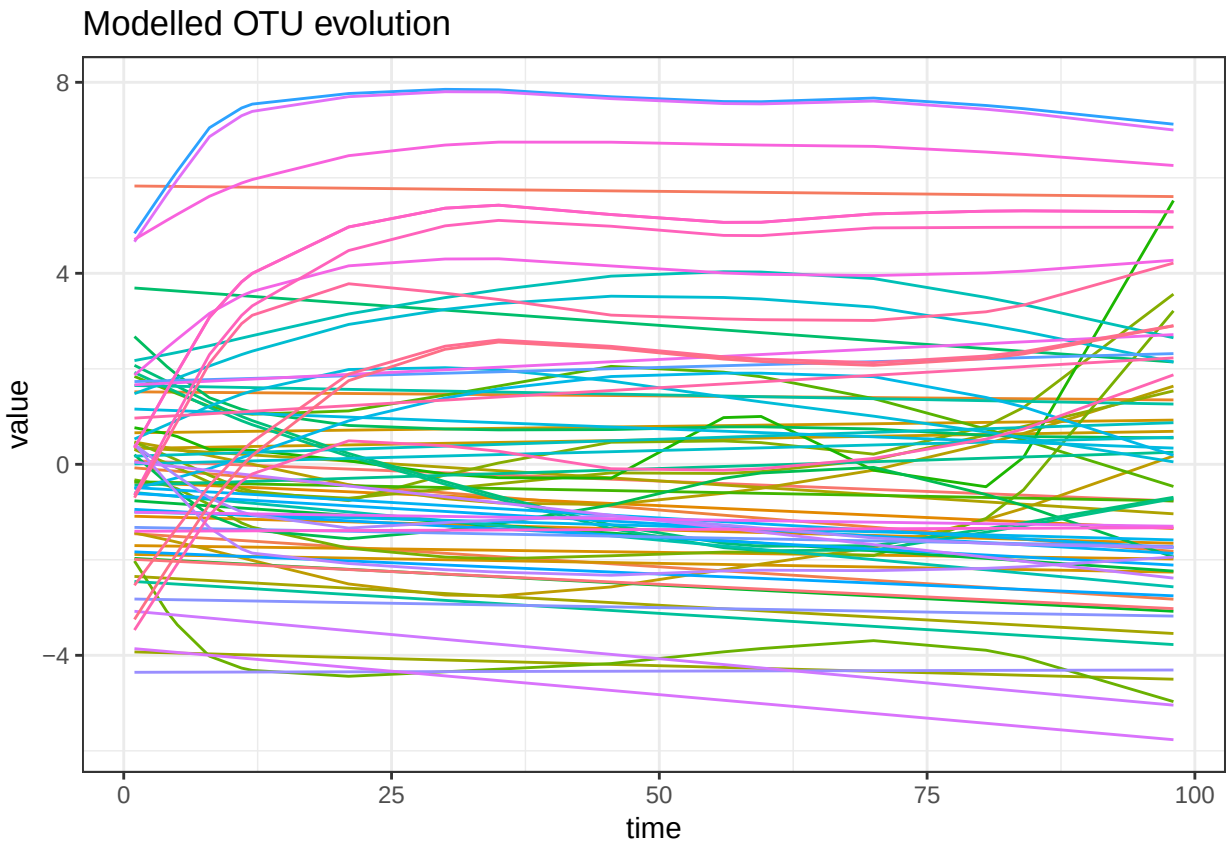
More straight lines in every delivery mode. Less in pspline.

Filter

C-section

```
filter.spline.C.res <- wrapper.filter.splines(OTU_norm.C, spline.MILK.C.pspline)
index.filter.C <- (rownames(spline.MILK.C.pspline@predSpline) %in% filter.spline.C.res$to_keep) %>%
  which
```

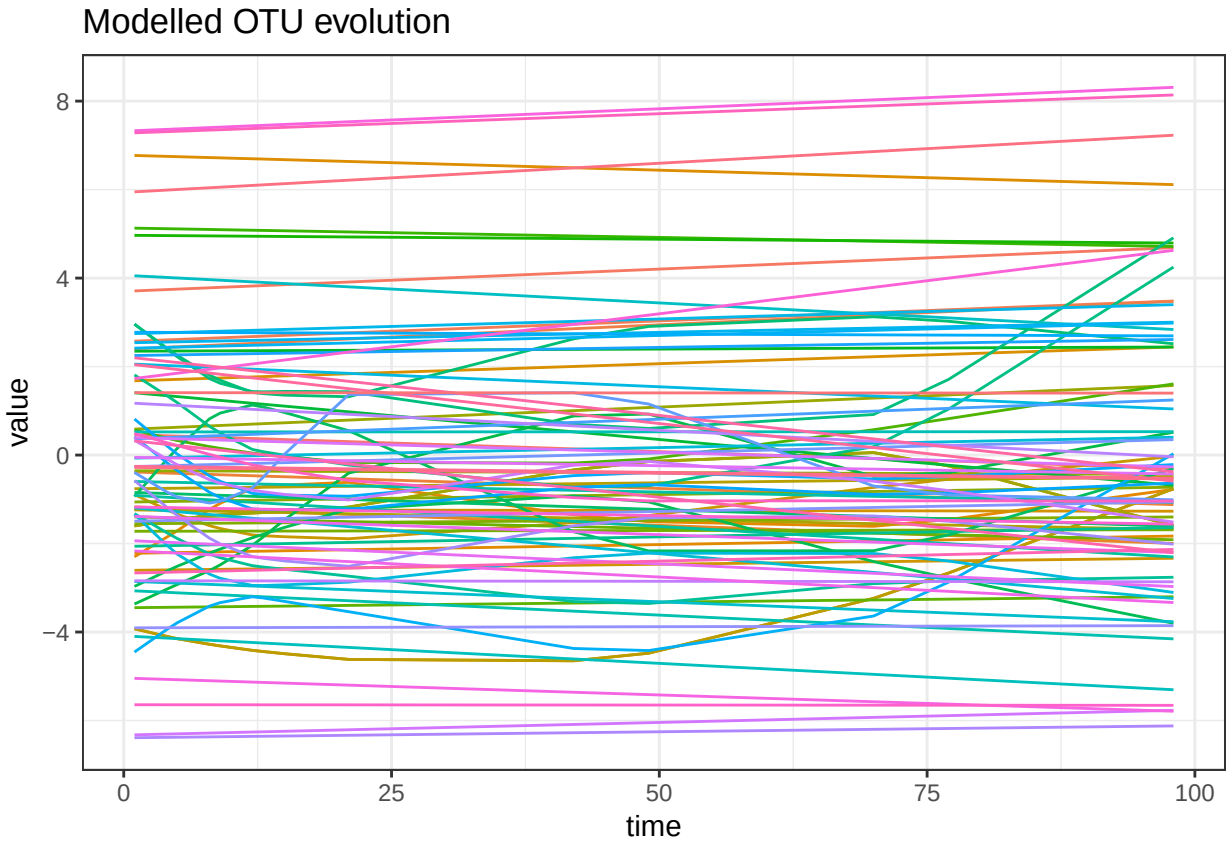
```
spline.data.C <- spline.MILK.C.pspline@predSpline[index.filter.C,] %>% t %>% as.data.frame()
spline.data.C %>% rownames_to_column("time") %>%
  gather(Features, value, - time) %>% mutate(time =as.numeric(time)) %>%
  ggplot(aes(x=time, y = value, col = Features)) + geom_line() + theme_bw() +
  theme(legend.position = "none") + ggtitle("Modelled OTU evolution")
```



Vaginal

```
filter.spline.V.res <- wrapper.filter.splines(OTU_norm.V, spline.MILK.V.pspline)
index.filter.V <- (rownames(spline.MILK.V.pspline@predSpline) %in% filter.spline.V.res$to_keep) %>%
  which()
```

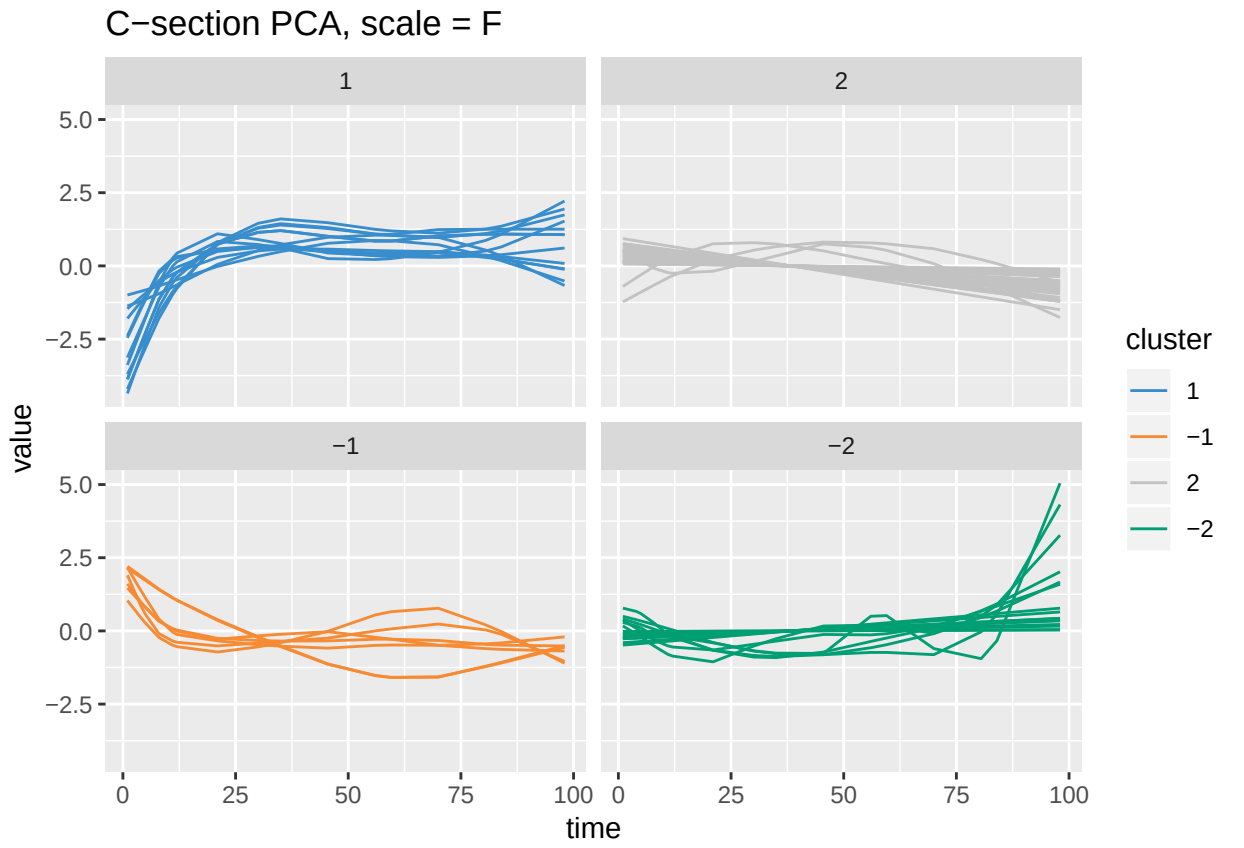
```
spline.data.V <- spline.MILK.V.pspline@predSpline[index.filter.V,] %>% t %>% as.data.frame()
spline.data.V %>% rownames_to_column("time") %>%
  gather(Features, value, - time) %>% mutate(time =as.numeric(time)) %>%
  ggplot(aes(x=time, y = value, col = Features)) + geom_line() + theme_bw() +
  theme(legend.position = "none") + ggtitle("Modelled OTU evolution")
```



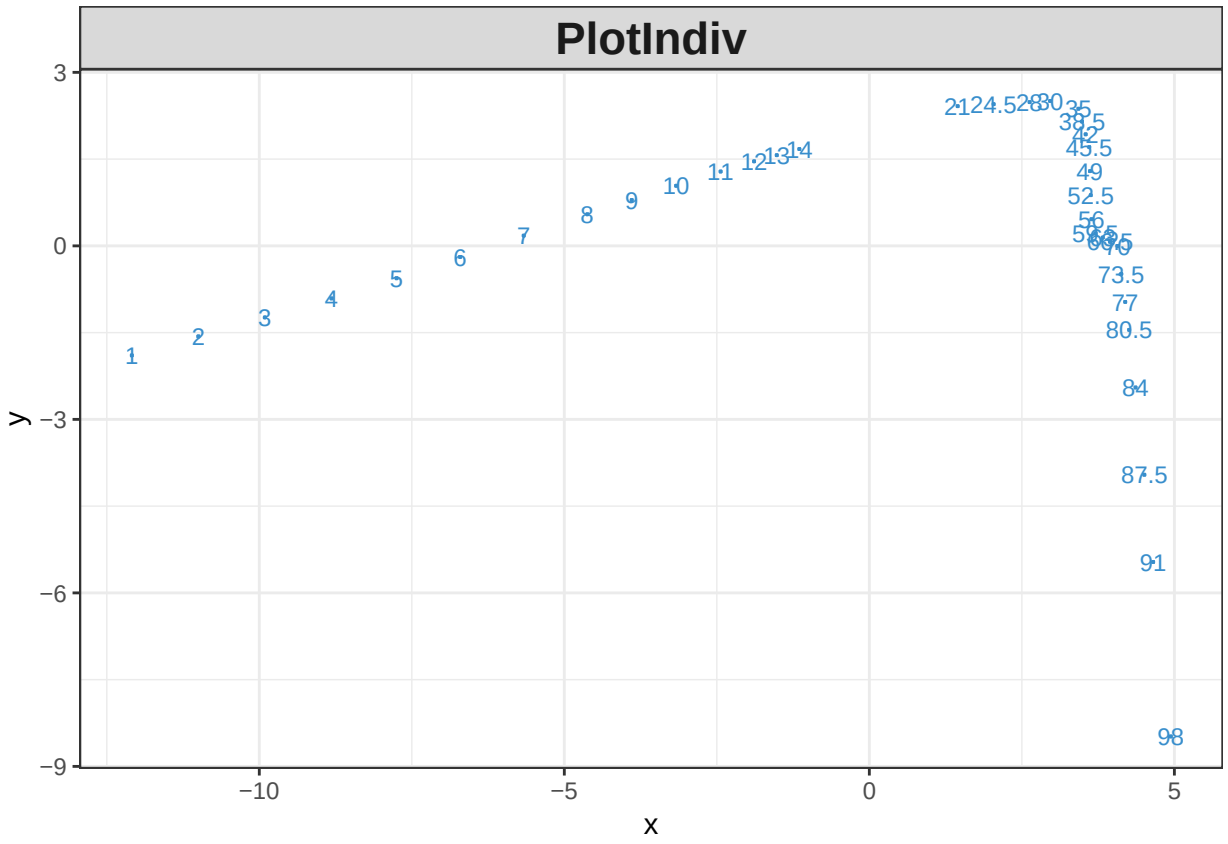
PCA Clustering

C-section

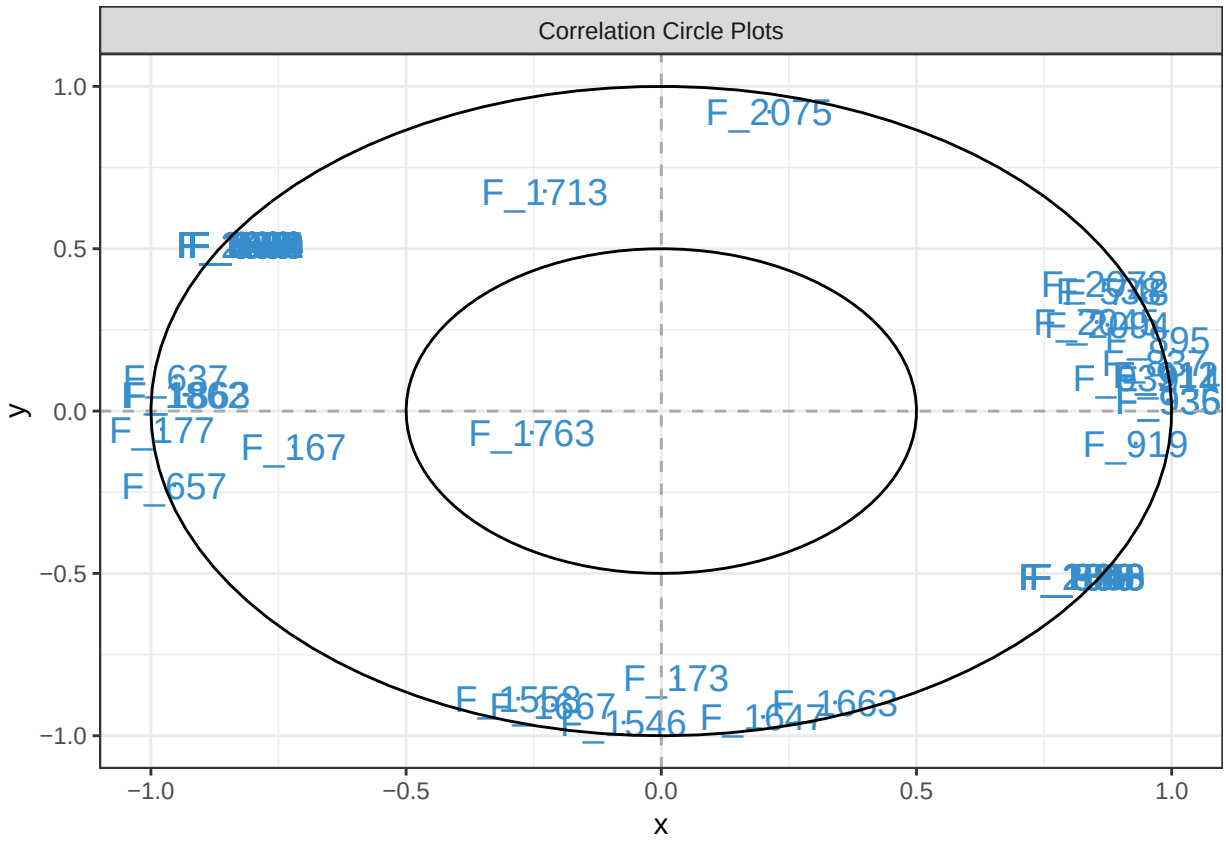
```
pca.res.C <- pca(spline.data.C, ncomp = 2, scale = F, center = T)
pca.plot(pca.res.C, title = "C-section PCA, scale = F")
```




```
plotIndiv(pca.res.C)
```



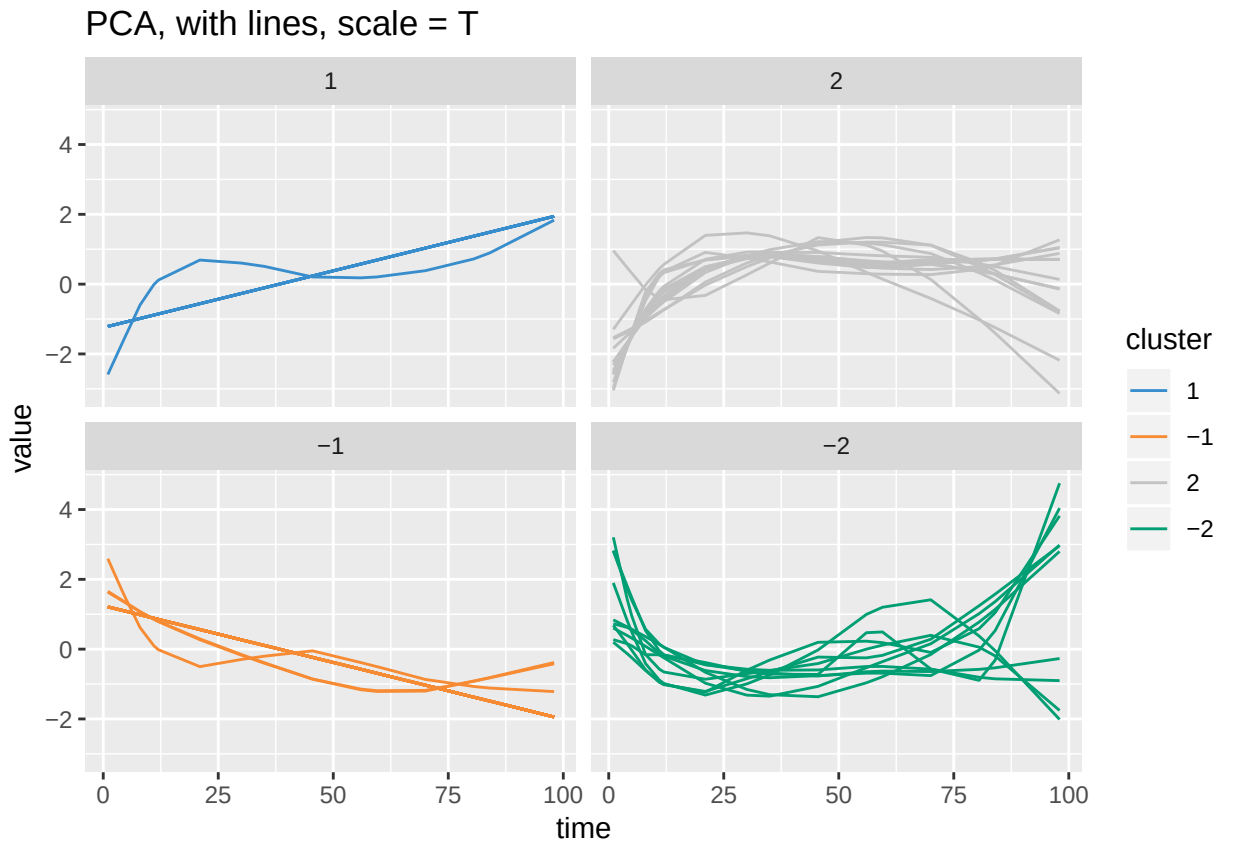

```
plotVar(pca.res.C)
```




```
pca.res.C <- pca(spline.data.C, ncomp = 2, scale = T, center = T)
pca.get_cluster(pca.res.C) %>% pull(cluster) %>% table
```

```
## .
## -2 -1  1  2
## 10 35 11 15
```

```
pca.plot(pca.res.C, title = "PCA, with lines, scale = T")
```



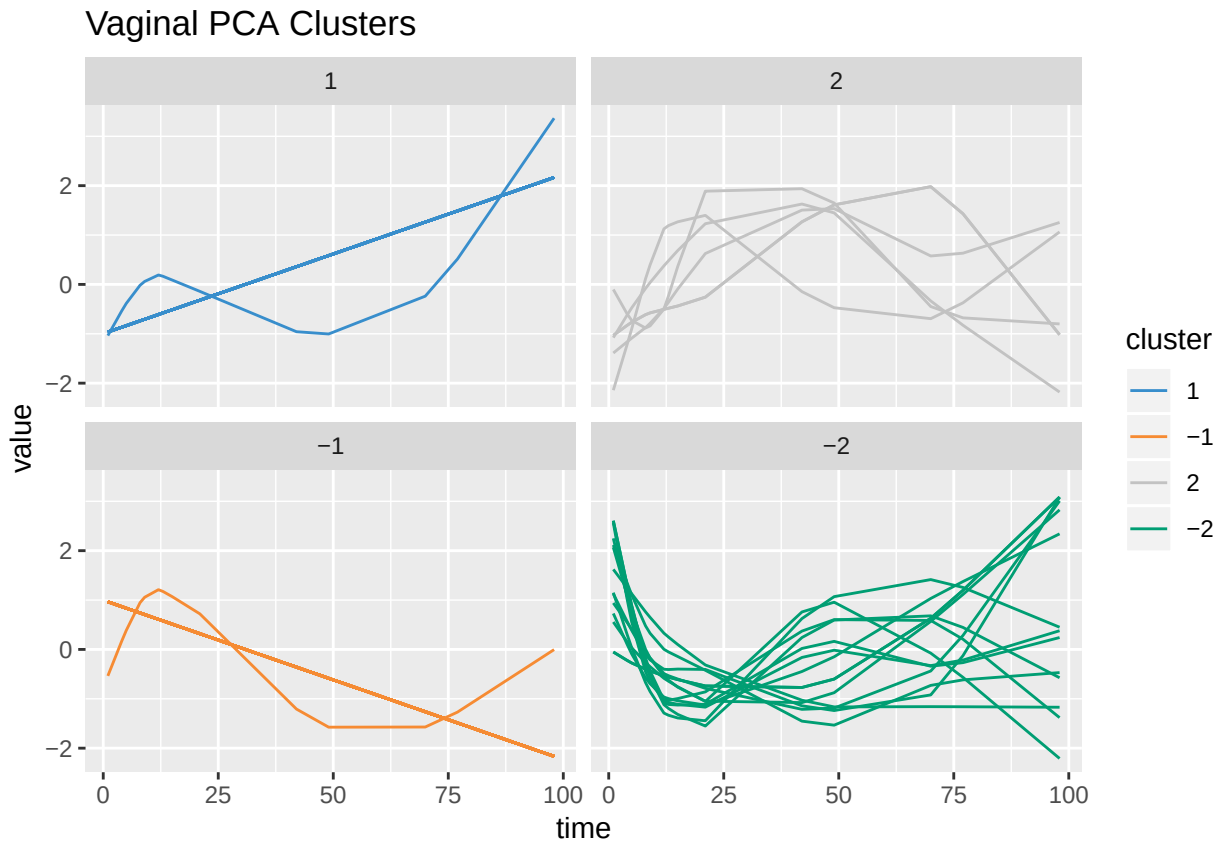

```
# paper
pca.plot(pca.res.C, title = "C-section PCA Clusters")
```


```
pca.res.C <- pca(spline.data.C, ncomp = 2, scale = F, center = T)

# silhouette coefficient for this clustering
wrapper.silhouette.pca(spline.data.C, ncomp = 2, scale = T, center=T)
```

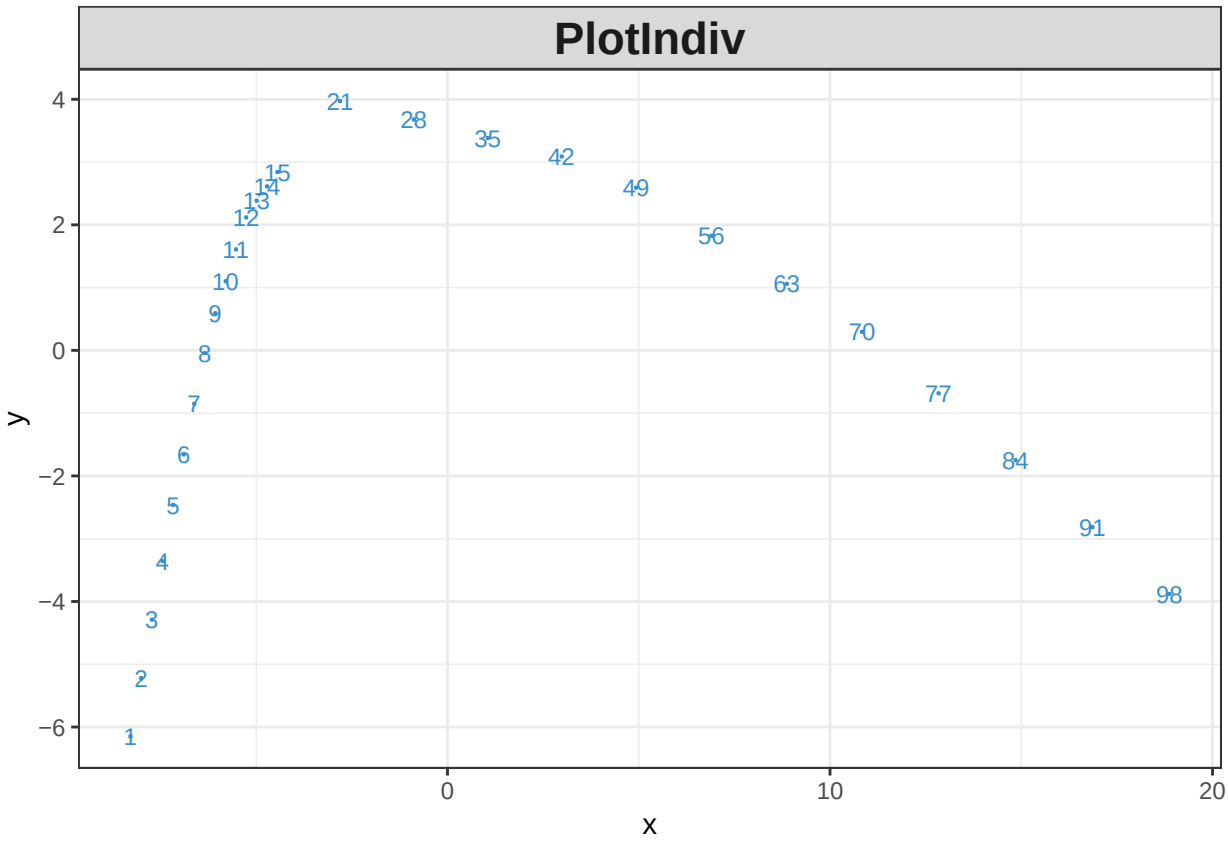
```
## [1] 0.8426561
```

Vaginal

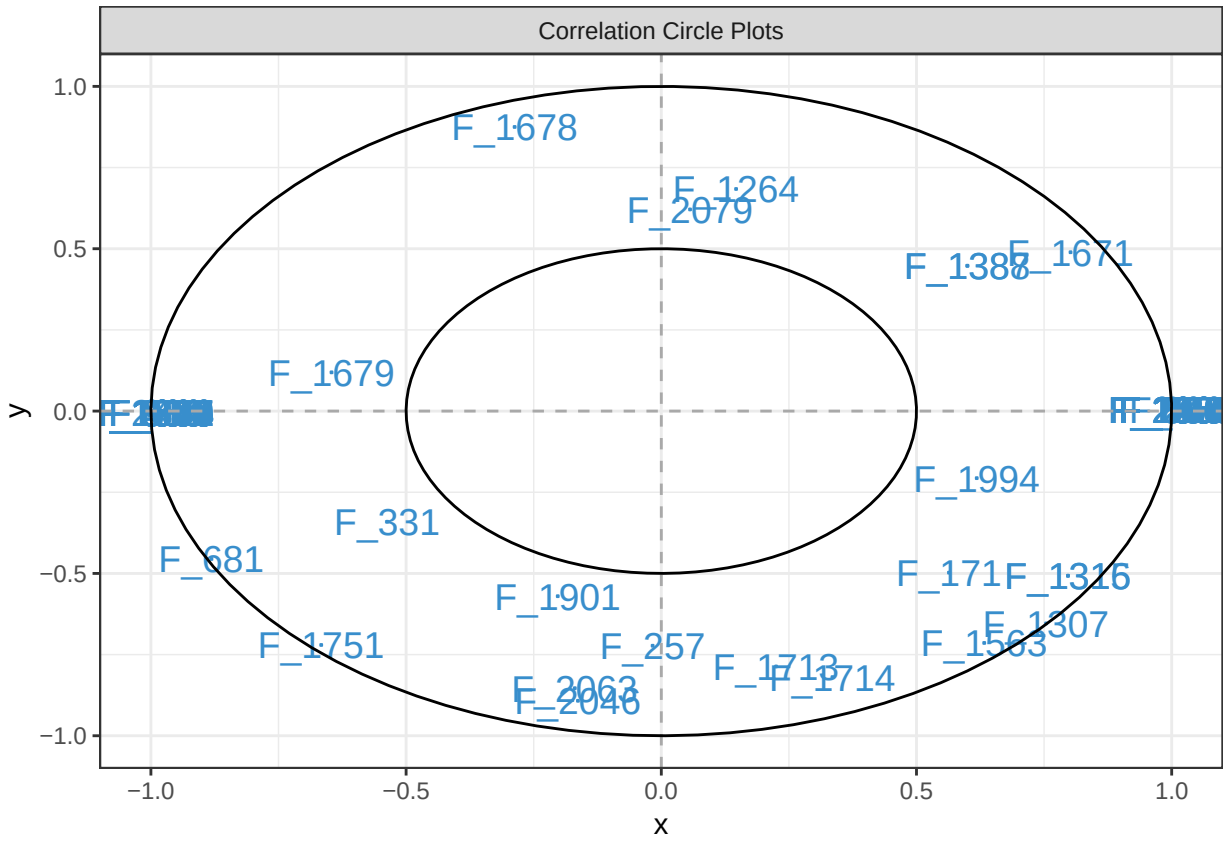
```
pca.res.V <- pca(spline.data.V, ncomp = 2, scale = T, center = T)
pca.plot(pca.res.V, title = "Vaginal PCA Clusters")
```




```
plotIndiv(pca.res.V)
```



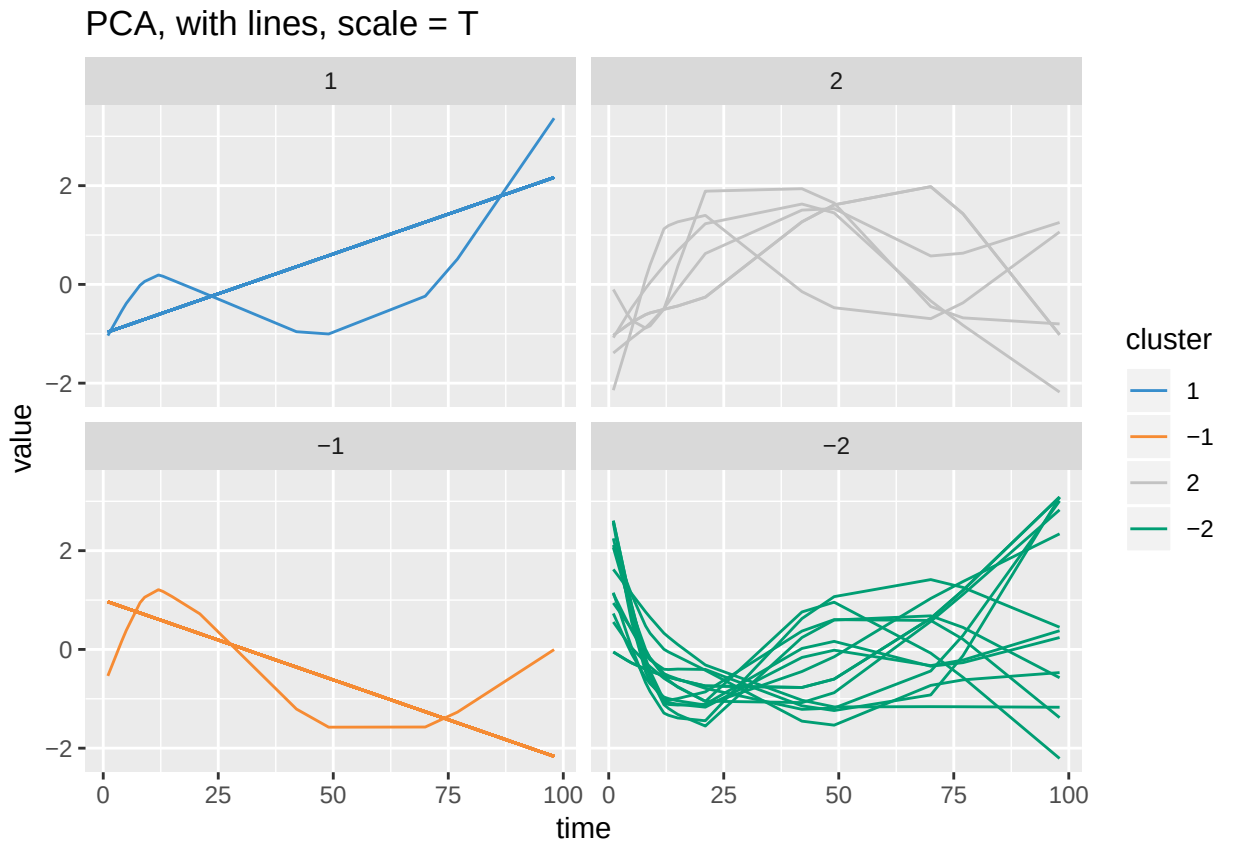

```
plotVar(pca.res.V)
```




```
pca.res.V <- pca(spline.data.V, ncomp = 2, scale = T, center = T)
pca.get_cluster(pca.res.V) %>% pull(cluster) %>% table
```

```
## .
## -2 -1  1  2
## 14 38 32  6
```

```
pca.plot(pca.res.V, title = "PCA, with lines, scale = T")
```



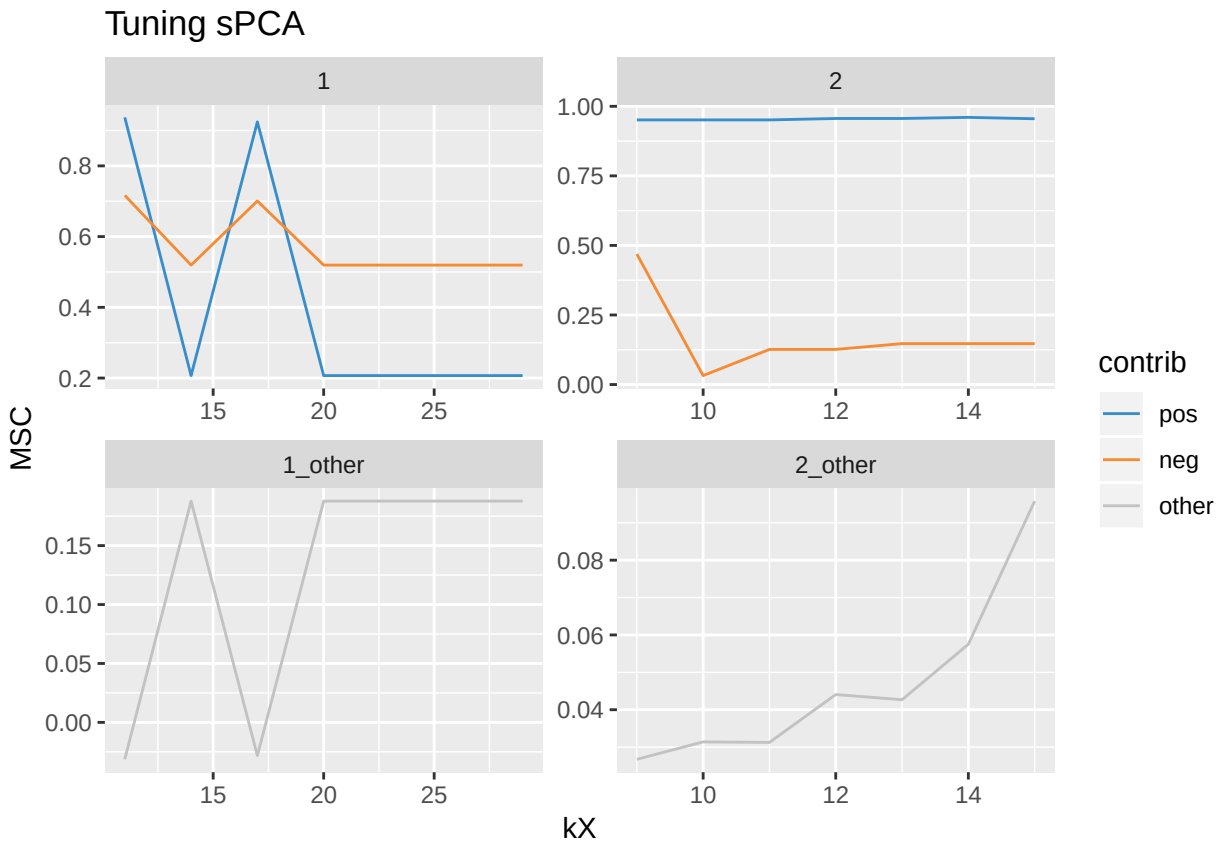

```
pca.res.V <- pca(spline.data.V, ncomp = 2, scale = F, center = T)

# silhouette coefficient for this clustering
wrapper.silhouette.pca(spline.data.V, ncomp = 2, scale = T, center=T)
```

```
## [1] 0.8730084
```

sparse PCA Clustering

C-section

```
keepX = list(seq(11,29, 3), seq(9,15,1))
res.tune.spca.C <- tune.spca(X = spline.data.C, ncomp = 2, keepX = keepX)
tune.spca.choice.keepX(res.tune.spca.C, draw = T)
```

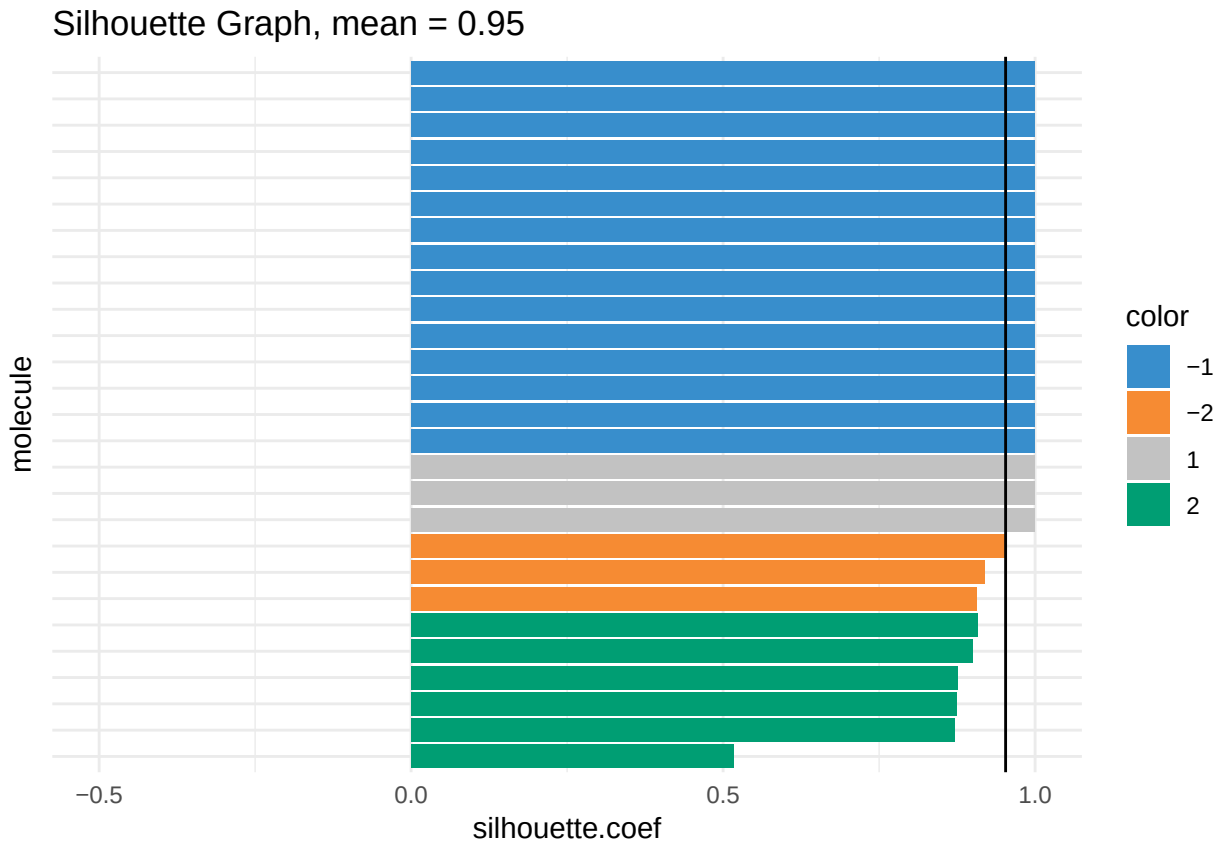



```
## [1] 11 NA
```

```
spca.res_f.C <- spca(spline.data.C, ncomp = 2, keepX = c(17,9))
pca.get_cluster(spca.res_f.C) %>% pull(cluster) %>% table
```

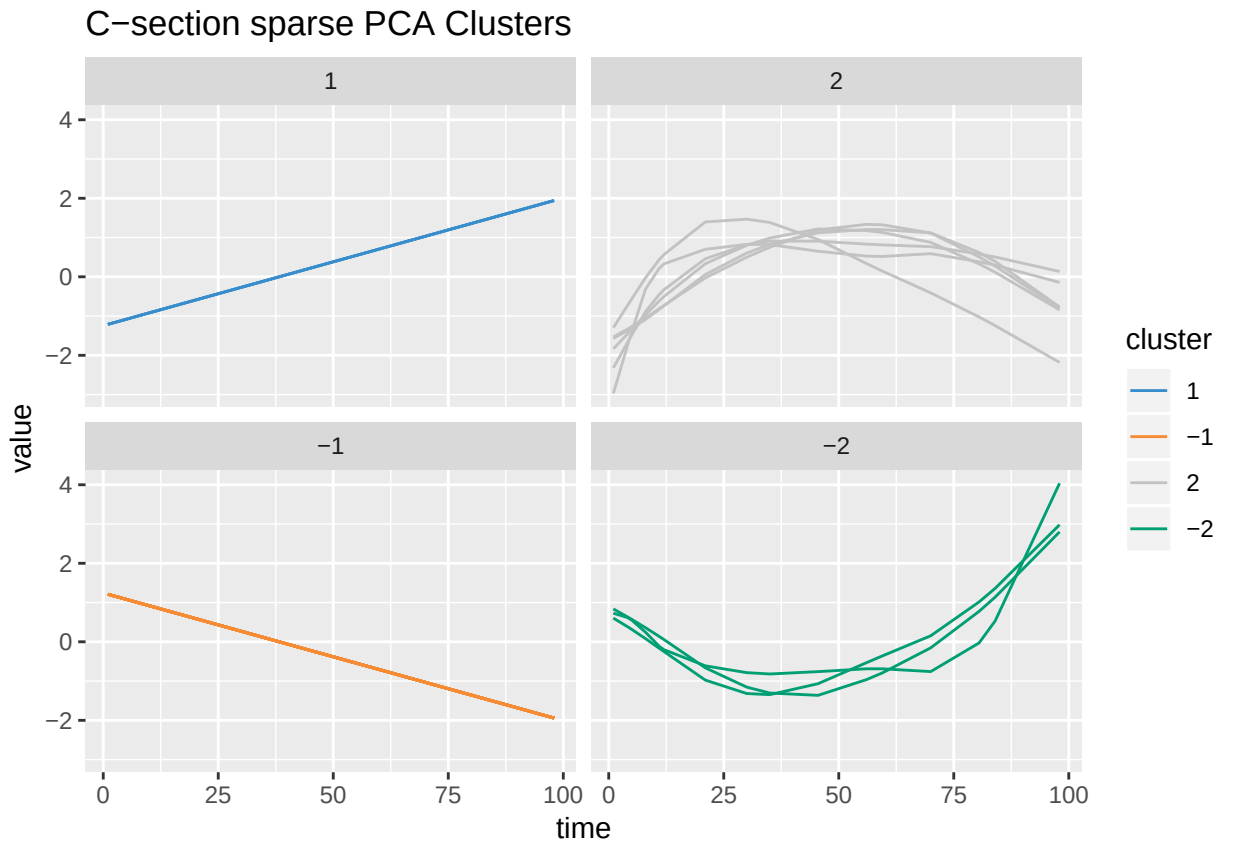
```
## .
## -2 -1  0  1  2
##  3 15 88  3  6
```

```
wrapper.silhouette.spca.paper(spline.data.C, keepX = c(17,9),  ncomp = 2, scale = T, center=T, plot.t =
```



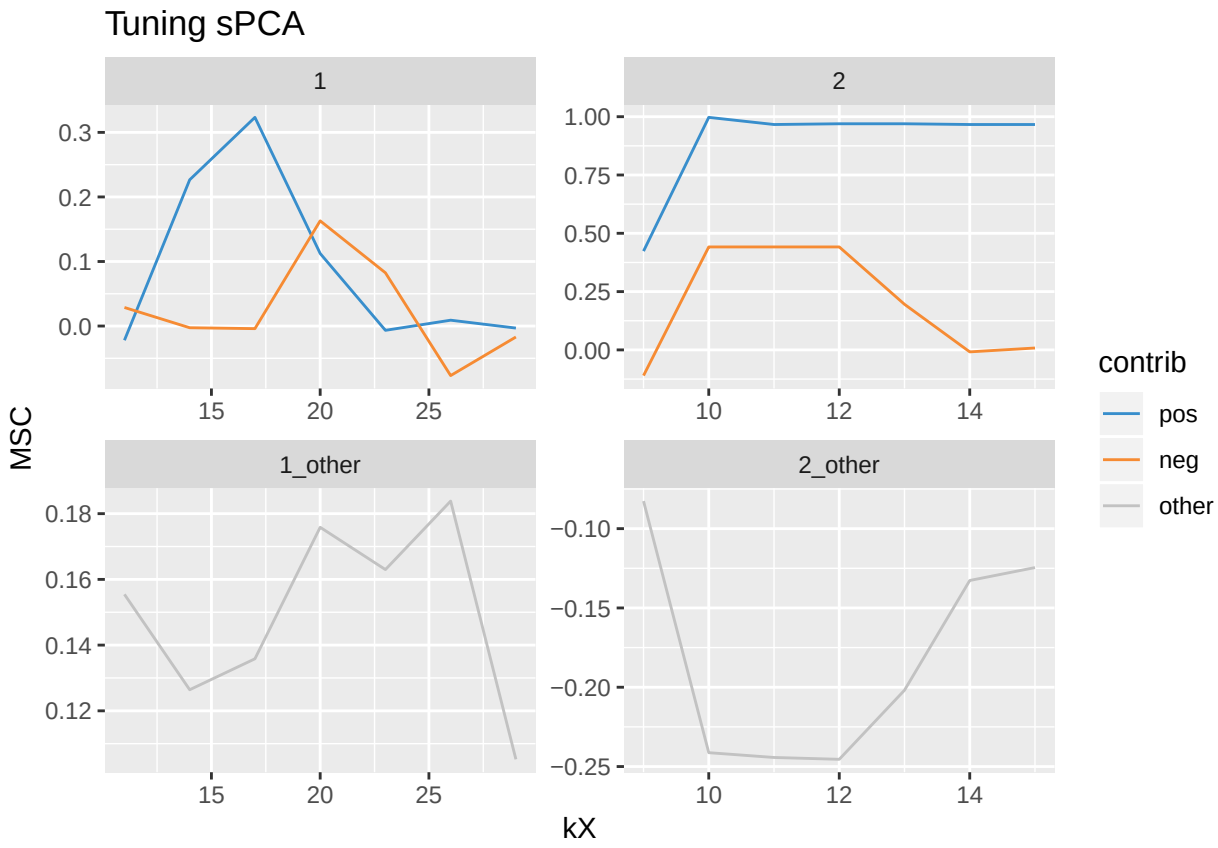

```
## [1] 0.9527507
```

```
spca.plot(spca.res_f.C, title = "C-section sparse PCA Clusters")
```



Vaginal

```
keepX = list(seq(11,29, 3), seq(9,15,1))
res.tune.spca.V <- tune.spca(X = spline.data.V, ncomp = 2, keepX = keepX)
tune.spca.choice.keepX(res.tune.spca.V, draw = T)
```

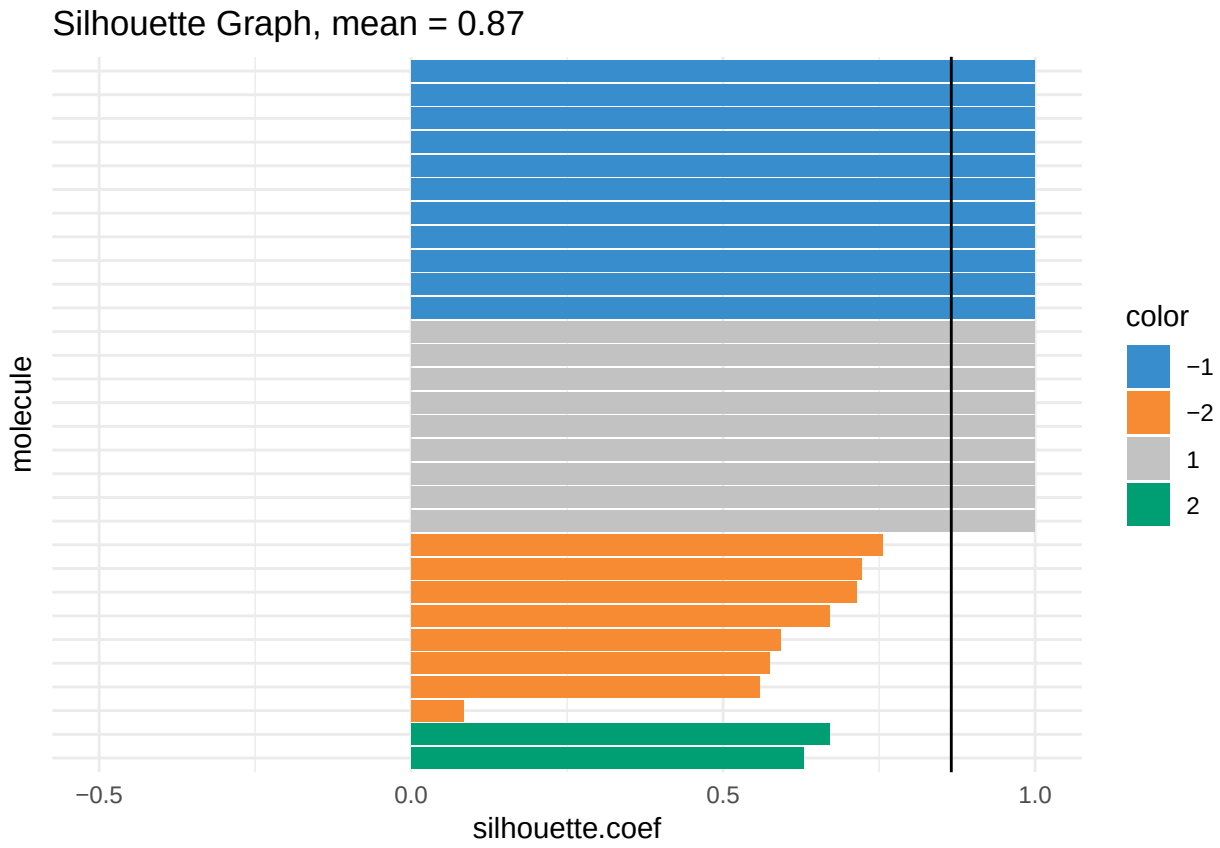



```
## [1] 17 10
```

```
spca.res_f.V <- spca(spline.data.V, ncomp = 2, keepX = c(17,10))
pca.get_cluster(spca.res_f.V) %>% pull(cluster) %>% table
```

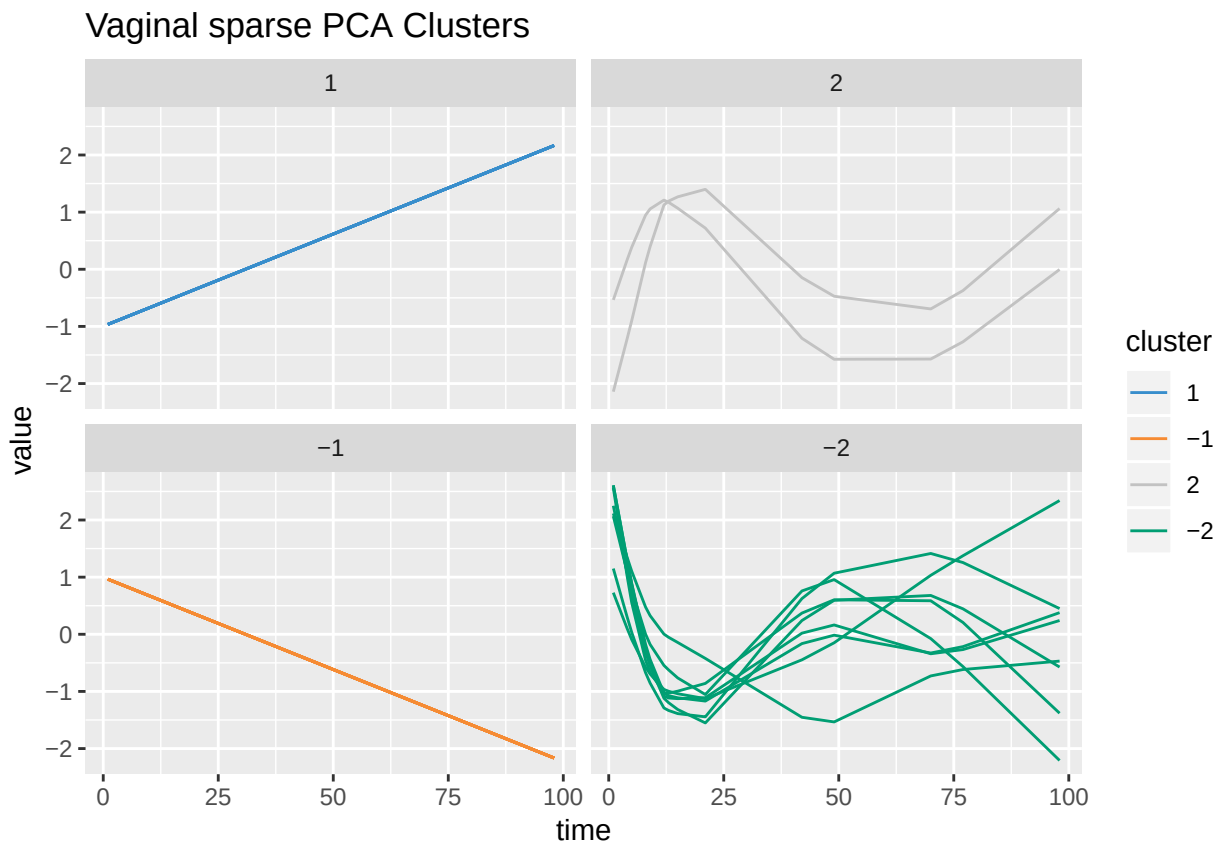
```
## .
##  -2  -1   0   1   2
##   8  11 120   9   2
```

```
wrapper.silhouette.spca.paper(spline.data.V, keepX = c(17,10),  ncomp = 2, scale = T, center=T, plot.t =
```




```
## [1] 0.8656932
```

```
spca.plot(spca.res_f.V, title = "Vaginal sparse PCA Clusters")
```



Results

Silhouette coefficient :

	PCA	PCA w/o lines	sPCA	sPCA w/o lines
C-section	0.84	0.69	0.95	0.72
Vaginal	0.87	0.44	0.86	0.58

nb of molecules :

	0 (after filter)	1
C-section	42	29
Vaginal	68	22

Comparison with fPCA

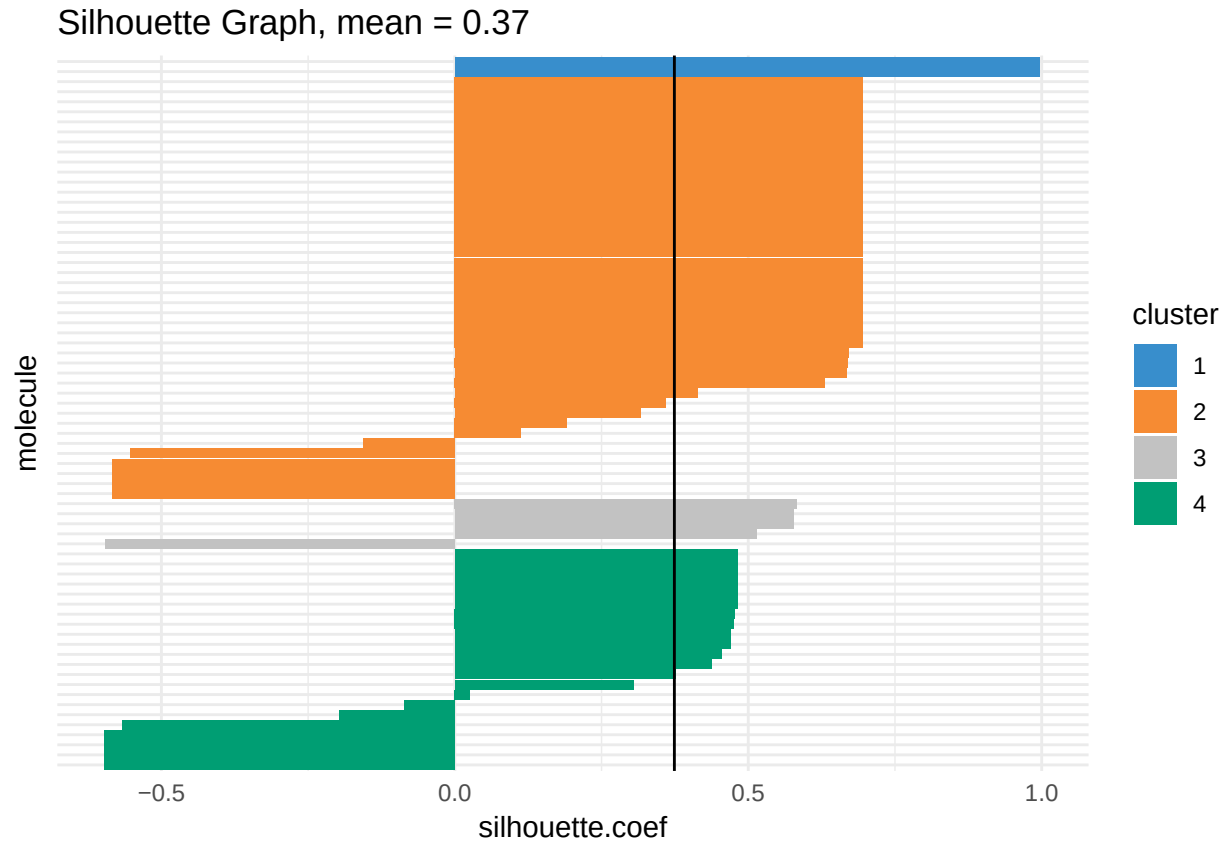
C-section

```
library(fdapace)
```

```
data <- as.matrix(spline.data.C)

# prepare fclust input
FPCA_input <- MakeFPCAInputs(IDs = colnames(data) %>% rep(each=dim(data)[1]),
                             tVec = rep(rownames(data) %>% as.numeric(),dim(data)[2]),
                            yVec = data)
set.seed(123)
fclust.res <- FClust(FPCA_input$Ly, FPCA_input$Lt,
                     optnsFPCA = list(userBwCov= 2, FVEthreshold = 0.90),
                     k = 4, cmethod = "EMCluster")

tmp <- bind_cols(as.data.frame(colnames(data)),
                 as.data.frame(as.character(fclust.res$cluster))) %>%
    set_names(c("molecule", "cluster"))

DF <- Spearman_distance(data)
B <- Add_Cluster_metadata(DF, tmp)
SC.fpca.1 <- Slhouette_coef_df(B)
mean(SC.fpca.1$silhouette.coef)
```

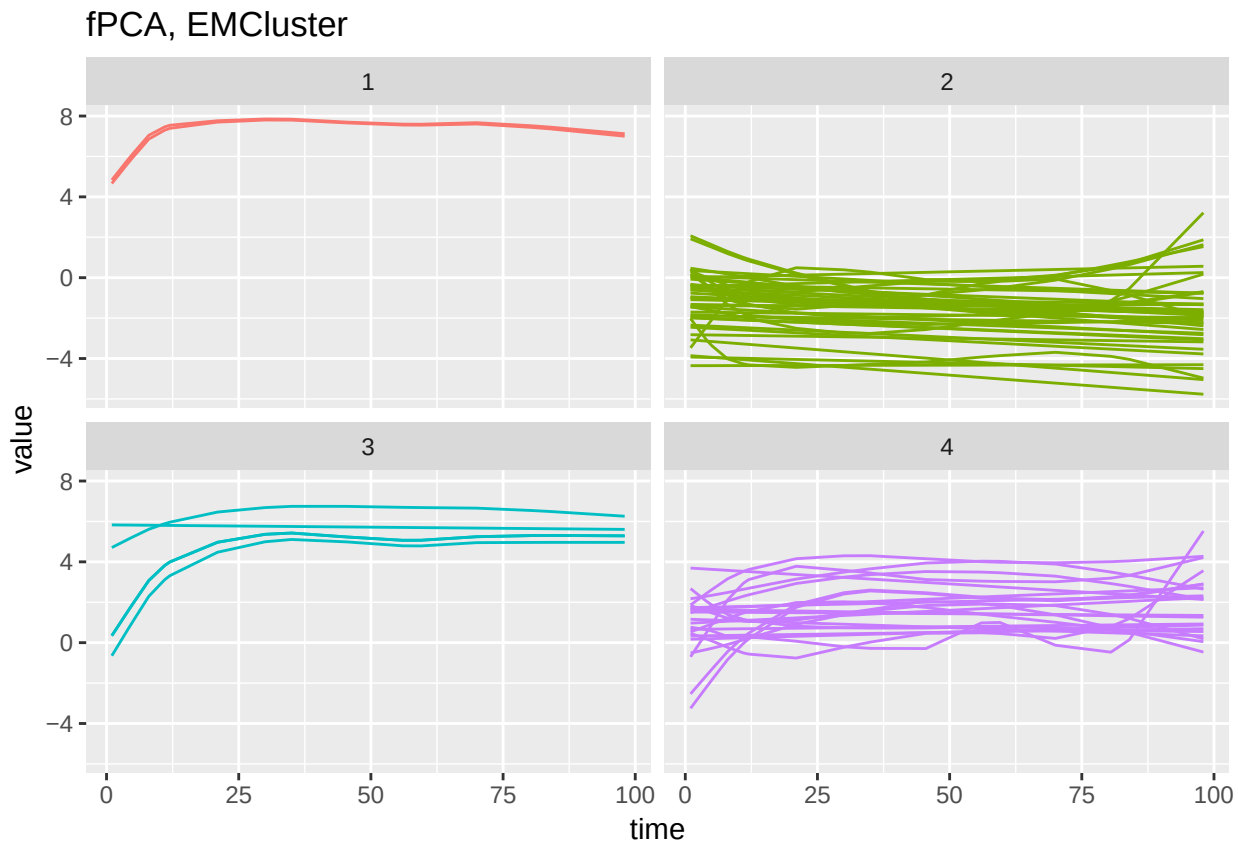
```
## [1] 0.3740866
```

```
#plot_silhouette_order_color(SC.fpca.1)
plot_fig.paper2(SC.fpca.1)
```




```
## plot clusters
data %>% as.data.frame() %>% rownames_to_column("time") %>%
```

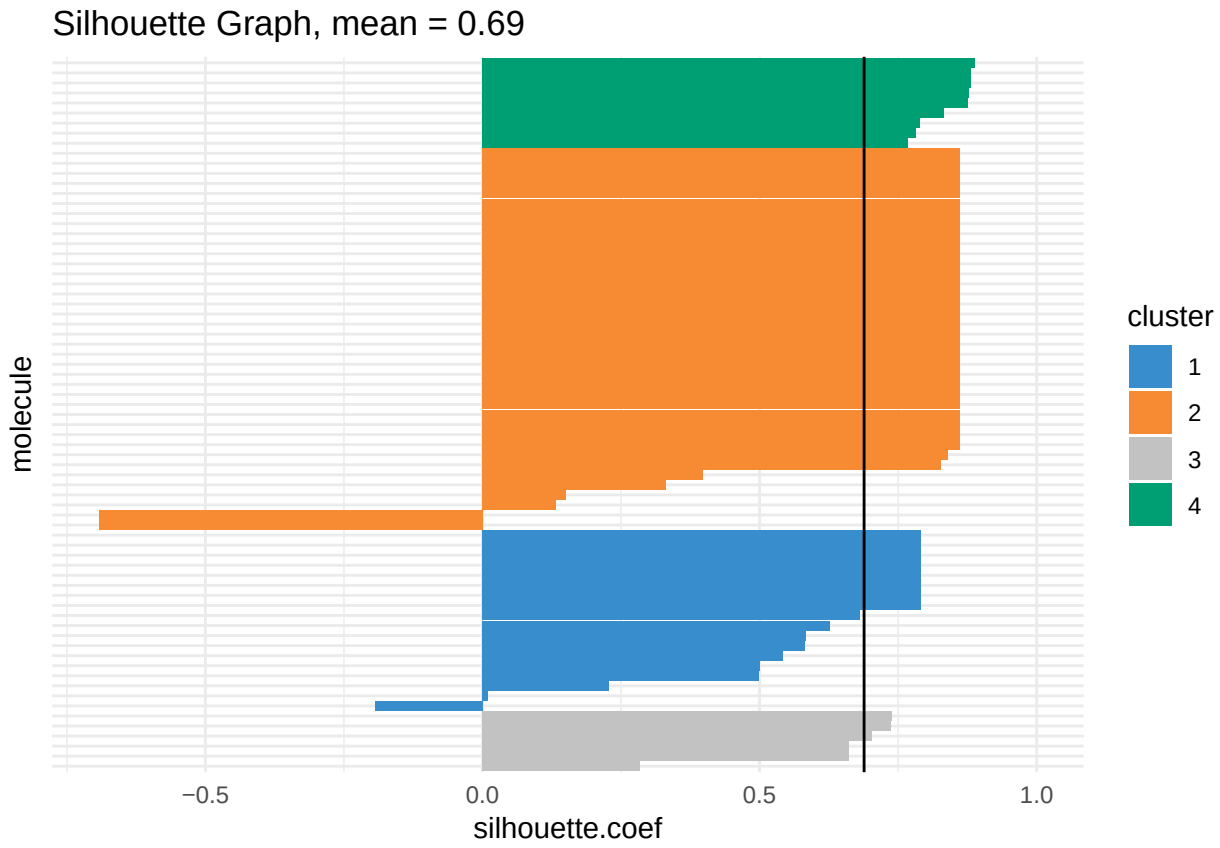
```
  gather(molecule, value, -time) %>%
  left_join(tmp) %>%   # add cluster metadata %>%
  mutate(time = as.numeric(time)) %>%
  ggplot(aes(x=time, y=value, group=molecule, color = as.factor(cluster))) +
  geom_line() + facet_wrap(~as.factor(cluster)) + theme(legend.position="none") +
  ggtitle("fPCA, EMCluster")
```



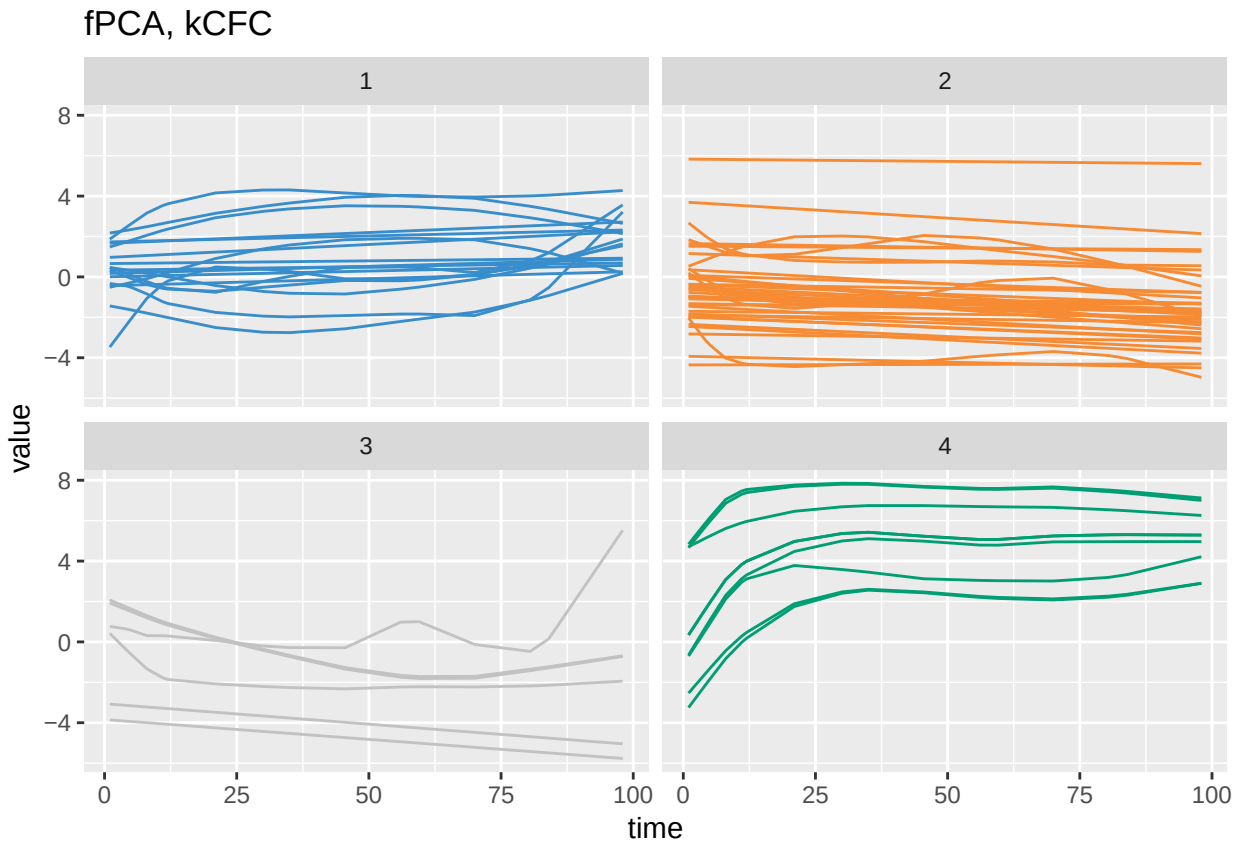

```
# idem with
set.seed(123)
fclust.res.2 <- FClust(FPCA_input$Ly, FPCA_input$Lt,
                       optnsFPCA = list(userBwCov= 2, FVEthreshold = 0.90),
                       k = 4, cmethod = "kCFC")

tmp <- bind_cols(as.data.frame(colnames(data)),
                 as.data.frame(as.character(fclust.res.2$cluster))) %>%
    set_names(c("molecule", "cluster"))

B <- Add_Cluster_metadata(DF, tmp)
SC.fpca.2 <- Slhouette_coef_df(B)
mean(SC.fpca.2$silhouette.coef)
```

```
## [1] 0.6885726
```

```
#plot_silhouette_order_color(SC.fpca.2)
plot_fig.paper2(SC.fpca.2)
```




```
## plot clusters
data %>% as.data.frame() %>% rownames_to_column("time") %>%
  gather(molecule, value, -time) %>%
  left_join(tmp) %>%   # add cluster metadata %>%
  mutate(time = as.numeric(time)) %>%
  ggplot(aes(x=time, y=value, group=molecule, color = as.factor(cluster))) +
  geom_line() + facet_wrap(~as.factor(cluster)) + theme(legend.position="none") +
  scale_color_manual(values=color.mixo(1:4)) + ggtitle("fPCA, kCFC")
```



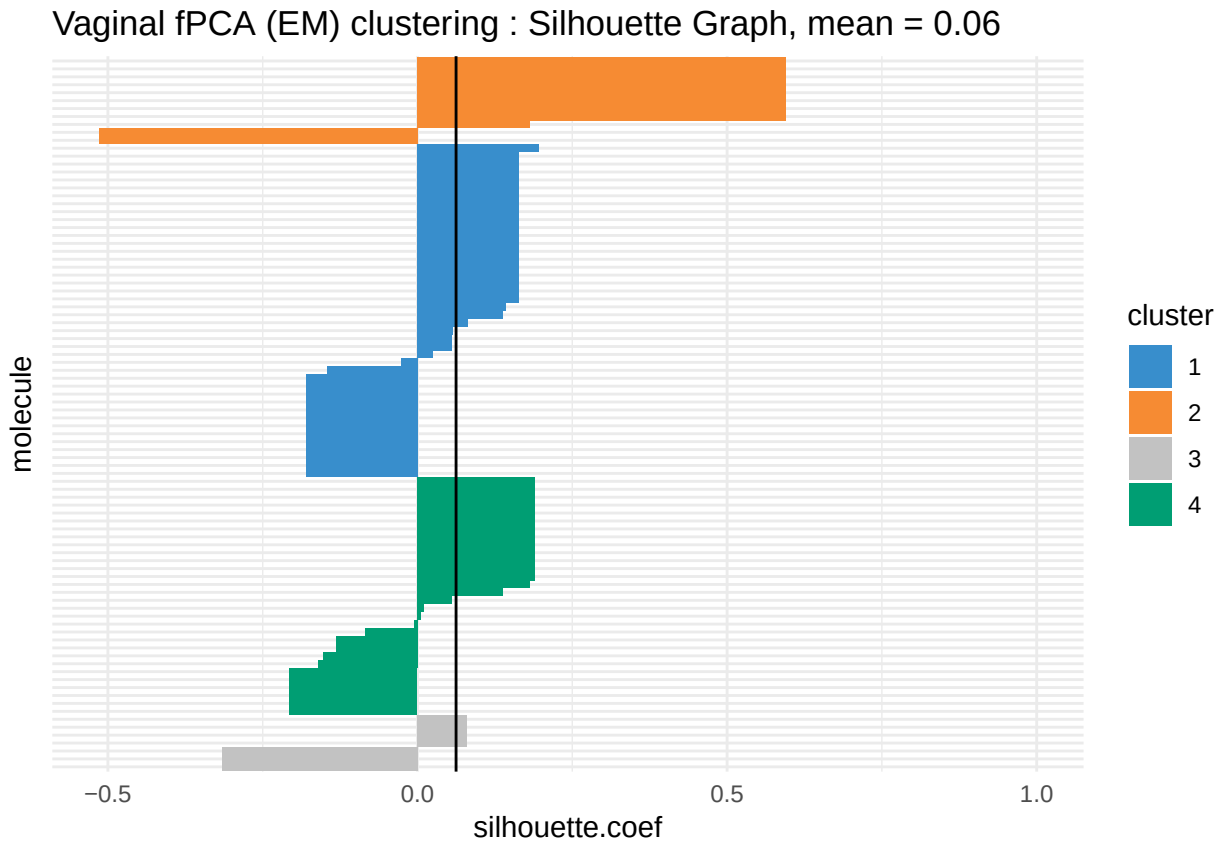
Vaginal

```
data <- as.matrix(spline.data.V)

# prepare fclust input
FPCA_input <- MakeFPCAInputs(IDs = colnames(data) %>% rep(each=dim(data)[1]),
                             tVec = rep(rownames(data) %>% as.numeric(),dim(data)[2]),
                             yVec = data)
set.seed(123)
fclust.res <- FClust(FPCA_input$Ly, FPCA_input$Lt,
                     optnsFPCA = list(userBwCov= 2, FVEthreshold = 0.90),
                     k = 4, cmethod = "EMCluster")

tmp <- bind_cols(as.data.frame(colnames(data)),
                 as.data.frame(as.character(fclust.res$cluster))) %>%
    set_names(c("molecule", "cluster"))

DF <- Spearman_distance(data)
B <- Add_Cluster_metadata(DF, tmp)
SC.fpca.1 <- Slhouette_coef_df(B)
mean(SC.fpca.1$silhouette.coef)
```

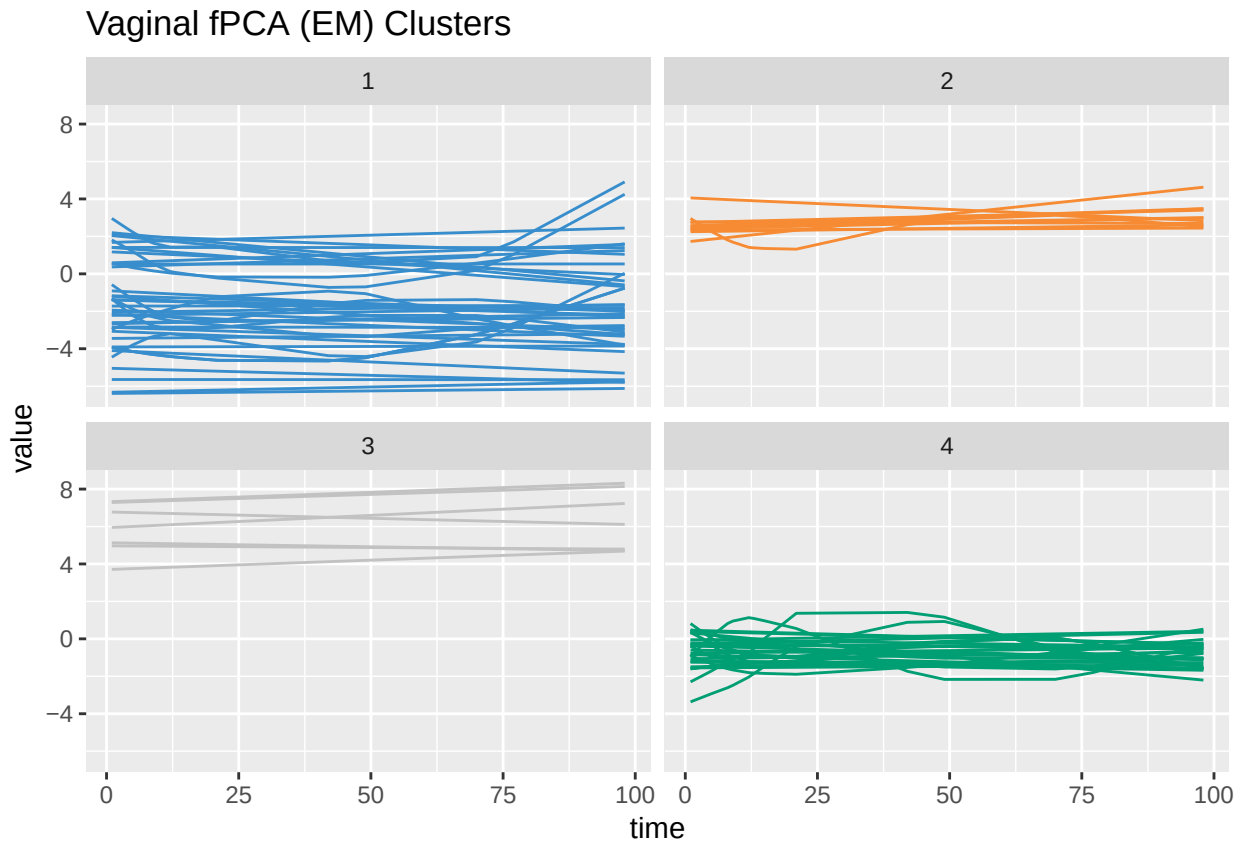
```
## [1] 0.06230024
```

```
#plot_silhouette_order_color(SC.fpca.1)
```

```
title = "Vaginal fPCA (EM) clustering : "
plot_fig.paper2(SC.fpca.1, title)
```



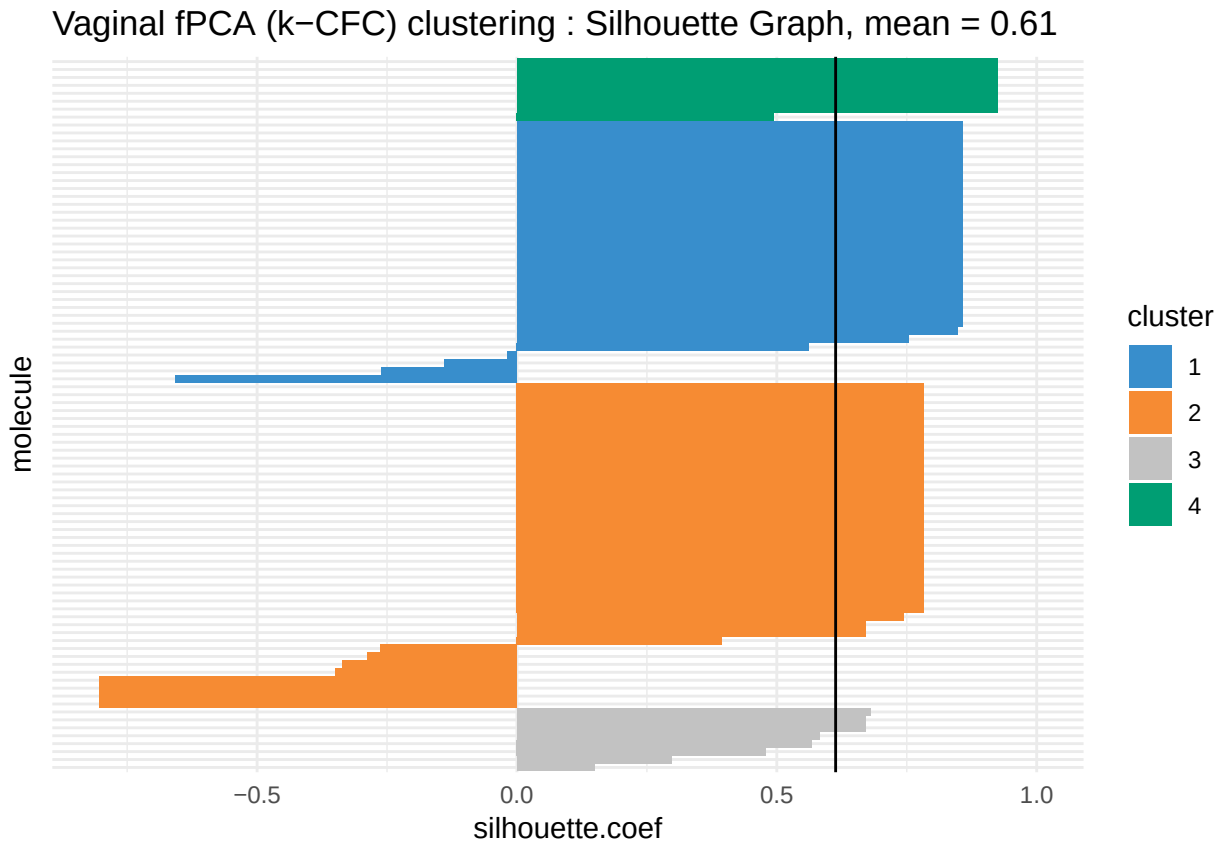

```
## plot clusters
data %>% as.data.frame() %>% rownames_to_column("time") %>%
  gather(molecule, value, -time) %>%
  left_join(tmp) %>%   # add cluster metadata %>%
  mutate(time = as.numeric(time)) %>%
  ggplot(aes(x=time, y=value, group=molecule, color = as.factor(cluster))) +
  geom_line() + facet_wrap(~as.factor(cluster)) + theme(legend.position="none") +
  ggtitle("Vaginal fPCA (EM) Clusters") +
  scale_color_manual(values = color.mixo(1:4))
```




```
# idem with
set.seed(123)
fclust.res.2 <- FClust(FPCA_input$Ly, FPCA_input$Lt,
                       optnsFPCA = list(userBwCov= 2, FVEthreshold = 0.90),
                       k = 4, cmethod = "kCFC")
tmp <- bind_cols(as.data.frame(colnames(data)),
                 as.data.frame(as.character(fclust.res.2$cluster))) %>%
    set_names(c("molecule", "cluster"))

B <- Add_Cluster_metadata(DF, tmp)
SC.fpca.2 <- Slhouette_coef_df(B)
mean(SC.fpca.2$silhouette.coef)
```

```
## [1] 0.6131915
```

```
#plot_silhouette_order_color(SC.fpca.2)
title = "Vaginal fPCA (k-CFC) clustering : "
plot_fig.paper2(SC.fpca.2, title)
```




```
## plot clusters
data %>% as.data.frame() %>% rownames_to_column("time") %>%
  gather(molecule, value, -time) %>%
  left_join(tmp) %>%   # add cluster metadata %>%
  mutate(time = as.numeric(time)) %>%
  ggplot(aes(x=time, y=value, group=molecule, color = as.factor(cluster))) +
  geom_line() + facet_wrap(~as.factor(cluster)) + theme(legend.position="none") +
  scale_color_manual(values=color.mixo(1:4)) + ggtitle("Vaginal fPCA (k-CFC) Clusters")
```


```
sessionInfo()
```

```
## R version 3.5.2 (2018-12-20)
## Platform: x86_64-redhat-linux-gnu (64-bit)
## Running under: Fedora 29 (Workstation Edition)
##
## Matrix products: default
## BLAS/LAPACK: /usr/lib64/R/lib/libRblas.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] fdapace_0.4.0     bindrcpp_0.2.2    pspline_1.0-18
##  [4] geiger_2.0.6      ape_5.2           usethis_1.4.0
##  [7] devtools_2.0.1    clValid_0.6-6     edci_1.1-3
## [10] mclust_5.4.2      cluster_2.0.7-1   dynOmics_1.2
## [13] lmeSplines_1.1-10 nlme_3.1-137      knitr_1.20
```

```
## [16] reshape2_1.4.3    lmms_1.3.3        lmtest_0.9-36
## [19] zoo_1.8-4         tseries_0.10-46   mixOmics_6.6.0
## [22] lattice_0.20-38   MASS_7.3-51.1     forcats_0.3.0
## [25] stringr_1.3.1     dplyr_0.7.8       purrr_0.2.5
## [28] readr_1.3.0       tidyr_0.8.2       tibble_1.4.2
## [31] ggplot2_3.1.0     tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
##  [1] colorspace_1.3-2    class_7.3-14        rprojroot_1.3-2
##  [4] htmlTable_1.12      corpcor_1.6.9       base64enc_0.1-3
##  [7] fs_1.2.6            rstudioapi_0.8      remotes_2.0.2
## [10] RSpectra_0.13-1     mvtnorm_1.0-8       lubridate_1.7.4
## [13] xml2_1.2.0          splines_3.5.2       pkgload_1.0.2
## [16] Formula_1.2-3       jsonlite_1.6        broom_0.5.1
## [19] compiler_3.5.2      httr_1.4.0          backports_1.1.2
## [22] assertthat_0.2.0    Matrix_1.2-15       lazyeval_0.2.1
## [25] cli_1.0.1           acepack_1.4.1       htmltools_0.3.6
## [28] prettyunits_1.0.2   tools_3.5.2         igraph_1.2.2
## [31] coda_0.19-2         gtable_0.2.0        glue_1.3.0
## [34] Rcpp_1.0.0          cellranger_1.1.0    gdata_2.18.0
## [37] ps_1.2.1            rvest_0.3.2         gtools_3.8.1
## [40] scales_1.0.0        subplex_1.5-4       hms_0.4.2
## [43] RColorBrewer_1.1-2  yaml_2.2.0          quantmod_0.4-13
## [46] curl_3.2            memoise_1.1.0       gridExtra_2.3
## [49] rpart_4.1-13        latticeExtra_0.6-28 stringi_1.2.3
## [52] desc_1.2.0          checkmate_1.8.5     TTR_0.23-4
## [55] caTools_1.17.1.1    pkgbuild_1.0.2      rlang_0.3.0.1
## [58] pkgconfig_2.0.2     matrixStats_0.54.0  bitops_1.0-6
## [61] pracma_2.2.2        evaluate_0.12       bindr_0.1.1
## [64] htmlwidgets_1.3     tidyselect_0.2.5    processx_3.2.1
## [67] deSolve_1.21        plyr_1.8.4          magrittr_1.5
## [70] R6_2.3.0            Hmisc_4.1-1         gplots_3.0.1
## [73] generics_0.0.2      foreign_0.8-71      pillar_1.3.0
## [76] haven_2.0.0         withr_2.1.2         xts_0.11-2
## [79] nnet_7.3-12         survival_2.43-3     modelr_0.1.2
## [82] crayon_1.3.4        rARPACK_0.11-0      KernSmooth_2.23-15
## [85] ellipse_0.4.1       rmarkdown_1.10.14   grid_3.5.2
## [88] readxl_1.1.0        data.table_1.11.8   callr_3.1.0
## [91] htmldeps_0.1.1      digest_0.6.18       numDeriv_2016.8-1
## [94] munsell_0.5.0       sessioninfo_1.1.1   quadprog_1.5-5
```