Aug 05, 2022

CONNECTOR, fitting and clustering of longitudinal data.

Simone Pernice[1], Roberta Sirovich[2], Elena Grassi[3], Marco Viviani[3], Martina Ferri[3], Francesco Sassi[3], Luca Alessandri'[4], Dora Tortarolo[1], Raffaele A. Calogero[4], Livio Trusolino[3], Andrea Bertotti[3], Marco Beccuti[1], Martina Olivero[3], Francesca Cordero[1]

[1]Department of Computer Science, University of Torino, 10149, Turin, Italy;

[2]Department of Mathematics G. Peano, University of Torino, 10123, Turin, Italy;

[3]Candiolo Cancer Institute, FPO-IRCCS, 10060 Candiolo, Italy;

[4]Department of Molecular Biotechnology and Health Sciences, University of Torino, 10126 Torino, Italy.

Simone Pernice: First author;

Roberta Sirovich: First author;

Elena Grassi: Corresponding author;

Francesca Cordero: Corresponding author

dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

Simone Pernice
University of Turin

ABSTRACT

The transition from the evaluation of a single time point to the examination of the entire dynamic evolution of a system is possible only in the presence of the proper framework. The strong variability of dynamic evolution makes the definition of an explanatory procedure for data fitting and data clustering challenging.

Here we present CONNECTOR, a data-driven framework able to analyze and inspect longitudinal data in a straightforward and revealing way. When used to analyze tumor growth kinetics over time in 1599 patient-derived xenograft (PDX) growth curves from ovarian and colorectal cancers, CONNECTOR allowed the aggregation of time-series data through an unsupervised approach in informative clusters.

Through the lens of a new perspective of mechanism interpretation, CONNECTOR shed light onto novel model aggregations and identified unanticipated molecular associations with response to clinically approved therapies.

DOI

dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

EXTERNAL LINK

https://qbioturin.github.io/connector/

Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

PROTOCOL CITATION

Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero 2022. CONNECTOR, fitting and clustering of longitudinal data.. **protocols.io** https://protocols.io/view/connector-fitting-and-clustering-of-longitudinal-d-xdvfi66

FUNDERS ACKNOWLEDGEMENT

AIRC
Grant ID: 20697

AIRC
Grant ID: 22802

AIRC 5x1000
Grant ID: 21091

AIRC/CRUK/FC AECC Accelerator Award
Grant ID: 22795

European Research Council Consolidator Grant
Grant ID: 724748—BEAT

H2020 COLOSSUS
Grant ID: 754923

H2020 INFRAIA
Grant ID: 731105 EDIReX

FPRC-ONLUS, 5x1000 Ministero della Salute 2016

KEYWORDS

Longitudinal data, functional data analysis, clustering

LICENSE

CREATED

Jan 23, 2019

LAST MODIFIED

Aug 05, 2022

Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

PROTOCOL INTEGER ID

19605

BEFORE STARTING

Before starting the analysis, the softwares R must be installed.

To install R follow the information reported in:
https://www.r-project.org/

In particular, **CONNECTOR** is built under R 4.1.0, for previous versions we suggest to install the right version of the package *locfit*:

```
install_version("locfit", version = "1.5-9.2")
```

Note that R should be built with *tcltk* support. To check this, you should run *capabilities("tcltk")* from R. If it returns FALSE then we suggest to check how to cope with this from here.

Finally, before installing **CONNECTOR**, the following R packages has to be installed:

```
install.packages(c('cowplot', 'fda', 'flexclust','ggplot2',
'MASS',
                   'Matrix', 'plyr','ggplotify', 'RColorBrewer',
                   'readxl', 'reshape2', 'splines', 'statmod',
                   'sfsmisc','shinyWidgets', 'viridis',
'dashboardthemes',

'shinybusy','shinydashboard','shinyjs','tidyr',
                   'shinyFiles','devtools'))"
```

How to install CONNECTOR

1 To install **CONNECTOR** you can use the **devtools** R package:

```
library(devtools)
install_github("https://github.com/qBioTurin/connector",
ref="master")
```

Docker

1.1 Docker

If you want to use Docker (which is not mandatory), you need to have it installed on your machine. For more info see this document:

Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

https://docs.docker.com/engine/installation/.

How to install docker

Ensure your user has the rights to run docker (without the use of *sudo*). To create the docker group and add your user:

- Create the docker group.

```
$ sudo groupadd docker
```

- Add your user to the docker group.

```
$ sudo usermod -aG docker $USER
```

- Log out and log back in so that your group membership is re-evaluated.

How to run CONNECTOR from docker

- **Without installing CONNECTOR.**

Download the image from dockerhub

```
$ docker pull qbioturin/connector:latest
```

Run the RStudio Docker image with CONNECTOR installed and work with RStudio in the browser

```
$ docker run --cidfile=dockerID -e DISABLE_AUTH=true -p
8787:8787 -d qbioturin/connector:latest
 $ open http://localhost:8787/
```

- **From CONNECTOR.**

It is possible to download and run the RStudio Docker image by using the CONNECTOR functions called *downloadContainers* and *docker.run* as follows:

```
> downloadContainers()
> docker.run()
```

Introduction

2 Introduction

The transition from the evaluation of a single time point to the examination of the entire

Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

dynamic evolution of a system is possible only in the presence of the proper framework. Here we introduce **CONNECTOR**, a data-driven framework able to analyze and inspect longitudinal high-dimensional data in a straightforward and revealing way. **CONNECTOR** is a tool for the unsupervised analysis of longitudinal data, that is, it can process any sample consisting of measurements collected sequentially over time. CONNECTOR is built on the model-based approach for clustering functional data presented in

> James, Gareth M and Sugar, Catherine A. Clustering for sparsely sampled functional data. Journal of the American Statistical Association.
> https://doi.org/10.1198/016214503000189

Let us note that this approach is particularly effective when observations are sparse and irregularly spaced, as growth curves usually are.

The **CONNECTOR** framework is based on the following steps:

(1) The **pre-processing step** consists of a visualization of the longitudinal data by a line plot and a time grid heat map helping in the inspection of the sparsity of the time points.

(2) Then, by means of the FDA analysis, the sampled curves are processed by **CONNECTOR** with a functional clustering algorithm based on a mixed effect model. This step requires a **model selection phase**, in which the dimension of the spline basis vector measures and number of clusters are computed that help the user to properly set the two free parameters of the model. The selection of the optimal set of parameter is supported by the generation of several plots.

(3) Once the model selection phase is completed, the **output** of **CONNECTOR** is composed of several graphical visualizations to easily mine the results.

Case study on tumor growth dataset.

3 To illustrate how **CONNECTOR** works we use patient-derived xenografts (PDXs) lines from 21 models generated from chemotherapy-naive high-grade serous epithelial ovarian cancer.

Data Importing

4 Data Importing

The analysis starts from two distinct files.

1. An excel file reporting the discretely sampled curve data. The longitudinal data must be reported in terms of time *t* and y-values *y*. Each sample is described by two columns: the first column, named *time*, includes the lags; while the second column, named with the *ID sample*, contains the list of *y* values. Note that, each sample must have different *ID*

Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

sample. Hence, the 21 tumor growth curves are collected in a file composed of 48 columns.
2. A csv (or txt) file contains the annotations associated with the sampled curves. The first column reports the **IDSample**, the other columns report the features relevant for the analysis, one per column. Note that the column *ID sample* must contain the same ID names which appear in the excel file of the sampled curves.

The two files are imported by the *DataImport* function, which arguments are the file names.

```
# Get the full path names of the example files from the CONNECTOR
package

TimeSeriesFile<-system.file("data", "475dataset.xlsx", package =
"connector")
AnnotationFile <-system.file("data", "475info.txt", package =
"connector")

# Import the data

CONNECTORList<-DataImport(TimeSeriesFile = TimeSeriesFile,
                          AnnotationFile = AnnotationFile)

## ################################
## ######## Summary ##############
##
##  Number of curves: 21 ;
##  Min curve length:  7 ; Max curve length:  18 .
## ################################
```

A list of four objects is created:

```

str(CONNECTORList)

## List of 4
##  $ Dataset :'data.frame':    265 obs. of  3 variables:
##   ..$ ID         : int [1:265] 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ Observation: num [1:265] 63.5 0 78.1 137.9 188.4 ...
##   ..$ Time       : num [1:265] 9 16 23 31 36 44 51 57 64 72 ...
##  $ LenCurv : int [1:21] 15 15 15 14 14 7 7 11 7 11 ...
##  $ LabCurv :'data.frame':    21 obs. of  5 variables:
##   ..$ IDSample    : chr [1:21] "475_P1b" "475_P1bP2a"
"475_P1bP2b" "475_P1bP2aP3a" ...
##   ..$ Progeny     : chr [1:21] "  P1" "  P2" "  P2" "  P3" ...
##   ..$ Source      : chr [1:21] "  P0" "  475_P1b" "  475_P1b" "
475_P1bP2a" ...
```


Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

```
##   ..$ Real.Progeny: chr [1:21] "  P3" "  P4" "  P4" "  P5" ...
##   ..$ ID          : int [1:21] 1 2 3 4 5 6 7 8 9 10 ...
##  $ TimeGrid: num [1:66] 3 5 6 8 9 10 12 13 14 15 ...
```

Let us note that the data can be also imported by using two data frames:

1. *TimeSeriesFrame* : dataframe of three columns storing the time series. The first columns, called "*ID*", stores the sample identification. The second column, labeled "*Observation*", contains the observations over the time, and the third column, called "*Time*", reports the time point.

2. *AnnotationFrame* : dataframe with a number of rows equal to the number of samples in the *TimeSeriesFrame*. The first column must be called "*ID*" reporting the sample identifiers. The other columns correspond to the features associate with the respective sample (one column per sample). If the dataframe is composed of one column, in the *CONNECTORList* only the feature *ID* will be reported.

Hence, *CONNECTORList* can be generated by exploiting the *DataFrameImport* function.

```
# Import the data from dataframes

CONNECTORList<-DataImport(TimeSeriesDataFrame,
                          AnnotationFrame)
```

Data Visualization

5 Data Visualization

The *PlotTimeSeries* function generates the plot of the sampled curves, coloured by the selected feature from the *AnnotationFile*.

Figure 1 reports the line plot generated by the following code:

```
CurvesPlot<-PlotTimeSeries(data = CONNECTORList,
                           feature = "Progeny")
CurvesPlot
```


Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

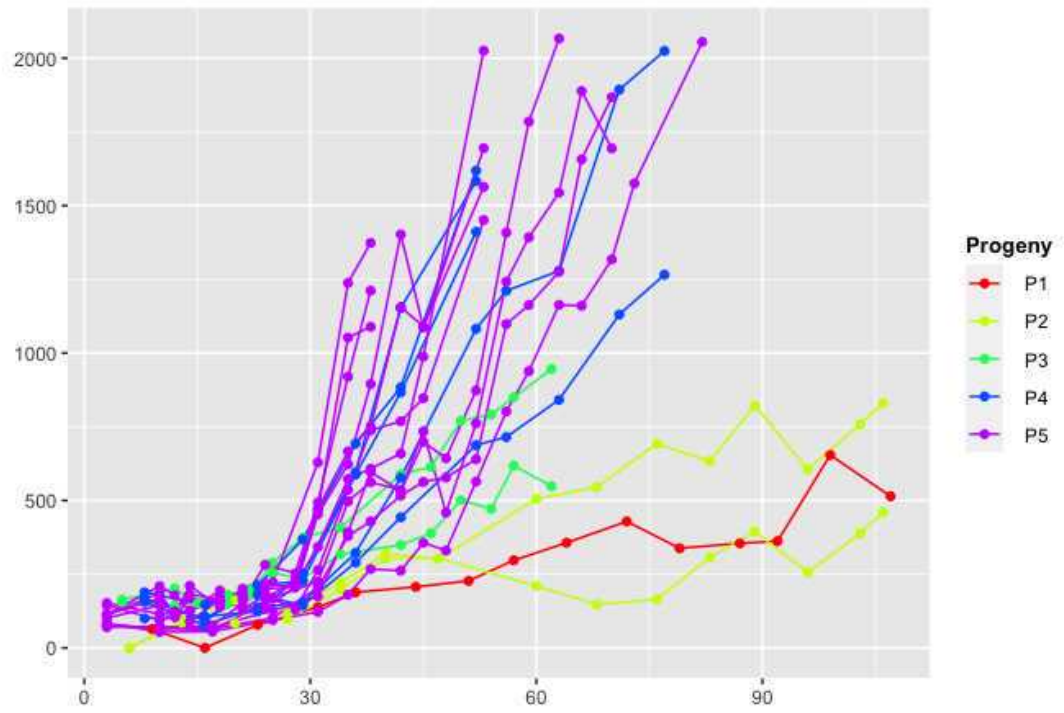

Fig.1 Sampled curves, coloured by progeny feature.

As we can see from Fig.1, only few samples have observations at the last time points. The *DataVisualization*
function plots the time grid heat map helping in the inspection of the sparsity of the time points. The *DataTruncation* function have been developed to truncate the time series at specific time value.

```
# Growth curves and time grid visualization
Datavisual<-DataVisualization(data = CONNECTORList,
                              feature = "Progeny",
                              labels = c("Time","Volume","Tumor
Growth"))
Datavisual
```


Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

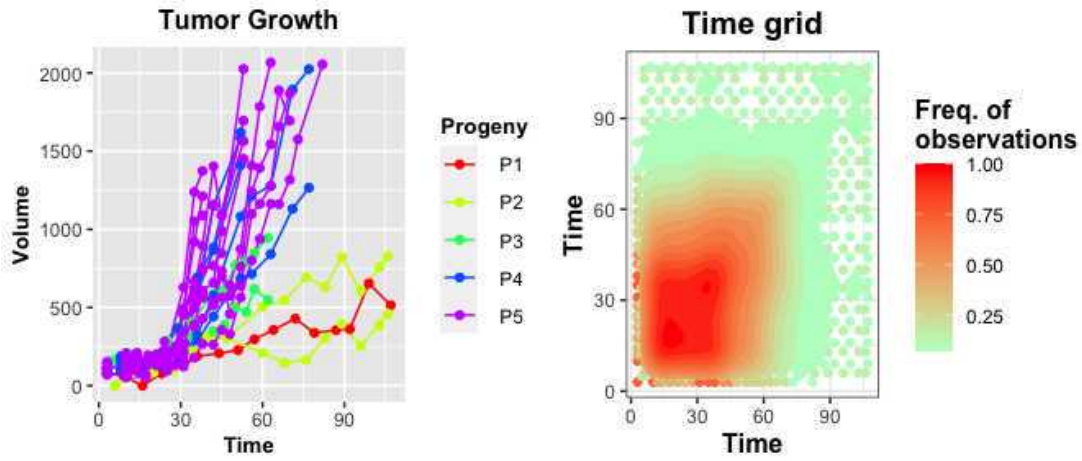

Fig.2 Sampled curves, coloured by progeny feature (left panel) and time grid density (right panel).

In Figure 2 the time grid density is showed. Each point $p_{(x,y)}$ is defined by a pair of coordinates $p_{(x,y)} = (x,y)$ and by a colour. $p_{(x,y)}$ is defined if only if exists at least one sample with two observations at time x and y. The colour encodes the number of samples in which $p_{(x,y)}$ is reported. In our example, according to Figure 2 we decide to truncate the observations at 70 days.

```
# Data truncation

trCONNECTORList<-DataTruncation(data = CONNECTORList,
                                feature="Progeny",
                                truncTime = 70,
                                labels = c("Time","Volume","Tumor
Growth"))

## ################################################
## ######## Summary of the trunc. data ############
##
##  Number of curves: 21 ;
##  Min curve length:  7 ; Max curve length:  16 ;
##
##  Number of truncated curves: 6 ;
##  Min points deleted:  2 ; Max points deleted:  6 ;
## ################################################
```

The output of the function *DataTruncation* is a list of six objects reporting the truncated versions of the data importing functions.
Moreover, the plot stored in

Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

```
trCONNECTORList$PlotTimeSeries_plot
```

shows the sampled curves plot with a vertical line indicating the truncation time, see Figure 3 .

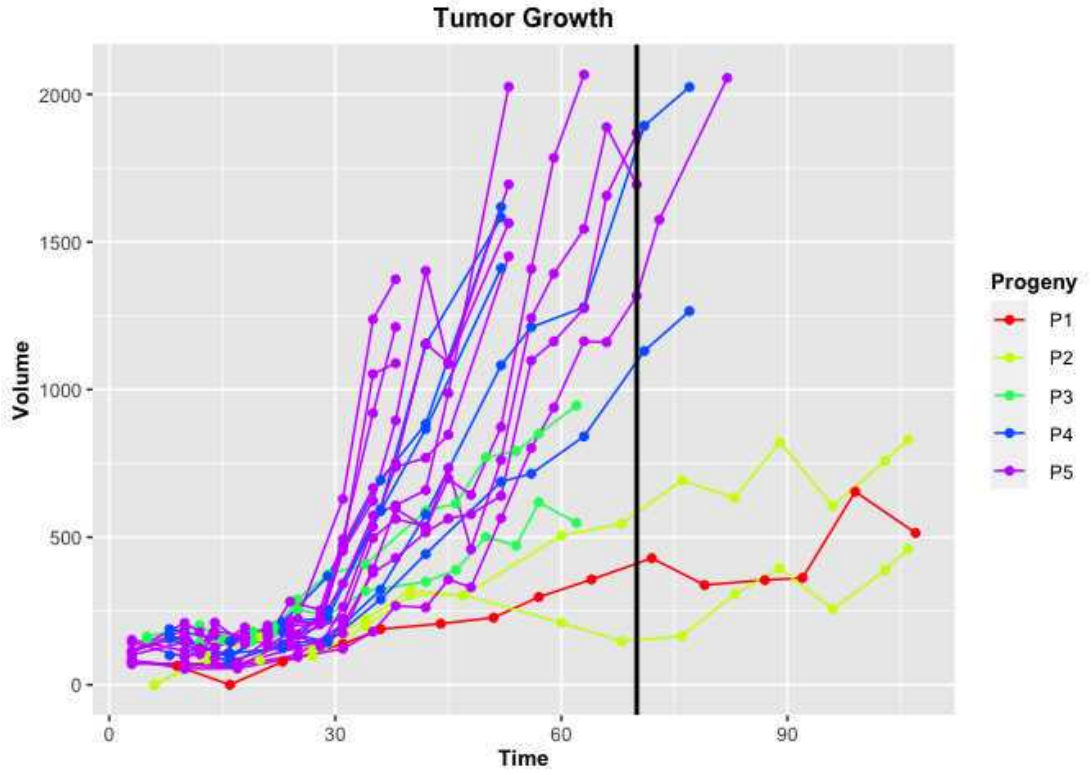

Fig. 3 Sampled curves, coloured by progeny feature and truncation lag (vertical solid black line).

Model Selection Tools

6 Model Selection Tools

Before running the fitting and clustering approach, we have to properly chose the two free parameters:

1. the spline basis dimension, p;
2. the number of clusters, G.

We developed several functions to enable the user to properly set these free parameters.

7 The spline basis dimension - p

The dimension of the spline basis can be chose by exploiting the *BasisDimension.Choice* function, by taking the
$p \in [p_{min}, p_{max}]$ value corresponding to the largest cross-validated likelihood, as proposed in

Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

James, Gareth M and Hastie, Trevor J and Sugar, Catherine A. Principal component models for sparse functional data. Biometrika. https://doi.org/10.1093/biomet/87.3.587

Where the interval of values $[p_{min}, p_{max}]$ is given by the user. In particular, a ten-fold cross-validation approach is implemented: firstly the data are split into 10 equal-sized parts, secondly the model is fitted considering 9 parts and the computation of the log-likelihood on the excluded part is performed.

In details two plots are returned:

1. The *$CrossLogLikePlot* return the plot of the mean tested log-likelihoods versus the dimension of the basis, see Figure 4. Each gray dashed line corresponds to the cross-log-likelihood values obtained on different test/learning sets and the solid black line is their mean value. The resulting plot could be used as a guide to choose the largest cross-validated likelihood. Specifically, the optimal value of p is generally the smallest ensuring the larger values of the mean cross-log-likelihood function.
2. The *$KnotsPlot* shows the time series curves (top) together with the knots distribution over the time grid (bottom), see Figure 5. The knots divide the time domain into contiguous intervals, and the curves are fitted with separate polynomials in each interval. Hence, this plot allows the user to visualize whether the knots properly split the time domain considering the distribution of the sampled observations. The user should choose p such that the number of cubic polynomials (i.e., $p - 1$) defined in each interval is able to follow the curves dynamics.

```
# ten-fold cross-validation
CrossLogLike<-BasisDimension.Choice(data = trCONNECTORList,
                                    p = 2:6 )
CrossLogLike$CrossLogLikePlot
```


Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

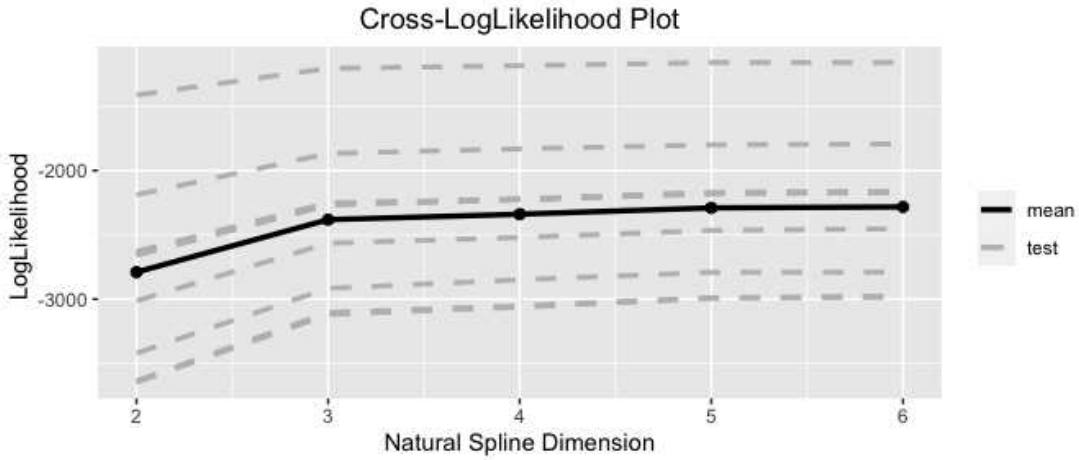

Fig. 4 Cross-validated loglikelihood functions.

```
CrossLogLike$KnotsPlot
```

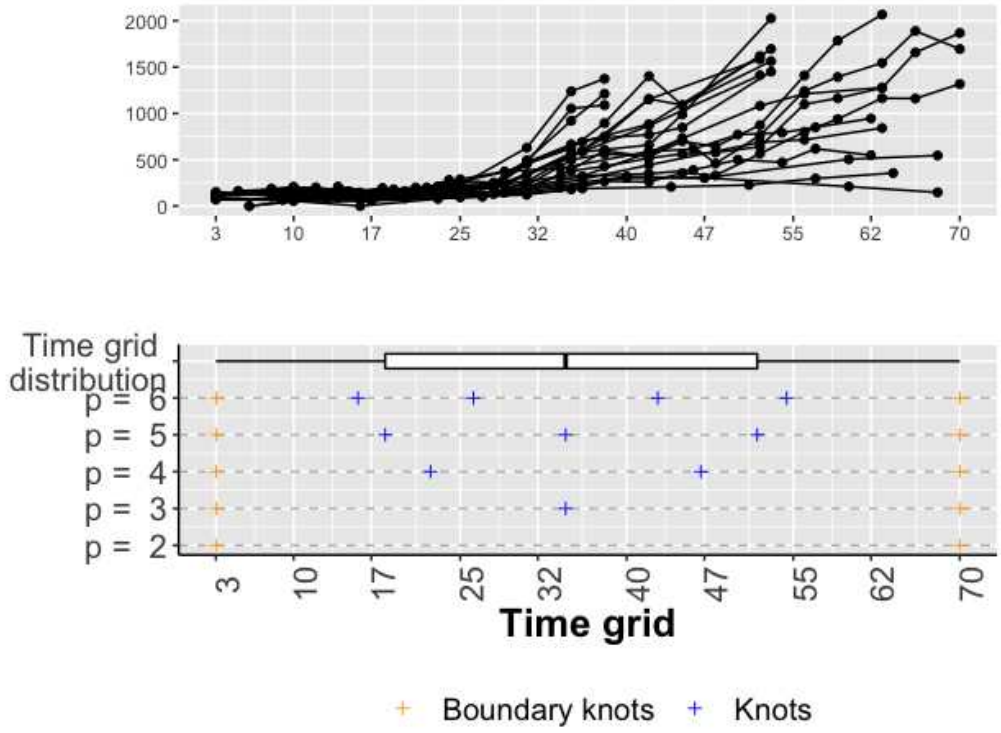

Fig. 5 Knots distribution.

In our example, the optimal value of p is 3.

8 The number of clusters - G

CONNECTOR provides two different plots to properly guide to set the number of clusters.

Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

Two measures of proximity are introduced:

1. the *total tightness (T)* and the *functional Davied-Bouldin index (fDB):* as the number of clusters increases, the total tightness decreases to zero, the value which is attained when the number of fitted clusters is equal to the number of sampled curves;
2. *the functional David Bouldin* (fDB) index: it is a cluster separation measure.

A proper number of clusters can be inferred as large enough to let the total tightness drop down to little values over which it does not decrease substantially. Hence, we look for the location of an *elbow* the plot of the total tightness against the number of clusters. Furthermore, the optimal choice of clusters, then, will be that which minimizes this average blend.

Let us note that the random initialization of the k-means algorithm to get initial cluster memberships into the FCM algorithm and the stochasticity characterizing the method lead to some variability among different runs. For these reasons, multiple runs are necessary to identify the most frequent clustering fixed a number of clusters (*G*).
To effectively take advantage of those two measures, **CONNECTOR** supplies the function *ClusterAnalysis* which repeats the clustering procedure a number of times equal to the parameter *runs* and for each of the number of clusters given in the parameter *G*.

```
ClusteringList <-ClusterAnalysis(data = trCONNECTORList,
                                 G = 2:5,
                                 p = p,
                                 runs = 100)
```

The output of the function is a list of three objects

- (i) *$Clusters.List* : the list of all the clustering divisions obtained varying among the input *G* values,
- (ii) *$seed* : the seed sets before running the method,
- (iii) *$runs*: the number of runs.

Specifically, the object storing the clustering obtained with *G* = 2 (i.e., *ClusteringList$Clusters.List$G2*) is a list of three elements:

1. *$ClusterAll*: the list of all the possible FCM parameters values that can be obtained through each run. In details, the item *$ParamConfig.Freq* reports the number of times that the respectively parameters configuration (stored in *$FCM*) is found. Let us note, that the sum might be not equal to the number of runs since errors may occur;

2. *$ErrorConfigurationFit*: list of errors that could be obtained by running the FCM method with extreme parameters configurations;

3. *$h.selected*: the dimension of the mean space, denoted as *h*, selected to perform the analysis. In details, this parameter gives a further parametrization of the mean curves allowing a lower-dimensional representation of the curves with means in a restricted

Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

subspace. Its value is choose such that the inequality $h \leq \min(G-1, p)$ holds. Better clusterings are obtained with higher values of h, but this may lead to many failing runs. In this cases h values is decreased until the number of successful runs are higher than a specific constraint which can be defined by the user (by default is 100% of successful runs).

```
# the output structure

str(ClusteringList, max.level = 2, vec.len=1)

## List of 4
##  $ Clusters.List:List of 4
##   ..$ G2:List of 2
##   ..$ G3:List of 2
##   ..$ G4:List of 2
##   ..$ G5:List of 2
##  $ CONNECTORList:List of 6
##   ..$ Dataset            :'data.frame':  241 obs. of  3
variables:
##   ..$ LenCurv            : num [1:21] 9 9 ...
##   ..$ LabCurv            :'data.frame':  21 obs. of  5
variables:
##   ..$ TimeGrid           : num [1:50] 3 5 ...
##   ..$ ColFeature         : chr [1:5] "#FF0000" ...
##   ..$ PlotTimeSeries_plot:List of 9
##   .. ..- attr(*, "class")= chr [1:2] "gg" ...
##  $ seed         : num 2404
##  $ runs         : num 100
str(ClusteringList$Clusters.List$G2, max.level = 3, vec.len=1)
## List of 2
##  $ ClusterAll:List of 2
##   ..$ :List of 2
##   .. ..$ FCM             :List of 4
##   .. ..$ ParamConfig.Freq: int 43
##   ..$ :List of 2
##   .. ..$ FCM             :List of 4
##   .. ..$ ParamConfig.Freq: int 57
##  $ h.selected: num 1
```

The function *IndexesPlot.Extrapolation* generated two plots reported in Figure 6. In details, the tightness and fDB indexes are plotted for each value of G. Each value of G is associated with a violin plot created by the distribution of the index values collected from the runs performed. The blue line shows the most probable configuration which
will be exploited hereafter.

```
IndexesPlot.Extrapolation(ClusteringList)-> indexes
```


Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

```
indexes$Plot
```

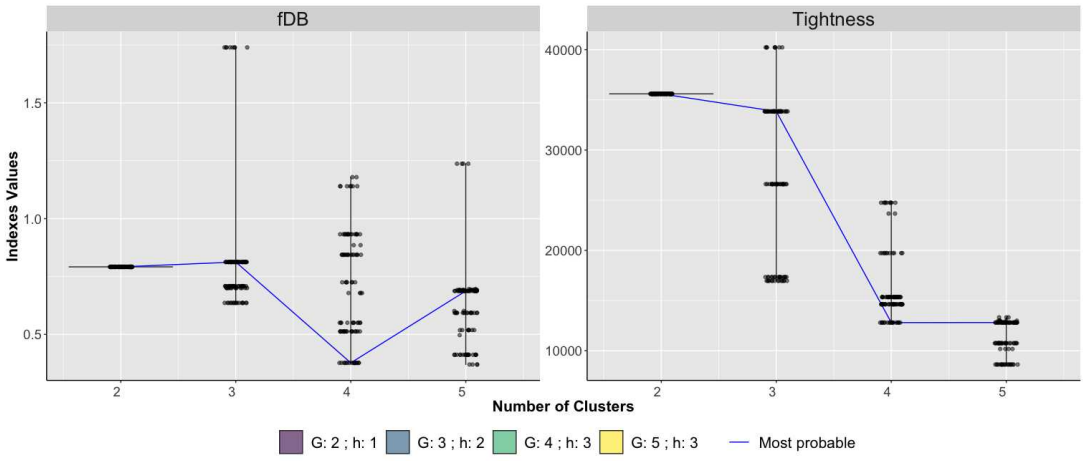

Fig. 6 Violin Plots of the total tightness (T) calculated on each run and for different number of clusters G (right panel). Violin Plots of the fDB calculated on each run and for different number of clusters G (left panel)

G = 4 corresponds to the optimal choice for our example, since it represents the elbow for the tightness indexes (right panel) and explicitly minimizes the fDB indexes (left panel).The variability of the two measures among runs, exhibited in Figure 6 , is related to the random initialization of the k-means algorithm to get initial cluster memberships from points. The stability of the clustering procedure can be visualized through the stability matrix extrapolated by the function *ConsMatrix.Extrapolation*, as shown in Figure 7. The plot informs about the stability of the final clustering across different runs. Indeed, each cell of the matrix is coloured proportionally to the frequency of the two corresponding curves belonging to the same cluster across different runs. Hence the larger the frequencies are (which corresponds to warmer colours of the cells in the plot), the more stable is the final clustering. For each cluster the average stability value is reported.

```
ConsMatrix<-ConsMatrix.Extrapolation(stability.list =
ClusteringList)

str(ConsMatrix, max.level = 2, vec.len=1)

## List of 4
##  $ G2:List of 2
##   ..$ ConsensusMatrix: num [1:21, 1:21] 1 1 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   ..$ ConsensusPlot  :List of 9
##   .. ..- attr(*, "class")= chr [1:2] "gg" ...
##  $ G3:List of 2
##   ..$ ConsensusMatrix: num [1:21, 1:21] 1 1 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   ..$ ConsensusPlot  :List of 9
##   .. ..- attr(*, "class")= chr [1:2] "gg" ...
##  $ G4:List of 2
```


Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

```
##   ..$ ConsensusMatrix: num [1:21, 1:21] 1 1 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   ..$ ConsensusPlot  :List of 9
##   .. ..- attr(*, "class")= chr [1:2] "gg" ...
##  $ G5:List of 2
##   ..$ ConsensusMatrix: num [1:21, 1:21] 1 0.81 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   ..$ ConsensusPlot  :List of 9
##   .. ..- attr(*, "class")= chr [1:2] "gg" ...

ConsMatrix$G4$ConsensusPlot
```

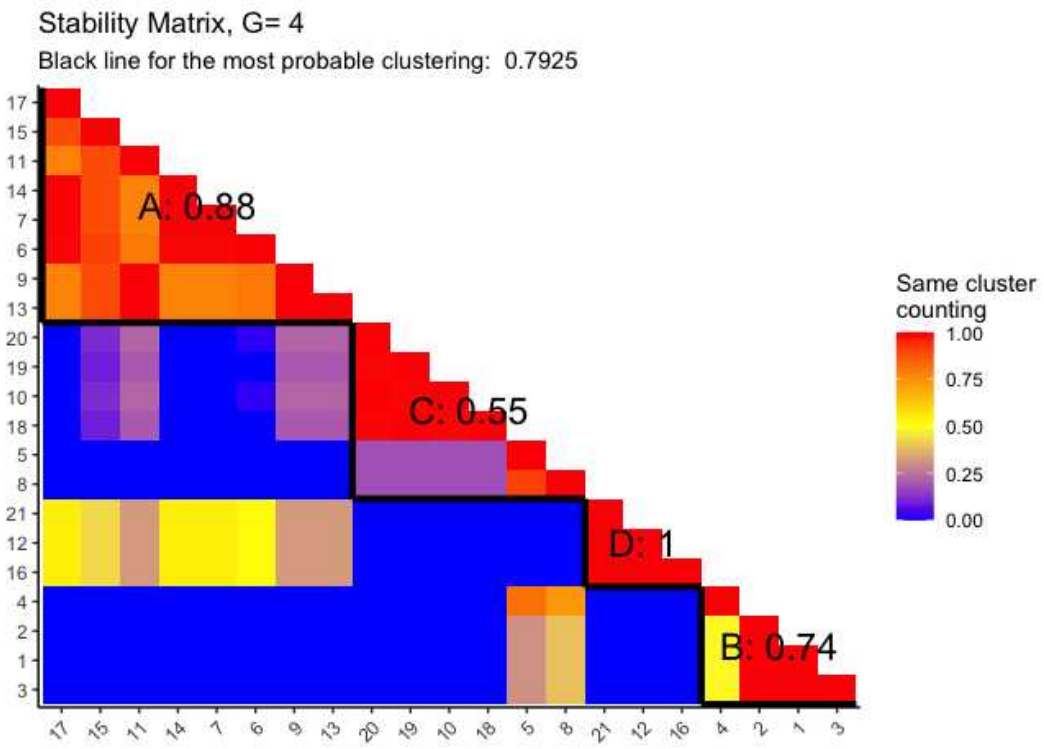

Fig. 7 Stability Matrix for G = 4.

Once the free parameters are all set, the function *MostProbableClustering.Extrapolation* can be used to fix the most probable clustering with given dimension of the spline basis *p*, and number of clusters *G* and save the result in a dedicated object.

```
CONNECTORList.FCM.opt <- MostProbableClustering.Extrapolation(
                                                stability.list =
ClusteringList,
                                                G = 4 )
```

Output

Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

9 Output

The output of **CONNECTOR** consists of the line plot of the samples grouped by the **CONNECTOR** cluster, the discriminant plot, the discriminant function, and the estimation of the entire curve for each single subject.

10 CONNECTOR clusters

CONNECTOR provides the function *ClusterWithMeanCurve* which plots the sampled curves grouped by cluster membership, together with the mean curve for each cluster, see Figure 8 .

The function prints as well the values for S_{k1} (the total standard deviation of the cluster k1), $M_{k1,k2}$ (the distance between mean-curves of k1--th and k2--th cluster), R_{k1} (a measure of how good the cluster is) and fDB indexes. The _1 and _2 denote the first and second derivatives respectively of the corresponding index.

```
FCMplots<- ClusterWithMeanCurve(clusterdata =
CONNECTORList.FCM.opt,
                                feature = "Progeny",
                                labels = c("Time","Volume"),
                                title = "FCM model")
##
## ######## S indexes ############
##
##
## |          |        S|       S_1|      S_2|
## |:---------|--------:|---------:|--------:|
## |Cluster B | 1702.753| 114.57314| 5.128564|
## |Cluster A |  654.418|  29.34886| 1.021555|
## |Cluster C | 2791.051| 136.25605| 6.430096|
##
## ############################################################
## ############################################################
## ######## M indexes ############
##
##
## |          | Cluster B| Cluster A| Cluster C|
## |:---------|---------:|---------:|---------:|
## |Cluster B |     0.000|  2851.265|  5731.600|
## |Cluster A |  2851.265|     0.000|  8831.041|
## |Cluster C |  5731.600|  8831.041|     0.000|
##
## ############################################################
## ######## R indexes ############
##
```


Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

```
##
## |          |         R|       R_1|      R_2|
## |:---------|---------:|---------:|--------:|
## |Cluster B | 0.8267106| 1.0914505| 1.530825|
## |Cluster A | 0.8267106| 1.0914505| 1.530825|
## |Cluster C | 0.7840400| 0.8803222| 1.255647|
##
## ################################################################
## ######## fDB indexes ############
##
##
## |       fDB|    fDB_1|    fDB_2|
## |---------:|--------:|--------:|
## | 0.8124871| 1.021074| 1.439099|
##
## ################################################################
```

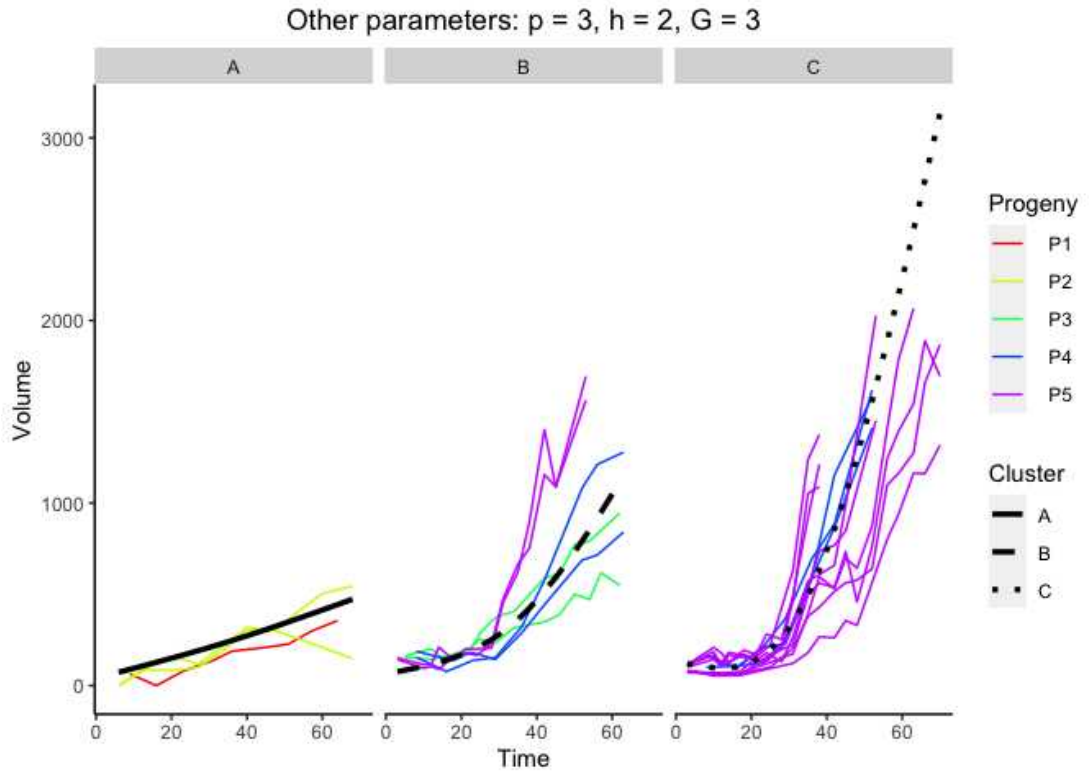

Fig. 8 Sampled curves grouped by cluster membership.

Finally, to inspect the composition of the clusters, the function *CountingSamples* reports the number and the name of samples in each cluster according to the feature selected by the user.

```
NumberSamples<-CountingSamples(clusterdata = CONNECTORList.FCM.opt,
```


Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

```
                                feature = "Progeny")

str(NumberSamples, max.level = 2)
## List of 2
##  $ Counting    :'data.frame':    15 obs. of  3 variables:
##   ..$ Cluster: chr [1:15] "A" "A" "A" "A" ...
##   ..$ Progeny: Factor w/ 5 levels "  P1","  P2",..: 1 2 3 4 5 1
2 3 4 5 ...
##   ..$ n      : int [1:15] 1 2 0 0 0 0 0 2 2 2 ...
##  $ ClusterNames:'data.frame':    21 obs. of  2 variables:
##   ..$ ID     : int [1:21] 1 2 3 4 5 6 7 8 9 10 ...
##   ..$ Cluster: chr [1:21] "A" "A" "A" "B" ...
```

11 Discrimination Plot

A detailed visualization of the clusterization of the sampled curves can be obtained by means of the *DiscriminantPlot* function.
The low-dimensional plots of curve datasets, enabling a visual assessment of clustering. The curves can be projected the into the lower dimensional space of the mean space, in this manner the curve can be plot as points (with coordinates the functional linear discriminant components).
In details, when h is equal to 1 or 2, than the *DiscriminantPlot* function return the plots of the curves projected onto the h dimensional mean space:

1. when h = 1, the functional linear discriminant α_1 is plotted versus its variability;
2. when h = 2, the functional linear discriminant α_1 is plotted versus the functional linear discriminant α_2.

If h > 2, the principal component analysis is exploited to extrapolate the components of the h-space with more explained variability (which is reported in brackets), that are then plotted together.

In the case study here described, we get the plots in Figure 9 which is colored by cluster membership and in Figure 10 which is colored by the *"Progeny"* feature.

```
DiscrPlt<-DiscriminantPlot(clusterdata = CONNECTORList.FCM.opt,
                           feature = "Progeny")
DiscrPlt$ColCluster
```


Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

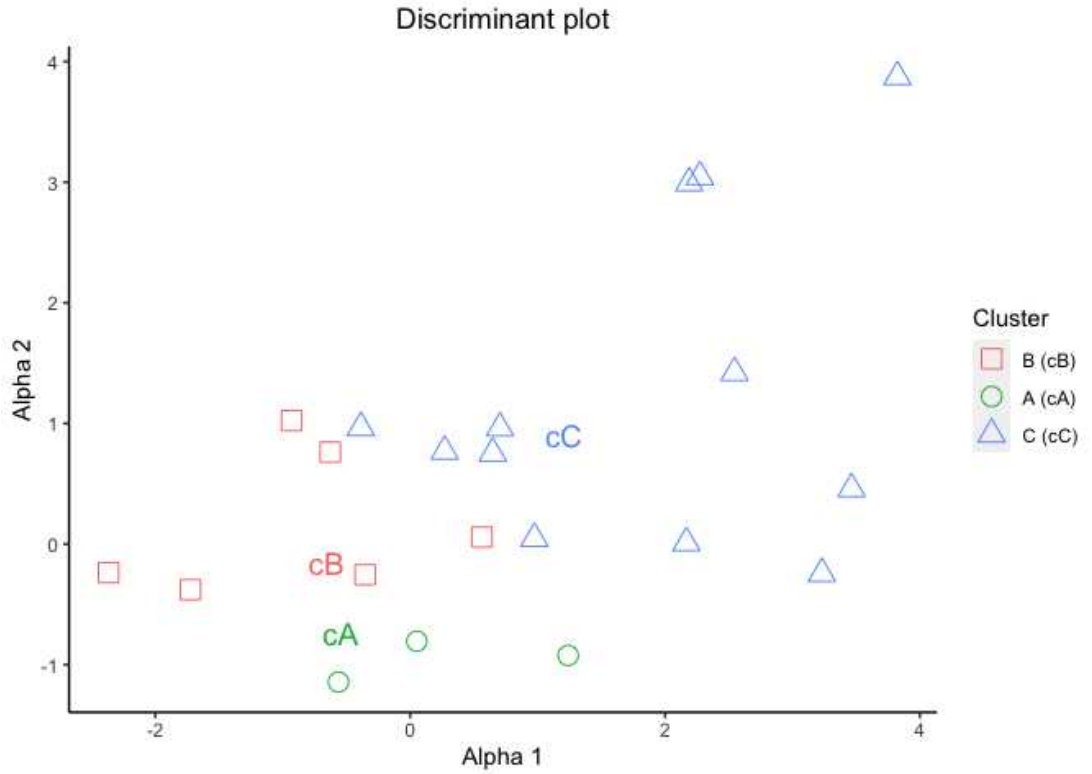

Fig. 9 Curves projected onto the 2-dimensional mean space: symbols are coloured by cluster membership.

```
DiscrPlt$ColFeature
```


Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

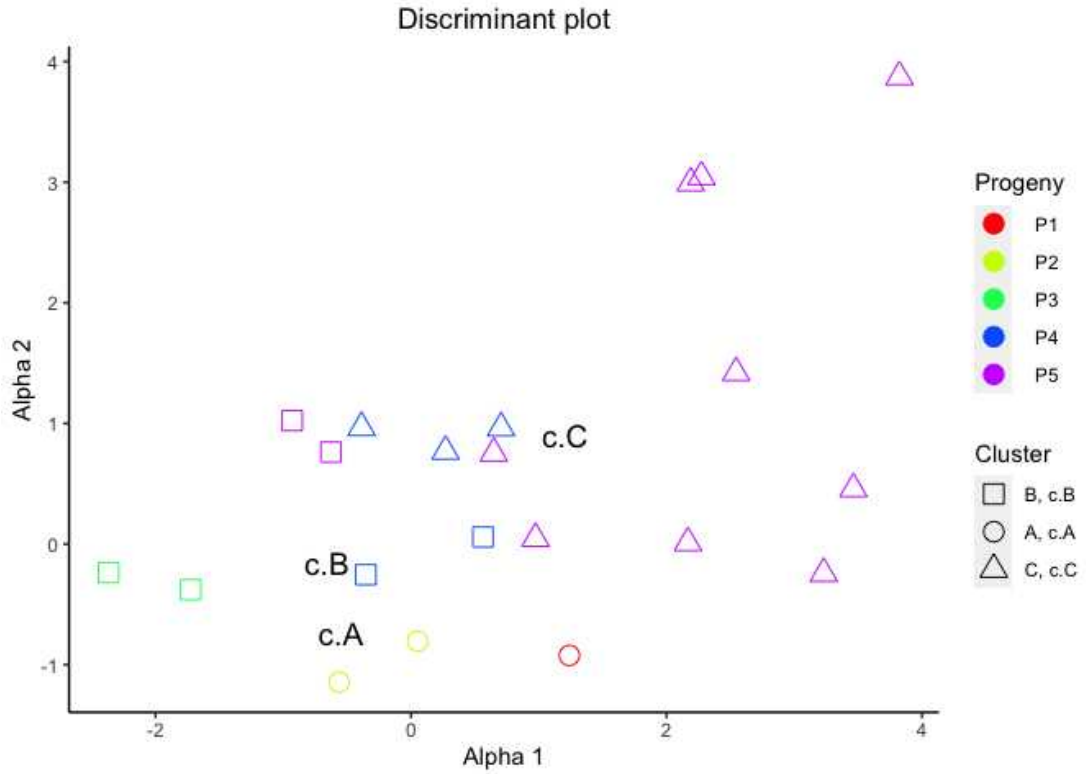

Fig. 10 Curves projected onto the 2-dimensional mean space: symbols are coloured by progeny.

12 Discrimination Function

There will be *h* discriminant functions and each curve shows the times with higher discriminatory power, which are the times corresponding to largest absolute (positive or negative) values on the y-axis, see Figure 11.

.

```
MaxDiscrPlots<-MaximumDiscriminationFunction(clusterdata =
CONNECTORList.FCM.opt)

MaxDiscrPlots[[1]]
```


Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

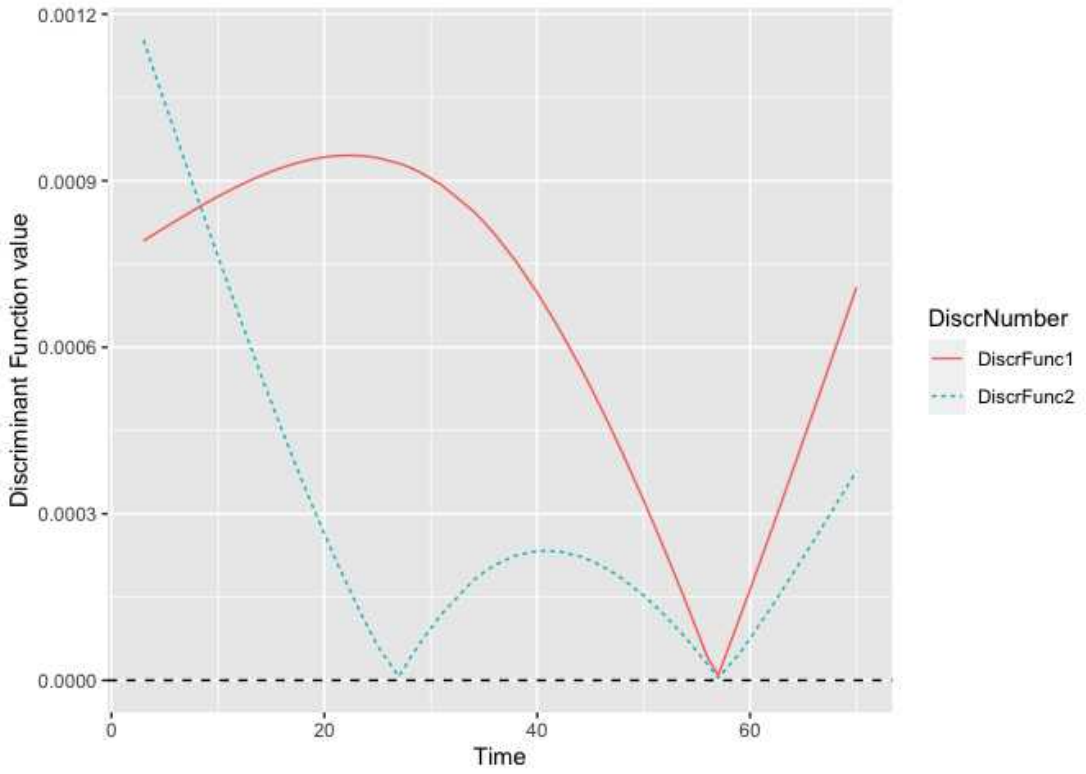

Fig. 11 Discriminant curves.

Large absolute values correspond to large weights and hence large discrimination between clusters. From the Figure 11 it is possible assess that earlier measurements are crucial in determining cluster assignment as well as latter ones.

13 Estimated Curve

The functional clustering procedure predict unobserved portions of the true curves for each subject. The estimated curves are returned by **CONNECTOR** and plotted with confidence intervals as well that is possible to visualize using the *Spline.plots* function.

```
PlotSpline = Spline.plots(FCMplots)

PlotSpline$*1*
```


Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1

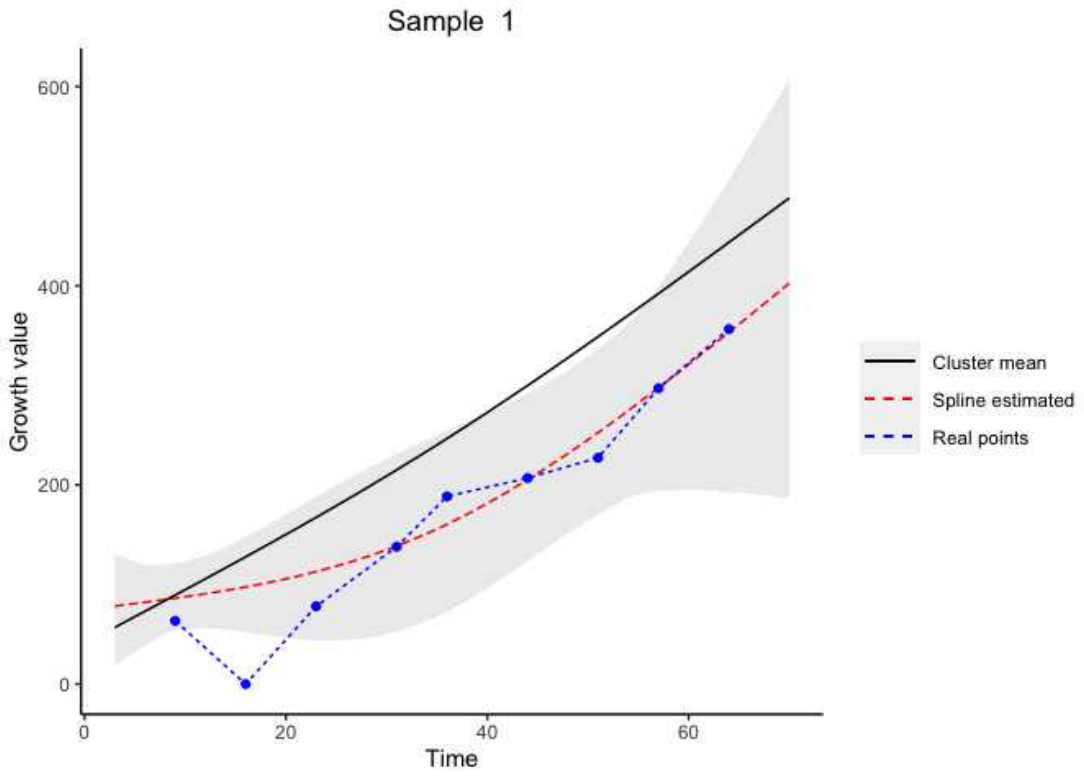

Fig. 12 Fitting of the Sample with ID = 1: in blue the observations characterinzing the curve, in red the estimated spline from the fitting, and in black the mean curve of the associated cluster. The grey area represents the confidence interval.

Citation: Simone Pernice, Roberta Sirovich, Elena Grassi, Marco Viviani, Martina Ferri, Francesco Sassi, Luca Alessandri', Dora Tortarolo, Raffaele A. Calogero, Livio Trusolino, Andrea Bertotti, Marco Beccuti, Martina Olivero, Francesca Cordero CONNECTOR, fitting and clustering of longitudinal data. https://dx.doi.org/10.17504/protocols.io.8epv56e74g1b/v1