plotFPOP Vignette

Vincent Runge

LaMME Université d'Evry. October 22, 2017

Introduction

`plotFPOP` is a package providing plotting functions for dynamic programming changepoint algorithms with penalty using functional pruning. The package provides simple (written in R) functions and algorithms to plot the functional cost in 1d and 2d at consecutive times. The goal is to give an easy and simple access to the shape of the functional cost in order to better understand its updates. The package also contains simulation functions to generate segmented data under a gaussian law in dimension 1 and 2.

This vignette describes the usage of `plotFPOP` in R.

First, let us install the package:

```
#install.packages("plotFPOP")
```

and load it:

```
library(plotFPOP)
```

Data simulation

We simulate 1d data with the function dataG1

```
dataG1(mean = c(0.5,1), tau = c(0.3,1), n= 20)
```

```
##  [1]  1.28486024  1.13768263  0.71437132 -2.47158982  0.03544287
##  [6]  2.48156386  0.97020333  1.84294029  2.16281230  1.84624056
## [11] -0.23973259  0.87487750  1.05330642  1.72847655  0.73981162
## [16]  1.11869106  1.62204480  1.18753080  0.84434671  2.18604017
```

This function has two vector arguments : 'mean' to specify the consecutive means on each segment, 'tau' for the relative positions of the last point on each segment. The segmentation is based on gaussian laws with a unique standard deviation 'sigma'. 'n' is the number of data to simulate.

We simulate 2d data with the function dataG1

```
dataG2(mean1 = c(0.5,1), mean2 = c(0,0.5), tau = c(0.3,1), n= 20)
```

```
##            [,1]       [,2]      [,3]       [,4]       [,5]      [,6]
## [1,] -0.6164109  1.3341529 2.2586920  0.5555203 -1.7942769  0.469673
## [2,] -0.8853159 -0.8399116 0.3883085 -0.4982347 -0.4934506 -1.766957
##          [,7]      [,8]      [,9]    [,10]     [,11]       [,12]     [,13]
## [1,] 1.522656 0.5345967 0.6950930 1.326665 2.2836179  0.05897509 1.8555920
## [2,] 1.074624 0.4846147 0.8466814 1.876206 0.7536903 -0.53397958 0.7094074
##           [,14]    [,15]       [,16]    [,17]      [,18]      [,19]
## [1,]  0.2762663 1.582449  0.01629552 2.012360 -0.2327084 -0.6380278
## [2,] -1.4459190 0.154810 -0.94506240 1.365998  1.5401176  0.3952069
```

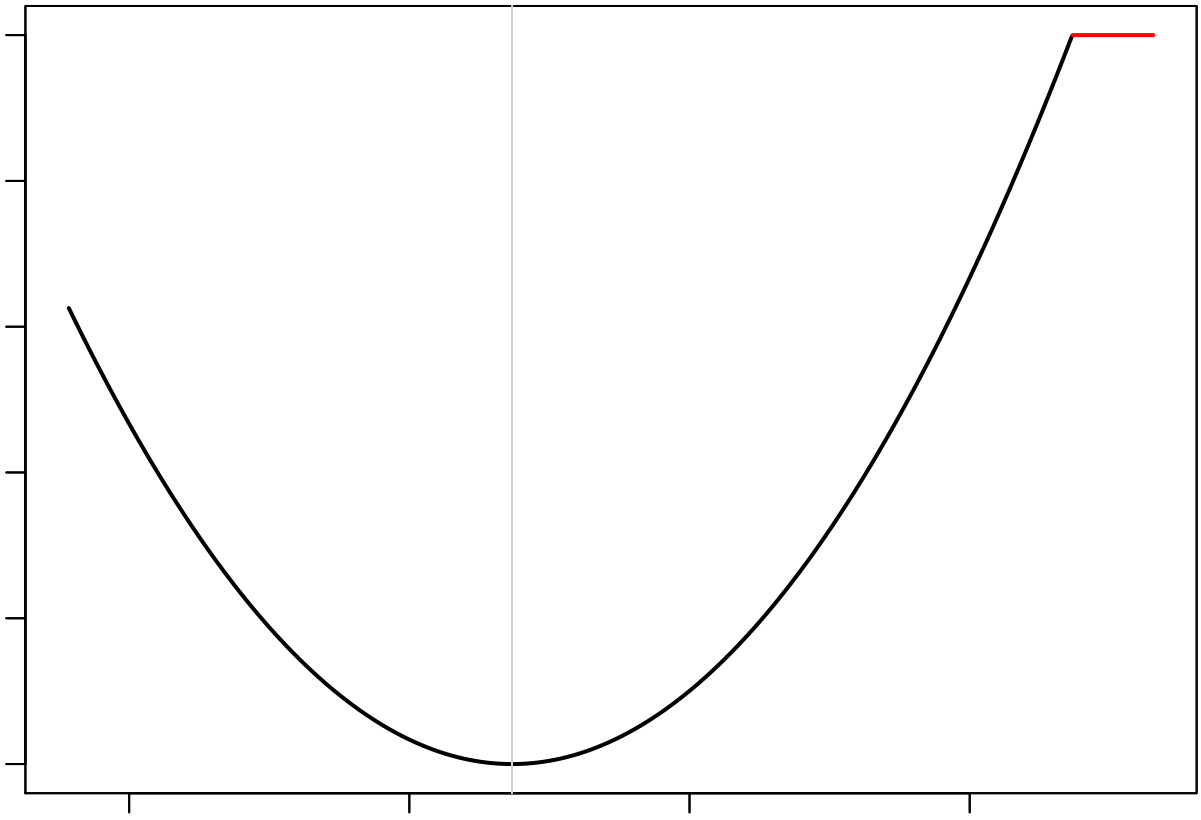
```
##           [,20]
## [1,] -0.4073608
## [2,]  0.5422631
```

With this function, a segmentation in dimension 2 is simulated. Two vectors of means has to be given : 'mean1' and 'mean2'. The other arguments are the same as for dataG1. The obtained segmentation is also based on gaussian distribution with a constant standard deviation.

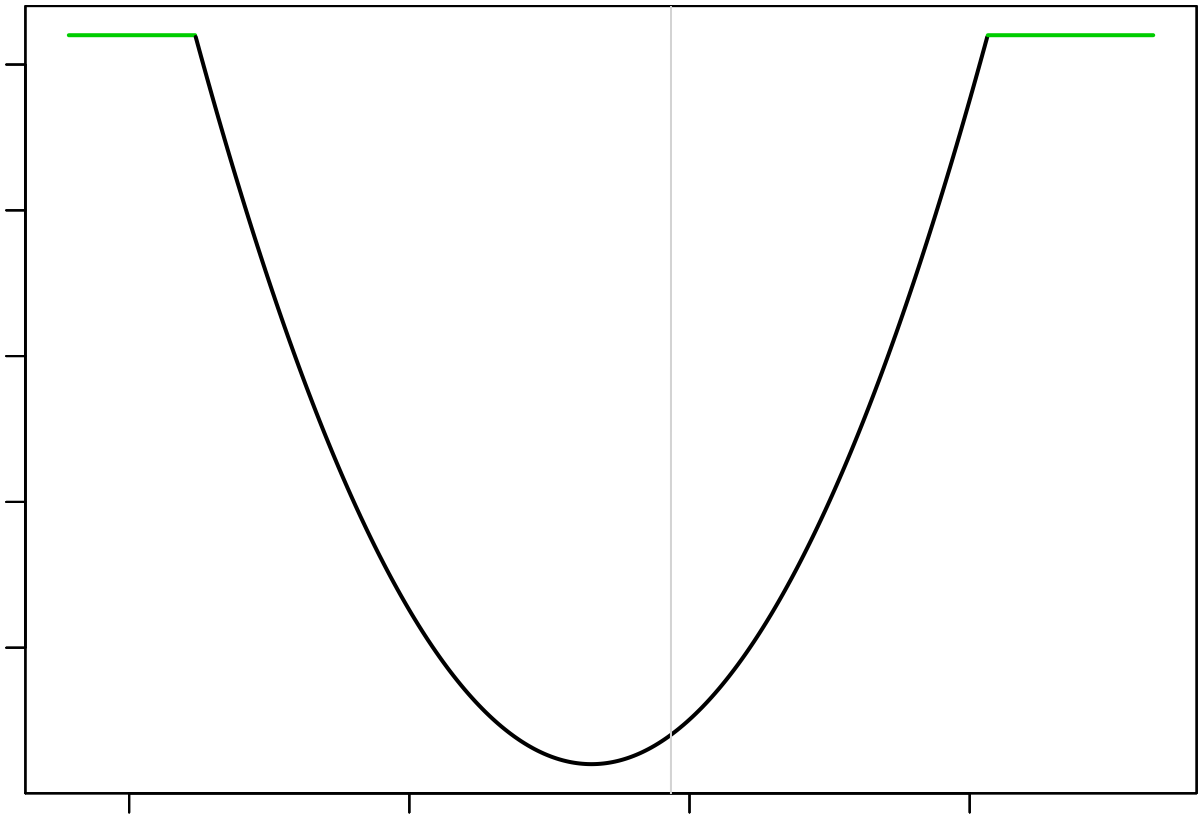
Plots

```
data <- dataG1(sigma = 0.5, n=5)
fpop1d(data,1)
```

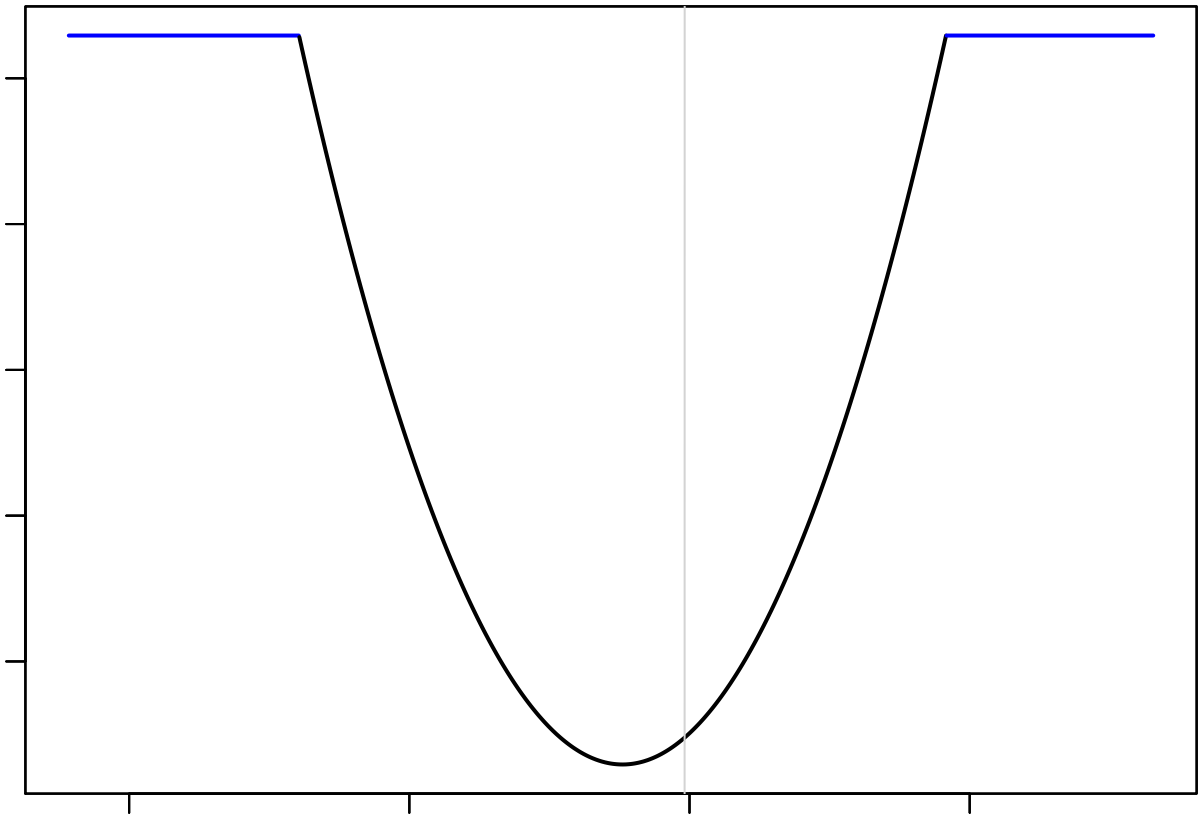
```
## [1] 1 2
```



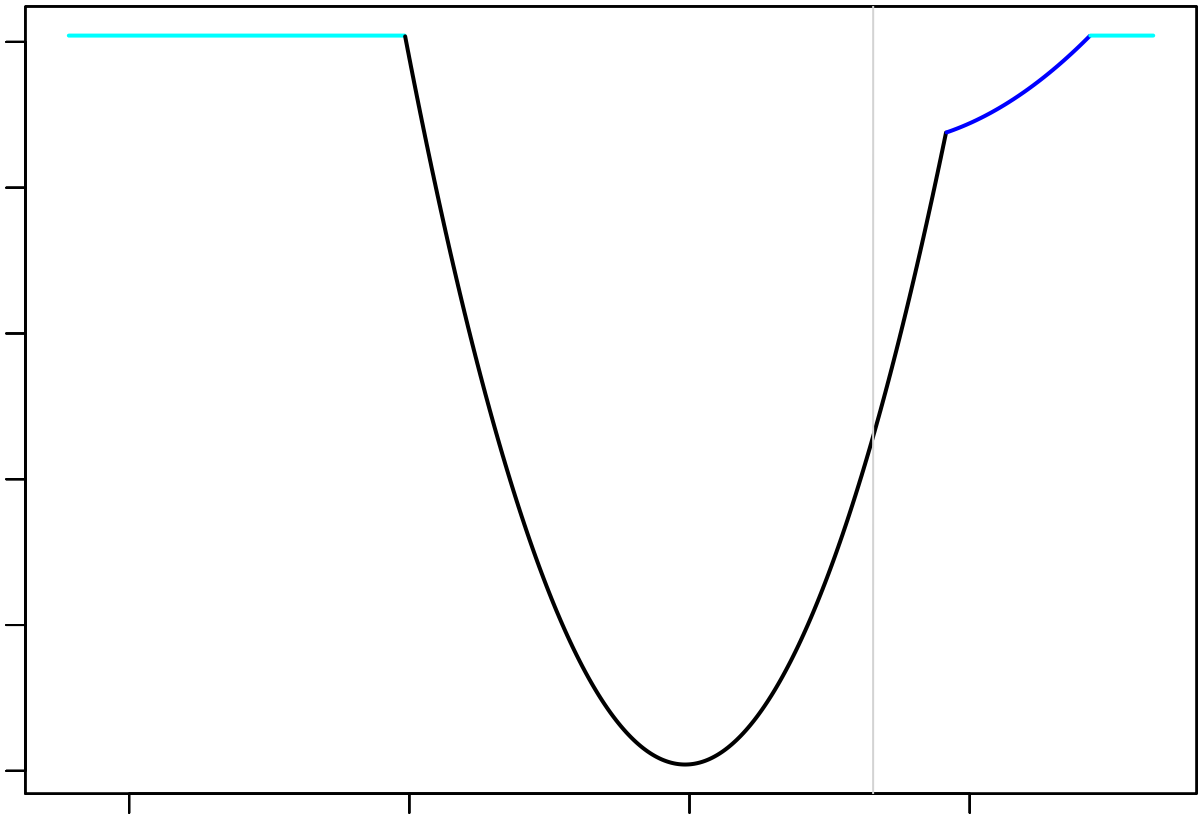
```
## [1] 1 3
```



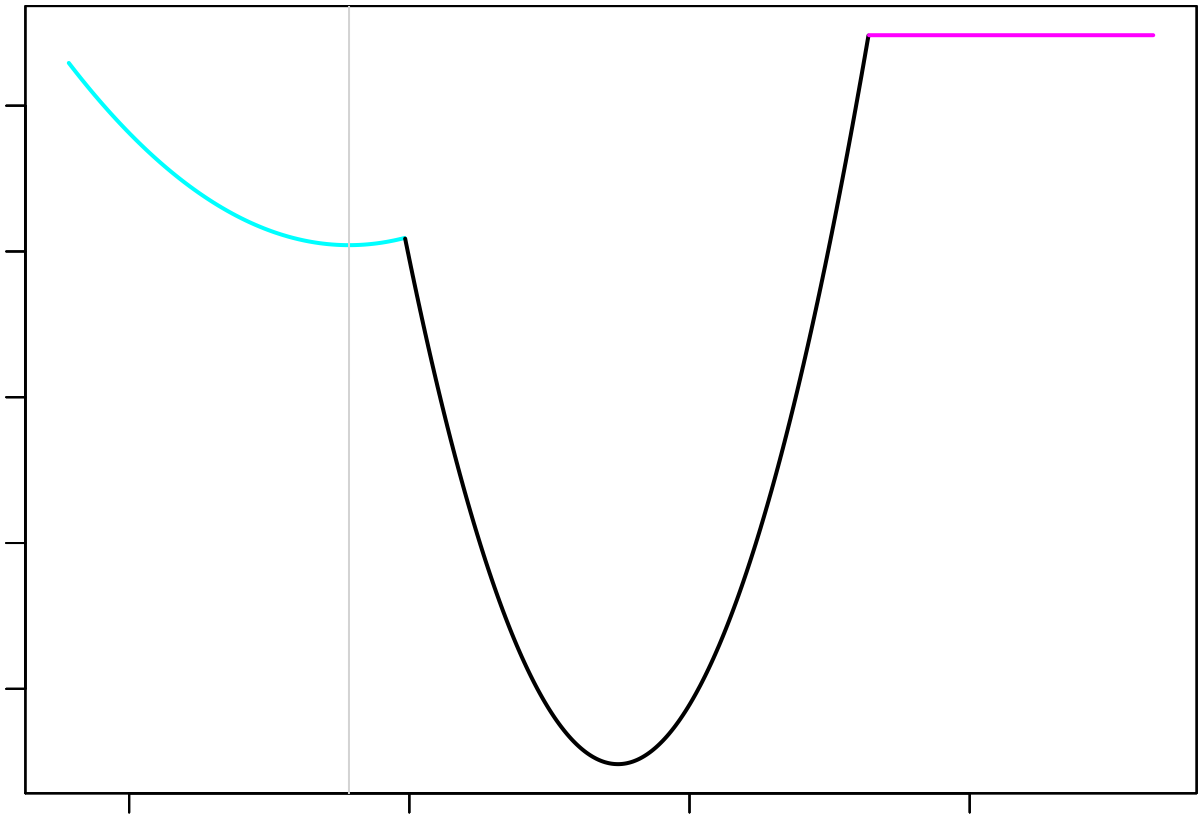
```
## [1] 1 4
```



```
## [1] 1 4 5
```



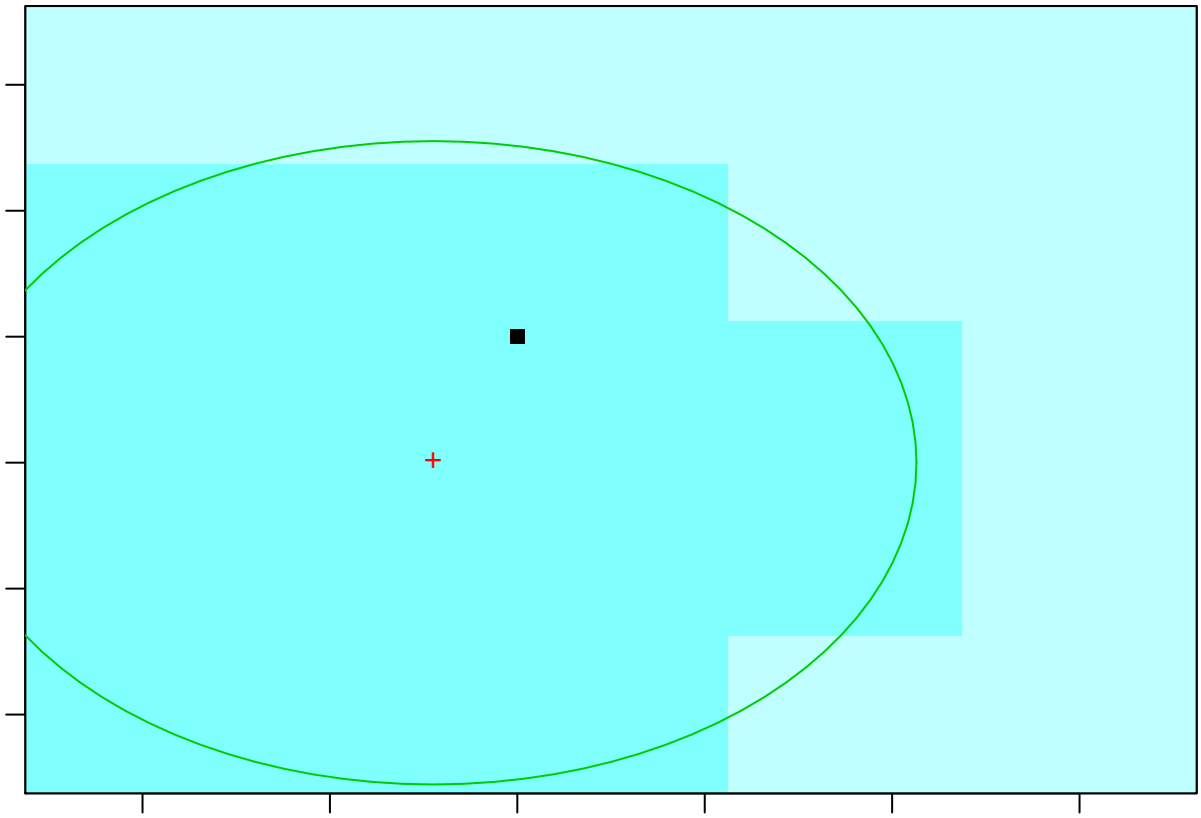
```
## [1] 1 5 6
```



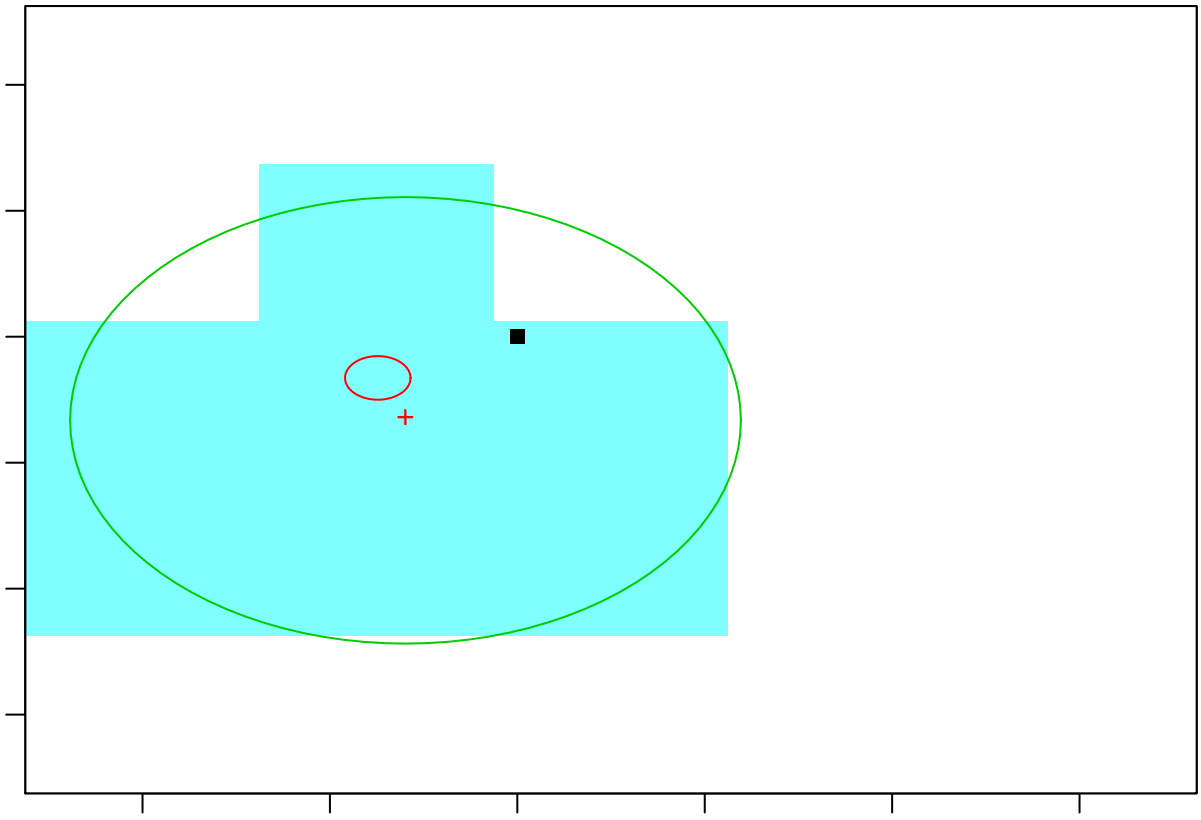
This function ‘fpop1d’ returns consecutive plots and consecutive vectors containing the indexes of present labels for data in dimension 1. The argument ‘beta’ is a positive number: the penalty to use in this penalized version of the dynamic programming algorithm. The last argument ‘order’ is a boolean, if true, the vector of labels gives the labels arising from left to right in the functional cost.

```
data <- dataG2(mean1 = 0, mean2 = 0, tau = 1, sigma = 0.5, n=5)
fpop2d(data, nb = 300, 1, circle = TRUE)
```

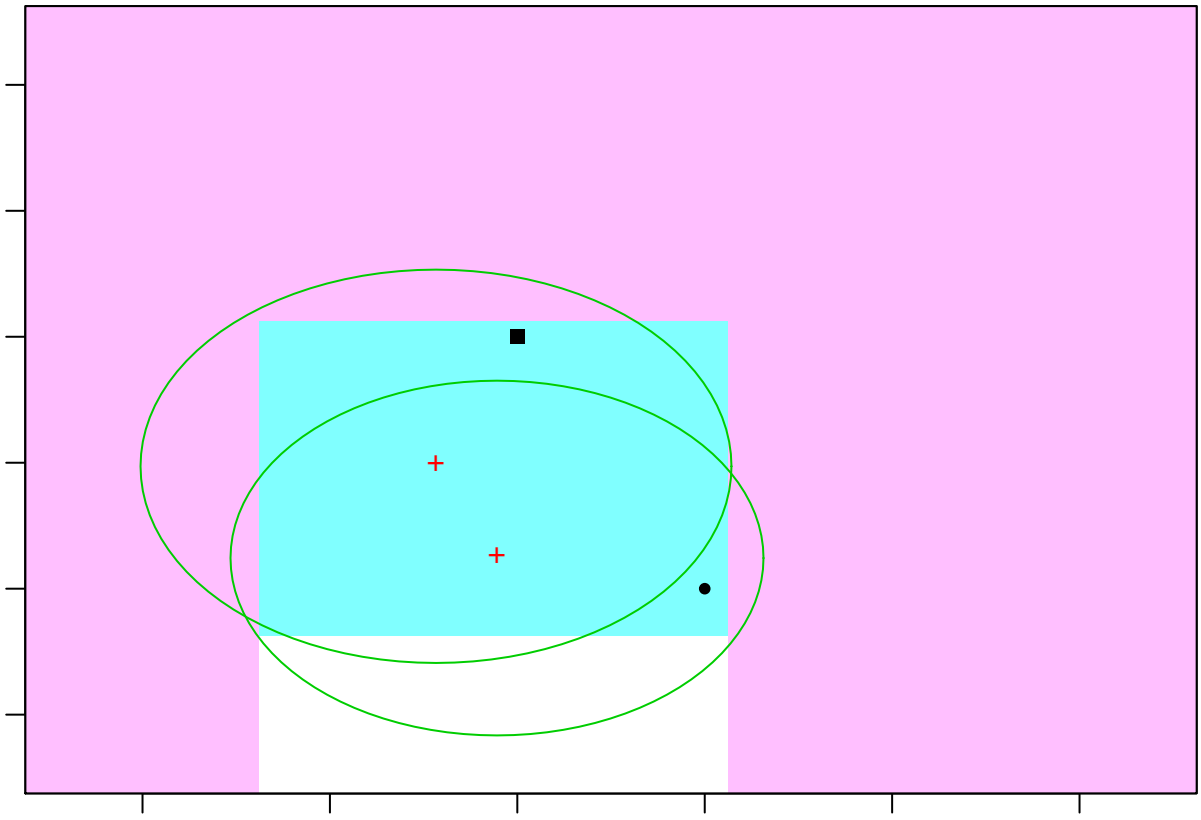
```
## present: 1
```



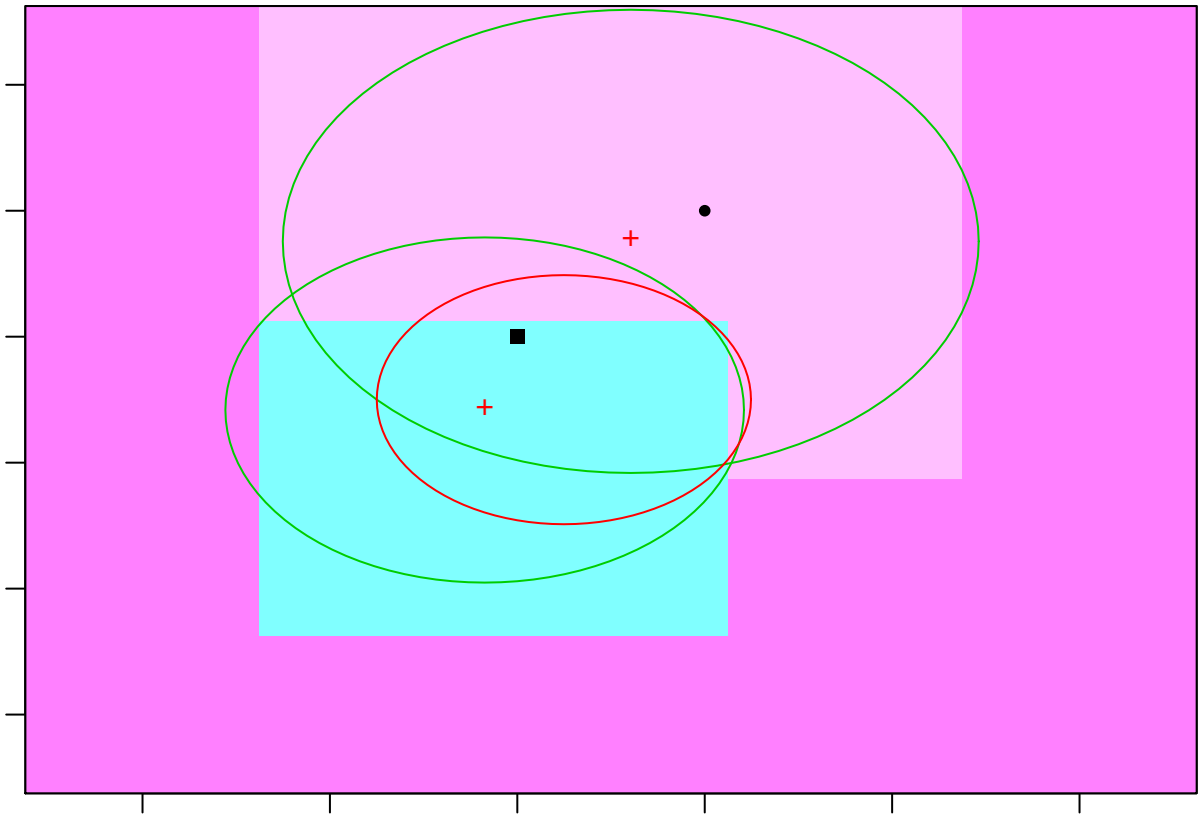
```
## present: 1
```



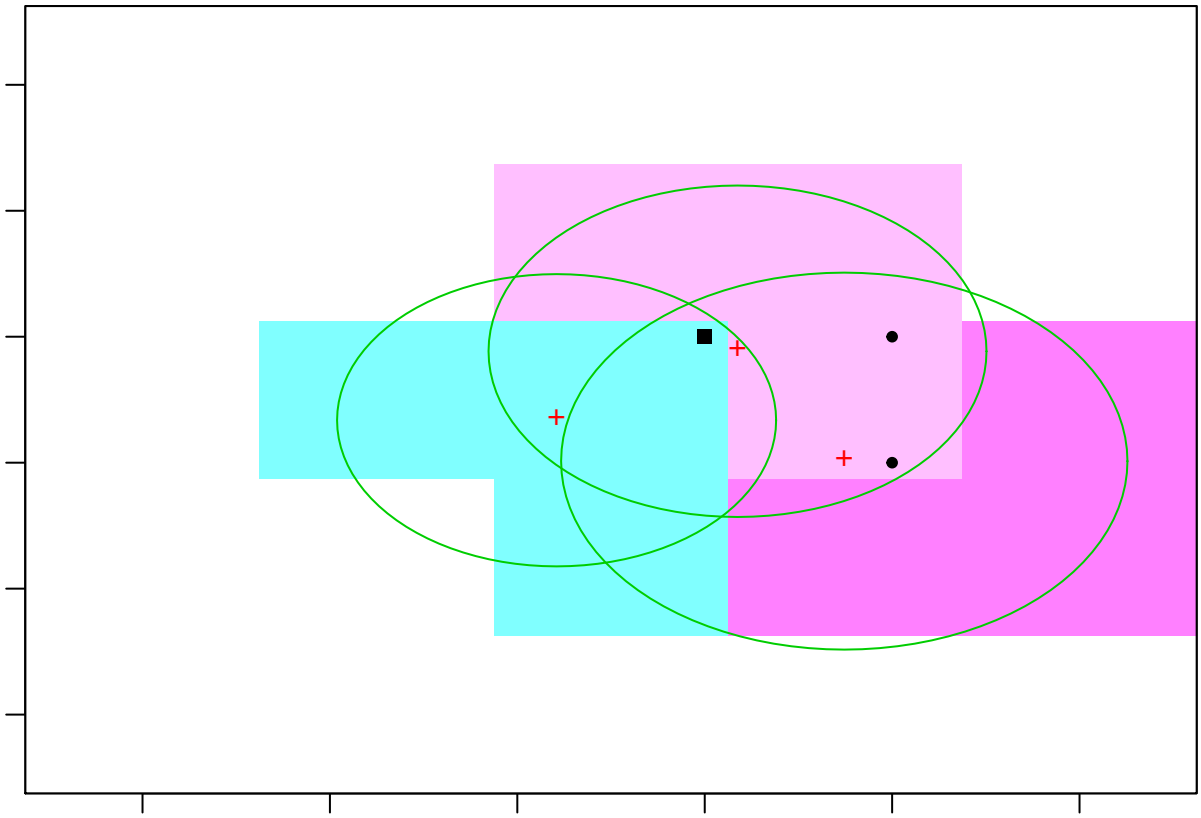
```
## present: 1 3
```



```
## present: 1 4
```



```
## present: 1 4 5
```



This function ‘fpop2d’ shows the shape of the functional cost in dimension 2 at level ‘minimum of this function’ + beta. Each color represents a different label. Argument ‘precision’ is used to force the size of the matrix to plot. The boolean ‘circle’, if set to true, is used to display the circles of intersection at each time step.

Back to Top