flowMeans: Non-parametric Flow Cytometry Data Gating

Nima Aghaeepour

October 18, 2010

naghaeep@bccrc.ca

Contents

1 Licensing

Under the Artistic License, you are free to use and redistribute this software.

2 Overview

We apply a non-parametric approach to perform automated gating of cell populations in flow cytometry data. Our clustering results are obtained with counting the number of modes in every single dimension followed by multidimensional clustering. Then adjacent clusters (in terms of Euclidean or Mahalanobis distance) are merged. The number of clusters is determined using change point detection algorithm based on piecewise linear regression. This approach allows multiple clusters to represent the same population. This enables our framework to find non-spherical cell populations. Development of flowMeans is partly motivated by the advent of high-throughput flow cytometry. Using the k-means algorithm, flowMeans avoids using complex statistical models. This, results in improving the runtime of current model-based population identification methods without decreasing their accuracy.

3 Example

The *flowMeans* function runs the main clustering procedure:

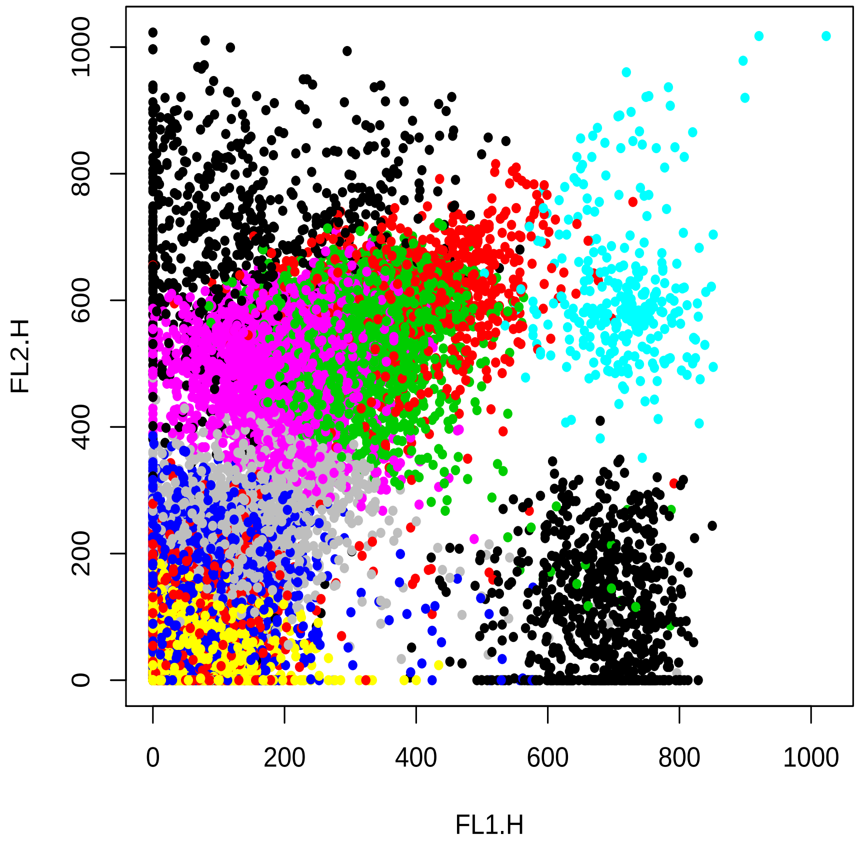
```
> library(flowMeans)

> data(x)
> res <- flowMeans(x, varNames = c("FL1.H", "FL2.H", "FL3.H", "FL4.H"),
+     MaxN = 10)
```

The input arguement *varNames* must contain a vector of parameter names that need to be analysed. By providing the parameter names, the user must make sure that no extra parameters (e.g., time and ID) are passed to the *flowMeans* function as it will not remove them automatically.

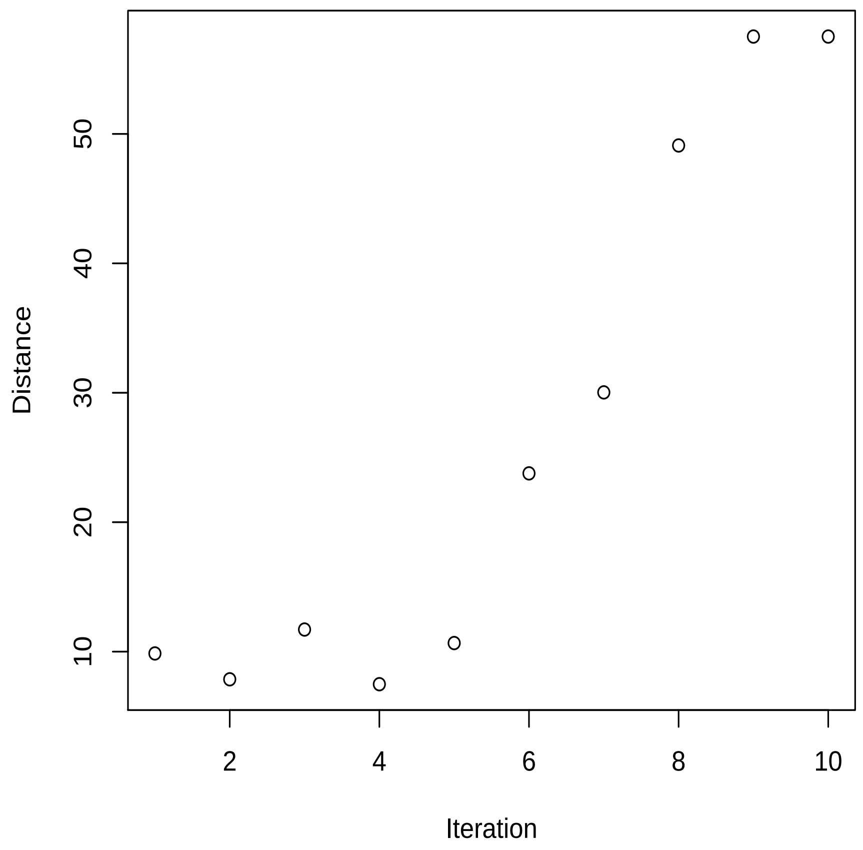
The *Labels* value is a list of cluster membership labels at each iteration. As an example, this value can be used to visualize the first iteration:

```
> plot(x[, c(3, 4)], col = res@Labels[[1]], pch = 20)
```



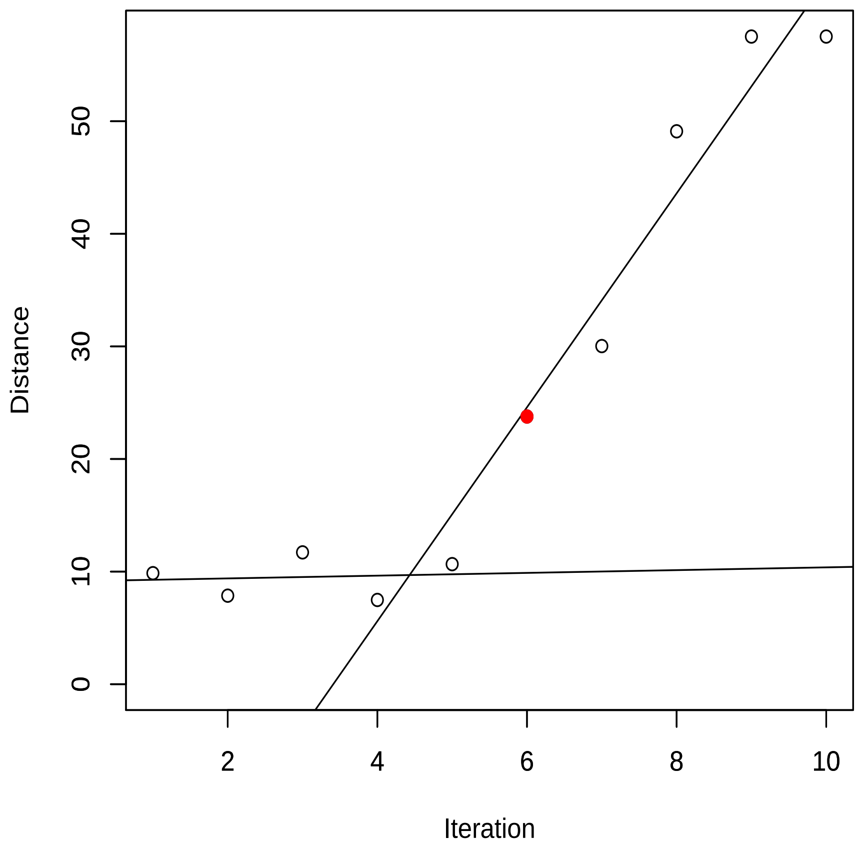
The *Mins* vector contains the minimum distances between the merged clusters at each iteration:

```
> plot(res@Mins, xlab = "Iteration", ylab = "Distance")
```



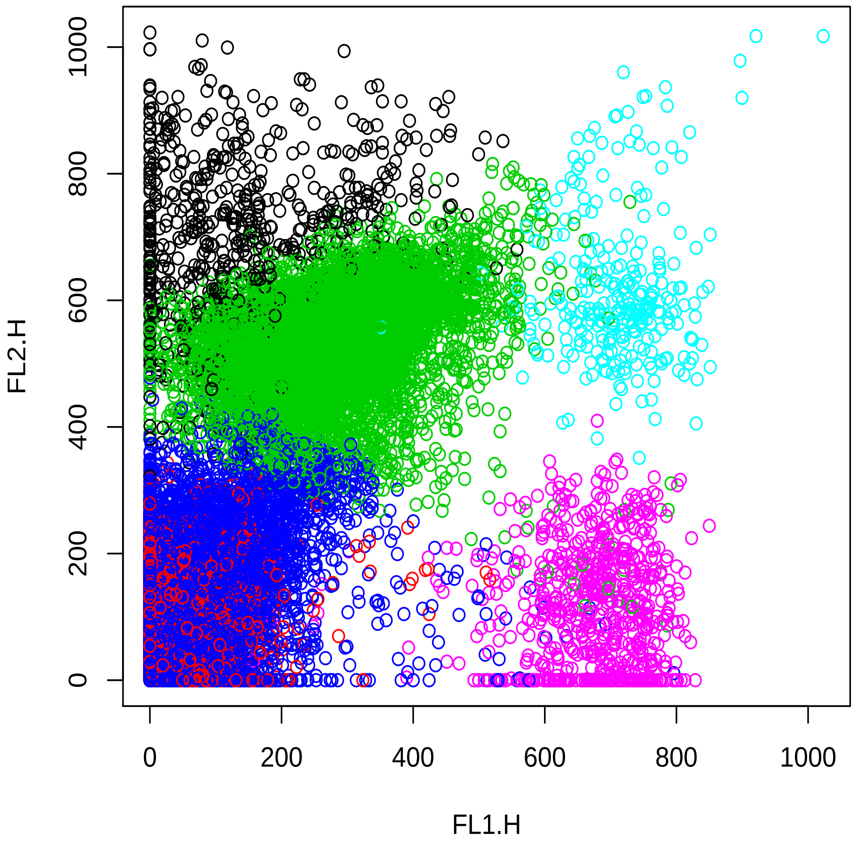
The *changepointDetection* function can be used to find the change point in the chart of minimum distances and iterations to select the correct number of clusters. The *MinIndex* value indicates the index of the change point:

```
> plot(res@Mins, xlab = " ", ylab = " ", xlim = c(1, res@MaxN),
+     ylim = c(0, max(res@Mins)))
> ft <- changepointDetection(res@Mins)
> abline(ft$l1)
> abline(ft$l2)
> par(new = TRUE)
> plot(ft$MinIndex + 1, res@Mins[ft$MinIndex + 1], col = "red",
+     xlab = "Iteration", ylab = "Distance", xlim = c(1, res@MaxN),
+     ylim = c(0, max(res@Mins)), pch = 19)
```



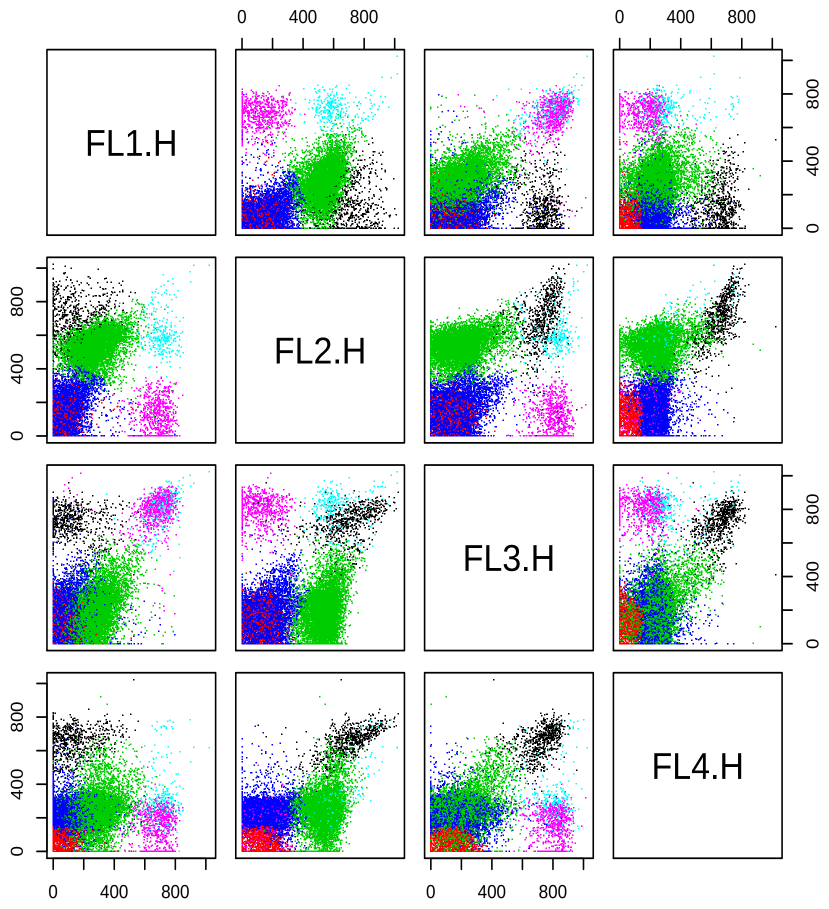
The *Label* vector indicates the cluster membership labels. The *plot* function can be used to visualize this:

```
> plot(x, res, c("FL1.H", "FL2.H"))
```



The cluster membership labels can also be visualized in multi-dimensions:

```
> plot(x, res, c("FL1.H", "FL2.H", "FL3.H", "FL4.H"), pch = ".")
```


FL1.H
FL2.H
FL3.H
FL4.H
0
400
800