The flowPeaks: a flow cytometry data clustering algorithm that uses K-means and peak finding

Yongchao Ge

March 19, 2012

yongchao.ge@gmail.com

1 Licensing

Under the Artistic License, you are free to use and redistribute this software. However, we ask you to cite the following paper if you use this software for publication.

- Ge Y. et al, flowPeaks: a fast unsupervised clustering forflow cytometry data via K-means and density peak finding, Manuscript, 2012

2 Overview

We combine the ideas from the finite mixture model and histogram spatial exploration together to find the clustering of the flow cytometry data. This new algorithm, in which we called `flowPeaks`, can be applied to high dimensional data and identify irregular shape clusters. The algorithm first uses K-means algorithm with a large K to partition the population into many compact clusters. These partitioned data allow us to generate a smoothed density function. All local peaks are exhaustedly found by exploring the density function and the cells are clustered by the associated local peak. The algorithm flowPeaks ia automatic, fast and reliable and robust to the cluster shape and outliers. Details can be seen in the paper by Ge (2012).

3 Installation

3.1 All users

When you are reading this and find the R package is already available in the Bioconductor package repository. You can install it directly from Bioconductor.

- **Windows users**: select the menu "Packages" and then click "Select repositories..." and choose "BioC software". And then select the menu 'Packages" click "install R package(s)..." and then look for the package flowPeaks.
- **Linux users**: This also works for Windows users. Type the following after you have invoked R

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("flowPeaks")
```

If this succeeds, congratulations, you can ignore the rest of this section and skip to Section 4.

3.2 Windows Users

Please read section 3.1 to install the R package from Bioconductor before proceed this. If you have the prebuilt binary of **flowPeaks** zip file, you can install the package by selecting the menu "Packages", and then "Install packages from a local zip file", and then point to prebuilt binary of **flowPeaks** zip file.

To build **flowPeaks** from the source by using Rtools is very not straightforward. R novices are not encouraged to try this. Experienced R users need to carefully follow the instruction of the Rtools (`http://www.murdoch-sutherland.com/Rtools/`) and `http://cran.r-project.org/doc/manuals/R-admin.html#The-Windows-toolset`. The GSL library needs to be downloaded from the file `local.zip` at `http://www.stats.ox.ac.uk/pub/Rtools/goodies/Win32` and the `GSL_LIB` in the file `flowPeaks/src/Makevars.win` should be pointed to the top folder of the extracted file `local.zip`. The top folder should contain three subfolders: include, bin and lib. The users are not encouraged to compile their own gsl library by MinGW or Visual Studio. Most likely their own version of gsl library is not going to work.

3.3 Linux Users

To build the **flowPeaks** package from the source, make sure that the following is present on your system:

- C++ compiler
- GNU Scientific Library (GSL)

A C++ compiler is needed to build the package as the core function is coded in C++. GSL can be downloaded directly from `http://www.gnu.org/software/gsl/` and follow its instructions to install the GSL from the source code. Alternatively, GSL can also be installed from your linux specific package manager (for example, Synaptic Package Manager for Ubuntu system). Other than the GSL binary library, please make sure the GSL development package is also installed, which includes the header files when building **flowPeaks** package.

Now you are ready to install the package:

```
R CMD INSTALL flowPeaks_x.y.z.tar.gz
```

If GSL is installed at some non-standard location such that it cannot be found when installing **flowPeaks**. You need to do the following

1. Find out the GSL include location (`<path-to-include>`) where the GSL header files are stored in the sub folder `gsl`, and GSL library location (`<path-to-lib>`) where the lib files are stored. If the GSL's `gsl_config` can be run, you can get them easily by `gsl-config --cflags` and `gsl-config --libs`
2. In the file `flowPeaks/src/Makevars`, you may need to change the last two lines as below:
   - `PKG_CXXFLAGS = $(GSL_CFLAGS) -I. -I<path-to-include>`
   - `PKG_LIBS = $(GSL_LIBS) -L<path-to-lib>`

4 Examples

To illustrate how to use the core functions of this package, we use a barcode flow cytometry data, which can be accessed by the command data(barcode). The barcode data is just a simple data matrix of 180,000 rows and 3 columns. The clustering analysis is done by using the following commands.

```
> library(flowPeaks)
```

```
> data(barcode)
> summary(barcode)
```

```
  Pacific.blue      Alexa             APC
 Min.   : 153   Min.   :  81.0   Min.   :  92
 1st Qu.:1198   1st Qu.: 602.0   1st Qu.: 503
 Median :1900   Median : 699.0   Median :1817
 Mean   :1817   Mean   : 688.7   Mean   :1565
 3rd Qu.:2522   3rd Qu.: 789.0   3rd Qu.:2393
 Max.   :3503   Max.   :1119.0   Max.   :3369
```

```
> fp<-flowPeaks(barcode)
```

```
Starting the flow Peaks analysis...

    Task A: compute kmeans...
        step 0, set the intial seeds, tot.wss=5.47711e+09
        step 1, do the rough EM, tot.wss=3.52259e+09 at 0.881 sec
        step 2, do the fine transfer of Hartigan-Wong Algorithm
                 tot.wss=3.52028e+09 at 1.271 sec
        ...finished summarization at 1.323 sec

    Task B: find peaks...finished at 1.734 sec
```

```
> plot(fp,idx=c(1,3))
```

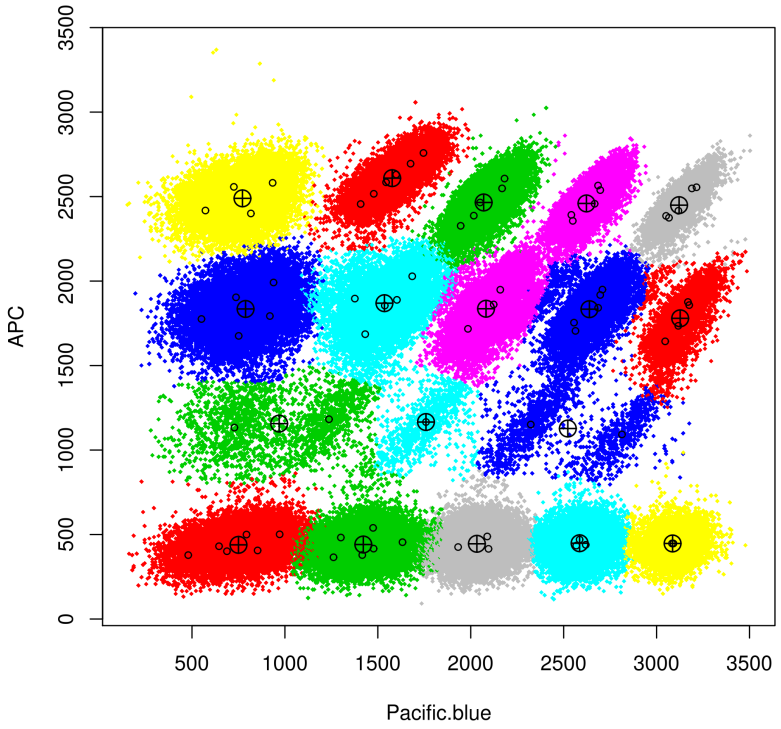

Figure 1: The scatter plot of clustering results for the first column and the third column. Two clusters may share the same color as the automatic color specification is not unique for all clusters. The users may have option to change the col option in the plot function to have their own taste of color specifications. The flowPeaks cluster are drawn in different colors with the centers ($\oplus$), the underlying K-means cluster centers are indicated by $\circ$.

If we want to visualize the results, we can draw scatter plot for any two columns of the data matrix. The result is shown in Figure 1. Different colors specify different clusters, and the dots represent the center of the initial K-means, and triangles are the peaks found by flowPeaks. We can see all pairwise scatter plots, the R commands and the plot are shown in Figure 2.

We find out that the data displays better clustering in the Pacific.blue and APC. We could choose to redo the clustering just focusing on these two dimensions as shown in Figure 3.

```
> par(mfrow=c(2,2))
> plot(fp,idx=c(1,2,3))
```

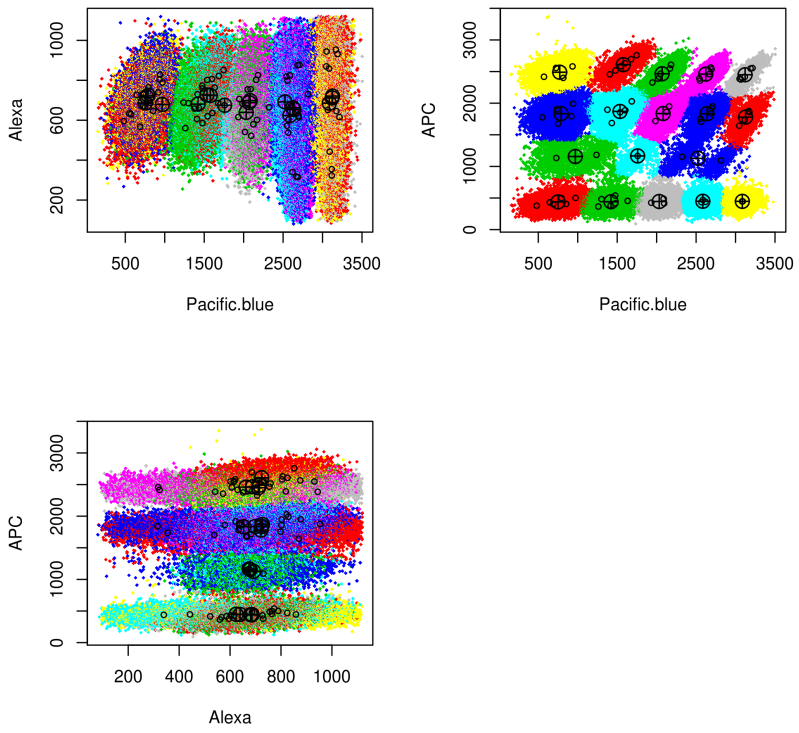

Figure 2: Pairwise plot of all columns

Since the clustering is on two dimension, the plot in Figure 3 is able to give you the boundary of the final clusters (in bold line) and the Voronoi boundary of the K-means. These boundaries are not visible for the data clustering in higher dimension as the projection onto 2D will have many clusters overlapped.

We can evaluate the cluster performance with the gold standard clustering stored in barcode.cid by

```
> evalCluster(barcode.cid,fp2$peaks.cluster,method="Vmeasure")
```

```
[1] 0.9983352
```

We could remove the outliers that are far away from the cluster center or can not be ambiguously assigned to one of the neighboring cluster as shown in Figure 4.

```
> fp2<-flowPeaks(barcode[,c(1,3)])
> plot(fp2)
```

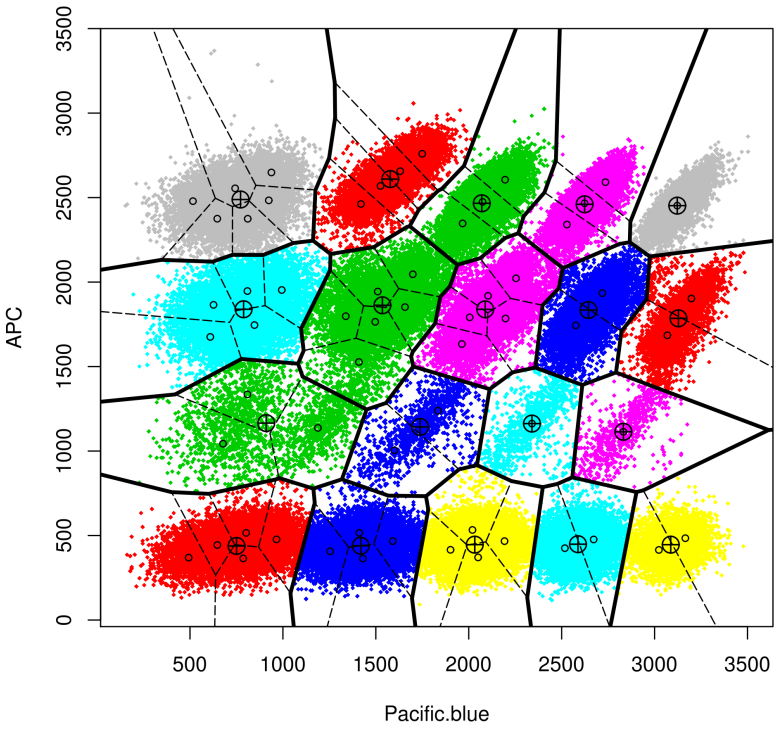

Figure 3: Clustering results based on only the Pacific.blue and APC

```
> fpc<-assign.flowPeaks(fp2,fp2$x)
> plot(fp2,classlab=fpc,drawboundary=FALSE,
+    drawvor=FALSE,drawkmeans=FALSE,drawlab=TRUE)
```

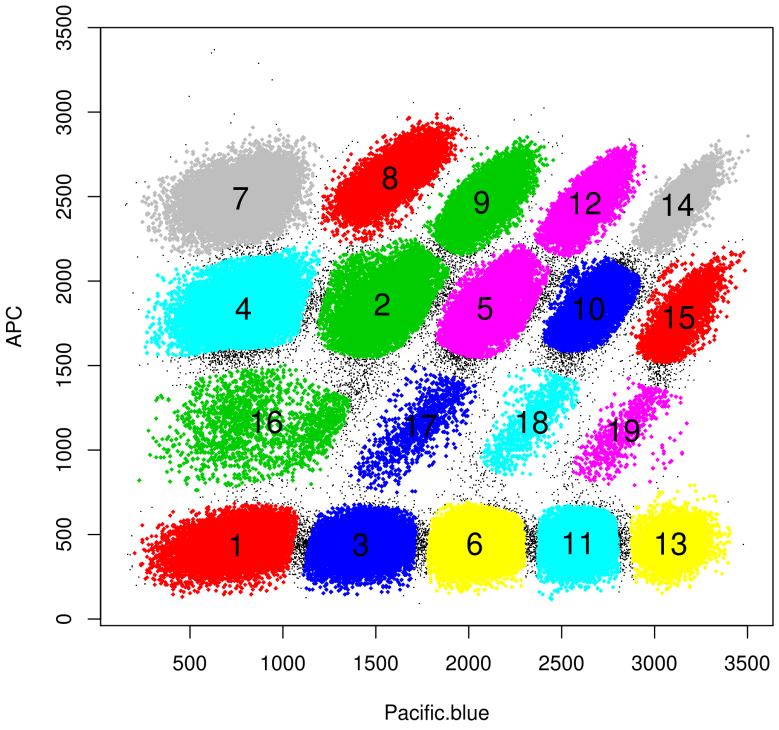

Figure 4: The plot of the barcode after the outliers have been identified as black points