# Clustering Cells by NF-kB Intensity

*Joshua Cook*

*9/7/2019*

## Contents



## 0.1 Overview

### 0.1.1 Purpose of this document

This document is meant to serve as an example of how to conduct data analysis in R Markdown. It is meant to accompany an introductory page available to the BMI 713 course at Harvard Medical School.

For coding in R, I tend to follow a mixture of Google's R Style Guide and that encouraged by the Tidyverse.

### 0.1.2 The mock analysis

The goal of the analysis coducted below was to separate cells into two groups based on the intesnity of the immunofluorescent signal of nuclear NF-$\kappa$B. The Mock Data section created the data for two populations with normally-distributed NF-$\kappa$B signals. The Clustering section then classified each cell into one of two groups based on its fluorescence intensity.

## 0.2 Mock Data

### 0.2.1 Creation

Here, I create some mock data consisting of three columns: `nucleus_id`, `nfkb_intensity`, and `real_group`. It is intended to represent the nuclear intensity of NF-$\kappa$B in a microscopy image containing `n_cells`cells. There are two groups in this population, each with a different normally-distributed fluorescence intensity. In a real analysis, the true identity of the groups would not be known, but this will be used for validation in this example analysis.

```
# total number of cells in a microscopy image
n_cells <- 1e5
# randomly select number of cells for group 1
```

```
n_group1 <- sample(1:n_cells, 1)
# make test data
#   nucleus ID: a unique label for each nucleus in the image
#   nfkb_intensity: intensity of the NK-kB probe in the cells
#   real group: the known label for each cell
df <- tibble(
    nucleus_id = paste("nucleus", 1:n_cells),
    nfkb_intensity = c(
        rnorm(n_group1, mean = 50, sd = 15),
        rnorm(n_cells - n_group1, mean = 100, sd = 15)
    ),
    real_group = c(
        rep(1, n_group1),
        rep(2, n_cells - n_group1)
    )
)
```

### 0.2.2 Shuffle

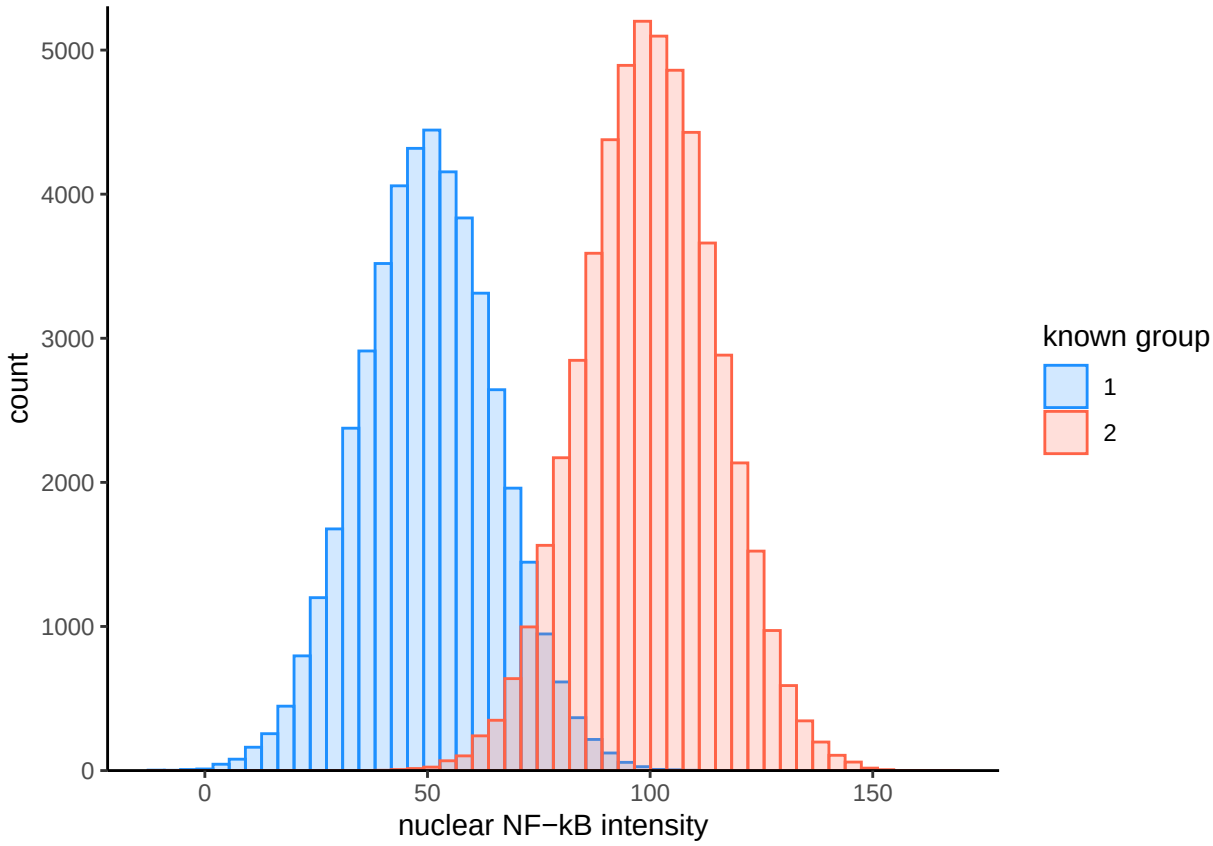
To be sure that I am not somehow cheating because the data frame is in order by group, I shuffled the rows of the data frame.

```
# shuffle the rows by shuffling the row indices
df <- df[sample(1:nrow(df)), ]
```

```
#> # A tibble: 100,000 x 3
#>    nucleus_id    nfkb_intensity real_group
#>    <chr>                 <dbl>      <dbl>
#>  1 nucleus 99669          97.3          2
#>  2 nucleus 85166         100.           2
#>  3 nucleus 18511          48.0          1
#>  4 nucleus 48619         119.           2
#>  5 nucleus 31030          25.1          1
#>  6 nucleus 28892          43.9          1
#>  7 nucleus 83846          97.0          2
#>  8 nucleus 79990         100.           2
#>  9 nucleus 28193          40.6          1
#> 10 nucleus 31801          47.0          1
#> # ... with 99,990 more rows
```

### 0.2.3 Visualization of the groups

A histogram of the intensity data is shown below. The coloration is of the real groups that would normally not be known. I randomly selected the number of cells for group 1 (between 1 and the total number of cells). This value was 46031, meaning there are $5.3969 \times 10^4$ cells in group 2.




## 0.3 Clustering

### 0.3.1 Installing 'mclust'

For this analysis, I used the 'mclust' package to identify the two groups in the data. It can be downloaded from CRAN using the following command. (Note that I inlcude `eval=FALSE` in the chunk header so that I don't install the package every time I knit the document.) A thorough explanation of the many features of this package are available in the "A quick tour of mclust" vignette.

```
# install 'mclust' from CRAN
install.packages("mclust")
```

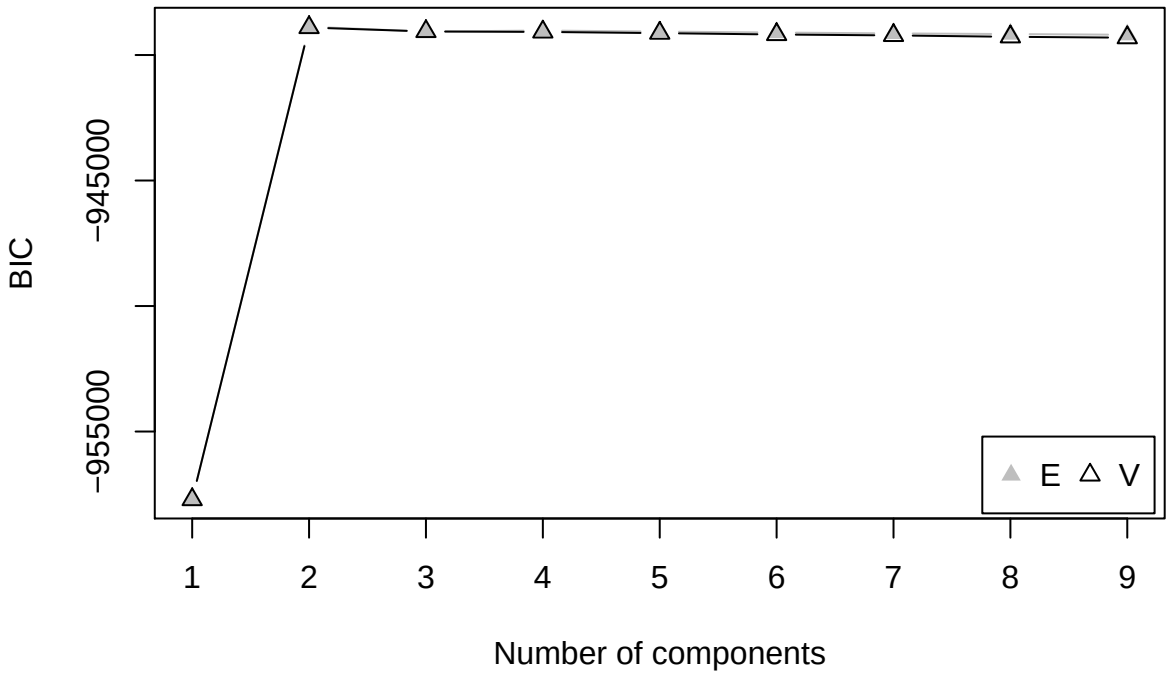
### 0.3.2 Using BIC to identify the number of clusters

The Bayesian Information Criterion (BIC) can be used to predict how many clusters exist in the data.

```
# identify the optimal number of groups using the BIC
nfkb_bic <- mclustBIC(df$nfkb_intensity, verbose = FALSE)
```

A summary of the results, shown below, along with the plot, clearly indicate that there are likely two clusters in the data (as expected). The `nfkb_bic` object can be used in the clustering function, next.

```
#> Best BIC values:
#>                E,2          V,2         E,4
#> BIC      -938871.2 -938903.81388 -939023.6868
#> BIC diff       0.0     -32.61857    -152.4915
```



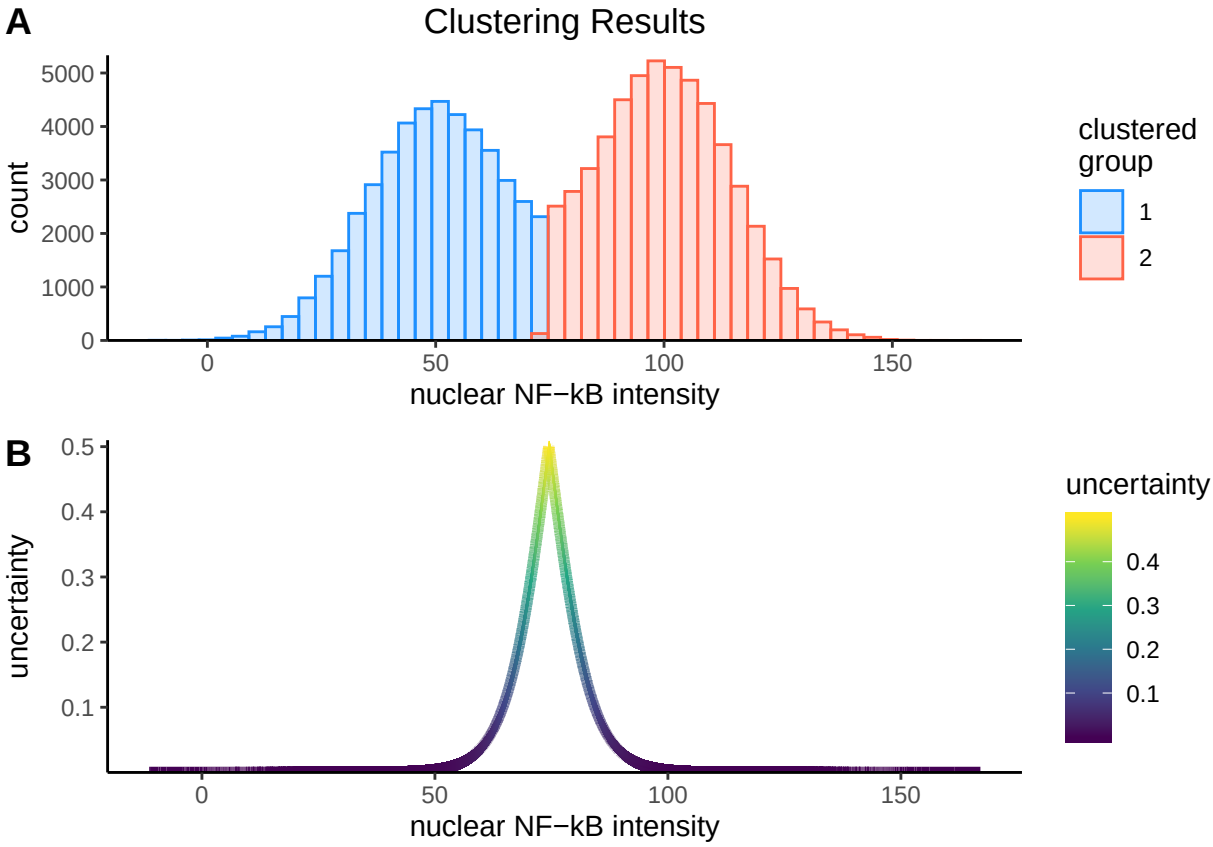

### 0.3.3 Clustering by NK-$\kappa$B intensity

The `Mclust` function actually does the clustering of the data. At first glance, the summary statistics indicate that the clustering was quite successful.

```
# cluster the intensity values
fit <- Mclust(df$nfkb_intensity, x = nfkb_bic)
```

```
#> ----------------------------------------------------
#> Gaussian finite mixture model fitted by EM algorithm
#> ----------------------------------------------------
#>
#> Mclust E (univariate, equal variance) model with 2 components:
#>
#>  log-likelihood      n df       BIC       ICL
#>       -469412.6 100000  4 -938871.2 -950027.9
#>
#> Clustering table:
#>     1     2
#> 45977 54023
#>
#> Mixing probabilities:
#>         1         2
#> 0.4609054 0.5390946
#>
#> Means:
#>         1         2
#>  50.09366 100.06082
#>
#> Variances:
#>       1       2
#> 224.354 224.354
```

As shown by the summary of the model, the means of the two groups were very acurately predicted as 50.1 and 100.1. The resultant classification of each cell by the NF-$\kappa$B intensity is shown below (**A**). Some cells were mislabeled comapred to their real group because they had intensity values more simillar to the other group. Because there are only two groups that are easy to distinguish by eye, the clusters were split into two about halfway between the means. Accordingly, the uncertainty of each classification increased as this boundary was approached (**B**).




The table below shows the concoradance between the real groups (vertical) and the clustered groups (horizontal). The values on the diagonal represent the number of correct classifications whereas the off-diagonal values are the number of misclassified cells.

```
tbl <- table(df$real_group, df$clustered)
colnames(tbl) <- paste("cls", colnames(tbl))
rownames(tbl) <- paste("real", rownames(tbl))
tbl
```

```
#>
#>          cls 1 cls 2
#>   real 1 43597  2434
#>   real 2  2380 51589
```

### 0.3.4 Accuracy

To quantify the success of the clustering, the accuracy was calculated. This metric is defined as follows:

$$ACC = \frac{\text{correctly classified}}{\text{total cells}}$$

**Thus, the accuracy of the classification was 0.952.**

---

### 0.3.5 Session Info

Below is a summary of the R session used for the R Markdown file.

```
sessionInfo()
```

```
#> R version 3.6.1 (2019-07-05)
#> Platform: x86_64-apple-darwin15.6.0 (64-bit)
#> Running under: macOS Mojave 10.14.4
#>
#> Matrix products: default
#> BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
#> LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
#>
#> locale:
#> [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#>  [1] forcats_0.4.0   stringr_1.4.0   dplyr_0.8.3     purrr_0.3.2
#>  [5] readr_1.3.1     tidyr_0.8.3     tibble_2.1.3    ggplot2_3.2.1
#>  [9] tidyverse_1.2.1 cowplot_1.0.0   mclust_5.4.5
#>
#> loaded via a namespace (and not attached):
#>  [1] tidyselect_0.2.5  xfun_0.9          haven_2.1.1
#>  [4] lattice_0.20-38   colorspace_1.4-1  generics_0.0.2
#>  [7] vctrs_0.2.0       viridisLite_0.3.0 htmltools_0.3.6
#> [10] yaml_2.2.0        utf8_1.1.4        rlang_0.4.0
#> [13] pillar_1.4.2      glue_1.3.1        withr_2.1.2
#> [16] modelr_0.1.5      readxl_1.3.1      munsell_0.5.0
#> [19] gtable_0.3.0      cellranger_1.1.0  rvest_0.3.4
#> [22] evaluate_0.14     labeling_0.3      knitr_1.24
#> [25] fansi_0.4.0       broom_0.5.2       Rcpp_1.0.2
#> [28] backports_1.1.4   scales_1.0.0      jsonlite_1.6
#> [31] hms_0.5.1         digest_0.6.20     stringi_1.4.3
#> [34] grid_3.6.1        cli_1.1.0         tools_3.6.1
#> [37] magrittr_1.5      lazyeval_0.2.2    crayon_1.3.4
#> [40] pkgconfig_2.0.2   zeallot_0.1.0     xml2_1.2.2
#> [43] lubridate_1.7.4   assertthat_0.2.1  rmarkdown_1.15
#> [46] httr_1.4.1        rstudioapi_0.10   R6_2.4.0
#> [49] nlme_3.1-141      compiler_3.6.1
```