flowDensity

Automated alternative to the current manual gating practice

Jafar Taghiyar, Mehrnoush Malek

October 30, 2017

Contents

1 Licensing

2 Introduction

Expert humans use flowJo software to manually gate FCS data files either individually or by setting a static gate to apply on all the files. The former is very tedious specially when there is a large number of files and the cost for the latter is to ignore characteristics of individual samples.

flowDensity is a supervised clustering algorithm based on density estimation techniques designed specifically to overcome these problems. It automates the current practice of manual 2D gating and adjusts the gates for each FCS data file individually.

Although automated flow cytometry methods developed to date have focused on fully automated analysis which is especially suited for discovery, they seldom match manual results where this is desirable (e.g., for diagnosis). In contrast, flowDensity aims to gate

predefined cell populations of interest where the gating strategy, *i.e.*, sequence of gates, is known. This helps it take advantage of expert knowledge and as a result it often matches manual results very well. In addition, since flowDensity uses only two dimensions at a time, it is very fast and requires mush less computational power.

3 How to use flowDensity?

In order to use flowDensity, the gating strategy is required. A gating strategy here means the sequence of 2D gates needed to apply one at a time on a FCS file to eventually extract the cell subset of interest.

A 2D gate consists of two channels (dimensions) or equivalently a phenotype with two markers. In addition, the corresponding expression level for each channel is given. For example, phenotype "CD19+CD20-" has markers CD19 and CD20 with expression values "positive" and "negative", respectively.

To use flowDensity, this 2D gate is input to the function *flowDensity(.)*. The channels in the gate are used for the *channels* argument and the expression values are used for the *position* argument of the function.

Let assume for example that CD19 is on channel "PerCP-Cy5-5-A" and CD20 is on channel "APC-H7-A". Therefore, the corresponding input arguments are:

channels=c(PerCP-Cy5-5-A, APC-H7-A) and *position=c(TRUE,FALSE)*.

In general, *channels* argument can be set using either names of the channels or their corresponding indices (column numbers in the FCS file) and *position* argument could be one the four logical pairs *(TRUE,FALSE)*, *(FALSE,TRUE)*, *(FALSE,FALSE)* and *(TRUE,TRUE)*. If the user needs to set the thresholds for only one of the channels, then position for the other channel must be set to *NA*.

The expression value "high" can also be identified by flowDensity but some specific (optional) arguments must be set accordingly. More on this later.

In addition to the above arguments, either *cell.population* or *flow.frame* argument is required where the former is an object of class "CellPopulation" loaded from flowDensity namespace and the latter is a "flowFrame" object loaded from flowCore namespace.

4 Examples

In this section we present several examples to elaborate how to use the *flowDensity(.)* function.

4.1 Extracting Bcell

This example shows how to use flowDensity to extract B cells by using the gating strategy Singlet/viability-CD3-/CD19+CD20+ or equivalently lsinglets/Bcell. The Figures 1-2 show this.

```
> library(flowCore)
> library(flowDensity)
```

```
> library(GEOmap)
> data_dir <- system.file("extdata", package = "flowDensity")
> load(list.files(pattern = 'sampleFCS_1', data_dir, full = TRUE))
> f

flowFrame object ''
with 23000 cells and 13 observables:
              name desc  range      minRange maxRange
$P1          FSC-A <NA> 262144 -111.00000000 262143.0
$P2          FSC-H <NA> 262144    0.00000000 262143.0
$P3          SSC-A <NA> 262144 -111.00000000 262143.0
$P4          SSC-H <NA> 262144    0.00000000 262143.0
$P5          APC-A CD38 262144   -0.04496801      4.5
$P6       APC-H7-A CD20 262144    0.57425502      4.5
$P7         FITC-A <NA> 262144    0.17100441      4.5
$P8  PerCP-Cy5-5-A CD19 262144   -0.06225046      4.5
$P9         V450-A  CD3 262144    0.15907409      4.5
$P10        V500-A  IgD 262144    0.22768370      4.5
$P11          PE-A CD24 262144   -0.05101856      4.5
$P12      PE-Cy7-A CD27 262144   -0.23198773      4.5
$P13          Time <NA> 262144    0.00000000 262143.0
211 keywords are stored in the 'description' slot

> sngl <- flowDensity(f,channels = c("FSC-A","FSC-H"),position = c(F,F),
+                     percentile =c(.99999,.99999),use.percentile = c(T,T),
+                     ellip.gate = T,scale = .99 )

> plotDens(f,c(1,2))
> lines(sngl@filter,type="l")


> bcell <- flowDensity(sngl, channels=c(9, 3),
+                      position=c(FALSE, NA))

> plot(getflowFrame(sngl), bcell)
```

4.2 Gating rare cell populations

To emulate the practice of expert humans for identification of high, flowDensity provides two parameters that help fine tune the algorithm for identification of small cell populations. These parameters are set by providing following arguments in the *flowDensity(.)* function:

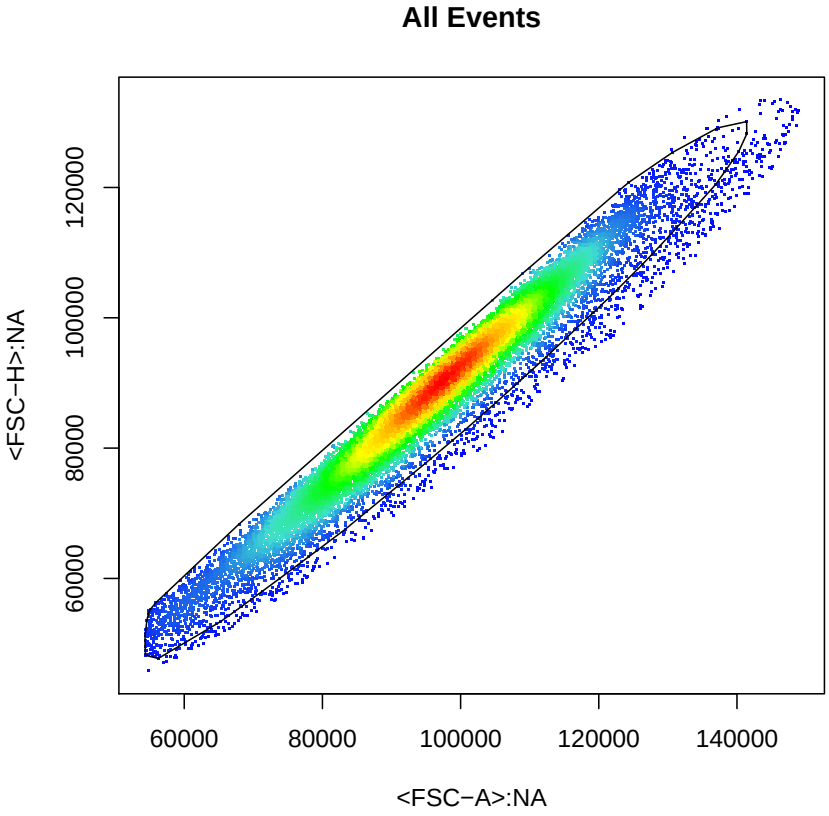

Figure 1: Gating singlets using ellipsoid gating

- *upper*: This argument is used to identify small cell subsets present at the tail or head of the density distribution curve where they are typically camouflaged due to the presence of adjacent large cell populations. If it is set to "TRUE" ("FALSE"), flowDensity checks the tail (head) of the density distribution. If it is required to use *upper* for one channel and not the other, the "NA" value should be used; for example *upper=c(FALSE,NA)*.

- *use.upper*: This argument is only used when the user wants to force the algorithm to use the upper argument no matter how many peaks are found in the density distribution.

- *percentile*: This argument gets a value of $[0, 1)$ and provides the ability to set a threshold based on the percentile of the density distribution. To force using this threshold, argument *use.percentile* should be set to "TRUE", otherwise the percentile threshold will be automatically used when appropriate.

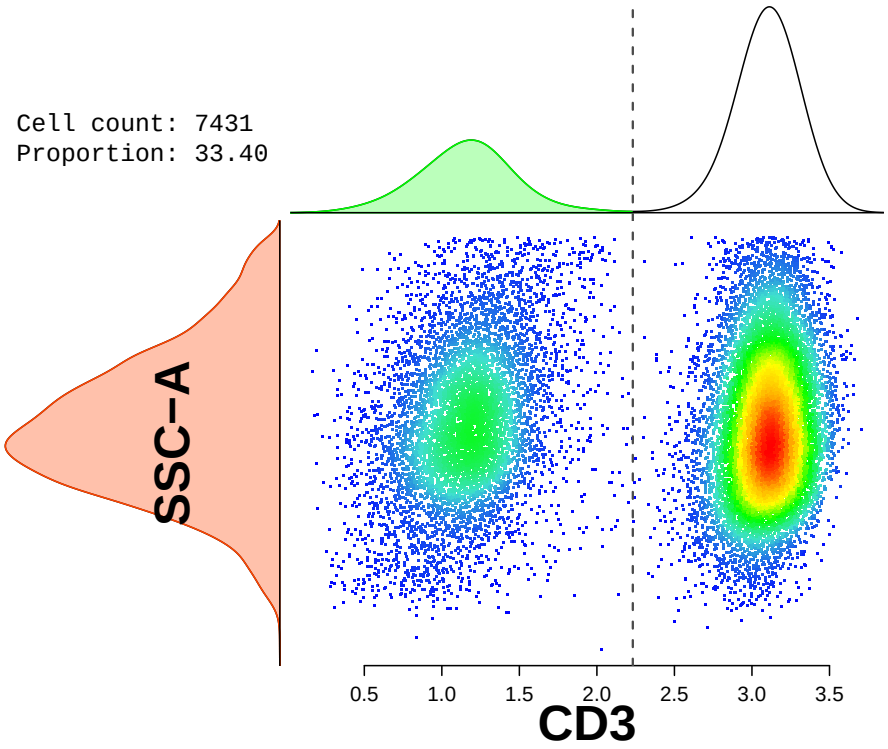

Figure 2: B cells from gating strategy lymphocytes/singlets/CD3-

For example, we can use flowDensity to extract "plasmablasts" cell population as follows:

```
> CD19pCD20n <- flowDensity(obj=bcell, channels=c(8, 6),
+                           position=c(T,F))
> plasmablasts <- flowDensity(obj=CD19pCD20n, channels=c(5, 12),
+                             position=c(T, T))
> plotDens(getflowFrame(CD19pCD20n), plasmablasts@channels, pch=19)
> points(plasmablasts@filter, type='l', col=2, lwd=2)
```

To overcome this problem, flowDensity tracks the slope of the curve of the density distribution by comparing the slope of a window of points on the curve with specific length to examine if it drops below a a threshold relative to the adjacent windows. This way, once the large cell subset ends and the rare one starts the dramatic change in the slope can be detected by flowDensity and the threshold is set. We call this technique as *slopeTrack*

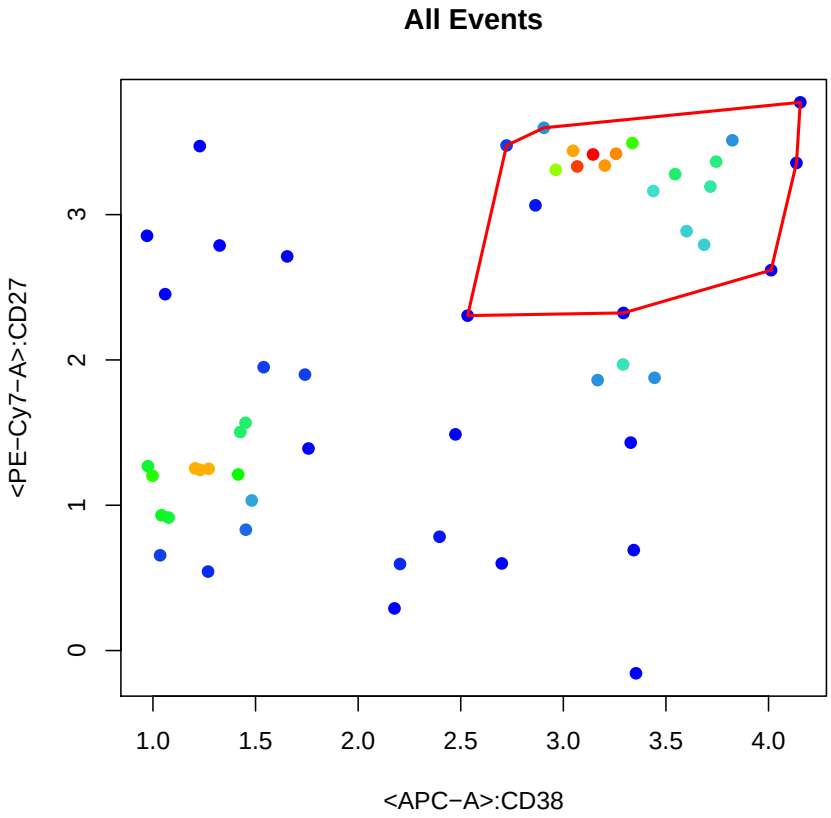

Figure 3: plasmablasts cell population, a small subset of B cells

In seldom cases the slope varies slowly and smoothly so that no relatively huge change is sensed by this technique. If such, the 90^{th} percentile is used as a gate. A rule of thumb is that if the spread of the density distribution is mostly around the mean, *i.e.*, the standard deviation is small relative to the mean, then most likely the trackSlope returns better results than 90^{th}. If neither of these techniques are bale to set a proper threshold, the peak value plus a multiplier of the standard deviation is chosen as the threshold.

flowDensity is able to decide on which of these methods to use. However, the user can also modify this decision by setting certain parameters specifically for tricky cell populations.

In Figure 3, flowDensity has been used to gate "Plasma blasts" cell population which is a rare cell subset of CD3+CD19+CD20- cell population. On the x-axis for the marker CD38 the slopeTrack technique is used whereas on the y-axis for the markerCD27 HLA-DR the peak plus 1.5×standard deviation gives a proper gate. [1]

To plot the gating results, there is custom *plot(.)* method where its arguments are a flowFrame object and an object of class CellPopulation. For example:

where *getflowFrame(.)* function is an accessor to get the flowFrame of a CellPopulation object.

4.3 Multiple calls for a single cell population identification

```
> f <- nmRemove(f, c("FSC-A", "SSC-A"))
> #plotDens(f, c("FSC-A", "SSC-A"))
> #plot(f, lymph)
```

flowDensity can be used recursively to gate a cell population of interest.

In Figure 4, flowDensity has been used to gate "lymphocytes" from CD45 vs SSC. In order to gate lymphocytes more accurate and tighter, flowDensity can be called several times. First time it finds the thresholds for both channels, then returns SSC-CD45+ as an input for the second call. In the last call thresholds of CD45 from the first call and thresholds of SSC from the second call is given to flowDensity to draw ellipse around the lymphocyte population. Figure 5 illustrates the last step. In some cases CD45 has only one peak so the percentile of 0.25 is given to flowDensity to detect the right population. For SSC 0.85 would give the optimum threshold.

```
> load(list.files(pattern = 'sampleFCS_2', data_dir, full = TRUE))
> f2

flowFrame object ''
with 7000 cells and 14 observables:
              name   desc  range    minRange maxRange
$P1          FSC-A   <NA> 262144  0.00000000 262143.0
```

[1] The multiplier 1.5 is the default value of the algorithm. However, it can be both set by user or set via analyzing the density distribution by flowDensity

```
$P2          FSC-H   <NA> 262144  0.00000000 262143.0
$P3          FSC-W   <NA> 262144  0.00000000 262143.0
$P4          SSC-A   <NA> 262144  0.00000000 262143.0
$P5          SSC-H   <NA> 262144  0.00000000 262143.0
$P6          SSC-W   <NA> 262144  0.00000000 262143.0
$P7         FITC-A Lambda 262144  0.16328094      4.5
$P8           PE-A  Kappa 262144  0.11965745      4.5
$P9  PerCP-Cy5-5-A    CD5 262144  0.52215808      4.5
$P10      PE-Cy7-A   CD10 262144  0.83157577      4.5
$P11         APC-A   CD19 262144  0.49399988      4.5
$P12      APC-H7-A   CD20 262144  0.69680349      4.5
$P13        V450-A   CD38 262144  0.37618959      4.5
$P14        V500-A   CD45 262144 -0.02725884      4.5
204 keywords are stored in the 'description' slot
```

```
> channels <- c("V500-A", "SSC-A")
> # First call to flowDensity
> tmp.cp1 <- flowDensity(obj=f2, channels=channels,
+                        position=c(TRUE, FALSE), percentile=c(0.25, NA))
> # Second call to flowDensity
> tmp.cp2 <- flowDensity(obj=tmp.cp1, channels=channels,
+                         position=c(TRUE, FALSE), gates=c(FALSE, NA),
+                         percentile=c(NA, 0.85))
> # Final call to flowDensity
> lymph <- flowDensity(obj=f2, channels=channels,
+                      position=c(TRUE, FALSE), gates=c(tmp.cp1@gates[1],
+                      tmp.cp2@gates[2]), ellip.gate=TRUE, scale=.99)

> plot(f2, tmp.cp1)

> plot(f2, tmp.cp2)

> plotDens(f2, channels=channels)
> points(lymph@filter, type="l", col=2, lwd=2)
```

4.4 Gating cells using a control sample

To utilize matched control samples (*e.g.* FMO controls), the *flowDensity(.)* function has parameters that allow control data to be included. When this option is used, the gating threshold is calculated in the control data and applied to the stimulated data. Control samples are added using two parameters:

- *use.control*: When set to "TRUE", flowDensity uses matched control data to calculate gating thresholds. This argument can be set for both channels. For example: *use.control=c(TRUE, FALSE).*

- *control*: This argument accepts flowFrame or CellPopulation objects containing control data matched to the specified stimulated data (passed in the *obj* argument). Control samples can be included for one or both of the channels. If no control is to be used, the argument should be passed an "NA" value (default). For example, if the first channel should be gated using a control but the second channel should be gated normally (using the stimulated data), the user would specify *control=c(fmo.data, NA)*.

When control data is used, the other gating arguments (*upper, percentile, n.sd*, etc.) are applied to finding the threshold in the control sample instead of the stimulated sample.

For example, an FMO control (*i.e.* negative control) for the 'BV421-A' channel can be used for gating as follows:

```
> load(list.files(pattern = 'sampleFCS_3.Rdata', data_dir, full = TRUE))
> f3

flowFrame object ''
with 8000 cells and 20 observables:
                  name desc  range     minRange maxRange
$P1              FSC-A <NA> 262144  0.000000000 262143.0
$P2              FSC-W <NA> 262144  0.000000000 262143.0
$P3              SSC-A <NA> 262144  0.000000000 262143.0
$P4              SSC-W <NA> 262144  0.000000000 262143.0
$P5              APC-A <NA> 262144  1.424618765      4.5
$P6  Alexa Fluor 700-A <NA> 262144  1.693370490      4.5
$P7          APC-Cy7-A <NA> 262144 -0.006329766      4.5
$P8  Alexa Fluor 488-A <NA> 262144 -0.058480480      4.5
$P9      PerCP-Cy5-5-A <NA> 262144  1.653328450      4.5
$P10           BV421-A <NA> 262144  0.603735357      4.5
$P11           BV500-A <NA> 262144  0.054259242      4.5
$P12           BV570-A <NA> 262144  1.136510670      4.5
$P13           BV605-A <NA> 262144  0.647546933      4.5
$P14           BV650-A <NA> 262144  0.788049917      4.5
$P15           BV700-A <NA> 262144  0.914095648      4.5
$P16           BV785-A <NA> 262144  1.240968280      4.5
$P17              PE-A <NA> 262144  0.156357679      4.5
$P18          PE-Cy5-A <NA> 262144  0.736939054      4.5
$P19          PE-Cy7-A <NA> 262144  0.512480878      4.5
$P20              Time <NA> 262144  0.000000000 262143.0
269 keywords are stored in the 'description' slot

> load(list.files(pattern = 'sampleFCS_3_FMO', data_dir, full = TRUE))
> f3.fmo

flowFrame object ''
with 8000 cells and 20 observables:
```

```
                  name desc  range     minRange maxRange
$P1              FSC-A <NA> 262144  0.000000000 262143.0
$P2              FSC-W <NA> 262144  0.000000000 262143.0
$P3              SSC-A <NA> 262144  0.000000000 262143.0
$P4              SSC-W <NA> 262144  0.000000000 262143.0
$P5              APC-A <NA> 262144  1.424618765      4.5
$P6  Alexa Fluor 700-A <NA> 262144  1.693370490      4.5
$P7          APC-Cy7-A <NA> 262144 -0.006329766      4.5
$P8  Alexa Fluor 488-A <NA> 262144 -0.058480480      4.5
$P9      PerCP-Cy5-5-A <NA> 262144  1.653328450      4.5
$P10           BV421-A <NA> 262144  0.606985391      4.5
$P11           BV500-A <NA> 262144  0.054259242      4.5
$P12           BV570-A <NA> 262144  1.136510670      4.5
$P13           BV605-A <NA> 262144  0.647546933      4.5
$P14           BV650-A <NA> 262144  0.788049917      4.5
$P15           BV700-A <NA> 262144  0.914095648      4.5
$P16           BV785-A <NA> 262144  1.240968280      4.5
$P17              PE-A <NA> 262144  0.156357679      4.5
$P18          PE-Cy5-A <NA> 262144  0.736939054      4.5
$P19          PE-Cy7-A <NA> 262144  0.512480878      4.5
$P20              Time <NA> 262144  0.000000000 262143.0
269 keywords are stored in the 'description' slot
```

```
> f3.gated <- flowDensity(obj=f3, channels=c('BV421-A', 'FSC-A'),
+                         position = c(TRUE, NA),use.control = c(TRUE, F)
+                         , control = c(f3.fmo, NA))
```

```
[1] "Only one peak is found, set standard deviation, percentile, or upper arguments accor
[1] "Cutoff is based on inflection point."
```

```
> f3.fmo.gated <- flowDensity(obj=f3.fmo, channels=c('BV421-A', 'FSC-A'),
+                             position=c(TRUE, NA),
+                             gates=c(f3.gated@gates[1], NA))
```

When only one peak is present in density, flowDensity prints out a message that can be spressed by verbose=FALSE for each of the marker. This message infroms you about how cutoff was calculated based on the present arguments (percentile, upper, sd.threshold).

```
> plot(f3.fmo, f3.fmo.gated)
```

```
> plot(f3, f3.gated)
```

Figure 6 shows the gating of 'BV421-A' on both stimulated and control samples.

For finer control, additional gating arguments can be passed that will be applied to the control sample. For example, the below example will gate using the 98th percentile in control data:

```
> f3.gated.98p <- flowDensity(obj=f3, channels=c('BV421-A', 'FSC-A'),
+                             position = c(TRUE, NA),use.percentile = c(TRUE, NA),
+                             percentile = 0.98, use.control = c(TRUE, FALSE),
+                             control = c(f3.fmo, NA))
> f3.fmo.gated.98p <- flowDensity(obj=f3.fmo, channels=c('BV421-A', 'FSC-A'),
+                                 position = c(TRUE, NA),
+                                 gates=c(f3.gated.98p@gates[1], NA))

> plot(f3.fmo, f3.fmo.gated.98p)

> plot(f3, f3.gated.98p)
```

Figure 7 Shows the gating based on 98th percentile gate in FMO data.

Note: When using controls, setting *position=TRUE* will treat the data as a negative control and extract the population above the threshold. Setting *position=FALSE* will treat it as a positive control.

4.5 Selecting threshold using deGate()

Another option beside flowDensity is deGate() function, which gives better control over cutoffs. The output is either a number or a vector of all possible cutoffs if all.cuts=T. In the example below, some of the possibilities are provided

```
> load(list.files(pattern = 'sampleFCS_2', data_dir, full = TRUE))
> thresholds <- deGate(obj = f2,channel = 9)
> #Percentile default is .95, which can be changed
> thresholds.prcnt <- deGate(f2,channel = 9,use.percentile=T,percentile=.3)
> thresholds.lo <- deGate(f2,channel = 9,use.upper=T,upper=F,alpha = .9)
> thresholds.hi <- deGate(f2,channel = 9,use.upper=T,upper=T,alpha = .9)

> plotDens(f2,c(9,12))
> abline(v=c(thresholds,thresholds.prcnt,thresholds.lo,thresholds.hi),col=c(1,2,3,4))
```

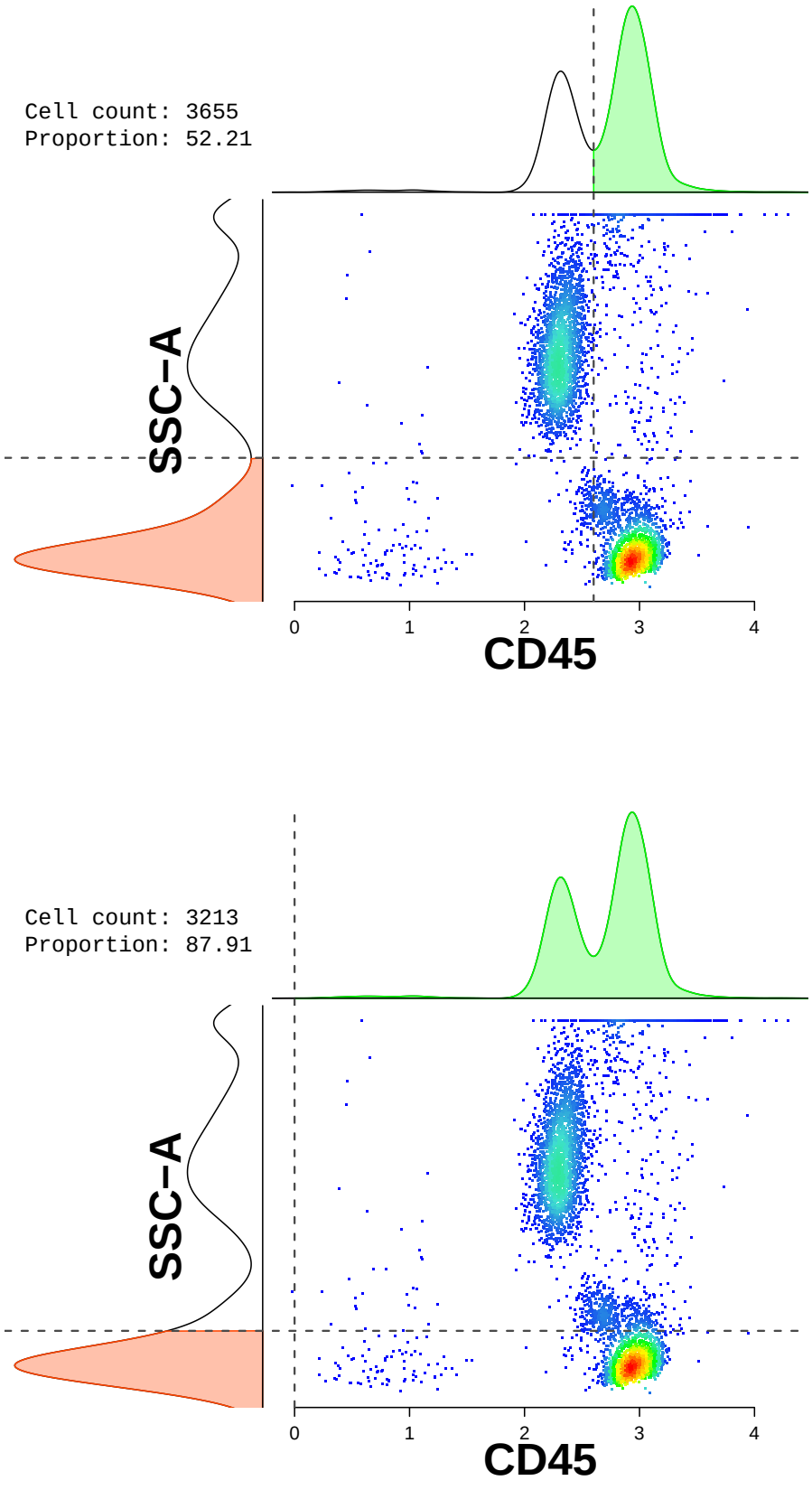

Figure 4: Two steps of gating lymphocytes

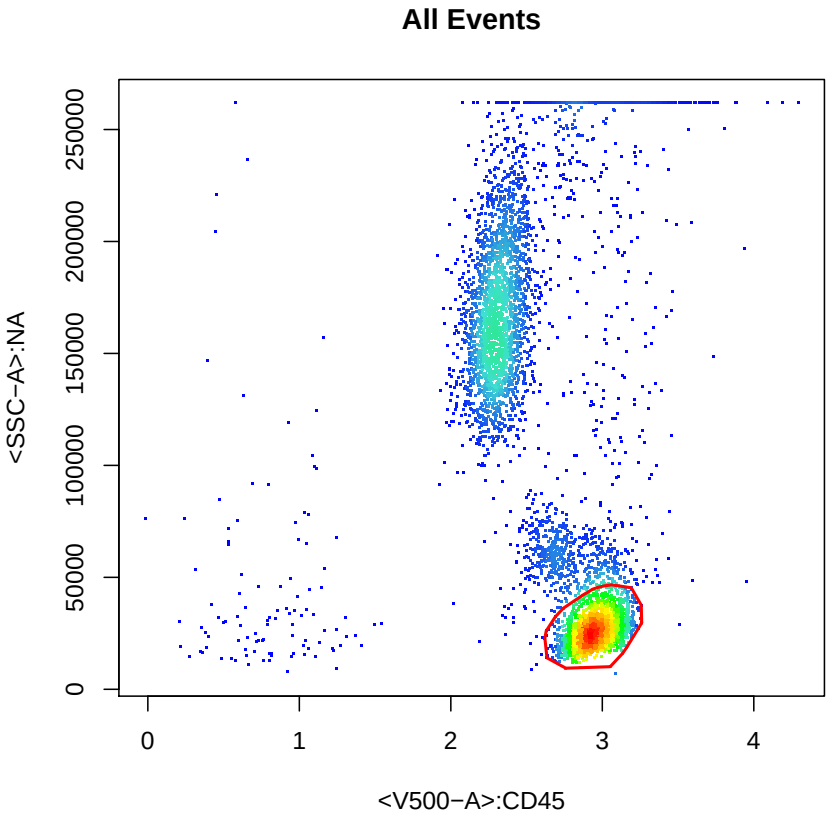

Figure 5: lymphocyte cels

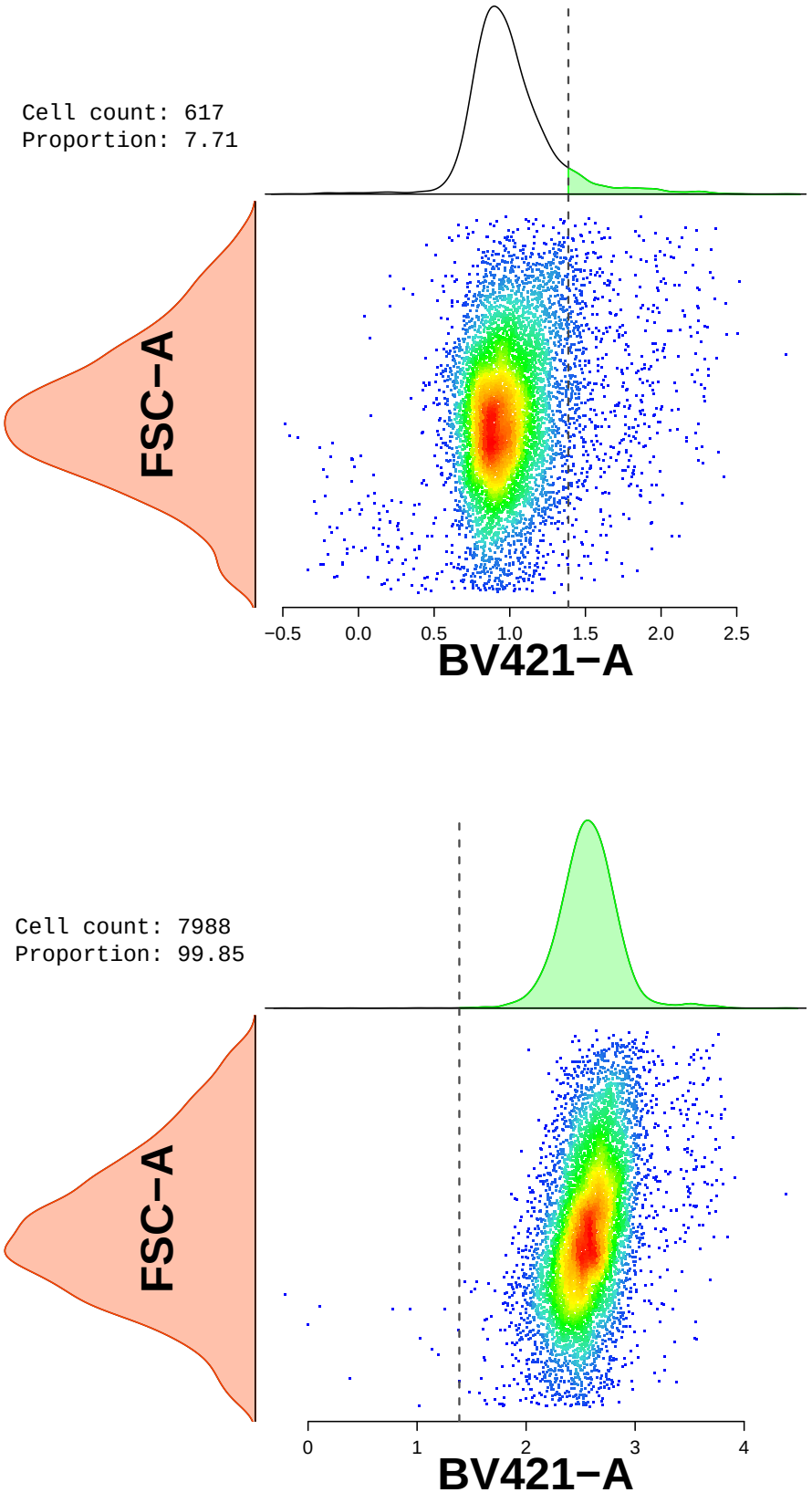

Figure 6: Default gate in FMO data applied to itself (top) and stimulated data (bottom)

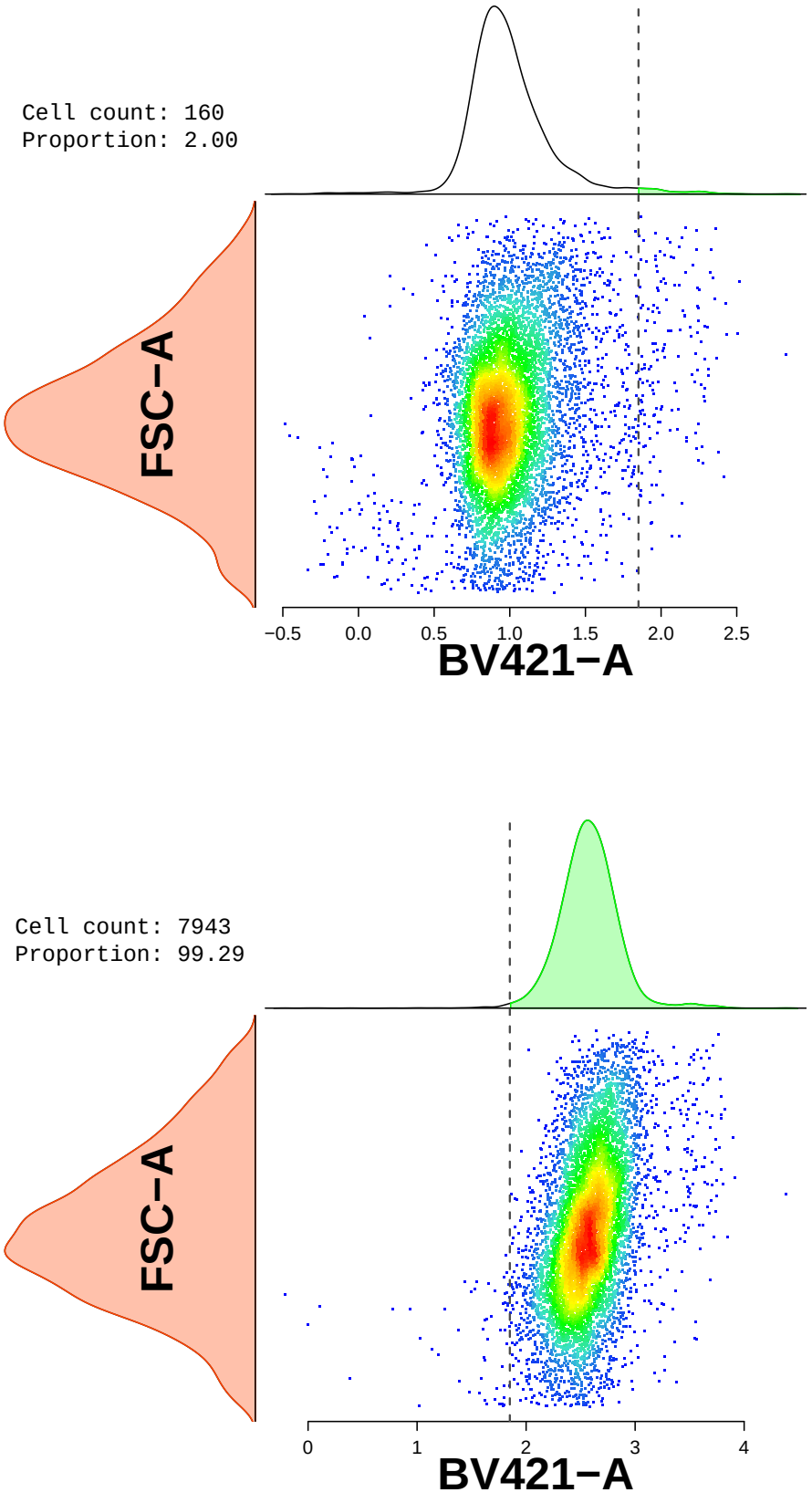

Figure 7: 98th percentile gate in FMO data applied to itself (top) and stimulated data (bottom)

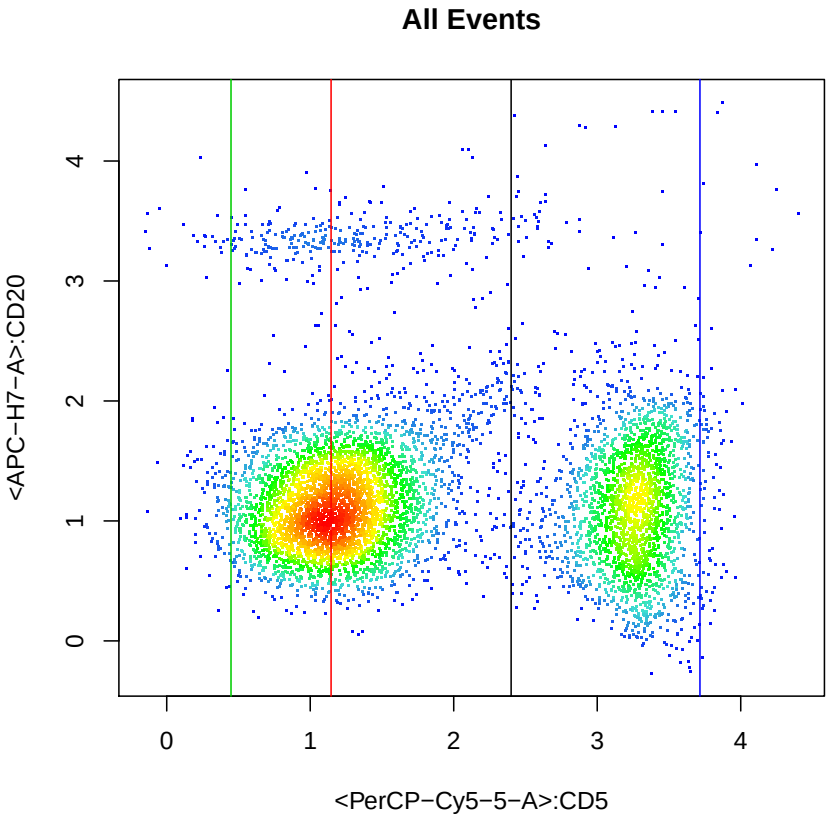

Figure 8: using upper,percentile and default of deGate