Gating

Anja Bråthen Kristoffersen

last updated

date()

## [1] "Thu Apr 07 09:49:46 2022"

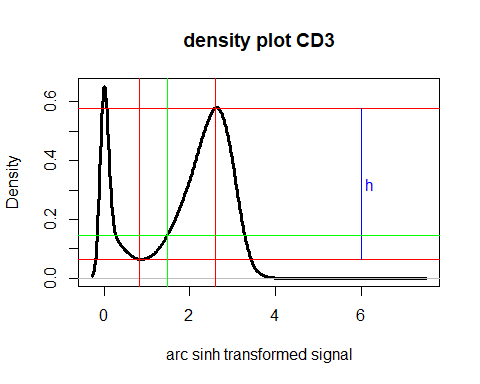
# Gating based on density plot

Different function can be used based on how the gate will be placed

## Function find\_gaussian\_gates\_second\_top()

To find lower gate for upper signal function find\_gaussian\_gates\_second\_top() can be used. It takes the variables

* data: here arc sinh transformed data are used
* channel: which marker/channel you want to gate, here CD3
* lower\_gate\_percent: tells where the lower gate is placed. See figure and explanation below, here 15%
* upper\_gate\_percent: tells where the upper gate is placed. See figure and explanation below, here not used

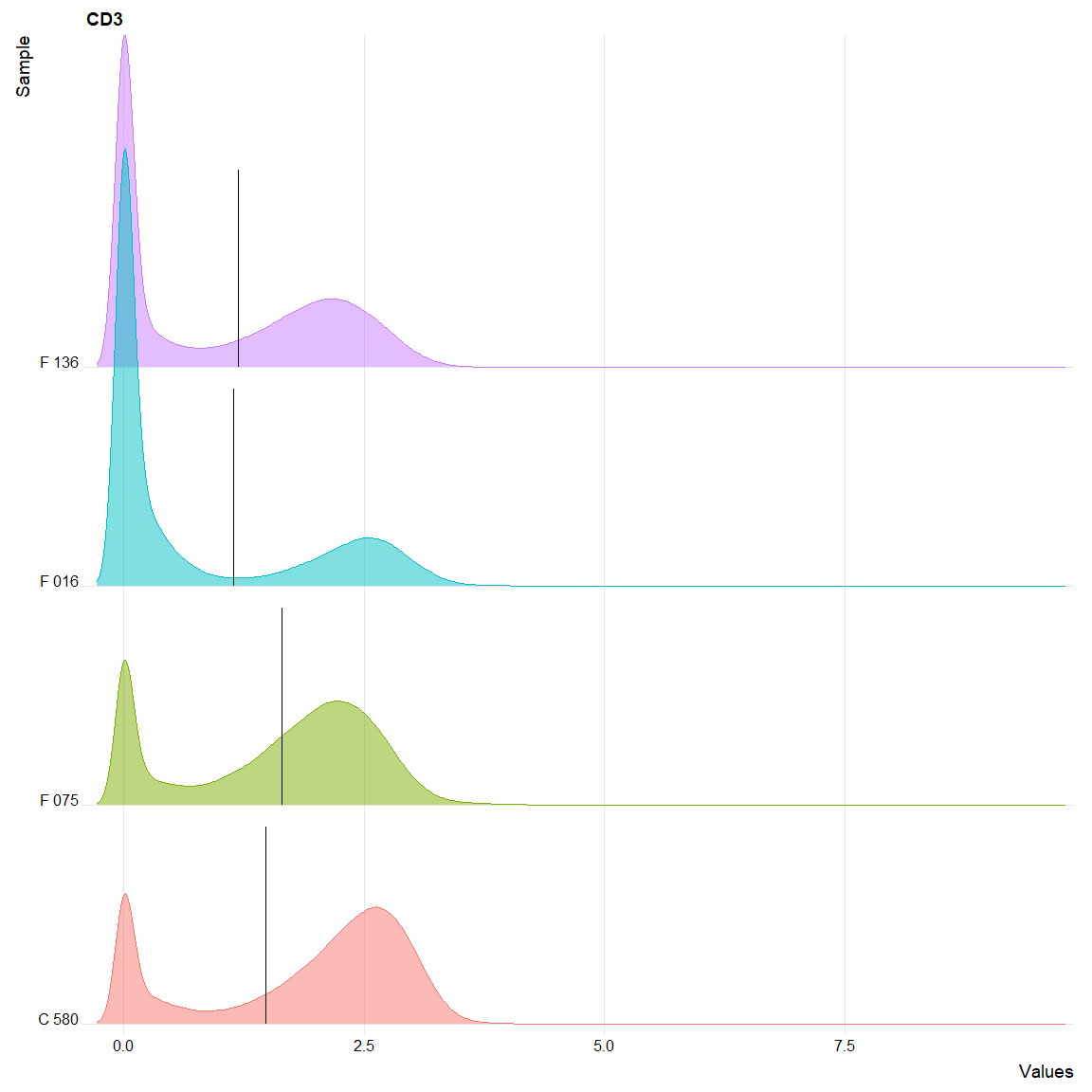


The density plot of the signal is here plotted in black. Then the first bottom and the next top are found. Here shown with red lines. The hight h between these lines are calculated, here shown in blue. The green horizontal line is found based on the lower percentage, so it is the height of the lower horizontal red line + lower\_gate\_percent/100 \* h. Where the green horizontal line first crosses the density curve after the bottom here shown with a green vertical line. The green vertical line is the lower gate produced by the function find\_gaussian\_gates\_second\_top()

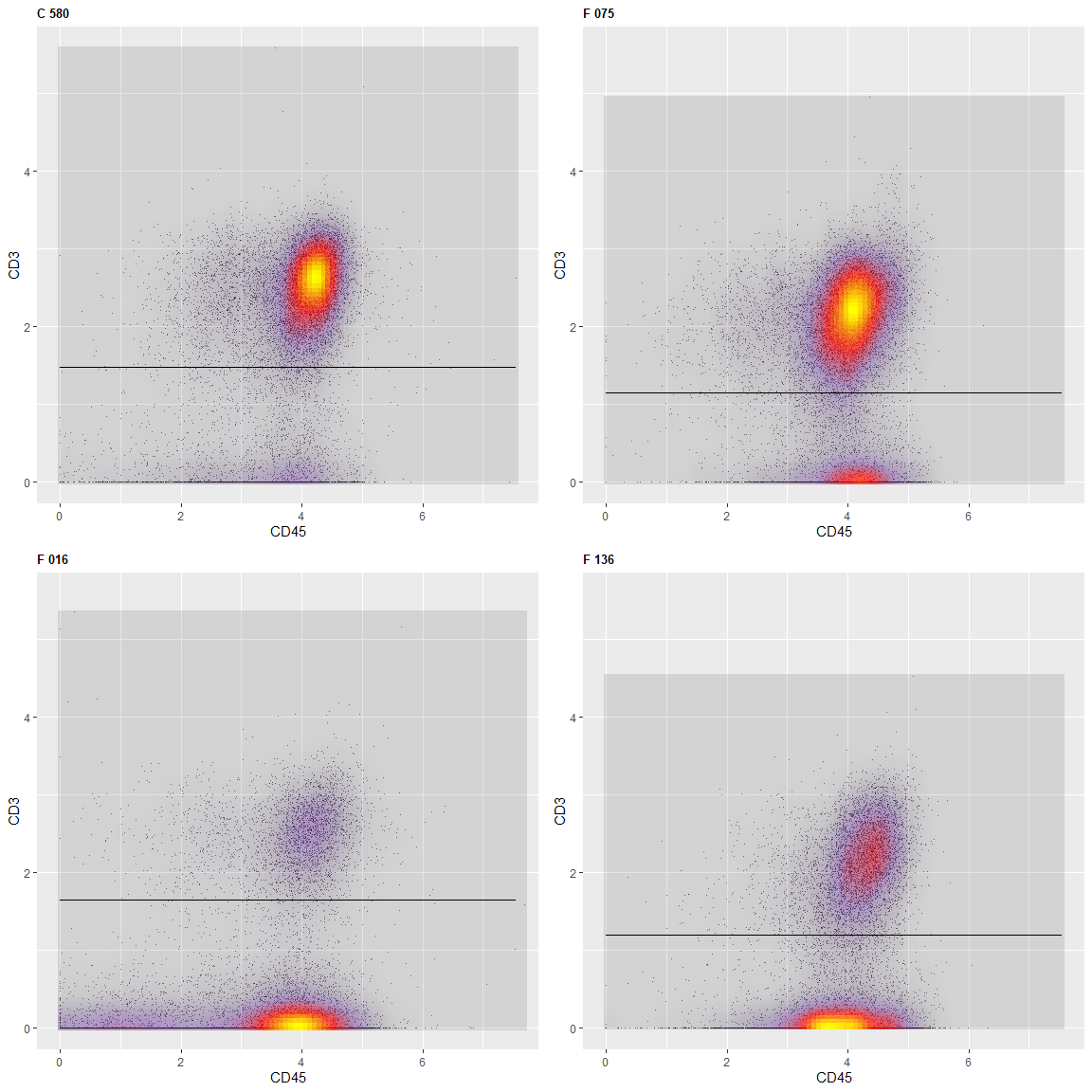
This will then be done for many files simultaneously and plotted together either as a density plot or as a signal-signal plot

split <- find\_gaussian\_gates\_second\_top(data = data, channel = kanal, lower\_gate\_percent = 15, upper\_gate\_percent = 0.001)  
  
result\_first\_gating <- split$lower\_gates  
  
density\_plots <- density\_plot(data = data, channel = kanal, plot\_title = file\_names\_short, lower\_gate = split$lower\_gates, main\_title = x)  
density\_plots

## Picking joint bandwidth of 0.0936



signal <- signal\_signal\_plot(data = data, random\_events = random\_event\_for\_plotting, channel1 = CD45, channel2 = kanal, ylow = split$lower\_gates, xname = "CD45", yname = x, plot\_title = file\_names\_short, title\_size = 10)  
plot\_multiple\_signals(signal)



## TableGrob (2 x 2) "arrange": 4 grobs  
## z cells name grob  
## 1 1 (1-1,1-1) arrange gtable[layout]  
## 2 2 (1-1,2-2) arrange gtable[layout]  
## 3 3 (2-2,1-1) arrange gtable[layout]  
## 4 4 (2-2,2-2) arrange gtable[layout]

where the gate is placed varies a little from file to file.

### save the result of the gating per cell per file

The result of the gating is here saved per cell as either 0 or 1 where 0 means lower than gate and 1 higher. If more gate you can save the level for each cell as 0, 1, 2, 3 etc. The result is saved in a list result which is designed to be equal to your list data.

result <- list()  
  
#this for loop makes a list of data frame to store the result in. All values stored by now is NA  
for(j in 1:length(data)){  
 mat <- as.data.frame(matrix(NA, ncol = 1, nrow = nrow(data[[j]])))  
 colnames(mat) <- c("above high gate")  
 result[[j]] <- mat  
}  
  
#this for loop store the information for each cell, if it is positive (above the gate) as TRUE (1) or below the gate as FALSE (0)  
for(j in 1:length(data)){  
 result[[j]][,"above high gate"] <- data[[j]][, kanal] > split$lower\_gates[j]  
}

The percentage of each file that are of a given level can be calculated from the result by using the function prosent\_per\_channel(). Here you find the percent that are higher than the found gate.

percents <- data.frame(prosent\_per\_channel(result, values = 1))  
  
  
percents

## above.high.gate  
## 1 71.43952  
## 2 70.74283  
## 3 27.08563  
## 4 46.26240

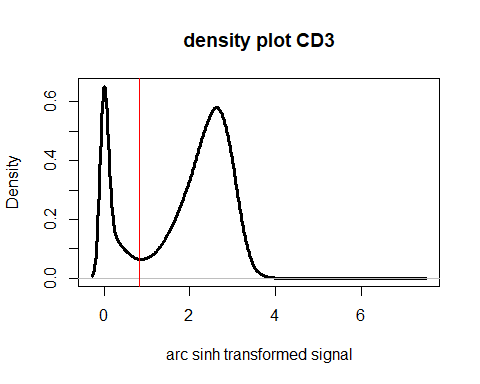
### if R have problem finding the gate you want.

* you can try to use the variable minimum in find\_gaussian\_gates\_second\_top(). That will force the dataset to either use this value as the minimum or find the first minimum above this value as the minimum.
* you might also just define your own gate. By using the same value for all files or only overwrite the gate with a value for some selected files.

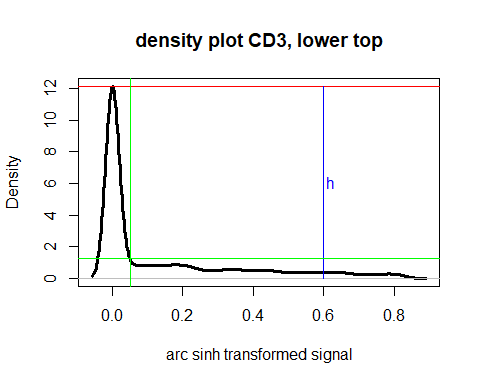
## Function find\_gaussian\_gates\_first\_top()

If you rather want to place the gates based on the first top you can use the function find\_gaussian\_gates\_first\_top(). It takes the variables:

* data: here arc sinh transformed data are used
* channel: which marker/channel you want to gate, here CD3
* lower\_gate\_percent: tells where the lower gate is placed. See figure and explanation below, here not used
* upper\_gate\_percent: tells where the upper gate is placed. See figure and explanation below, here 10 %



The first bottom is found, here shown with a red line, and everything above the red line is not used further in the calculation. Make a new density plot of the lower signal to find the gate.



## [1] 0.04990727

The density plot of the lower signal is here plotted in black. Then the first top is found. Here shown with red horizontal lines. The hight h between this line and 0 are calculated, here shown in blue. The green horizontal line is found based on the upper percentage, so it is upper\_gate\_percent/100 \* h. Where the green horizontal line first crosses the density curve after the top here shown with a green vertical line. The green vertical line is the upper gate produced by the function find\_gaussian\_gates\_first\_top()

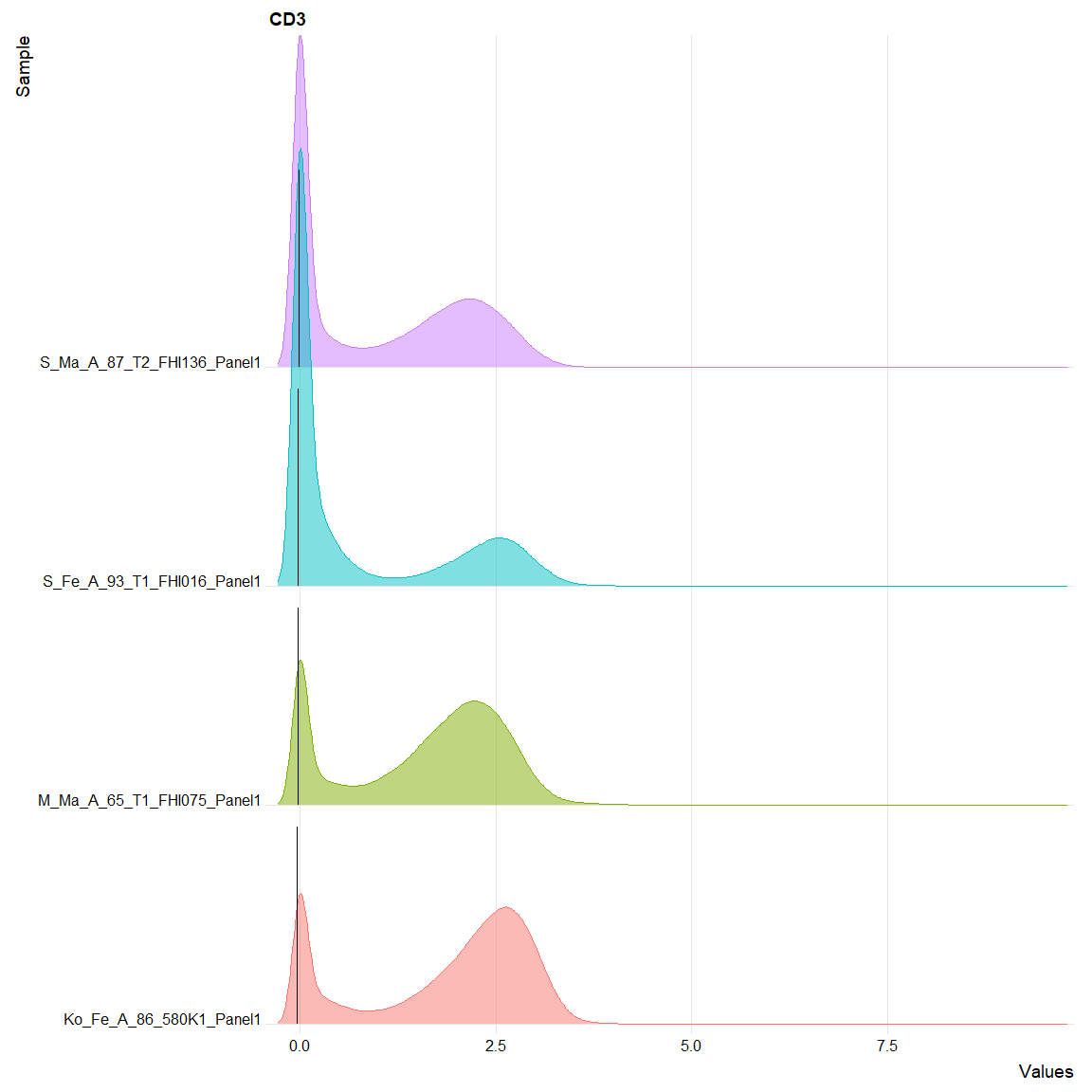
This will then be done for many files simultaneously and plotted together either as a density plot or as a signal-signal plot

split <- find\_gaussian\_gates\_first\_top(data = data, channel = kanal, lower\_gate\_percent = 10, upper\_gate\_percent = 10)  
split$upper\_gates

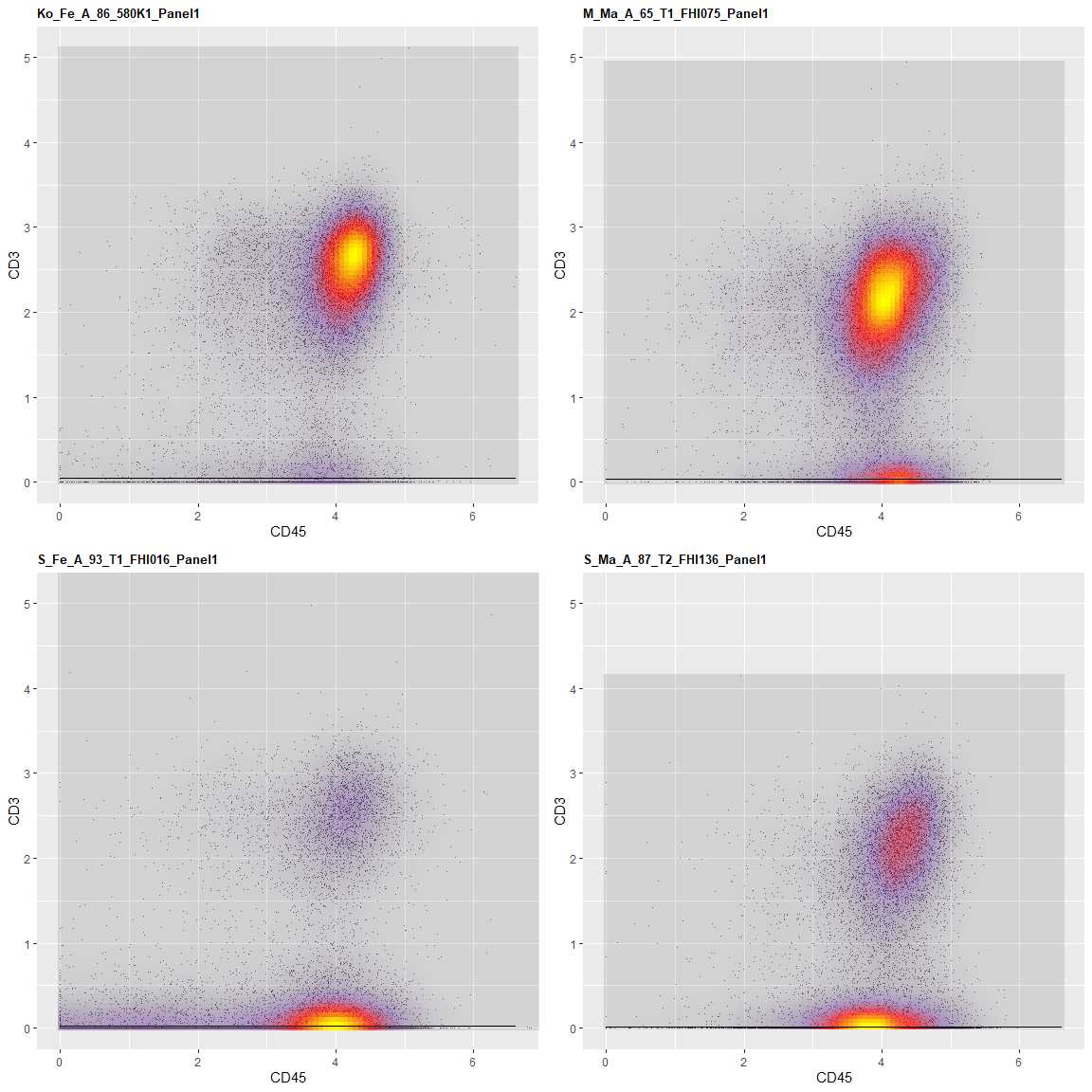
## [1] 0.049907272 0.030980456 0.027901743 0.007379025

density\_plots <- density\_plot(data = data, channel = kanal, plot\_title = file\_names, lower\_gate = split$lower\_gates, main\_title = x)  
density\_plots

## Picking joint bandwidth of 0.0936



result\_second\_gating <- split$upper\_gates  
  
  
signal <- signal\_signal\_plot(data = data, random\_events = random\_events(number\_of\_events(data)), channel1 = CD45, channel2 = kanal, yhigh = split$upper\_gates, xname = "CD45", yname = x, plot\_title = file\_names, title\_size = 10)  
plot\_multiple\_signals(signal)



## TableGrob (2 x 2) "arrange": 4 grobs  
## z cells name grob  
## 1 1 (1-1,1-1) arrange gtable[layout]  
## 2 2 (1-1,2-2) arrange gtable[layout]  
## 3 3 (2-2,1-1) arrange gtable[layout]  
## 4 4 (2-2,2-2) arrange gtable[layout]

Unfortunately, it looks like the gate is placed in the middle of the top. This is “just” a smoothing and visualisation problem. If we count how many cells that are above the upper gate found here we save the result of the gating in the list result and use the function prosent\_per\_channel() to find how many percent of the cells in each file that are above the gate.

result <- list()  
for(j in 1:length(data)){  
 mat <- as.data.frame(matrix(NA, ncol = 2, nrow = nrow(data[[j]])))  
 colnames(mat) <- c("above first gate", "above second gate")  
 result[[j]] <- mat  
}  
  
for(j in 1:length(data)){  
 result[[j]][,"above first gate"] <- data[[j]][, kanal] > result\_first\_gating[j]  
 result[[j]][,"above second gate"] <- data[[j]][, kanal] > result\_second\_gating[j]  
}  
  
  
  
percents <- data.frame(prosent\_per\_channel(result, values = 1))  
percents$above.first.gate

## [1] 71.43952 70.74283 27.08563 46.26240

Since we now have the percentages above each of the gates we can also calculate the percentages between the gates.

percents$above.second.gate - percents$above.first.gate

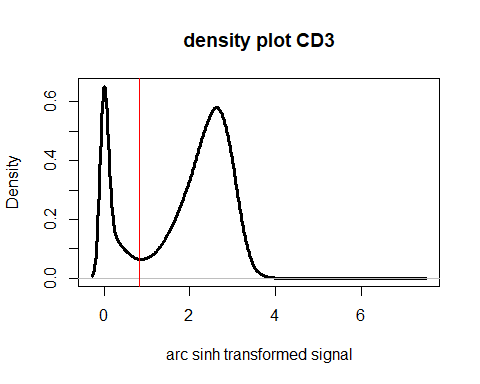
## [1] 14.57035 13.82924 26.62275 17.83158

## Function find\_split\_first\_second\_top()

If you rather want to place the gates based on the minimum between the first and second top you can use the function find\_split\_first\_second\_top(). It takes the variables:

* data: here arc sinh transformed data are used
* channel: which marker/channel you want to gate, here CD3
* minimum: which is the minimum value that you want the gate to have, if not used the default is 0

Here the gates are found as the minimum between the first and second top as shown in the figure below.



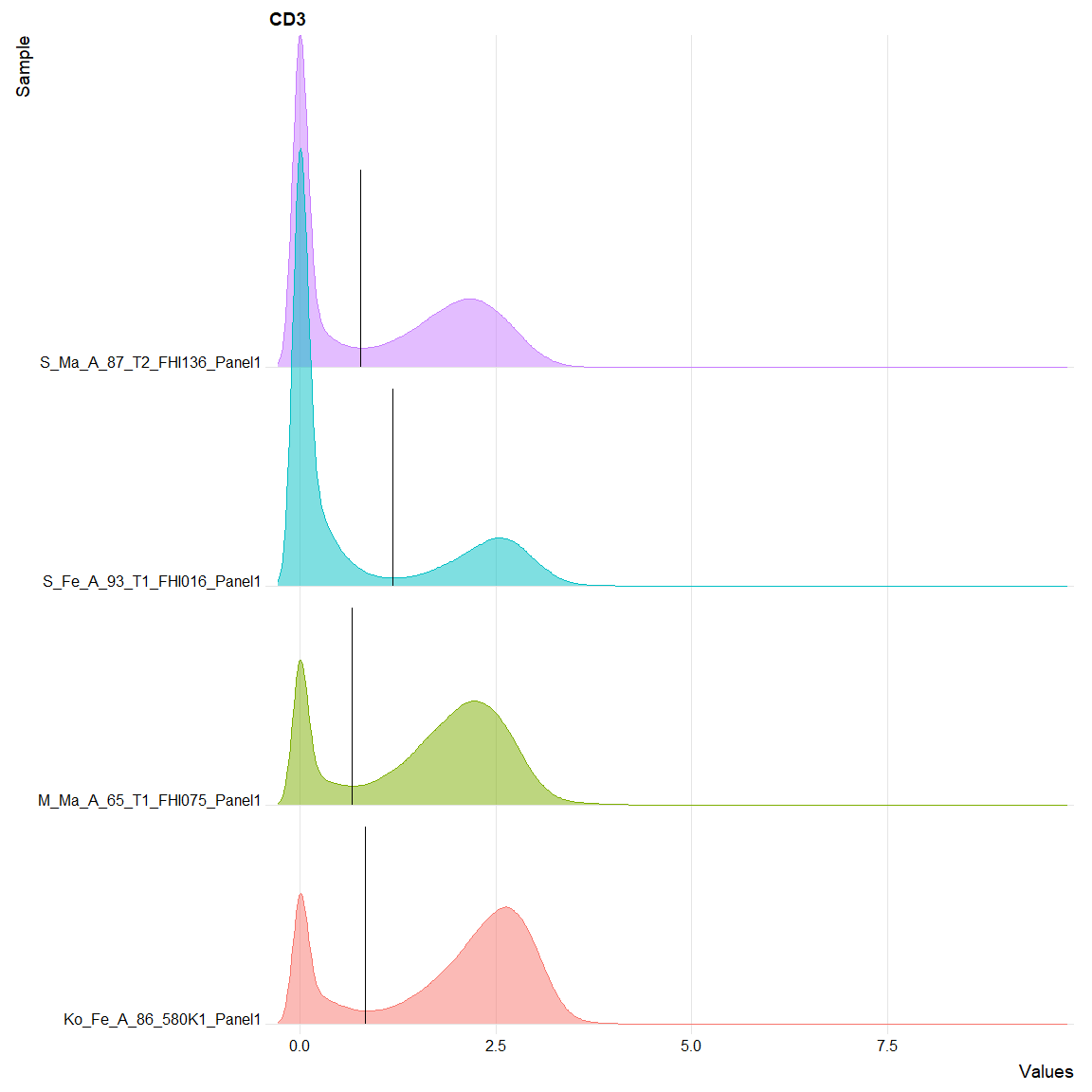
The code for doing this is, where also density plot and signal-signal plots are made.

split <- find\_split\_first\_second\_top(data = data, channel = kanal)  
  
split

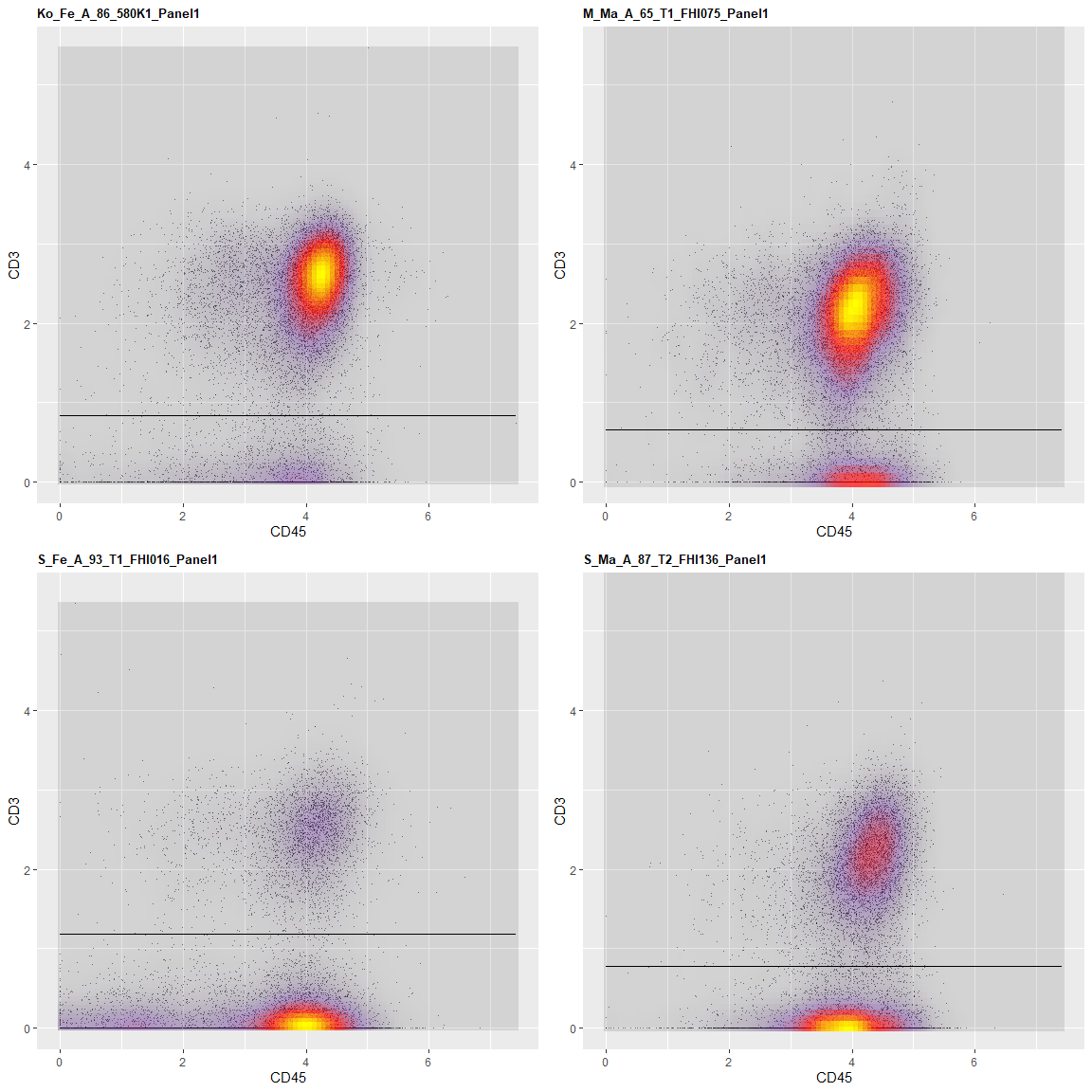
## [1] 0.8338336 0.6609022 1.1869257 0.7786559

density\_plots <- density\_plot(data = data, channel = kanal, plot\_title = file\_names, lower\_gate = split, main\_title = x)  
density\_plots

## Picking joint bandwidth of 0.0936



result\_third\_gating <- split  
  
  
signal <- signal\_signal\_plot(data = data, random\_events = random\_events(number\_of\_events(data)), channel1 = CD45, channel2 = kanal, yhigh = split, xname = "CD45", yname = x, plot\_title = file\_names, title\_size = 10)  
plot\_multiple\_signals(signal)



## TableGrob (2 x 2) "arrange": 4 grobs  
## z cells name grob  
## 1 1 (1-1,1-1) arrange gtable[layout]  
## 2 2 (1-1,2-2) arrange gtable[layout]  
## 3 3 (2-2,1-1) arrange gtable[layout]  
## 4 4 (2-2,2-2) arrange gtable[layout]

The data is then stored in result.

result <- list()  
for(j in 1:length(data)){  
 mat <- as.data.frame(matrix(NA, ncol = 3, nrow = nrow(data[[j]])))  
 colnames(mat) <- c("above first gate", "above second gate", "above third gate")  
 result[[j]] <- mat  
}  
  
for(j in 1:length(data)){  
 result[[j]][,"above first gate"] <- data[[j]][, kanal] > result\_first\_gating[j]  
 result[[j]][,"above second gate"] <- data[[j]][, kanal] > result\_second\_gating[j]  
 result[[j]][,"above third gate"] <- data[[j]][, kanal] > result\_third\_gating[j]  
}

And the percentages above each of the gates found by using the function prosent\_per\_channel()

percents <- data.frame(prosent\_per\_channel(result, values = c(1, 1, 1)))  
percents

## above.first.gate above.second.gate above.third.gate  
## 1 71.43952 86.00987 77.07246  
## 2 70.74283 84.57207 76.26572  
## 3 27.08563 53.70839 29.33951  
## 4 46.26240 64.09398 50.64475