flowLearn training/test data for IMPC

This worksheet demonstrates the use of training and test data for the IMPC bone marrow panel. The data is organized into one .rmd file per sample. Each file contains a list of gates with each gate having densities (using different granularities), threshold values, and a list of booleans indicating the events belonging to the gated population.

Libraries and functions

```
source("functions.R")
```

Parameters

Granularities were calculated for $granularity \in \{8, 16, 32, 64, 128, 256, 512\}$ features

```
# Granularity of the density, i.e. number of features
granularity <- "64"

# Population of interest
population <- "NOT(Plasma)"
```

Loading data

One file per sample, meaning one file corresponds to one training vector.

```
# Path to training files
dirName <- 'trainingFiles'
sampleName <- 'BM 8,2f,24,2f,15_L000096187_016.labelled'
sample <- readRDS(paste(dirName, '/', sampleName, '.rds', sep = ''))
```

Accessing data

The loaded sample contains a list with a number of fields named by the gated population.

```
# Print list of populations
print(names(sample))
```

```
##  [1] "Singlets"            "Live"                "Lymphocytes"
##  [4] "CD45"                "NOT(Granulocyte Pre)" "Granulocyte Pre"
##  [7] "CD3 T-cell"          "NOT(CD3 T-cell)"     "Plasma"
## [10] "NOT(Plasma)"         "B-cell"              "Myeloid"
## [13] "CD43+"               "CD43-"               "HFA"
## [16] "HFB"                 "HFC"                 "HFD"
## [19] "HFE"                 "HFF"
```

Each list entry contains a object of class TrainingGate.

```
print(class(sample[[population]]))
```

```
## [1] "TrainingGate"
## attr(,"package")
## [1] ".GlobalEnv"
```

The slots are:

```
print(slotNames("TrainingGate"))
```

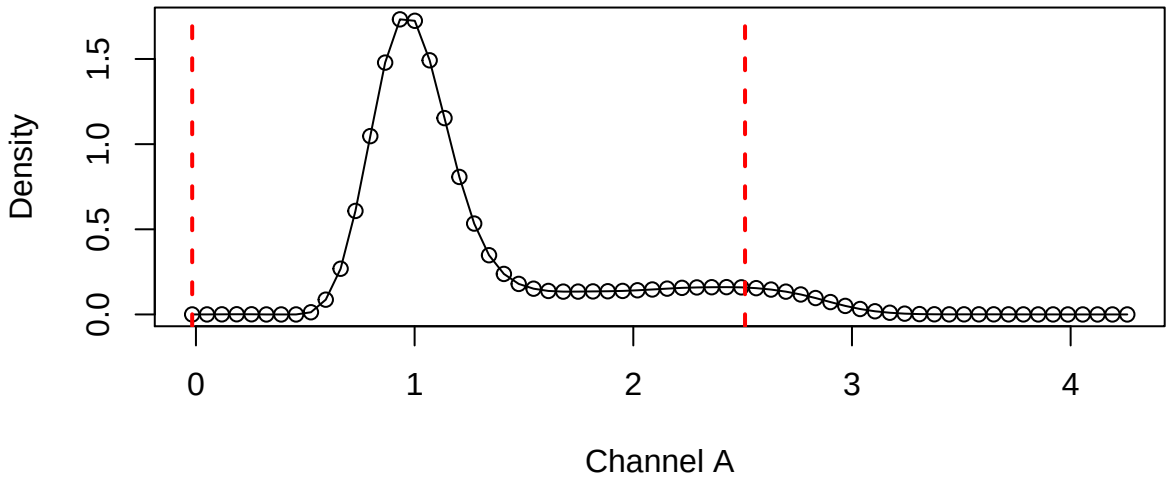
```
##  [1] "parentName"      "channelA"        "channelB"
##  [4] "thresholdALow"   "thresholdAHigh"  "thresholdBLow"
##  [7] "thresholdBHigh"  "negate"          "gateAssignments"
## [10] "densitiesA"      "densitiesB"
```

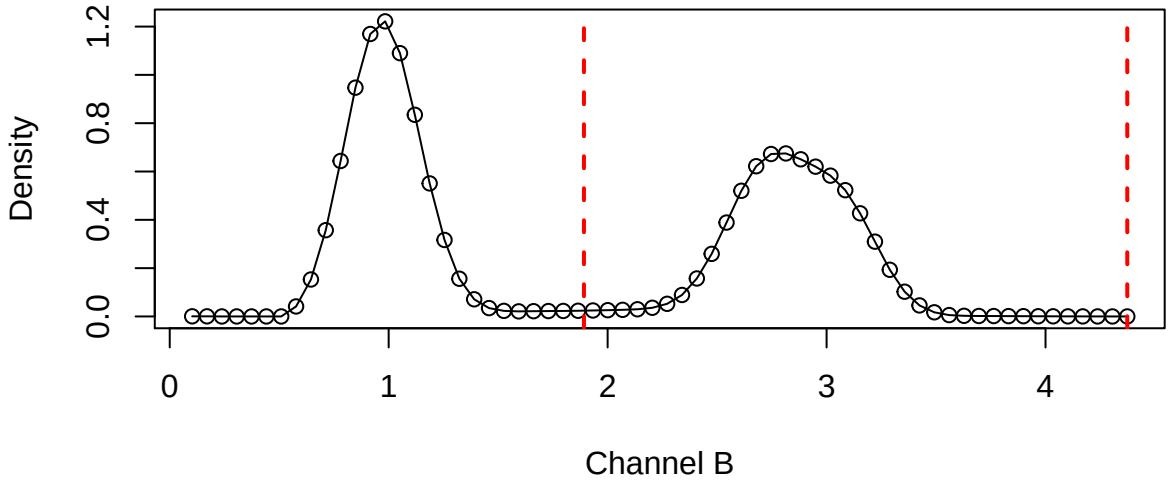
- `parentName`: Name of the parent population
- `channelA`: Index of the first channel (A)
- `channelB`: Index of the second channel (B)
- `thresholdALow`: Low threshold of the channel A. The threshold might be `NaN` indicating that only one or no threshold was used for gating the population using this channel.
- `thresholdAHigh`: High threshold of the channel A. The threshold might be `NaN` indicating that only one or no threshold was used for gating the population using this channel.
- `thresholdBLow`: Low threshold of the channel B. The threshold might be `NaN` indicating that only one or no threshold was used for gating the population using this channel.
- `thresholdBHigh`: High threshold of the channel B. The threshold might be `NaN` indicating that only one or no threshold was used for gating the population using this channel.
- `negate`: TRUE if the population was negated, meaning that not the gated population is of interest but everything else
- `gateAssignments`: List of logicals with length = number of events, where each entry is TRUE if the event belongs to the population of interest
- `densitiesA`: A list with calculated densities for channel A for different granularities. Each entry is the returned value of Rs density function.
- `densitiesB`: A list with calculated densities for channel B for different granularities. Each entry is the returned value of Rs density function.

Plotting densities and thresholds

I provided a convenience function for plotting the densities of channel A and B for one particular gate. Have a look at this function to know how to use the different low and high threshold values.

```
plotGate(sample[[population]], granularity)
```





Accessing density data

For each granularity, there is a density object containing different fields. The most important fields are `x` and `y` describing the density distribution.

```
x <- sample[[population]]@densitiesA[[granularity]]$x
y <- sample[[population]]@densitiesA[[granularity]]$y
```