flowBin: a Package for Combining Multitube Flow Cytometry Data

Kieran O'Neill'

October 30, 2025

koneill@bccrc.ca

Contents

1 Licensing

Under the Artistic License, you are free to use and redistribute this software.

2 Introduction

flowBin is a package for combining multitube flow cytometry data together using common population markers included in each tube.

3 An example of constructing a flowExprSet de novo

Although for the main example we will use a pre-constructed flowSample, it is useful to understand how one may be constructed de novo (using the example data from flowFP):

```
> library(flowBin)
> library(flowFP)
> data(fs1)
> show(fs1)
```

```
A flowSet with 7 experiments.

column names(7): FS Lin SS Log ... FL4 Log FL5 Log
```

Let's take a look at the parameters stored in this flowSet:

```
> fsApply(fs1, function(x){x@parameters@data[,'desc']})
```

```
                    $P1S     $P2S     $P3S           $P4S         $P5S
FI05_000942_001.LMD "FS Lin" "SS Log" "IgG1-FITC"    "IgG1-PE"    "CD45-ECD"
FI05_000942_002.LMD "FS Lin" "SS Log" "Kappa-FITC"   "Lambda-PE"  "CD45-ECD"
FI05_000942_003.LMD "FS Lin" "SS Log" "CD7-FITC"     "CD4-PE"     "CD45-ECD"
FI05_000942_004.LMD "FS Lin" "SS Log" "CD15-FITC"    "CD13-PE"    "CD45-ECD"
FI05_000942_005.LMD "FS Lin" "SS Log" "CD14-FITC"    "CD11c-PE"   "CD45-ECD"
FI05_000942_006.LMD "FS Lin" "SS Log" "HLA-DR-FITC"  "CD117-PE"   "CD45-ECD"
FI05_000942_007.LMD "FS Lin" "SS Log" "CD5-FITC"     "CD19-PE"    "CD45-ECD"
                    $P6S       $P7S
FI05_000942_001.LMD "IgG1-PC5" "IgG1-PC7"
FI05_000942_002.LMD "CD19-PC5" "CD20-PC7"
FI05_000942_003.LMD "CD8-PC5"  "CD2-PC7"
FI05_000942_004.LMD "CD16-PC5" "CD56-PC7"
FI05_000942_005.LMD "CD64-PC5" "CD33-PC7"
FI05_000942_006.LMD "CD34-PC5" "CD38-PC7"
FI05_000942_007.LMD "CD3-PC5"  "CD10-PC7"
```

P1, P2 and P5 are our common parameters, while 3,4,6,7 are the tube-specific measurement parameters. Also, tube 1 appears to contain isotype controls (IgG1). So, to make a flowSample:

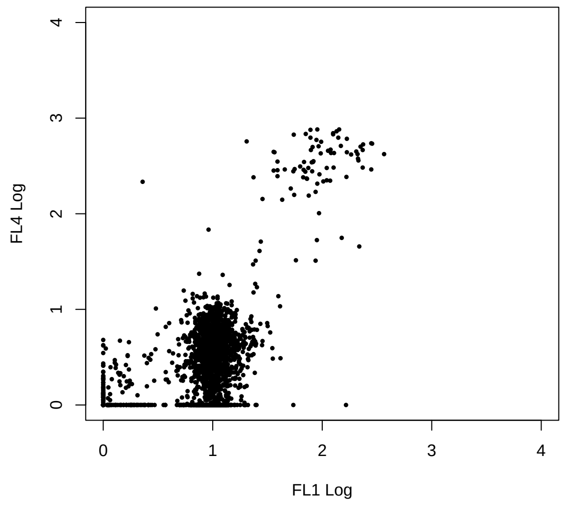
```
> aml.sample <- new('FlowSample',
+                                                           name='Example flowSample',
+                              tube.set=as.list(fs1@frames),
+                              control.tubes=c(1),
+                              bin.pars=c(1,2,5),
+                              measure.pars=list(c(3,4,6,7)))
> show(aml.sample)

A FlowSample containing 7 tubes

Control tube(s):  1
Parameters to bin:  1 2 5
Parameters to measure: 3 4 6 7

>
```

4 Looking a little more closly at the example data

For our example, we will use a leukemia diagnostic panel for one patient, downsampled for inclusion in the package. This example data set was taken from FlowRepository data set `FR-FCM-ZZYA`, which contains full data for 359 patients, and is a good data set to try out flowBin on. (It is also most likely the same data set which the flowFP example was taken from).

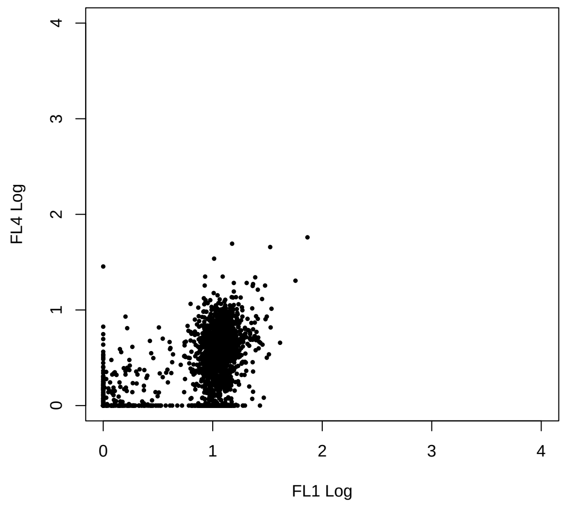
Let's plot two of the population (binning) markers in tube 1:

```
> data(amlsample)
> tube1.frame <- tube.set(aml.sample)[[1]]
> show(tube1.frame)

flowFrame object '0409.FCS'
with 2000 cells and 7 observables:
       name      desc     range    minRange  maxRange
$P1  FS Lin    FS Lin      1024 0.000000000      1023
$P2  SS Log    SS Log      1024 0.000400391         4
$P3 FL1 Log IgG1-FITC      1024 0.000400391         4
$P4 FL2 Log   IgG1-PE      1024 0.000400391         4
$P5 FL3 Log  CD45-ECD      1024 0.000400391         4
$P6 FL4 Log  IgG1-PC5      1024 0.000400391         4
$P7 FL5 Log  IgG1-PC7      1024 0.000400391         4
180 keywords are stored in the 'description' slot

> plot(exprs(tube1.frame)[,c(5,2)], pch=16, cex=0.6, xlim=c(0,4), ylim=c(0,4))
```



Let's look at two of the measurment markers in tube 1, which is a negative control tube:

```
> plot(exprs(tube1.frame)[,c(3,6)], pch=16, cex=0.6, xlim=c(0,4), ylim=c(0,4))
```

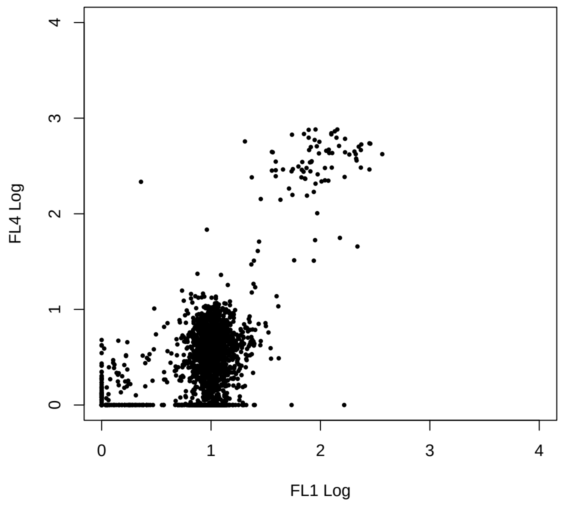


We can see that they are well in the lower end of the expression range. By contrast, here are the same channels in a measurement tube, with specific antibodies conjugated to them:

```
> tube7.frame <- tube.set(aml.sample)[[7]]
> show(tube7.frame)
```

```
flowFrame object '0415.FCS'
with 2000 cells and 7 observables:
       name     desc     range    minRange  maxRange
$P1  FS Lin   FS Lin      1024 0.000000000      1023
$P2  SS Log   SS Log      1024 0.000400391         4
$P3 FL1 Log CD5-FITC      1024 0.000400391         4
$P4 FL2 Log  CD19-PE      1024 0.000400391         4
$P5 FL3 Log CD45-ECD      1024 0.000400391         4
$P6 FL4 Log  CD3-PC5      1024 0.000400391         4
$P7 FL5 Log CD10-PC7      1024 0.000400391         4
180 keywords are stored in the 'description' slot
```

```
> plot(exprs(tube7.frame)[,c(3,6)], pch=16, cex=0.6, xlim=c(0,4), ylim=c(0,4))
```



Notice how most of the cells are still in the negative region, but there is a clear (but small) positive population.

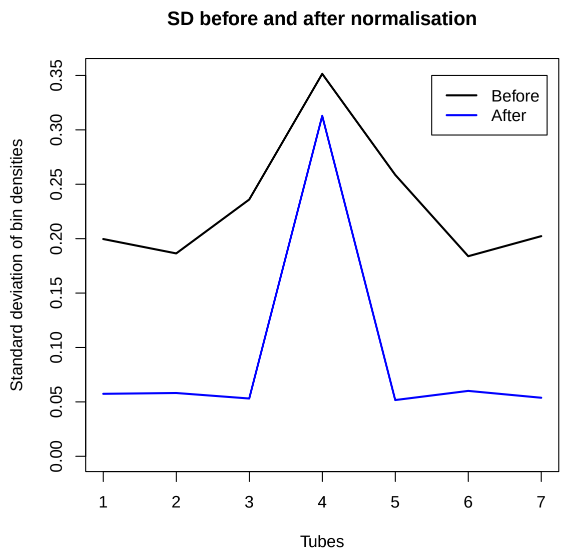
5 Quantile normalisation

While the negative controls can allow us to account for minor technical variation across tubes in the measurement markers, we do not have that luxury for the binning markers. However, since each tube is an aliquot from a common biological sample, stained with the same antibodies, we expect that they should all have the same underlying distribution. So, flowBin provides functionality to quantile normalise the binning markers across tubes.

```
> normed.sample <- quantileNormalise(aml.sample)
```

There is also a function to perform a quick check on the performance of the normalisation, using the quality control functionality of flowFP to measure the average standard deviation in probability bin densities across tubes. Here we plot the before and after densities (lower is better).

```
> qnorm.check <- checkQNorm(aml.sample, normed.sample, do.plot=F)
> plot(qnorm.check$sd.before, type='l', lwd=2,
+          ylim=c(0, max(qnorm.check$sd.before)),
+          xlab='Tubes',
+          ylab='Standard deviation of bin densities',
+          main='SD before and after normalisation')
> lines(qnorm.check$sd.after, lwd=2, col='blue')
> legend(x=5.5, y=0.35, legend=c('Before', 'After'), lwd=2, col=c('black', 'blue'))
```



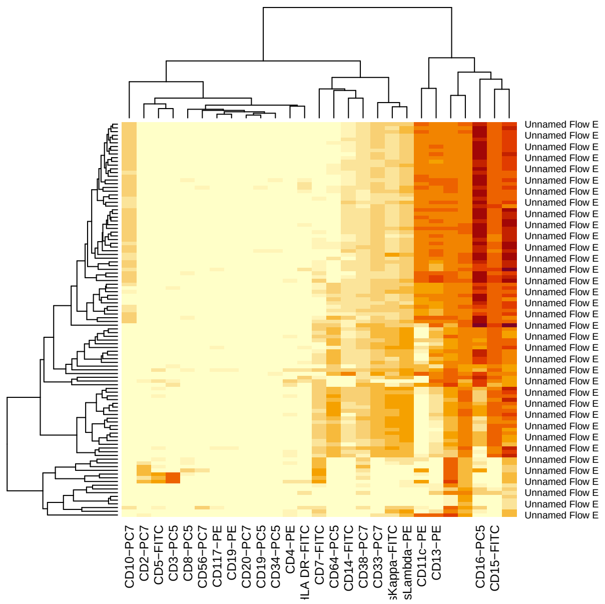
Quantile normalisation definitely seems to have improved things, although tube 4 might be worth examining for QC purposes. For this example, we'll run with it, though.

6 Running flowBin

```
> tube.combined <- flowBin(tube.list=aml.sample@tube.set,
+ bin.pars=aml.sample@bin.pars,
+ control.tubes=aml.sample@control.tubes,
+ expr.method='medianFIDist',
+ scale.expr=T)
```

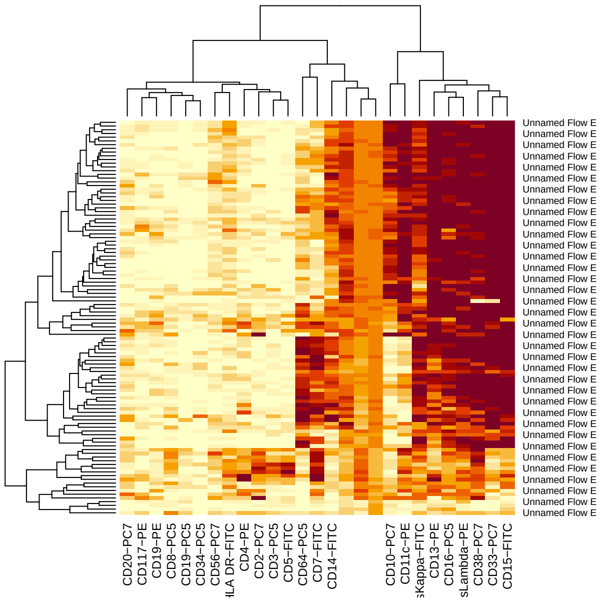
We use scale.expr to scale the results to the interval $[0, 1]$ by dividing by their range as specified in the flowFrame. For our example, this puts the FSC channel on the same scale as the others, fascilitating plotting (and other downstream uses):

```
> heatmap(tube.combined, scale='none')
```



We can try another method of determining bin expression, propPos, which sets cutoffs at the 98th percentile of the negative control, and counts what proportion of events fall above the cutoff.

```
> tube.propPos <- flowBin(tube.list=aml.sample@tube.set,
+ bin.pars=aml.sample@bin.pars,
+ control.tubes=aml.sample@control.tubes,
+ expr.method='propPos',
+ scale.expr=T)

> heatmap(tube.propPos, scale='none')
```


Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
Unnamed Flow E
CD20−PC7
CD117−PE
CD19−PE
CD8−PC5
CD19−PC5
CD34−PC5
CD56−PC7
ILA DR−FITC
CD4−PE
CD2−PC7
CD3−PC5
CD5−FITC
CD64−PC5
CD7−FITC
CD14−FITC
CD10−PC7
CD11c−PE
sKappa−FITC
CD13−PE
CD16−PC5
sLambda−PE
CD38−PC7
CD33−PC7
CD15−FITC