

Flow cytometry workflow for active, dormant, and dead microbial populations

M.L. Larsen

08 September, 2015

Overview

The microbes within communities have wide ranging metabolic activities that are influenced by resource availability. We can classify their metabolic activity with DNA and RNA content. Here, we seek to enumerate the fractions of the active (greater RNA than DNA), dormant (less RNA to DNA), and dead populations of microbes within freshwater ecosystems.

Document workflow

1. Setup work environment, load flow cytometry data for processing
2. Diagnostics with control data
3. Define static and dynamics gates
4. Process data files
5. University Lake Time Series Analysis
6. Indiana Ponds Analysis

Data Collection

Data for this project were collected as part of a long term sampling regime on University Lake in Bloomington, Indiana and wildlife refuge ponds scattered through south central Indiana. All data were collected in the Indiana University Flow Cytometry Core Facility operated by Christiane Hassel and processed by M.L. Larsen.

Data Processing

1. R Version and Package information

Source code used in this workflow were provided by R. Nikolic.

All analyses were completed with R version 3.2.2 (2015-08-14) and the following packages:

<i>Package</i>	<i>Version</i>	<i>Module Description</i>
flowCore	1.34.9	data gating
flowQ	1.28.0	flow data quality assurance
flowViz	1.32.0	visualization of flow data

1. Load data

The data for this project were collected in the IU Flow Cytometry Core Facility (C. Hassel) using eFluor fixed viability dye, Hoescht 33342, and Pyronin Y.

```
## A flowSet with 7 experiments.  
##  
##   column names:  
##   FSC-A SSC-A FSC PMT-A Alexa Fluor 488-A PI (B)-A Pacific Blue-A APC-A
```

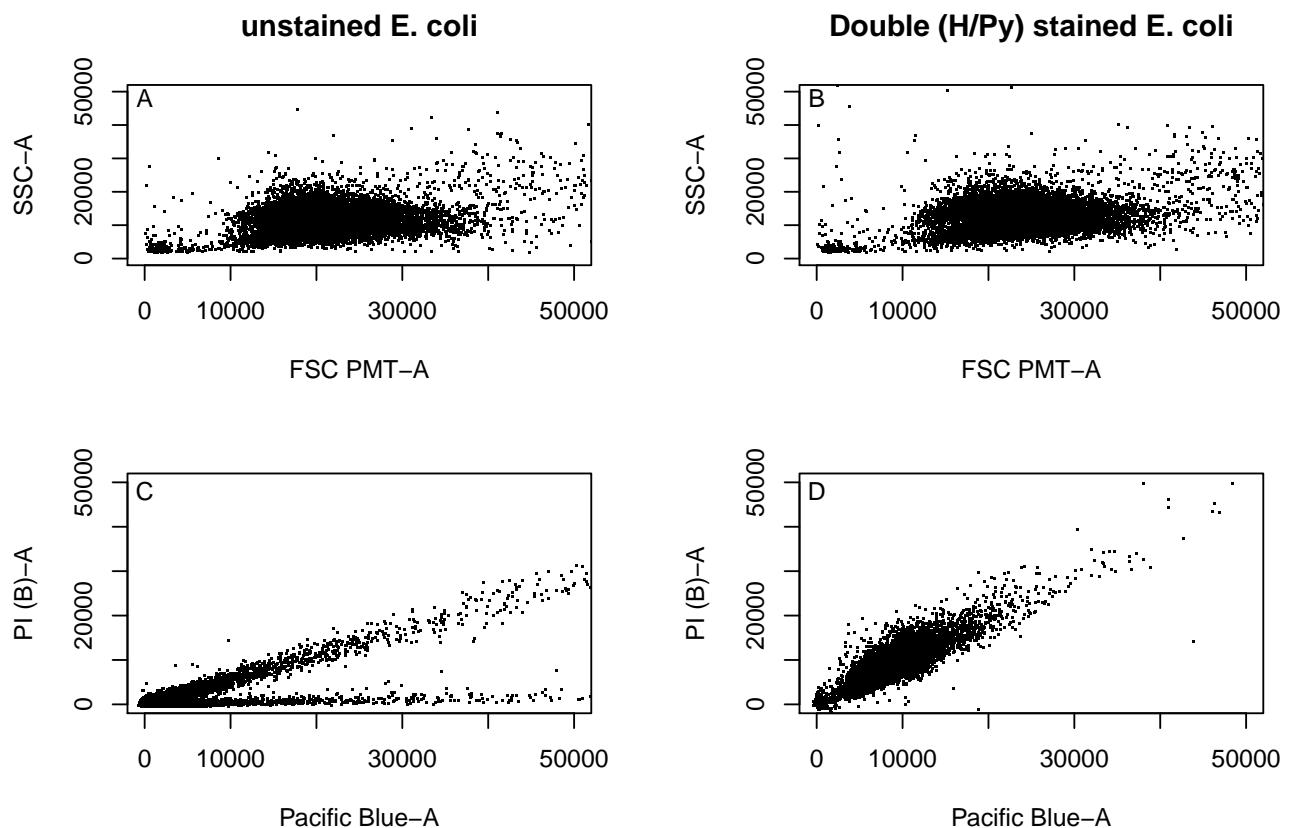
This experiment contains 7 samples.

2. Diagnostics with control data

2.1 Initial Visualization

Diagnostically evaluating the control data can help establish instrument malfunction, poor run quality, or the need for data preprocessing (e.g. compensations, transformation, etc.). `flowViz` provides the functionality to visually evaluate the data while `f1owQ` analyzes data quality assurance.

Bivariate density plots between stained and unstained samples show shifts in the data clustering (Figure 3 B,D). The *E. coli* standards in our control batch show very similar distributions in the FSC v SSC plots, however, the distribution changes substantially with the Pacific Blue v PI plot because of the presence of the stain in the cells. We should expect a diagonal shift up in the stained group.



2.2 Preprocessing

2.2.1 Compensation

Compensation is a normalization step by which excess fluorescence from the events read is removed.

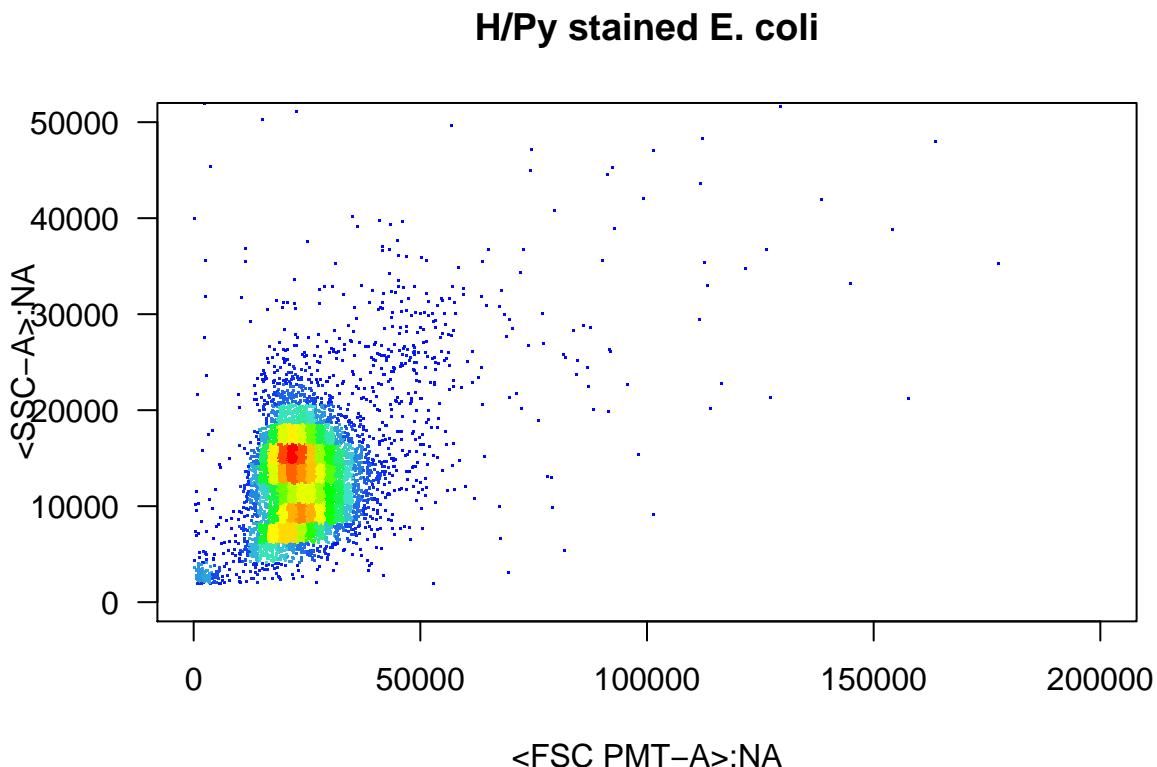
```
fs.controls[[1]]@description$'SPILL'

fs.controls.comp <- fsApply(fs.controls,function(frame){
  #extract compensation matrix from keywords
  comp <- keyword(frame)$`SPILL`
  new_frame <- compensate(frame,comp)
  new_frame
})

fs.controls.comp
summary(fs.controls.comp)
```

2.2.2 Visualize and Remove Margin events

Boundary, or marginal events, are common in the generation of flow cytometry data and should be considered as noise. These are events that fall out of the dynamic range that the flow cytometer can detect a signal. Before moving forward with the transformation, we must first classify and remove these boundary events with `flowQ` and `qaProcess.marginevents`, `qaProcess.cellnumber`, `qaProcess.timeline` and `qaProcess.timeflow`.

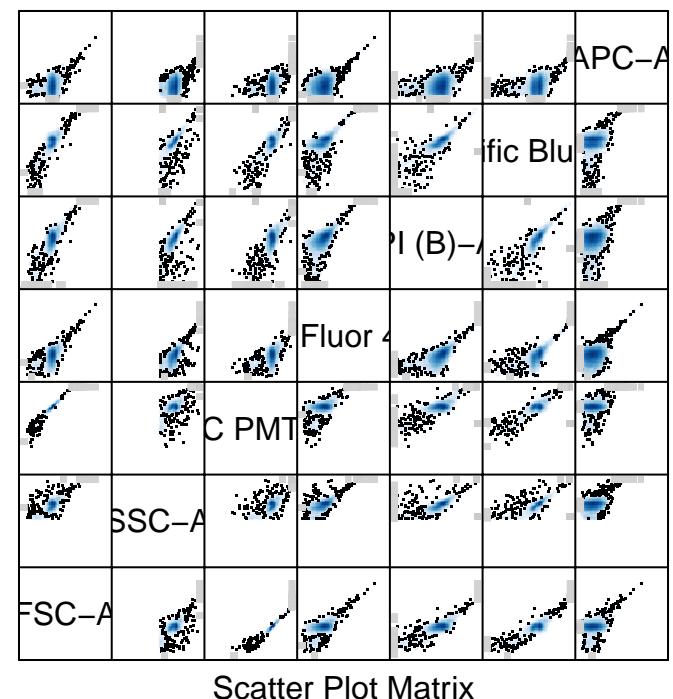
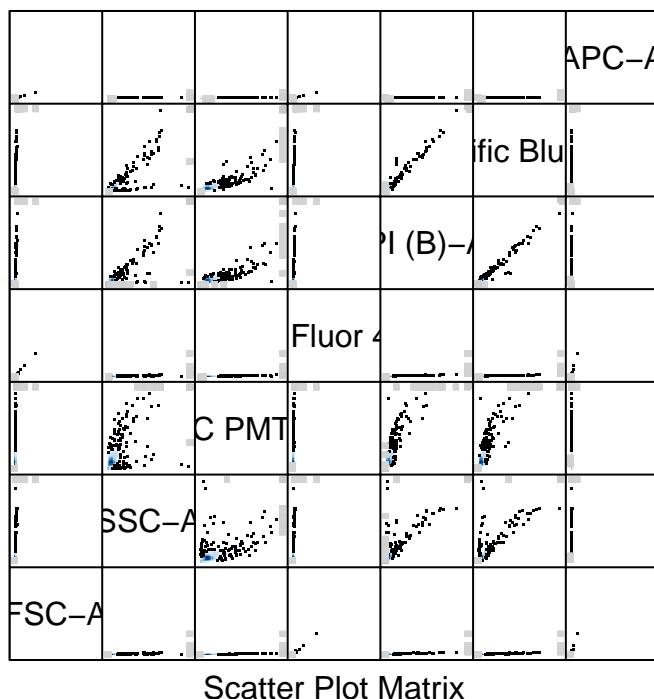


2.2.2.1 Time Anomalies `qaProcess.timeflow` & `qaProcess.timeline` can be used to detect disturbances in flow over time. Good even, flow through the machine shows that the instrument is running smoothly. These timeflow plots visualize acquisition rate over time, so the number of events that are recorded in a given time interval. The summary QA report for all the data in a `flowSet` can be found in the output.

Based on the QA analysis, this flowset is ready for to be checked for transformation. However, there may be some cases where the marginal events need to be removed and/or the data normalized.

2.2.3 Transformation

Data transformations can be done in-line or out-of-line techniques. Using the compensated data, For more information about choosing the appropriate transformation, see Spidlen et al. 2006.



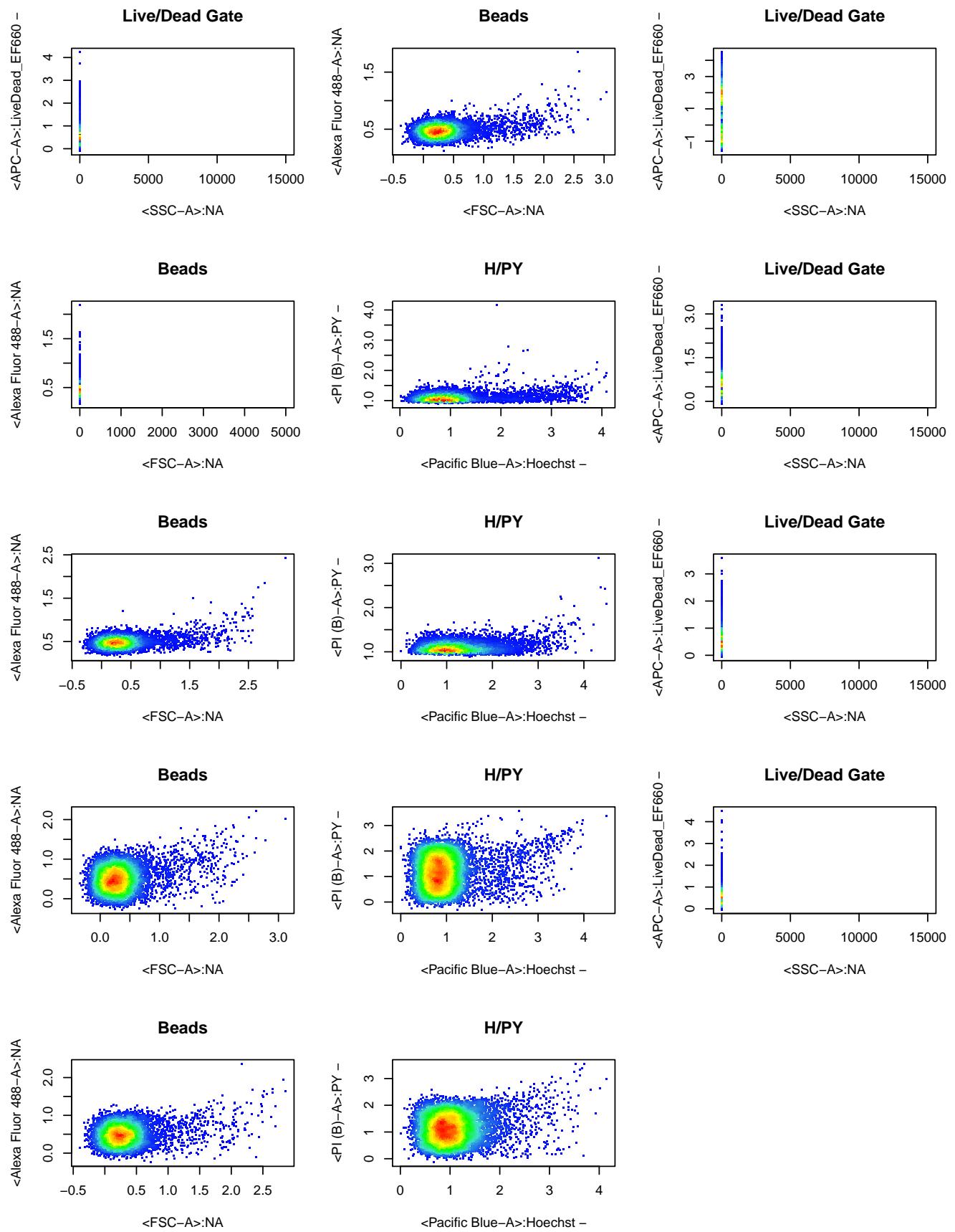
3. Define gates/filters using `flowDensity`

A flow cytometry `workFlow` object uses filters, or gates, that are related to the fluorescence of the eFluor 660 (a fixed viability dye), Hoechst 33342, or Pyronin-Y. This process uses all of the information to construct informational sorting for the data sets.

Stain	Activity	Use
Molecular Probes eFluor 660	fixed viability dye	Live vs. dead
Hoechst 33342	DNA specific binding	Active vs. dormant
Pyronin Y	RNA specific binding	Active vs. dormant

For each of the single color controls, we use static gating to create the initial gate sets. After constructing filters, we can then count populations of events.

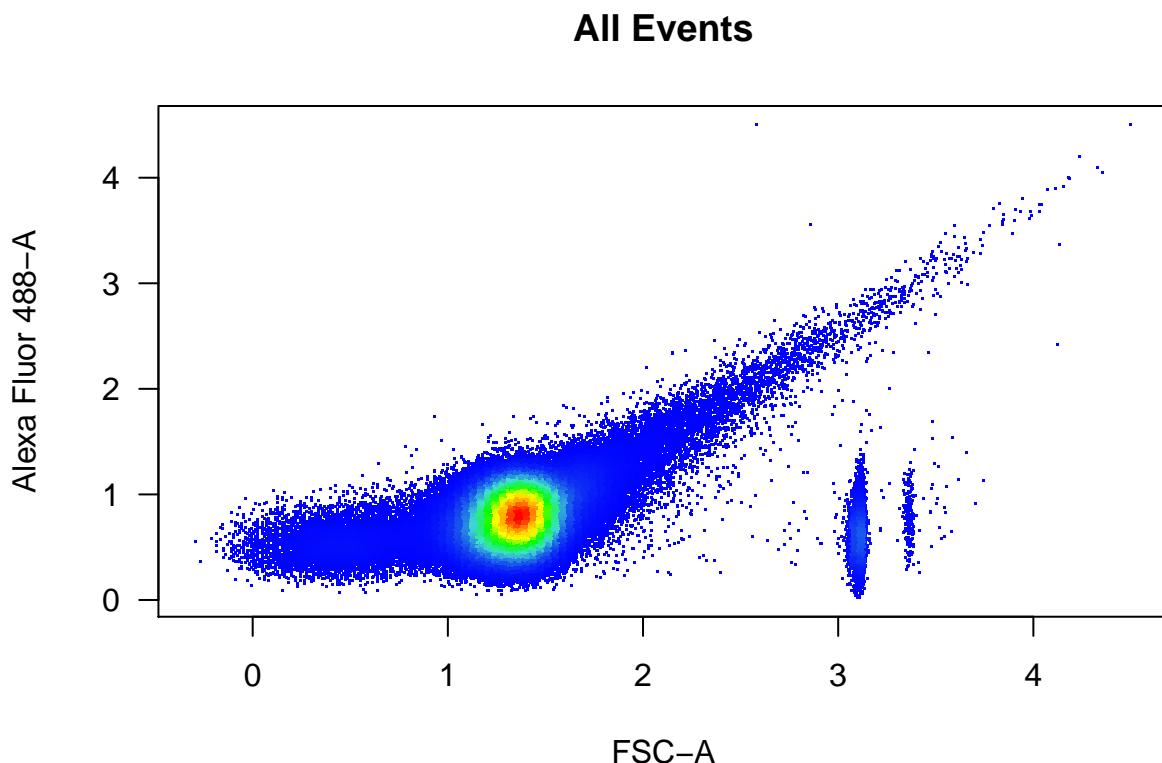
Let's look at the data in their respective channels.



3.1 Create static gates

Creating a gate for the beads produces a standardized count for estimating population density (events/mL). Beads are most easily seen in the FSC-A v. Alexa Fluor 488. Use of the Live/Dead fixed viability dye requires that we use a negative gating process. Cells with compromised membranes will take up the stain while those with intact membranes will not.

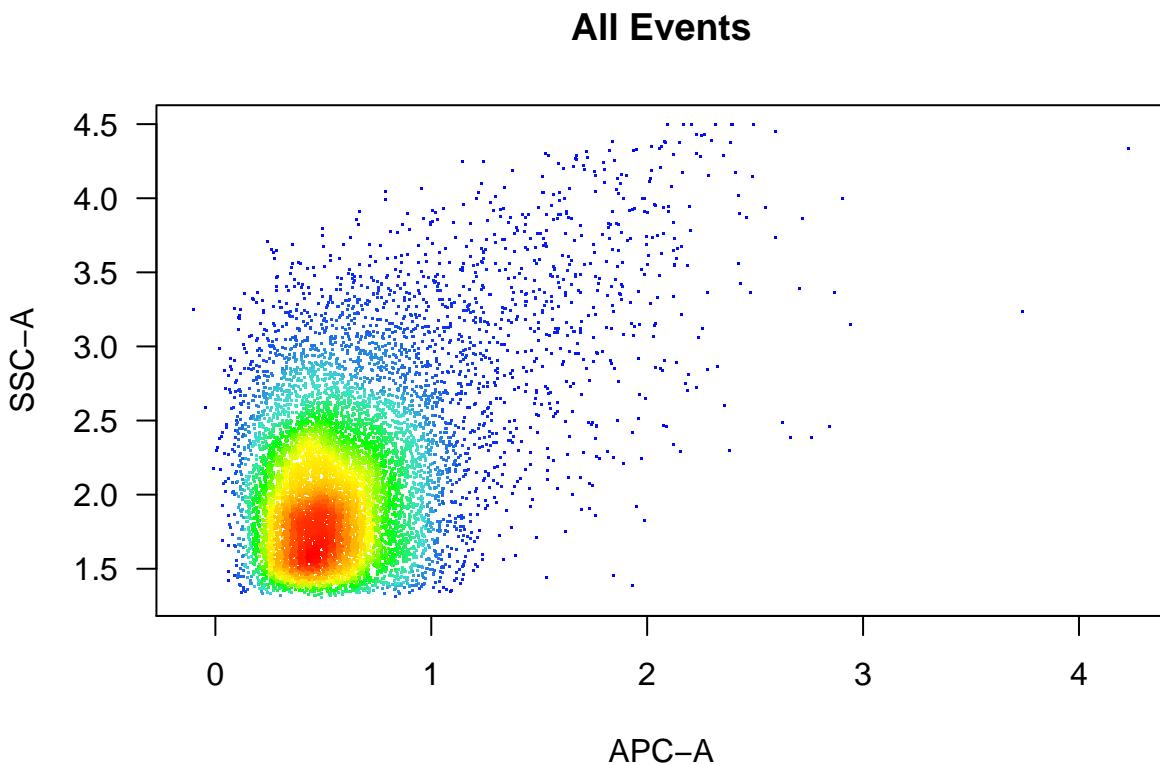
```
# Beads
samp.beads <- fs.trans[[6]]
bead.chnl <- c("FSC-A", "Alexa Fluor 488-A")
bead.cols <- c(1,4)
beads.plot <- plotDens(samp.beads, bead.chnl, devn = FALSE,
                        xlab = "FSC-A", ylab = "Alexa Fluor 488-A", las = 1)
```



```
bead.gate <- rectangleGate(filterId = "beads",
                            "FSC-A" = c(3.0,3.2),
                            "Alexa Fluor 488-A" = c(0,1.25))

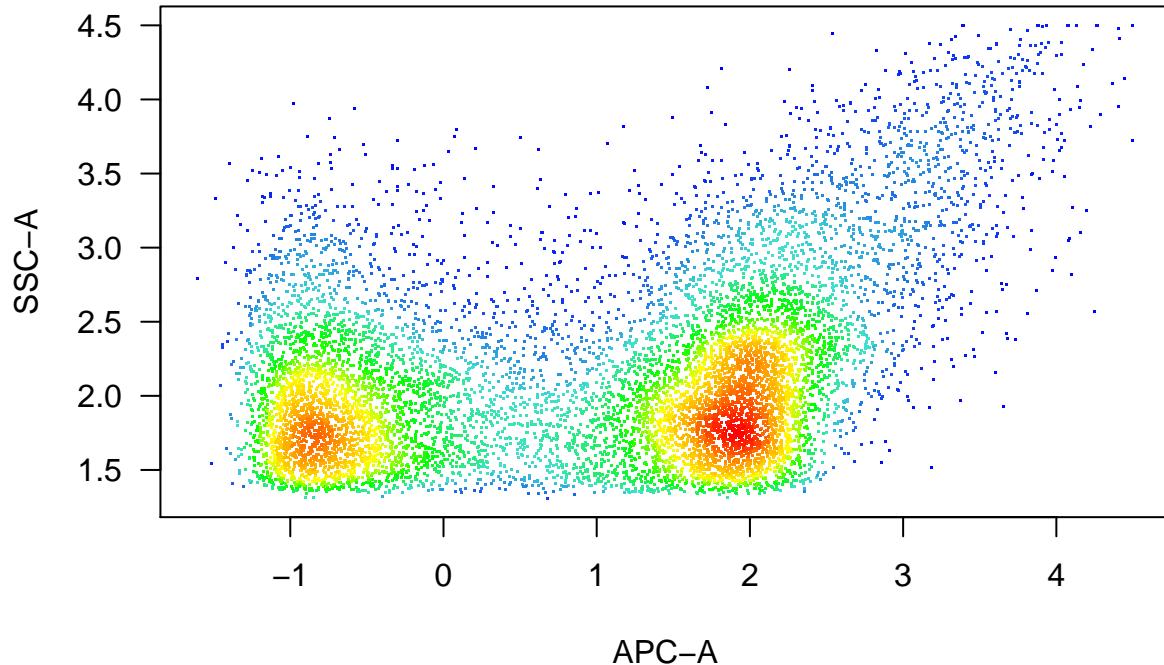
# Live/Dead gating from eFluor 660 - based on negative staining
## Must use unstained control and check with stained control
control.ld <- fs.trans[[5]]
test.ld <- fs.trans[[3]]
ld.chnl <- c("APC-A", "SSC-A")
ld.cols <- c(2,7)
```

```
plot.new()  
  
plotDens(control.ld, ld.chnl, devn = FALSE,  
          xlab = "APC-A", ylab = "SSC-A", las = 1)
```



```
plotDens(test.ld, ld.chnl, devn = FALSE,  
          xlab = "APC-A", ylab = "SSC-A", las = 1)
```

All Events



```
live.gate <- rectangleGate(filterId = "live",
                            "APC-A" = c(0,max(ld.cols[1])),
                            "SSC-A" = c(1.15,4.5))

dead.gate <- rectangleGate(filterId = "dead",
                            "APC-A" = c(max(ld.cols[[1]]),Inf))

ldfilt <- kmeansFilter(filterID = "ldfilt", "APC-A" = c("live","dead"))

# Test output
bead.results <- filter(fs.trans, bead.gate)
live.results <- filter(fs.trans, live.gate)
dead.results <- filter(fs.trans, dead.gate)

#live.dead.results <- filter(fs.trans, ld.filt)

summary(bead.results)

## filter summary for frame '061815_002_Hst.fcs'
## beads+: 0 of 10000 events (0.00%)
##
## filter summary for frame '061815_002_Hst_PY.fcs'
## beads+: 0 of 10000 events (0.00%)
##
## filter summary for frame '061815_002_LD.fcs'
```

```
##  beads+: 0 of 10000 events (0.00%)
##
## filter summary for frame '061815_002_PY.fcs'
##  beads+: 0 of 10000 events (0.00%)
##
## filter summary for frame '061815_002_UN.fcs'
##  beads+: 1 of 10000 events (0.01%)
##
## filter summary for frame '061815_Ecoli_beads.fcs'
##  beads+: 9548 of 471916 events (2.02%)
##
## filter summary for frame '061815_Ecoli_beads_PY_Hst.fcs'
##  beads+: 0 of 10000 events (0.00%)
```

```
summary(live.results)
```

```
## filter summary for frame '061815_002_Hst.fcs'
##  live+: 9951 of 10000 events (99.51%)
##
## filter summary for frame '061815_002_Hst_PY.fcs'
##  live+: 9964 of 10000 events (99.64%)
##
## filter summary for frame '061815_002_LD.fcs'
##  live+: 3791 of 10000 events (37.91%)
##
## filter summary for frame '061815_002_PY.fcs'
##  live+: 9892 of 10000 events (98.92%)
##
## filter summary for frame '061815_002_UN.fcs'
##  live+: 9894 of 10000 events (98.94%)
##
## filter summary for frame '061815_Ecoli_beads.fcs'
##  live+: 460947 of 471916 events (97.68%)
##
## filter summary for frame '061815_Ecoli_beads_PY_Hst.fcs'
##  live+: 9969 of 10000 events (99.69%)
```

```
summary(dead.results)
```

```
## filter summary for frame '061815_002_Hst.fcs'
##  dead+: 27 of 10000 events (0.27%)
##
## filter summary for frame '061815_002_Hst_PY.fcs'
##  dead+: 22 of 10000 events (0.22%)
##
## filter summary for frame '061815_002_LD.fcs'
##  dead+: 2839 of 10000 events (28.39%)
##
## filter summary for frame '061815_002_PY.fcs'
```

```

##  dead+: 98 of 10000 events (0.98%)
##
## filter summary for frame '061815_002_UN.fcs'
##  dead+: 103 of 10000 events (1.03%)
##
## filter summary for frame '061815_Ecoli_beads.fcs'
##  dead+: 6082 of 471916 events (1.29%)
##
## filter summary for frame '061815_Ecoli_beads_PY_Hst.fcs'
##  dead+: 16 of 10000 events (0.16%)

# Create table of results for each sample
## the idea here is to pull out the character values from the output and stash them
#strsplit(output, " ")

```

```

actdorm <- Subset(fs.trans, live.gate)
actdorm

```

```

## A flowSet with 7 experiments.
##
## column names:
## FSC-A SSC-A FSC PMT-A Alexa Fluor 488-A PI (B)-A Pacific Blue-A APC-A

```

```

sampleNames(actdorm)

```

```

## [1] "061815_002_Hst.fcs"          "061815_002_Hst_PY.fcs"
## [3] "061815_002_LD.fcs"           "061815_002_PY.fcs"
## [5] "061815_002_UN.fcs"           "061815_Ecoli_beads.fcs"
## [7] "061815_Ecoli_beads_PY_Hst.fcs"

```

```

summary(actdorm)

```

```

## $`061815_002_Hst.fcs`
## FSC-A SSC-A FSC PMT-A Alexa Fluor 488-A PI (B)-A Pacific Blue-A APC-A
## Min.   -0.3696 1.303   -1.401    0.1508  0.8899   -0.004141
## 1st Qu.  0.1277 1.703    1.640    0.4107  1.0190    0.832900
## Median   0.2666 2.010    1.840    0.4715  1.0670    1.086000
## Mean     0.3208 2.123    1.899    0.4761  1.0900    1.192000
## 3rd Qu.  0.4221 2.410    2.067    0.5383  1.1370    1.421000
## Max.    2.5730 4.499    4.500    1.3950  2.2400    4.159000
##             APC-A
## Min.   0.0002741
## 1st Qu. 0.3659000
## Median  0.5395000
## Mean    0.5739000
## 3rd Qu. 0.7343000
## Max.   1.9600000
## 

```

```

## $`061815_002_Hst_PY.fcs`
##          FSC-A SSC-A FSC PMT-A Alexa Fluor 488-A PI (B)-A Pacific Blue-A
## Min.    -0.3956 1.303   -1.448      -0.2452  -0.137     0.008606
## 1st Qu.  0.1162 1.691    1.628      0.3088  0.768     0.749400
## Median   0.2551 2.008    1.828      0.4715  1.129     0.974400
## Mean     0.2945 2.105    1.868      0.4820  1.141     1.042000
## 3rd Qu.  0.4008 2.396    2.049      0.6397  1.509     1.229000
## Max.     2.7000 4.464    4.500      1.5550  3.254     3.933000
##          APC-A
## Min.    0.003101
## 1st Qu. 0.373700
## Median  0.535500
## Mean    0.570000
## 3rd Qu. 0.726700
## Max.    2.000000
##
## $`061815_002_LD.fcs`
##          FSC-A SSC-A FSC PMT-A Alexa Fluor 488-A PI (B)-A Pacific Blue-A
## Min.    -0.3696 1.309   -1.312      0.1778  0.9159    0.03802
## 1st Qu.  0.1048 1.665    1.578      0.4046  1.0180    0.63690
## Median   0.2320 1.928    1.757      0.4654  1.0620    0.83290
## Mean     0.2505 2.015    1.769      0.4649  1.0830    0.83980
## 3rd Qu.  0.3682 2.271    1.954      0.5262  1.1300    1.02800
## Max.     2.2340 4.211    4.215      0.8573  1.5560    3.08900
##          APC-A
## Min.    0.0002741
## 1st Qu. 0.9864000
## Median  1.5130000
## Mean    1.3390000
## 3rd Qu. 1.7960000
## Max.    1.9990000
##
## $`061815_002_PY.fcs`
##          FSC-A SSC-A FSC PMT-A Alexa Fluor 488-A PI (B)-A Pacific Blue-A
## Min.    -0.3422 1.312   -1.298      -0.2728  -0.2953    0.00171
## 1st Qu.  0.1277 1.663    1.643      0.2853  0.6820    0.65730
## Median   0.2666 1.968    1.848      0.4776  1.1590    0.85760
## Mean     0.3054 2.083    1.885      0.4881  1.1630    0.93030
## 3rd Qu.  0.4221 2.361    2.079      0.6860  1.6320    1.07900
## Max.     2.5950 4.496    4.500      1.5750  3.2450    3.79500
##          APC-A
## Min.    0.003101
## 1st Qu. 0.377600
## Median  0.551400
## Mean    0.608600
## 3rd Qu. 0.767800
## Max.    1.993000
##
## $`061815_002_UN.fcs`
##          FSC-A SSC-A FSC PMT-A Alexa Fluor 488-A PI (B)-A Pacific Blue-A

```

```

## Min. -0.3762 1.309 -1.496 0.1192 0.8883 0.03491
## 1st Qu. 0.1277 1.688 1.602 0.4046 1.0150 0.66290
## Median 0.2551 1.995 1.793 0.4654 1.0690 0.85760
## Mean 0.3043 2.107 1.846 0.4691 1.0900 0.92300
## 3rd Qu. 0.4008 2.391 2.019 0.5322 1.1410 1.06800
## Max. 3.0450 4.383 4.500 1.2180 1.9600 3.73800
##
## APC-A
## Min. 0.005939
## 1st Qu. 0.381400
## Median 0.539500
## Mean 0.599900
## 3rd Qu. 0.745500
## Max. 1.998000
##
## $`061815_Ecoli_beads.fcs`
## FSC-A SSC-A FSC PMT-A Alexa Fluor 488-A PI (B)-A Pacific Blue-A
## Min. -0.2911 2.350 -2.042 0.04907 0.7035 -0.1257
## 1st Qu. 1.2700 2.996 3.323 0.64550 1.1470 1.3170
## Median 1.3580 3.146 3.400 0.79060 1.2930 1.5690
## Mean 1.3490 3.116 3.394 0.79130 1.3150 1.5250
## 3rd Qu. 1.4400 3.236 3.479 0.92410 1.4580 1.7660
## Max. 3.1030 4.497 4.500 2.48800 3.1800 4.3350
##
## APC-A
## Min. 0.0002741
## 1st Qu. 0.4245000
## Median 0.6225000
## Mean 0.6383000
## 3rd Qu. 0.8362000
## Max. 2.0000000
##
## $`061815_Ecoli_beads_PY_Hst.fcs`
## FSC-A SSC-A FSC PMT-A Alexa Fluor 488-A PI (B)-A Pacific Blue-A
## Min. -0.2272 2.363 1.240 -0.02481 0.04991 0.2335
## 1st Qu. 1.3240 3.043 3.371 0.97300 2.81000 2.8500
## Median 1.4080 3.196 3.447 1.17000 2.97200 3.0170
## Mean 1.3960 3.162 3.438 1.13500 2.92600 2.9710
## 3rd Qu. 1.4870 3.278 3.523 1.32600 3.10100 3.1110
## Max. 2.9480 4.494 4.500 2.59500 4.31300 4.3260
##
## APC-A
## Min. 0.005939
## 1st Qu. 0.479800
## Median 0.700100
## Mean 0.706700
## 3rd Qu. 0.923900
## Max. 1.984000

```

```

# function to create ratio for fluorescence
test <- actdorm[[7]]

```

```

# subset data fluorescence data from flowframe

```

```

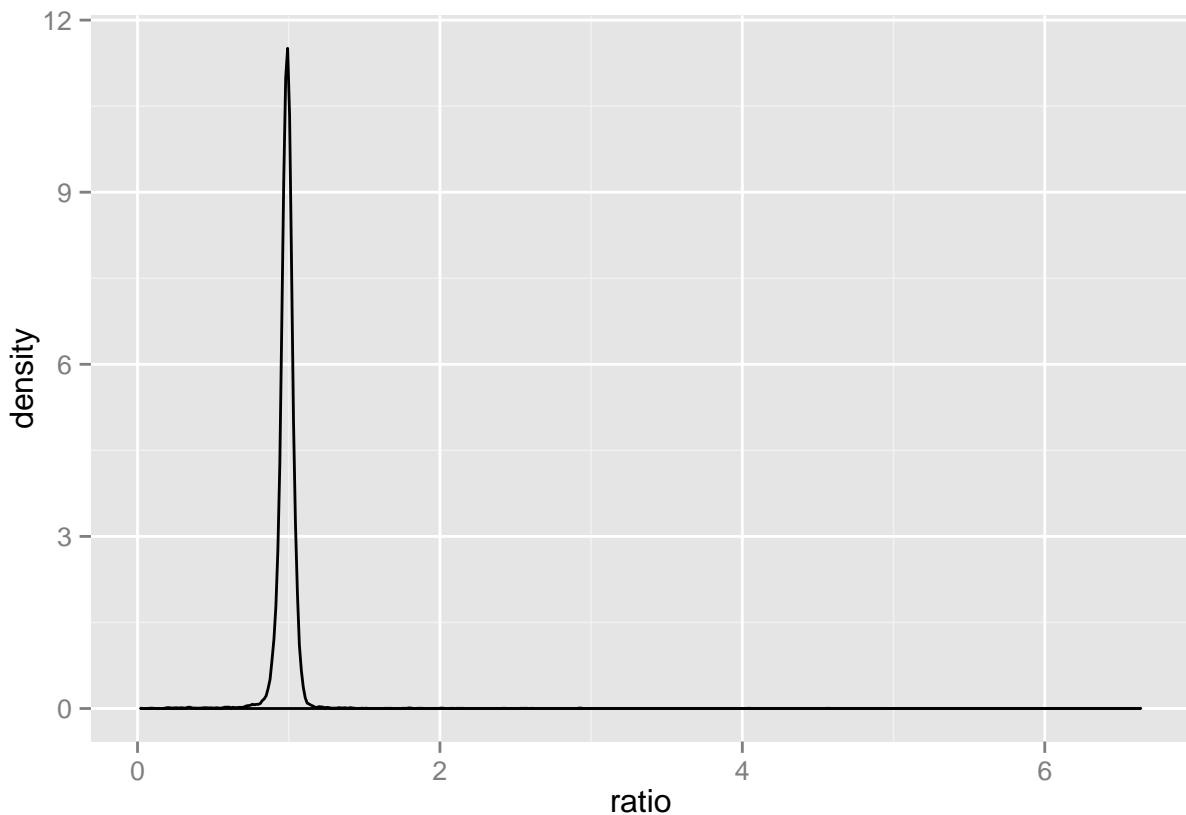
DNA <- exprs(test[, "Pacific Blue-A"])
RNA <- exprs(test[, "PI (B)-A"])

# Calculate RNA/DNA ratio
RDratio <- RNA/DNA
dat <- data.frame(DNA, RNA, RDratio)
colnames(dat) <- c("DNA", "RNA", "ratio")

# Calculate population densities and relative percentages
min <- 1-sd(RDratio); max <- 1+sd(RDratio)
live.pop.dens <- length(RDratio[RDratio > min])
act.pop.dens <- length(RDratio[RDratio > min & RDratio < max])
per.dorm <- (live.pop.dens - act.pop.dens)/live.pop.dens
per.act <- (act.pop.dens)/live.pop.dens

# Visualize distribution
ggplot(dat, aes(x = ratio)) + geom_density()

```



```
## NEED TO ADD IN POLYGON SHADING FOR REGIONS + LABELS
```

```

results <- matrix(NA, nrow = length(sampleNames(fs.trans)), ncol = 9)
results <- as.data.frame(results)
colnames(results) <- c("sample", "NA", "ratio.min", "ratio.max",
                       "live.dens", "act.dens", "dorm.dens",
                       "act.perc", "dorm.perc")

```

```

for(i in 1:length(sampleNames(fs.trans))){  

  DNA <- exprs(fs.trans[[i]][,"Pacific Blue-A"])  

  RNA <- exprs(fs.trans[[i]][,"PI (B)-A"])  

  RDratio <- RNA/DNA  

  dat <- data.frame(DNA,RNA,RDratio)  

#make plot  

  p <- ggplot(dat,aes(x = dat[,3])) + geom_density()  

  print(p)  

  range(RDratio)  

  min <- 1-sd(RDratio)  

  max <- 1+sd(RDratio)  

  live.pop.dens <- length(RDratio[RDratio > min])  

  dorm.pop.dens <- length(RDratio[RDratio > min & RDratio < max])  

  act.pop.dens <- live.pop.dens-dorm.pop.dens  

  per.act <- (live.pop.dens - dorm.pop.dens)/live.pop.dens  

  per.dorm <- (act.pop.dens)/live.pop.dens  

  results[i,1] <- sampleNames(fs.trans)[[i]]  

  results[i,2] <- NA  

  results[i,3] <- min  

  results[i,4] <- max  

  results[i,5] <- live.pop.dens  

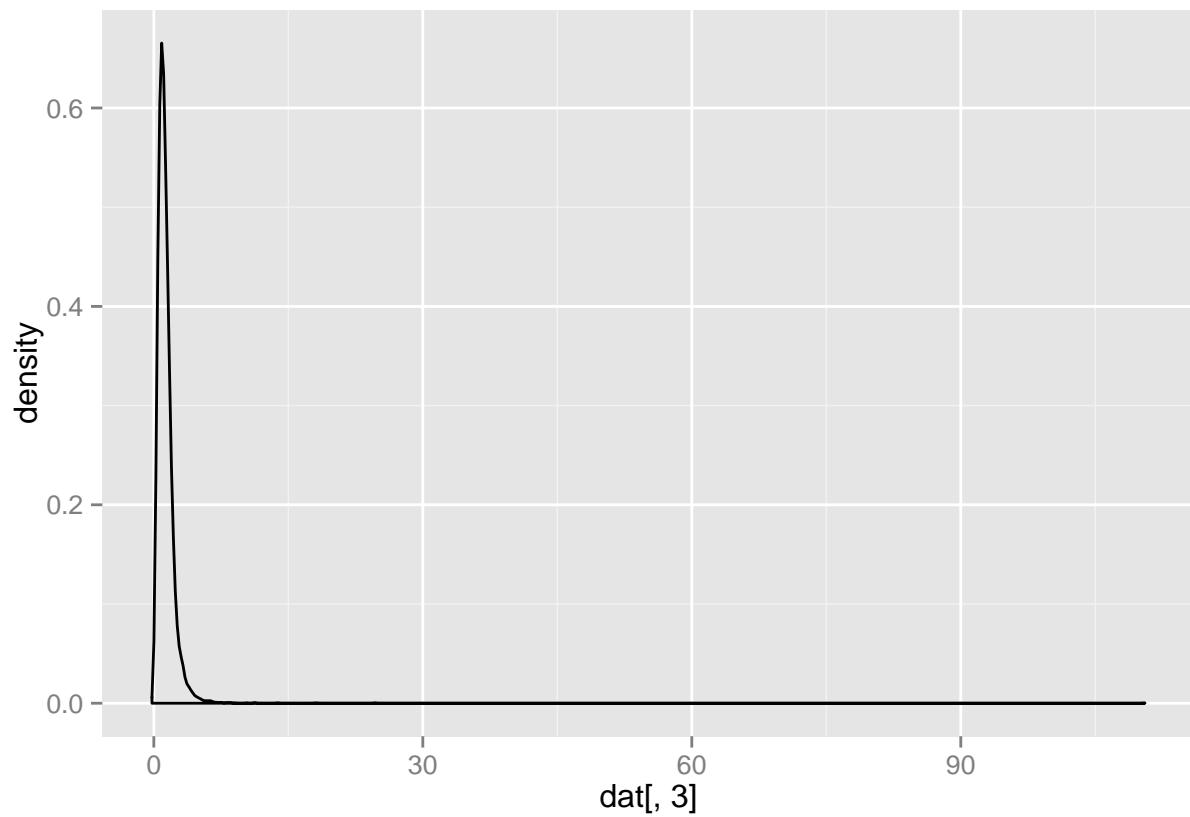
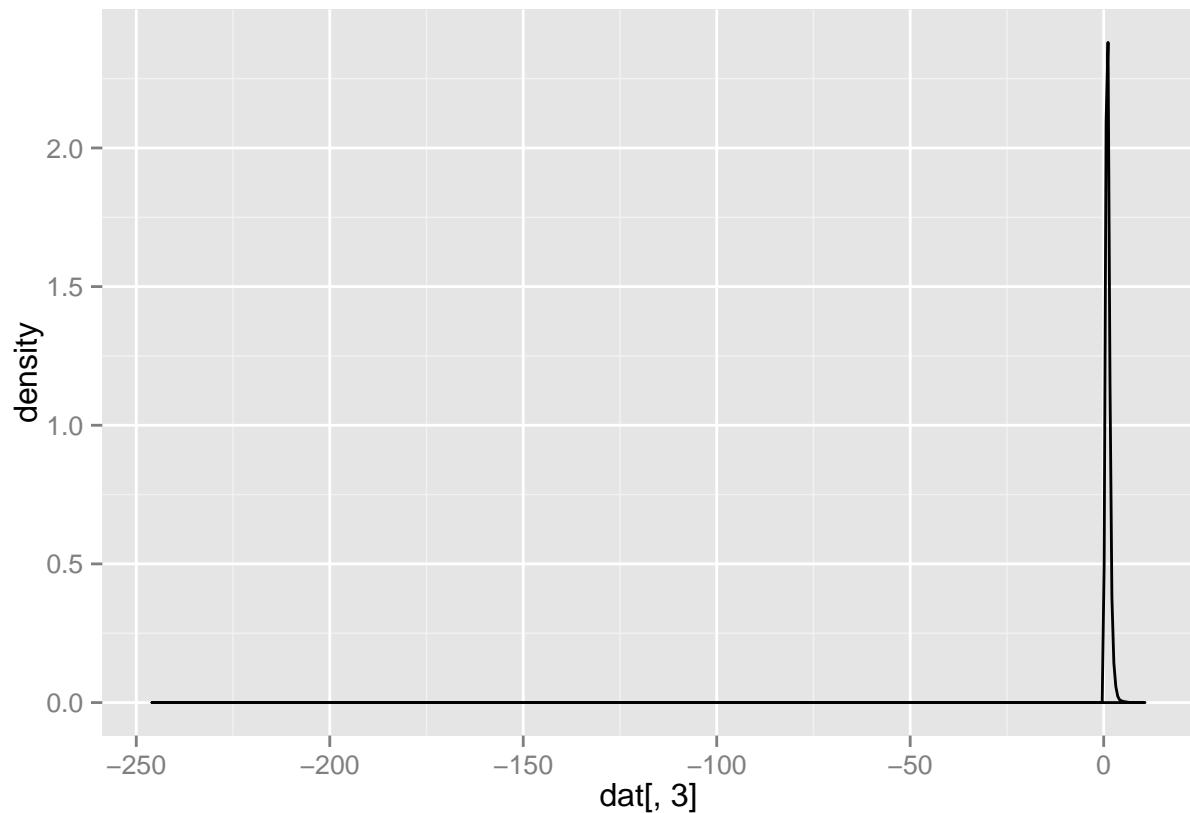
  results[i,6] <- act.pop.dens  

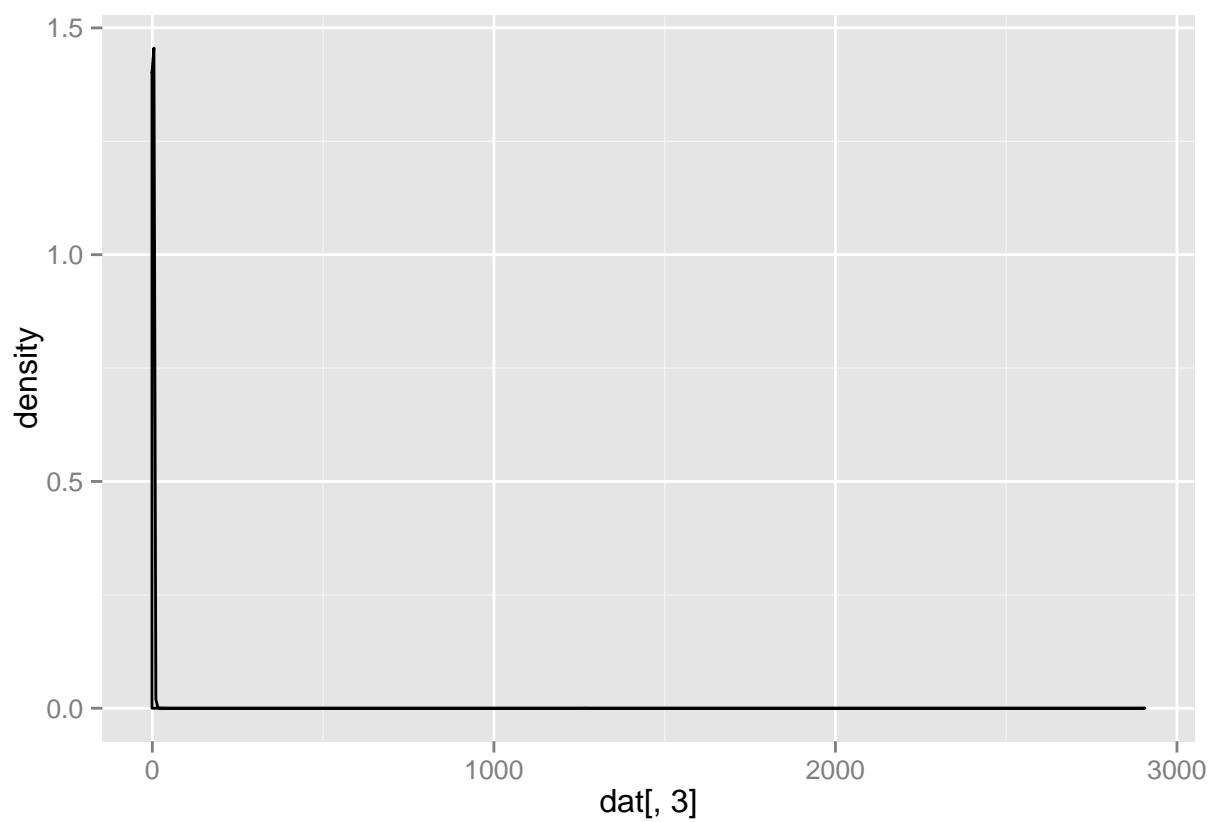
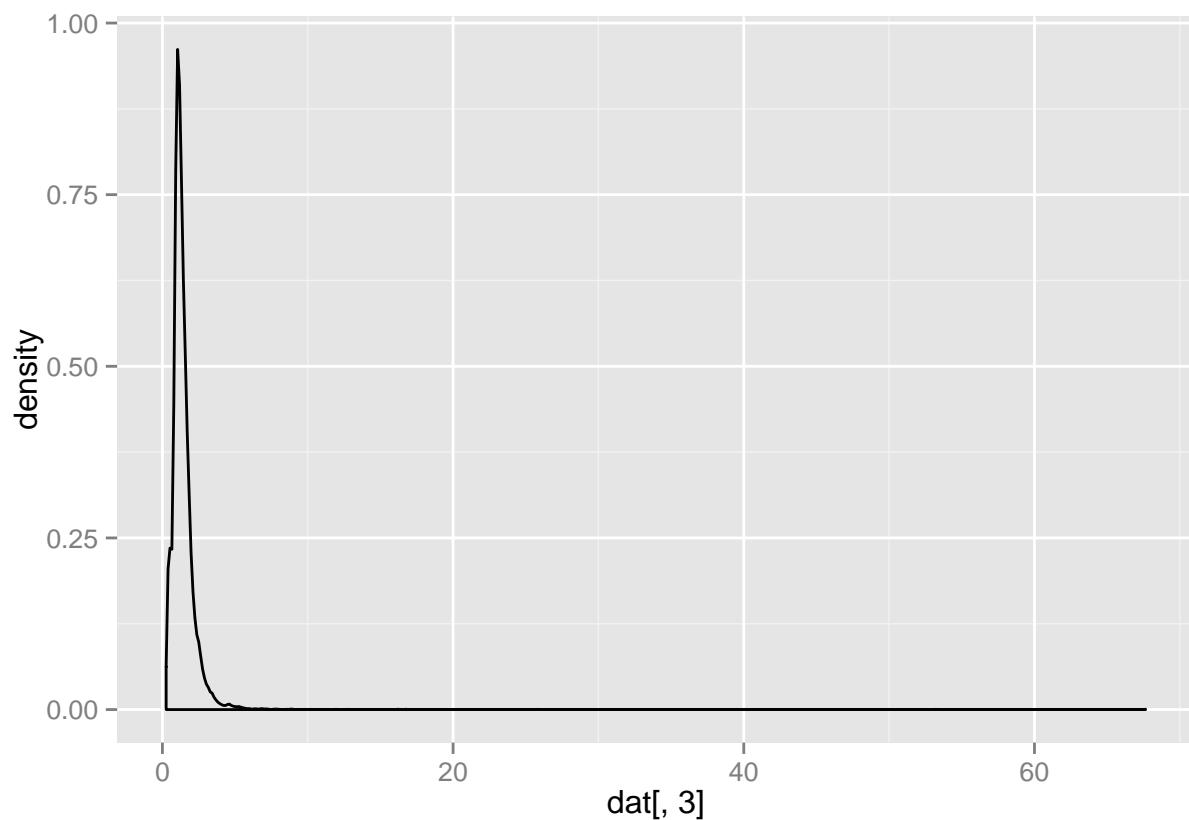
  results[i,7] <- dorm.pop.dens  

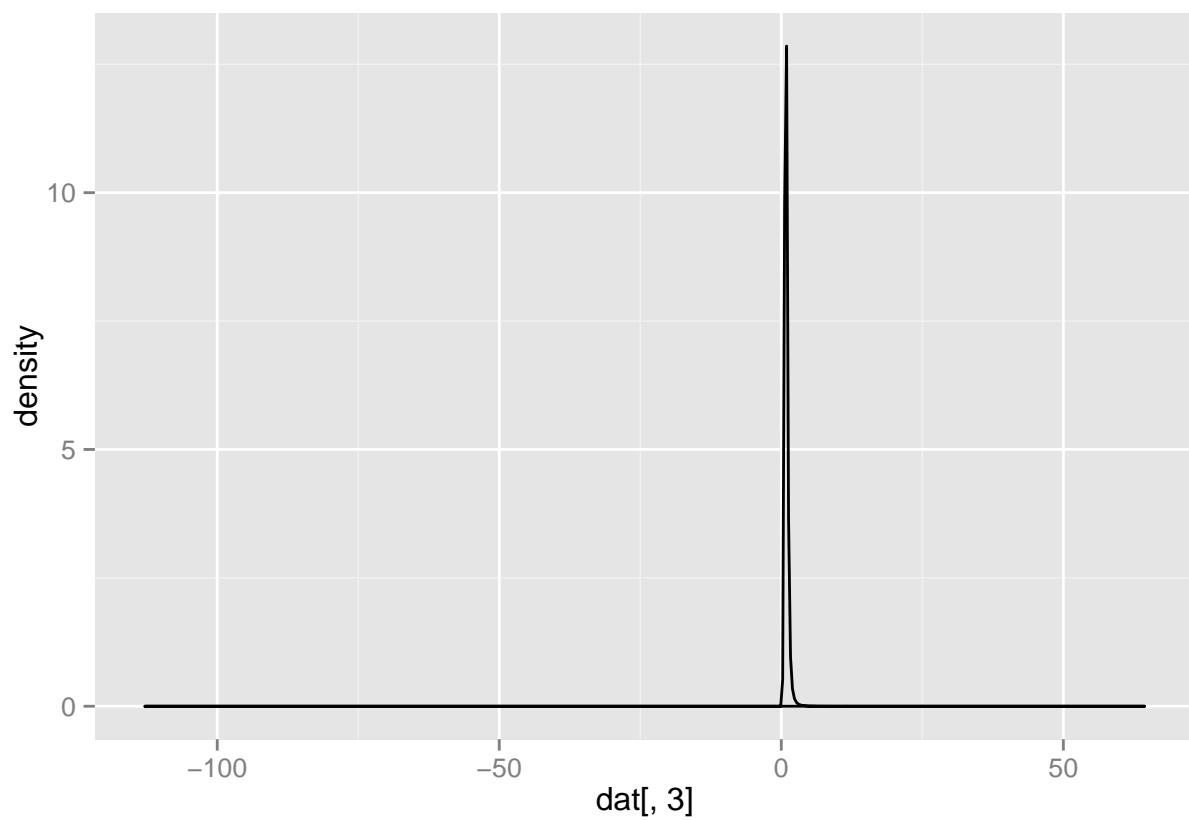
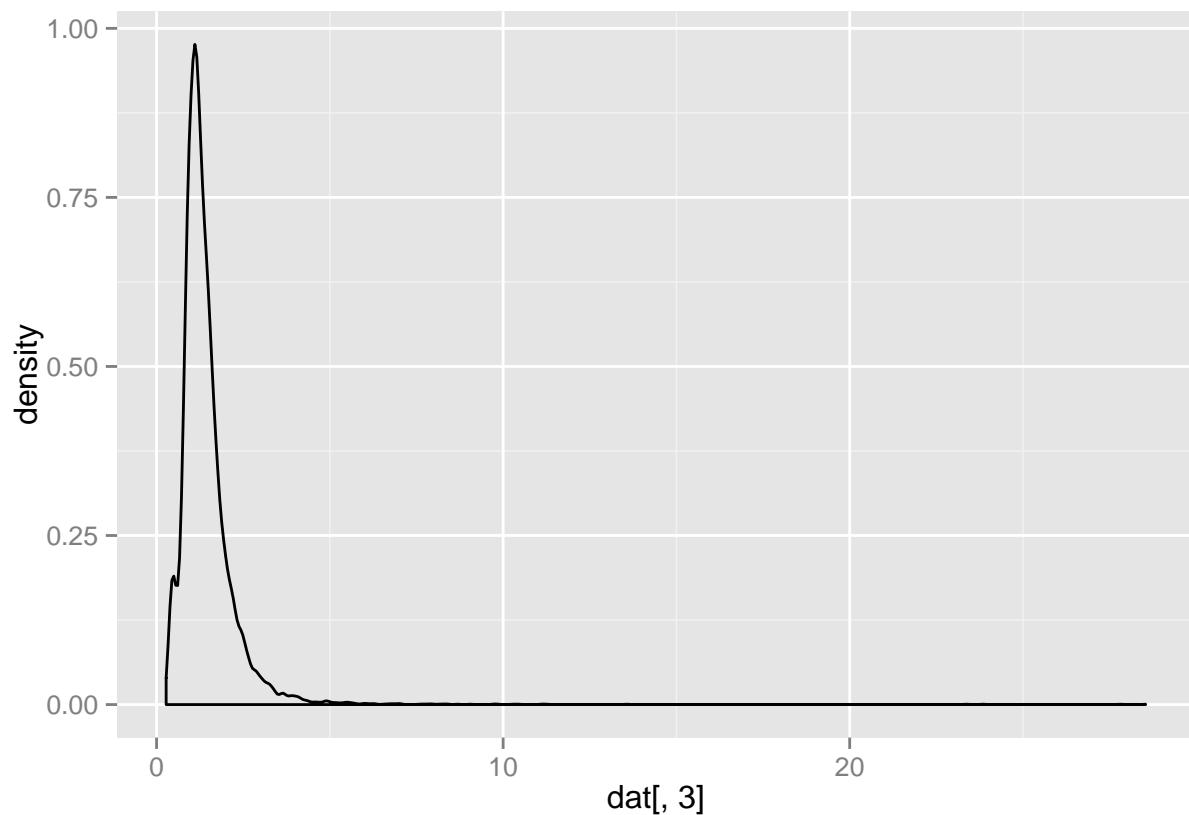
  results[i,8] <- round(per.act*100, digits = 3)  

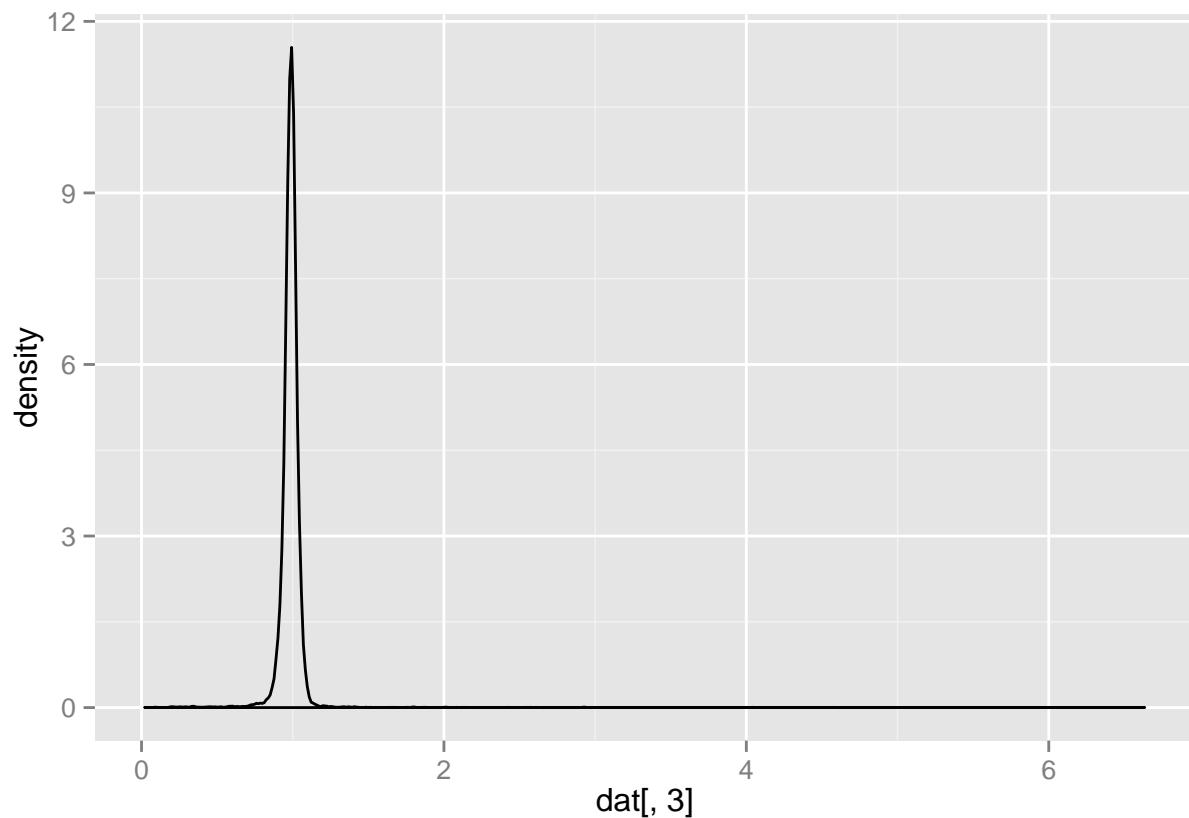
  results[i,9] <- round(per.dorm*100, digits = 3)
}

```









```
results
```

```
##           sample NA ratio.min ratio.max live.dens
## 1 061815_002_Hst.fcs NA -1.5366470 3.536647    9999
## 2 061815_002_Hst_PY.fcs NA -0.4273458 2.427346   10000
## 3 061815_002_LD.fcs NA -0.1694238 2.169424   10000
## 4 061815_002_PY.fcs NA -30.7054645 32.705464   10000
## 5 061815_002_UN.fcs NA  0.1001156 1.899884   10000
## 6 061815_Ecoli_beads.fcs NA  0.5423753 1.457625  454678
## 7 061815_Ecoli_beads_PY_Hst.fcs NA  0.8868391 1.113161    9682
##   act.dens dorm.dens act.perc dorm.perc
## 1      54     9945    0.540     0.540
## 2     786     9214    7.860     7.860
## 3    1023     8977   10.230    10.230
## 4      3     9997    0.030     0.030
## 5    1620     8380   16.200    16.200
## 6   25058    429620    5.511     5.511
## 7      93     9589    0.961     0.961
```

4. Batch sample processing

Active populations are cells that have RNA > DNA **Dormant** population have RNA = DNA **Dead** population is filtered out with the fixed viability dye

Create data file that contains information about the samples as well as calculations for each population following gates

4.1 Test output generation with a single batch (bch06182015-1)

4.2 Multibatch processing

Script to be added.