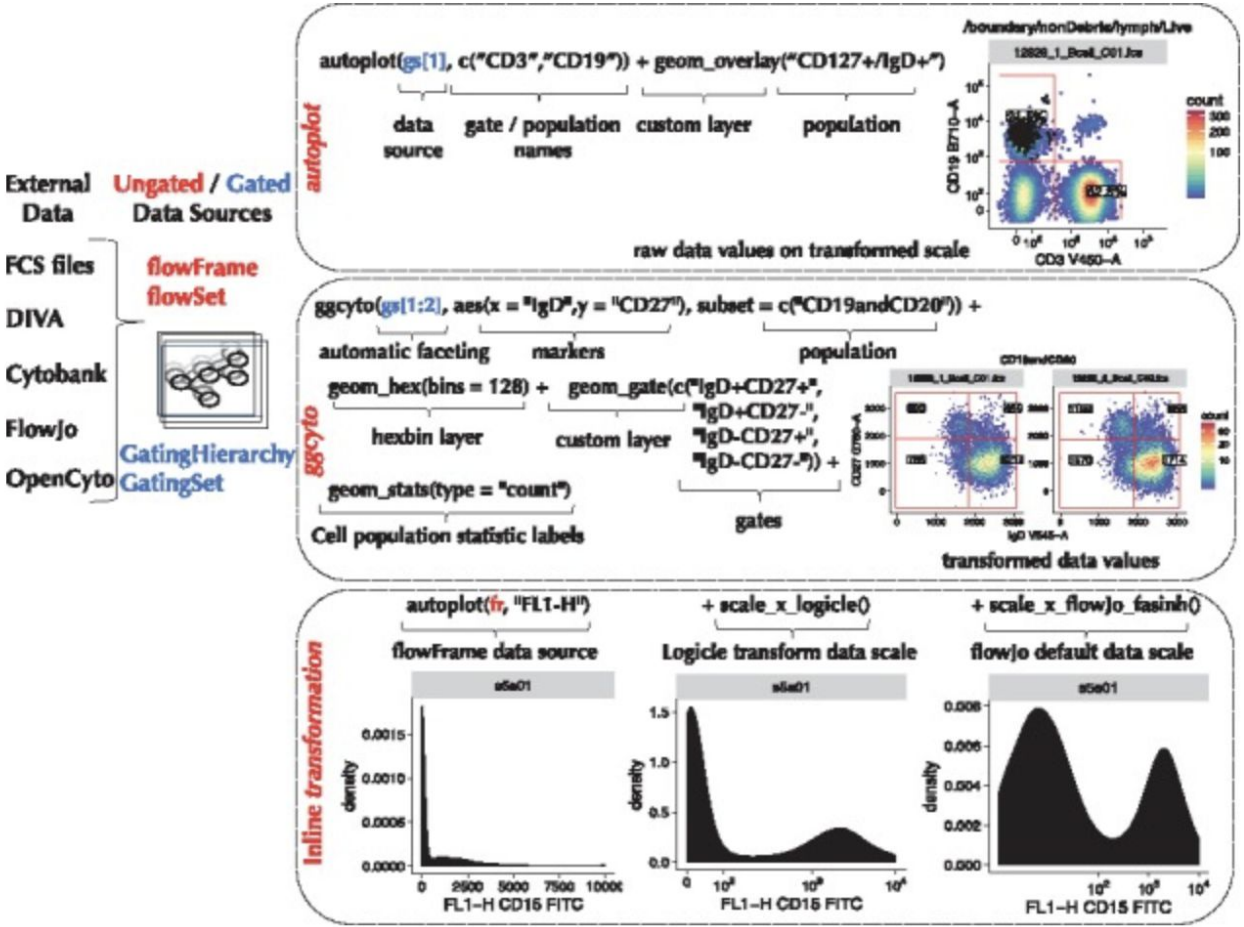
RaMP Spring 2021 Update

R and Flow Cytometry
Elizabeth Roebker

Background

- Cytometry (FCM) is the primary assay for immune monitoring in clinical and research applications
- Ggcyto is a BioConductor package that provides a unified visualization interface to complex, ungated, gated, and/or annotated cytometry data structures
  - Built on the ggplot2 package
  - Ggcyto also allows for data and axes transformation, back-gating visualization, plot faceting by experimental meta-data variables
- Analogously, autoplot can be used in order to create plots from flowSet and flowFrame objects (for ungated data) or GatingHierarchy and GatingSet objects (for gated data)


External Data
Ungated / Gated Data Sources
FCS files
DIVA
Cytobank
FlowJo
OpenCyto
flowFrame
flowSet
GatingHierarchy
GatingSet
autoplot
autoplot(gs[1], c("CD3","CD19")) + geom_overlay("CD127+/IgD+")
data source
gate / population names
custom layer
population
raw data values on transformed scale
/boundary/nonDebris/lymph/Live
CD19 BV10-A
CD3 V450-A
count
ggcyto
ggcyto(gs[1:2], aes(x = "IgD",y = "CD27"), subset = c("CD19andCD20")) +
automatic faceting
markers
population
geom_hex(bins = 128) +
geom_gate(c("IgD+CD27+", "IgD+CD27-", "IgD-CD27+", "IgD-CD27-")) +
geom_stats(type = "count")
hexbin layer
custom layer
gates
Cell population statistic labels
transformed data values
Inline transformation
autoplot(fr, "FL1-H")
+ scale_x_logicle()
+ scale_x_flowJo_fasinh()
flowFrame data source
Logicle transform data scale
flowJo default data scale
s5e01
density
FL1-H CD15 FITC

Purpose

- Open source software for computational cytometry has been gaining popularity over the past few years
    - Why? This software has highlighted the importance of trying to standardize experimental and computational aspects of cytometry data analysis
- The R/BioConductor platform hosts the largest collection of open source cytometry software covering all aspects of data analysis
        - Provides infrastructure to analyze cytometry with all of the relevant experimental and gating and cell population annotations
        - **This enables completely reproducible data analysis**
- For ungated files, you can use the automated gating template functionality of the openCyto package in order to detect sub-populations that can be applied to whole datasets
    - The automation of this process saves the researcher from having to manually gate for matched experiments
        - Saving time while improving reproducibility and objectivity

Supplementary Information

Bioinformatics. 2018 Nov 15; 34(22): 3951–3953.
Published online 2018 Jun 1. doi: 10.1093/bioinformatics/bty441
PMCID: PMC6223365
PMID: 29868771

ggCyto: next generation open-source visualization software for cytometry

Phu Van, Wenxin Jiang, Raphael Gottardo, and Greg Finak

Jonathan Wren, Associate Editor

Associated Data

▸ Supplementary Materials

Abstract

Go to:

Motivation

Open source software for computational cytometry has gained in popularity over the past few years. Efforts such as FlowCAP, the Lyoplate and Euroflow projects have highlighted the importance of efforts to standardize both experimental and computational aspects of cytometry data analysis. The R/BioConductor platform hosts the largest collection of open source cytometry software covering all aspects of data analysis and providing infrastructure to represent and analyze cytometry data with all relevant experimental, gating and cell population annotations enabling fully reproducible data analysis. Data visualization frameworks to support this infrastructure have lagged behind.

Supplementary Information for ‘ggcyto: Next generation visualization software for flow cytometry’

Phu T. Van, Mike Jiang, Raphael Gottardo & Greg Finak

May 21, 2018

Contents

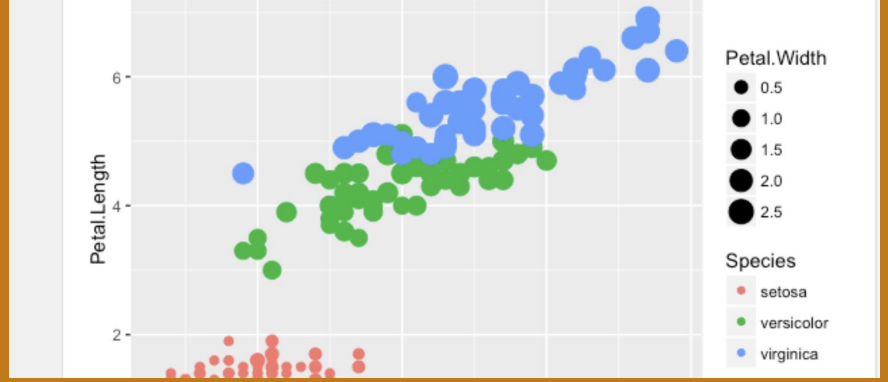
R Markdown

- Turns analyses done via R into organized presentations, reports, etc.

````
```{r}
summary(iris)
```
````

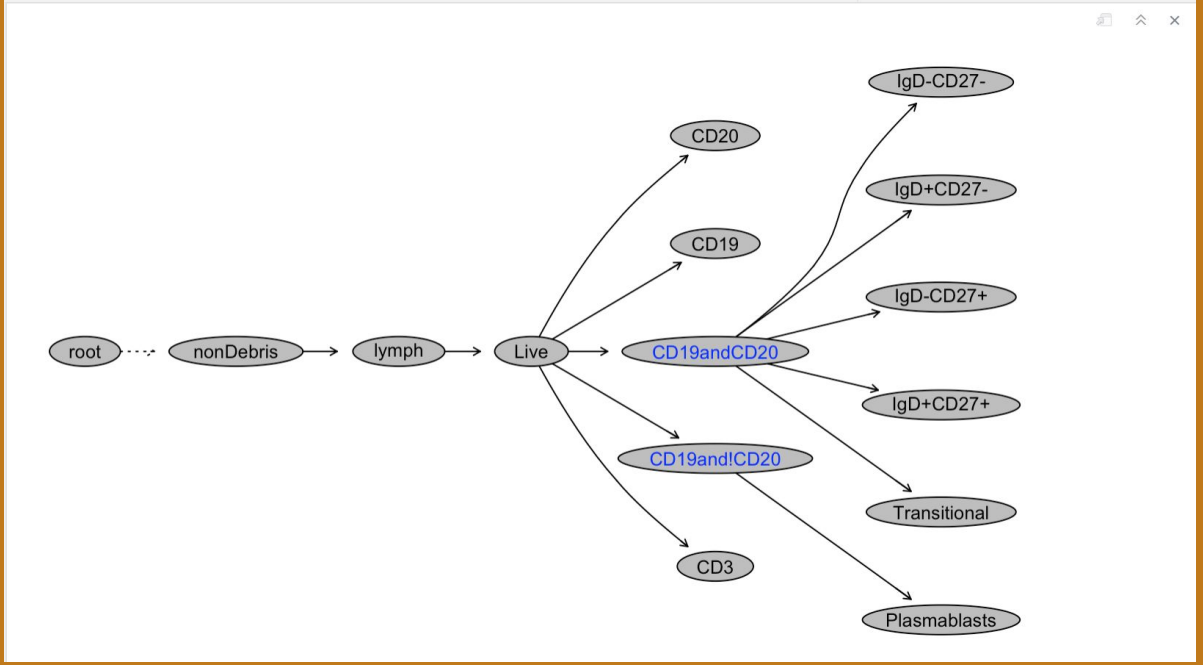
```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

````
```{r}
library(ggplot2)
qplot(Sepal.Length, Petal.Length, data = iris, color = Species, size =
Petal.Width)
```
````



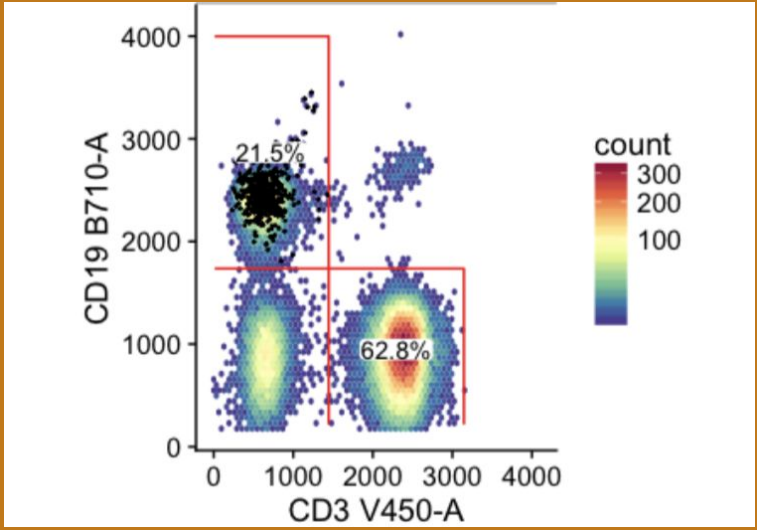
Packages

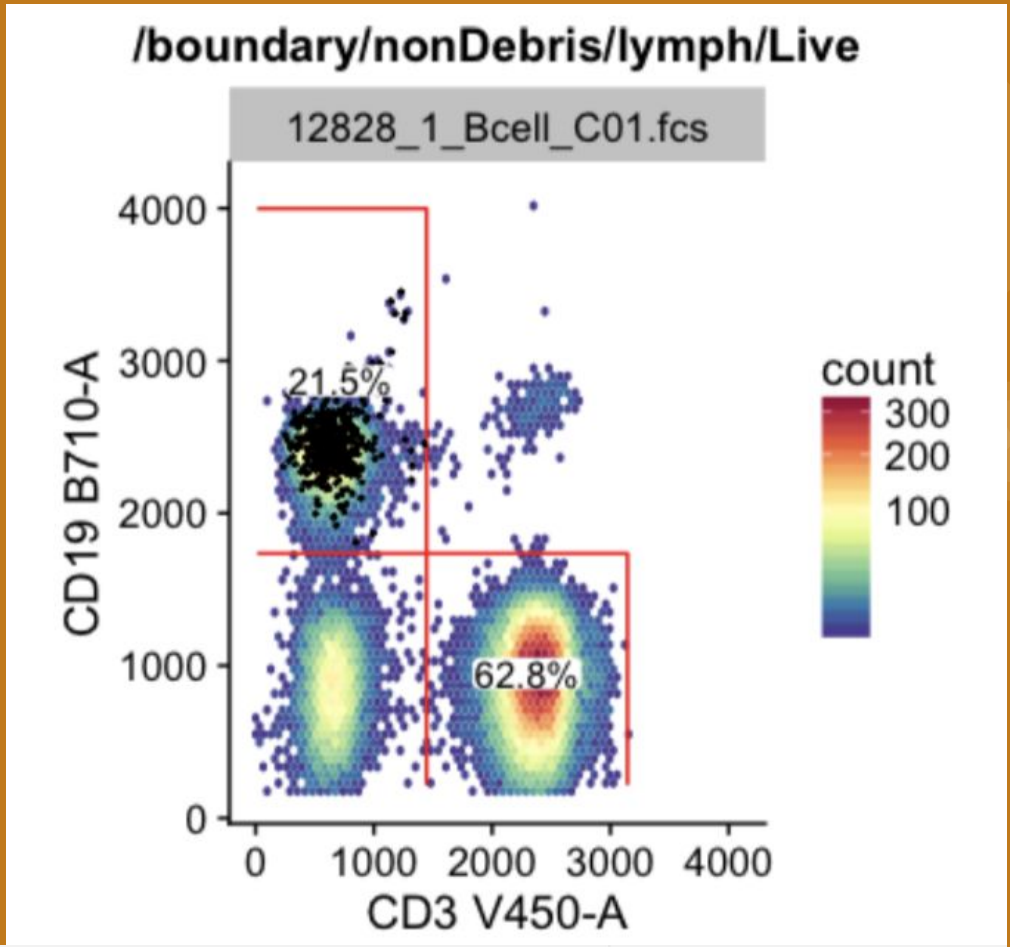
```
##Packages
#need to import all for the code to work
```{r}
install.packages("flowCore")
library("flowCore")
install.packages("mvtnorm")
library("mtvnorm")
install.packages("ggplot2")
library("ggplot2")
#install.packages("ggfortify")
#library(ggfortify)
install.packages("ncdfFlow")
library("ncdfFlow")
install.packages("RcppAramadillo")
library("RcppArmadillo")
install.packages("BH")
library("BH")
library("xtable")
install.packages("cowplot")
library("cowplot")

if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("flowCore")
BiocManager::install("ggcyto")
library(ggcyto)
BiocManager::install("flowWorkspace")
library(flowWorkspace)
BiocManager::install("flowWorkspaceData")
library(flowWorkspaceData)
browseVignettes("ggcyto")
```
```

GvHD dataset

```
## Code
```{r}
data("GvHD") #load data set
dataDir <- system.file("extdata", package="flowWorkspaceData") #system.file finds the full names of files in
packages.
gs <- load_gs(list.files(dataDir, pattern = "gs_bcell_auto",full =TRUE))
#loads a gating set, in this case with the gs_bcell_auto pattern
plot(gs, bool=TRUE)
```
```



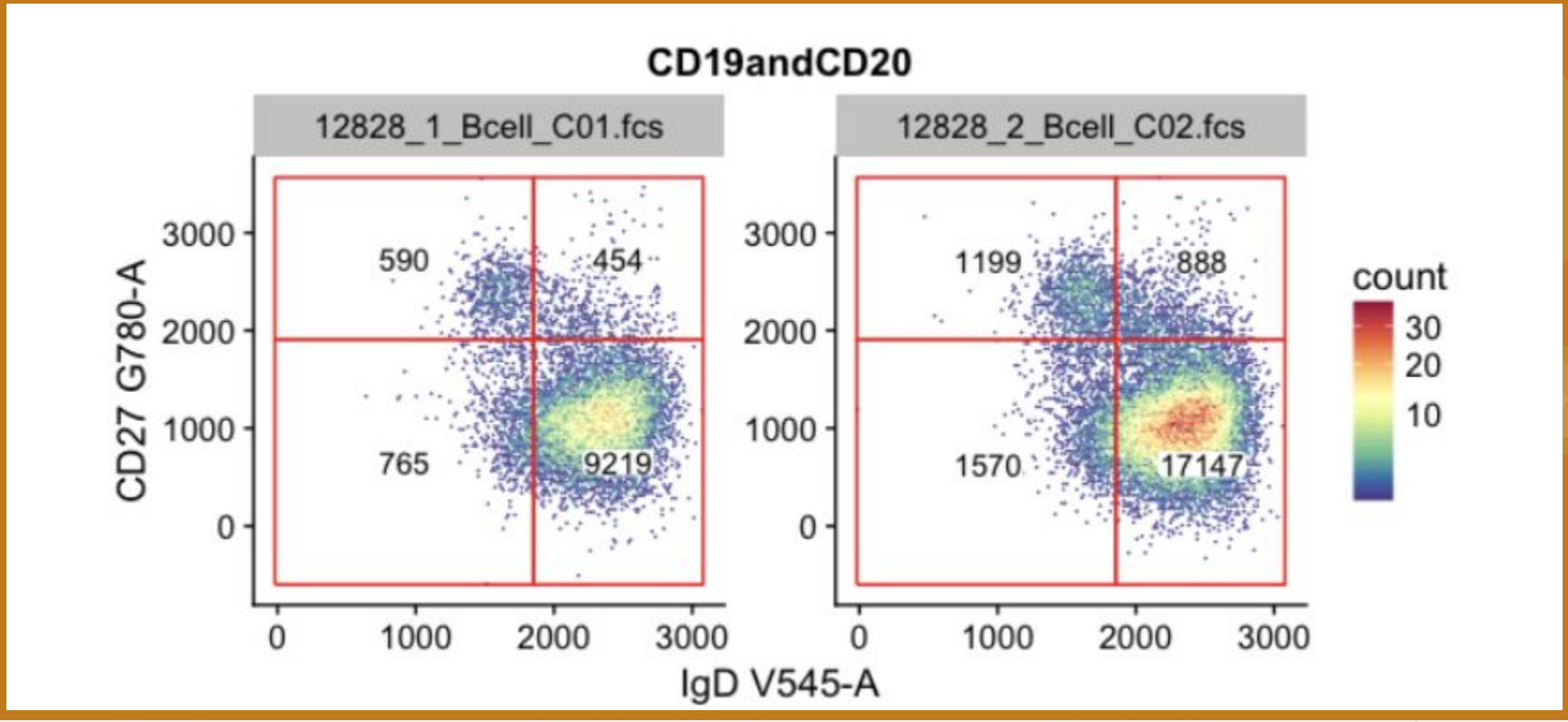
```{r}
# Below we associate a transformation (FlowJo biexponential) with the GatingSet since it did
# not have one. It is applied to all channels except FS and SSC (first two columns)
#gs@transformation <- transformerList(colnames(gs)[-(1:2)], flowjo_biexp_trans())
#use transform instead of transformerList

gs@transformation <- transform(colnames(gs)[-(1:2)], flowjo_biexp_trans())

#transformation is a virtual class to abstact transformations
#flowjo_biexp_trans() is used for constructing biexponential transformation object
#this allows R to plot a gating set

# For the GvHD data we select subjects 5 and 7, and Visit 5 and 6.
# We extract those sample "name"s and use that to subset the GvHD flowSet.
fs <- GvHD[subset(pData(GvHD), Patient %in%5:7 & Visit %in% c(5:6))
           [["name"]]]

autoplot(gs[1], gate = c("CD3","CD19"), bins = 64) + geom_overlay(data = "IgD+CD27+",size=0.25)

ggcyto(gs[1:2], mapping = aes(x="IgD",y="CD27"), subset = c("CD19andCD20")) +
  geom_hex(bins=128) +
  geom_gate(c("IgD+CD27+","IgD+CD27-","IgD-CD27+","IgD-CD27-")) +
  geom_stats(type = "count")
```

CD3 versus CD19





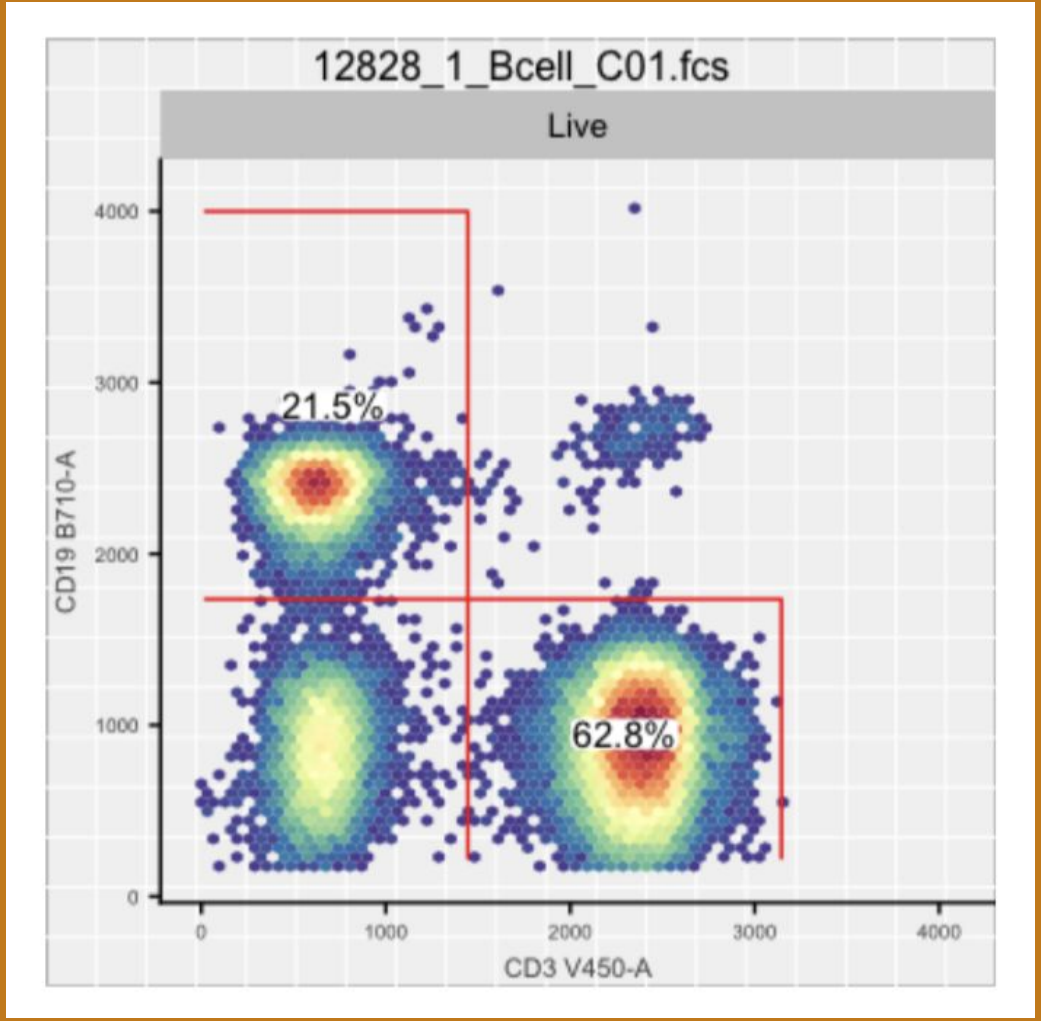

```
autoplot(gs[1], gate = c("CD3","CD19"), bins = 64) + geom_overlay(data = "IgD+CD27+",size=0.25)
```




```
ggcyto(gs[1:2], mapping = aes(x="IgD",y="CD27"), subset = c("CD19andCD20")) +
  geom_hex(bins=128) +
  geom_gate(c("IgD+CD27+","IgD+CD27-","IgD-CD27+","IgD-CD27-")) +
  geom_stats(type = "count")
```

Gated Data

```
autoplot(gs[[1]], gate = c("CD3",
"CD19"), bins = 64)
```



Next Steps

- Import Madeline's FCS files
- Replicate processes shown in this presentation
- Compare R version of plot to Madeline's FlowJo version
- FlowSOM

```
read.FCS("Compensation_Lineage -APC-Cy7.fcs")

setwd("Downloads/FW__FCS_files_test/")
FCS<-read.FCS("Compensation_CD45-AF700.fcs")
```

References

- https://www.bioconductor.org/packages/release/bioc/html/ggcyto.html
- **https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6223365/#sup1**
- **https://www.fredhutch.org/en/news/spotlight/2018/07/vidd_finak_bioinfo.html**
- **http://127.0.0.1:22265/library/ggcyto/doc/Top_features_of_ggcyto.html**
- **https://www.bioconductor.org/packages/devel/bioc/vignettes/FlowSOM/inst/doc/FlowSOM.pdf**