

```
Console Terminal x
~/
R version 3.5.3 (2019-03-11) -- "Great Truth"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

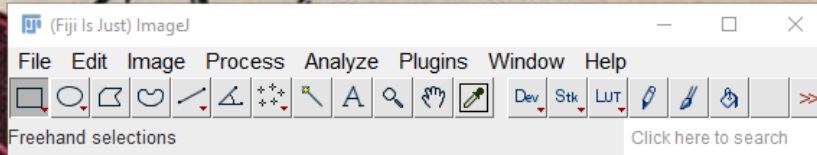
> install.packages(c("ggplot2", "autothresldr", "dplyr", "purrr", "jsonlite", "shiny"))
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.5/ggplot2_3.2.1.tgz'
Content type 'application/x-gzip' length 3961383 bytes (3.8 MB)
=====
downloaded 3.8 MB

trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.5/autothresldr_1.3.5.tgz'
Content type 'application/x-gzip' length 2155819 bytes (2.1 MB)
=====
downloaded 2.1 MB

trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.5/dplyr_0.8.3.tgz'
Content type 'application/x-gzip' length 6265040 bytes (6.0 MB)
=====
downloaded 6.0 MB

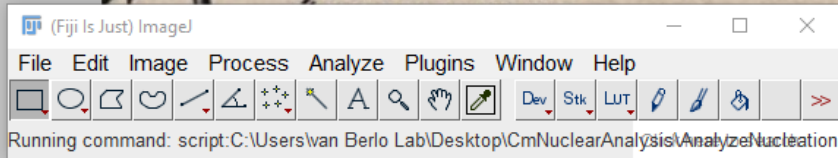
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.5/purrr_0.3.3.tgz'
Content type 'application/x-gzip' length 408939 bytes (399 KB)
=====
downloaded 399 KB

trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.5/jsonlite_1.6.tgz'
```



```
AnalyzeNucleation.py
File Edit Language Templates Run Tools Git Tabs Options
[+] [-]
van Berlo Lab
AnalyzeNucleation.py
1 from itertools import product
2 import os, re
3 import json
4
5 from fiji.util.gui import GenericDialogPlus as GenericDialog
6 from ij.io import Opener
7 import ij
8 from ij import ImagePlus, ImageStack, IJ, WindowManager
9 from ij.plugin import HyperStackConverter, ChannelSplitter, ImageCalculator
10 from ij.plugin.filter import BackgroundSubtractor, Analyzer
11 from ij.plugin.frame import RoiManager
12 from ij.process import StackStatistics, AutoThresholder
13 from ij.measure import ResultsTable
14
15
16 methodsList = ["Default",
17 "Huang",
18 "IJ_IsoData",
19 "Intermodes",
20 "IsoData",
21 "Li",
22 "MaxEntropy",
23 "Mean",
24 "MinError",
25 "Minimum",
26 "Moments",
27 "Otsu",
28 "Percentile",
29 "RenyiEntropy",
30 "Shanbhag",
31 "Triangle",
32 "Yen"]
33
34 def getCombos(dict, groups):
35     retval = {}
36     for d in dict:
37         idTuple = tuple(d[g] for g in groups)
38         z = retval.get(idTuple)
39         if z is None: retval[idTuple] = []
40         retval[idTuple].append(d)
```

Run Batch Kill  persistent Show Errors Clear



AnalyzeNucleation.py (Running)

File Edit Language Templates Run Tools Git Tabs Options

[+] [-]

van Berlo Lab

```
1 from itertools import product
2 import os, re
3 import json
4
5 from fiji.util.gui import GenericDialogPlus as GenericDialog
6 from ij.io import Opener
7 import ij
8 from ij import ImagePlus, ImageStack, IJ, WindowManager
9 from ij.plugin import ImagePlus2D, ImagePlus2DProcessor
10 from ij.plugin.filter import Threshold
11 from ij.plugin.frame import ColorConverter
12 from ij.process import BinaryImage, BinaryProcessor, ImageProcessor
13 from ij.measure import Measurements, ResultsTable
14
15
16 methodsList = ["Default",
17 "Huang",
18 "IJ_IsoData",
19 "Intermodes",
20 "IsoData",
21 "Li",
22 "MaxEntropy",
23 "Mean",
24 "MinError",
25 "Minimum",
26 "Moments",
27 "Otsu",
28 "Percentile",
29 "RenyiEntropy",
30 "Shanbhag",
31 "Triangle",
32 "Yen"]
33
34 def getCombos(dict, groups):
35     retval = {}
36     for d in dict:
37         idTuple = tuple(d[g] for g in groups)
38         z = retval.get(idTuple)
39         if z is None: retval[idTuple] = []
40         retval[idTuple].append(d)
```

Output Folder

Output folder  Browse...

OK Cancel

















































Run Batch Kill  persistent Show Errors Clear ▶

Started AnalyzeNucleation.py at Mon Nov 25 13:57:12 CST 2019

Image folder location	C:\Users\van Berlo Lab\Desktop	Browse...
Image filename pattern	(?P<row>[A-H])(?P<column>\d+)(?P<site>\d+)(?P<channel>[a-zA-Z]+)_001.tif	
Group By	site, channel	
Nuclear Stain Channel Name	DAPI	
Cardiomyocyte Image Channel Name	GFP	
<input type="checkbox"/> Are images stitched by well		
Nuclear thresholding method	Otsu	
<input checked="" type="checkbox"/> Threshold nuclei by stack?		
<input type="checkbox"/> Are the cardiomyocyte images brightfield?		
Cardiomyocyte thresholding method	Triangle	
<input checked="" type="checkbox"/> Threshold cardiomyocytes by stack?		
Rows per well	8	
Columns per well	7	
Nuclear minimum size (pixels)	50	
Cardiomyocyte minimum size (pixels)	500	

OK

Cancel

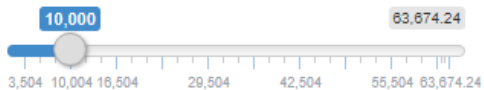
Name	Date modified	Type	Size
 savedSettings	11/25/2019 1:59 PM	JSON File	1 KB
 cm_column=12_row=H	10/17/2019 3:23 PM	CSV File	56 KB
 nuclej_column=12_row=H	10/17/2019 3:23 PM	CSV File	704 KB
 nucleilink_column=12_row=H	10/17/2019 3:23 PM	CSV File	72 KB
 cm_column=11_row=H	10/17/2019 3:20 PM	CSV File	56 KB
 nucleilink_column=11_row=H	10/17/2019 3:20 PM	CSV File	72 KB
 nuclej_column=11_row=H	10/17/2019 3:20 PM	CSV File	702 KB
 cm_column=10_row=H	10/17/2019 3:18 PM	CSV File	58 KB
 nuclej_column=10_row=H	10/17/2019 3:18 PM	CSV File	719 KB
 nucleilink_column=10_row=H	10/17/2019 3:18 PM	CSV File	75 KB
 cm_column=9_row=H	10/17/2019 3:17 PM	CSV File	62 KB
 nuclej_column=9_row=H	10/17/2019 3:17 PM	CSV File	751 KB
 nucleilink_column=9_row=H	10/17/2019 3:17 PM	CSV File	77 KB
 cm_column=8_row=H	10/17/2019 3:14 PM	CSV File	62 KB
 nucleilink_column=8_row=H	10/17/2019 3:14 PM	CSV File	81 KB
 nuclej_column=8_row=H	10/17/2019 3:14 PM	CSV File	797 KB
 cm_column=7_row=H	10/17/2019 3:13 PM	CSV File	61 KB
 nuclej_column=7_row=H	10/17/2019 3:13 PM	CSV File	694 KB
 nucleilink_column=7_row=H	10/17/2019 3:13 PM	CSV File	71 KB
 cm_column=6_row=H	10/17/2019 3:11 PM	CSV File	59 KB
 nuclej_column=6_row=H	10/17/2019 3:11 PM	CSV File	706 KB
 nucleilink_column=6_row=H	10/17/2019 3:11 PM	CSV File	71 KB
 cm_column=5_row=H	10/17/2019 3:08 PM	CSV File	59 KB
 nuclej_column=5_row=H	10/17/2019 3:08 PM	CSV File	612 KB
 nucleilink_column=5_row=H	10/17/2019 3:08 PM	CSV File	62 KB
 cm_column=4_row=H	10/17/2019 3:06 PM	CSV File	55 KB
 nuclej_column=4_row=H	10/17/2019 3:06 PM	CSV File	712 KB
 nucleilink_column=4_row=H	10/17/2019 3:06 PM	CSV File	72 KB
 cm_column=3_row=H	10/17/2019 3:05 PM	CSV File	61 KB
 nucleilink_column=3_row=H	10/17/2019 3:05 PM	CSV File	75 KB
 nuclej_column=3_row=H	10/17/2019 3:05 PM	CSV File	736 KB
 cm_column=2_row=H	10/17/2019 3:03 PM	CSV File	51 KB
 nuclej_column=2_row=H	10/17/2019 3:03 PM	CSV File	673 KB
 nucleilink_column=2_row=H	10/17/2019 3:03 PM	CSV File	69 KB
 cm_column=1_row=H	10/17/2019 3:02 PM	CSV File	52 KB
 nuclej_column=1_row=H	10/17/2019 3:02 PM	CSV File	621 KB
 nucleilink_column=1_row=H	10/17/2019 3:02 PM	CSV File	63 KB
 cm_column=12_row=G	10/17/2019 3:00 PM	CSV File	51 KB
 nucleilink_column=12_row=G	10/17/2019 3:00 PM	CSV File	63 KB
 nuclej_column=12_row=G	10/17/2019 2:59 PM	CSV File	618 KB
 cm_column=11_row=G	10/17/2019 2:58 PM	CSV File	63 KB
 nucleilink_column=11_row=G	10/17/2019 2:58 PM	CSV File	69 KB
 nuclej_column=11_row=G	10/17/2019 2:58 PM	CSV File	679 KB
 cm_column=10_row=G	10/17/2019 2:56 PM	CSV File	63 KB
 nuclej_column=10_row=G	10/17/2019 2:56 PM	CSV File	786 KB
 nucleilink_column=10_row=G	10/17/2019 2:56 PM	CSV File	79 KB
 cm_column=9_row=G	10/17/2019 2:53 PM	CSV File	67 KB
 nuclej_column=9_row=G	10/17/2019 2:52 PM	CSV File	921 KB

AnalyzeMultinucleatedServer.R x

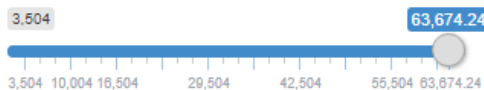
Run App

```
1 folderName = "C:/Users/van Berlo Lab/Desktop/Results/BL6 20d 1"
2 library(shiny)
3 knitr::opts_chunk$set(echo=FALSE,results='hide',fig.keep='all')
4 library(jsonlite)
5 savedParams=F
6 setwd(folderName)
7 if (savedParams){
8   paramlist = as.list(unlist(read_json("params.json")))
9   #This automatically assigns the read analysis parameters to R's global environment.
10  list2env(paramlist, environment())
11 }
12 library(ggplot2)
13 th = theme_set(theme_bw() + theme(panel.grid.major = element_blank(),
14                                 panel.grid.minor = element_blank(),
15                                 text = element_text(size=12),
16                                 axis.title = element_text(face="bold")))
17 library(dplyr)
18 library(purrr)
19 library(automethodr)
20 library(rlang)
21 groupSet = character(0)
22 find_metadata = function(filename){
23   strings = unlist(map(unlist(
24     regmatches(filename, gregexpr("_[a-zA-Z1-9]+[a-zA-Z0-9]+", filename)),
25     ~substring(., 2)))
26   keys = unlist(map(unlist(
27     regmatches(strings, gregexpr("^\\w+=", strings)),
28     ~substring(., 1, nchar(.)-1)))
29   values = unlist(map(unlist(
30     regmatches(strings, gregexpr("=\\w+$", strings)),
31     ~substring(., 2)))
32   setNames(values, keys)
33 }
34
35
```

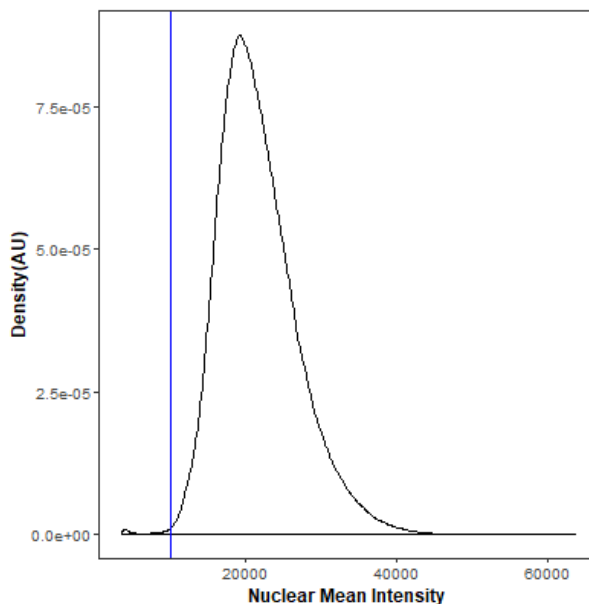
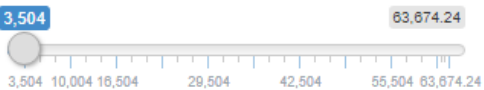
### Minimum Nuclear Mean Intensity Threshold



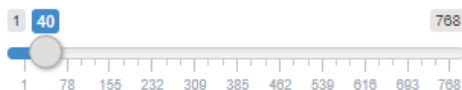
### Upper X limit of Graph



### Lower X limit of Graph



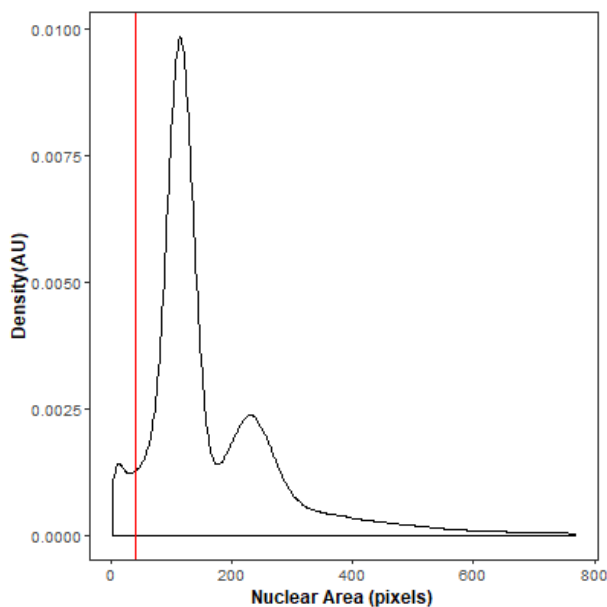
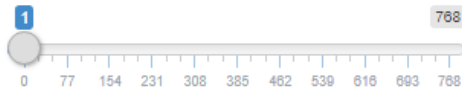
### Minimum Nuclear Area Threshold



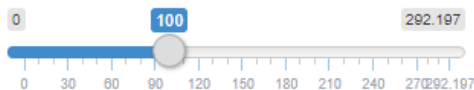
### Upper X limit of Graph



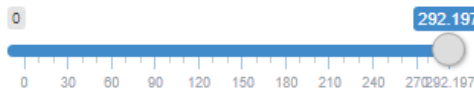
### Lower X limit of Graph



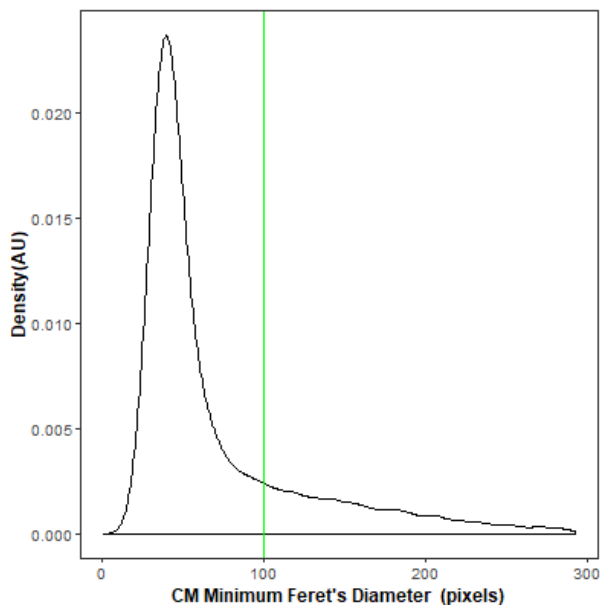
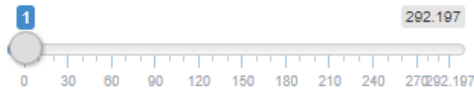
### Maximum Cardiomyocyte Minimum Feret's Diameter Threshold



### Upper X limit of Graph



### Lower X limit of Graph

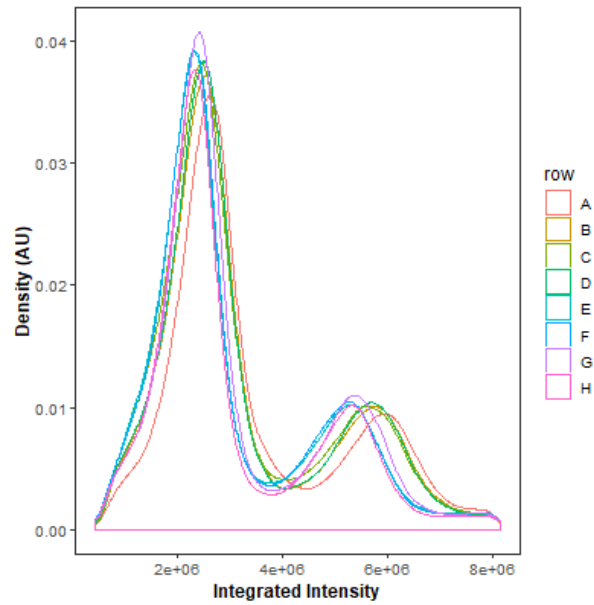
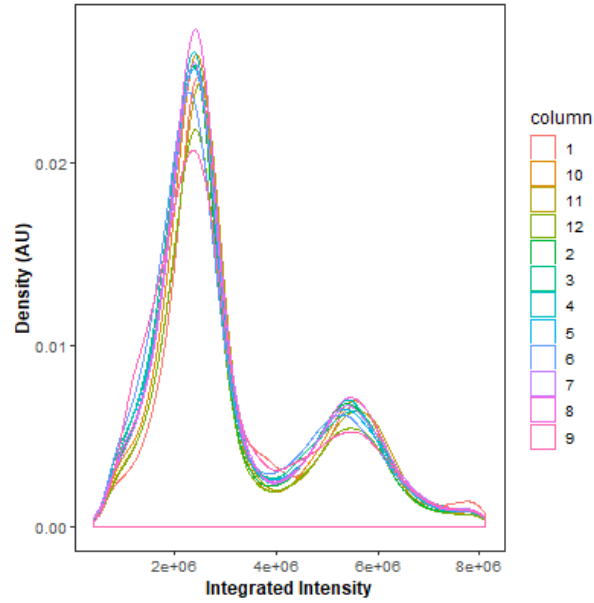
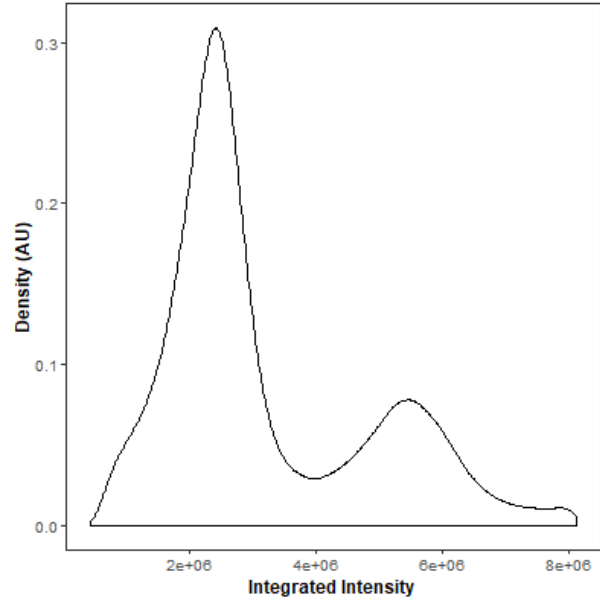


Apply Selected Thresholds?

Plot Intensity Distribution

Apply Selected Thresholds?

Plot Intensity Distribution



Normalize Separately by group?



Normalize Separately by group?

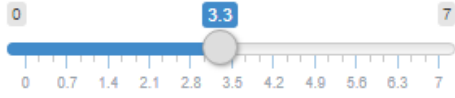
Calculate Ploidy

Plot Estimated Ploidy Distribution

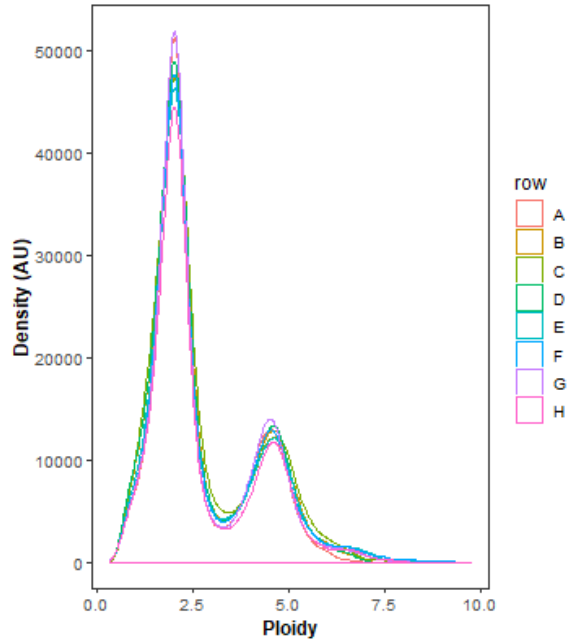
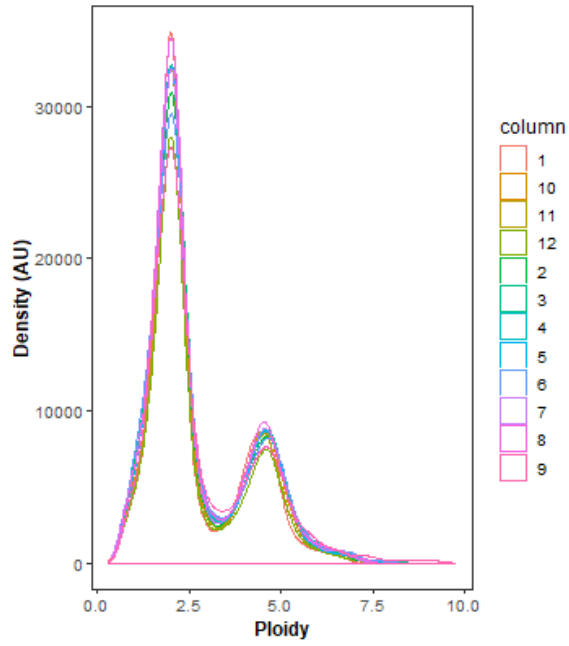
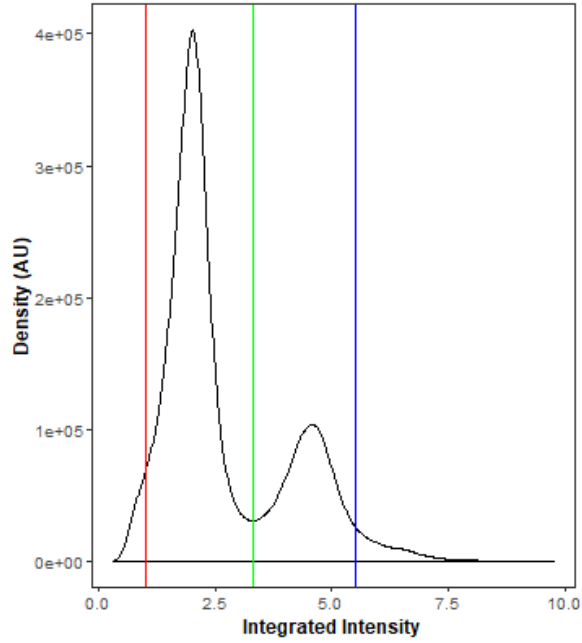
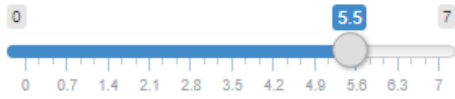
Minimum diploid threshold



Threshold between diploid and tetraploid



Maximum tetraploid threshold



Calculate Cardiomyocyte Ploidy and Nucleation Distributions

Plot and Save Into Output Folder

