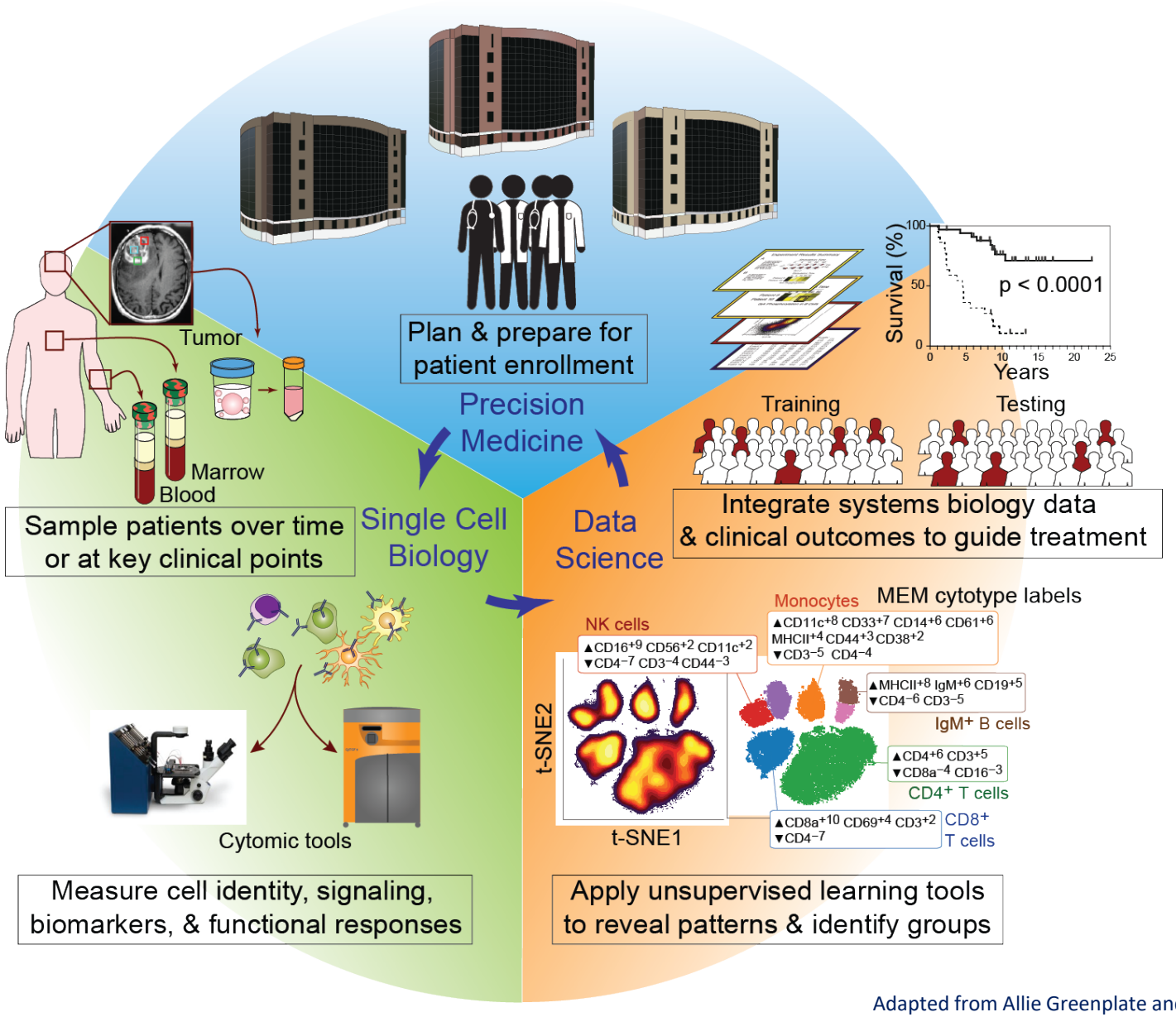
High Dimensional Data Analysis for Flow Cytometry

Sierra Barone
Justine Sinnaeve
12/15/2019

Goal: Systematically Dissect Cellular Mechanisms Across Time, Treatments, Tissues, & Tumor Types

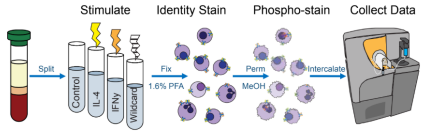


Adapted from Allie Greenplate and Kirsten Diggins

Flow Cytometry Workflow from Data Collection to Deep Analysis

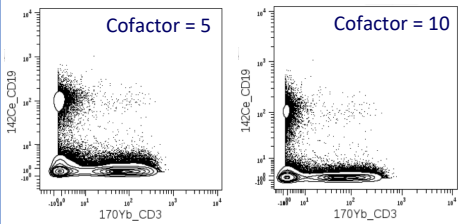
Data collection

1) Panel design
2) Data collection



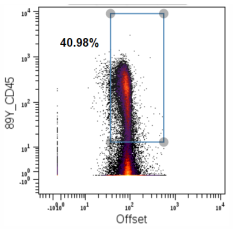
Data processing

3) Normalization
4) Concatenation
5) Scale transformation



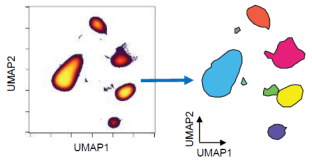
Distinguishing initial populations

6) Live single cell gating
7) Focal population gating



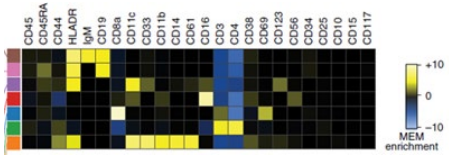
Revealing cell subsets

8) Feature selection
9) Dimensionality reduction
10) Identify cell clusters



Characterizing cell subsets

11) Feature comparison
12) Model populations
13) Learn cell identity
14) Statistical testing



How much can be automated?

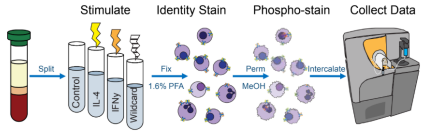
How do we select tools and use them well?

Workflow Steps Adapted from Diggins et al., *Methods* 2015

Some Preprocessing is Necessary Before Data Analysis

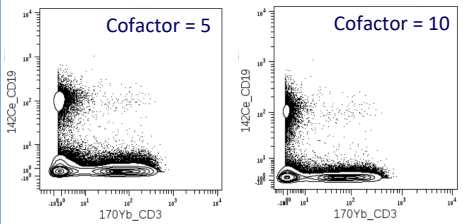
Data collection

1) Panel design
2) Data collection



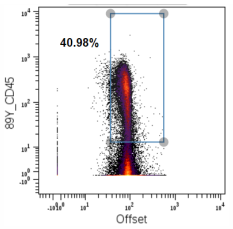
Data processing

3) Normalization
4) Concatenation
5) Scale transformation



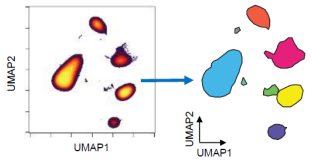
Distinguishing initial populations

6) Live single cell gating
7) Focal population gating



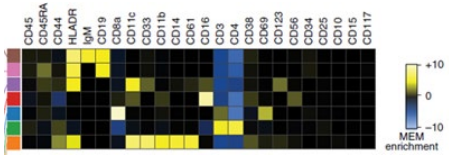
Revealing cell subsets

8) Feature selection
9) Dimensionality reduction
10) Identify cell clusters



Characterizing cell subsets

11) Feature comparison
12) Model populations
13) Learn cell identity
14) Statistical testing



How much can be automated?

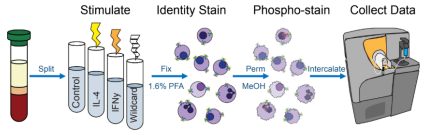
How do we select tools and use them well?

Workflow Steps Adapted from Diggins et al., *Methods* 2015

Scale Transformation Impacts Data Analysis

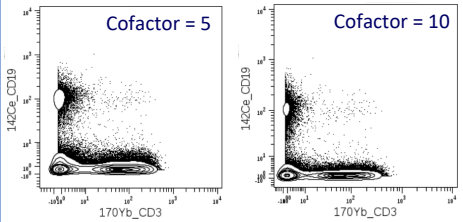
Data collection

1) Panel design
2) Data collection



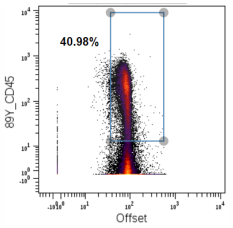
Data processing

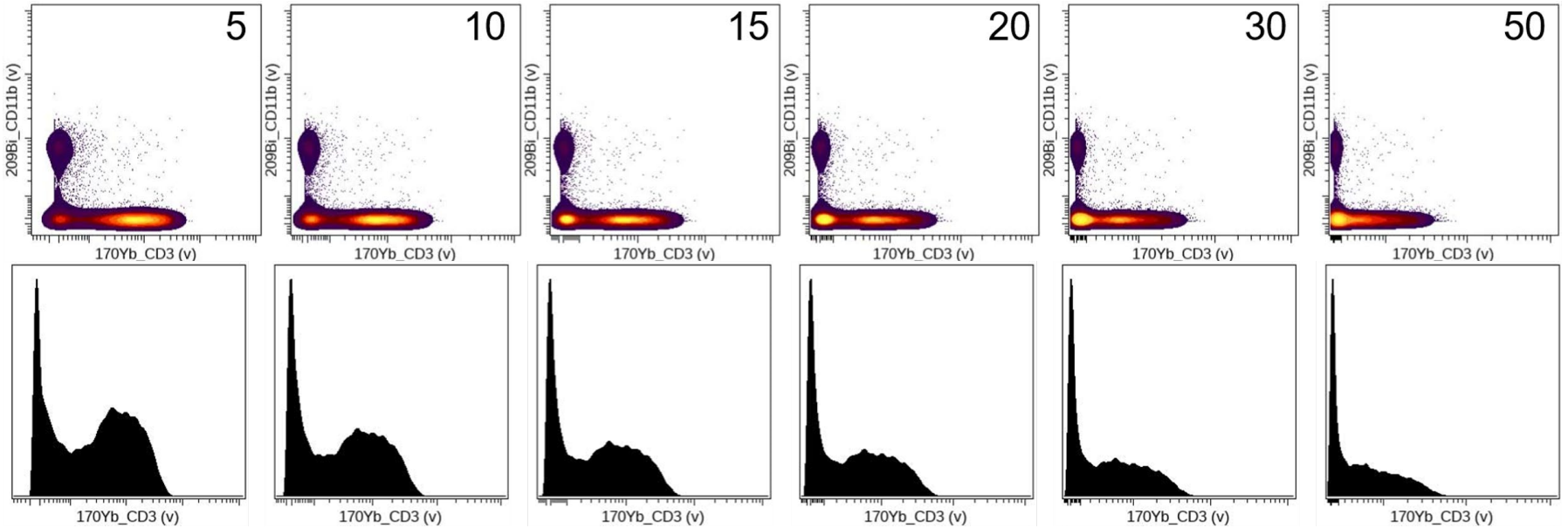
3) Normalization
4) Concatenation
5) Scale transformation



Distinguishing initial populations

6) Live single cell gating
7) Focal population gating

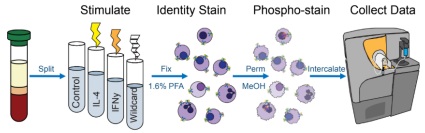




Biaxial Gating is Used to Identify Live Cells of Interest

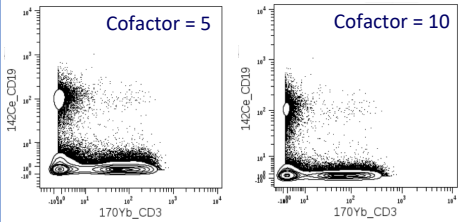
Data collection

1) Panel design
2) Data collection



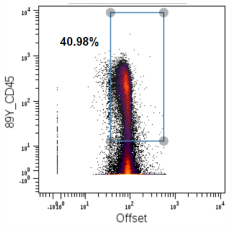
Data processing

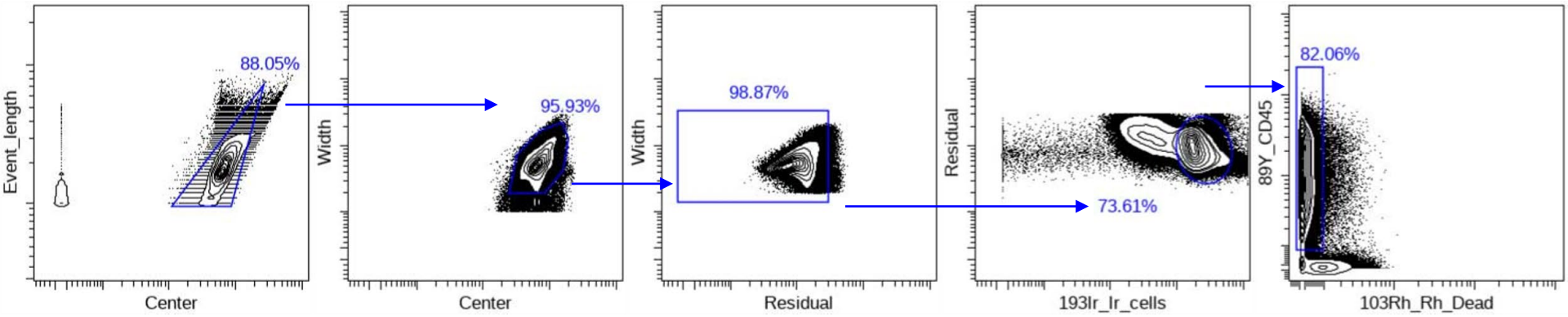
3) Normalization
4) Concatenation
5) Scale transformation



Distinguishing initial populations

6) Live single cell gating
7) Focal population gating

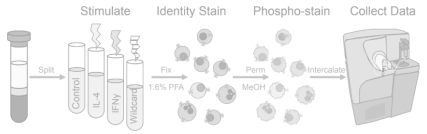




Dimensionality Reduction, Clustering, and Feature Comparisons are Combined in a Streamlined Workflow

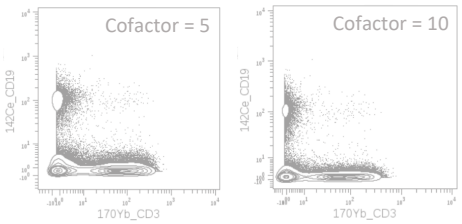
Data collection

1) Panel design
2) Data collection



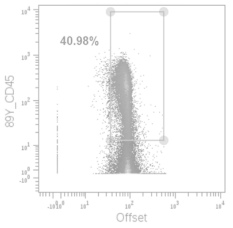
Data processing

3) Normalization
4) Concatenation
5) Scale transformation



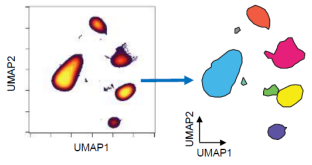
Distinguishing initial populations

6) Live single cell gating
7) Focal population gating



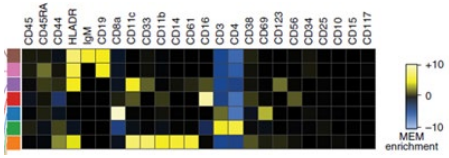
Revealing cell subsets

8) Feature selection
9) Dimensionality reduction
10) Identify cell clusters



Characterizing cell subsets

11) Feature comparison
12) Model populations
13) Learn cell identity
14) Statistical testing



How much can be automated?

How do we select tools and use them well?

Workflow Steps Adapted from Diggins et al., *Methods* 2015

Installation and Intro to Working in R

Install latest versions of R and Rstudio

To install R (language/environment) go to link below and follow instructions:

https://cran.r-project.org/

To install RStudio (IDE), go to link below and follow instructions to download free desktop version:

https://www.rstudio.com/products/rstudio/download/

PC Users: To install R Tools, go to the link below and download the recommended version:

https://cran.r-project.org/bin/windows/Rtools/

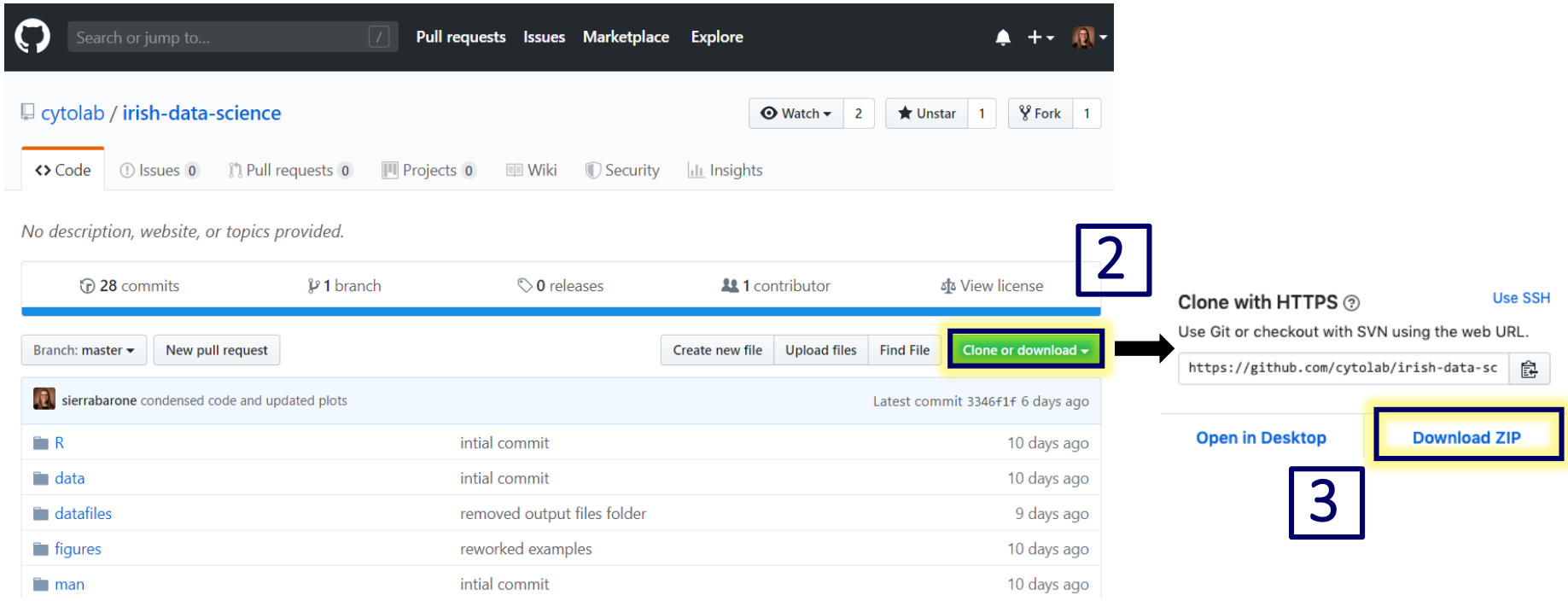
MAC Users: To install XQuartz, go to the link and download:

https://www.xquartz.org/

Download scripts, data, and R packages from GitHub

1 Go to link below and download repository:

https://github.com/cytolab/irish-data-science



Irish-data-science repository contents

1 installation script (R markdown, .rmd)

3 example analysis scripts (.rmd)

Data files (.fcs)

MEM package (.r, .rproj, etc.)

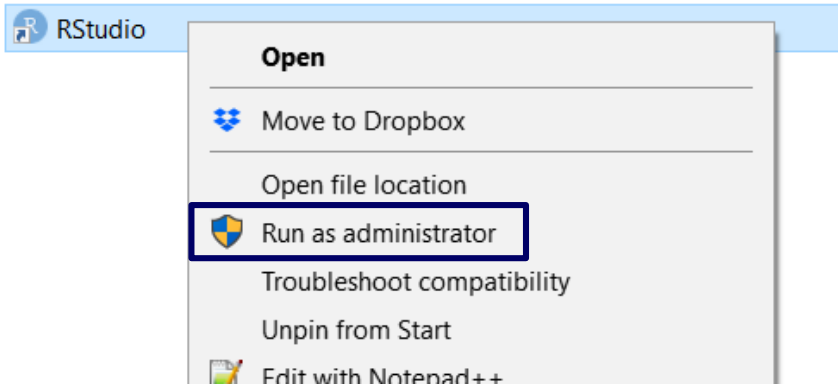
Documentation files (.rd, .md)

Other misc. files (.txt, .pdf, .rdata, etc.)

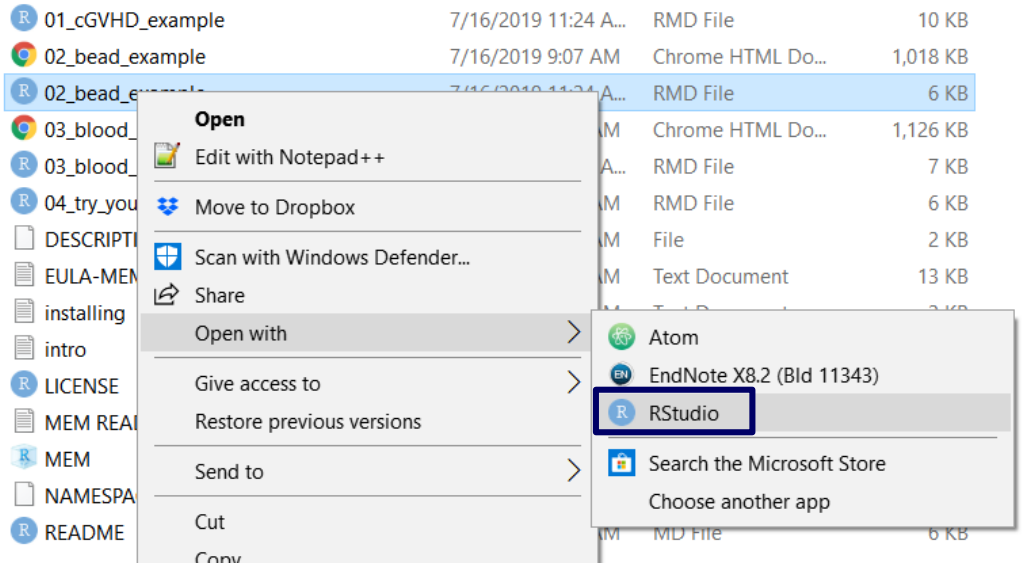
*make sure you unzip downloaded folder

RStudio

For PC users, run RStudio as administrator



For all, open .Rmd files with RStudio



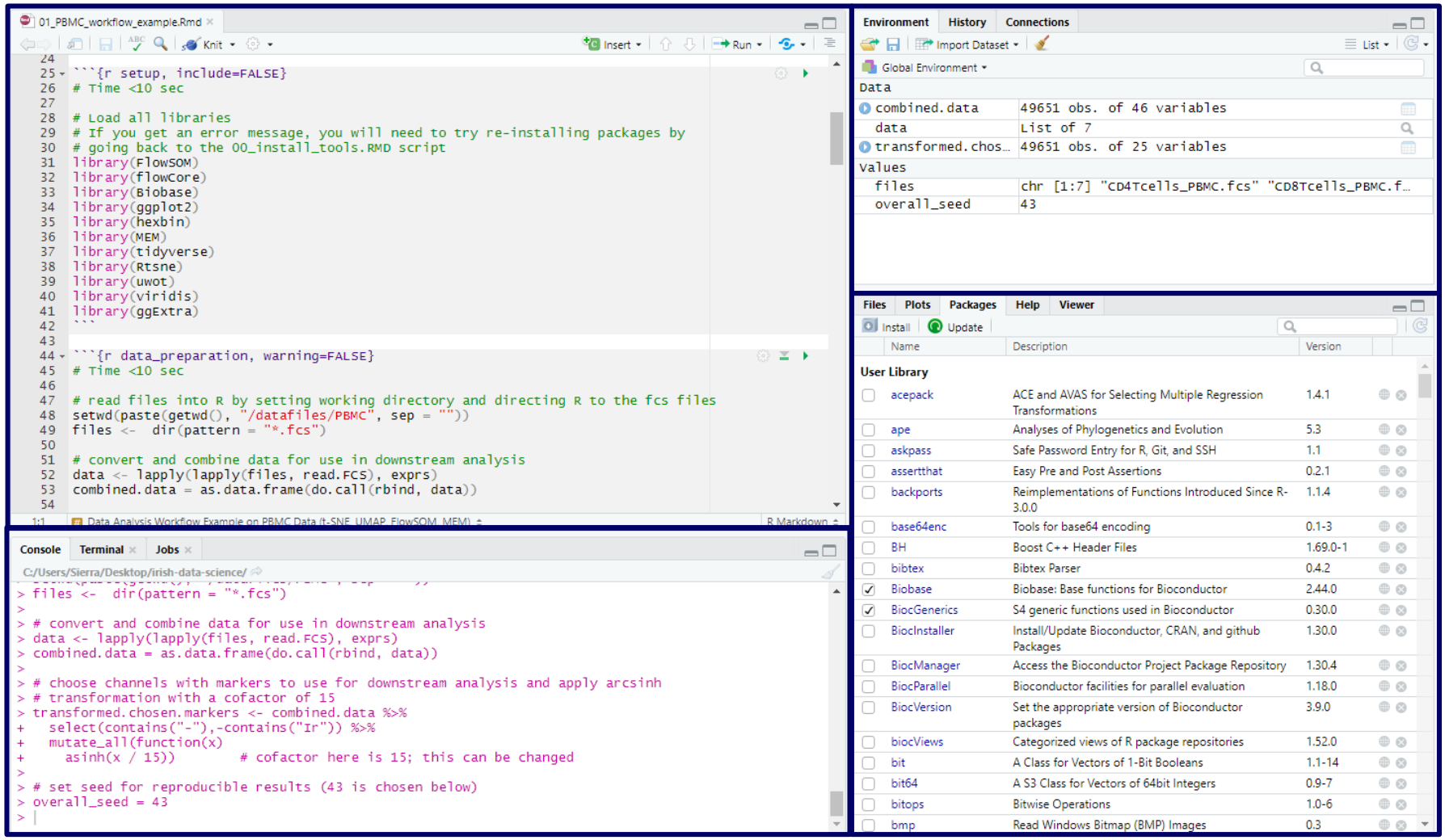
*make sure you unzip downloaded folder

Open 00_install_tools.rmd and 01_PBMC_extended_workflow_example.rmd

*make sure console is open

Working Script and Code

Environment

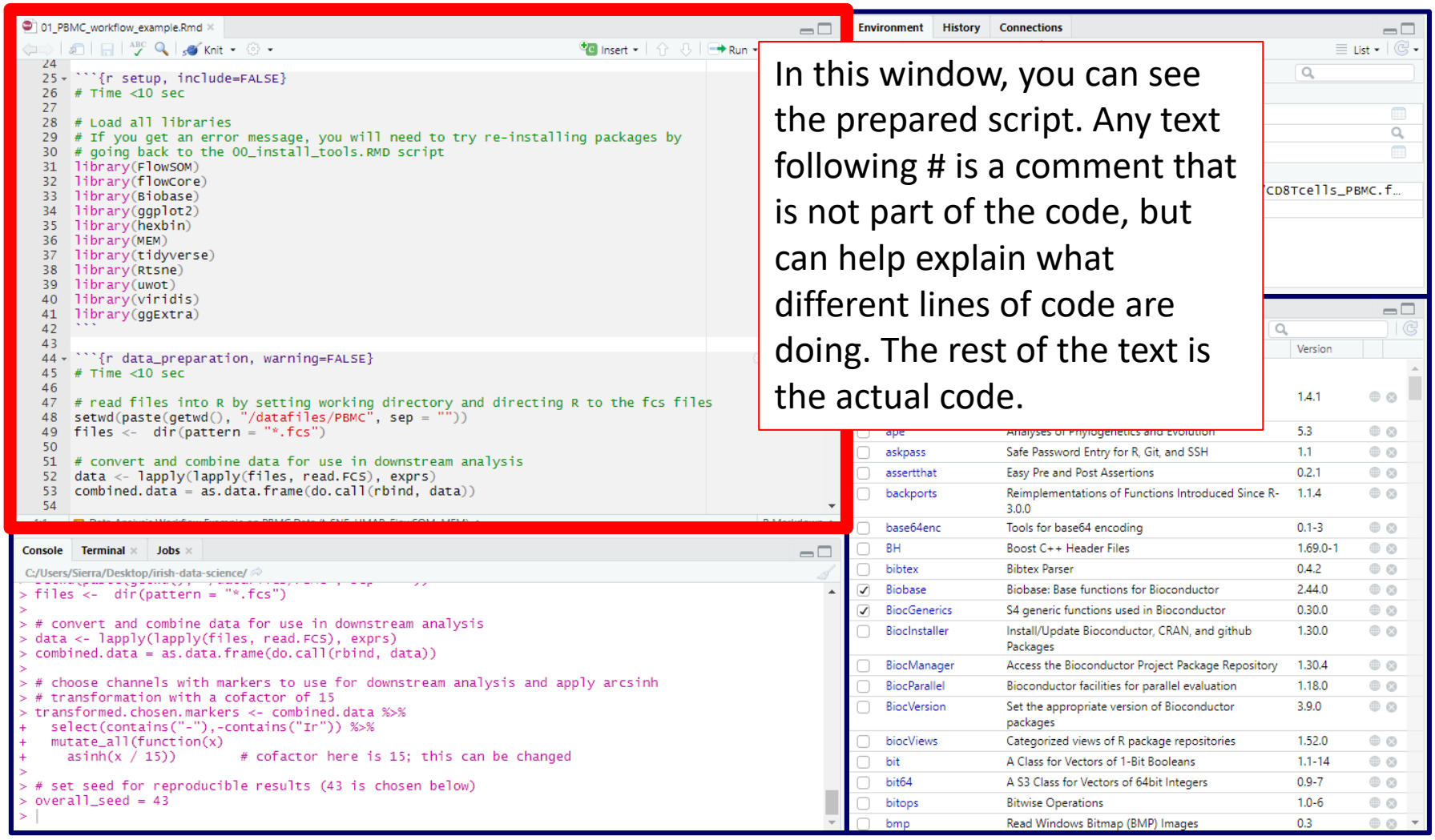


Console

Plots, Files, Help, etc.

Working Script and Code

Environment



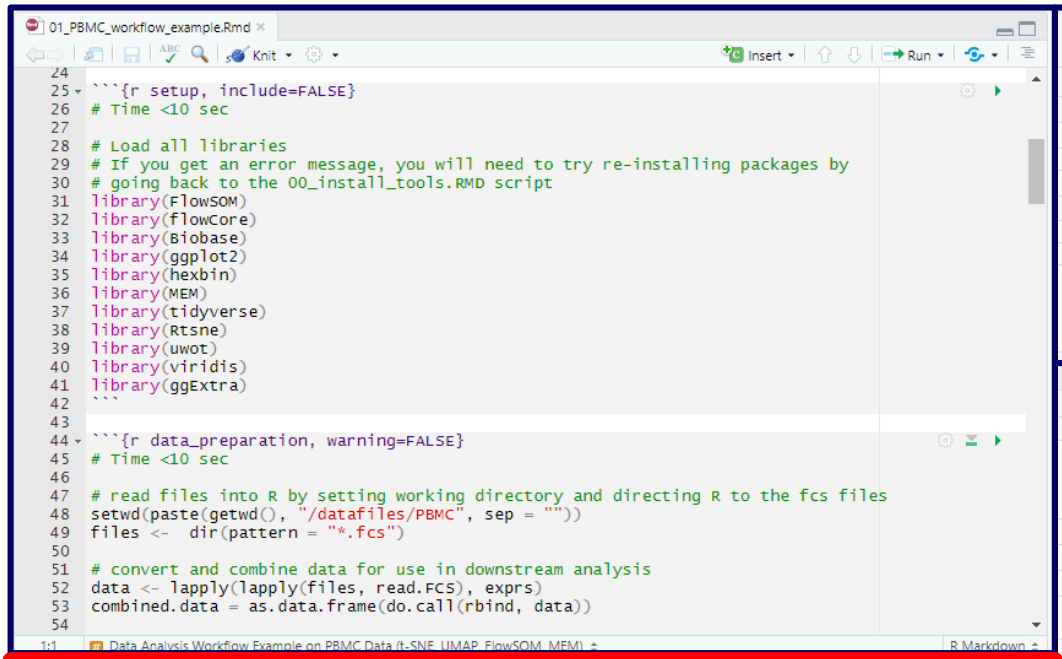
In this window, you can see the prepared script. Any text following # is a comment that is not part of the code, but can help explain what different lines of code are doing. The rest of the text is the actual code.

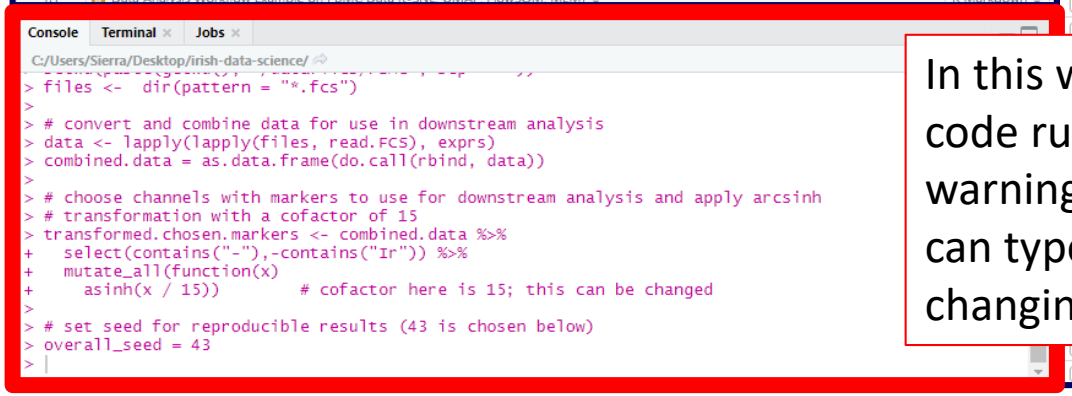
Console

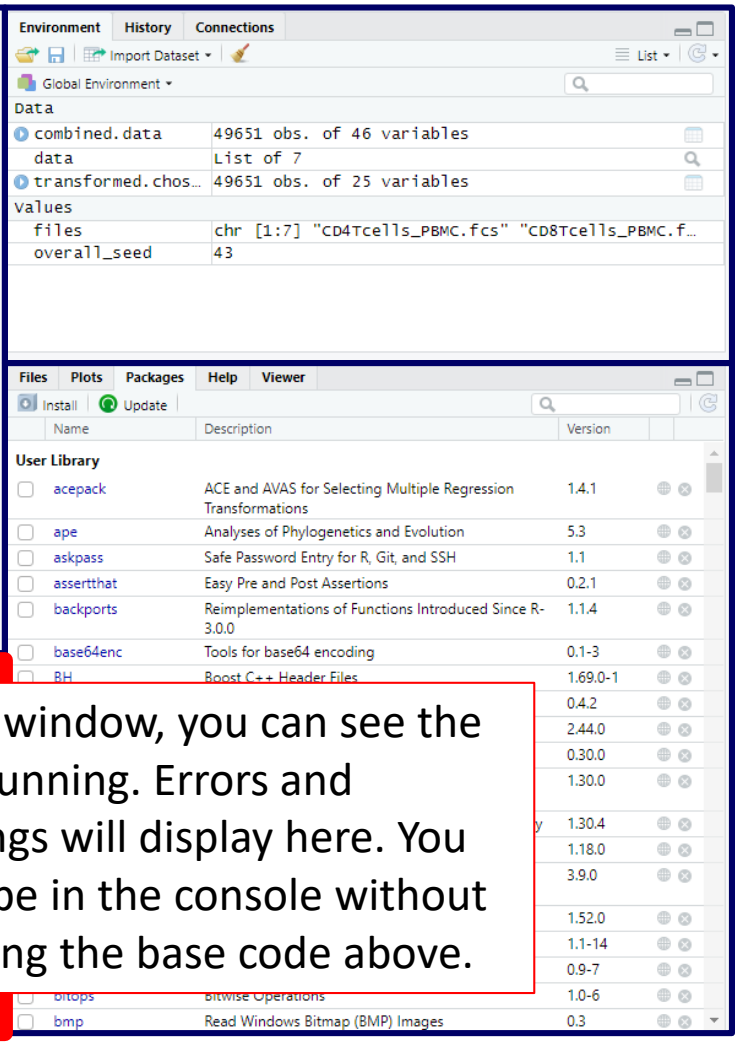
Plots, Files, Help, etc.

Working Script and Code

Environment







In this window, you can see the code running. Errors and warnings will display here. You can type in the console without changing the base code above.

Console

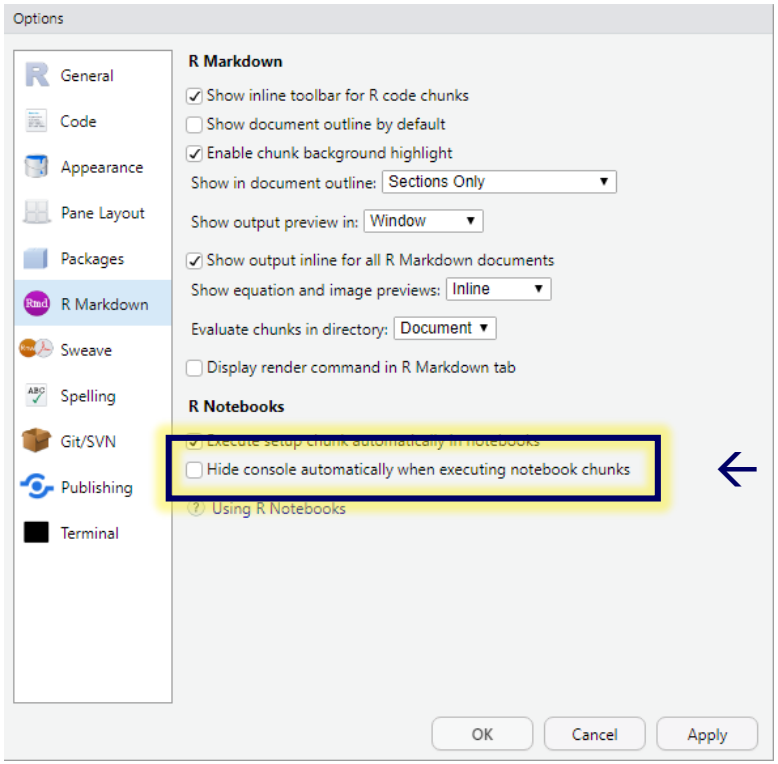
Plots, Files, Help, etc.

*make sure console is open

Make sure console stays open

1 Tools → Global Options

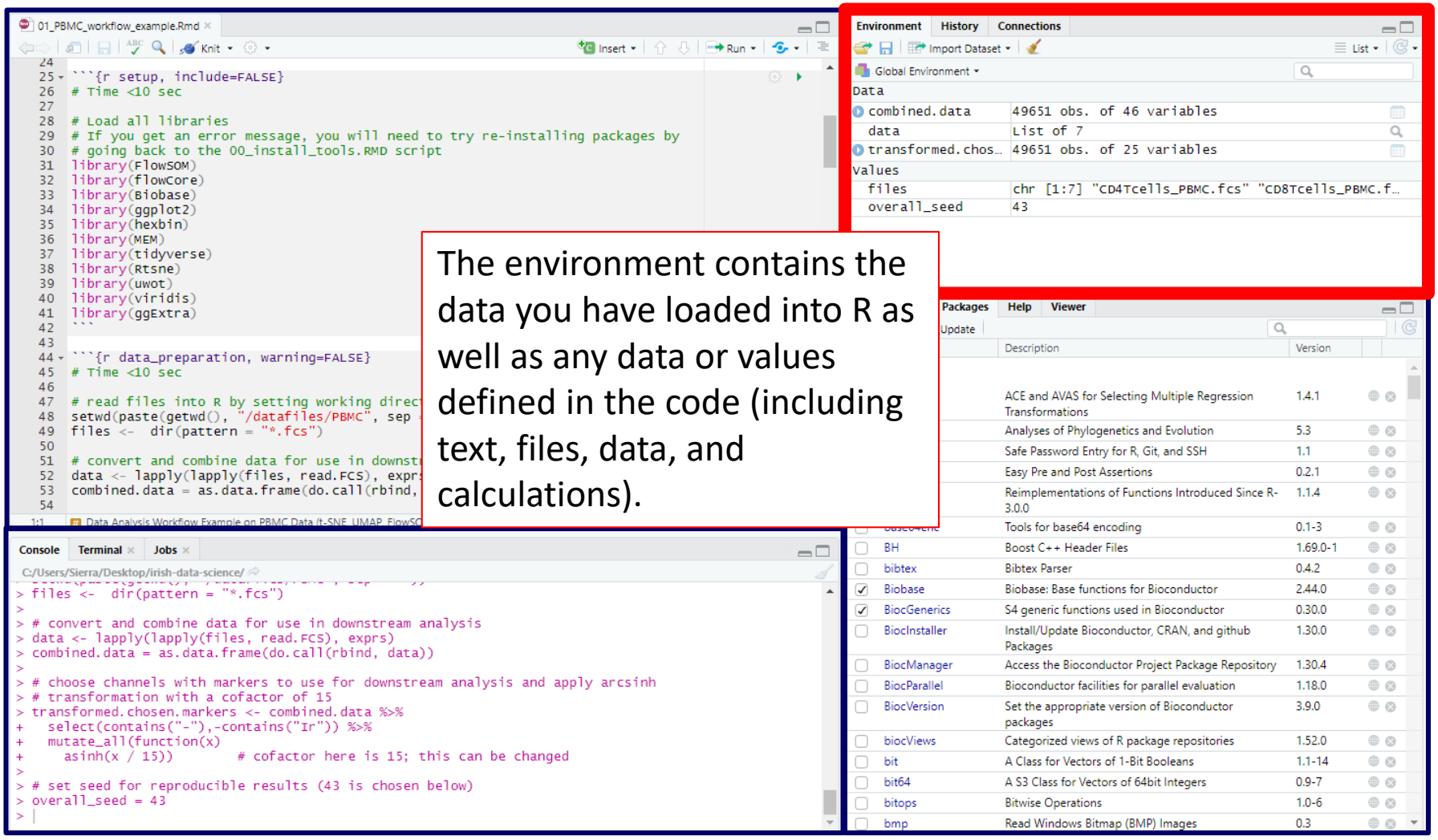
2



← Uncheck this option if checked

Working Script and Code

Environment



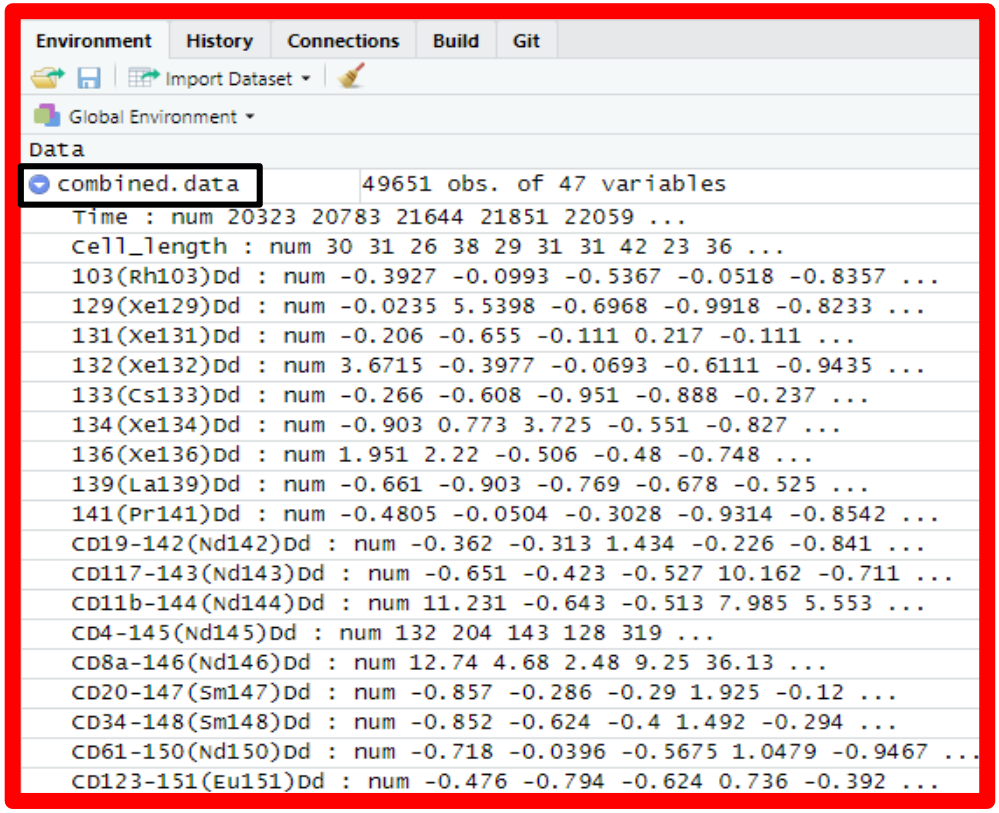
The environment contains the data you have loaded into R as well as any data or values defined in the code (including text, files, data, and calculations).

Console

Plots, Files, Help, etc.

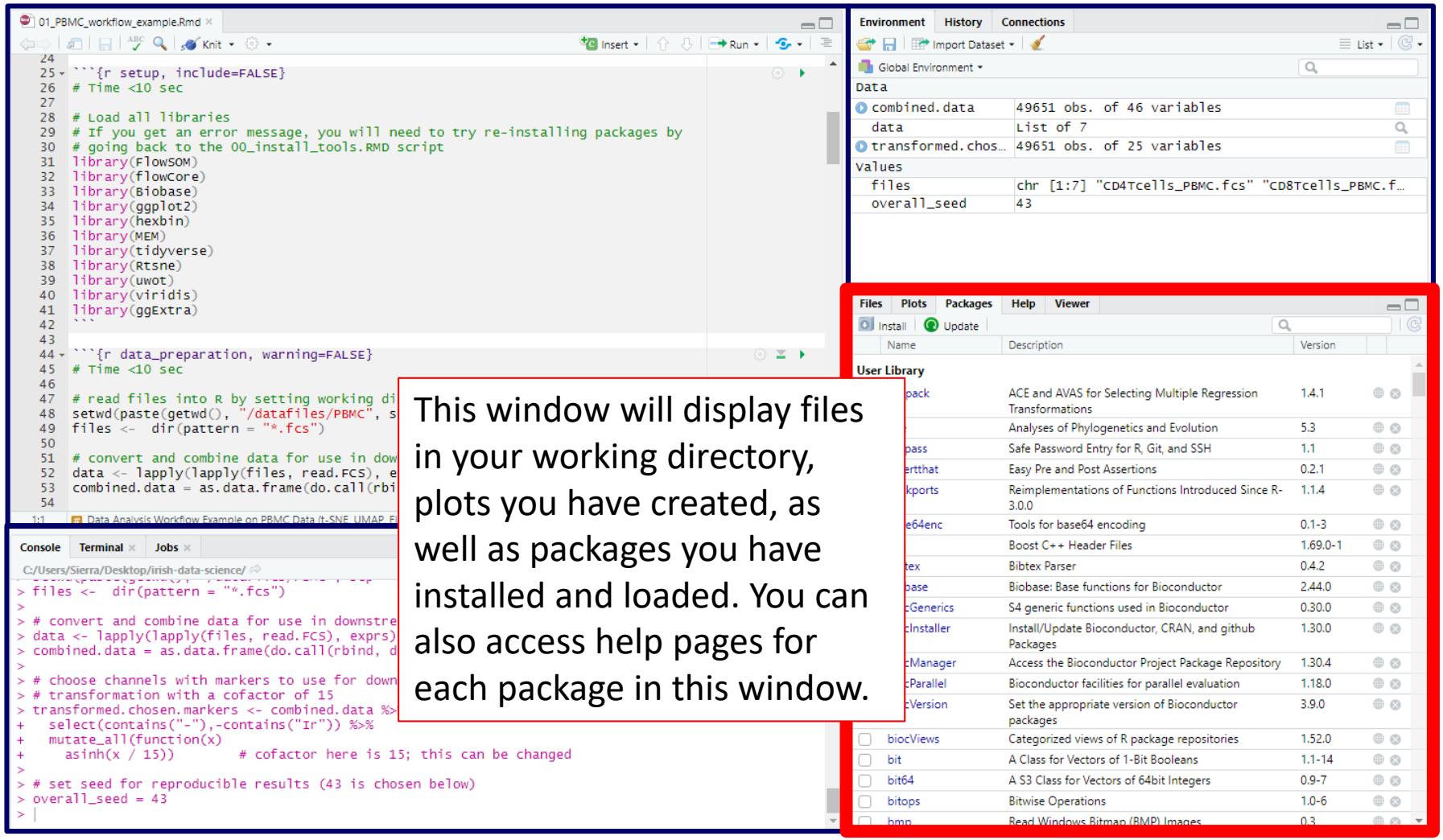
Environment

You can view each variable created by clicking the blue arrow or the variable's name in the environment. This will show you features for each event.



Working Script and Code

Environment

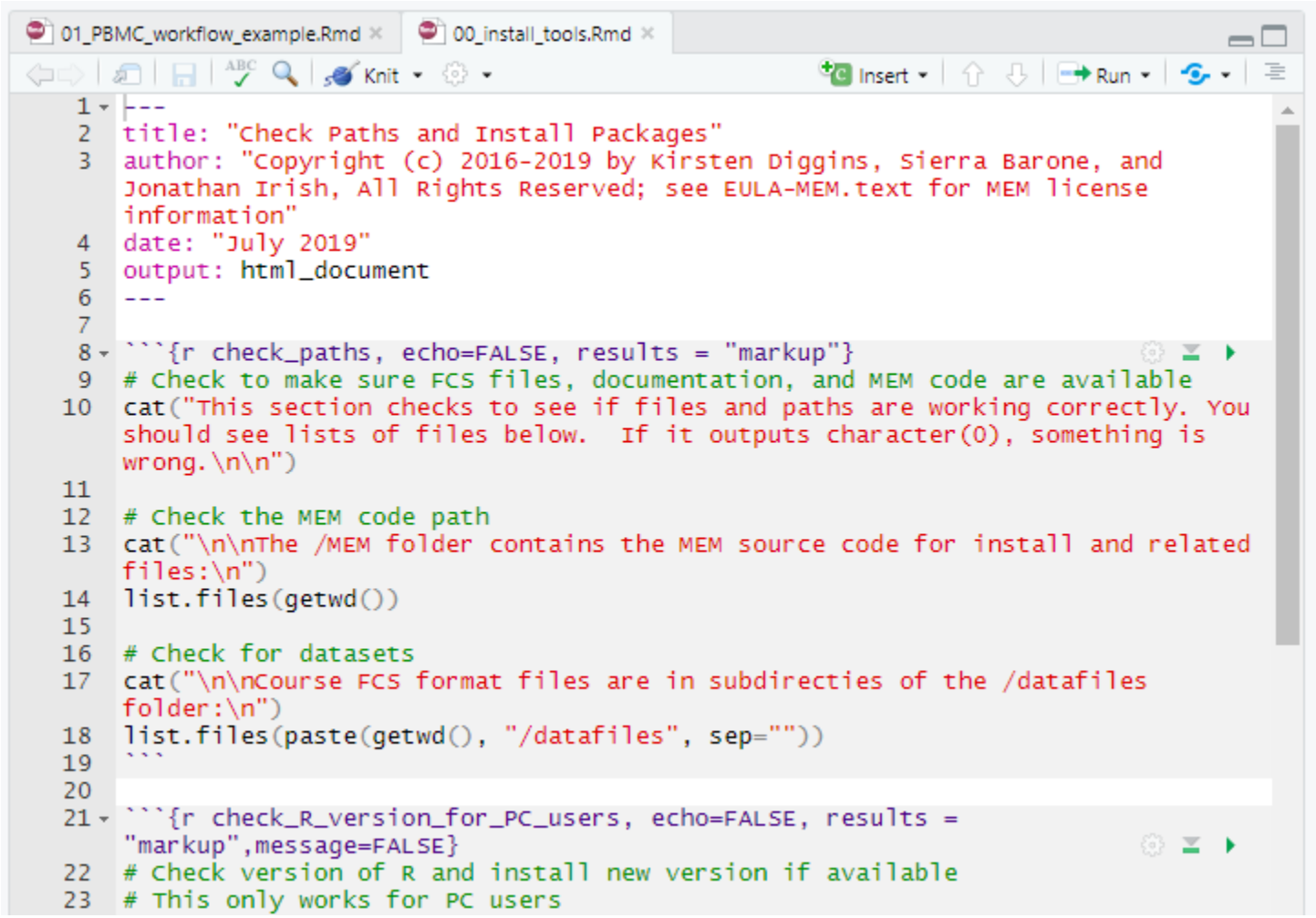


Console

Plots, Files, Help, etc.

Open 00_install_tools.rmd

1



Open **00_install_tools.rmd** and begin installing required packages

CNTL-ENTER or COMMAND-RETURN to run a single line of code

OR

Press play to run entire section of code

Code Section Title

```
```{r check_paths, echo=FALSE, results = "markup"}
# Check to make sure FCS files, documentation, and MEM code are available
cat("This section checks to see if files and paths are working correctly. You
should see lists of files below.  If it outputs character(0), something is
wrong.\n\n")

# Check the MEM code path
cat("\n\nThe /MEM folder contains the MEM source code for install and related
files:\n")
list.files(getwd())

# Check for datasets
cat("\n\nCourse FCS format files are in subdirecties of the /datafiles
folder:\n")
list.files(paste(getwd(), "/datafiles", sep=""))
```
```

This section checks that the files we will need are accessible in our working directory

```
```{r installation_notes, echo=FALSE, results = "markdown"}
# Print the contents a help file that explains installing packages
writeLines(readLines(paste(getwd(), "installing.txt", sep="/")))
```
```

This section prints installation text

Open **00_install_tools.rmd** and begin installing required packages

```{r install_bioconductor_packages, echo=FALSE, results = "hide"}
# install bioconductor and flow cytometry tools for R
cat("If this works, you should see 4 sets of messages about downloading files
that end in a message saying something like package 'BiocManager' successfully
unpacked and MD5 sums checked.  You should see this for BioCManager, Biobase,
flowCore, and FlowSOM.\n\n")
install.packages("BiocManager", repos = "http://cran.us.r-project.org")

if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("flowCore")
BiocManager::install("FlowSOM")
```

This section downloads Bioconductor and flow cytometry tools we will need

```{r test_flow_installs, echo=FALSE, results = "markdown"}
# Load and test whether bioconductor and flow packages are installed
cat("If this works, you may see Attaching Package messages or no message at
all; that's good.  If you get a warning, go back to the last CHUNK.\n\n")
library(FlowSOM)
library(flowCore)
library(Biobase)
```

This section tests to make sure Bioconductor and flow cytometry tools are installed

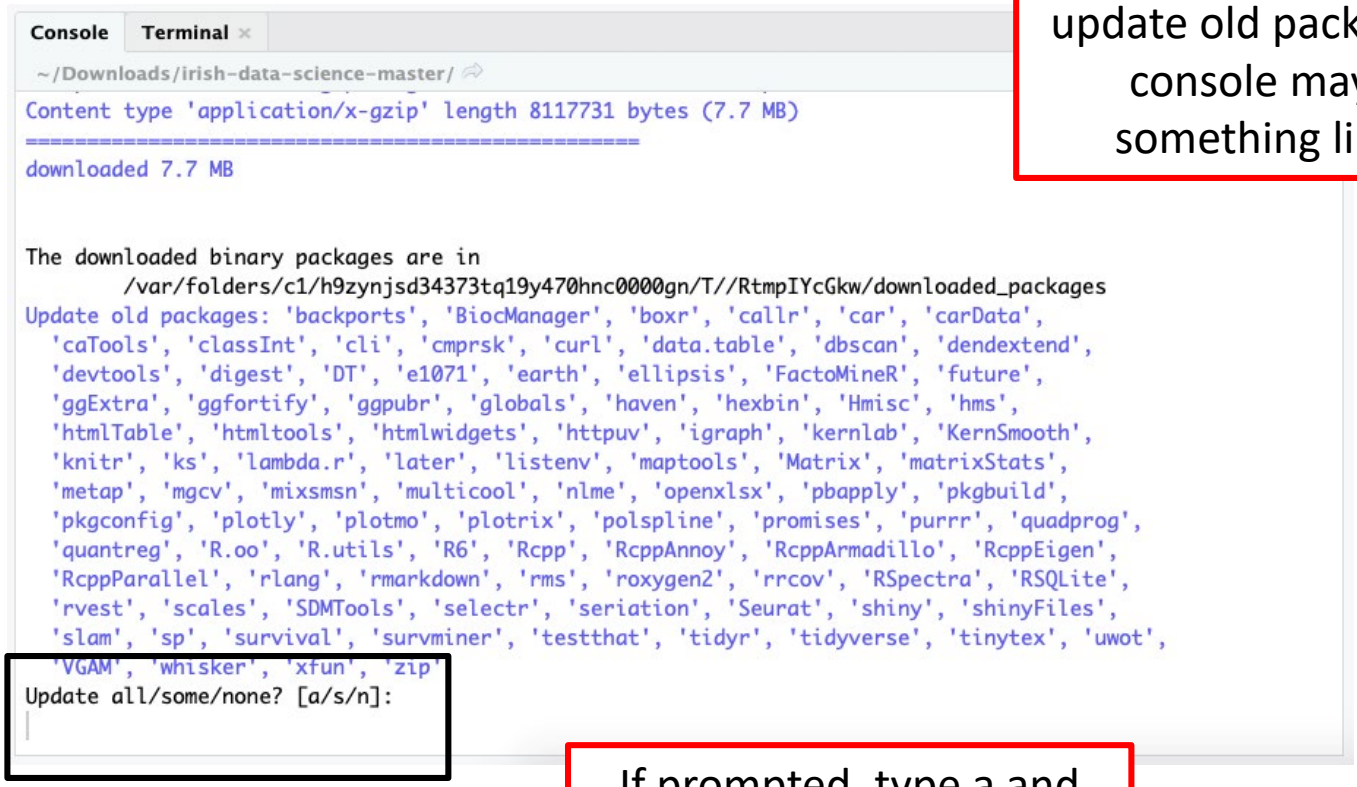
```{r install_ggplots, echo=FALSE, results = "markup"}
# install plotting packages
cat("If this works, you will see text about packages being downloaded.\n\n")
install.packages("gplots", repos = "http://cran.us.r-project.org")
install.packages("ggplot2", repos = "http://cran.us.r-project.org")
install.packages("hexbin", repos = "http://cran.us.r-project.org")
install.packages("viridis", repos = "http://cran.us.r-project.org")
install.packages("ggExtra", repos = "http://cran.us.r-project.org")
```

```{r load_gplots, echo=FALSE, results = "markup"}
# Load and test whether gplots and ggplot2 packages are installed
cat("If this works, you may see Attaching Package messages or no message at
all; that's good.  If you get a warning, go back to the last CHUNK.\n\n")
library(gplots)
library(ggplot2)
library(hexbin)
library(viridis)
library(ggExtra)
```

The next sections install and load the tools to make plots

You may be prompted to enter a value into the console

You may be asked to update old packages. The console may look something like this



If prompted, type a and then enter/return in the console

Signs that the code is running

Green bar alongside numbered line of code

Play button turns to a clock or red square

Working Script and Code

```
60 ```{r run_t-SNE}
61 # Time ~5 min
62 set.seed(overall_seed)
63
64 # the line below will run t-SNE on the scaled surface markers (to see help page
65 # for t-SNE, type "?Rtsne -- enter" in console)
66
67 # you can view t-SNE progress by opening up the console below
68 mytSNE = Rtsne(
69   transformed.chosen.markers,                    # input scaled data
```

```
85 ```

Performing PCA
Read the 49651 x 25 data matrix successfully!
OpenMP is working. 1 threads.
Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
Computing input similarities...
Building tree...
 - point 10000 of 49651
 - point 20000 of 49651
 - point 30000 of 49651

86
87 ```{r plot_t-SNE}
88 # Time <10 sec
```

Pinwheel in space between sections/chunks of code

Console

Stop sign in console panel

Open **00_install_tools.rmd** and begin installing required packages

```{r install_MEM, echo=FALSE, results = "markup"}
# install MEM, load it, and test if it is all set
cat("If this works, you should see several lines about installing files, then
DONE (MEM) near the end.  The MEM help page will also open in the Help menu in
RStudio.\n\n")

# If you have previously installed MEM, you may get an error message. If this
is the case, try restarting your RStudio session
install.packages(getwd(), type="source", repos=NULL)
library(MEM)
?MEM

# OR
# install.packages("devtools", repos = "http://cran.us.r-project.org")
# devtools::install_github("cytolab/mem")
```

This section installs and loads the marker enrichment modeling tool

```{r install_last_packages, echo=FALSE, results = "markup"}
# install the last packages for UMAP, t-SNE and other tools
print("You may see a bunch of messages, this is OK as long as they are not
errors.\n\n")
install.packages("tidyverse", repos = "http://cran.us.r-project.org")
install.packages("Rtsne", repos = "http://cran.us.r-project.org")
install.packages("uwot", repos = "http://cran.us.r-project.org")
install.packages("RColorBrewer", repos = "http://cran.us.r-project.org")
```

```{r load_last_packages, echo=FALSE, results = "markup"}
# Load and test the last libraries
library(tidyverse)
library(Rtsne)
library(uwot)
library(RColorBrewer)
```

These sections install and load the other tools we will use for analysis

Open 01_PBMC_extended_workflow_example.rmd and work through the example

01_PBMC_extended_workflow_example.rmd

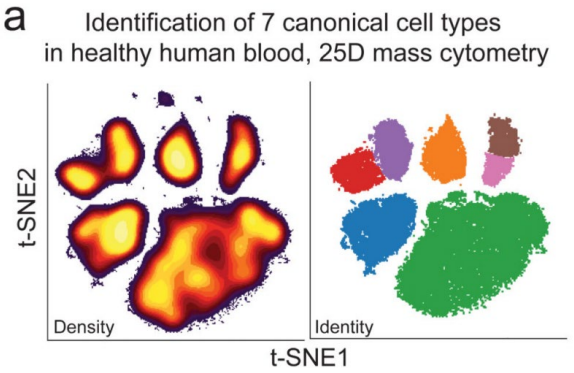
```
---
title: "Data Analysis Workflow Example on PBMC Data (t-SNE, UMAP, FlowSOM, MEM)"
author: "Copyright (c) 2016-2019 by Kirsten Diggins, Sierra Barone, and Jonathan Irish, All
Rights Reserved; see EULA-MEM.text for MEM license information"
date: "October 2019"
output:
  pdf_document:
    latex_engine: xelatex
  html_document:
    df_print: paged
editor_options:
  chunk_output_type: inline
---

This data set contains 7 FCS (flow cytometry standard) files. Each FCS file
contains single cell data for one cell subset that is a well-established,
phenotypically distinct population. This is mass cytometry data for healthy
human PBMC (peripheral blood mononuclear cells). The populations were expert
gated following a t-SNE analysis. The first section of the code will run two
dimensionality reduction tools, UMAP and t-SNE, on the data set. Next, you
will run FlowSOM on the both the UMAP and t-SNE axes to cluster, or group
together, the various cell populations. Finally, you will run MEM to see
enrichment scores for each of the FlowSOM clusters or populations that have been
expert gated. The goal of this exercise is to run several computational tools on
a single cell data set to get a feel for the workflow used in the Irish lab as
well as compare the various analysis methods. The method for comparison of the
cell populations by automated or manual analysis is RMSD.

```{r setup, include=FALSE}
# Time <10 sec

# Load all libraries
# If you get an error message, you will need to try re-installing packages by
# going back to the 00_install_tools.RMD script
library(FlowSOM)
library(flowCore)
library(Biobase)
library(ggplot2)
library(hexbin)
library(MEM)
library(tidyverse)
library(Rtsne)
library(uwot)
library(viridis)
library(ggExtra)
library(RColorBrewer)
```
```

A description of the code and its purpose



This section loads the necessary libraries

Diggins et al., *Nature Methods* 2017

01_PBMC_extended_workflow_example.rmd

Data Preparation

```r
```{r data_preparation, warning=FALSE}
# Time <10 sec

# read files into R by setting working directory and directing R to the fcs
# files
setwd(paste(getwd(), "/datafiles/PBMC", sep = ""))
files <-  dir(pattern = "*.fcs")

# convert and combine data for use in downstream analysis
data <- lapply(lapply(files, read.FCS), exprs)
combined.data = as.data.frame(do.call(rbind, mapply(
  cbind, data, "cluster" = c(1:length(data)), SIMPLIFY = F)))

# choose channels with markers to use for downstream analysis and apply
arcsinh
# transformation with a cofactor of 15
transformed.chosen.markers <- combined.data %>%
  select(contains("-"),-contains("Ir")) %>%
  mutate_all(function(x)
    asinh(x / 15))        # cofactor here is 15; this can be changed

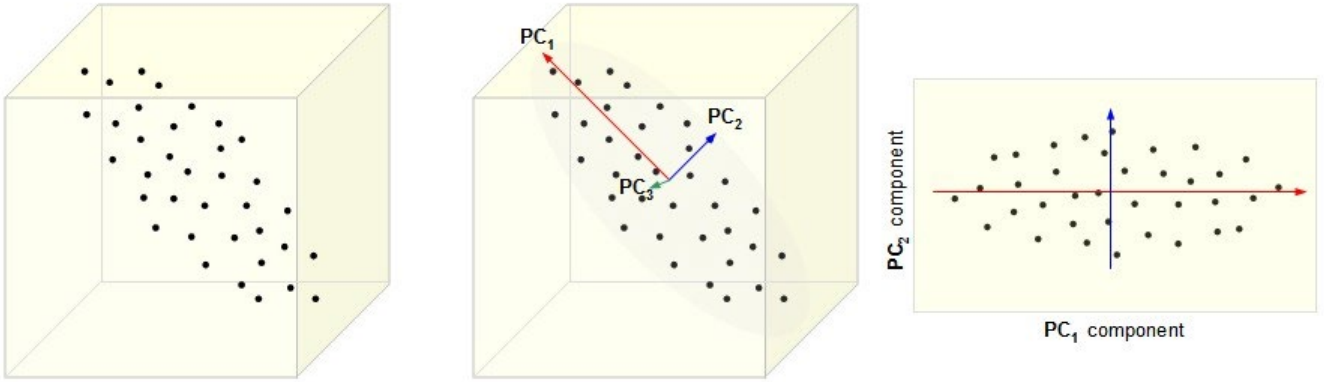
# set seed for reproducible results (43 is chosen below)
overall_seed = 43
```
```

Read the data files into R and format for analysis

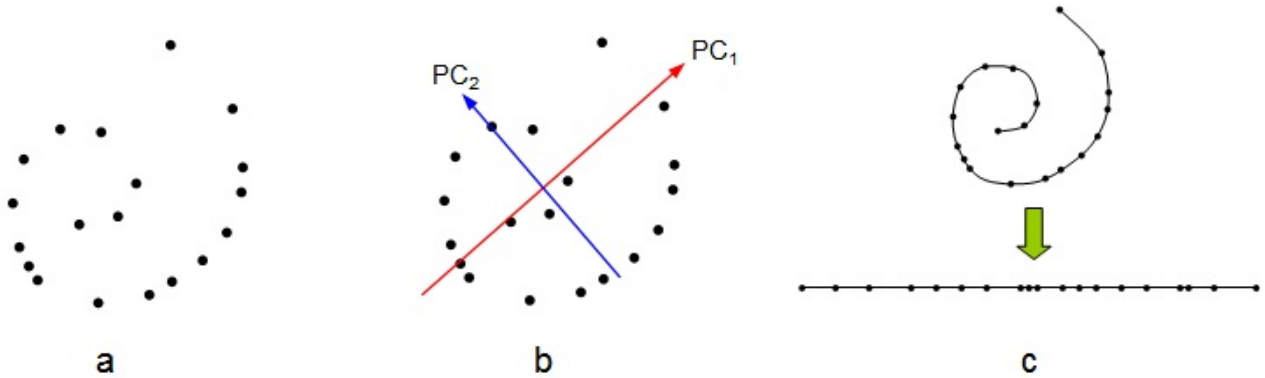
Select channels and scale the data

Choose parameters

PCA is a Linear Dimensionality Reduction Tool



An illustration of PCA. **a)** A data set given as 3-dimensional points. **b)** The three orthogonal Principal Components (PCs) for the data, ordered by variance. **c)** The projection of the data set into the first two PCs, discarding the third one.

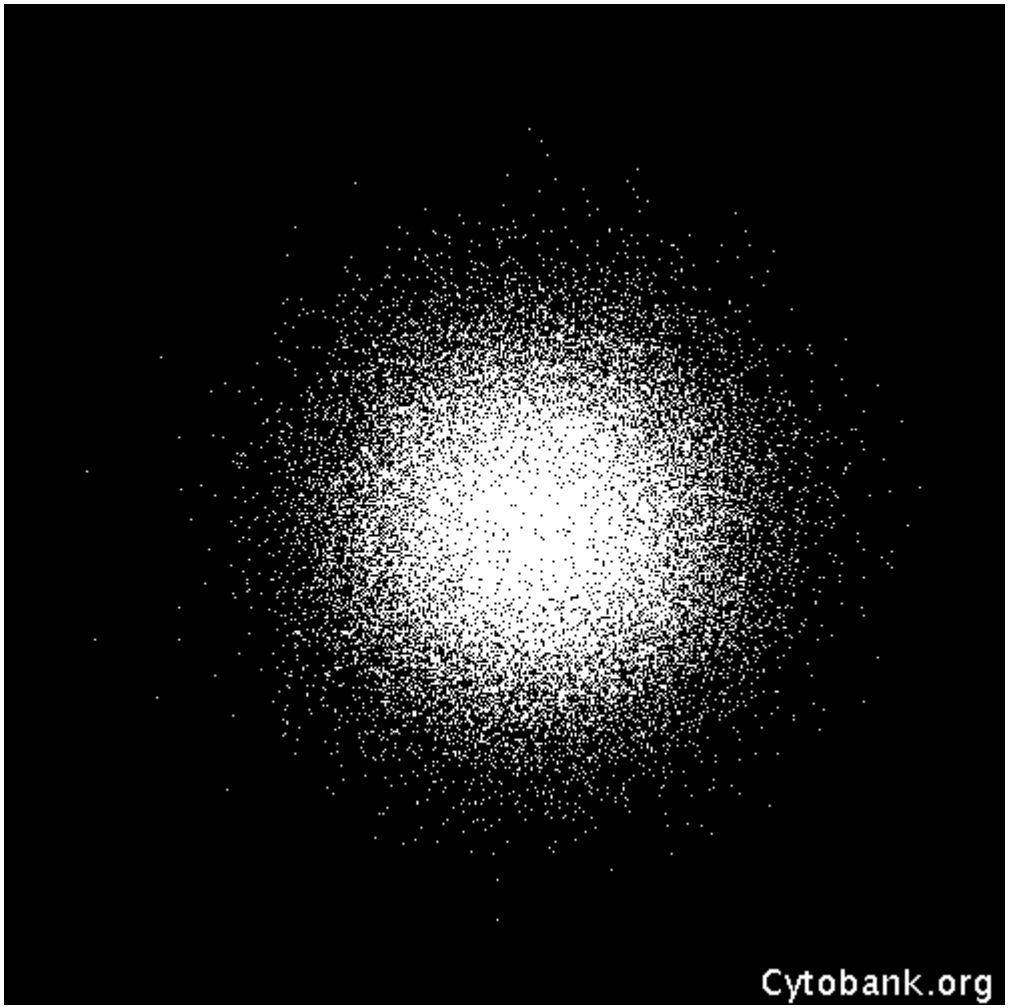


Effects of dimensionality reduction on an inherently non-linear data set. **a)** The original data given as a two-dimensional set. **b)**PCA identifies two PCs as contributing significantly to explain the data variance. **c)** However, the inherent topology (connectivity) of the data helps identify the set as being one-dimensional, but non-linear.

t-Distributed Stochastic Neighbor Embedding is a Dimensionality Reduction Tool

minimizes the divergence between two distributions (one that measures pairwise similarities of input objects and one that measures pairwise similarities of corresponding low-dimensional points)

Parameters:
-perplexity
-iterations
-seed

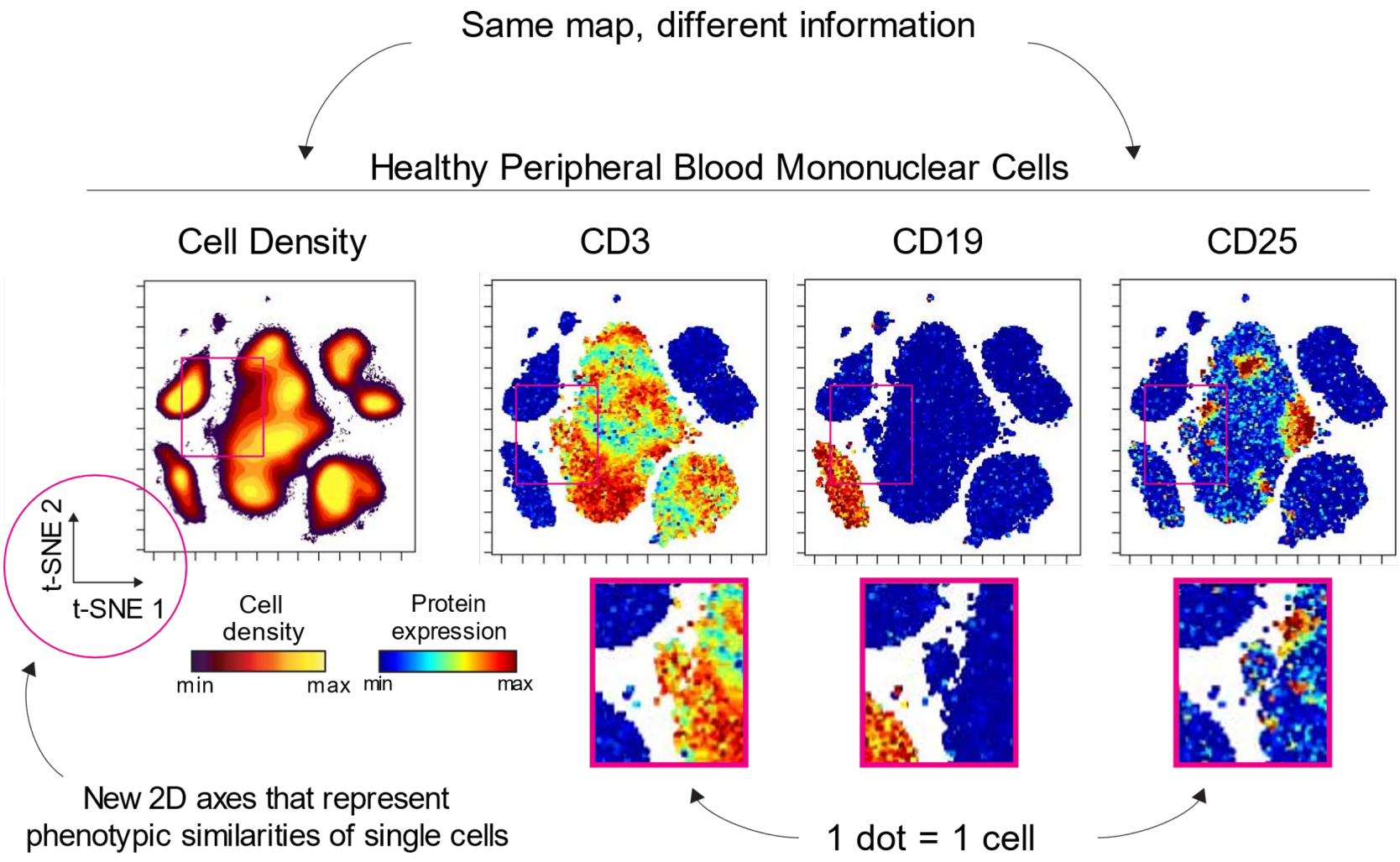


developed by Laurens van der Maaten and Geoffrey Hinton

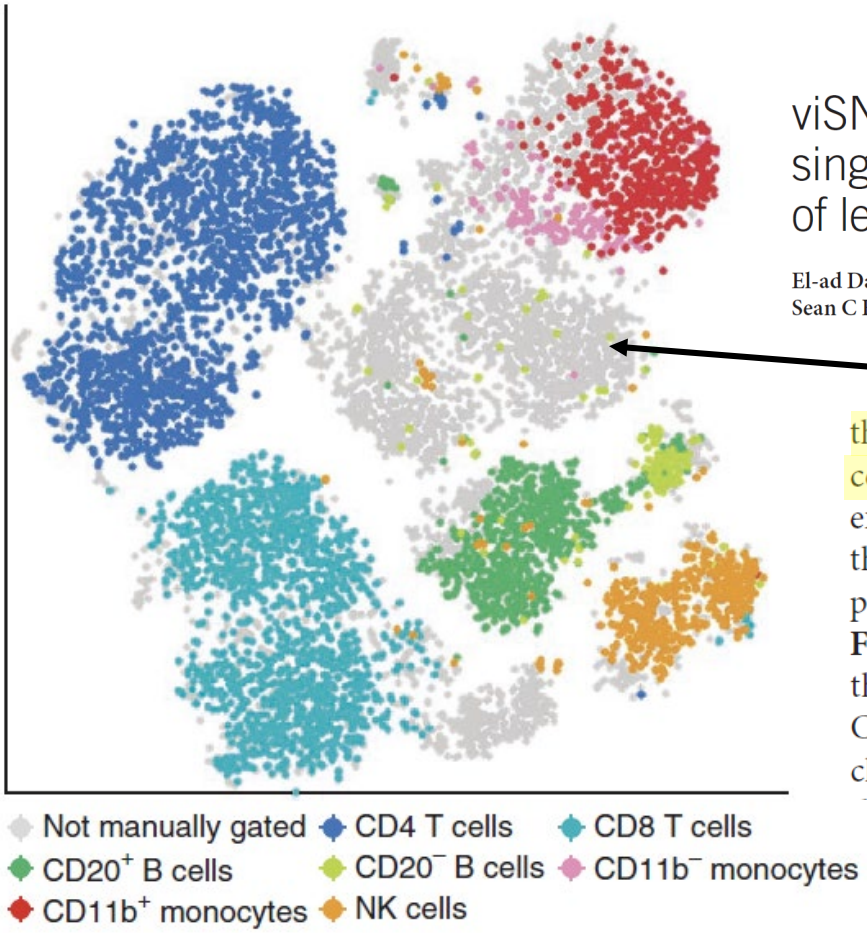
PRESS PLAY ON t-SNE chunk in RMD

Animation created by Cytobank team from iterations of viSNE / t-SNE using Healthy PBMC (26 features)

t-SNE Analysis Allows 2D Visualization of High Dimensional Single Cell Data



t-SNE can Help to Identify Cells Otherwise Lost by Expert Identification



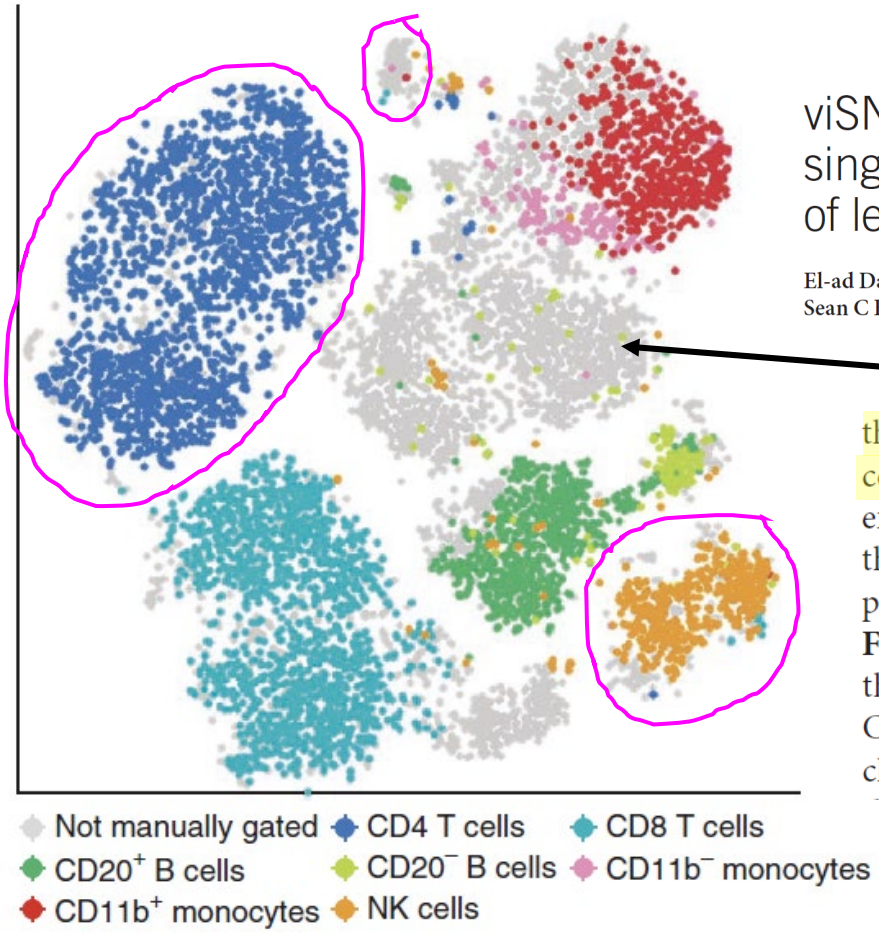
viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia

El-ad David Amir[1], Kara L Davis[2,3], Michelle D Tadmor[1,3], Erin F Simonds[2,3], Jacob H Levine[1,3], Sean C Bendall[2,3], Daniel K Shenfeld[1,3], Smita Krishnaswamy[1], Garry P Nolan[2,4] & Dana Pe'er[1,4]

In all cases, the viSNE gate included cells that were not classified by the expert manually gated biaxial plots; these cells are labeled in gray in the viSNE map. Examination of the marker expression of these cells reveals that they are typically just beyond the threshold of one marker, but the viSNE classification is strongly supported based on the expression of all other markers. For example, in **Figure 1d**, wherein cells are colored for CD11b marker expression, the cells in the gated region express the canonical monocyte marker CD33 (**Supplementary Fig. 1b**). However, only 47% of these cells were classified as monocytes by the manual gating (**Fig. 1b**).

Amir et al., *Nature Biotechnology* 2013

Experts can use t-SNE Axes to Select Cells of Interest



viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia

El-ad David Amir[1], Kara L Davis[2,3], Michelle D Tadmor[1,3], Erin F Simonds[2,3], Jacob H Levine[1,3], Sean C Bendall[2,3], Daniel K Shenfeld[1,3], Smita Krishnaswamy[1], Garry P Nolan[2,4] & Dana Pe'er[1,4]

In all cases, the viSNE gate included cells that were not classified by the expert manually gated biaxial plots; these cells are labeled in gray in the viSNE map. Examination of the marker expression of these cells reveals that they are typically just beyond the threshold of one marker, but the viSNE classification is strongly supported based on the expression of all other markers. For example, in **Figure 1d**, wherein cells are colored for CD11b marker expression, the cells in the gated region express the canonical monocyte marker CD33 (**Supplementary Fig. 1b**). However, only 47% of these cells were classified as monocytes by the manual gating (**Fig. 1b**).

Amir et al., *Nature Biotechnology* 2013

01_PBMC_extended_workflow_example.rmd

Run t-SNE

This section will run a t-SNE analysis on the PBMC data with set parameters

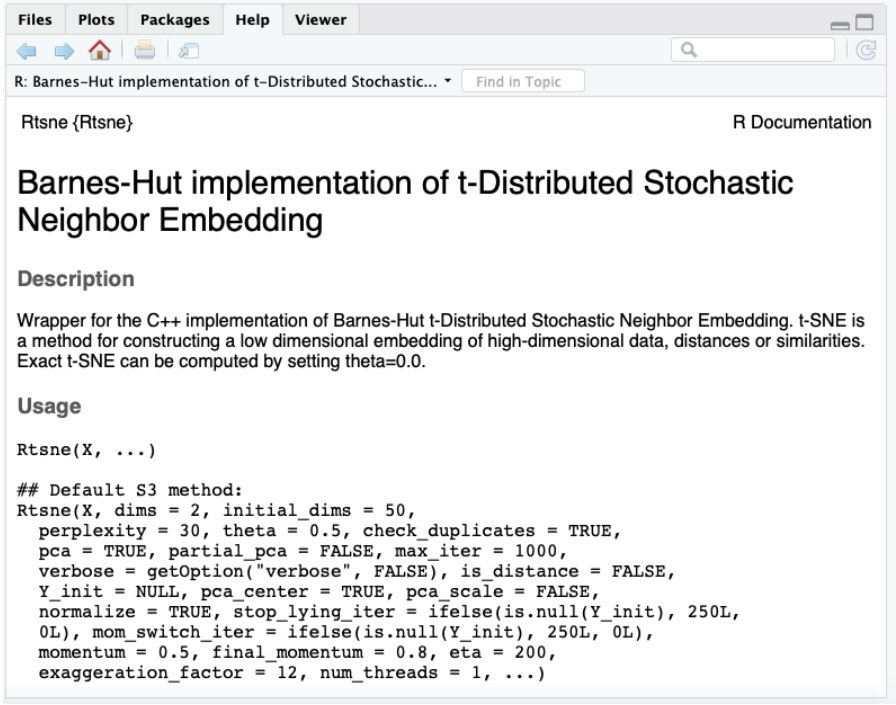
You can choose the resulting number of dimensions, the perplexity, and the iterations

This section will plot the t-SNE results. Two plots will appear, a “flat” dot plot and a density plot

```
72  ```{r run_t-SNE}
73  # Time ~5 min
74  set.seed(overall_seed)
75
76  # the line below will run t-SNE on the scaled surface markers (to see help page
77  # for t-SNE, type "?Rtsne -- enter" in console)
78
79  # you can view t-SNE progress by opening up the console below
80  mytSNE = Rtsne(
81    transformed.chosen.markers,                          # input scaled data
82
83    dims = 2,                                            # number of final
84                                                         # dimensions
85
86    initial_dims = length(transformed.chosen.markers), # number of initial
87                                                         # dimensions
88
89    perplexity = 30,           # perplexity (similar to # of nearest neighbors,
90                               # will scale with data sets, cannot be greater than
91                               # the number of events minus 1 divided by 3)
92    check_duplicates = FALSE,
93    max_iter = 1000,           # number of iterations
94    verbose = TRUE
95  )
96  tsne.data = as.data.frame(mytSNE$Y)
97  ```
98
99  ```{r plot_t-SNE}
100 # Time <10 sec
101
102 # setting aspect ratio for plots
103 range <- apply(apply(tsne.data, 2, range), 2, diff)
104 graphical.ratio.tsne <- (range[1] / range[2])
105
106 # t-SNE flat dot plot and density dot plot (1 dot = 1 cell)
107 tsne.plot <- data.frame(x = tsne.data[, 1], y = tsne.data[, 2])
108
```

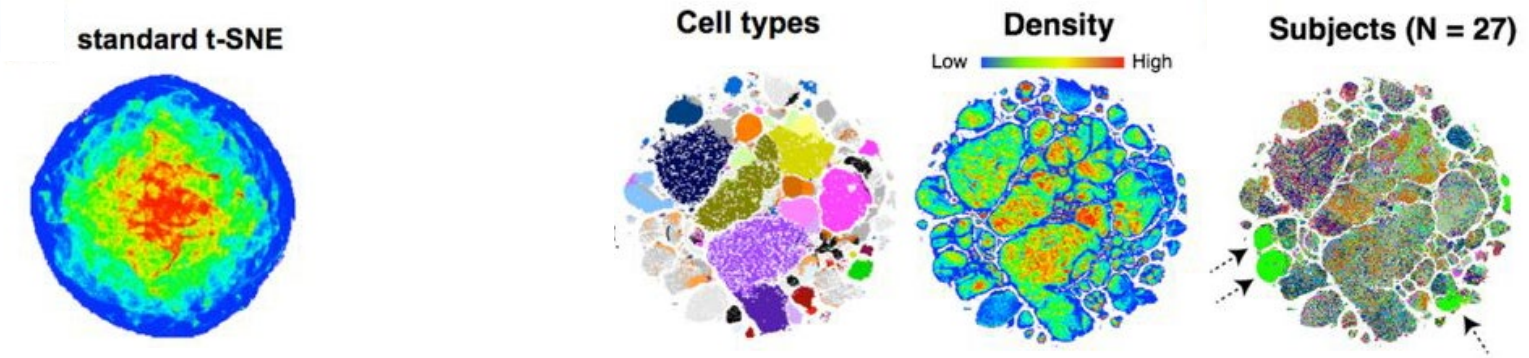
For help pages for tools type…

?Rtsne → enter
?umap → enter
?FlowSOM → enter
?MEM → enter



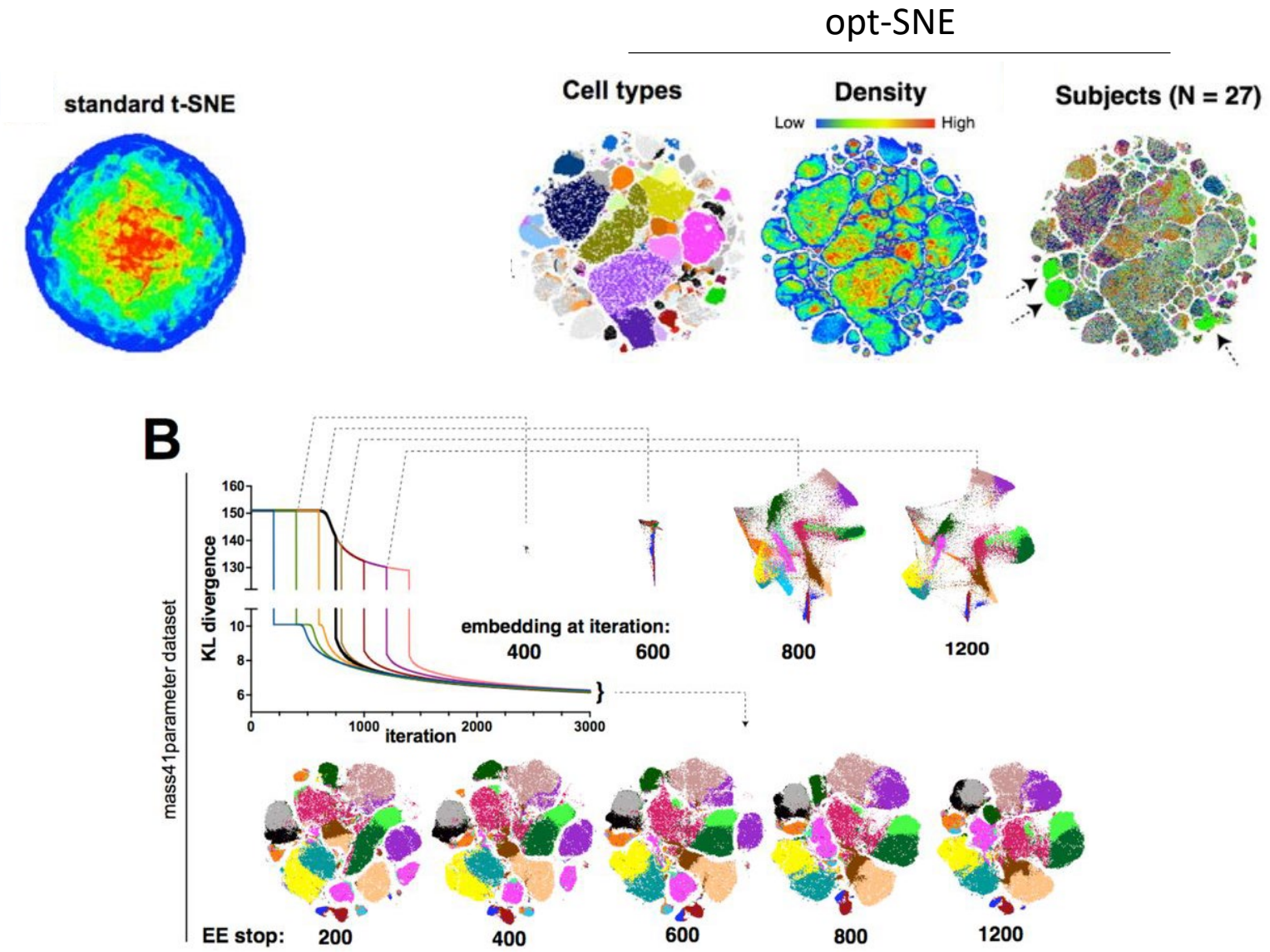
…in console to the right of >

opt-SNE is an Implementation of t-SNE for Large Data Sets



Belkina et al., *BioRxiv* , 2019

opt-SNE is an Implementation of t-SNE for Large Data Sets



Belkina et al., *BioRxiv* , 2019

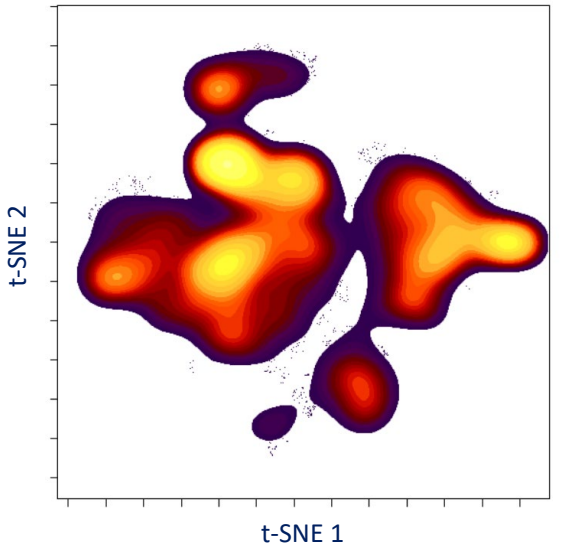
UMAP (Uniform Manifold Approximation and Projection) is Another Dimensionality Reduction Tool

Superior run times

Emphasis on both global and local structure in the data

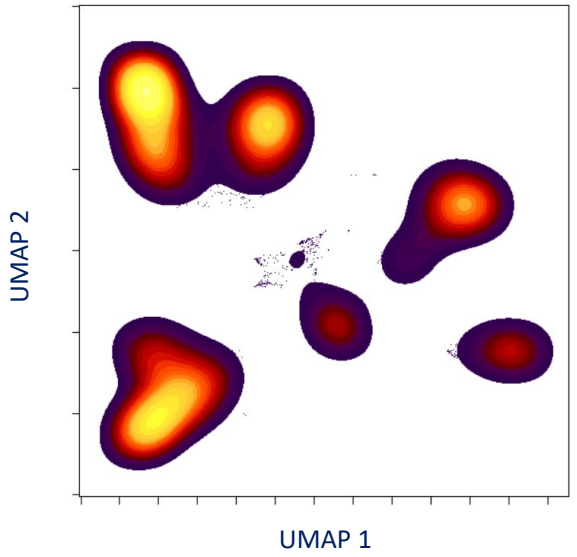
Ability to map new data onto the low-dimensional projection

t-SNE



VS.

UMAP



UMAP: McInnes et al., *arXiv* 2018; Becht et al., *Nature Biotech* 2019

Data used for t-SNE and UMAP: human PBMC

UMAP Preserves Local and Global Structure

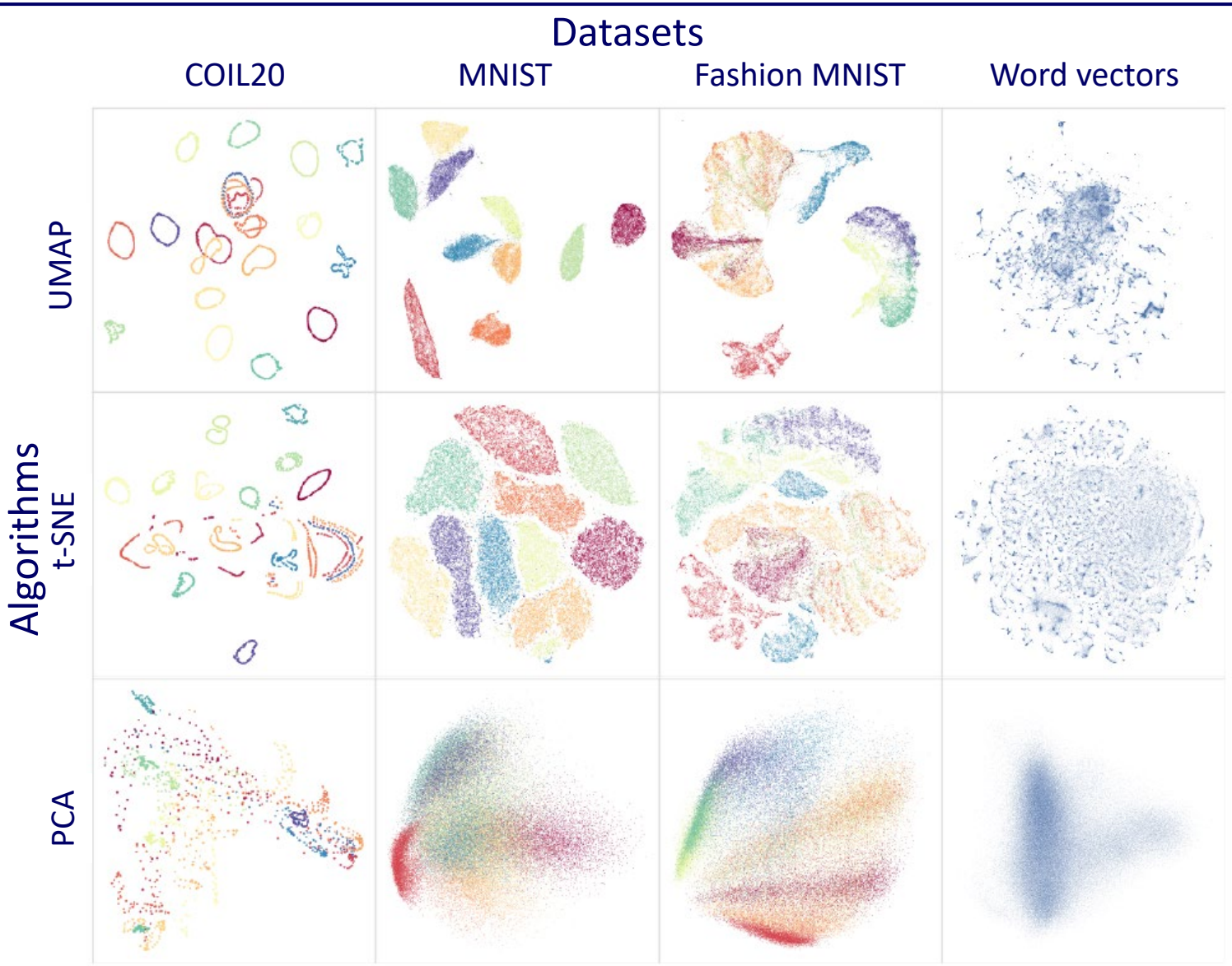


Figure 2: A comparison of several dimension reduction algorithms. UMAP reflects much of the large scale global structure, while also preserving the local fine structure similar to t-SNE.

McInnes et al., *arXiv* 2018

01_PBMC_extended_workflow_example.rmd

Run UMAP

This section will run a UMAP analysis on the PBMC data using set parameters

```r
# Time ~1 min
set.seed(overall_seed)
# Run UMAP on all scaled surface markers

# the line below will run UMAP on the data set (to see help page for UMAP, type
# "?UMAP -- enter" in console)

# you can view UMAP progress by opening up the console below
myumap <-
  umap(transformed.chosen.markers,  # input scaled data

       n_neighbors = 15,            # number of nearest neighbors to look at,
                                    # scales with data set

       n_threads = 1,               # this makes UMAP reproducible
       verbose = TRUE)
umap.data = as.data.frame(myumap)
```

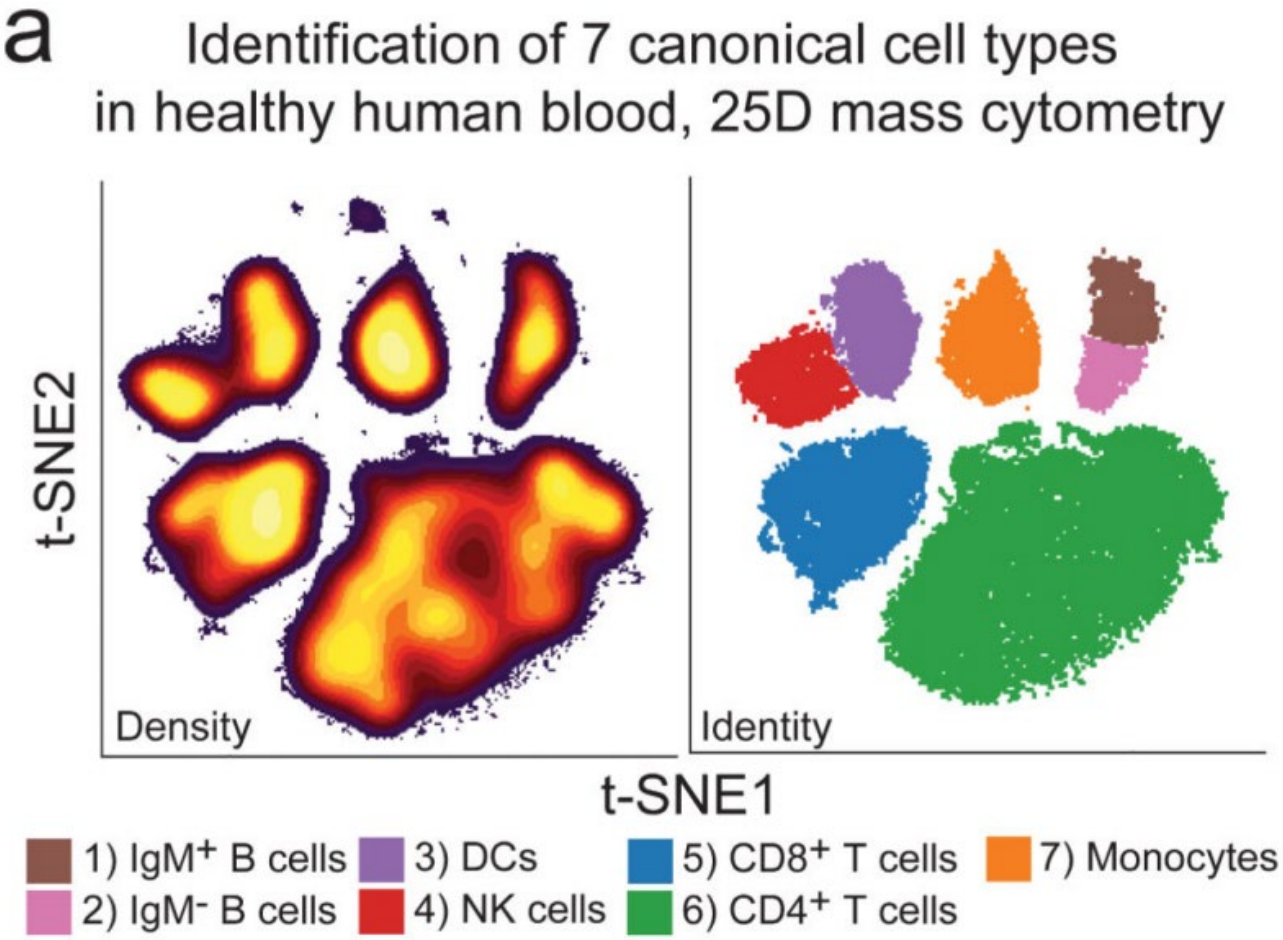
This section will plot the UMAP results. Two plots will appear, a “flat” dot plot and a density plot

```r
# Time <10 sec

# setting aspect ratio for plots
range <- apply(apply(umap.data, 2, range), 2, diff)
graphical.ratio.umap <- (range[1] / range[2])

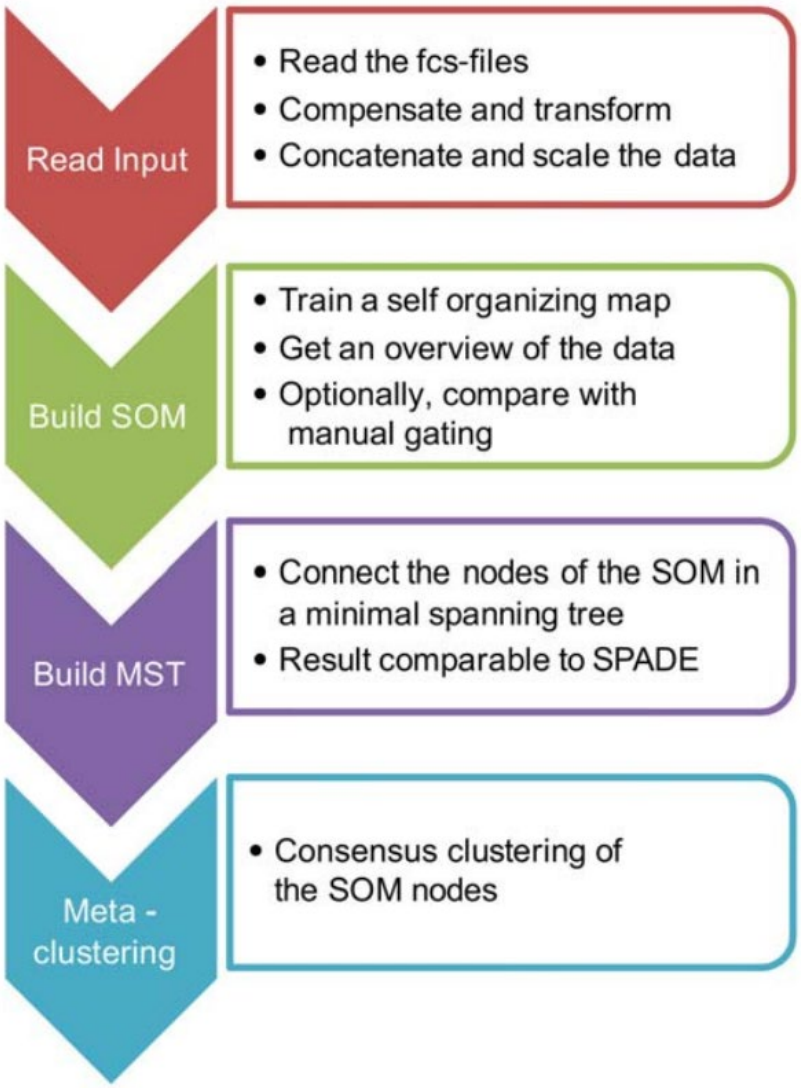
# UMAP flat dot plot and density dot plot (1 dot = 1 cell)
UMAP.plot <- data.frame(x = umap.data[, 1], y = umap.data[, 2])
```

Clusters can be Identified Based on Dimensionality Reduction Results



Diggins et al., *Nature Methods* 2017

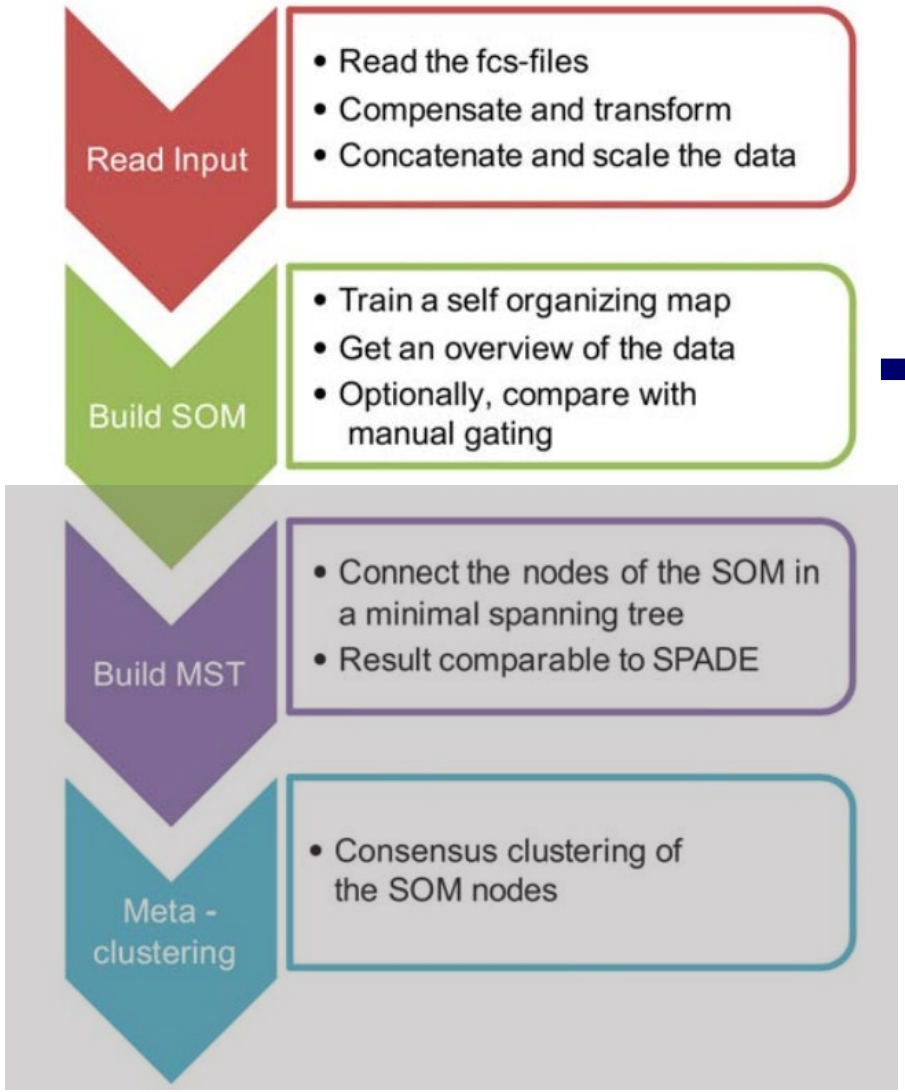
Clustering with FlowSOM: Self-organizing Maps



Van Gassen et al., *Cytometry A* 2015

See also, FlowSOM does well in a comparison of clustering tools: Weber & Robinson, *Cytometry A* 2017

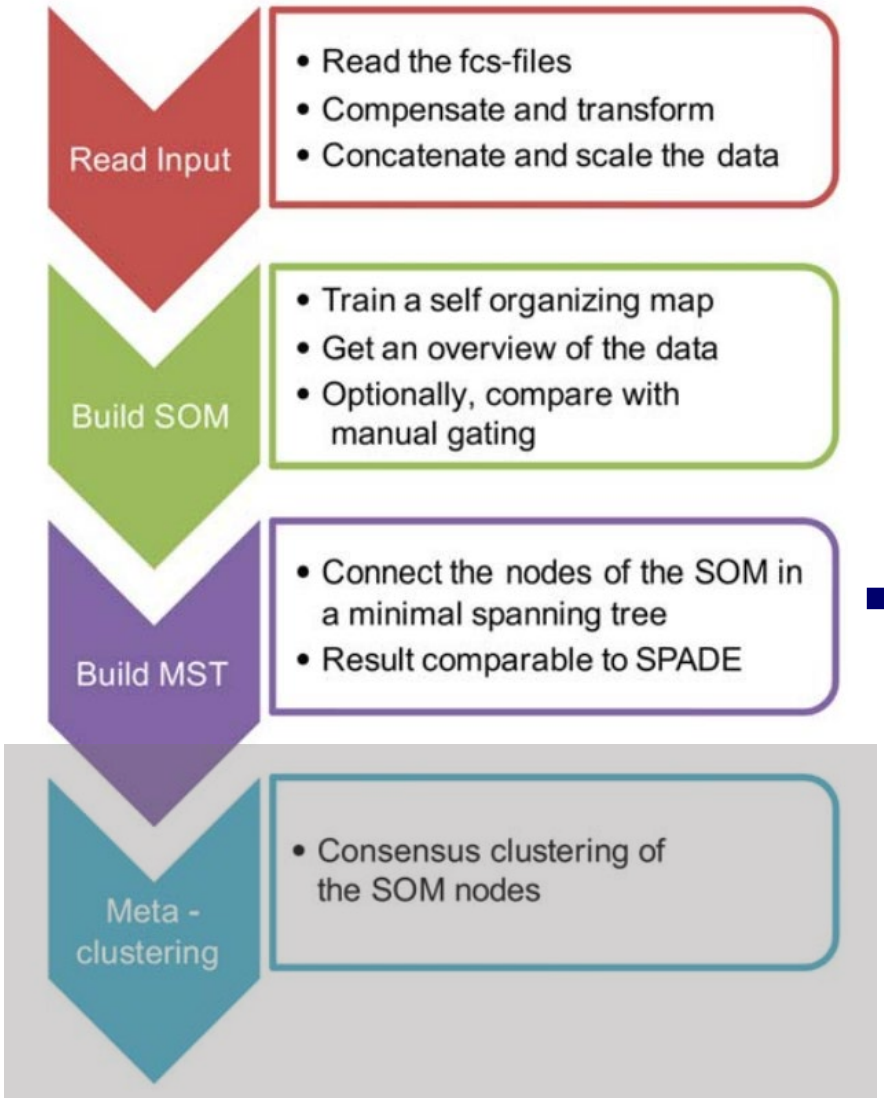
Clustering with FlowSOM: Self-organizing Maps



A SOM is created by assigning data points/cells to nodes based on their multidimensional phenotypes, updating this map repeatedly until each cell is assigned to a node with the most similar cells

Van Gassen et al., *Cytometry A* 2015
See also, FlowSOM does well in a comparison of clustering tools: Weber & Robinson, *Cytometry A* 2017

Clustering with FlowSOM: Self-organizing Maps



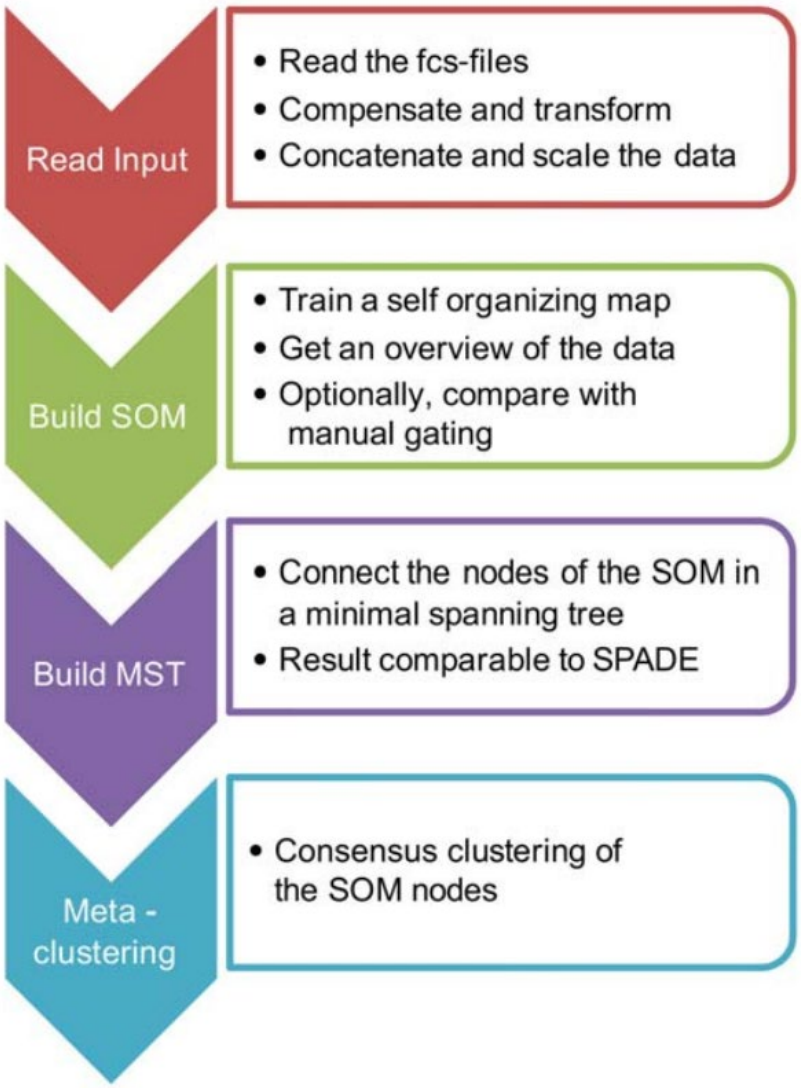
The next step is to arrange the nodes along a minimal spanning tree (MST), so that nodes that are most similar are closest on the tree

not used in our visualization

Van Gassen et al., *Cytometry A* 2015

See also, FlowSOM does well in a comparison of clustering tools: Weber & Robinson, *Cytometry A* 2017

Clustering with FlowSOM: Self-organizing Maps



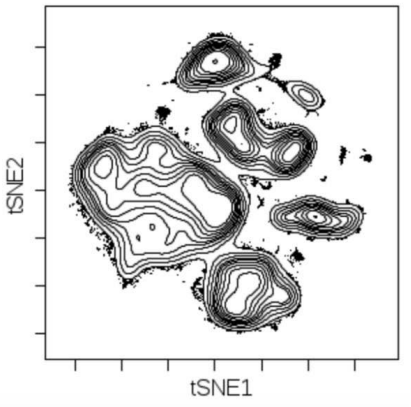
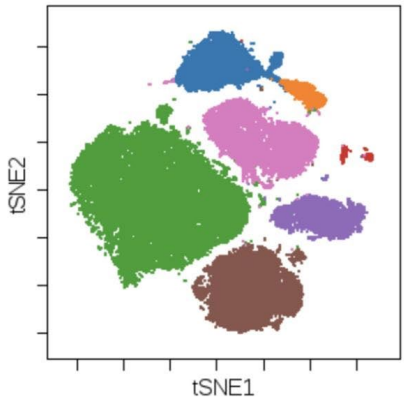
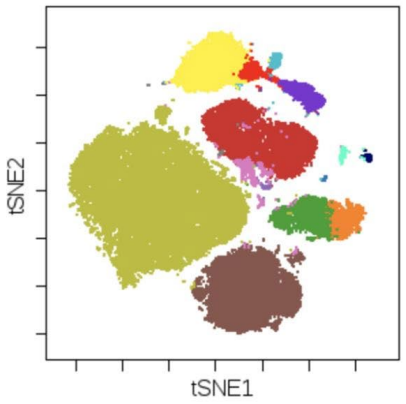
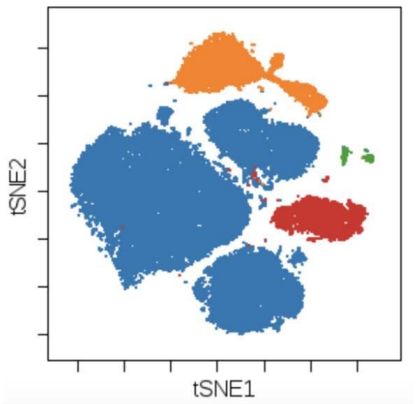
Finally, similar nodes are combined based on the number of desired clusters defined by the user. This desired number can be based on prior knowledge or a specific goal (i.e. minimizing intracluster variance)

Van Gassen et al., *Cytometry A* 2015

See also, FlowSOM does well in a comparison of clustering tools: Weber & Robinson, *Cytometry A* 2017

01_PBMC_extended_workflow_example.rmd

Run FlowSOM

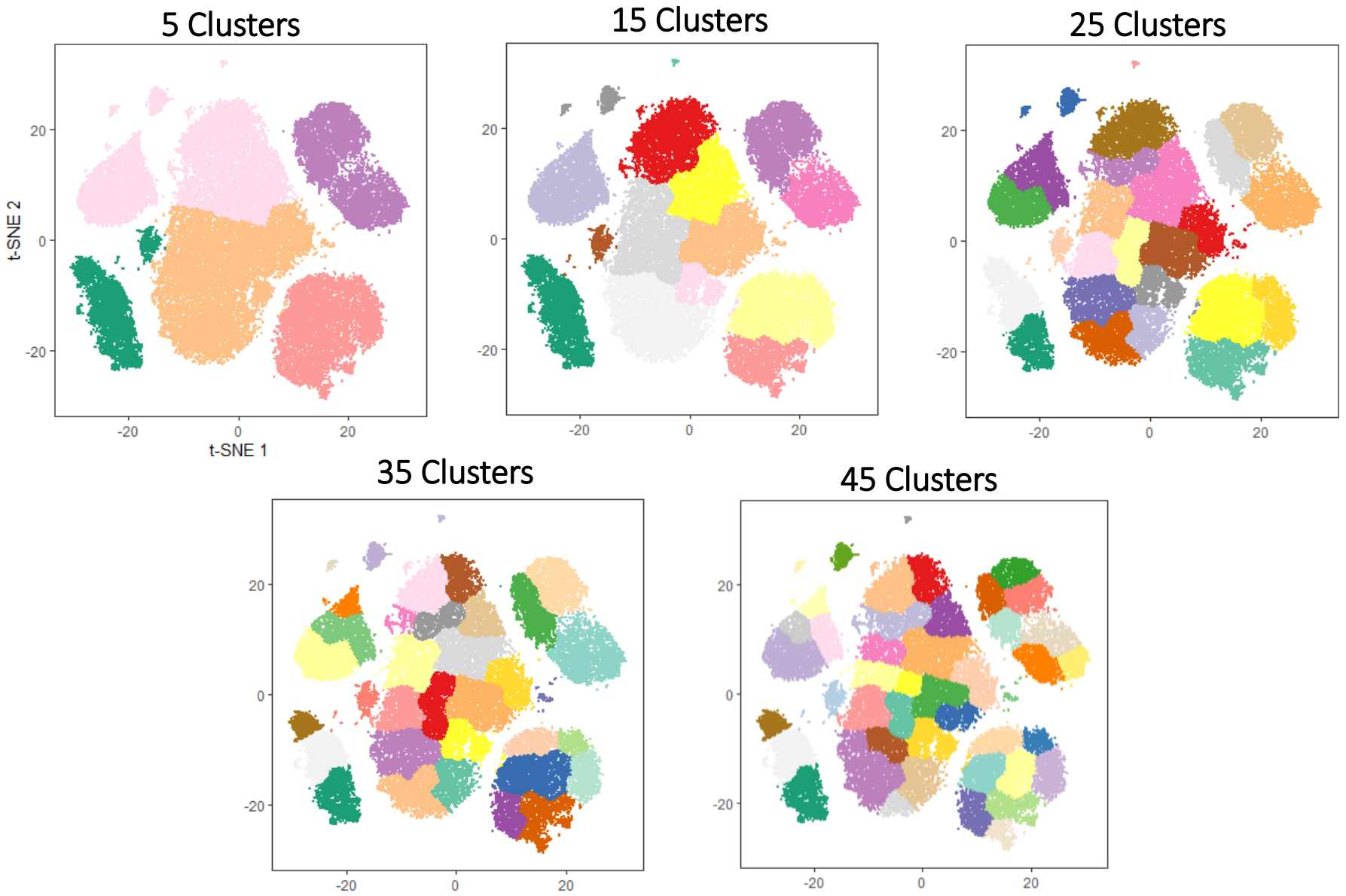
```
```{r run_FlowSOM_on_t-SNE}
# Time <10 sec

# create flowFrame for FlowSOM input (using t-SNE axes as input)
matrix <- as.matrix(tsne.data)
metadata <-
  data.frame(name = dimnames(matrix)[[2]],
             desc = dimnames(matrix)[[2]])
metadata$range <- apply(apply(matrix, 2, range), 2, diff)
metadata$minRange <- apply(matrix, 2, min)
metadata$maxRange <- apply(matrix, 2, max)
flowframe <- new("flowFrame",
                      exprs = matrix,
                      parameters = AnnotatedDataFrame(metadata))

# implement the FlowSOM on the data by running the line below (to see help page
# for FlowSOM, type "?FlowSOM --> enter" in console)
fsom <-
  FlowSOM(
    flowframe,      # input flowframe

    colsToUse = c(1:2),  # columns to use

    nClus = 10,          # target number of clusters (this can

    seed = overall_seed  # set seed for reproducibility
  )
FlowSOM.clusters.tsne <-
  as.matrix(fsom[[2]][fsom[[1]]$map$mapping[, 1]])
```
```

This section performs FlowSOM clustering on the t-SNE results

You can choose the parameters the clustering is performed on (t-SNE axes vs. measured markers) as well as a seed and desired number of clusters

01_PBMC_extended_workflow_example.rmd

Plot FlowSOM

```
211 ```{r plot_tsne_flowSOM_clusters}
212 # Time <10 sec
213 qual_col_pals = brewer.pal.info[brewer.pal.info$category
214 col_vector = unlist(mapply(brewer.pal, qual_col_pals$maxc
215                            rownames(qual_col_pals)))
216
217 # plot FlowSOM clusters on t-SNE axes
218 ggplot(tsne.plot) + coord_fixed(ratio=graphical.ratio.tsn
219   geom_point(aes(x=x, y=y, color=FlowSOM.clusters.tsne), cex = 0.5) +
220   labs(x = "t-SNE 1", y = "t-SNE 2",title = "FlowSOM Clustering on t-SNE Axes",
221       color = "Cluster") + theme_bw() +
222   guides(colour = guide_legend(override.aes = list(size=4))) +
223   scale_color_manual(values = sample(col_vector)) +
224   labs(caption = "Data from Diggins et al., Nat Methods 2017, 14: 275-278 \nFlow
    Repository: FR-FCM-ZY63") +
225   theme(panel.grid.major = element_blank(),
226             panel.grid.minor = element_blank())
227 ```
```

This section plots the identified clusters back onto the t-SNE axes and generates a plot (a colored version of the t-SNE plot from before)

Clustering with FlowSOM: Self-organizing Maps

Contour plot of viSNE map



FlowSOM metaclusters overlaid on viSNE map



 # metaclusters = 7



 # metaclusters = 15



 # metaclusters = 4

01_PBMC_extended_workflow_example.rmd

Run FlowSOM

```
```{r run_FlowSOM_varying_cluster_number}
# Time ~ 1-2 min
for (i in seq(5,45,by = 10)){

  matrix <- as.matrix(tsne.data)
metadata <-
  data.frame(name = dimnames(matrix)[[2]],
             desc = dimnames(matrix)[[2]])
metadata$range <- apply(apply(matrix, 2, range), 2, diff)
metadata$minRange <- apply(matrix, 2, min)
metadata$maxRange <- apply(matrix, 2, max)
flowframe <- new("flowFrame",
                      exprs = matrix,
                      parameters = AnnotatedDataFrame(metadata))
fsom <-
  FlowSOM(
    flowframe,
    colsToUse = c(1:2),
    nClus = i,
    seed = overall_seed
  )
FlowSOM.clusters.vary <-
  as.matrix(fsom[[2]][fsom[[1]]$map$mapping[, 1]])

legend.col = round(max(as.numeric(as.vector(FlowSOM.clusters.vary)))/3)
print(ggplot(tsne.plot) + coord_fixed(ratio=graphical.ratio.tsne) +
  geom_point(aes(x=x, y=y, color=FlowSOM.clusters.vary), cex = 0.5) +
  labs(x = "t-SNE 1", y = "t-SNE 2",title = "FlowSOM Clustering on t-SNE Axes",
       color = "Cluster") + theme_bw() +
  guides(colour = guide_legend(override.aes = list(size=4),
                               nrow = legend.col)) +
    scale_color_manual(values = sample(col_vector)) +
  labs(caption = "Data from Diggins et al., Nat Methods 2017, 14: 275-278 \nFlow
Repository: FR-FCM-ZY63") +
  theme(panel.grid.major = element_blank(),
            panel.grid.minor = element_blank()))}
```
```

This section runs multiple FlowSOM analyses, changing the number of clusters from 5 to 45 in increments of 10

FlowSOM Requires that Users Choose a Number of Clusters



Data from Diggins et al., *Nature Methods*, 2017

01_PBMC_extended_workflow_example.rmd

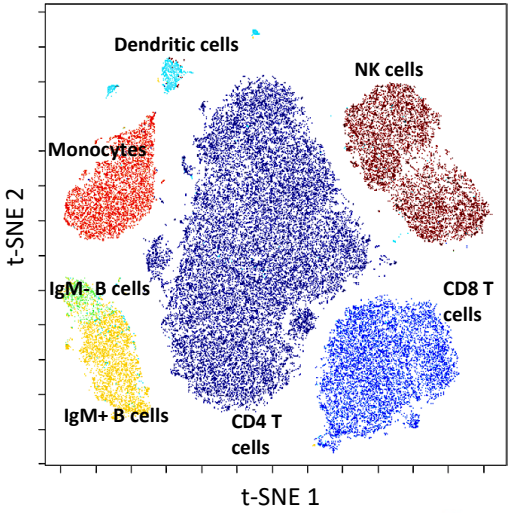
Run FlowSOM

```
266 ```{r run_FlowSOM_on_original markers}
267 # Time ~ 1 min
268
269 matrix <- as.matrix(transformed.chosen.markers)
270 metadata <-
271   data.frame(name = dimnames(matrix)[[2]],
272              desc = dimnames(matrix)[[2]])
273 metadata$range <- apply(apply(matrix, 2, range), 2, diff)
274 metadata$minRange <- apply(matrix, 2, min)
275 metadata$maxRange <- apply(matrix, 2, max)
276 flowframe <- new("flowFrame",
277                  exprs = matrix,
278                  parameters = AnnotatedDataFrame(metadata))
279
280 fsom <-
281   FlowSOM(
282     flowframe,
283     colsToUse = c(1:ncol(transformed.chosen.markers)),
284     nClus = 10,
285     seed = overall_seed
286   )
287 FlowSOM.clusters.OG <-
288   as.matrix(fsom[[2]][fsom[[1]]$map$mapping[, 1]])
289
290 ggplot(tsne.plot) + coord_fixed(ratio=graphical.ratio.tsne) +
291   geom_point(aes(x=x, y=y, color=FlowSOM.clusters.OG), cex = 0.3) +
292   labs(x = "t-SNE 1", y = "t-SNE 2",
293        title = "FlowSOM Clustering on Original Markers", color = "Cluster") +
294   theme_bw() + scale_color_manual(values = sample(col_vector)) +
295   guides(colour = guide_legend(override.aes = list(size=4))) +
296   labs(caption = "Data from Diggins et al., Nat Methods 2017, 14: 275-278 \nFlow
    Repository: FR-FCM-ZY63") +
297   theme(panel.grid.major = element_blank(),
298             panel.grid.minor = element_blank())
299 ```
```

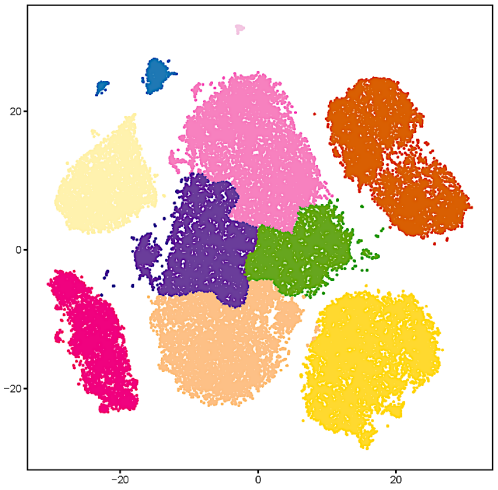
This section performs FlowSOM clustering on the protein markers in the cyTOF experiment

FlowSOM Clusters are Dependent on Input Parameters

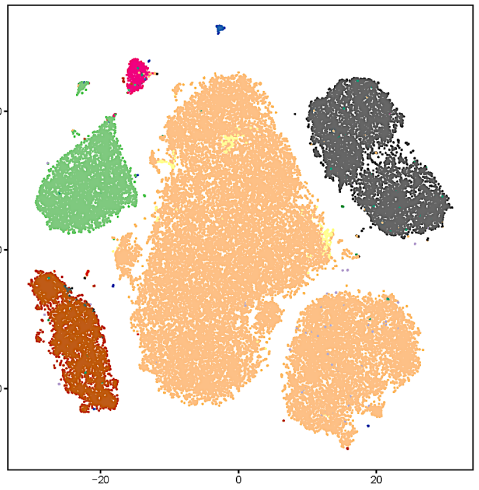
Major Populations Overlaid on t-SNE Axes

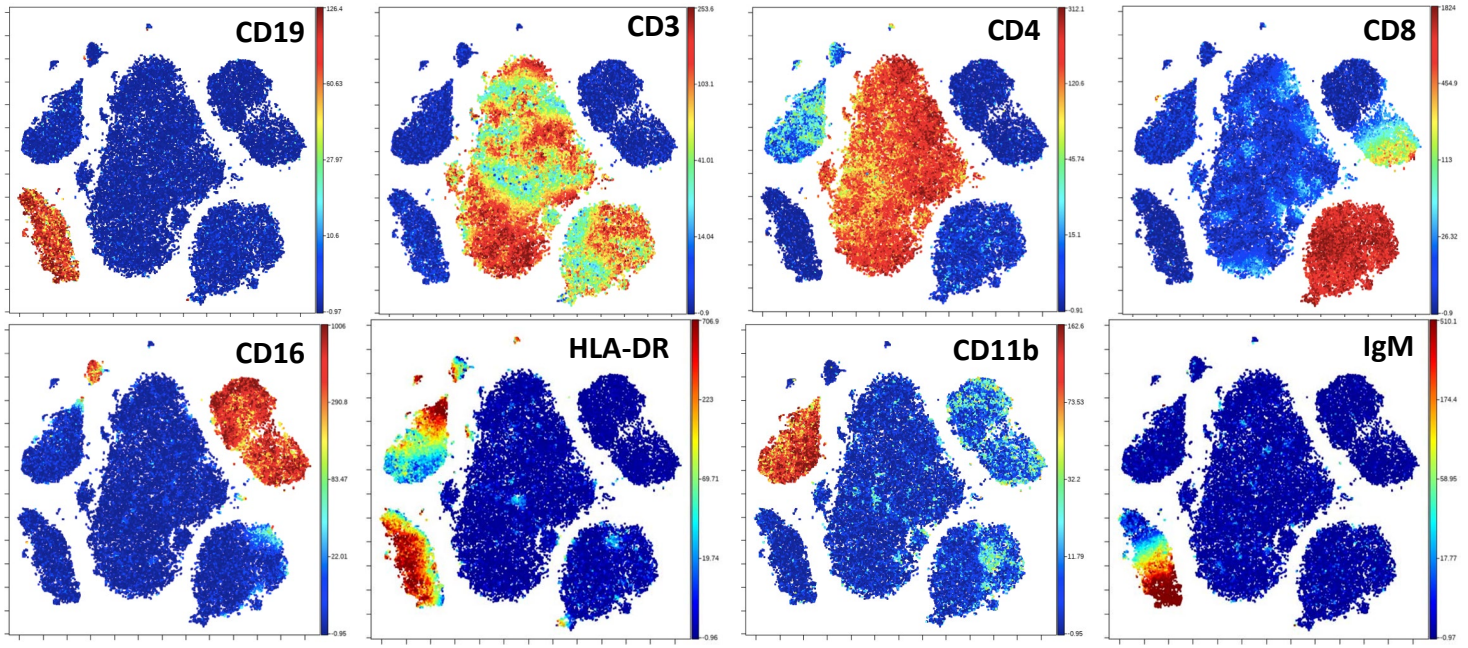


FlowSOM on t-SNE Axes (n = 10)



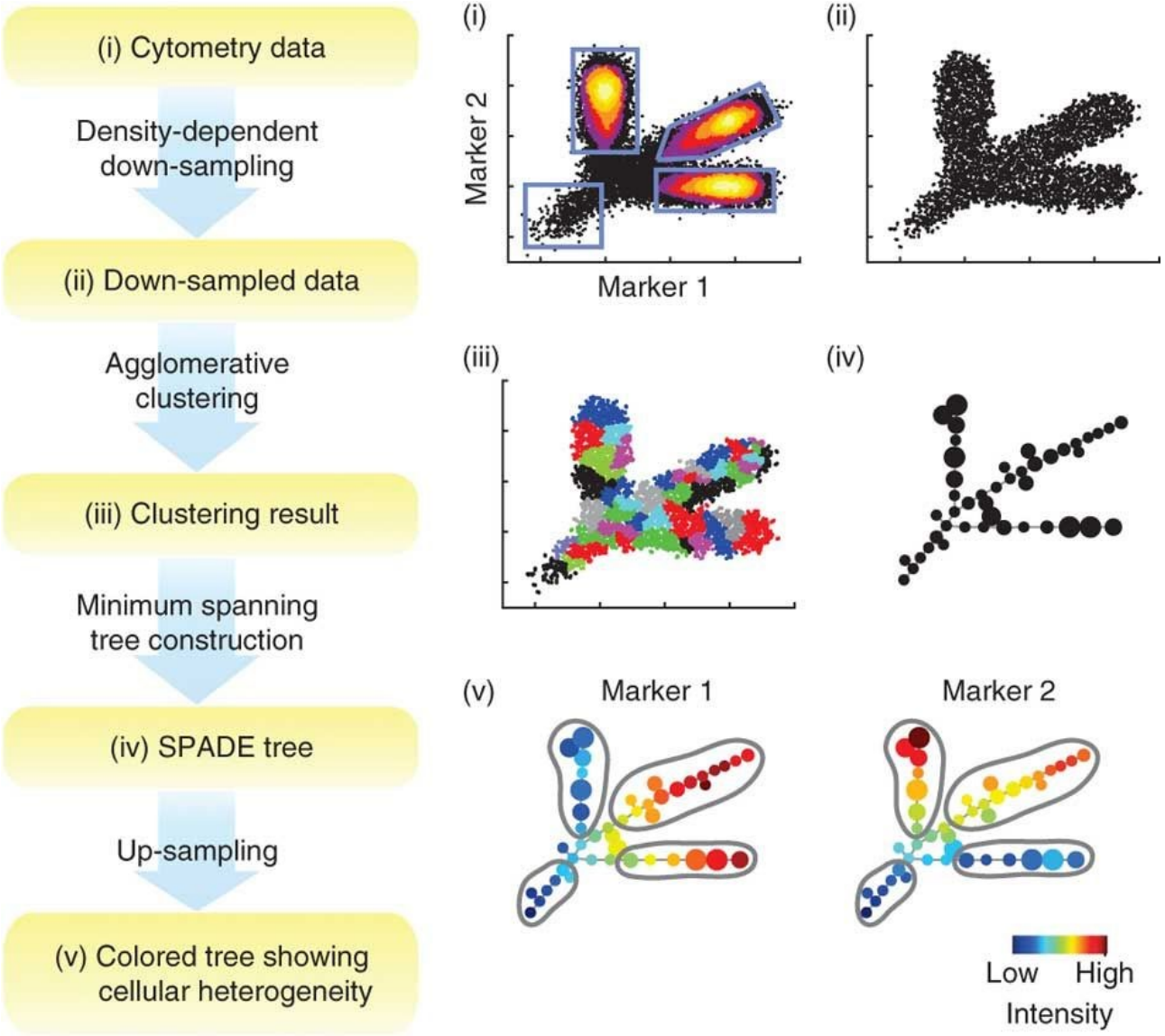
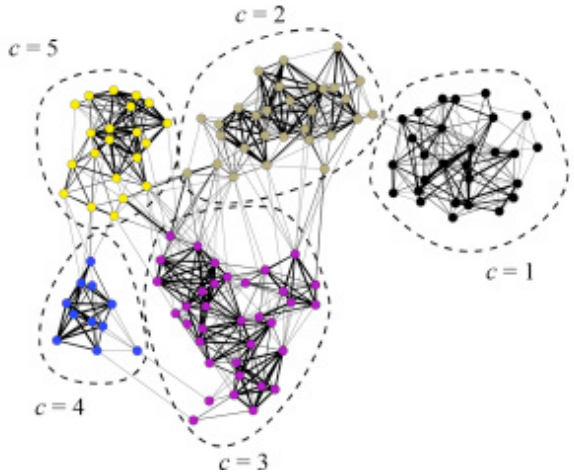
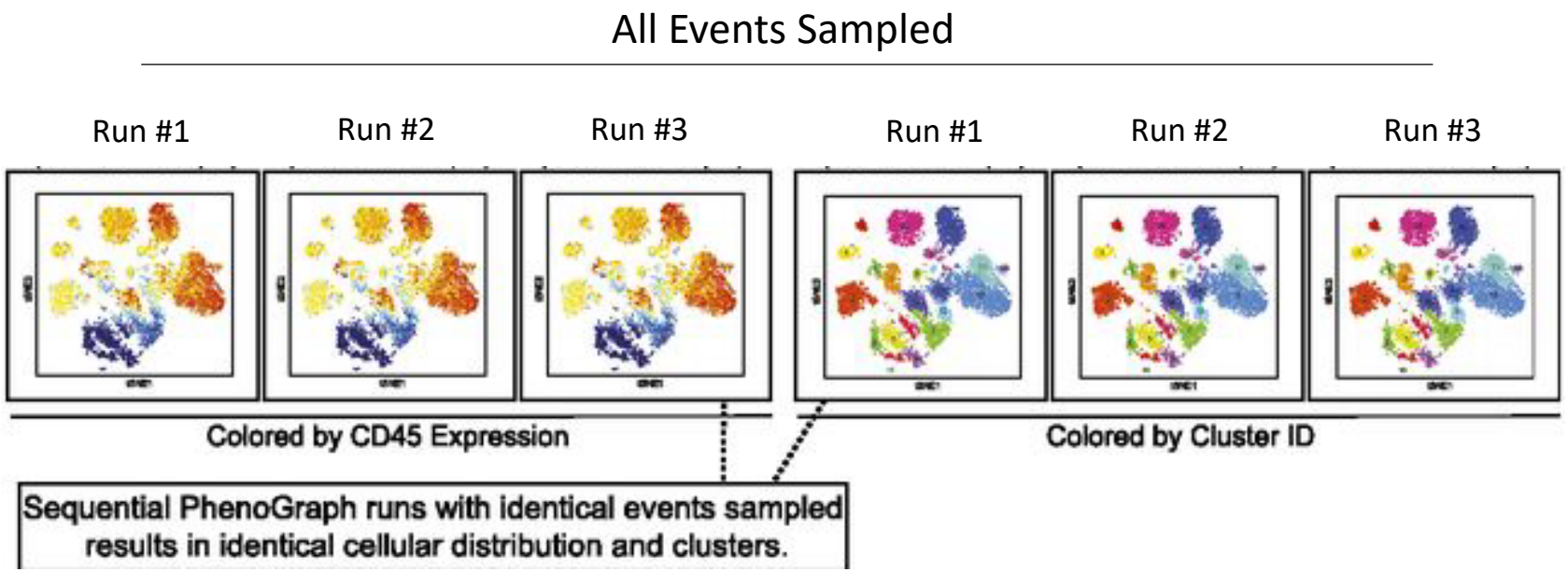
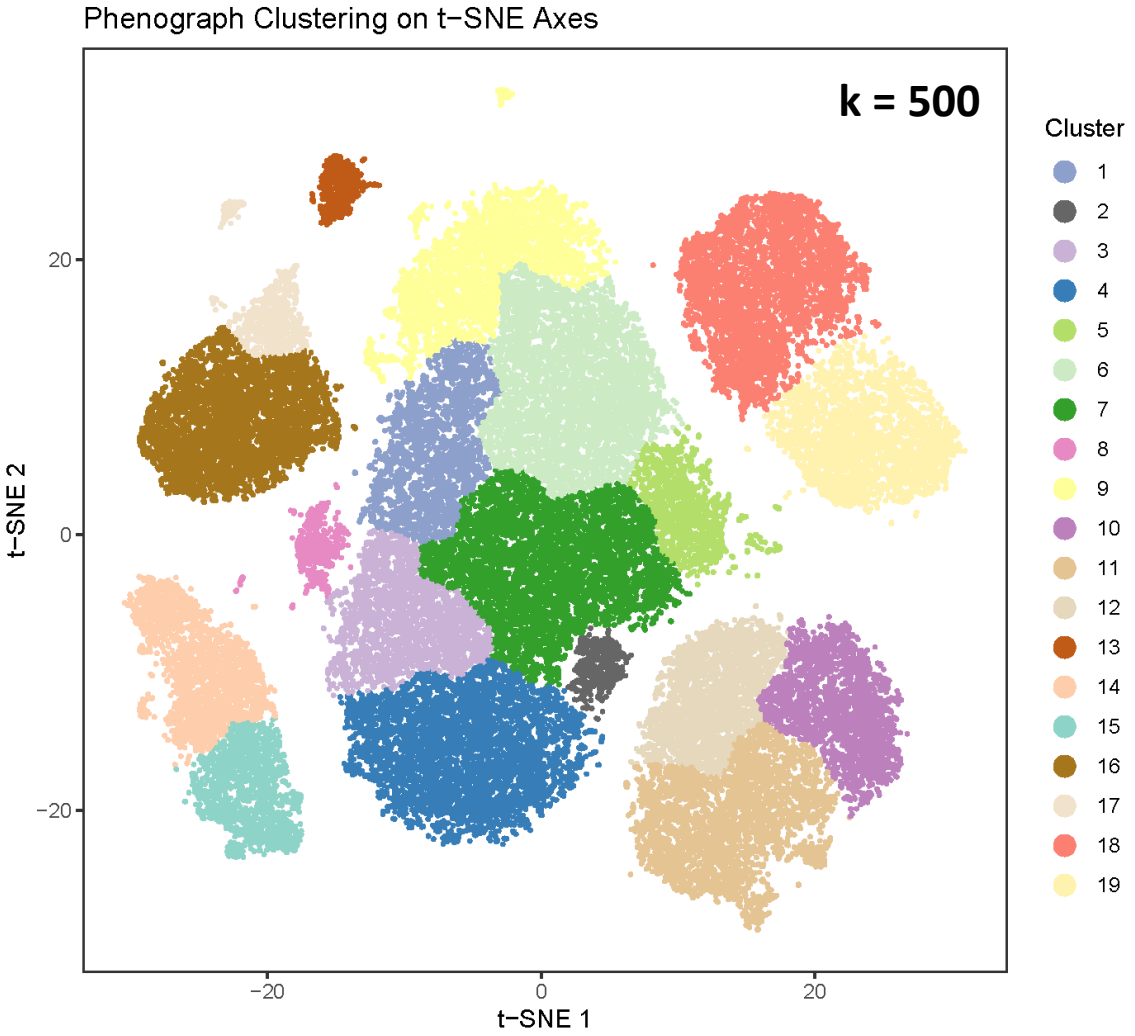
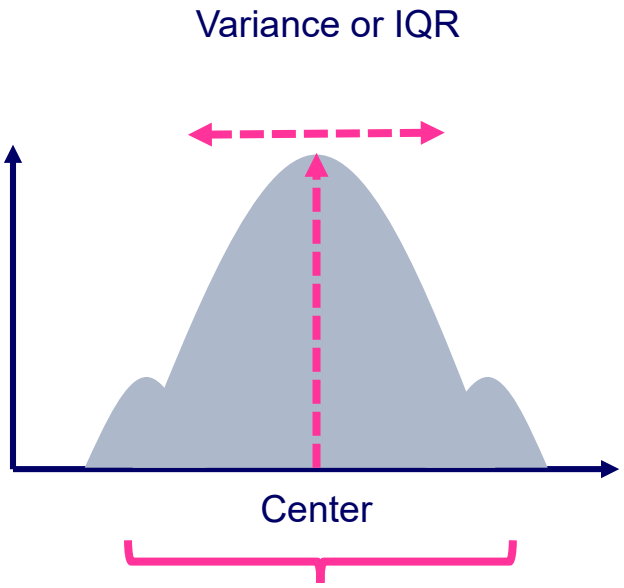
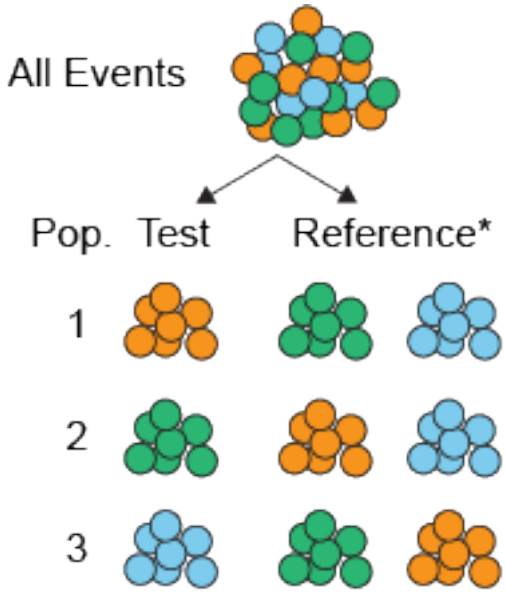
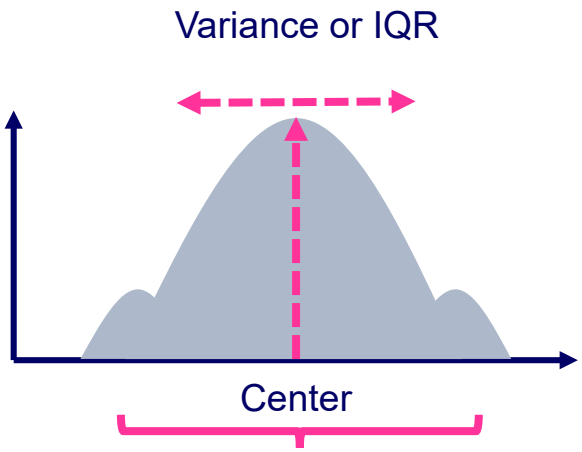
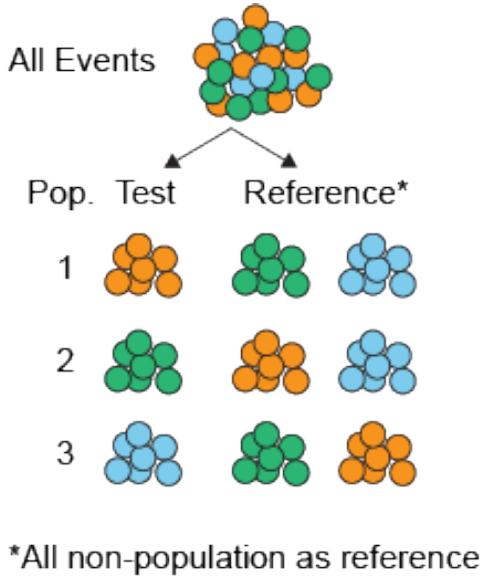
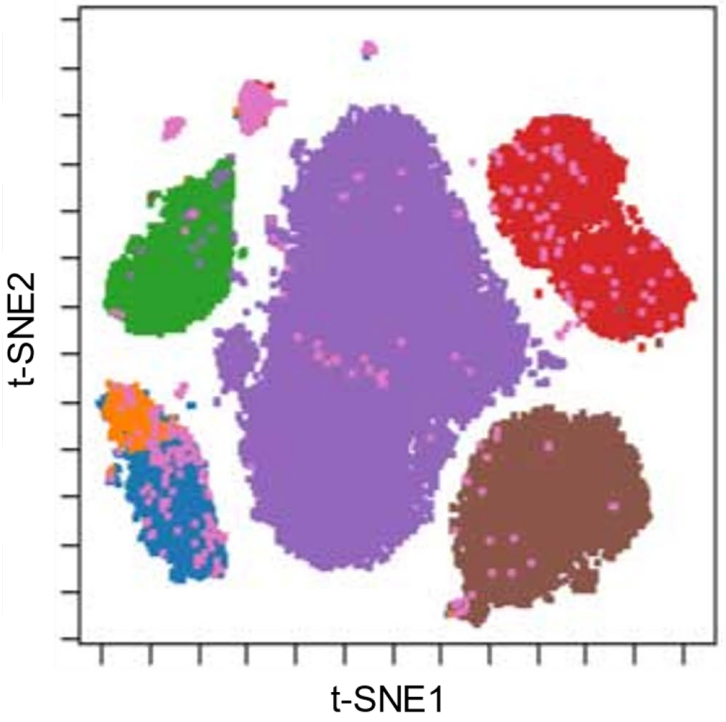
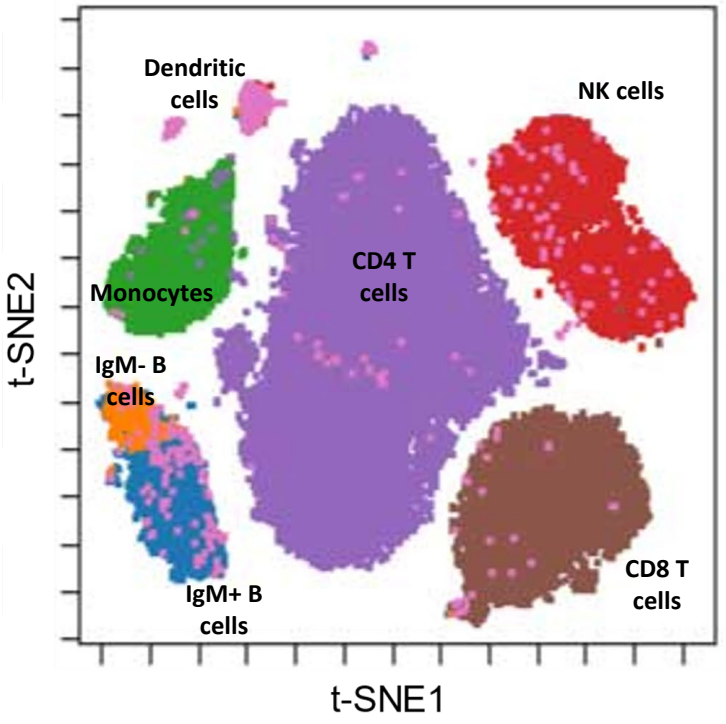
FlowSOM on Original Markers (n = 10)





01_PBMC_extended_workflow_example.rmd

Run FlowSOM

```r
```{r run_FlowSOM_on_UMAP}
# Time <10 sec

matrix <- as.matrix(umap.data)
metadata <-
  data.frame(name = dimnames(matrix)[[2]],
             desc = dimnames(matrix)[[2]])
metadata$range <- apply(apply(matrix, 2, range), 2, diff)
metadata$minRange <- apply(matrix, 2, min)
metadata$maxRange <- apply(matrix, 2, max)
flowframe <- new("flowFrame",
                      exprs = matrix,
                      parameters = AnnotatedDataFrame(metadata))
fsom <-
  FlowSOM(
    flowframe,
    colsToUse = c(1:2),
    nClus = 10,
    seed = overall_seed
  )
FlowSOM.clusters.umap <-
  as.matrix(fsom[[2]][fsom[[1]]$map$mapping[, 1]])

ggplot(UMAP.plot) + coord_fixed(ratio=graphical.ratio.umap) +
  geom_point(aes(x=x, y=y, color=FlowSOM.clusters.umap), cex = 0.5) +
  labs(x = "UMAP 1", y = "UMAP 2",title = "FlowSOM Clustering on UMAP Axes",
       color = "Cluster") + theme_bw() +
  guides(colour = guide_legend(override.aes = list(size=4)))+
  scale_color_manual(values = sample(col_vector))+
  labs(caption = "Data from Diggins et al., Nat Methods 2017, 14: 275-278 \nFlow Repository: FR-FCM-ZY63") +
  theme(panel.grid.major = element_blank(),
              panel.grid.minor = element_blank())
```
```

This section performs FlowSOM clustering on the UMAP results

Spanning-Tree Progression Analysis of Density-Normalized Events (SPADE) is an Alternative Clustering Tool



Qui et al., *Nature Biotechnology* 2011

Phenograph Uses K Nearest Neighbors to Build a Weighted Graph and Assign Cells to Clusters

Build Graphs

- Build a single-cell graph for each sample or dataset
  - Each cell is a node and is connected by edges to nearest neighbors

Cluster

- Partition each graph into distinct subpopulations
  - Based on density/interconnected nodes

Define

- Extract surface and signaling features for each subpopulation



Levine et al., *Cell,* 2015

Phenograph is Deterministic



Kimball et al., *J Immunology,* 2018

K Nearest Neighbor Settings Determine the Number of Clusters Identified by Phenograph



Data from Diggins et al., *Nature Methods*, 2017

Marker Enrichment Modeling Analysis Identifies Markers that are Specifically Expressed or Lacking on Populations

MEM accounts for variance and median of markers to identify enriched features on subsets of cells



Shape (skewness, symmetry
peaks, outliers, etc.)



*All non-population as reference

Adapted from Kirsten Diggins

MEM Quantifies Relative Enrichment by Combining Magnitude and Interquartile Range





MEM label

▲HLADR^{+10} CD20^{+9} CD19^{+7} IgM^{+5} CD34^{+3} CD45RA^{+3} CXCR4^{+2} CD47^{+2} CD33^{+2}
▼CD7^{-2}

Linear transformation to -10 to +10

If $MAG_{test} - MAG_{ref} < 0$, MEM = -MEM

Adapted from Kirsten Diggins

MEM Quantifies Relative Enrichment by Combining Magnitude and Interquartile Range



▲CD33^{+5}CD11c $^{+5}$CD61 $^{+4}$ CD14^{+4}CD11b $^{+3}$CD44 $^{+3}$ HLA-DR $^{+3}$
▼CD3^{-7}CD4 $^{-5}$CD8 $^{-3}$

▲CD4^{+4} CD3^{+4}CD44 $^{+2}$
▼CD16^{-9}CD8 $^{-7}$CD11c $^{-5}$ HLA-DR $^{-5}$CD69 $^{-3}$CD11b $^{-2}$ CD20^{-2}

▲CD11c^{+4}HLA-DR $^{+4}$ CD123^{+2}CD16 $^{+2}$
▼CD3^{-8}CD4 $^{-3}$CD8 $^{-3}$

▲CD16+6
▼CD4^{-10}CD3 $^{-6}$CD44 $^{-2}$

▲CD8+7 CD3^{+3}CD69 $^{+2}$
▼CD4^{-10}CD38 $^{-2}$

▲HLA-DR $^{+5}$IgM $^{+4}$CD19 $^{+3}$ CD20^{+3}
▼CD4^{-10}CD3 $^{-8}$CD8 $^{-4}$

▲HLA-DR $^{+4}$CD19 $^{+3}$CD20 $^{+3}$
▼CD4^{-9}CD3 $^{-8}$CD8 $^{-3}$

MEM Quantifies Relative Enrichment by Combining Magnitude and Interquartile Range



▲CD33^{+5}CD11c $^{+5}$CD61 $^{+4}$ CD14^{+4}CD11b $^{+3}$CD44 $^{+3}$ HLA-DR $^{+3}$
▼CD3^{-7}CD4 $^{-5}$CD8 $^{-3}$

▲CD4^{+4} CD3^{+4}CD44 $^{+2}$
▼CD16^{-9}CD8 $^{-7}$CD11c $^{-5}$ HLA-DR $^{-5}$CD69 $^{-3}$CD11b $^{-2}$ CD20^{-2}

▲CD11c^{+4}HLA-DR $^{+4}$ CD123^{+2}CD16 $^{+2}$
▼CD3^{-8}CD4 $^{-3}$CD8 $^{-3}$

▲CD16+6
▼CD4^{-10}CD3 $^{-6}$CD44 $^{-2}$

▲CD8+7 CD3^{+3}CD69 $^{+2}$
▼CD4^{-10}CD38 $^{-2}$

▲HLA-DR $^{+5}$IgM $^{+4}$CD19 $^{+3}$ CD20^{+3}
▼CD4^{-10}CD3 $^{-8}$CD8 $^{-4}$

▲HLA-DR $^{+4}$CD19 $^{+3}$CD20 $^{+3}$
▼CD4^{-9}CD3 $^{-8}$CD8 $^{-3}$

01_PBMC_extended_workflow_example.rmd

Run MEM

````
```{r run_MEM_on_FlowSOM_on_t-SNE}
# Time ~30 sec

# Run MEM on the FlowSOM clusters found from using t-SNE axes
cluster = as.numeric(as.vector((FlowSOM.clusters.tsne)))
MEM.data = cbind(transformed.chosen.markers, cluster)

MEM.values.tf = MEM(
  MEM.data,              # input data (last column must contain cluster values)

  transform = FALSE,     # data is already scaled in this case
  cofactor = 1,
  choose.markers = FALSE,
  markers = "all",       # use all transformed, chosen markers from pre
                         # selection

  choose.ref = FALSE,    # reference will be all other cells
  zero.ref = FALSE,
  rename.markers = FALSE,
  new.marker.names = "CD19,CD117,CD11b,CD4,CD8,CD20,CD34,CD61,CD123,CD45RA,CD45,CD10,CD3
3,CD11c,CD14,CD69,CD15,CD16,CD44,CD38,CD25,CD3,IgM,HLA-DR,CD56", # rename channels for
labels
  file.is.clust = FALSE,
  add.fileID = FALSE,
  IQR.thresh = NULL
)

# build MEM heatmap and output enrichment scores
build.heatmaps(
  MEM.values.tf,               # input MEM values

  cluster.MEM = "both",        # dendrogram for columns and rows

  display.thresh = 2,          # display threshold for MEM scores
  newWindow.heatmaps = FALSE,
  output.files = TRUE,         # makes txt and PDF files for heatmap and MEM
                               # scores

  labels = TRUE,               # include labels in heatmap
  only.MEMheatmap = FALSE
)
```
````

This section performs MEM analysis on the FlowSOM clusters based on the t-SNE results

You can choose the markers for the MEM analysis as well as their names and the reference population

This section produces heatmaps and MEM (enrichment) scores

01_PBMC_extended_workflow_example.rmd

Run MEM

```r
```{r run_MEM_on_FlowSOM_on_OG}
# Time ~30 sec

cluster = as.numeric(as.vector((FlowSOM.clusters.OG)))
MEM.data = cbind(transformed.chosen.markers, cluster)

MEM.values.ogf = MEM(
  MEM.data,
  transform = FALSE,
  cofactor = 1,
  choose.markers = FALSE,
  markers = "all",
  choose.ref = FALSE,
  zero.ref = FALSE,
  rename.markers = FALSE,
  new.marker.names = "CD19,CD117,CD11b,CD4,CD8,CD20,CD34,CD61,CD123,CD45RA,CD45,CD10,CD33,CD11c,CD14,CD69,CD15,CD16,CD44,CD38,CD25,CD3,IgM,HLA-DR,CD56", # rename channels for labels
  file.is.clust = FALSE,
  add.fileID = FALSE,
  IQR.thresh = NULL
)

build.heatmaps(
  MEM.values.ogf,
  cluster.MEM = "both",
  display.thresh = 2,
  newWindow.heatmaps = FALSE,
  output.files = TRUE,
  labels = TRUE,
  only.MEMheatmap = FALSE
)
```
```

This section performs MEM analysis on the FlowSOM clusters based on the original markers

```r
```{r run_MEM_on_FlowSOM_on_UMAP}
# Time ~30 sec

cluster = as.numeric(as.vector((FlowSOM.clusters.umap)))
MEM.data = cbind(transformed.chosen.markers, cluster)
```

This section performs MEM analysis on the FlowSOM clusters based on the UMAP results
```

01_PBMC_extended_workflow_example.rmd

Run MEM

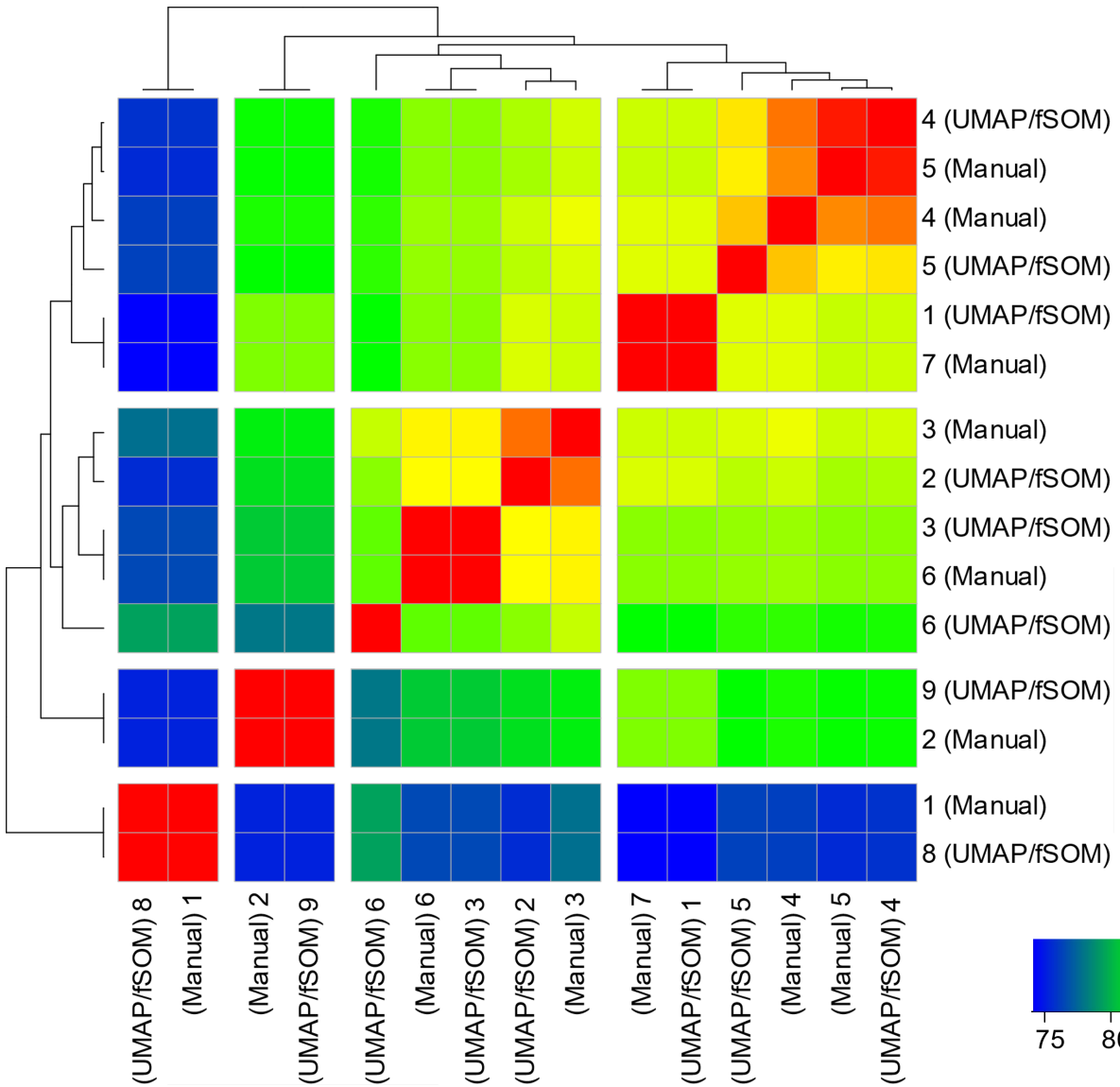
```
```{r run_MEM_on_manually_gated_pops}
# Time ~30 sec

MEM.values.orig = MEM(
  combined.data,
  transform = TRUE,
  cofactor = 15,
  choose.markers = FALSE,
  markers = "12:20,22:23,25:33,35:36,38:40",
  choose.ref = FALSE,
  zero.ref = FALSE,
  rename.markers = FALSE,
  new.marker.names = "CD19,CD117,CD11b,CD4,CD8,CD20,CD34,CD61,CD123,CD45RA,CD45,CD10,CD33,CD11c,CD14,CD69,CD15,CD16,CD44,CD38,CD25,CD3,IgM,HLA-DR,CD56",
  file.is.clust = FALSE,
  add.fileID = FALSE,
  IQR.thresh = NULL
)

build.heatmaps(
  MEM.values.orig,
  cluster.MEM = "both",
  display.thresh = 2,
  newWindow.heatmaps = FALSE,
  output.files = TRUE,
  labels = TRUE,
  only.MEMheatmap = FALSE
)
```
```

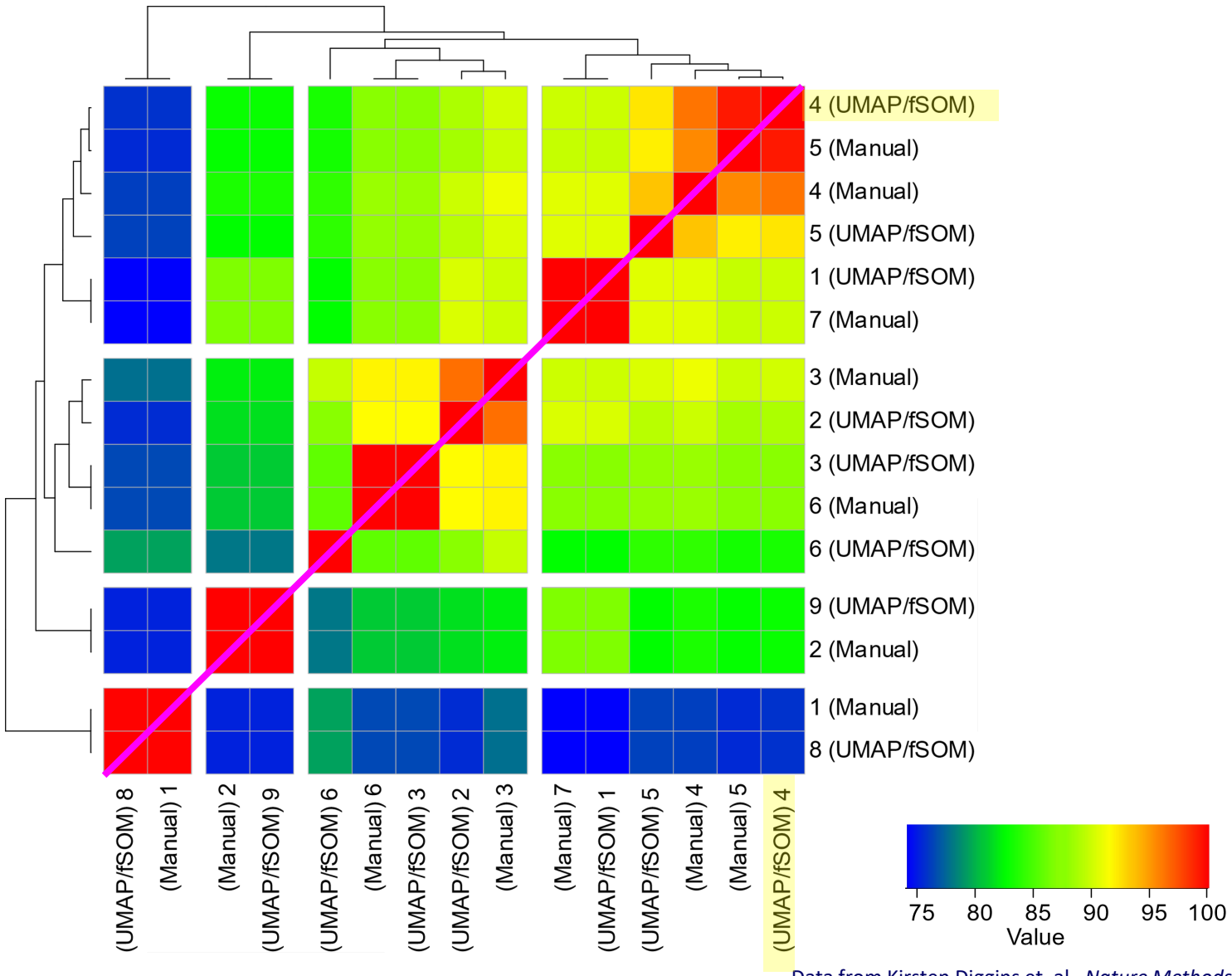
This section performs MEM analysis on the manually identified clusters

Root Mean Squared Distance is Used to Calculate the Similarity of Cell Populations Based on MEM Scores



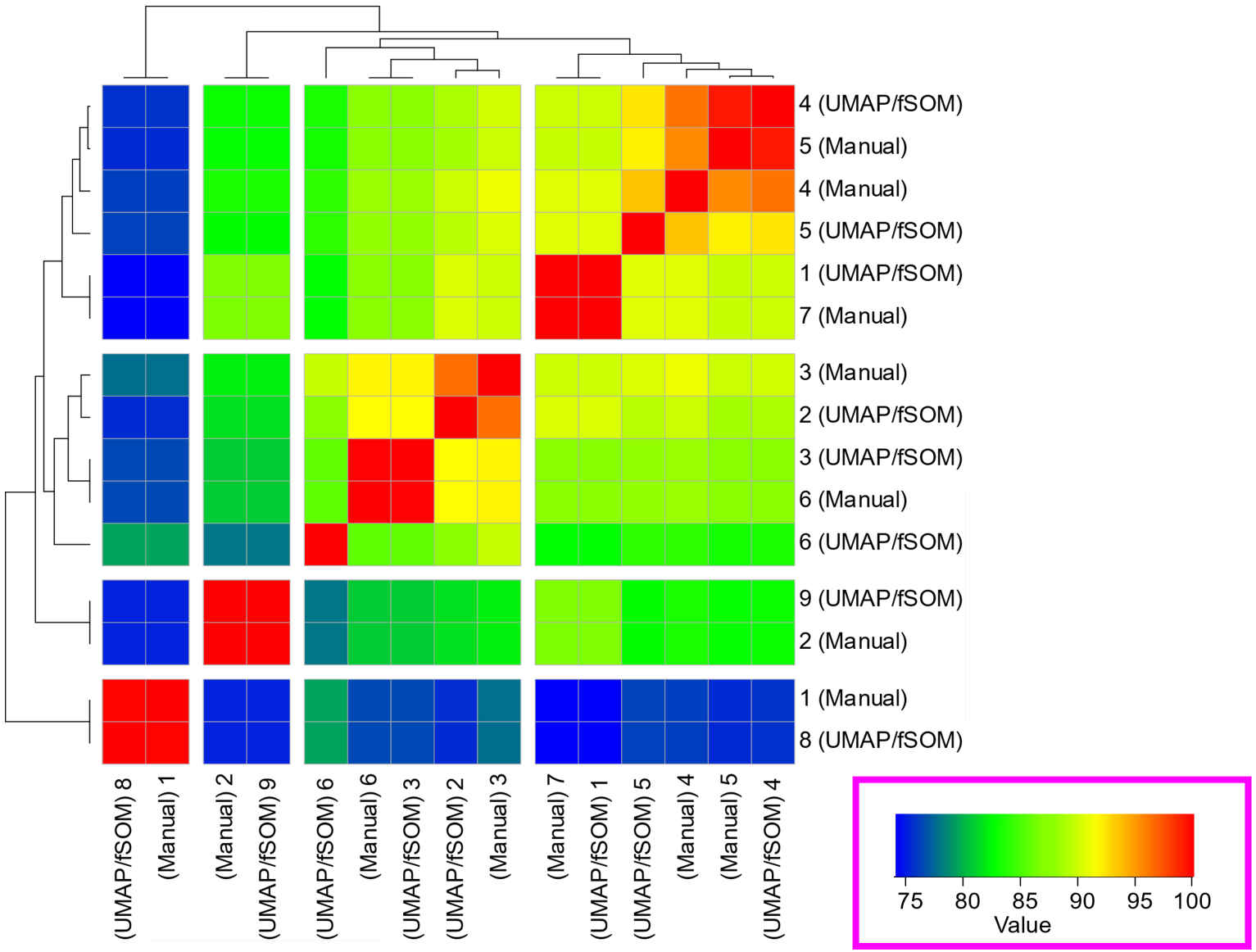
Data from Kirsten Diggins et. al, *Nature Methods*, 2017

RMSD Heatmap Output is Reflected Over the Diagonal Axis



Data from Kirsten Diggins et. al, *Nature Methods*, 2017

Heat Indicates Similarity in MEM Scores



Data from Kirsten Diggins et. al, *Nature Methods*, 2017

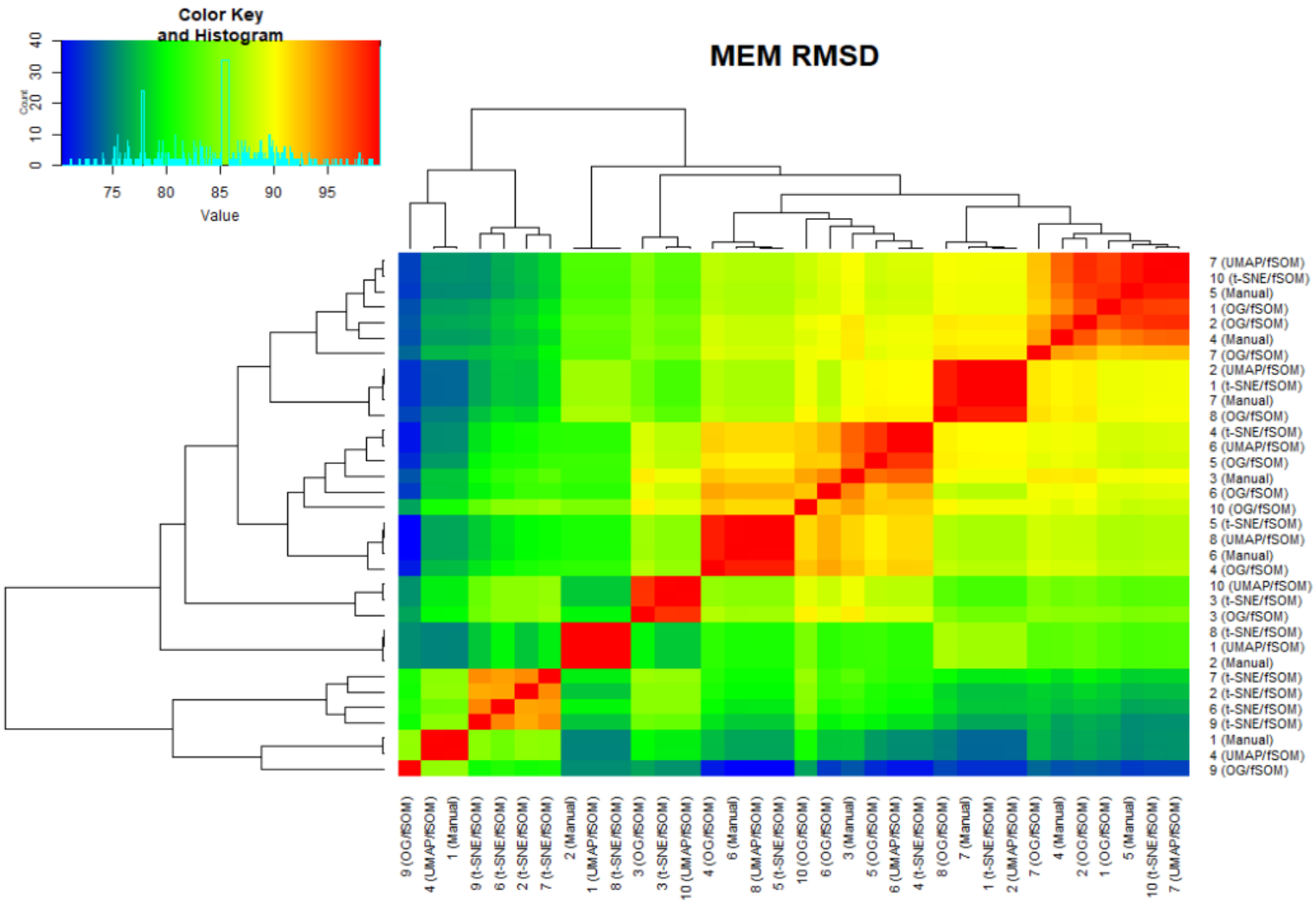
01_PBMC_extended_workflow_example.rmd

Run RMSD

```
469 ```{r RMSD for All Clusters}
470 # RMSD to compare labels from all populations (FlowSOM clusters vs. manually
471 # gated populations)
472
473 orig.MEM.scores = as.data.frame(MEM.values.orig[[5]])
474 rownames(orig.MEM.scores) = paste0(rownames(orig.MEM.scores), "
475
476 ogf.MEM.scores = as.data.frame(MEM.values.ogf[[5]])
477 rownames(ogf.MEM.scores) = paste0(rownames(ogf.MEM.scores), " (o
478
479 uf.MEM.scores = as.data.frame(MEM.values.uf[[5]])
480 rownames(uf.MEM.scores) = paste0(rownames(uf.MEM.scores), ' (UMA
481
482 tf.MEM.scores = as.data.frame(MEM.values.tf[[5]])
483 rownames(tf.MEM.scores) = paste0(rownames(tf.MEM.scores), ' (t-S
484
485 all.MEM.values = as.matrix(rbind(orig.MEM.scores, ogf.MEM.scores, uf.MEM.scores,
    tf.MEM.scores))
486
487 RMSD_vals <-
488   MEM_RMSD(
489     all.MEM.values,                # input all MEM values from clustering and
490                                    # expert gating
491     format = NULL,
492     newWindow.heatmaps = FALSE,
493     output.matrix = TRUE
494   )
495 ```
```

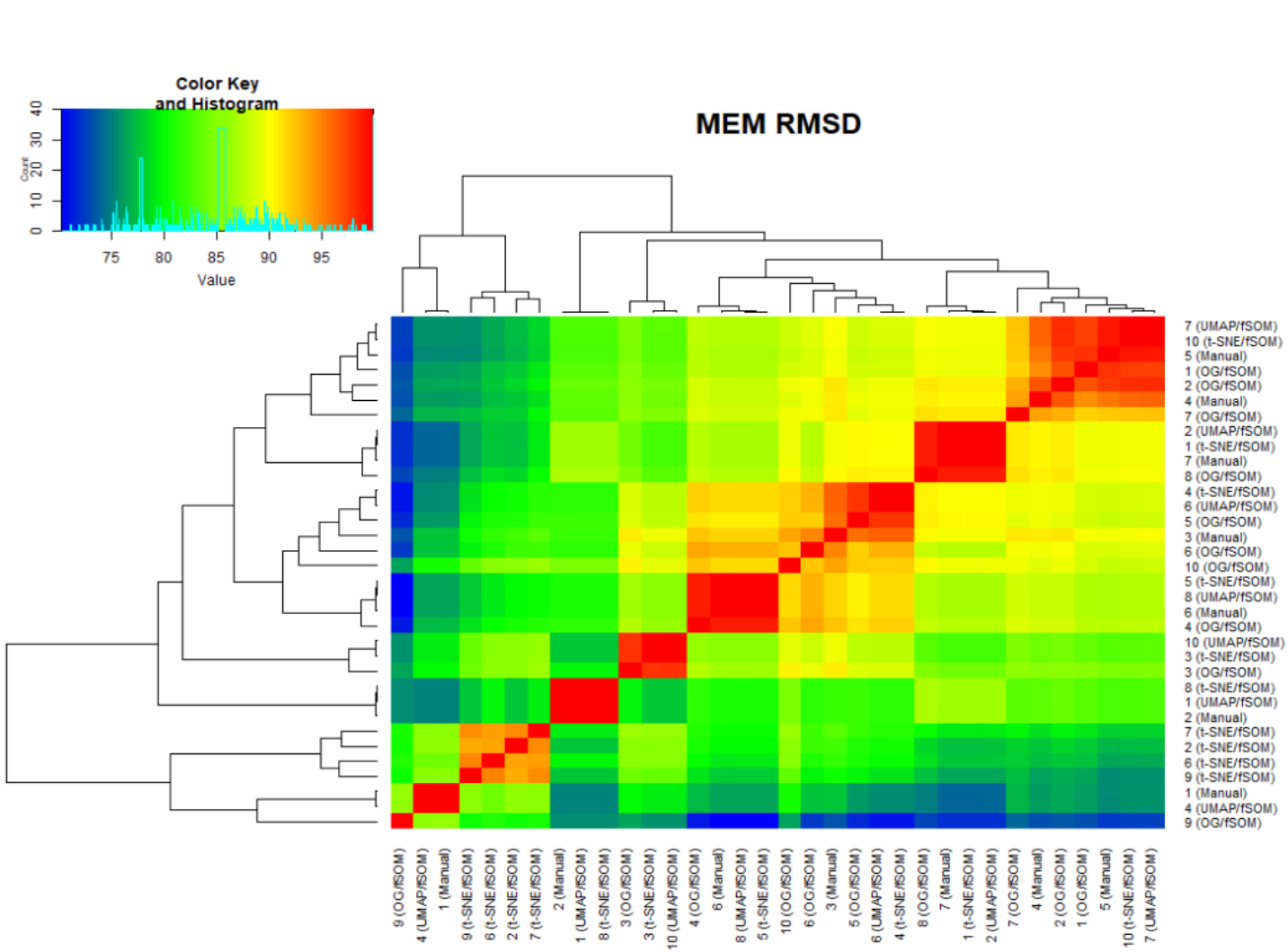
This section compares the root mean-squared distance between MEM labels from each analysis (manual clusters and flowSOM clusters from t-SNE, UMAP, and original markers)

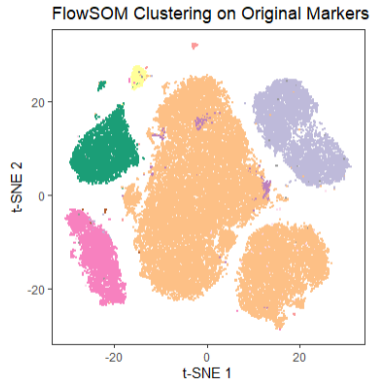
R Output for RMSD Demonstrates that Clusters are More Similar to Each other than Methods of Deriving Them

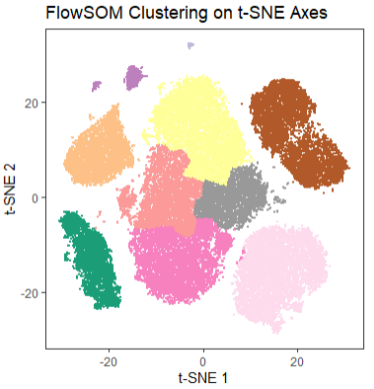


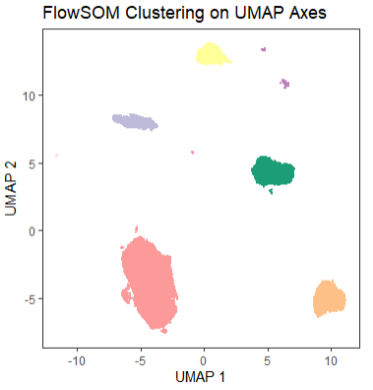
Data from Kirsten Diggins et. al, *Nature Methods*, 2017

R Output for RMSD Demonstrates that Clusters are More Similar to Each other than Methods of Deriving Them



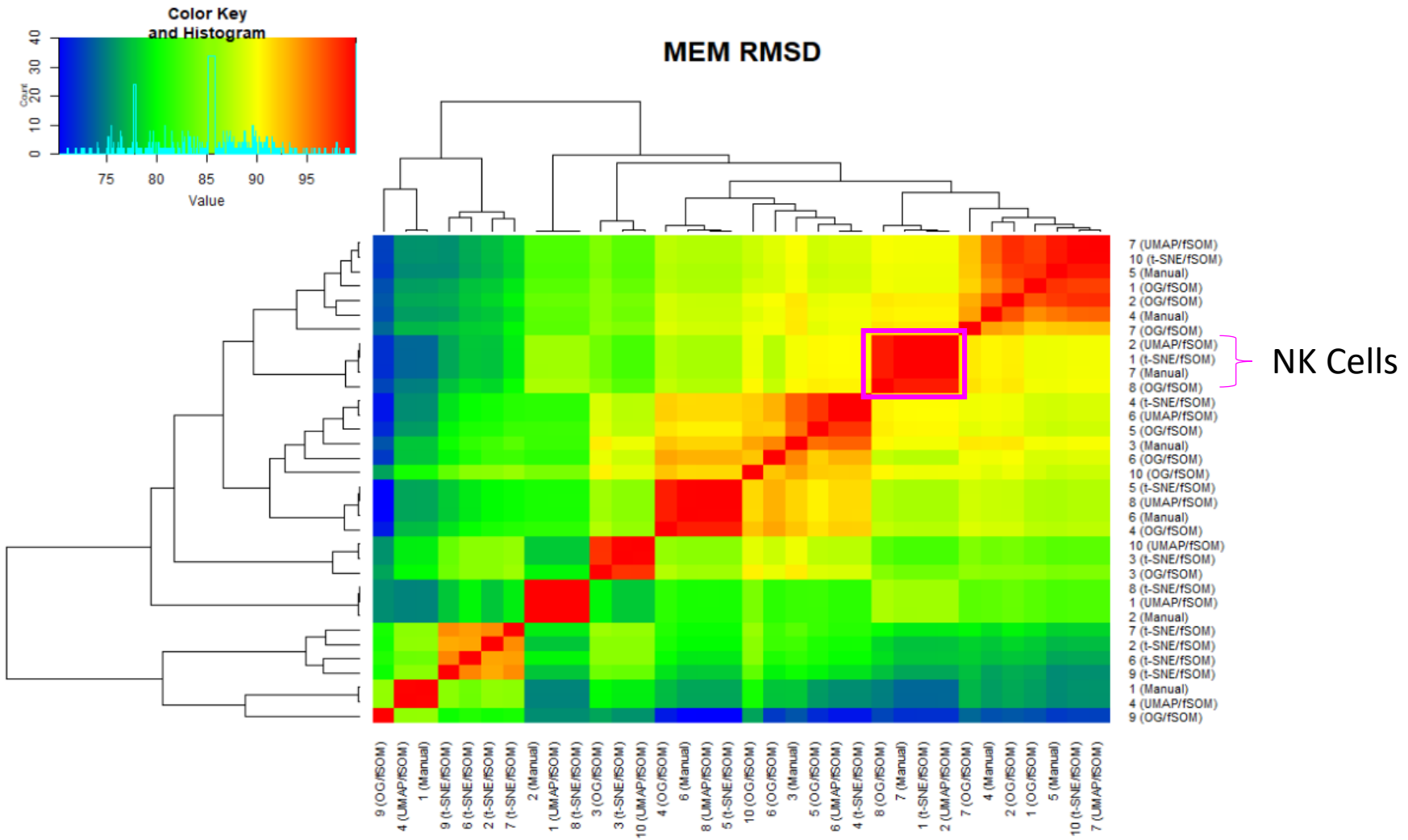






Data from Kirsten Diggins et. al, *Nature Methods*, 2017

R Output for RMSD Demonstrates that Clusters are More Similar to Each other than Methods of Deriving Them



▲CD16^{+6}
▼CD4^{-10} CD3^{-6} CD44^{-2}

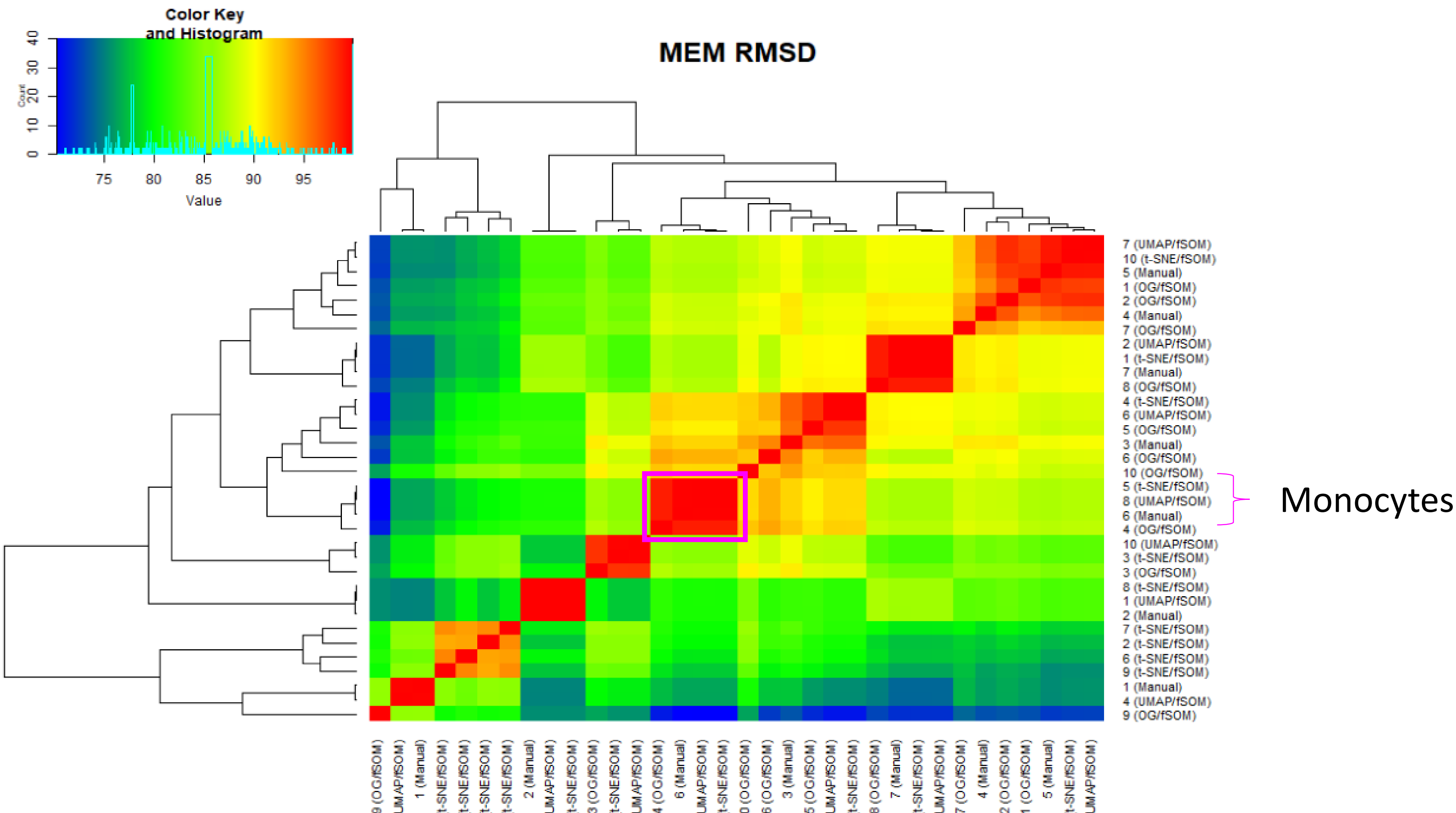
▲CD16^{+6}
▼CD4^{-10} CD3^{-6} CD44^{-2}

▲CD16^{+6}
▼CD4^{-10} CD3^{-6} CD44^{-2}

▲CD16^{+5}
▼CD4^{-9} CD3^{-5} CD44^{-2}

Data from Kirsten Diggins et. al, *Nature Methods*, 2017

R Output for RMSD Demonstrates that Clusters are More Similar to Each other than Methods of Deriving Them



▲CD33^{+5} CD11c^{+5} CD61^{+4} CD14^{+4} CD11b^{+3} CD44^{+3} HLA-DR^{+3} ▼CD3^{-7} CD4^{-5} CD8^{-3}

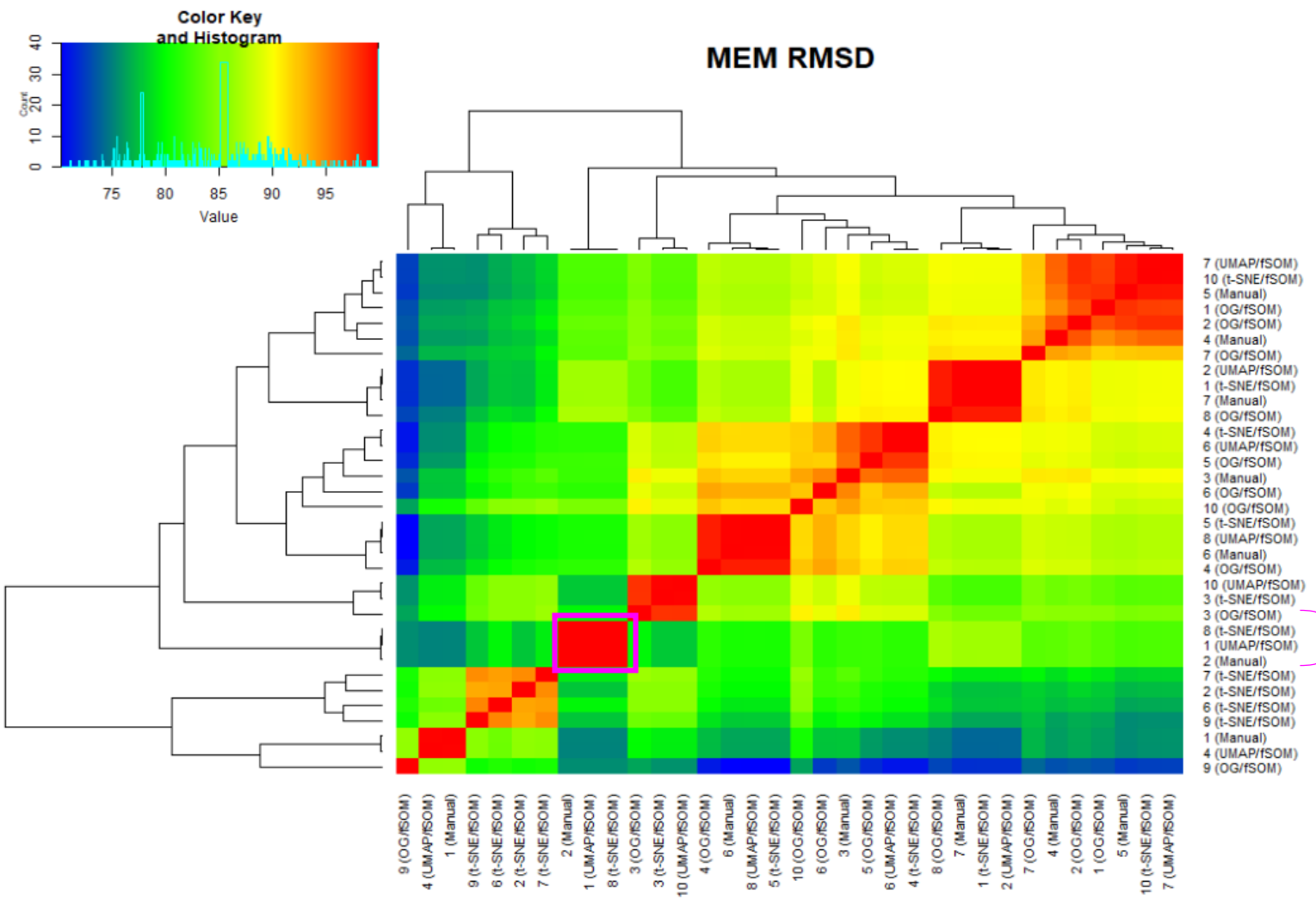
▲CD33^{+5} CD11c^{+5} CD61^{+4} CD14^{+4} CD11b^{+3} CD44^{+3} HLA-DR^{+3} ▼CD3^{-7} CD4^{-5} CD8^{-3}

▲CD33^{+5} CD11c^{+5} CD61^{+4} CD14^{+4} CD11b^{+3} CD44^{+3} HLA-DR^{+3} ▼CD3^{-7} CD4^{-5} CD8^{-3}

▲CD61^{+4} CD33^{+4} CD11c^{+4} CD14^{+4} CD11b^{+3} HLA-DR^{+3} CD44^{+2} ▼CD3^{-7} CD4^{-5} CD8^{-3}

Data from Kirsten Diggins et. al, *Nature Methods*, 2017

R Output for RMSD Demonstrates that Clusters are More Similar to Each other than Methods of Deriving Them



CD8 T Cells

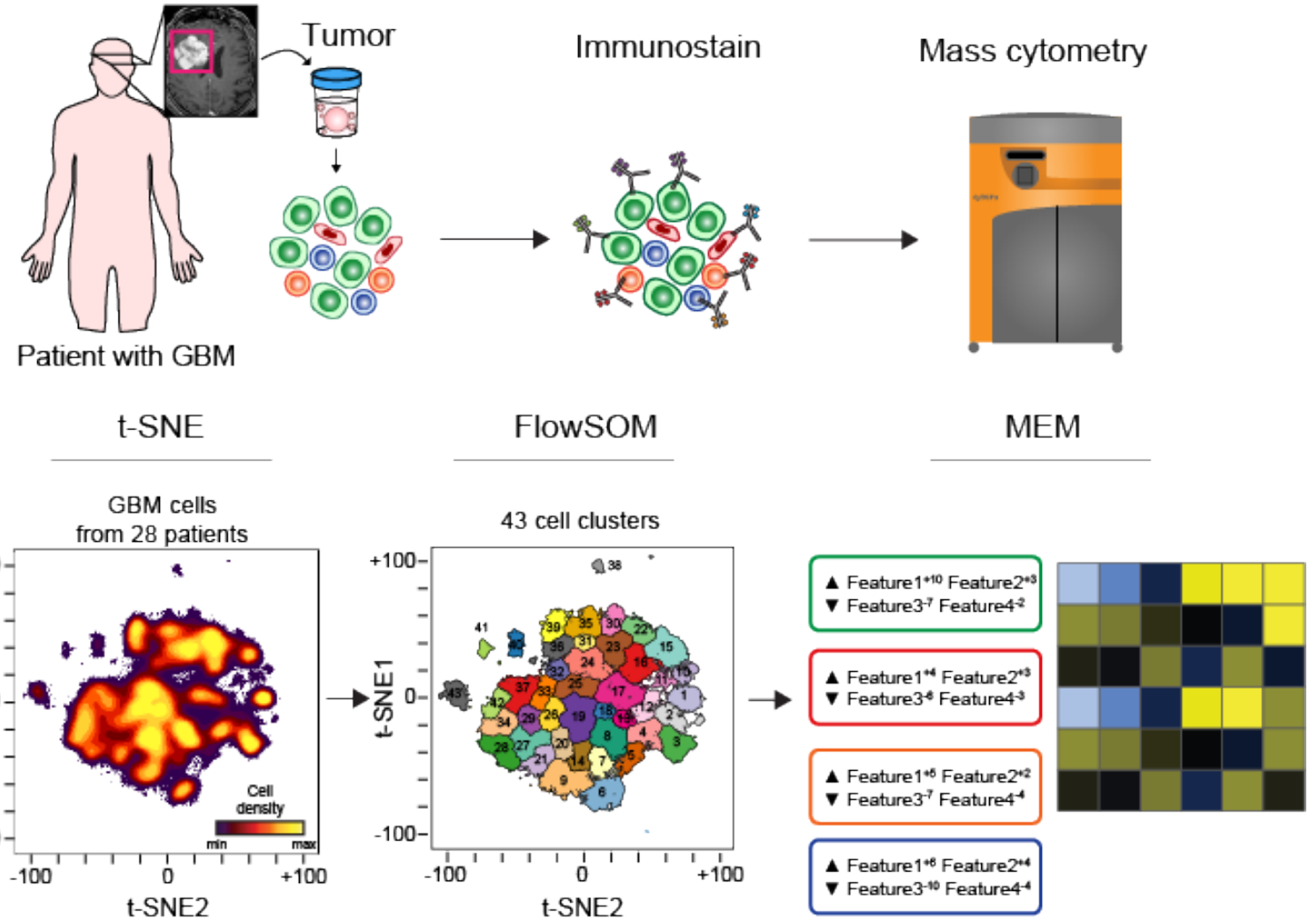
▲CD8^{+7} CD3^{+3} CD69^{+2}
▼CD4^{-10} CD38^{-2}

▲CD8^{+7} CD3^{+3} CD69^{+2}
▼CD4^{-10} CD38^{-2}

▲CD8^{+7} CD3^{+3} CD69^{+2}
▼CD4^{-10} CD38^{-2}

Data from Kirsten Diggins et. al, *Nature Methods*, 2017

t-SNE, FlowSOM, and MEM can be Used in a Data Analysis Workflow



https://www.biorxiv.org/content/10.1101/632208v1

Acknowledgements

Irish Lab

Jonathan Irish
Sierra Barone
Todd Bartkowiak
Caroline Roe
Madeline Hayes

Past Irish Lab Members

Nalin Leelatian
Kirsten Diggins
Jocelyn Gandelman
Allison Greenplate
Deon Dixie
Cara Wogsland

Ihrie Lab

Rebecca Ihrie
Justine Sinnaeve
Akshitkumar Mistry
Asa Brockman
Laura Winalski
Bret Mobley
Ethan Chervonski

Resources

Normalization
https://onlinelibrary.wiley.com/doi/full/10.1002/cyto.a.22271

Gaussian Gating
http://cytoforum.stanford.edu/download/file.php?id=242&sid=37e5ec0a3dedb53865bbbcb6a023c316

t-SNE
https://www.nature.com/articles/nbt.2594

Opt-SNE
https://www.biorxiv.org/content/10.1101/451690v3.full

UMAP
https://www.nature.com/articles/nbt.4314

FlowSOM
https://www.ncbi.nlm.nih.gov/pubmed/25573116

SPADE
https://www.nature.com/articles/nbt.1991

Phenograph
https://www.sciencedirect.com/science/article/pii/S0092867415006376

MEM
https://www.nature.com/articles/nmeth.4149

"A Beginner's Guide to Analyzing and Visualizing Mass Cytometry Data"
https://www.jimmunol.org/content/200/1/3

Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data
https://www.ncbi.nlm.nih.gov/pubmed/27992111