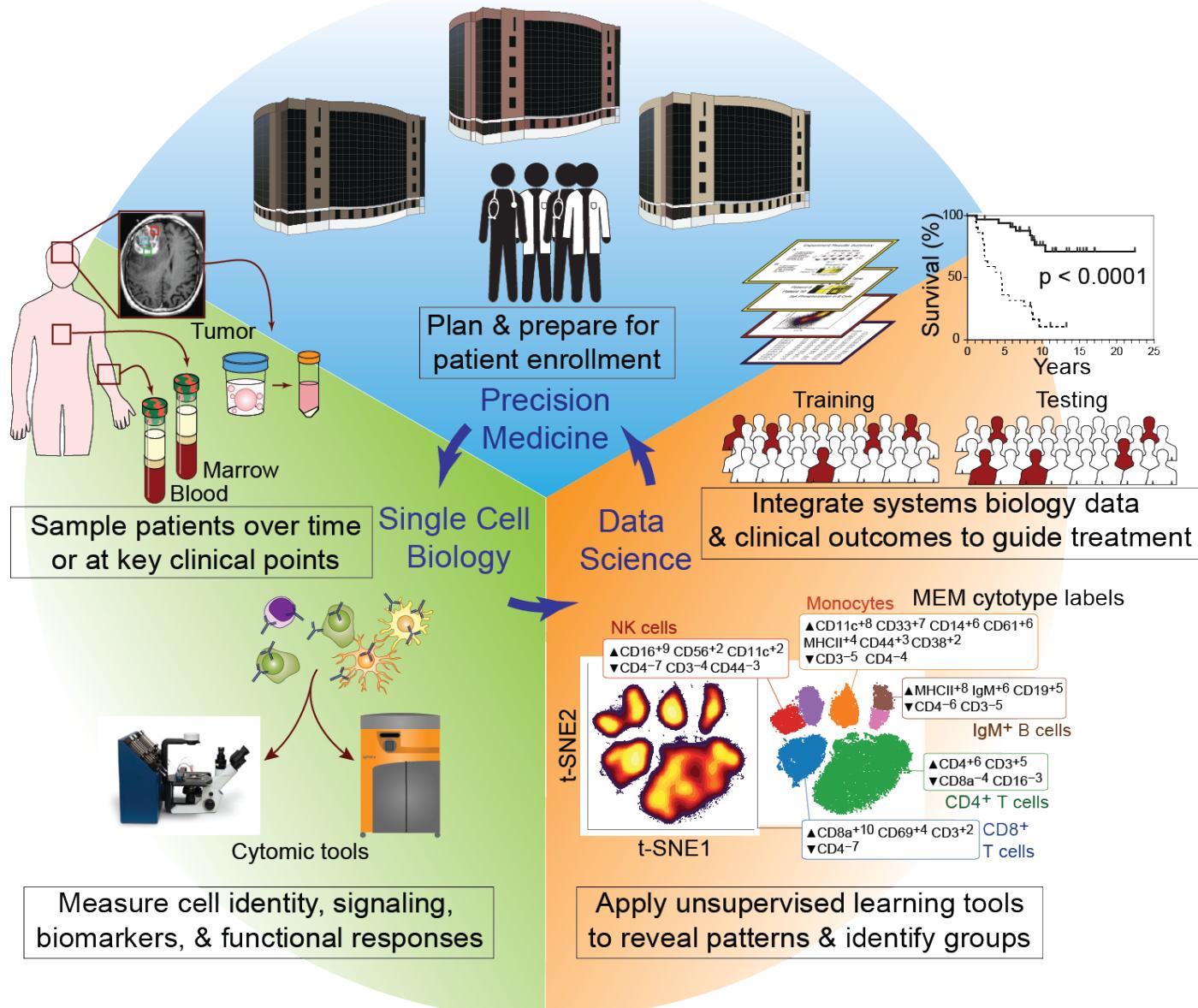


# Data Science Tools for Cancer Biologists

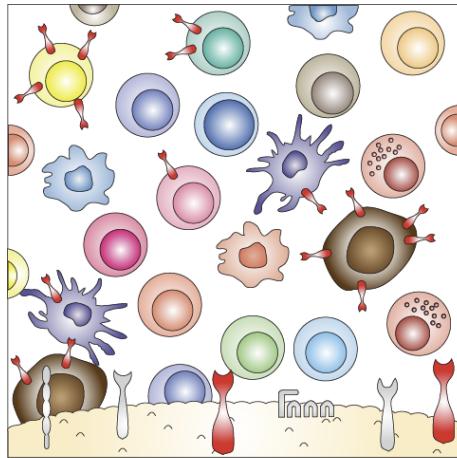
Sierra Barone  
Justine Sinnaeve  
8/22/2019

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



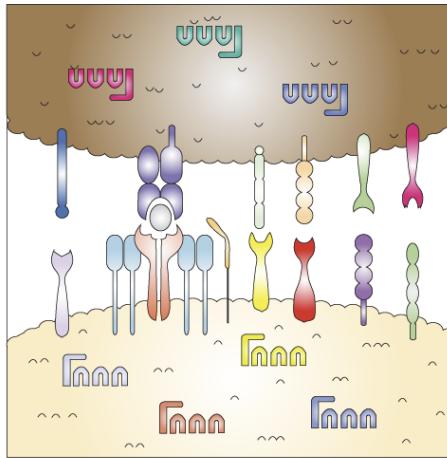
# Applications of Mass Cytometry in Cancer Biology

Microenvironment



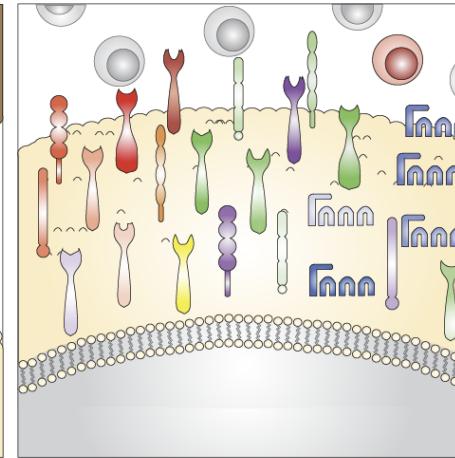
Track key biomarkers  
on all cell types;  
find cytokine producers

Cell:cell interactions



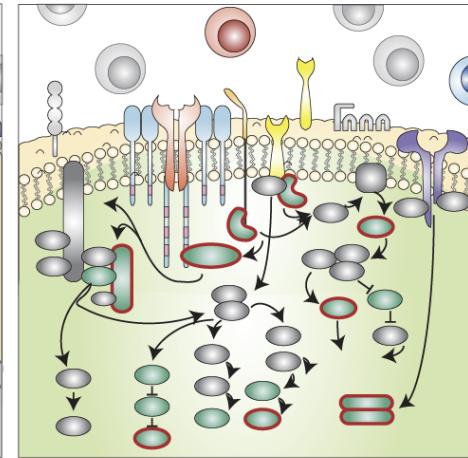
Monitor immune checkpoint  
proteins on T cells, APCs,  
& cancer cells

Immunophenotype



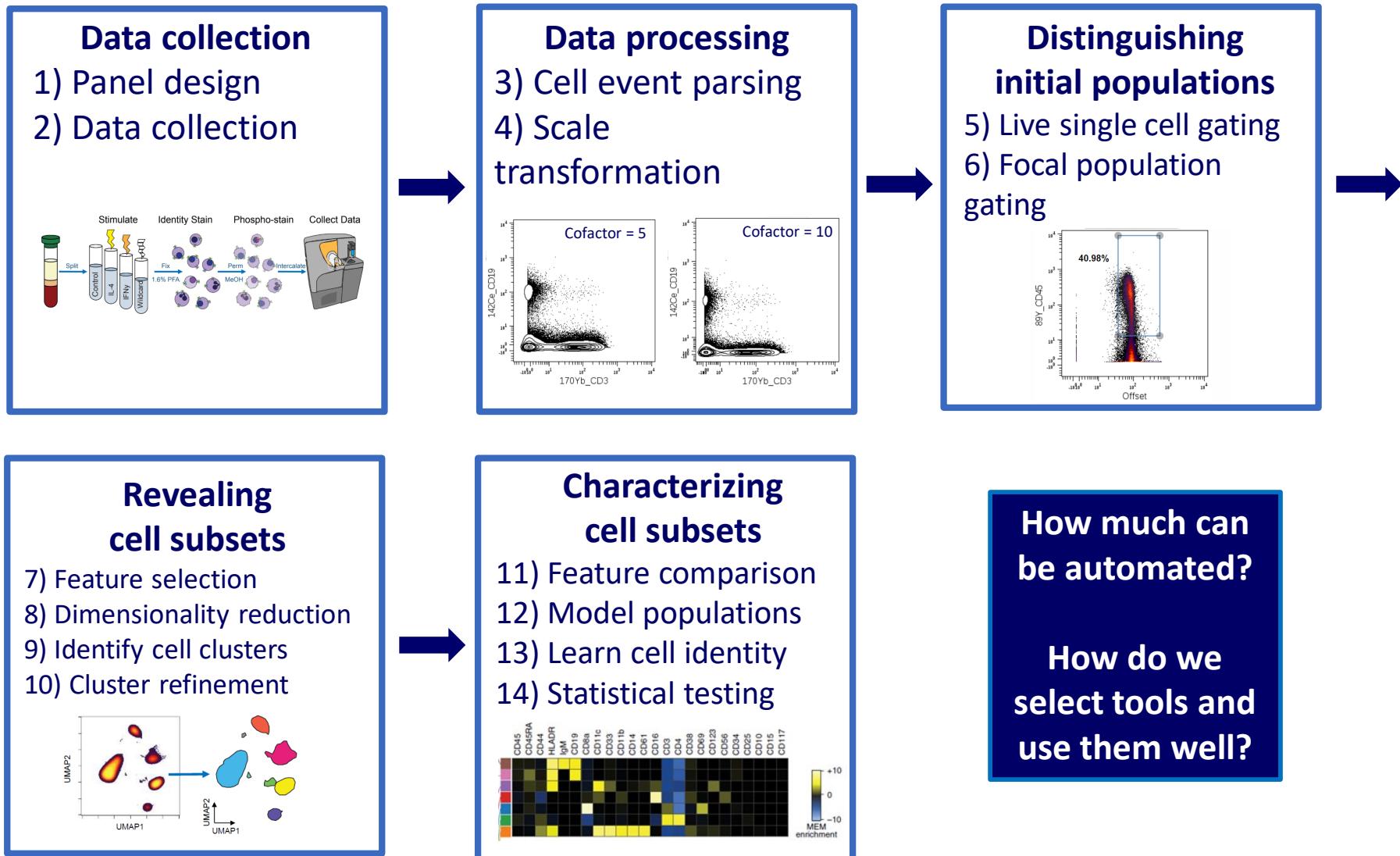
Measure differentiation;  
deep phenotype  
using fewer cells

Signaling & function

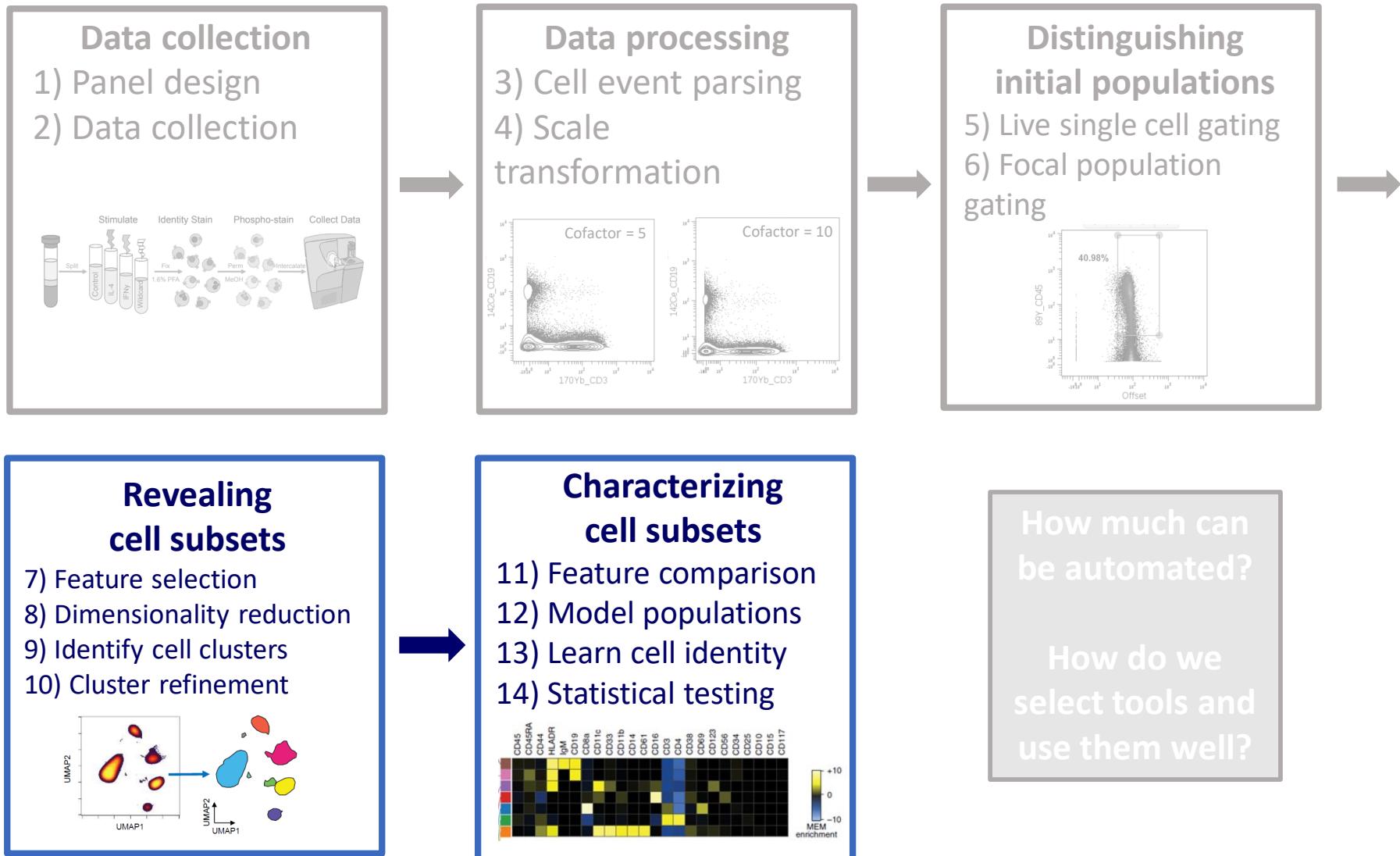


Dissect signaling changes;  
characterize mechanisms  
of treatment response

# Single Cell Biology Workflow



# Single Cell Biology Workflow



# Installation and Intro to Working in R

# Download scripts, data, and R packages from GitHub

1 Go to link below and download repository:

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

The screenshot shows the GitHub repository page for the 'irish-data-science' repository. At the top, there's a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below the bar, the repository name 'cytolab / irish-data-science' is displayed, along with statistics: 28 commits, 1 branch, 0 releases, 1 contributor, and a 'View license' link. A dropdown menu for 'Branch: master' and a 'New pull request' button are also visible. In the center, there's a commit history table with the following data:

File	Commit Message	Time Ago
R	initial commit	10 days ago
data	initial commit	10 days ago
datafiles	removed output files folder	9 days ago
figures	reworked examples	10 days ago
man	initial commit	10 days ago

At the bottom right of the screenshot, a yellow box highlights the 'Clone or download' button, which is step 2 in the process.

2

Clone with HTTPS [?](#) [Use SSH](#)  
Use Git or checkout with SVN using the web URL.  
<https://github.com/cytolab/irish-data-science>

[Open in Desktop](#)

[Download ZIP](#)

3

# Irish-data-science repository contents

1 installation script (R markdown, .rmd)

2 example analysis scripts (.rmd)

Data files (.fcs)

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

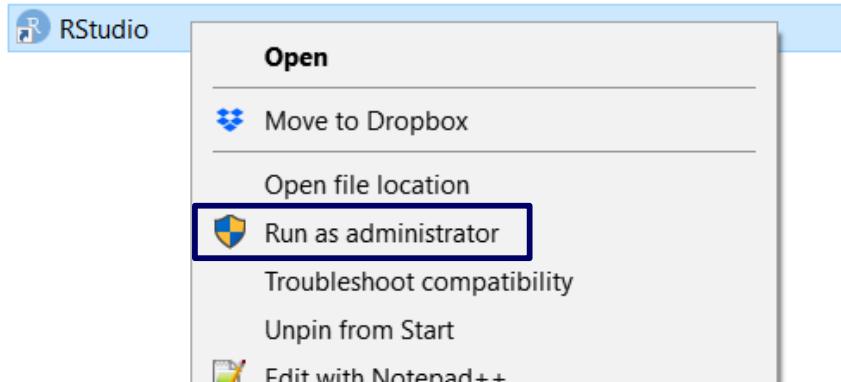
Documentation files (.rd, .md)

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

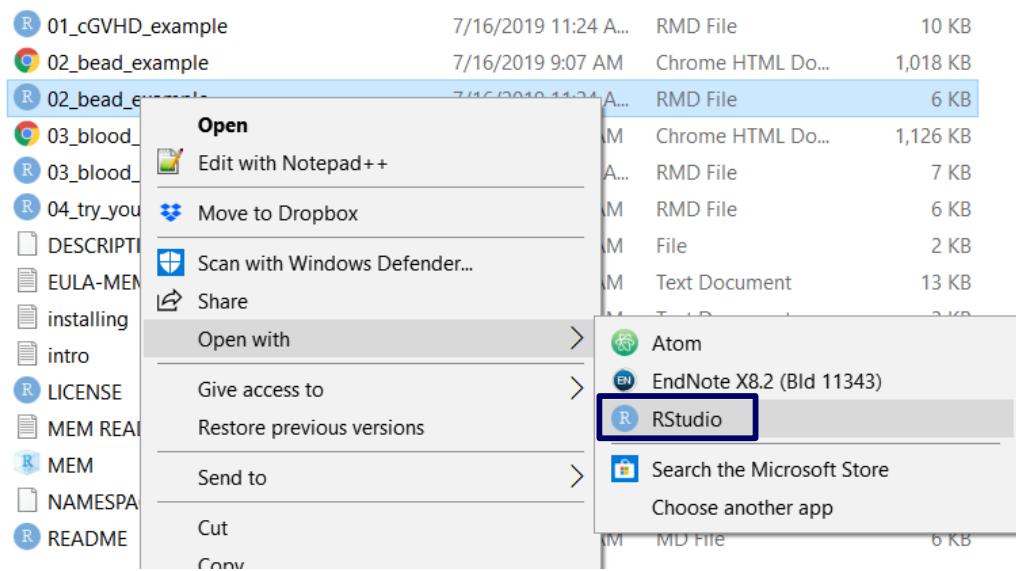
\*make sure you unzip downloaded folder

# 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\_workflow\_example.rmd

\*make sure console is open

# Working Script and Code

The screenshot shows the RStudio interface. On the left, a code editor displays a script named '01\_PBMC\_workflow\_example.Rmd'. The script includes sections for library loading and data preparation. On the right, the 'Console' tab shows the command-line output of running the script, which reads FCS files from a directory, converts them to a single data frame, and performs initial data processing steps like selecting markers and applying arcsinh transformation.

```
24
25  ```{r setup, include=FALSE}
26  # Time <10 sec
27
28 # Load all libraries
29 # If you get an error message, you will need to try re-installing packages by
30 # going back to the 00_install_tools.RMD script
31 library(FlowsOM)
32 library(flowCore)
33 library(Biobase)
34 library(ggplot2)
35 library(hexbin)
36 library(MEM)
37 library(tidyverse)
38 library(Rtsne)
39 library(uwot)
40 library(viridis)
41 library(ggExtra)
42
43
44  ```{r data_preparation, warning=FALSE}
45 # Time <10 sec
46
47 # read files into R by setting working directory and directing R to the fcs files
48 setwd(paste(getwd(), "/datafiles/PBMC", sep = ""))
49 files <- dir(pattern = "*.fcs")
50
51 # convert and combine data for use in downstream analysis
52 data <- lapply(lapply(files, read.FCS), exprs)
53 combined.data = as.data.frame(do.call(rbind, data))
54
```

1:1 Data Analysis Workflow Example on PBMC Data (t-SNE, UMAP, FlowSOM, MEM) + R Markdown +

Console Terminal Jobs

```
C:/Users/Sierra/Desktop/irish-data-science/
```

```
> 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, data))
>
> # 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
>
```

Console

# Environment

The screenshot shows the RStudio environment pane. It displays the 'Global Environment' with objects like 'combined.data' and 'transformed.chosen.markers'. Below the environment, the 'User Library' is shown as a table of installed packages, including Bioconductor packages like Biobase, BiocGenerics, and BiocManager, along with other utilities and base packages.

Name	Description	Version
acepack	ACE and AVAS for Selecting Multiple Regression Transformations	1.4.1
ape	Analyses of Phylogenetics and Evolution	5.3
askpass	Safe Password Entry for R, Git, and SSH	1.1
assertthat	Easy Pre and Post Assertions	0.2.1
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.4
base64enc	Tools for base64 encoding	0.1-3
BH	Boost C++ Header Files	1.69.0-1
bibtex	Bibtex Parser	0.4.2
Biobase	Biobase: Base functions for Bioconductor	2.44.0
BiocGenerics	S4 generic functions used in Bioconductor	0.30.0
BiocInstaller	Install/Update Bioconductor, CRAN, and github Packages	1.30.0
BiocManager	Access the Bioconductor Project Package Repository	1.30.4
BiocParallel	Bioconductor facilities for parallel evaluation	1.18.0
BiocVersion	Set the appropriate version of Bioconductor packages	3.9.0
biocViews	Categorized views of R package repositories	1.52.0
bit	A Class for Vectors of 1-Bit Booleans	1.1-14
bit64	A S3 Class for Vectors of 64bit Integers	0.9-7
bitops	Bitwise Operations	1.0-6
bmp	Read Windows Bitmap (BMP) Images	0.3

Plots, Files, Help, etc.

# Working Script and Code

# Environment

The screenshot shows the RStudio interface. The top-left pane displays the R Markdown script `01_PBMC_workflow_example.Rmd`. The code includes sections for setup, loading libraries, reading FCS files, and preparing data. The top-right pane shows the Environment browser with a list of loaded packages like `ape`, `askpass`, and `Biobase`. The bottom-left pane is the Console, showing the R command history for file reading and data processing. A red box highlights the script editor area, and a callout box points to it with the text: "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."

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.

Package	Description	Version
<code>ape</code>	Analyses of Phylogenetics and Evolution	5.3
<code>askpass</code>	Safe Password Entry for R, Git, and SSH	1.1
<code>assertthat</code>	Easy Pre and Post Assertions	0.2.1
<code>backports</code>	Reimplementations of Functions Introduced Since R-3.0.0	1.1.4
<code>base64enc</code>	Tools for base64 encoding	0.1-3
<code>BH</code>	Boost C++ Header Files	1.69.0-1
<code>bibtex</code>	Bibtex Parser	0.4.2
<code>Biobase</code>	Biobase: Base functions for Bioconductor	2.44.0
<code>BiocGenerics</code>	S4 generic functions used in Bioconductor	0.30.0
<code>BiocInstaller</code>	Install/Update Bioconductor, CRAN, and github Packages	1.30.0
<code>BiocManager</code>	Access the Bioconductor Project Package Repository	1.30.4
<code>BiocParallel</code>	Bioconductor facilities for parallel evaluation	1.18.0
<code>BiocVersion</code>	Set the appropriate version of Bioconductor packages	3.9.0
<code>biocViews</code>	Categorized views of R package repositories	1.52.0
<code>bit</code>	A Class for Vectors of 1-Bit Booleans	1.1-14
<code>bit64</code>	A S3 Class for Vectors of 64bit Integers	0.9-7
<code>bitops</code>	Bitwise Operations	1.0-6
<code>bmp</code>	Read Windows Bitmap (BMP) Images	0.3

Console

Plots, Files, Help, etc.

# Working Script and Code

The screenshot shows the RStudio interface. On the left, a script file named '01\_PBMC\_workflow\_example.Rmd' is open, displaying R code for data analysis. On the right, a 'Console' tab is active, showing the command-line history of the session. A red box highlights the console area.

```
24
25 ````{r setup, include=FALSE}
26 # Time <10 sec
27
28 # Load all libraries
29 # If you get an error message, you will need to try re-installing packages by
30 # going back to the 00_install_tools.RMD script
31 library(FlowsOM)
32 library(flowCore)
33 library(Biobase)
34 library(ggplot2)
35 library(hexbin)
36 library(MEM)
37 library(tidyverse)
38 library(Rtsne)
39 library(uwot)
40 library(viridis)
41 library(ggExtra)
42
43 ````{r data_preparation, warning=FALSE}
44 # Time <10 sec
45
46
47 # read files into R by setting working directory and directing R to the fcs files
48 setwd(paste(getwd(), "/datafiles/PBMC", sep = ""))
49 files <- dir(pattern = "*.fcs")
50
51 # convert and combine data for use in downstream analysis
52 data <- lapply(lapply(files, read.FCS), exprs)
53 combined.data = as.data.frame(do.call(rbind, data))
54
```

```
C:/Users/Sierra/Desktop/irish-data-science/
> 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, data))
>
> # 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
>
```

Console

# Environment

The screenshot shows the RStudio interface. On the right, the 'Environment' pane lists variables and their values. The 'Packages' pane shows a list of installed packages with their versions. A red box highlights the 'Console' area.

Environment

Name	Description	Version
combined.data	49651 obs. of 46 variables	
data	List of 7	
transformed.chos...	49651 obs. of 25 variables	

Packages

Name	Description	Version
acepack	ACE and AVAS for Selecting Multiple Regression Transformations	1.4.1
ape	Analyses of Phylogenetics and Evolution	5.3
askpass	Safe Password Entry for R, Git, and SSH	1.1
assertthat	Easy Pre and Post Assertions	0.2.1
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.4
base64enc	Tools for base64 encoding	0.1-3
BH	Boost C++ Header Files	1.69.0-1

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.

Plots, Files, Help, etc.

\*make sure console is open

# Working Script and Code

The screenshot shows the RStudio interface. On the left, a script file named "01\_PBMC\_workflow\_example.Rmd" is open, displaying R code for data analysis. On the right, a "Console" tab is active, showing the command-line history of the session. The console output matches the code in the script file.

```
24 25  ```{r setup, include=FALSE}
26 # Time <10 sec
27
28 # Load all libraries
29 # If you get an error message, you will need to try re-installing packages by
30 # going back to the 00_install_tools.RMD script
31 library(FlowsOM)
32 library(flowCore)
33 library(Biobase)
34 library(ggplot2)
35 library(hexbin)
36 library(MEM)
37 library(tidyverse)
38 library(Rtsne)
39 library(uwot)
40 library(viridis)
41 library(ggExtra)
42
43
44  ```{r data_preparation, warning=FALSE}
45 # Time <10 sec
46
47 # read files into R by setting working directory
48 setwd(paste(getwd(), "/datafiles/PBMC", sep = ""))
49 files <- dir(pattern = "*.fcs")
50
51 # convert and combine data for use in downstream analysis
52 data <- lapply(files, read.FCS), exprs)
53 combined.data = as.data.frame(do.call(rbind, data))
54
```

```
> files <- dir(pattern = "*.fcs")
>
> # convert and combine data for use in downstream analysis
> data <- lapply(files, read.FCS), exprs)
> combined.data = as.data.frame(do.call(rbind, data))
>
> # 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
>
```

Console

# Environment

The screenshot shows the RStudio environment pane. It displays the global environment with various objects listed, such as "combined.data", "data", "transformed.chosen.markers", "values", "files", and "overall\_seed". Below the environment pane is a "Packages" tab showing a list of installed packages with their descriptions and versions.

Environment History Connections

Global Environment

Data

- combined.data 49651 obs. of 46 variables
- data List of 7
- transformed.chosen.markers 49651 obs. of 25 variables

Values

- files chr [1:7] "CD4Tcells\_PBMC.fcs" "CD8Tcells\_PBMC.f...
- overall\_seed 43

Packages Help Viewer

Description	Version
ACE and AVAS for Selecting Multiple Regression Transformations	1.4.1
Analyses of Phylogenetics and Evolution	5.3
Safe Password Entry for R, Git, and SSH	1.1
Easy Pre and Post Assertions	0.2.1
Reimplementations of Functions Introduced Since R-3.0.0	1.1.4
Tools for base64 encoding	0.1-3
Boost C++ Header Files	1.69.0-1
Bibtex Parser	0.4.2
Biobase: Base functions for Bioconductor	2.44.0
S4 generic functions used in Bioconductor	0.30.0
Install/Update Bioconductor, CRAN, and github Packages	1.30.0
Access the Bioconductor Project Package Repository	1.30.4
Bioconductor facilities for parallel evaluation	1.18.0
Set the appropriate version of Bioconductor packages	3.9.0
Categorized views of R package repositories	1.52.0
A Class for Vectors of 1-Bit Booleans	1.1-14
A S3 Class for Vectors of 64bit Integers	0.9-7
Bitwise Operations	1.0-6
Read Windows Bitmap (BMP) Images	0.3

Plots, Files, Help, etc.

# Working Script and Code

The screenshot shows the RStudio interface. On the left, a script file named '01\_PBMC\_workflow\_example.Rmd' is open, displaying R code for setting up and preparing data. On the right, a terminal window shows the execution of this code, including commands like `setwd`, `lapply`, and `read.FCS` to process FCS files into a combined dataset.

```
24
25  ```{r setup, include=FALSE}
26  # Time <10 sec
27
28 # Load all libraries
29 # If you get an error message, you will need to try re-installing packages by
30 # going back to the 00_install_tools.RMD script
31 library(FlowsOM)
32 library(flowCore)
33 library(Biobase)
34 library(ggplot2)
35 library(hexbin)
36 library(MEM)
37 library(tidyverse)
38 library(Rtsne)
39 library(uwot)
40 library(viridis)
41 library(ggExtra)
42
43
44  ```{r data_preparation, warning=FALSE}
45 # Time <10 sec
46
47 # read files into R by setting working dir
48 setwd(paste(getwd(), "/datafiles/PBMC", sep = ""))
49 files <- dir(pattern = "*.fcs")
50
51 # convert and combine data for use in downstream analysis
52 data <- lapply(lapply(files, read.FCS), exprs)
53 combined.data = as.data.frame(do.call(rbind, data))
54
55 # choose channels with markers to use for downstream analysis
56 # transformation with a cofactor of 15
57 transformed.chosen.markers <- combined.data %>
+   select(contains("-"), !contains("Ir")) %>%
+   mutate_all(function(x)
+     asinh(x / 15)) # cofactor here is 15; this can be changed
58
59 # set seed for reproducible results (43 is chosen below)
60 overall_seed = 43
61
```

Console

# Environment

The screenshot shows the RStudio Environment tab. It displays the global environment with variables like 'combined.data' and 'data'. Below this, the 'User Library' tab is selected, showing a list of installed packages such as 'pack', 'ace', 'pass', 'erthat', 'kports', 'base64enc', 'tex', 'base', 'generics', 'biocManager', 'parallel', and 'version'. Each package entry includes its name, description, and version number.

Name	Description	Version
pack	ACE and AVAS for Selecting Multiple Regression Transformations	1.4.1
ace	Analyses of Phylogenetics and Evolution	5.3
pass	Safe Password Entry for R, Git, and SSH	1.1
erthat	Easy Pre and Post Assertions	0.2.1
kports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.4
base64enc	Tools for base64 encoding	0.1-3
tex	Boost C++ Header Files	1.69.0-1
base	Bibtex Parser	0.4.2
generics	S4 generic functions used in Bioconductor	0.30.0
biocInstaller	Install/Update Bioconductor, CRAN, and github Packages	1.30.0
biocManager	Access the Bioconductor Project Package Repository	1.30.4
parallel	Bioconductor facilities for parallel evaluation	1.18.0
version	Set the appropriate version of Bioconductor packages	3.9.0
biocViews	Categorized views of R package repositories	1.52.0
bit	A Class for Vectors of 1-Bit Booleans	1.1-14
bit64	A S3 Class for Vectors of 64bit Integers	0.9-7
bitops	Bitwise Operations	1.0-6
bmp	Read Windows Bitmap (BMP) Images	0.3

Plots, Files, Help, etc.

This window will display files in your working directory, plots you have created, as well as packages you have installed and loaded. You can also access help pages for each package in this window.

# Open 00\_install\_tools.rmd

1

```
1 ---  
2 title: "Check Paths and Install Packages"  
3 author: "Copyright (c) 2016-2019 by Kirsten Diggins, Sierra Barone, and  
4 Jonathan Irish, All Rights Reserved; see EULA-MEM.text for MEM license  
5 information"  
6 date: "July 2019"  
7 output: html_document  
8 ---  
9   
10 cat("This section checks to see if files and paths are working correctly. You  
11 should see lists of files below. If it outputs character(0), something is  
12 wrong.\n\n")  
13 # Check the MEM code path  
14 cat("\n\nThe /MEM folder contains the MEM source code for install and related  
15 files:\n")  
16 list.files(getwd())  
17 # Check for datasets  
18 cat("\n\nCourse FCS format files are in subdirecties of the /datafiles  
19 folder:\n")  
20 list.files(paste(getwd(), "/datafiles", sep=""))  
21   
22   
23 # This only works for PC users
```

Header

2

Code

3

# Open 00\_install\_tools.rmd and begin installing required packages

## 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=""))  
...`
```

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

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

OR

Press play to run  
entire section of  
code

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

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")  
```
```

The next sections install and load the tools to make plots

```
```{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)  
```
```

# 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")  
...```

```

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

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

```

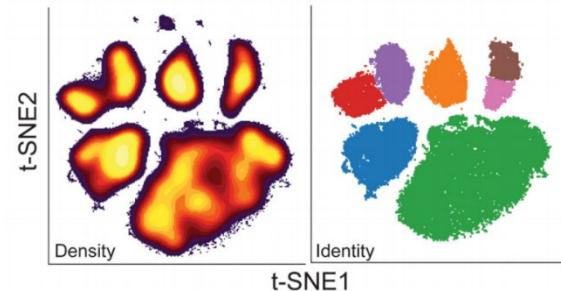
Open **01\_PBMC\_workflow\_example.rmd**  
and work through the example

# 01\_PBMC\_workflow\_example.rmd

```
01_PBMC_workflow_example.Rmd x
Insert Run
1 ---  
2 title: "Data Analysis workflow Example on PBMC Data (t-SNE, UMAP, FlowsOM,  
MEM)"  
3 author: "Copyright (c) 2016-2019 by Kirsten Diggins, Sierra Barone, and  
Jonathan Irish, All Rights Reserved; see EULA-MEM.text for MEM license  
information"  
4 date: "July 2019"  
5 output:  
6   pdf_document:  
7     latex_engine: xelatex  
8   html_document:  
9     df_print: paged  
10  editor_options:  
11    chunk_output_type: inline  
12 ---  
13  
14 This data set contains 7 FCS (flow cytometry standard) files. Each FCS file  
15 contains single cell data for one cell subset that is a well-established,  
16 phenotypically distinct population. This is mass cytometry data for healthy  
17 human PBMC (peripheral blood mononuclear cells). The populations were expert  
18 gated following a t-SNE analysis. The first section of the code will run two  
19 dimensionality reduction tools, UMAP and t-SNE, on the data set. Next, you  
20 will run FlowsOM on the UMAP axes to cluster the islands of cell populations.  
21 Finally, you will run MEM to see enrichment scores for each of the FlowsOM  
22 clusters. The goal of this exercise is to run several computational tools on  
23 a single cell data set to get a feel for the workflow used in the Irish lab.  
24  
25 ````{r setup, include=FALSE}  
26 # Time <10 sec  
27  
28 # Load all libraries  
29 # If you get an error message, you will need to try re-installing packages by  
30 # going back to the 00_install_tools.RMD script  
31 library(FlowsOM)  
32 library(flowCore)  
33 library(BioBase)  
34 library(ggplot2)  
35 library(hexbin)  
36 library(MEM)  
37 library(tidyverse)  
38 library(Rtsne)  
39 library(uwot)  
40 library(viridis)  
41 library(ggExtra)  
42 ````
```

A description of the code and its purpose

a Identification of 7 canonical cell types in healthy human blood, 25D mass cytometry



This section loads the necessary libraries

# 01\_PBMC\_workflow\_example.rmd

## Data Preparation

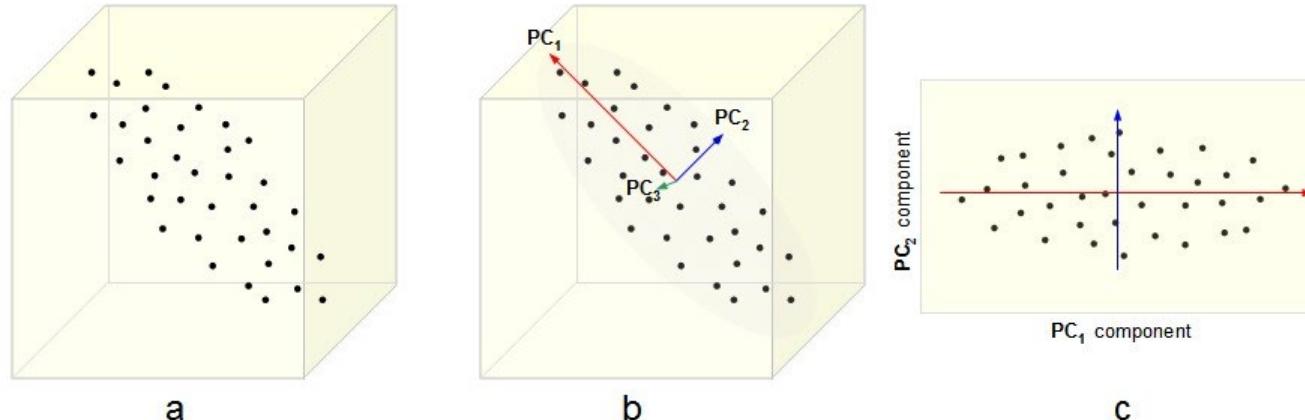
```
43
44 `r data_preparation, warning=FALSE}
45 # Time <10 sec
46
47 # read files into R by setting working directory and directing R to
# files
48 setwd(paste(getwd(), "/datafiles/PBMC", sep = ""))
49 files <- dir(pattern = "*.fcs")
50
51 # convert and combine data for use in downstream analysis
52 data <- lapply(lapply(files, read.FCS), exprs)
53 combined.data = as.data.frame(do.call(rbind, data))
54
55 # choose channels with markers to use for downstream analysis and apply arcsinh
56 # transformation with a cofactor of 15
57 transformed.chosen.markers <- combined.data %>%
58   select(contains("-"), -contains("Ir")) %>%
59   mutate_all(function(x)
60     asinh(x / 15))      # cofactor here is 15; this can be changed
61
62 # set seed for reproducible results (43 is chosen below)
63 overall_seed = 43
64
65
```

Read the data files into R and format for analysis

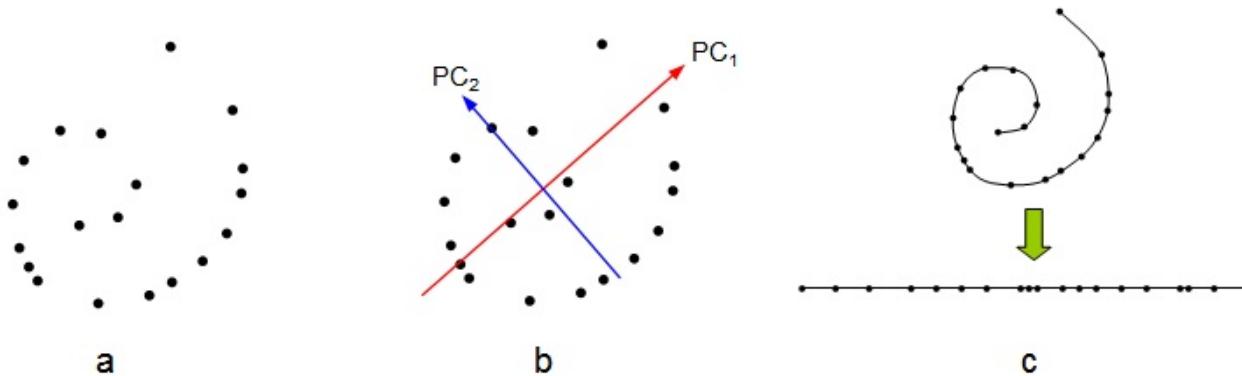
Select channels and scale the data

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.

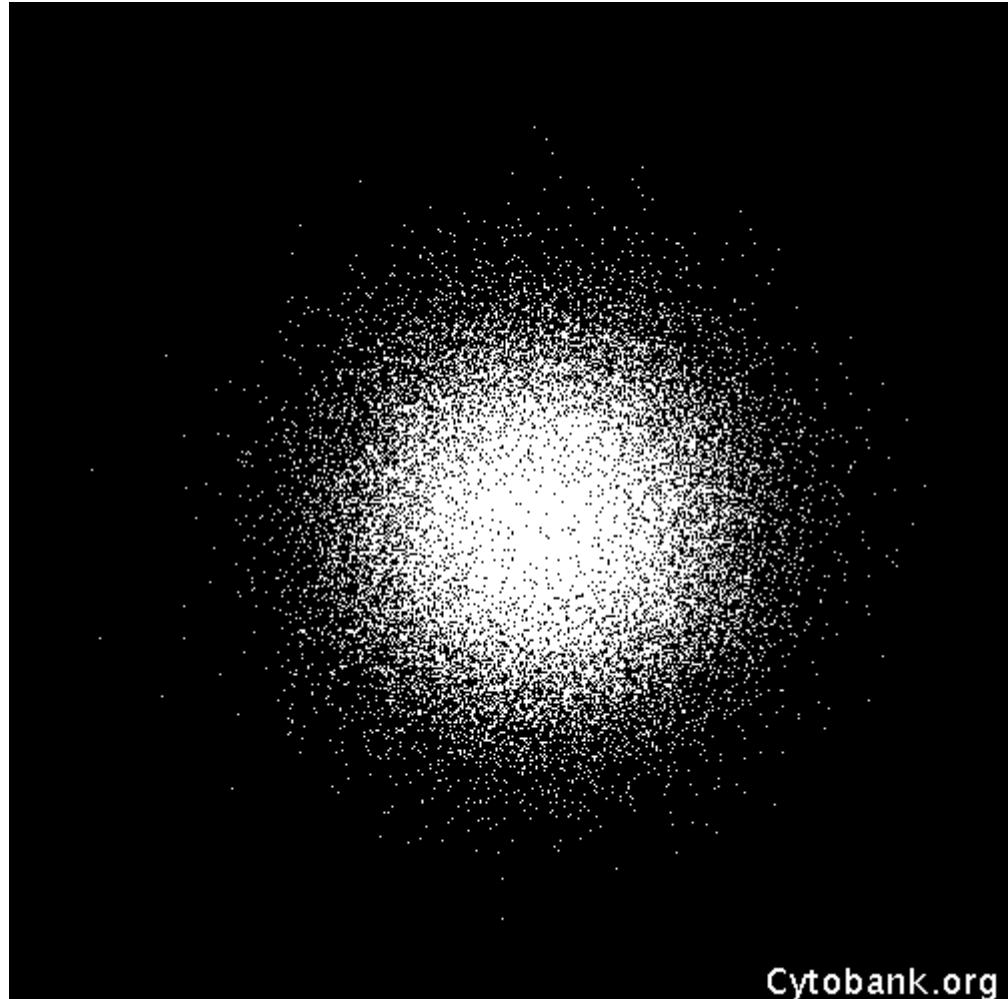


Effects of dimensionality reduction on an inherently non-linear data set. **a)** The original data given as a two-dimensional set. **b)** PCA identifies two PCs as contributing significantly to explain the data variance. **c)** However, the inherent topology (connectivity) of the data helps identify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

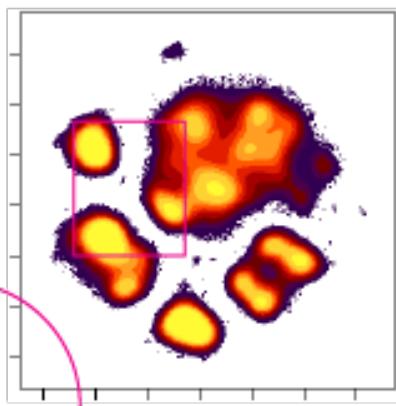
Animation created by Cytobank team from iterations of viSNE / t-SNE using Healthy PBMC (26 features)

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

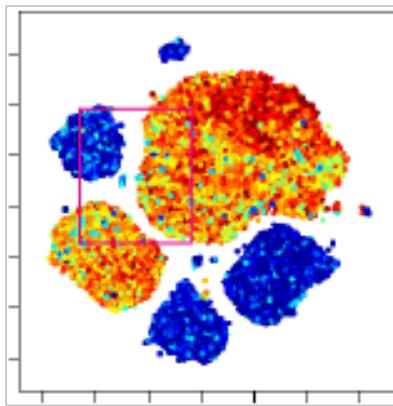
Same map, different information

Healthy Peripheral Blood Mononuclear Cells

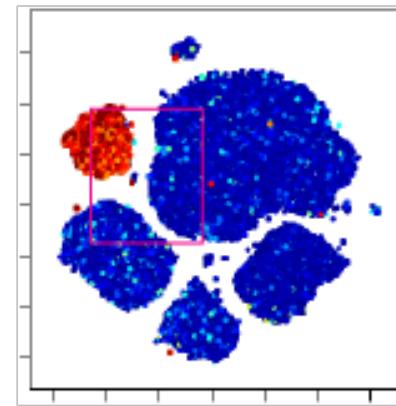
Cell Density



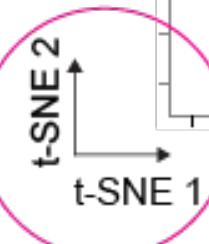
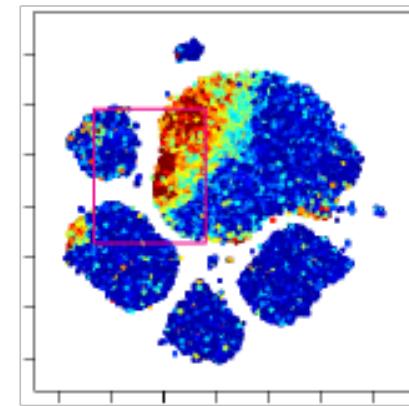
CD3



CD19

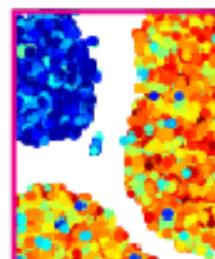


CD25



Cell  
density

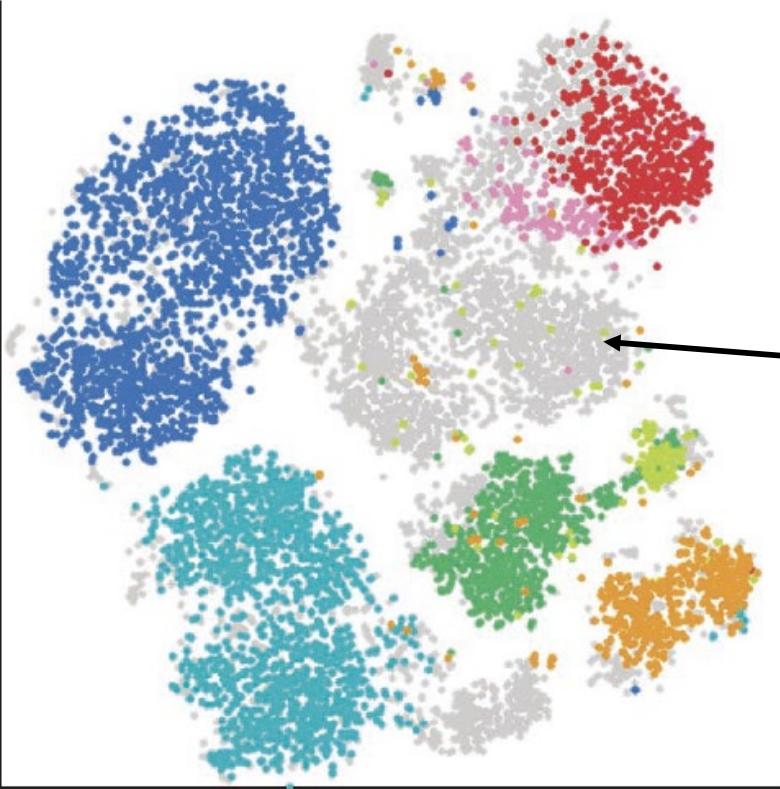
Protein  
expression



New 2D axes that represent phenotypic similarities of single cells

1 dot = 1 cell

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



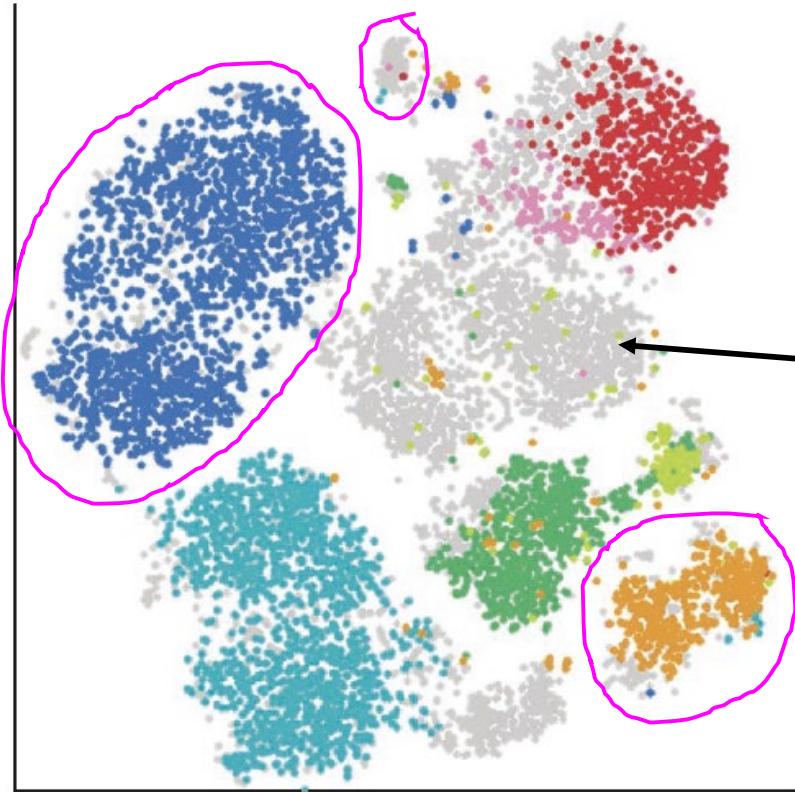
viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia

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

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**).

- Not manually gated
- CD4 T cells
- CD8 T cells
- CD20<sup>+</sup> B cells
- CD20<sup>-</sup> B cells
- CD11b<sup>-</sup> monocytes
- CD11b<sup>+</sup> monocytes
- NK cells

# 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<sup>1</sup>, Kara L Davis<sup>2,3</sup>, Michelle D Tadmor<sup>1,3</sup>, Erin F Simonds<sup>2,3</sup>, Jacob H Levine<sup>1,3</sup>, Sean C Bendall<sup>2,3</sup>, Daniel K Shenfeld<sup>1,3</sup>, Smita Krishnaswamy<sup>1</sup>, Garry P Nolan<sup>2,4</sup> & Dana Pe'er<sup>1,4</sup>

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**).

- Not manually gated
- CD4 T cells
- CD8 T cells
- CD20<sup>+</sup> B cells
- CD20<sup>-</sup> B cells
- CD11b<sup>-</sup> monocytes
- CD11b<sup>+</sup> monocytes
- NK cells

# 01\_PBMC\_workflow\_example.rmd

## Run t-SNE

```
66 ~ ````{r run_t-SNE}
67 # Time ~5 min
68
69 set.seed(overall_seed)
70
71 # the line below will run t-SNE on the scaled surface markers (to see help p
for t-SNE, type "?Rtsne -- enter" in console)
72 # you can view t-SNE progress by opening up the console below
73 mytsNE = Rtsne(
74   transformed.chosen.markers,                      # input scaled data
75   dims = 2,                                         # number of final
dimensions
76   initial_dims = length(transformed.chosen.markers), # number of initial
dimensions
77   perplexity = 30,                                  # perplexity (similar to # of nearest neighbors,
# will scale with data sets, cannot be greater than
# the number of events minus 1 divided by 3)
78   check_duplicates = FALSE,
79   max_iter = 1000,                                   # number of iterations
80   verbose = TRUE
81 )
82 tsne.data = as.data.frame(mytsNE$Y)
83 ``
84
85
86
87 ~ ````{r plot_t-SNE}
88 # Time <10 sec
89
90 # setting aspect ratio for plots
91 range <- apply(apply(tsne.data, 2, range), 2, diff)
92 graphical.ratio <- (range[1] / range[2])
93
94 # t-SNE flat dot plot and density dot plot (1 dot = 1 cell)
95 tsne.plot <- data.frame(x = tsne.data[, 1], y = tsne.data[, 2])
```

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

You can choose the resulting numbers 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

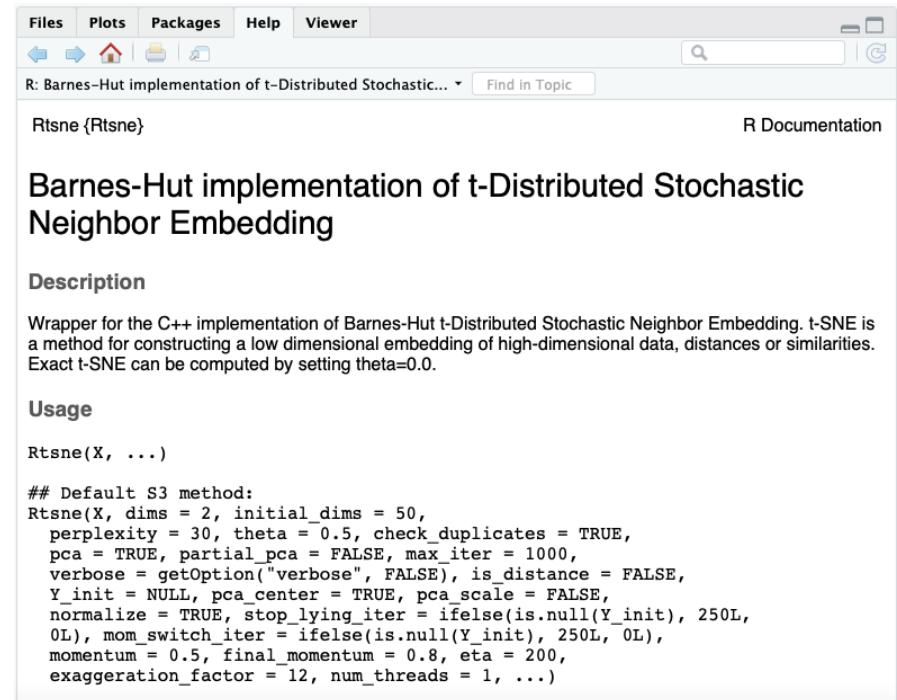
# For help pages for tools type...

?Rtsne -> enter

?umap -> enter

?FlowSOM -> enter

?MEM -> enter



The screenshot shows the RStudio interface with the 'Packages' tab selected in the top menu bar. A search bar at the top right contains the text 'R: Barnes-Hut implementation of t-Distributed Stochastic...'. Below the search bar, the title 'Rtsne {Rtsne}' is displayed, along with the 'R Documentation' link. The main content area is titled 'Barnes-Hut implementation of t-Distributed Stochastic Neighbor Embedding'. It includes a 'Description' section stating that it's a wrapper for the C++ implementation of Barnes-Hut t-SNE, which is used for constructing low-dimensional embeddings of high-dimensional data. It also includes a 'Usage' section with the following R code:

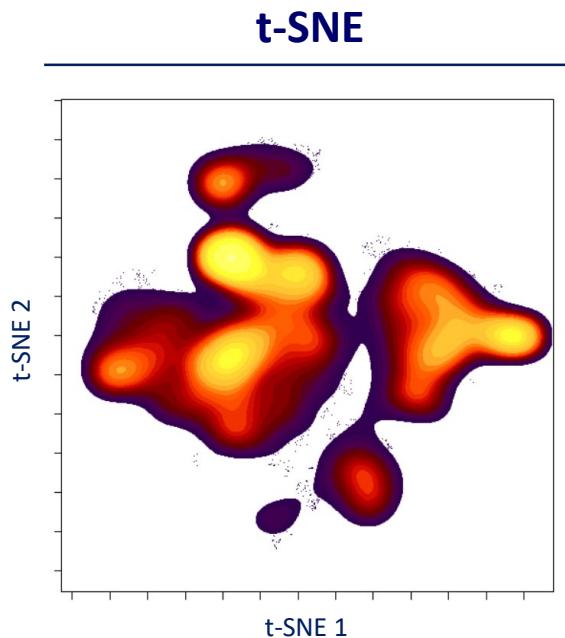
```
## Default S3 method:  
Rtsne(X, dims = 2, initial_dims = 50,  
      perplexity = 30, theta = 0.5, check_duplicates = TRUE,  
      pca = TRUE, partial_pca = FALSE, max_iter = 1000,  
      verbose = getOption("verbose", FALSE), is_distance = FALSE,  
      Y_init = NULL, pca_center = TRUE, pca_scale = FALSE,  
      normalize = TRUE, stop_lying_iter = ifelse(is.null(Y_init), 250L,  
      0L), mom_switch_iter = ifelse(is.null(Y_init), 250L, 0L),  
      momentum = 0.5, final_momentum = 0.8, eta = 200,  
      exaggeration_factor = 12, num_threads = 1, ...)
```

# ...in console to the right of >

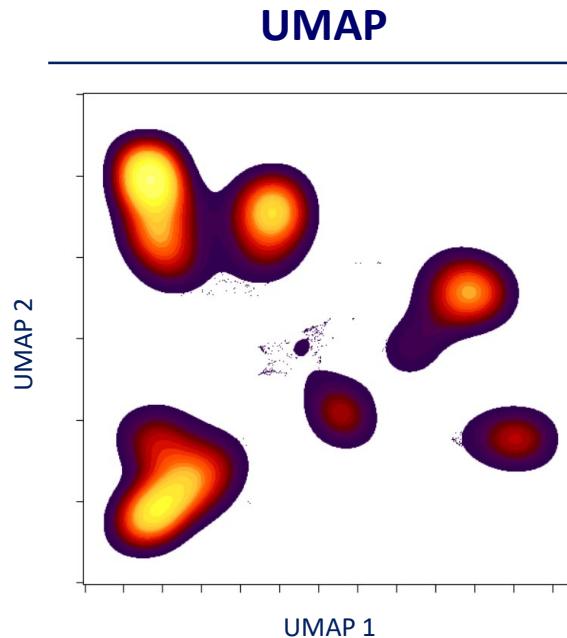
# UMAP (Uniform Manifold Approximation and Projection) is Another Dimensionality Reduction Tool

Superior run times

Emphasis on both global and local structure in the data



vs.



# UMAP Preserves Local and Global Structure

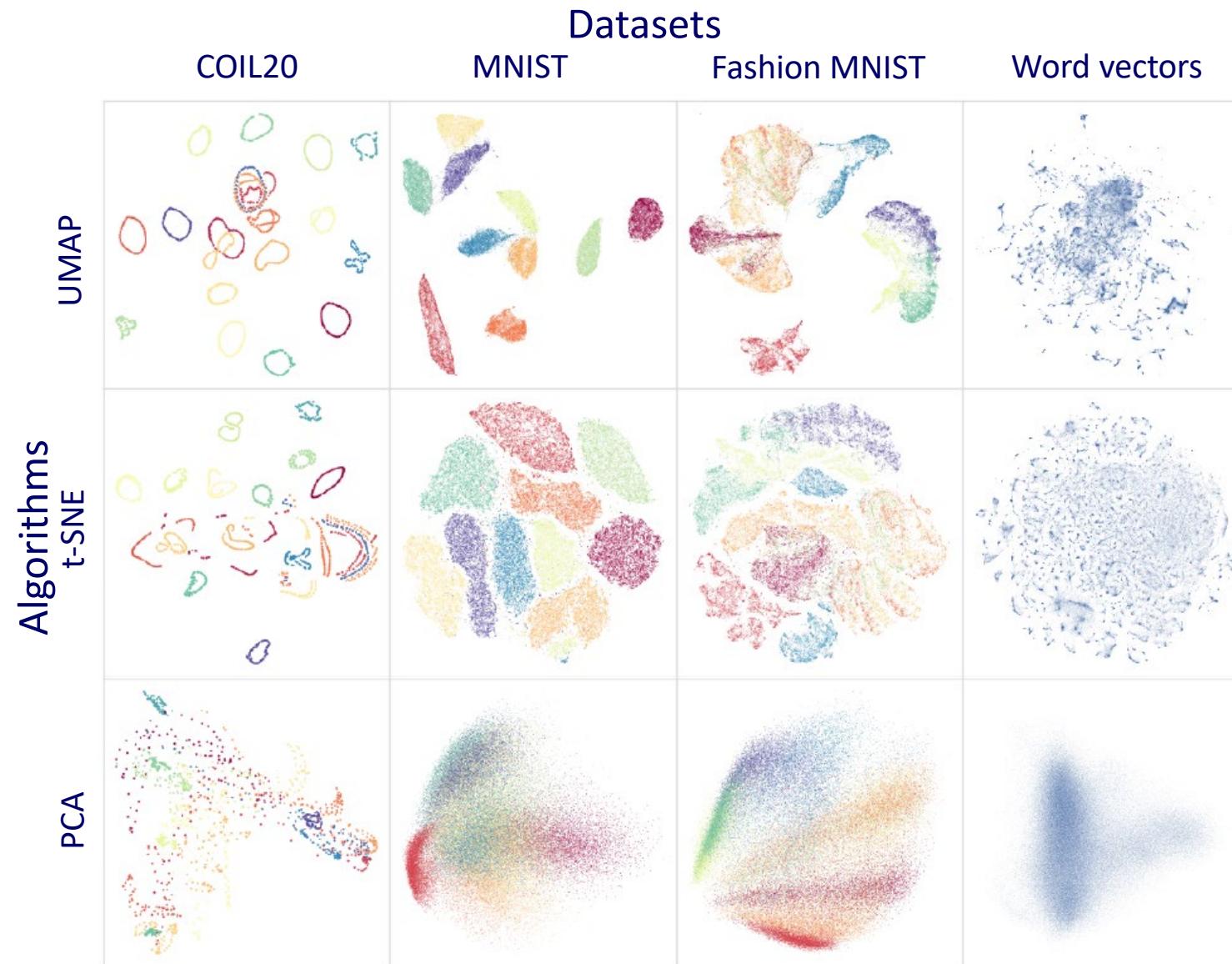


Figure 2: A comparison of several dimension reduction algorithms. UMAP reflects much of the large scale global structure, while also preserving the local fine structure similar to t-SNE.

# 01\_PBMC\_workflow\_example.rmd

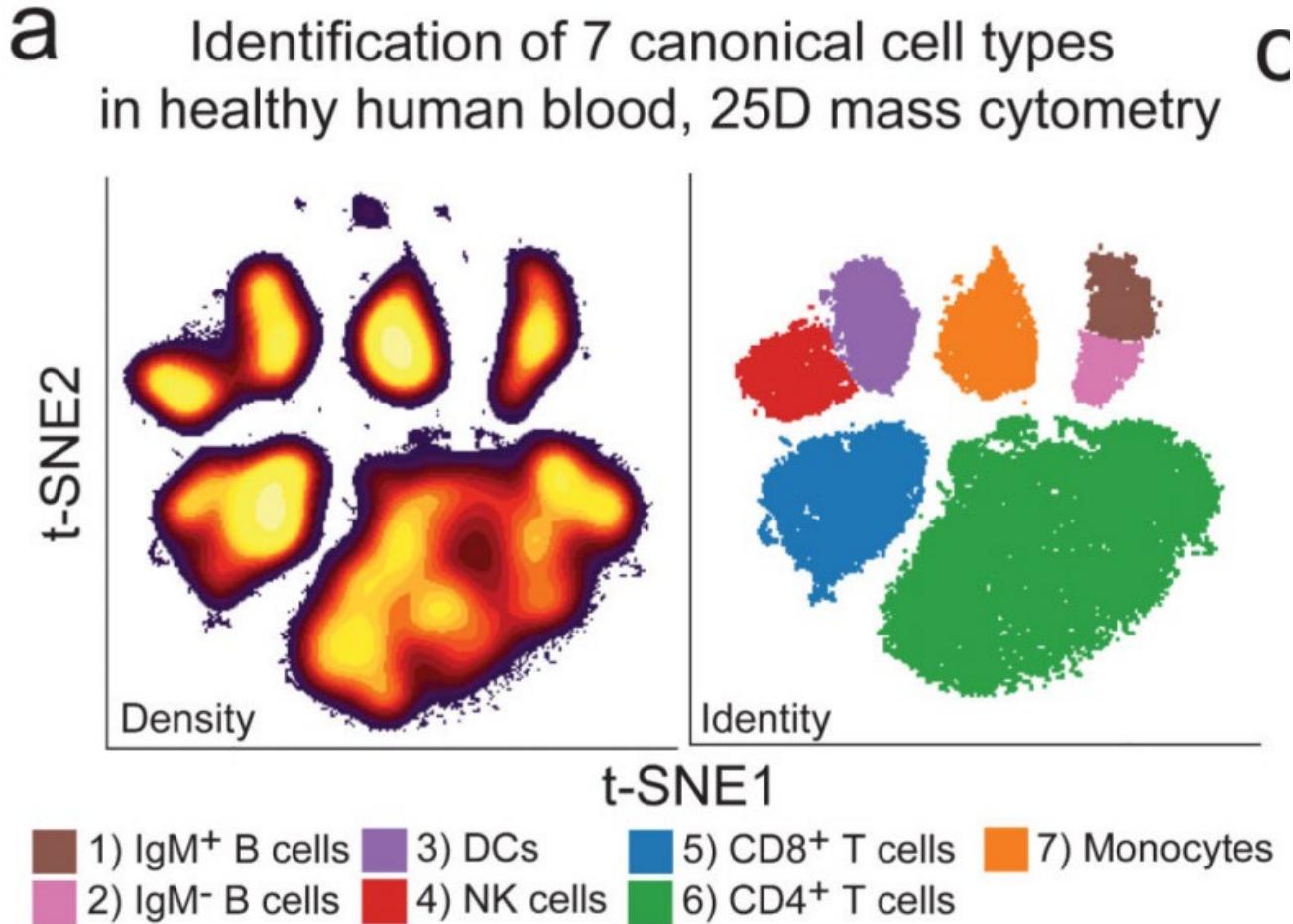
## Run UMAP

```
119 ````{r run_UMAP}
120 # Time ~1 min
121
122 # Run UMAP on all scaled surface markers
123 set.seed(overall_seed)
124
125 # the line below will run UMAP on the data set (to see help page for UMAP
# "UMAP -- enter" in console)
126 # you can view UMAP progress by opening up the console below
127 myumap <-
128   umap(transformed.chosen.markers, # input scaled data
129         n_neighbors = 15,          # number of nearest neighbors to look at,
scales with data set|
130         n_threads = 1,           # this argument makes UMAP reproducible
131         verbose = TRUE)
132 umap.data = as.data.frame(myumap)
133
134
135 ````{r plot_UMAP}
136 # Time <10 sec
137
138 # setting aspect ratio for plots
139 range <- apply(apply(umap.data, 2, range), 2, diff)
140 graphical.ratio <- (range[1] / range[2])
141
142 # UMAP flat dot plot and density dot plot (1 dot = 1 cell)
143 UMAP.plot <- data.frame(x = umap.data[, 1], y = umap.data[, 2])
144
145 # dot plot
146 ggplot(UMAP.plot) + coord_fixed(ratio = graphical.ratio) +
147   geom_point(aes(x = x, y = y), cex = 0.5) + labs(x = "UMAP 1", y = "UMAP 2",
title = "PBMC Data on UMAP
Axe")
148
149 theme_bw() +
```

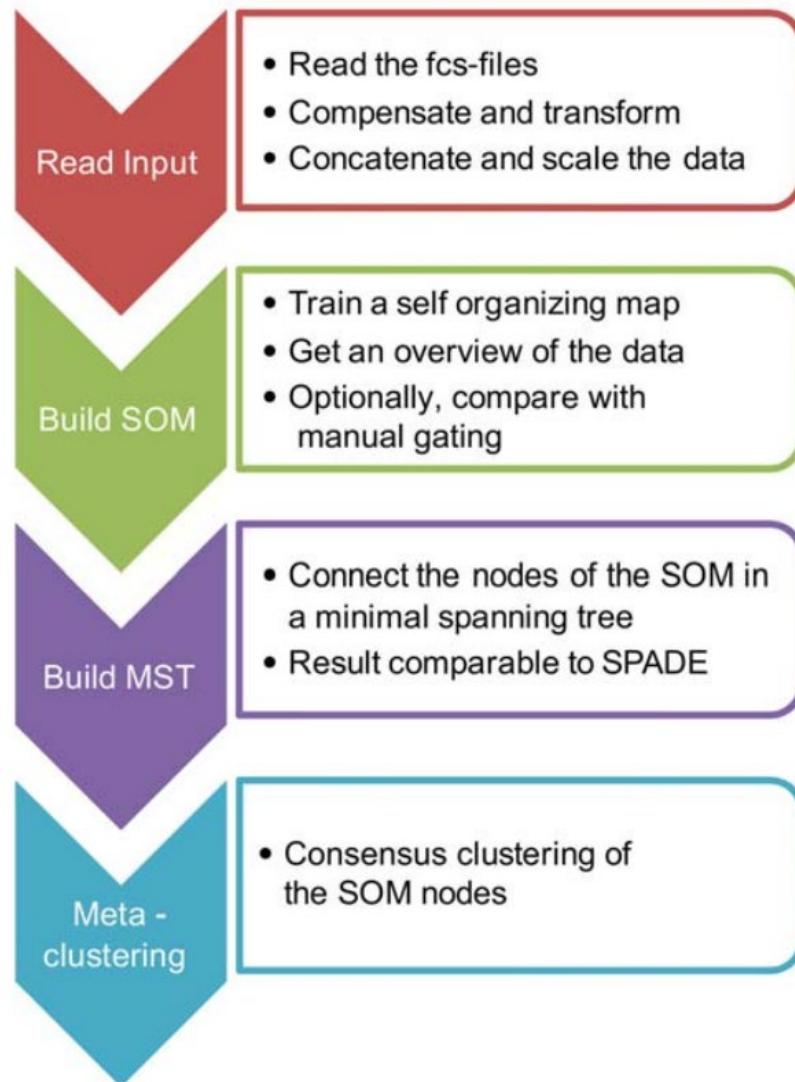
This section will run a UMAP analysis on the PBMC data using set parameters

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

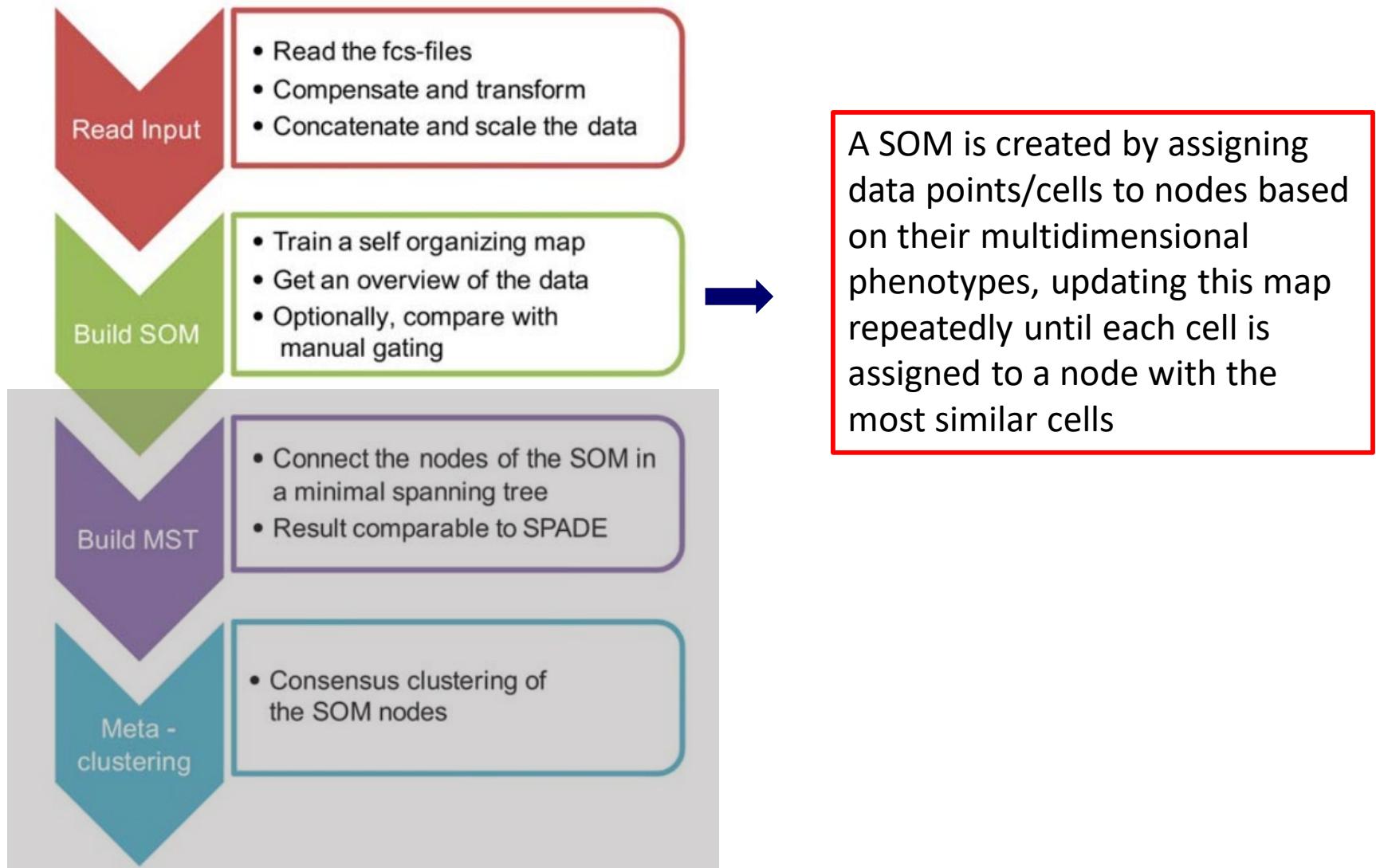
# Clusters can be Identified Based on Dimensionality Reduction Results



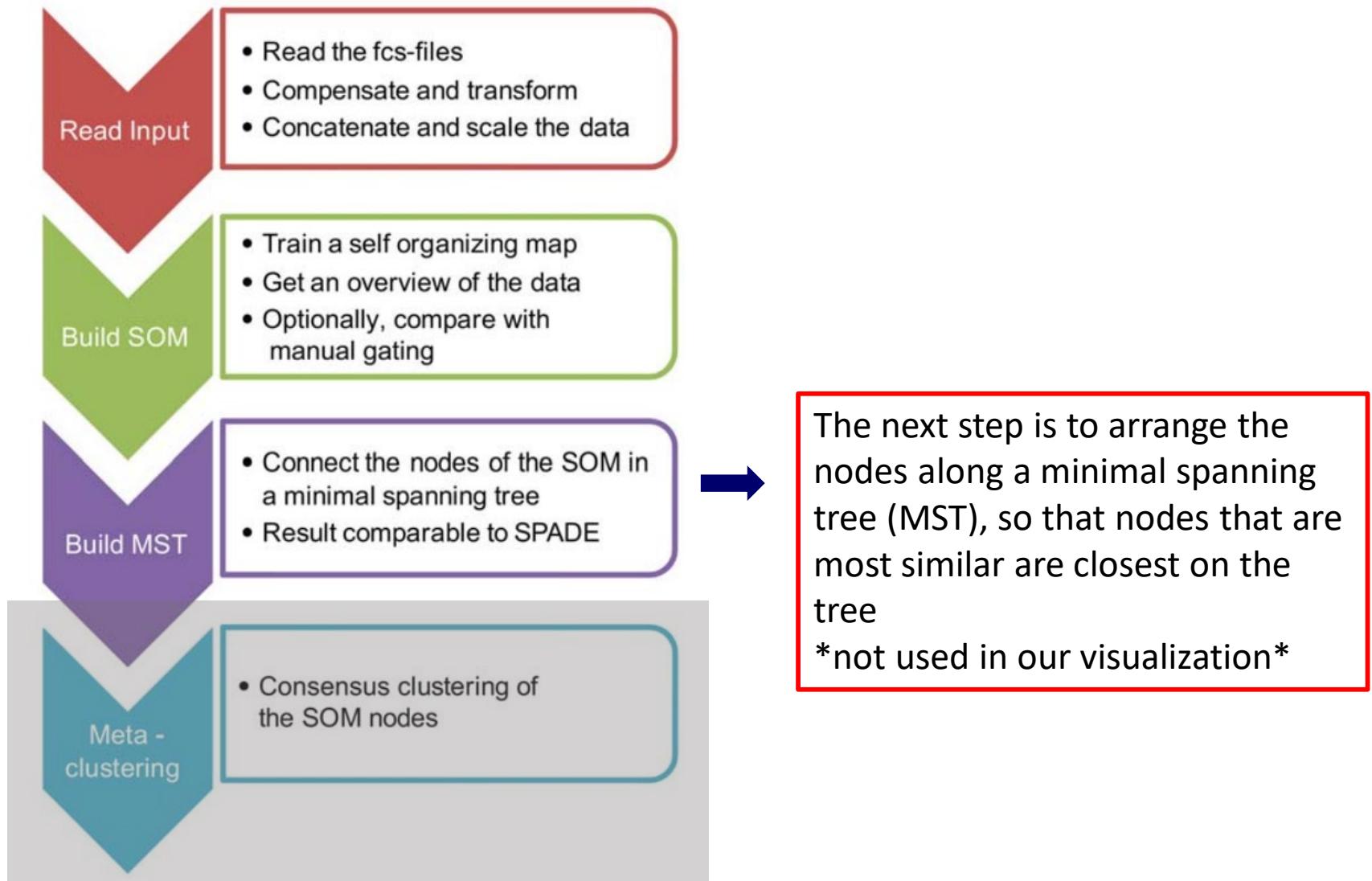
# Clustering with FlowSOM: Self-organizing Maps



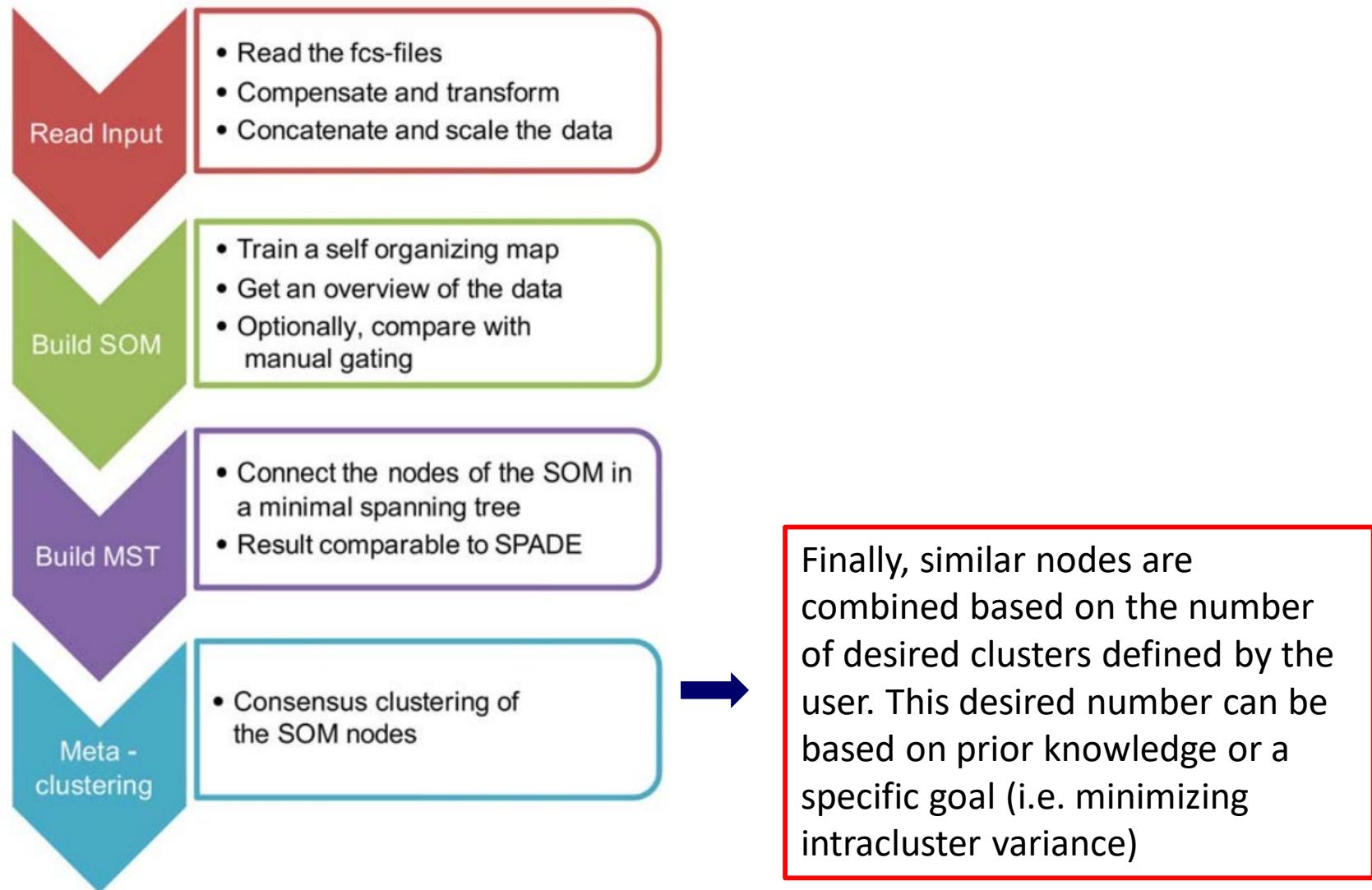
# Clustering with FlowSOM: Self-organizing Maps



# Clustering with FlowSOM: Self-organizing Maps

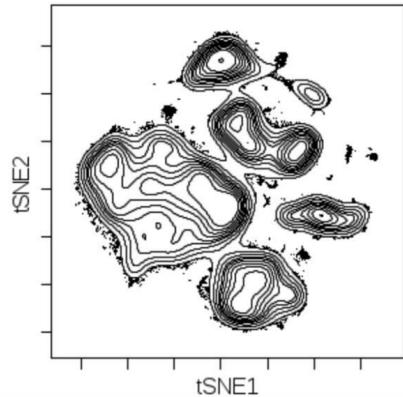


# Clustering with FlowSOM: Self-organizing Maps

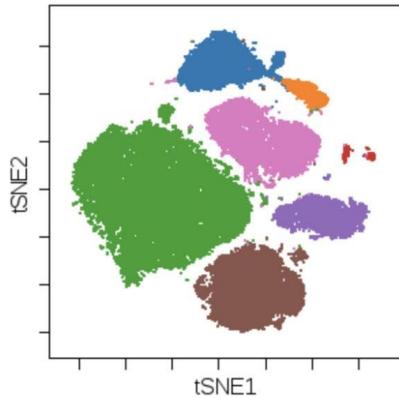


# Clustering with FlowSOM: Self-organizing Maps

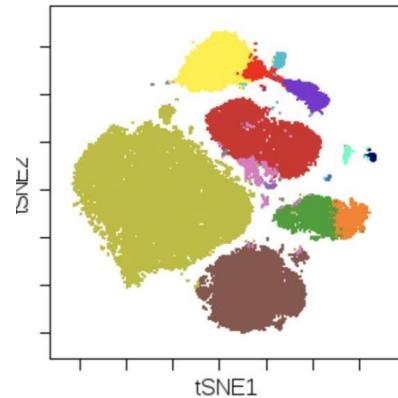
Contour plot of viSNE map



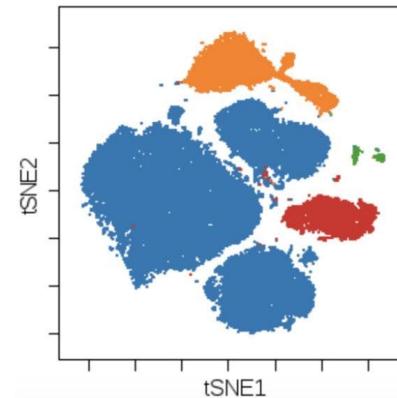
FlowSOM metaclusters overlaid on viSNE map



# metaclusters = 7



# metaclusters = 15



# metaclusters = 4

# 01\_PBMC\_workflow\_example.rmd

## Run FlowSOM

```
166 ````{r run_Flowsom}
167 # Time <10 sec
168
169 # create flowFrame for Flowsom input (using umap axes as input)
170 matrix <- as.matrix(umap.data)
171 metadata <-
172   data.frame(name = dimnames(matrix)[[2]],
173             desc = paste('UMAP', dimnames(matrix)[[2]]))
174 metadata$range <- apply(apply(matrix, 2, range), 2, diff)
175 metadata$minRange <- apply(matrix, 2, min)
176 metadata$maxRange <- apply(matrix, 2, max)
177 flowframe <- new("flowFrame",
178   exprs = matrix,
179   parameters = AnnotatedDataFrame(metadata))
180
181 # implement the Flowsom on the data by running the line below (to see help
182 # for Flowsom, type "Flowsom -- enter" in console)
183 fsom <-
184   Flowsom(
185     flowframe,      # input flowframe
186     colstouse = c(1:2), # columns to use
187     nclus = 10,       # target number of clusters (this can be changed)
188     seed = overall_seed # set seed for reproducibility
189   )
190 Flowsom.clusters <-
191   as.matrix(fsom[[2]][fsom[[1]]$map$mapping[, 1]])
192 ...
193 ````{r plot_clusters}
194 # Time <10 sec
195
196 # plot Flowsom clusters on UMAP axes
197 ggplot(UMAP.plot) + coord_fixed(ratio=graphical.ratio) +
198   geom_point(aes(x=x, y=y, color=Flowsom.clusters), cex = 0.5) +
199   labs(x = "UMAP 1", y = "UMAP 2", title = "Flowsom clustering on UMAP Axes",
200         color = "Cluster") + theme_bw() +
201   guides(colour = guide_legend(override.aes = list(size=4)))+
202   labs(caption = "Data from Diggins et al., Nat Methods 2017, 14: 275-278
203 \nFlow Repository: FR-FCM-ZY63") +
204   theme(panel.grid.major = element_blank(),
205         panel.grid.minor = element_blank())
206 ...
```

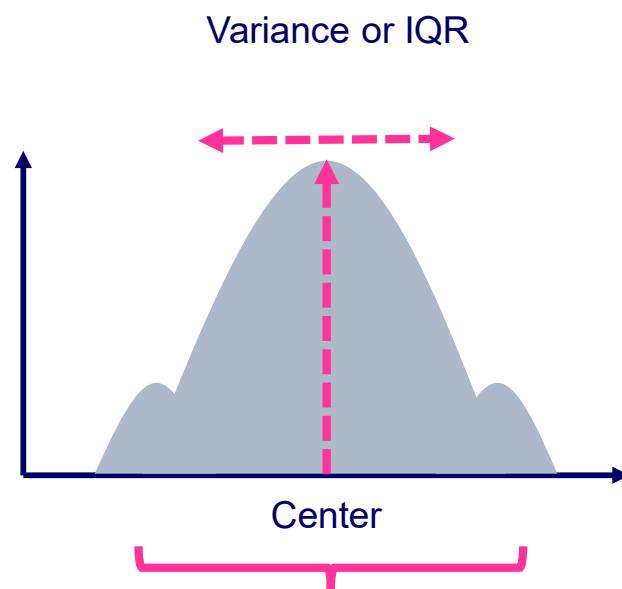
This section performs FlowSOM clustering on the UMAP results

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

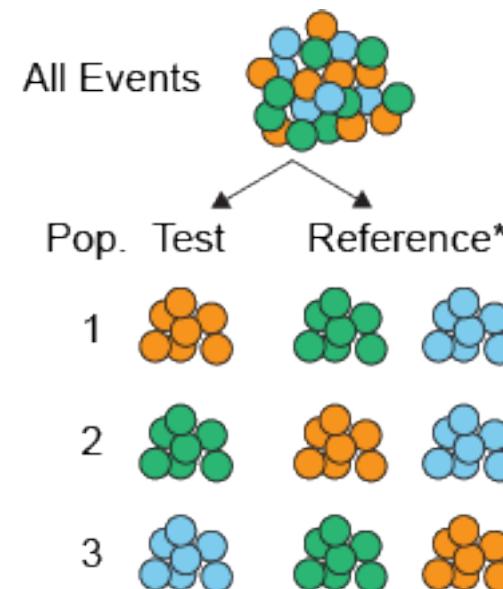
This section plots the identified clusters back onto the UMAP axes and generates a plot (a colored version of the UMAP plot from before)

# 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

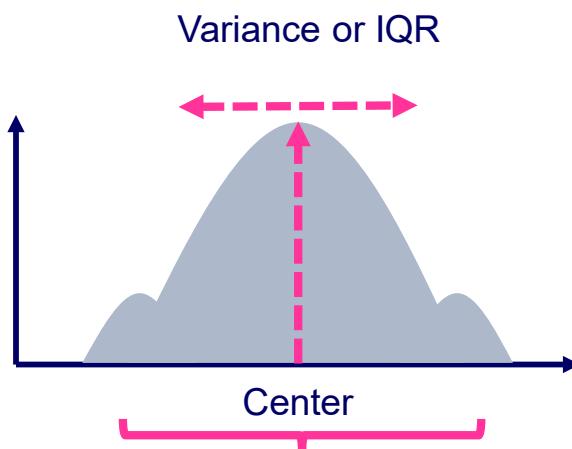


Shape (skewness, symmetry  
# peaks, outliers, etc.)

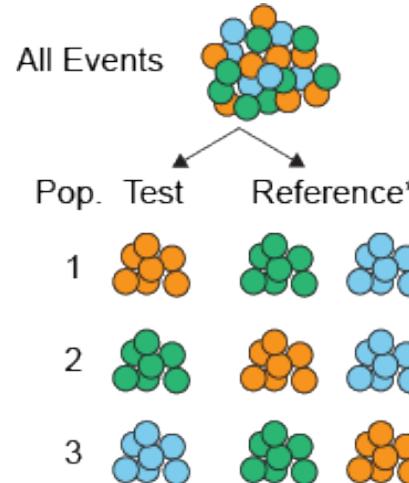


\*All non-population as reference

# MEM Quantifies Relative Enrichment by Combining Magnitude and Interquartile Range



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



MEM label

▲ HLADR<sup>+10</sup> CD20<sup>+9</sup> CD19<sup>+7</sup> IgM<sup>+5</sup> CD34<sup>+3</sup>  
CD45RA<sup>+3</sup> CXCR4<sup>+2</sup> CD47<sup>+2</sup> CD33<sup>+2</sup>  
▼ CD7<sup>-2</sup>

Linear transformation to -10 to +10

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

# 01\_PBMC\_workflow\_example.rmd

## Run MEM

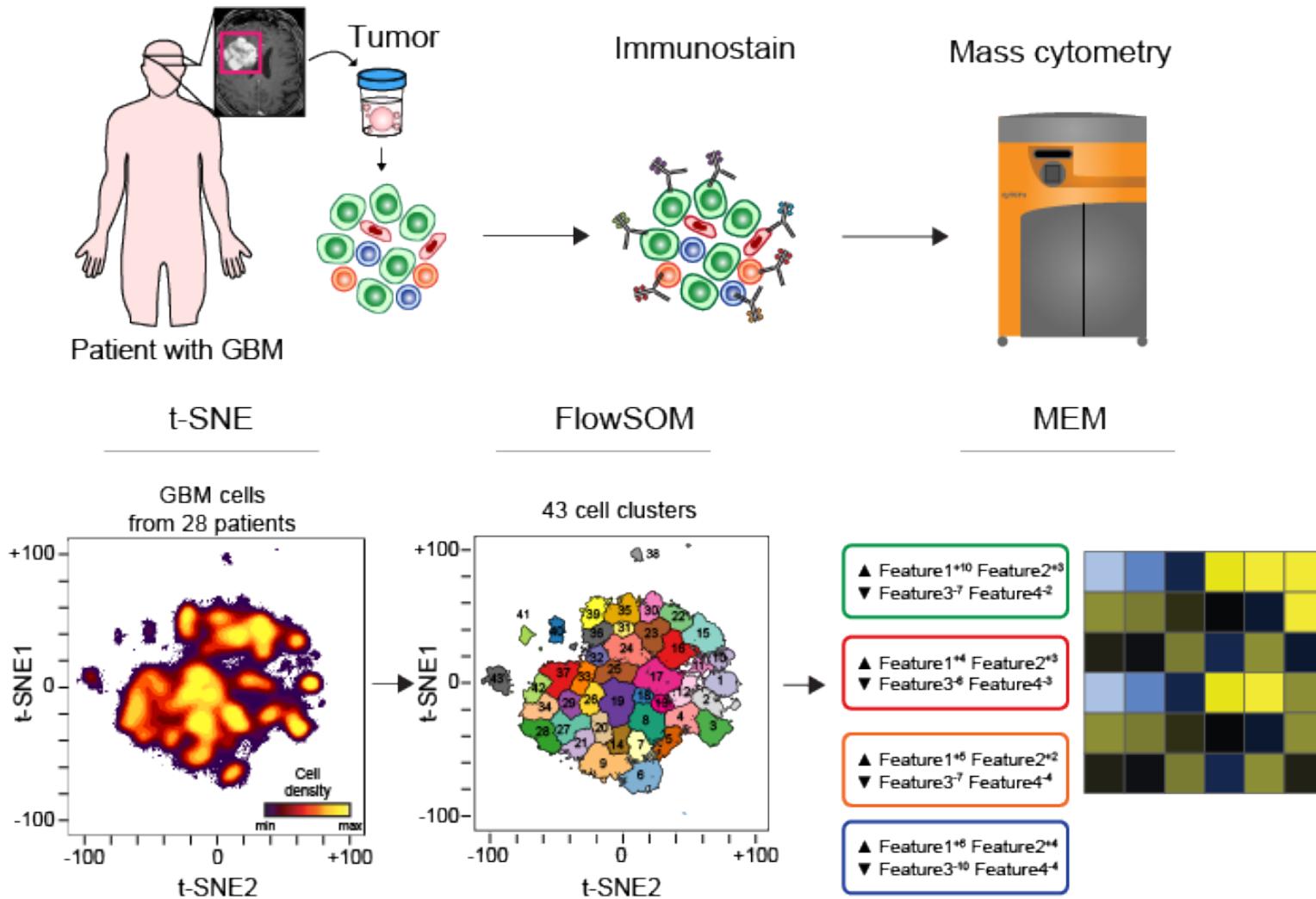
```
207 ````{r run_MEM}
208 # Time ~30 sec
209
210 # Run MEM on the FlowSOM clusters found from using UMAP axes
211 cluster = as.numeric(as.vector((FlowSOM.clusters)))
212 MEM.data = cbind(transformed.chosen.markers, cluster)
213
214 MEM.values = MEM(
215   MEM.data,           # input data (last column must contain cluster values)
216   transform = FALSE,  # data is already scaled in this case
217   cofactor = 1,
218   choose.markers = FALSE,
219   markers = "all",    # use all transformed, chosen markers from previous
220   selection
221   choose.ref = FALSE, # reference will be all other cells
222   zero.ref = FALSE,
223   rename.markers = FALSE,
224   new.marker.names =
225     "CD19,CD117,CD11b,CD4,CD8,CD20,CD34,CD61,CD123,CD45RA,CD45,CD10,CD33,CD69,CD15,CD16,CD44,CD38,CD25,CD3,IgM,HLA-DR,CD56", # rename channels
226   labels
227   file.is.clust = FALSE,
228   add.fileID = FALSE,
229   IQR.thresh = NULL
230 )
231
232 # build MEM heatmap and output enrichment scores
233 build.heatmaps(
234   MEM.values,          # input MEM values
235   cluster.MEM = "both", # dendrogram for columns and rows
236   display.thresh = 3,   # display threshold for MEM scores
237   newwindow.heatmaps = FALSE, # makes txt and PDF files for heatmap and MEM
238   output.files = TRUE,
239   scores
240   labels = TRUE,        # include labels in heatmap
241   only.MEMheatmap = FALSE
242 )````
```

This section performs MEM analysis on the identified FlowSOM clusters

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

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



# Acknowledgements

---

## Irish Lab

Jonathan Irish

## **Sierra Barone**

Todd Bartkowiak

Caroline Roe

Madeline Hayes

## Ihrie Lab

Rebecca Ihrie

## **Justine Sinnaeve**

Akshitkumar Mistry

Asa Brockman

Laura Winalski

Bret Mobley

Ethan Chervonski

## Past Irish Lab Members

Nalin Leelatian

## **Kirsten Diggins**

## **Jocelyn Gandelman**

Allison Greenplate

Deon Dixie

Cara Wogsland