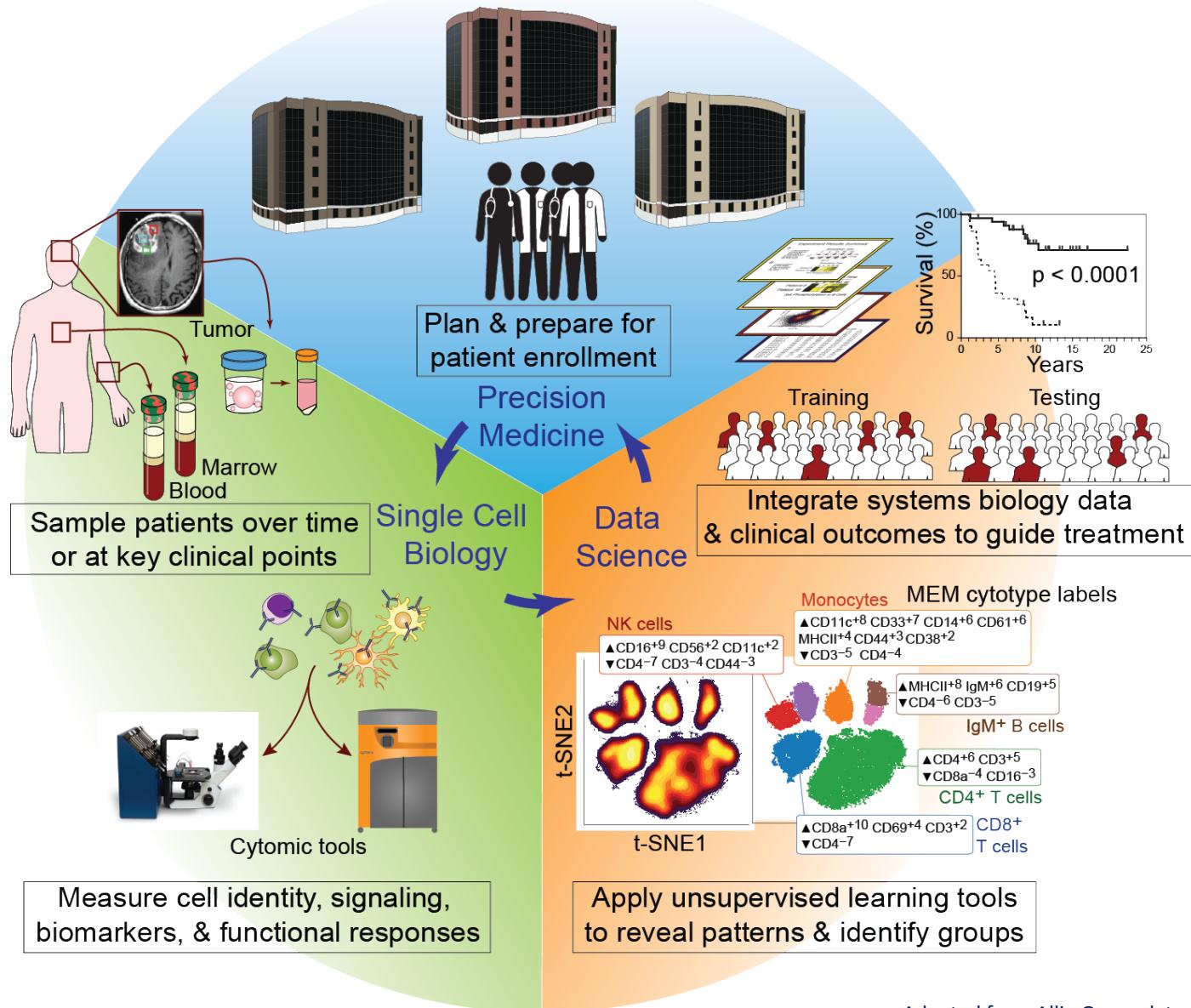


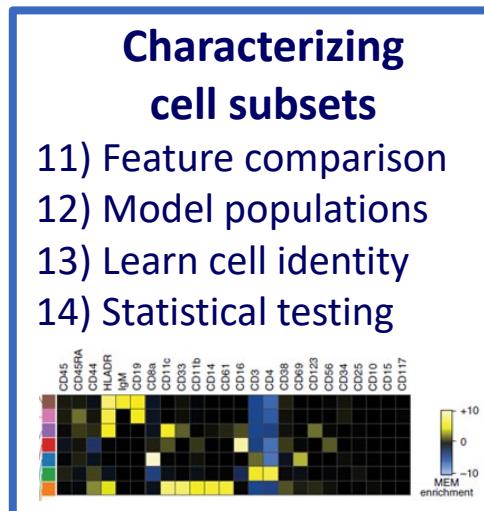
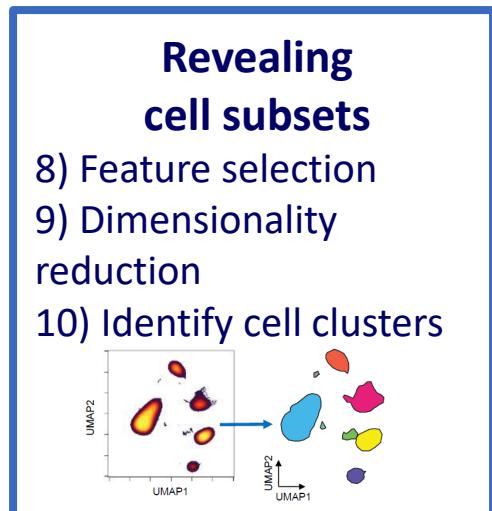
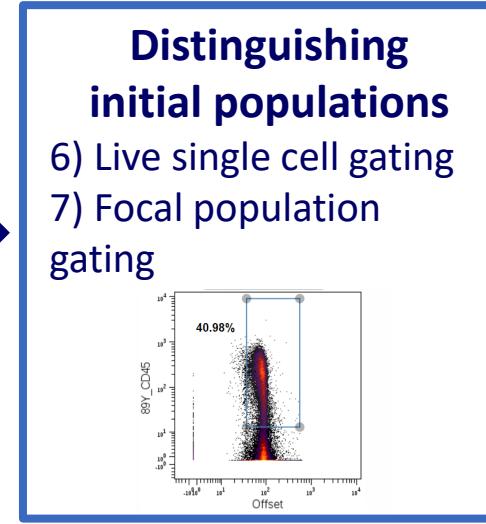
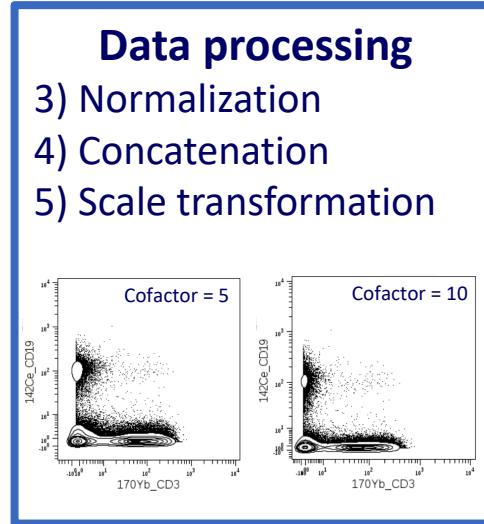
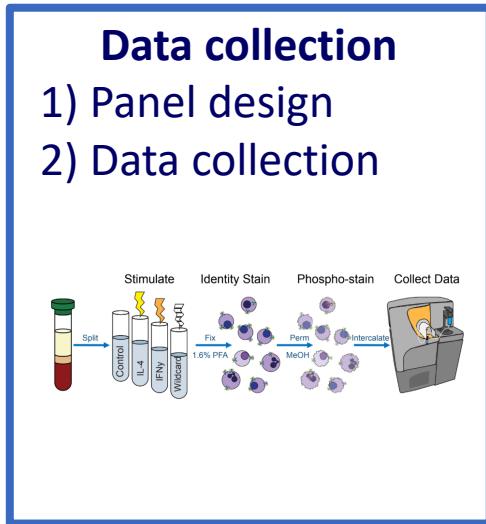
High Dimensional Data Analysis for Flow Cytometry

Sierra Barone
Justine Sinnaeve
12/15/2019

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



Flow Cytometry Workflow from Data Collection to Deep Analysis

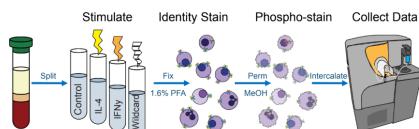


How much can be automated?
How do we select tools and use them well?

Some Preprocessing is Necessary Before Data Analysis

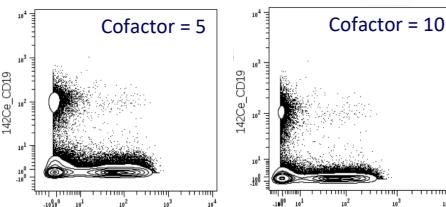
Data collection

- 1) Panel design
- 2) Data collection



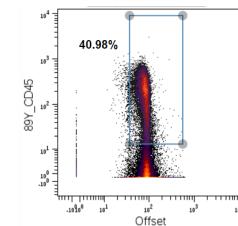
Data processing

- 3) Normalization
- 4) Concatenation
- 5) Scale transformation



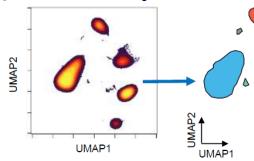
Distinguishing initial populations

- 6) Live single cell gating
- 7) Focal population gating



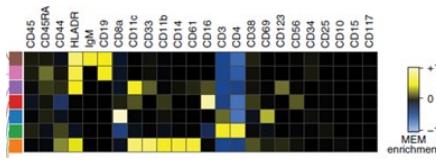
Revealing cell subsets

- 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



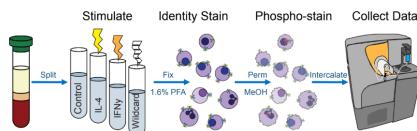
How much can be automated?

How do we select tools and use them well?

Scale Transformation Impacts Data Analysis

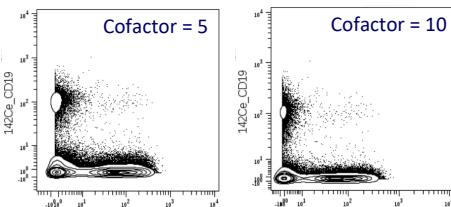
Data collection

- 1) Panel design
- 2) Data collection



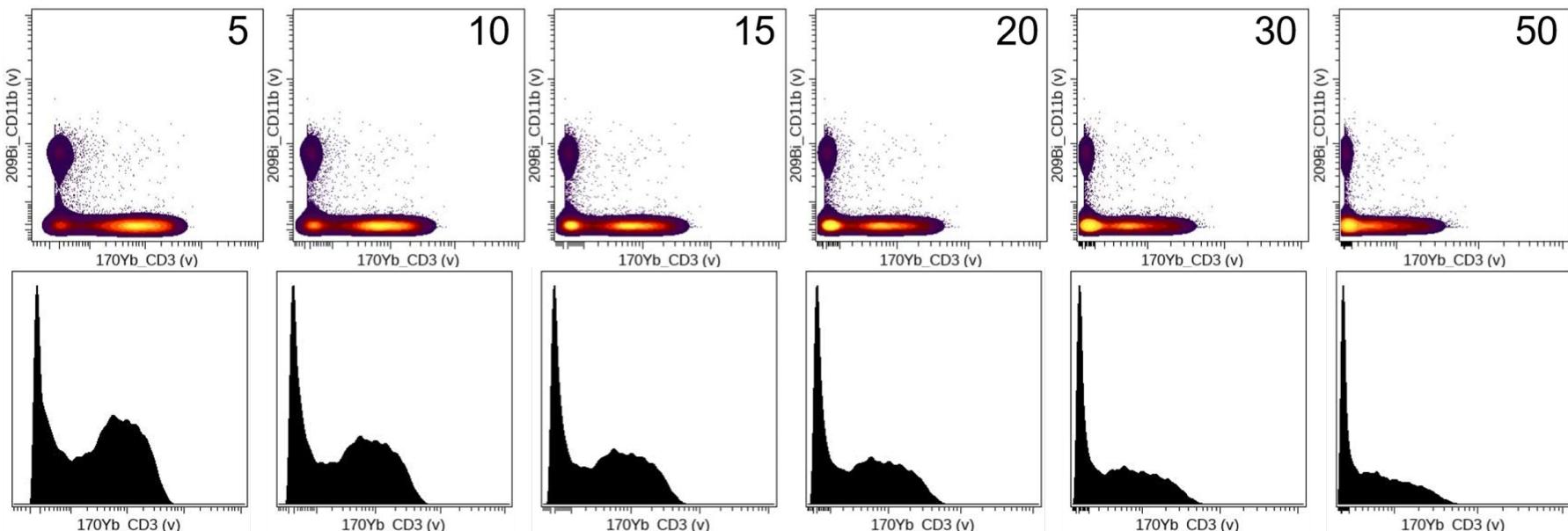
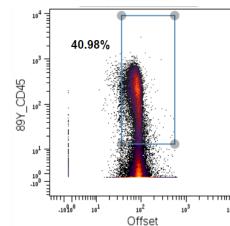
Data processing

- 3) Normalization
- 4) Concatenation
- 5) Scale transformation



Distinguishing initial populations

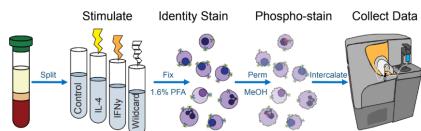
- 6) Live single cell gating
- 7) Focal population gating



Biaxial Gating is Used to Identify Live Cells of Interest

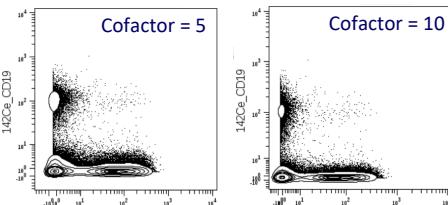
Data collection

- 1) Panel design
- 2) Data collection



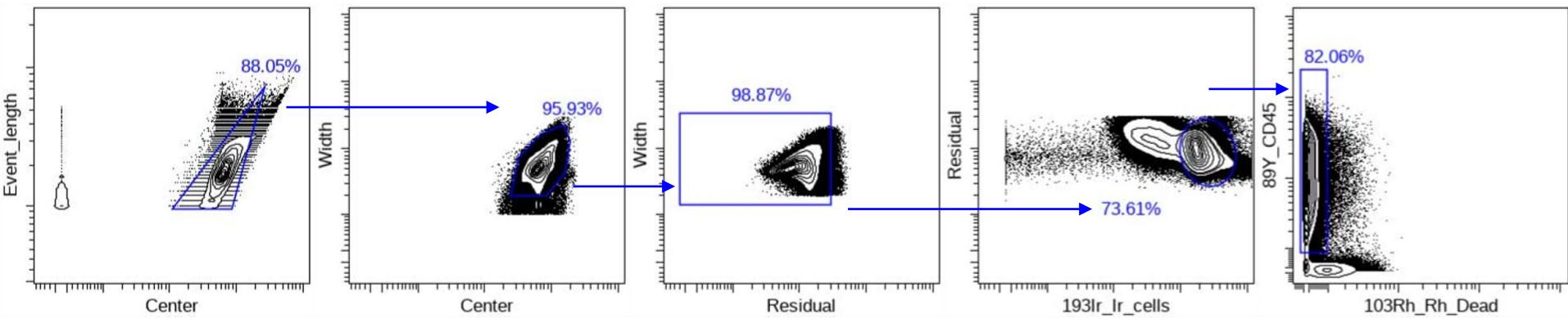
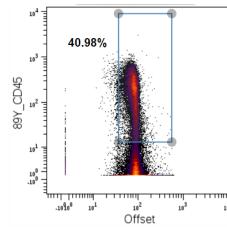
Data processing

- 3) Normalization
- 4) Concatenation
- 5) Scale transformation

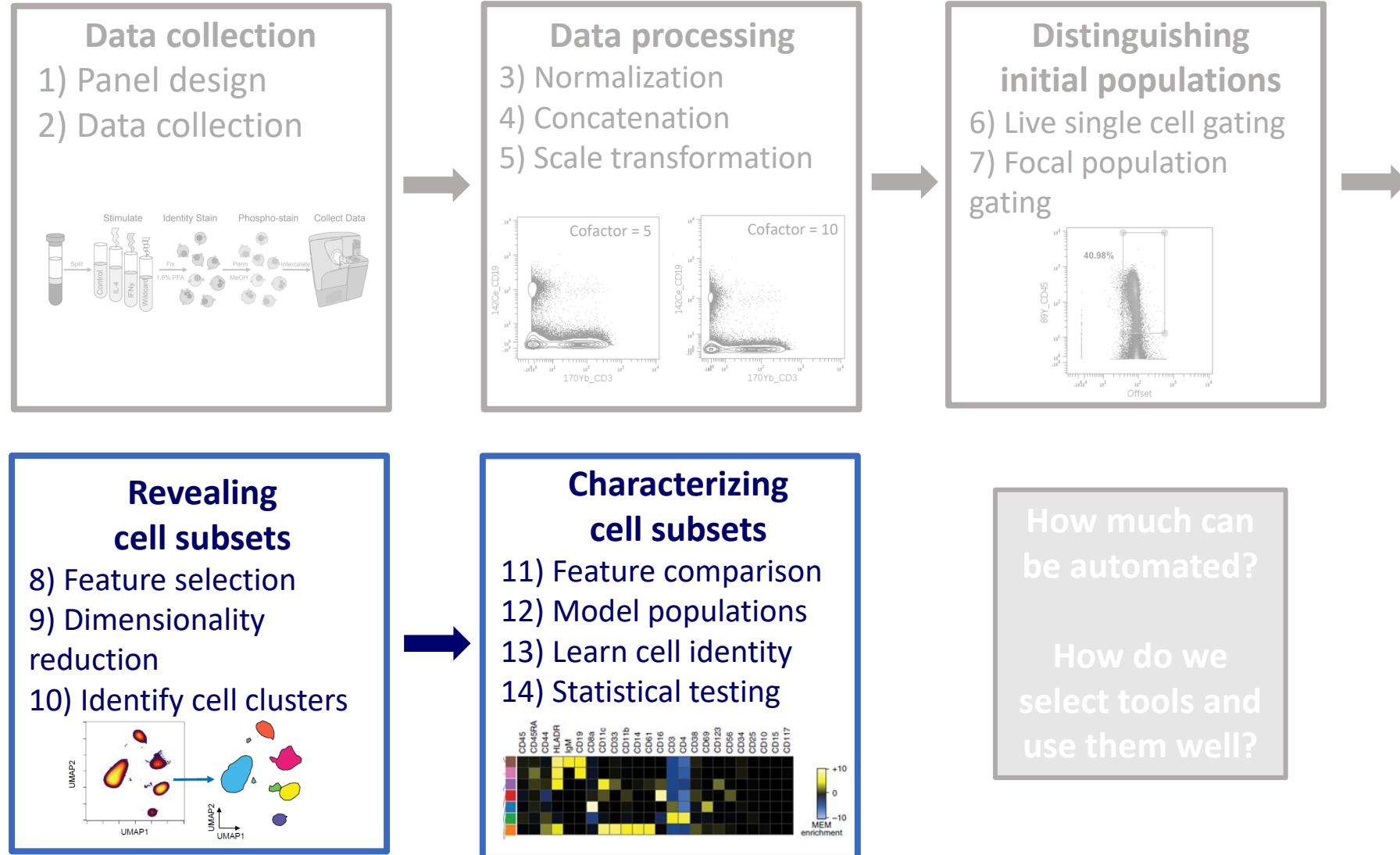


Distinguishing initial populations

- 6) Live single cell gating
- 7) Focal population gating



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



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>

No description, website, or topics provided.

28 commits 1 branch 0 releases 1 contributor View license

Branch: master New pull request Create new file Upload files Find File Clone or download

sierrabarone condensed code and updated plots

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

Latest commit 3346f1f 6 days ago

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)

3 example analysis scripts (.rmd)

Data files (.fcs)

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

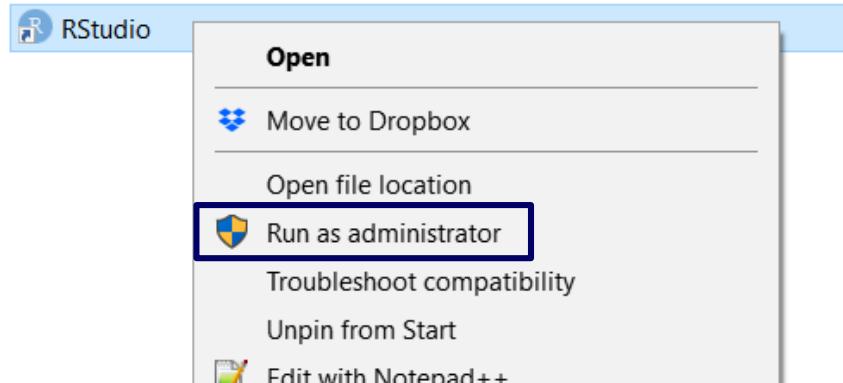
Documentation files (.rd, .md)

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

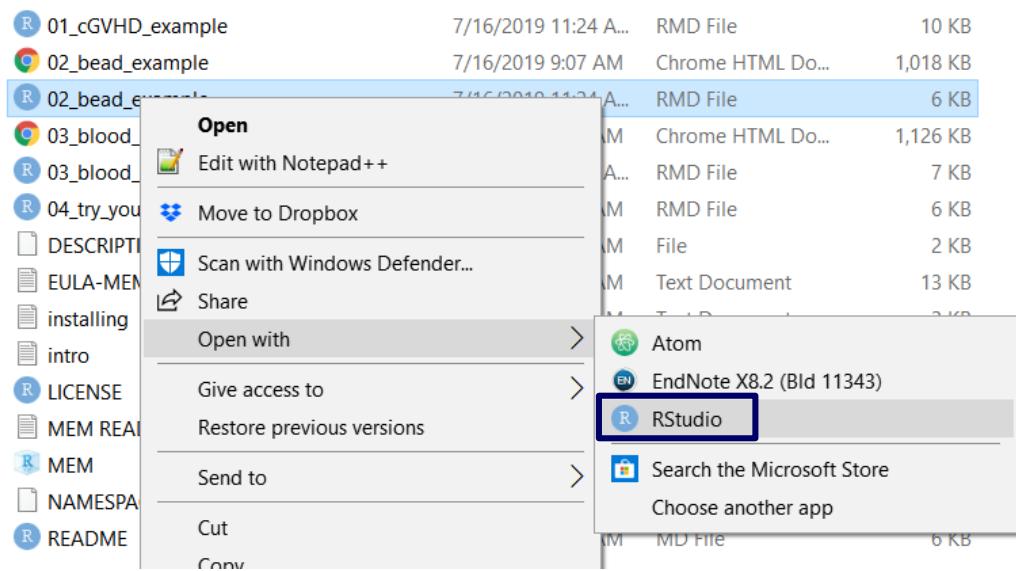
*make sure you unzip downloaded folder

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

The screenshot shows the RStudio interface. On the left, a code editor window displays a script named '01_PBMC_workflow_example.Rmd'. The script contains R code for setting up packages, reading FCS files, and combining data. On the right, a terminal window shows the command-line history of the session, including commands like `setwd`, `lapply`, and `combined.data`.

```
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
```

```
> 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 tab. It displays the global environment, showing variables like 'combined.data' and 'transformed.chosen.markers'. Below that, it shows the user library with a list of installed packages and their details.

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

The screenshot shows the RStudio interface. On the left is the script editor with code in R Markdown format. The code includes sections for setup, loading libraries, reading FCS files, and preparing data. The preview pane on the right shows the rendered version of the script.

```
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
```

The screenshot shows the RStudio console window with the execution history of the script. It includes commands for reading FCS files, combining data, and transforming it for downstream analysis.

```
> 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
>
```

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.

The screenshot shows the RStudio environment pane with a list of installed packages. The list includes packages like ape, askpass, assertthat, backports, base64enc, BH, bibtex, Biobase (selected), BiocGenerics, BiocInstaller, BiocManager, BiocParallel, BiocVersion, biocViews, bit, bit64, bitops, and bmp.

Package	Description	Version
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
<input checked="" type="checkbox"/> Biobase	Biobase: Base functions for Bioconductor	2.44.0
<input checked="" type="checkbox"/> 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

Console

Plots, Files, Help, etc.

Working Script and Code

The screenshot shows the RStudio interface. On the left is the 'Script Editor' window containing R code for a PBMC workflow. On the right is the 'Console' window where the same code is being run, showing the command history and output.

```
01_PBMC_workflow_example.Rmd
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 left is the 'Environment' window displaying global variables and their values. On the right is the 'Packages' window showing the installed package list.

Environment

Global Environment	Value	Type
combined.data	49651 obs. of 46 variables	data
data	List of 7	
transformed.chos...	49651 obs. of 25 variables	data
values		
files	chr [1:7] "CD4Tcells_PBMC.fcs" "CD8Tcells_PBMC.f...	character
overall_seed	43	integer

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
bitops		0.4.2
bmp	Read Windows Bitmap (BMP) Images	2.44.0
		0.30.0
		1.30.0
		1.30.4
		1.18.0
		3.9.0
		1.52.0
		1.1-14
		0.9-7
		1.0-6
		0.3

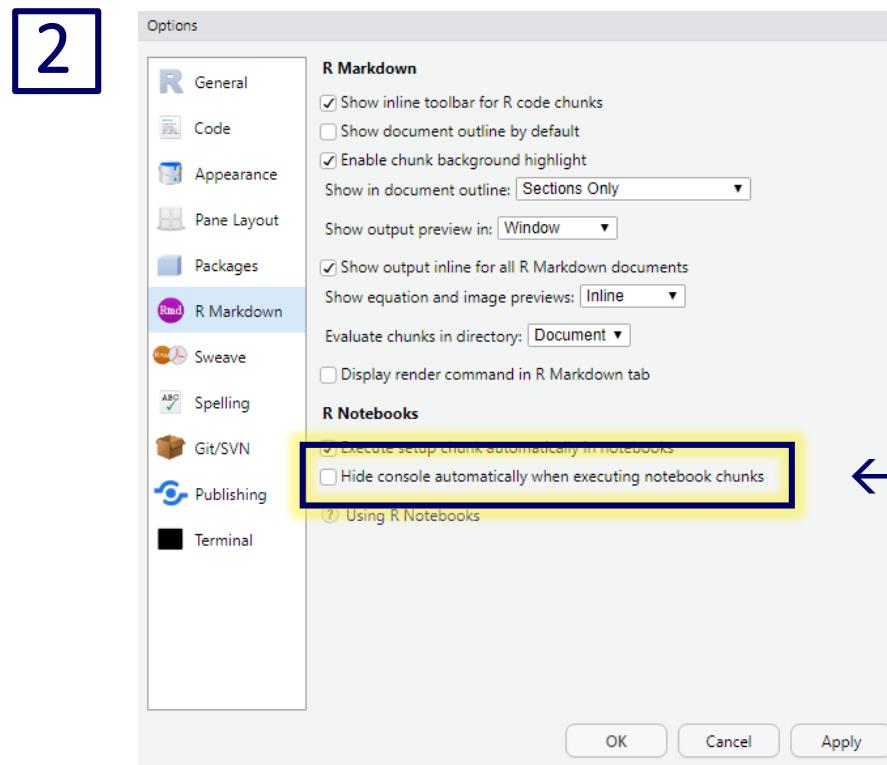
In this window, you can see the code running. Errors and warnings will display here. You can type in the console without changing the base code above.

Plots, Files, Help, etc.

*make sure console is open

Make sure console stays open

1 Tools → Global Options



← Uncheck this option if checked

Working Script and Code

The screenshot shows the RStudio interface. On the left is a code editor window titled "01_PBMC_workflow_example.Rmd" containing R code for setting up and loading libraries, reading FCS files, and combining data. Below it is a "Console" window showing the execution of the R code, including the creation of a "combined.data" object and its 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
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,
54
55
56
57
58
59
60
61 Data Analysis Workflow Example on PBMC Data t-SNE UMAP FlowSC
```

```
> 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 with the "Environment" and "Packages" panes highlighted by a red border. The "Environment" pane lists objects such as "combined.data", "data", "transformed.chosen.markers", "values", "files", and "overall_seed". The "Packages" pane lists various Bioconductor packages and their versions.

Environment

Object	Type	Description
combined.data	Global Environment	49651 obs. of 46 variables
data	Global Environment	List of 7
transformed.chosen.markers	Global Environment	49651 obs. of 25 variables
values	Global Environment	
files	Global Environment	chr [1:7] "CD4Tcells_PBMC.fcs" "CD8Tcells_PBMC.f...
overall_seed	Global Environment	43

Packages

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

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.

The screenshot shows the RStudio interface with a red border around the Environment tab. The Environment tab is selected, and the Global Environment section is visible. A variable named 'combined.data' is highlighted with a blue arrow icon and a black box, indicating it is the current selection. The variable is described as having 49651 observations and 47 variables. Below this, a list of variables is shown, each with a blue arrow icon and a short description of its type and values.

Variable	Description
Time	num 20323 20783 21644 21851 22059 ...
cell_length	num 30 31 26 38 29 31 31 42 23 36 ...
103(Rh103)Dd	num -0.3927 -0.0993 -0.5367 -0.0518 -0.8357 ...
129(Xe129)Dd	num -0.0235 5.5398 -0.6968 -0.9918 -0.8233 ...
131(Xe131)Dd	num -0.206 -0.655 -0.111 0.217 -0.111 ...
132(Xe132)Dd	num 3.6715 -0.3977 -0.0693 -0.6111 -0.9435 ...
133(Cs133)Dd	num -0.266 -0.608 -0.951 -0.888 -0.237 ...
134(Xe134)Dd	num -0.903 0.773 3.725 -0.551 -0.827 ...
136(Xe136)Dd	num 1.951 2.22 -0.506 -0.48 -0.748 ...
139(La139)Dd	num -0.661 -0.903 -0.769 -0.678 -0.525 ...
141(Pr141)Dd	num -0.4805 -0.0504 -0.3028 -0.9314 -0.8542 ...
CD19-142(Nd142)Dd	num -0.362 -0.313 1.434 -0.226 -0.841 ...
CD117-143(Nd143)Dd	num -0.651 -0.423 -0.527 10.162 -0.711 ...
CD11b-144(Nd144)Dd	num 11.231 -0.643 -0.513 7.985 5.553 ...
CD4-145(Nd145)Dd	num 132 204 143 128 319 ...
CD8a-146(Nd146)Dd	num 12.74 4.68 2.48 9.25 36.13 ...
CD20-147(Sm147)Dd	num -0.857 -0.286 -0.29 1.925 -0.12 ...
CD34-148(Sm148)Dd	num -0.852 -0.624 -0.4 1.492 -0.294 ...
CD61-150(Nd150)Dd	num -0.718 -0.0396 -0.5675 1.0479 -0.9467 ...
CD123-151(Eu151)Dd	num -0.476 -0.794 -0.624 0.736 -0.392 ...

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 packages and reading FCS files. On the right, a "Console" tab is active, showing the command-line history of the session, including commands for reading FCS files and combining data.

```
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
56 # choose channels with markers to use for downstream analysis
57 # transformation with a cofactor of 15
58 transformed.chosen.markers <- combined.data %>
59   select(contains("-"), !contains("Ir")) %>%
60   mutate_all(function(x)
61     asinh(x / 15)) # cofactor here is 15; this can be changed
62
63 # set seed for reproducible results (43 is chosen below)
64 overall_seed = 43
65
```

Console

Environment

The screenshot shows the RStudio interface focusing on the "Environment" and "User Library" panes. The "Environment" pane lists global variables like "combined.data" and "data". The "User Library" pane is highlighted with a red border and lists various Bioconductor packages such as BiocManager, BiocParallel, and BiocVersion.

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.

Name	Description	Version
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.

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 # Check to make sure FCS files, documentation, and MEM code are available  
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 # Check version of R and install new version if available  
23 # This only works for PC users
```

Header

2

Code

3

Open 00_install_tools.rmd and begin installing required packages

Code Section Title

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

OR

Press play to run
entire section of
code

```
```{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

The screenshot shows the RStudio interface with the 'Console' tab selected. The output window displays the following text:

```
Content type 'application/x-gzip' length 8117731 bytes (7.7 MB)
downloaded 7.7 MB

The downloaded binary packages are in
    /var/folders/c1/h9zynjsd34373tq19y470hnc0000gn/T//RtmpIYcGkw downloaded_packages
Update old packages: 'backports', 'BiocManager', 'boxr', 'callr', 'car', 'carData',
  'caTools', 'classInt', 'cli', 'cmprsk', 'curl', 'data.table', 'dbSCAN', 'dendextend',
  'devtools', 'digest', 'DT', 'e1071', 'earth', 'ellipsis', 'FactoMineR', 'future',
  'ggExtra', 'ggfortify', 'ggpubr', 'globals', 'haven', 'hexbin', 'Hmisc', 'hms',
  'htmlTable', 'htmltools', 'htmlwidgets', 'httpuv', 'igraph', 'kernlab', 'KernSmooth',
  'knitr', 'ks', 'lambda.r', 'later', 'listenv', 'maptools', 'Matrix', 'matrixStats',
  'metap', 'mgcv', 'mixSmsn', 'multicool', 'nlme', 'openxlsx', 'pbapply', 'pkgbuild',
  'pkgconfig', 'plotly', 'plotmo', 'plotrix', 'polyspline', 'promises', 'purrr', 'quadprog',
  'quantreg', 'R.oo', 'R.utils', 'R6', 'Rcpp', 'RcppAnnoy', 'RcppArmadillo', 'RcppEigen',
  'RcppParallel', 'rlang', 'rmarkdown', 'rms', 'roxygen2', 'rrcov', 'RSpectra', 'RSQLite',
  'rvest', 'scales', 'SDMTools', 'selectr', 'seriation', 'Seurat', 'shiny', 'shinyFiles',
  'slam', 'sp', 'survival', 'survminer', 'testthat', 'tidyR', 'tidyverse', 'tinytex', 'uwot',
  'VGAM', 'whisker', 'xfun', 'zip'
Update all/some/none? [a/s/n]:
```

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

Working Script and Code

Console

Signs that the code is running

Green bar alongside numbered line of 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
```

Play button turns to a clock or red square

```
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 Terminal x
~/Downloads/irish-data-science-master 2 ↵
```

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")  
...```

```

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)
...```

```

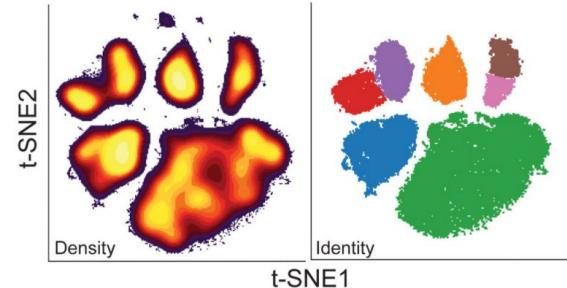
Open 01\_PBMC\_extended\_workflow\_example.rmd  
and work through the example

# 01\_PBMC\_extended\_workflow\_example.rmd

```
01_PBMC_extended_workflow_exa... x Go to file/function Addins ▾
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
4 Rights Reserved; see EULA-MEM.txt for MEM license information"
5 date: "October 2019"
6 output:
7 pdf_document:
8 latex_engine: xelatex
9 html_document:
10 df_print: paged
11 editor_options:
12 chunk_output_type: inline
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 both the UMAP and t-SNE axes to cluster, or group
21 together, the various cell populations. Finally, you will run MEM to see
22 enrichment scores for each of the FlowSOM clusters or populations that have been
23 expert gated. The goal of this exercise is to run several computational tools on
24 a single cell data set to get a feel for the workflow used in the Irish lab as
25 well as compare the various analysis methods. The method for comparison of the
26 cell populations by automated or manual analysis is RMSD.
27
28 ````{r setup, include=FALSE}
29 # Time <10 sec
30
31 # Load all libraries
32 # If you get an error message, you will need to try re-installing packages by
33 # going back to the 00_install_tools.RMD script
34 library(FlowSOM)
35 library(flowCore)
36 library(BioBase)
37 library(ggplot2)
38 library(hexbin)
39 library(MEM)
40 library(tidyverse)
41 library(Rtsne)
42 library(uwot)
43 library(viridis)
44 library(ggExtra)
45 library(RColorBrewer)
46 ````
```

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\_extended\_workflow\_example.rmd

## Data Preparation

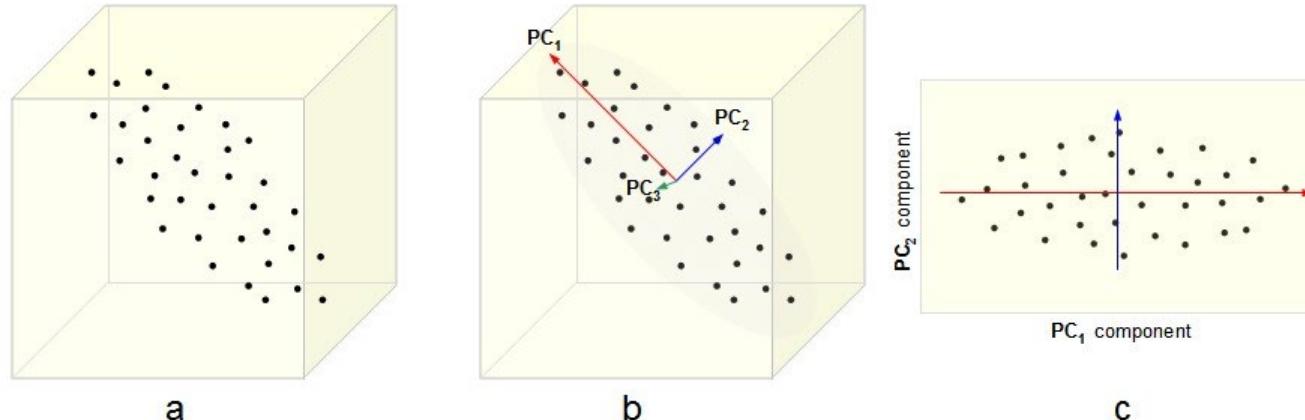
```
48 ````{r data_preparation, warning=FALSE}
49 # Time <10 sec
50
51 # read files into R by setting working directory and directing R to the fcs
52 # files
53 setwd(paste(getwd(), "/datafiles/PBMC", sep = ""))
54 files <- dir(pattern = "*.fcs")
55
56 # convert and combine data for use in downstream analysis
57 data <- lapply(lapply(files, read.FCS), exprs)
58 combined.data = as.data.frame(do.call(rbind, mapply(
59 cbind, data, "cluster" = c(1:length(data)), SIMPLIFY = F)))
60
61 # choose channels with markers to use for downstream analysis and apply
arcsinh
62 # transformation with a cofactor of 15
63 transformed.chosen.markers <- combined.data %>%
 select(contains("-"), -contains("Ir")) %>%
 mutate_all(function(x)
 asinh(x / 15)) # cofactor here is 15; this can be changed
64
65 # set seed for reproducible results (43 is chosen below)
66 overall_seed = 43
67
68 ````
```

Read the data files into R and format for analysis

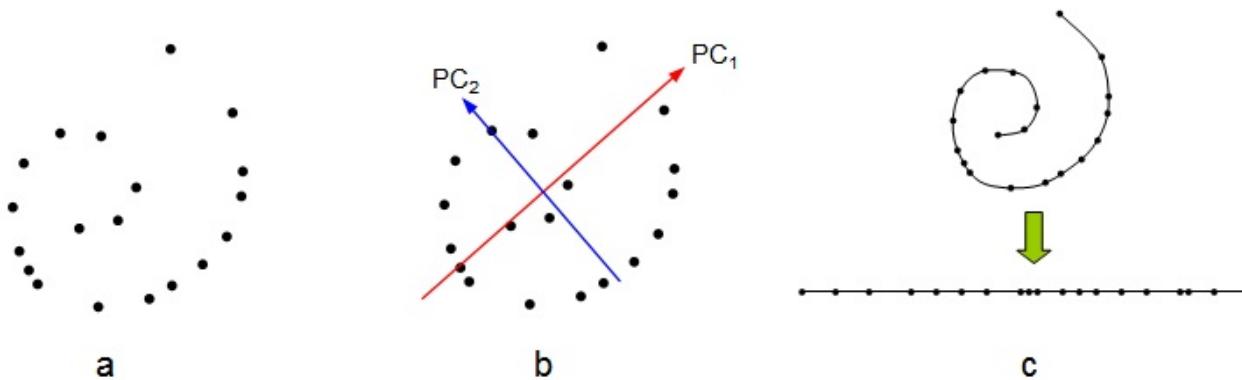
Select channels and scale the data

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.

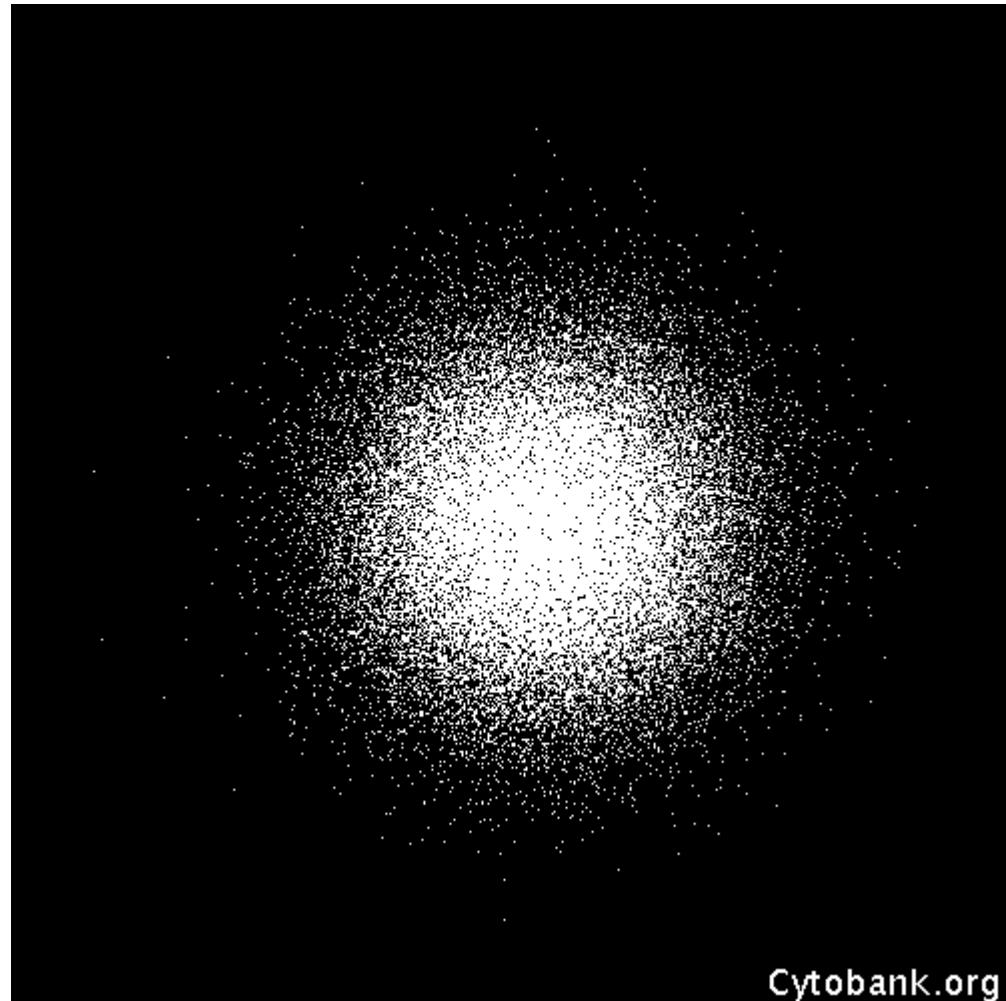


Effects of dimensionality reduction on an inherently non-linear data set. **a)** The original data given as a two-dimensional set. **b)** PCA identifies two PCs as contributing significantly to explain the data variance. **c)** However, the inherent topology (connectivity) of the data helps identify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



Cytobank.org

developed by Laurens van der Maaten and Geoffrey Hinton

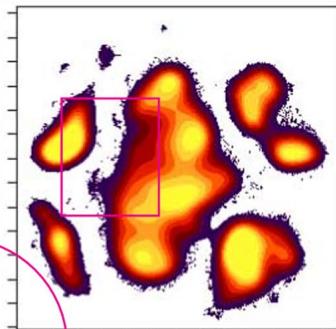
Animation created by Cytobank team from iterations of viSNE / t-SNE using Healthy PBMC (26 features)

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

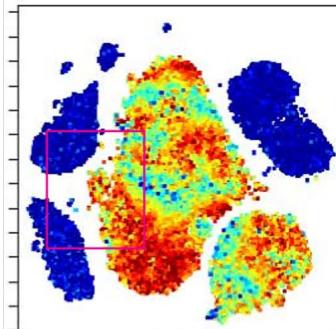
Same map, different information

Healthy Peripheral Blood Mononuclear Cells

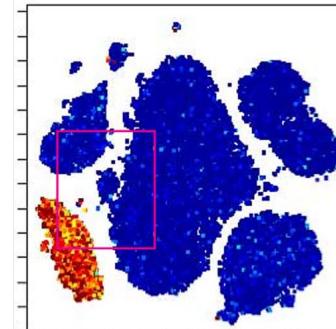
Cell Density



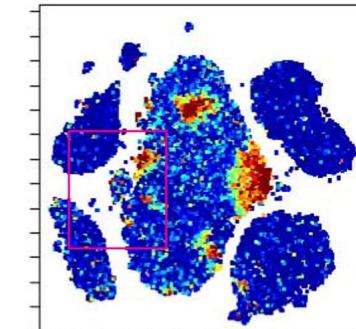
CD3



CD19



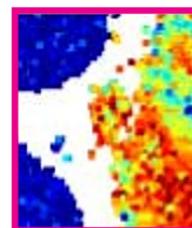
CD25



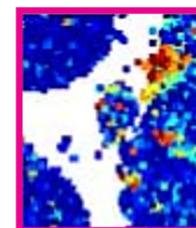
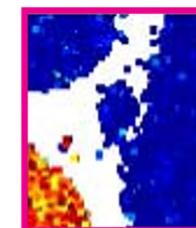
t-SNE 2  
↑  
t-SNE 1

Cell density      Protein expression  
min                  min  
                        max  
max                  max

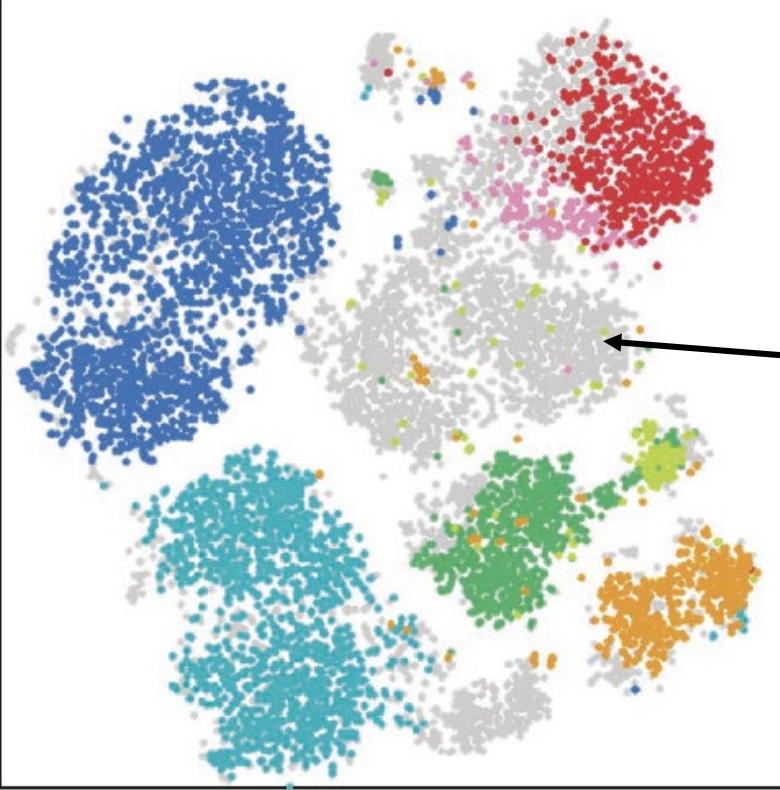
New 2D axes that represent phenotypic similarities of single cells



1 dot = 1 cell



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



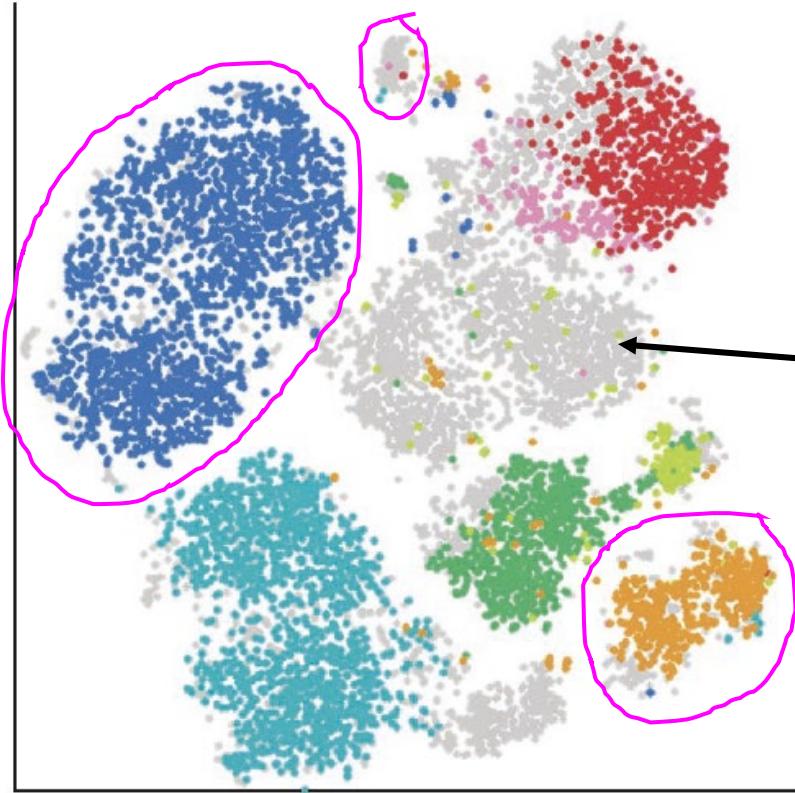
viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia

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\_extended\_workflow\_example.rmd

## Run t-SNE

```
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 dims = 2, # number of final
83 # dimensions
84
85 initial_dims = length(transformed.chosen.markers), # number of initial
86 # dimensions
87
88 perplexity = 30, # perplexity (similar to # of nearest neighbors,
89 # will scale with data sets, cannot be greater than
90 # the number of events minus 1 divided by 3)
91 check_duplicates = FALSE,
92 max_iter = 1000, # number of iterations
93 verbose = TRUE
94)
95 tsne.data = as.data.frame(mytsNE$Y)
96 ```
97
98 ````{r plot_t-SNE}
99 # Time <10 sec
100
101 # setting aspect ratio for plots
102 range <- apply(apply(tsne.data, 2, range), 2, diff)
103 graphical.ratio.tsne <- (range[1] / range[2])
104
105 # t-SNE flat dot plot and density dot plot (1 dot = 1 cell)
106 tsne.plot <- data.frame(x = tsne.data[, 1], y = tsne.data[, 2])
```

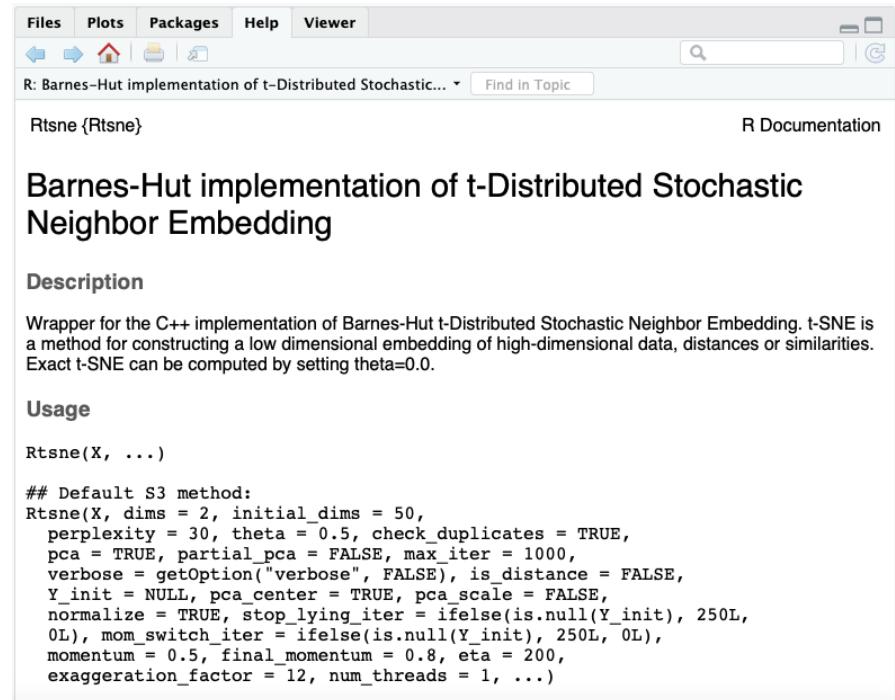
This section will run a t-SNE analysis on the PBMC data with set parameters

You can choose the resulting 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

# For help pages for tools type...

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

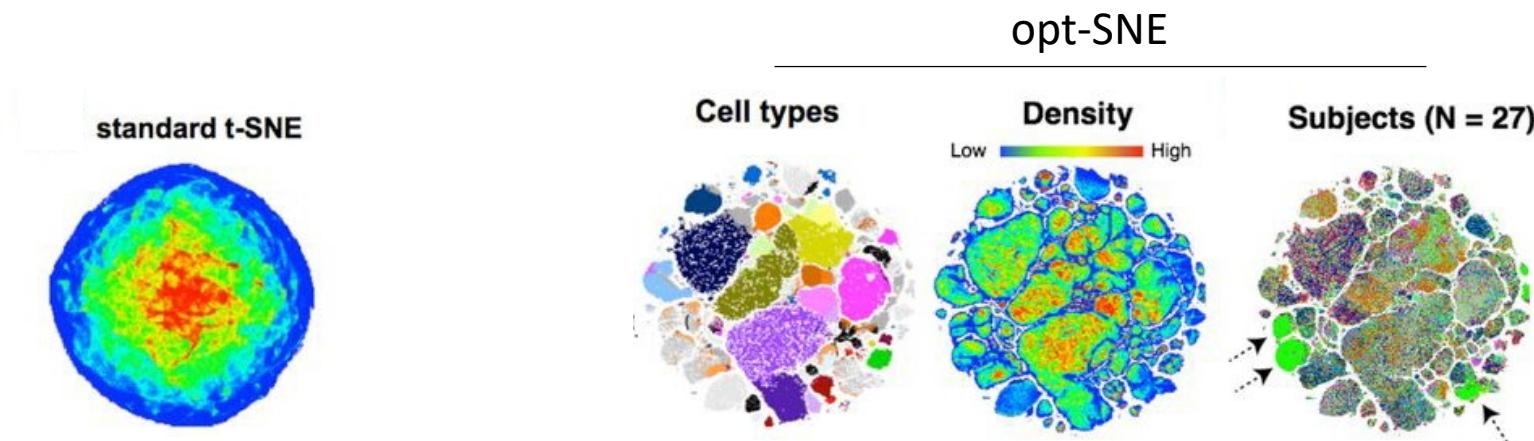


The screenshot shows the RStudio interface with the 'Packages' tab selected in the top menu bar. Below the menu is a toolbar with icons for back, forward, home, and search. The main area displays the R documentation for the 'Rtsne' package. The title is 'Barnes-Hut implementation of t-Distributed Stochastic Neighbor Embedding'. The 'Description' section states 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. The 'Usage' section shows the function signature: `Rtsne(X, ...)`. Below that is the detailed function code:

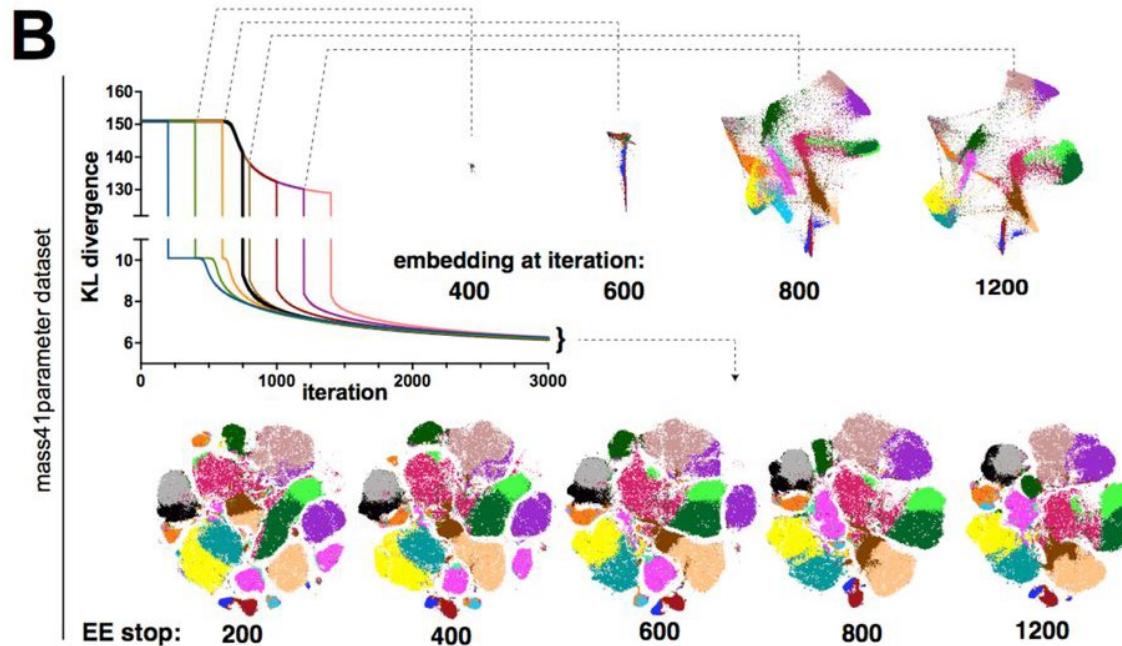
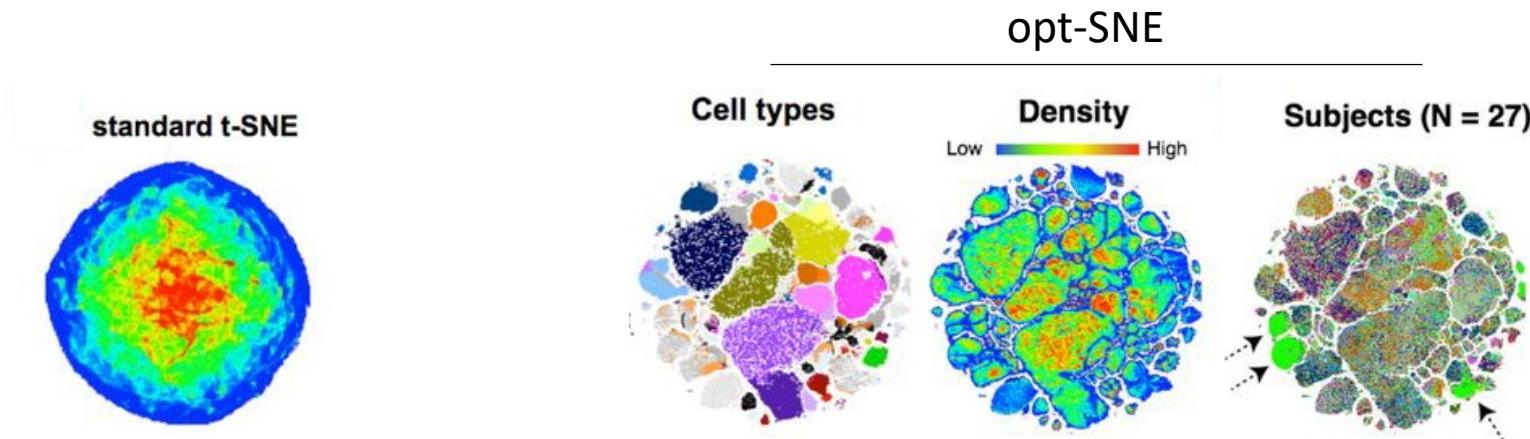
```
Default S3 method:
Rtsne(X, dims = 2, initial_dims = 50,
 perplexity = 30, theta = 0.5, check_duplicates = TRUE,
 pca = TRUE, partial_pca = FALSE, max_iter = 1000,
 verbose = getOption("verbose", FALSE), is_distance = FALSE,
 Y_init = NULL, pca_center = TRUE, pca_scale = FALSE,
 normalize = TRUE, 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 >

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



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

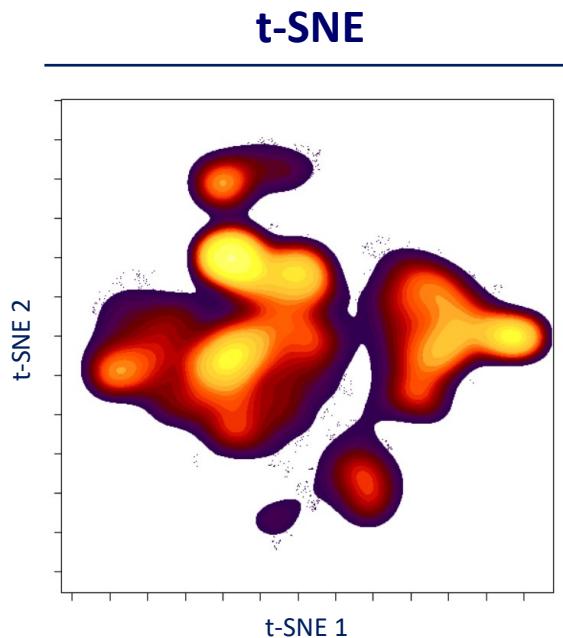


# UMAP (Uniform Manifold Approximation and Projection) is Another Dimensionality Reduction Tool

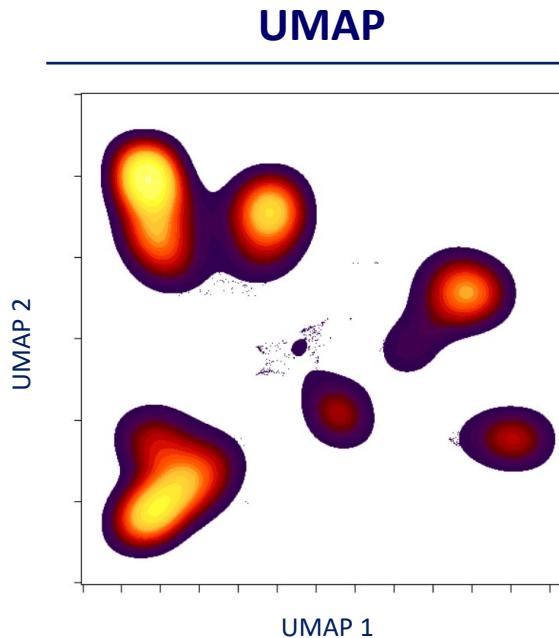
Superior run times

Emphasis on both global and local structure in the data

Ability to map new data onto the low-dimensional projection



vs.



# UMAP Preserves Local and Global Structure

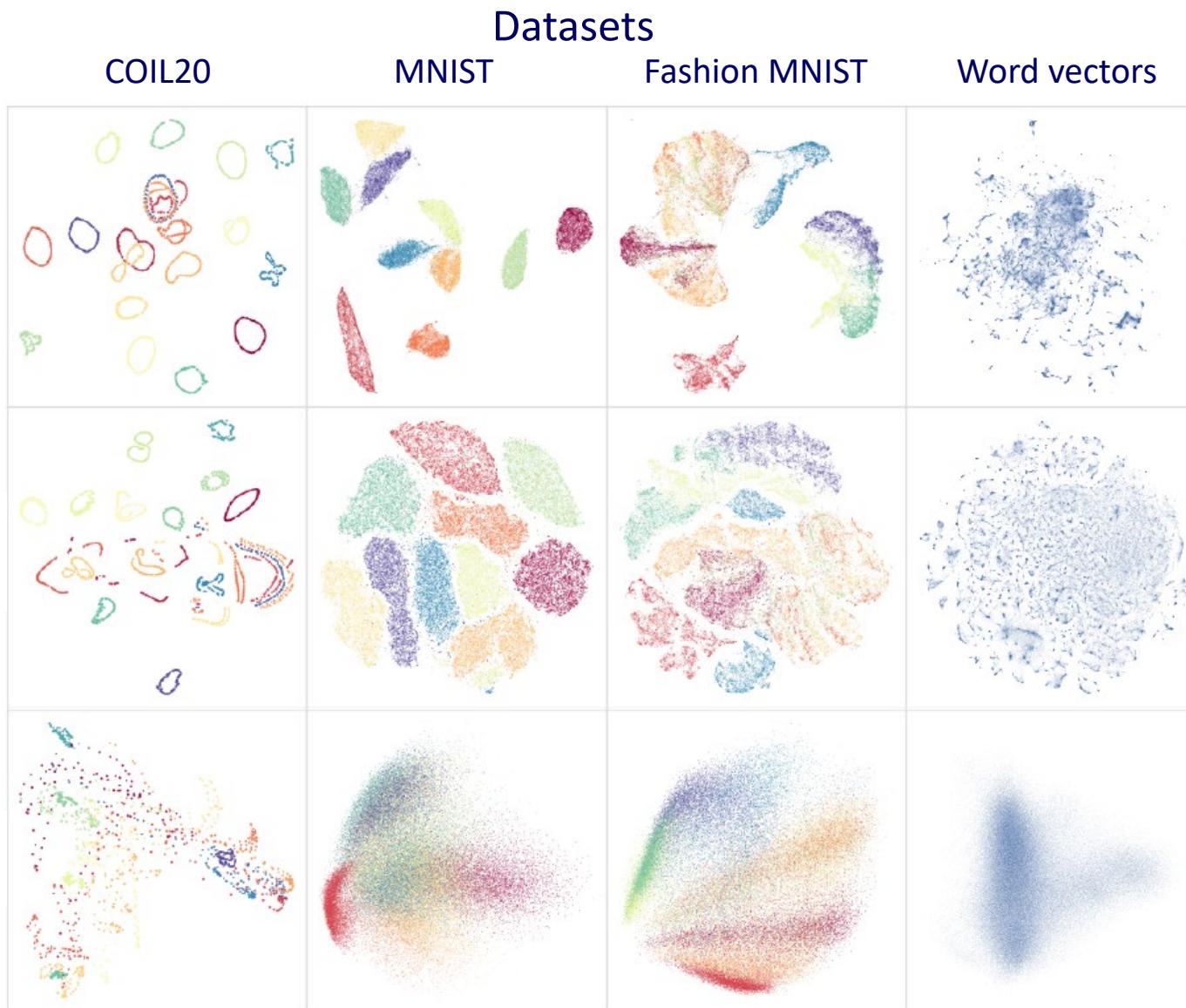


Figure 2: A comparison of several dimension reduction algorithms. UMAP reflects much of the large scale global structure, while also preserving the local fine structure similar to t-SNE.

# 01\_PBMC\_extended\_workflow\_example.rmd

## Run UMAP

```
130 ````{r run_UMAP}
131 # Time ~1 min
132 set.seed(overall_seed)
133 # Run UMAP on all scaled surface markers
134
135 # the line below will run UMAP on the data set (to see help page for UMAP, type
136 # "?UMAP -- enter" in console)
137
138 # you can view UMAP progress by opening up the console below
139 myumap <-
140 umap(transformed.chosen.markers, # input scaled data
141
142 n_neighbors = 15, # number of nearest neighbors to look at,
143 # scales with data set
144
145 n_threads = 1, # this makes UMAP reproducible
146 verbose = TRUE)
147 umap.data = as.data.frame(myumap)
148
149
150 ````{r plot_UMAP}
151 # Time <10 sec
152
153 # setting aspect ratio for plots
154 range <- apply(apply(umap.data, 2, range), 2, diff)
155 graphical.ratio.umap <- (range[1] / range[2])
156
157 # UMAP flat dot plot and density dot plot (1 dot = 1 cell)
158 UMAP.plot <- data.frame(x = umap.data[, 1], y = umap.data[, 2])
159
```

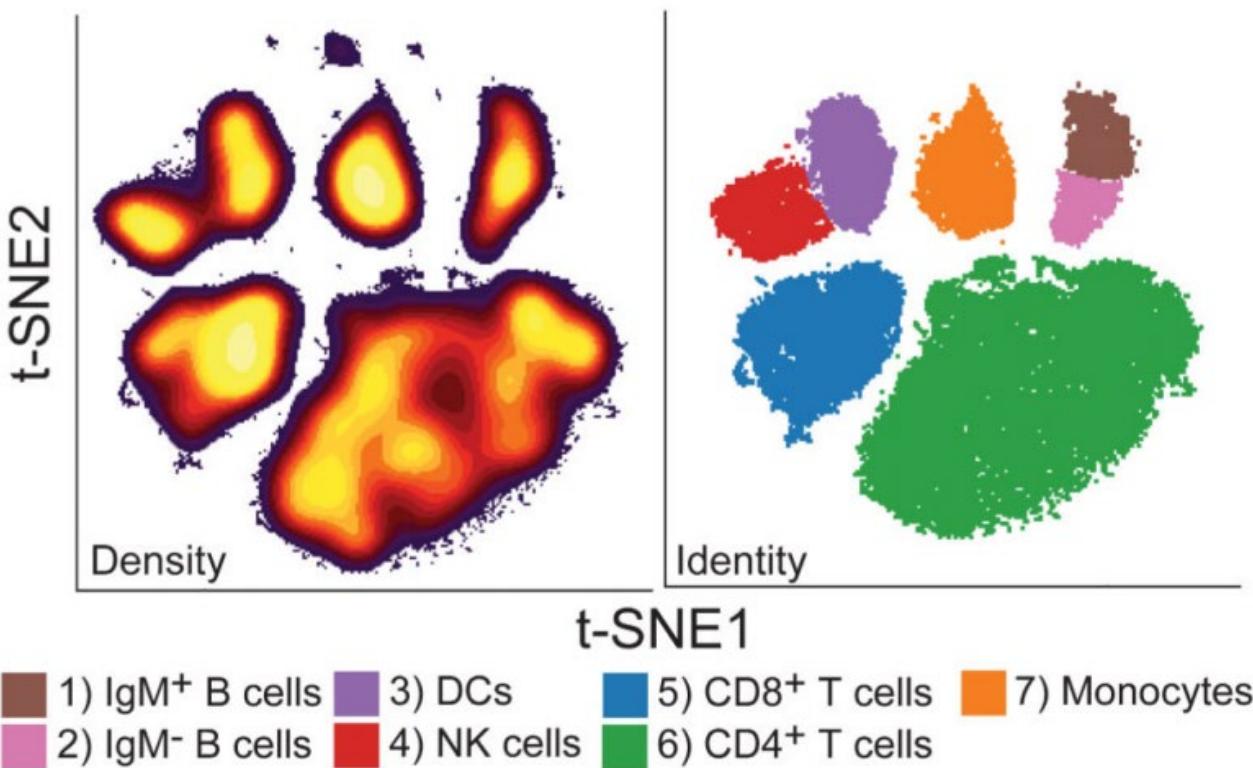
This section will run a UMAP analysis on the PBMC data using set parameters

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

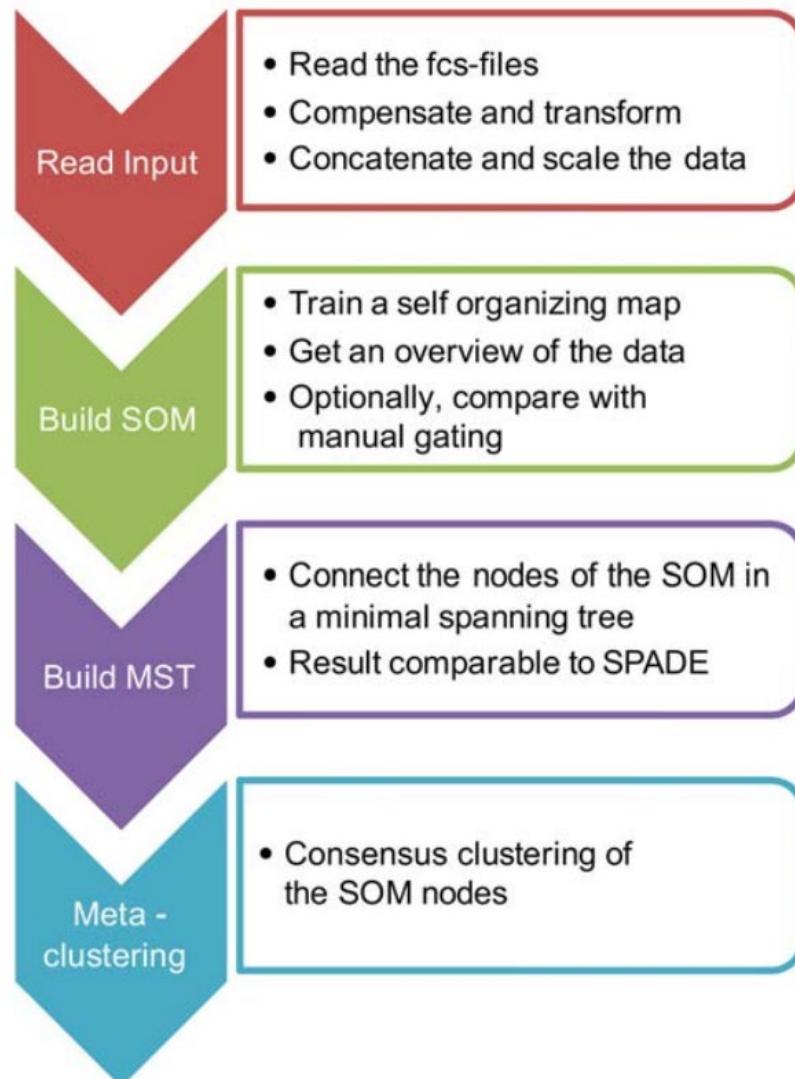
# Clusters can be Identified Based on Dimensionality Reduction Results

a

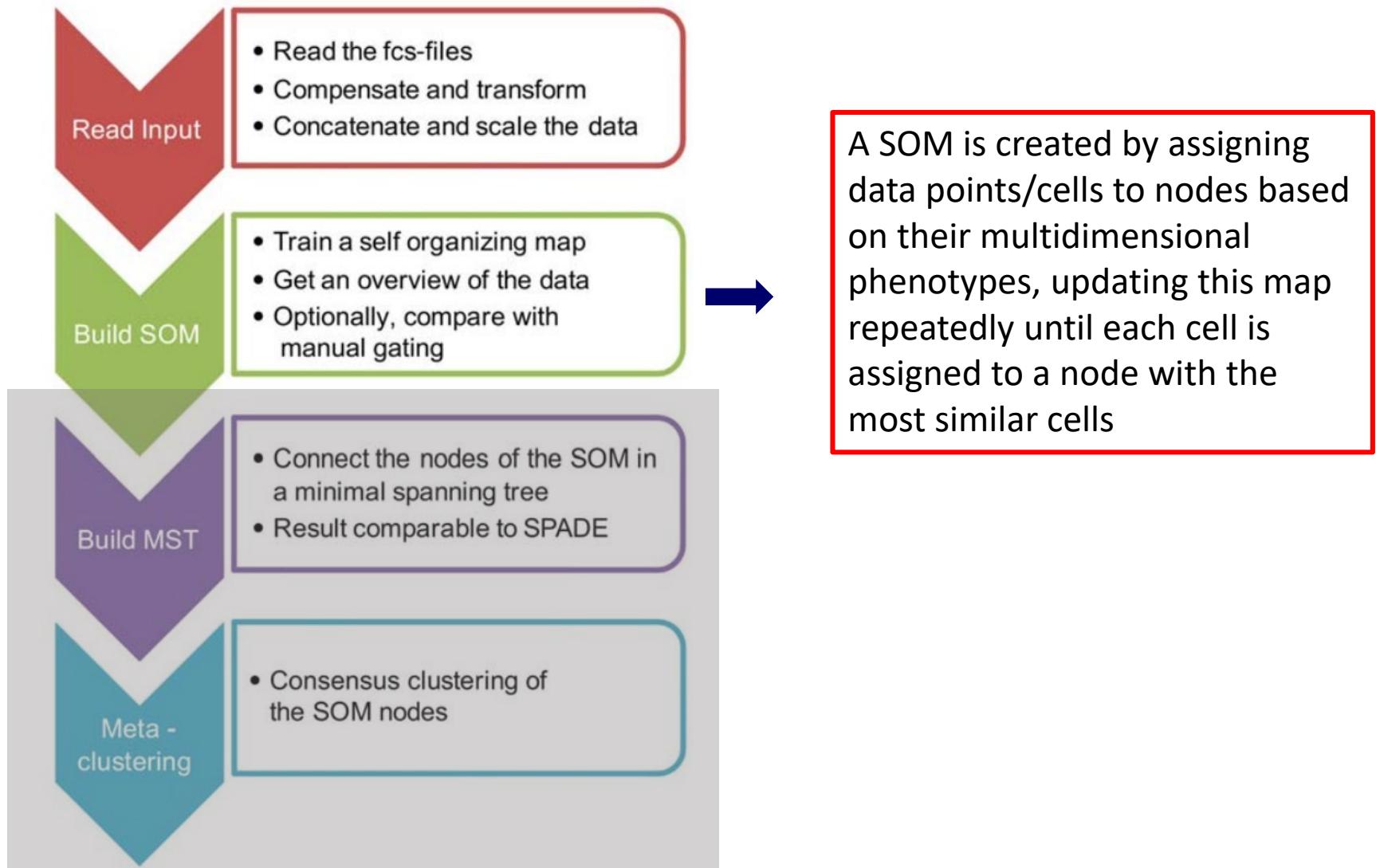
Identification of 7 canonical cell types  
in healthy human blood, 25D mass cytometry



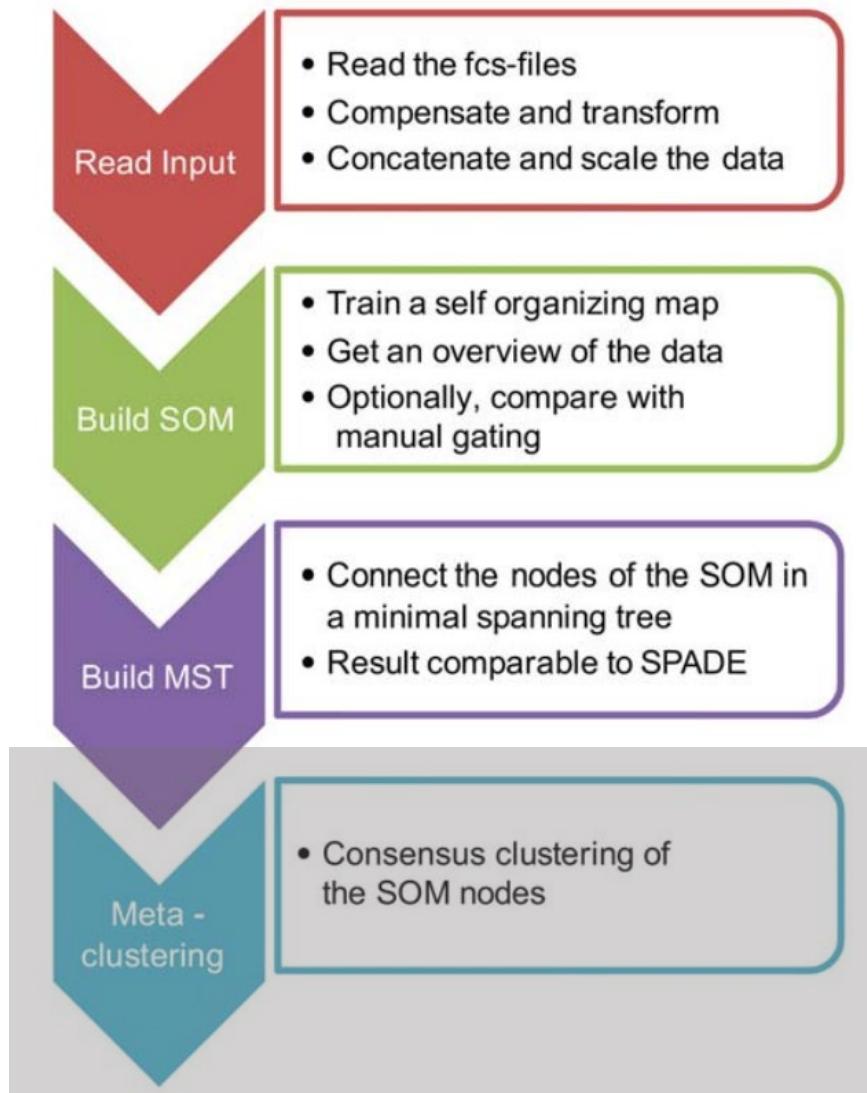
# Clustering with FlowSOM: Self-organizing Maps



# Clustering with FlowSOM: Self-organizing Maps

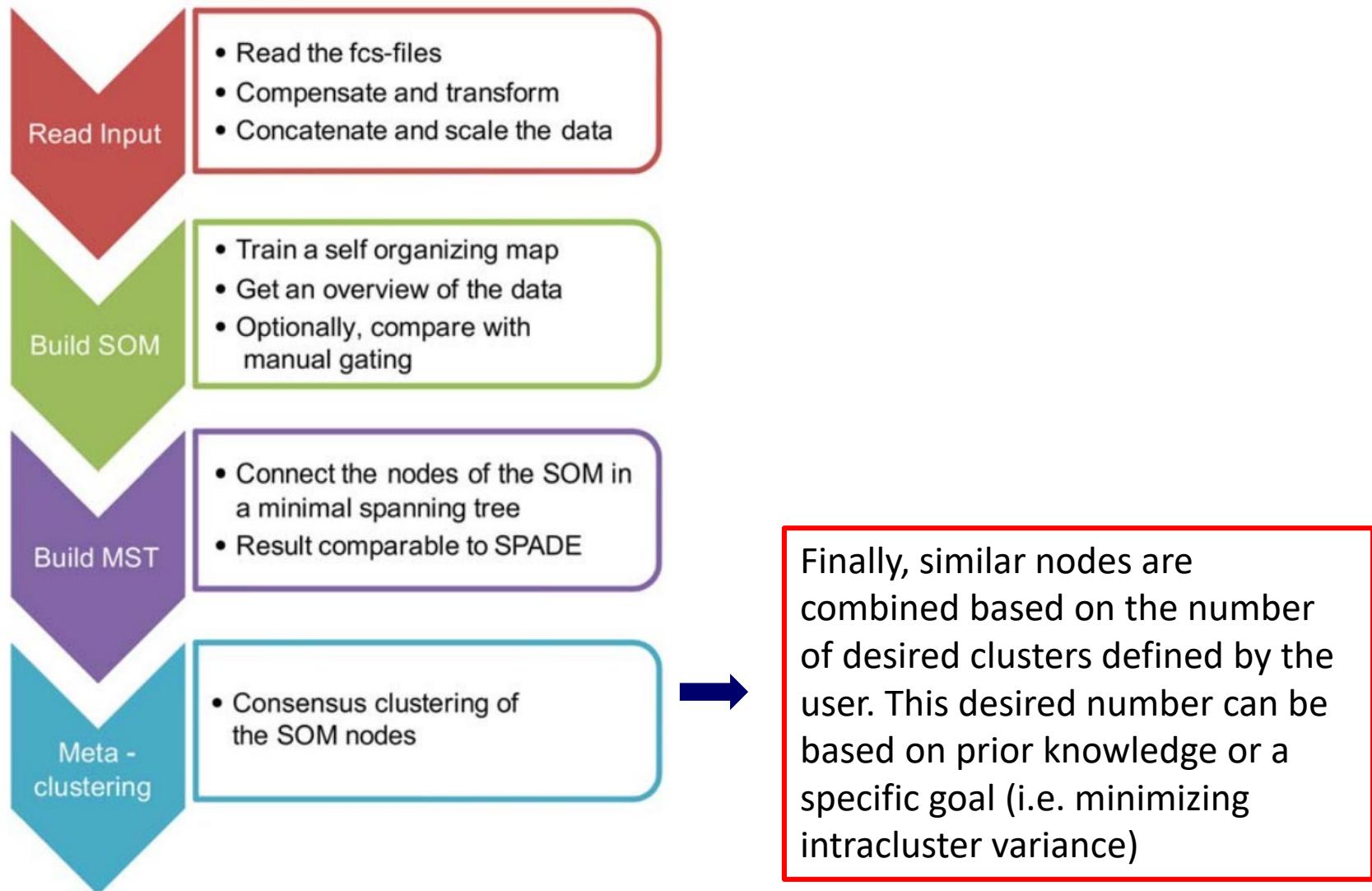


# Clustering with FlowSOM: Self-organizing Maps



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\*

# Clustering with FlowSOM: Self-organizing Maps



# 01\_PBMC\_extended\_workflow\_example.rmd

## Run FlowSOM

```
180 `r run_FlowSOM_on_t-SNE`
181 # Time <10 sec
182
183 # create flowFrame for FlowsOM input (using t-SNE axes as input)
184 matrix <- as.matrix(tsne.data)
185 metadata <-
186 data.frame(name = dimnames(matrix)[[2]],
187 desc = dimnames(matrix)[[2]])
188 metadata$range <- apply(apply(matrix, 2, range), 2, diff)
189 metadata$minRange <- apply(matrix, 2, min)
190 metadata$maxRange <- apply(matrix, 2, max)
191 flowframe <- new("flowFrame",
192 exprs = matrix,
193 parameters = AnnotatedDataFrame(metadata))
194
195 # implement the FlowsOM on the data by running the line below (to see help page
196 # for FlowsOM, type "?FlowsOM --> enter" in console)
197 fsom <-
198 FlowsOM(
199 flowframe, # input flowframe
200
201 colstouse = c(1:2), # columns to use
202
203 nclus = 10, # target number of clusters (this can
204
205 seed = overall_seed # set seed for reproducibility
206)
207 FlowsOM.clusters.tsne <-
208 as.matrix(fsom[[2]][fsom[[1]]mapmapping[, 1]])
209 ...
```

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

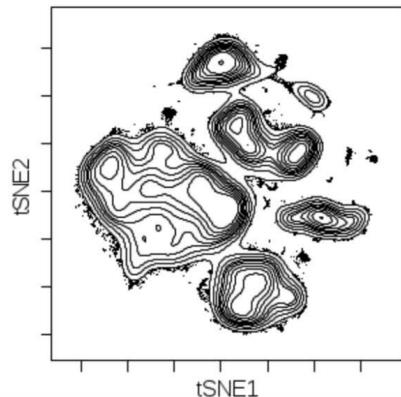
## Plot FlowSOM

```
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.ts
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
225 Repository: FR-FCM-ZY63") +
226 theme(panel.grid.major = element_blank(),
227 panel.grid.minor = element_blank())
228 ...
```

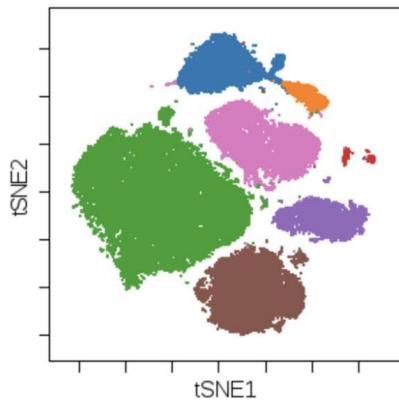
This section plots the identified clusters back onto the t-SNE axes and generates a plot (a colored version of the t-SNE plot from before)

# Clustering with FlowSOM: Self-organizing Maps

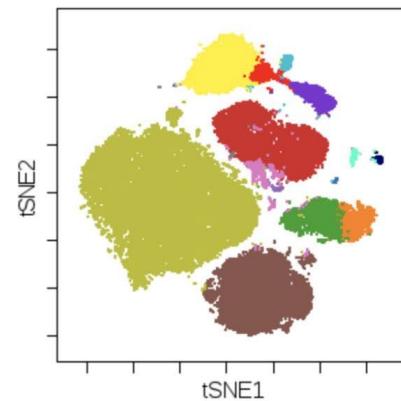
Contour plot of viSNE map



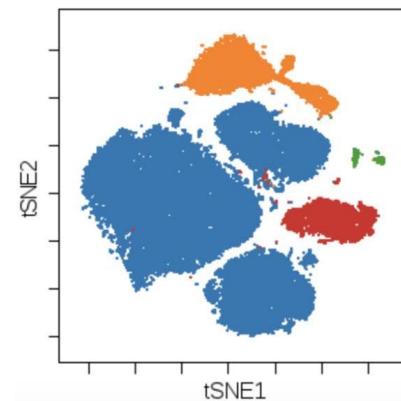
FlowSOM metaclusters overlaid on viSNE map



# metaclusters = 7



# metaclusters = 15



# metaclusters = 4

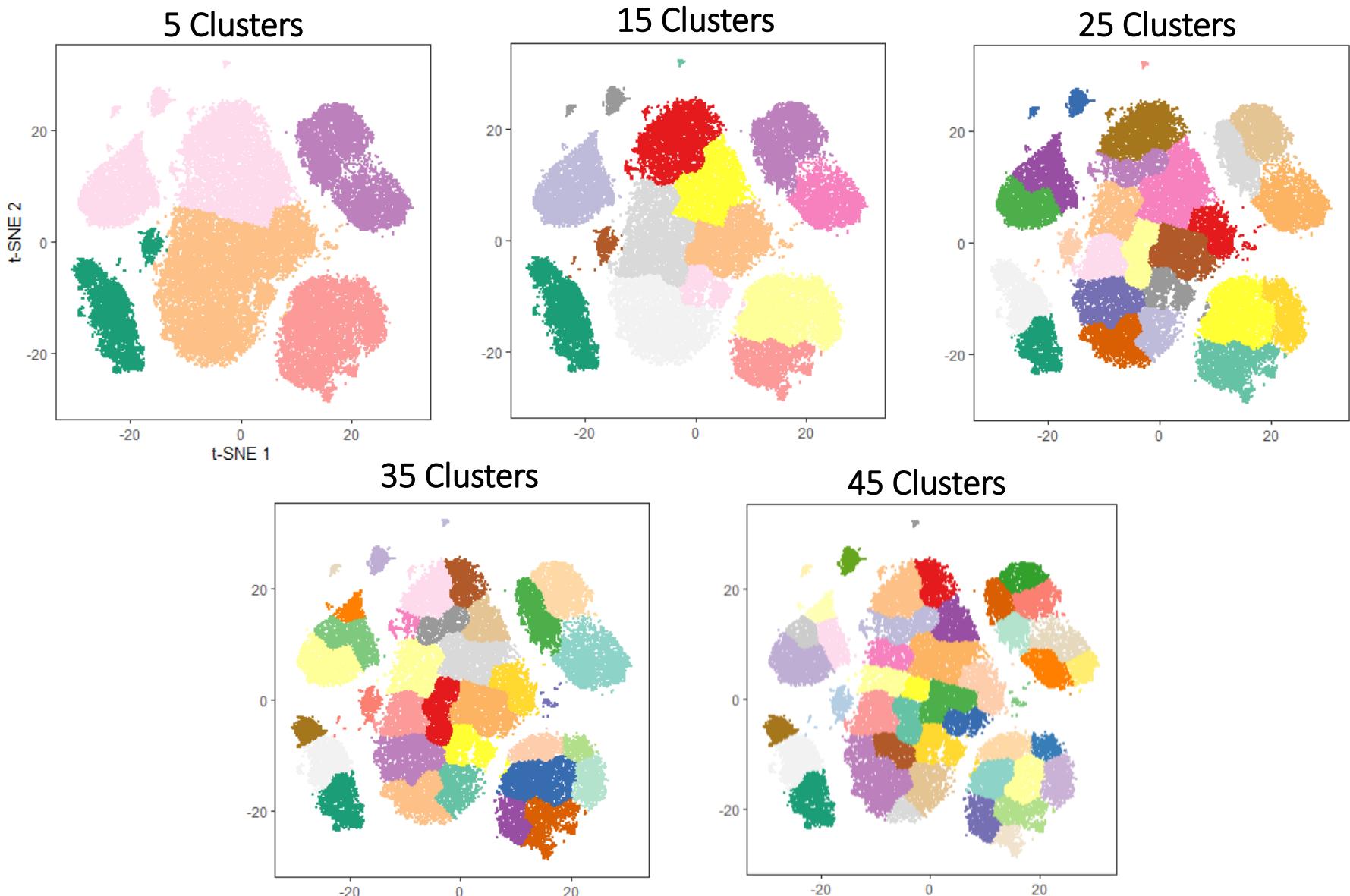
# 01\_PBMC\_extended\_workflow\_example.rmd

## Run FlowSOM

```
229 `r run_FlowSOM_varying_cluster_number`
230 # Time ~ 1-2 min
231 for (i in seq(5,45,by = 10)){
232 matrix <- as.matrix(tsne.data)
233 metadata <-
234 data.frame(name = dimnames(matrix)[[2]],
235 desc = dimnames(matrix)[[2]])
236 metadata$range <- apply(apply(matrix, 2, range), 2, diff)
237 metadata$minRange <- apply(matrix, 2, min)
238 metadata$maxRange <- apply(matrix, 2, max)
239 flowframe <- new("flowFrame",
240 exprs = matrix,
241 parameters = AnnotatedDataFrame(metadata))
242 fsom <-
243 FlowsOM(
244 flowframe,
245 colstoUse = c(1:2),
246 nclus = i,
247 seed = overall_seed
248)
249 FlowsOM.clusters.vary <-
250 as.matrix(fsom[[2]][fsom[[1]]mapmapping[, 1]])
251
252 legend.col = round(max(as.numeric(as.vector(FlowsOM.clusters.vary)))/3)
253 print(ggplot(tsne.plot) + coord_fixed(ratio=graphical.ratio.tsne) +
254 geom_point(aes(x=x, y=y, color=FlowsOM.clusters.vary), cex = 0.5) +
255 labs(x = "t-SNE 1", y = "t-SNE 2", title = "FlowSOM Clustering on t-SNE Axes",
256 color = "cluster") + theme_bw() +
257 guides(colour = guide_legend(override.aes = list(size=4),
258 nrow = legend.col)) +
259 scale_color_manual(values = sample(col_vector)) +
260 labs(caption = "Data from Diggins et al., Nat Methods 2017, 14: 275-278 \nFlow
261 Repository: FR-FCM-ZY63") +
262 theme(panel.grid.major = element_blank(),
263 panel.grid.minor = element_blank()))
264 ...
```

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



# 01\_PBMC\_extended\_workflow\_example.rmd

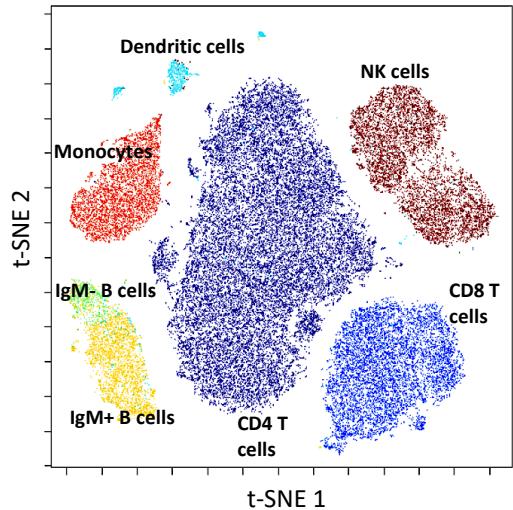
## Run FlowSOM

```
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]]mapmapping[, 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
297 Repository: FR-FCM-ZY63") +
298 theme(panel.grid.major = element_blank(),
299 panel.grid.minor = element_blank())
299 ...
```

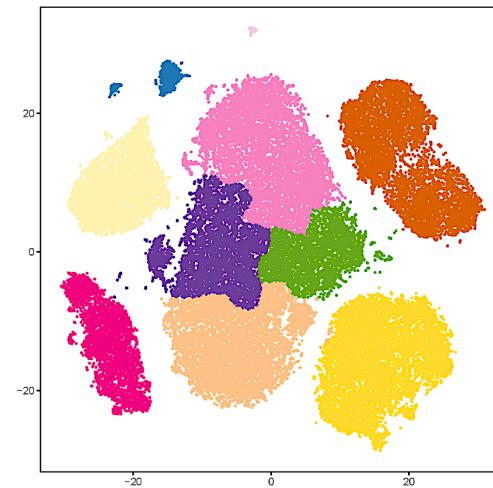
This section performs FlowSOM clustering on the protein markers in the cyTOF experiment

# FlowSOM Clusters are Dependent on Input Parameters

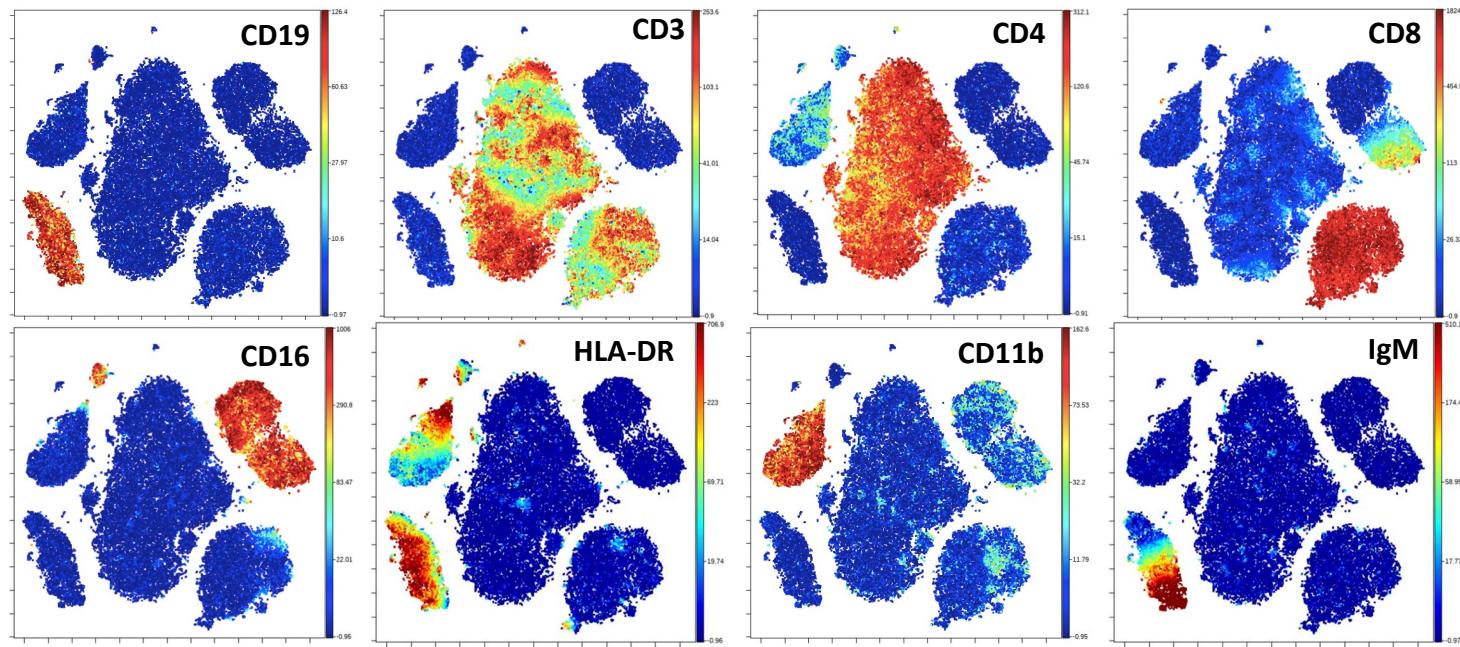
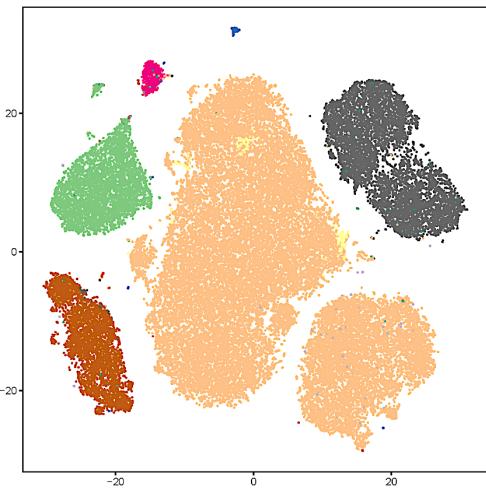
## Major Populations Overlaid on t-SNE Axes



## FlowSOM on t-SNE Axes (n = 10)



## FlowSOM on Original Markers (n = 10)



# 01\_PBMC\_extended\_workflow\_example.rmd

## Run FlowSOM

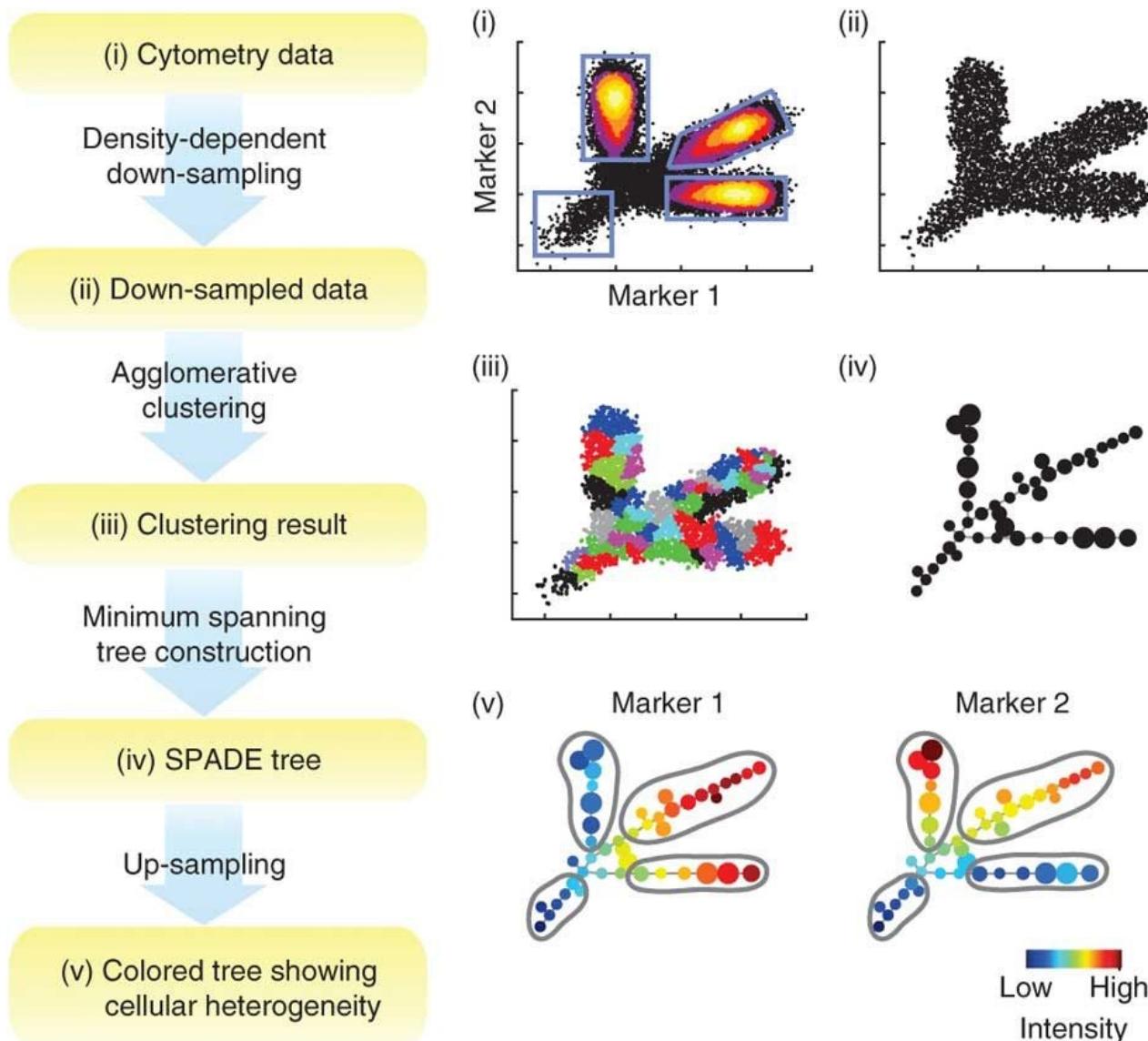
```
301 `r run_FlowSOM_on_UMAP`
302 # Time <10 sec
303
304 matrix <- as.matrix(umap.data)
305 metadata <-
306 data.frame(name = dimnames(matrix)[[2]],
307 desc = dimnames(matrix)[[2]])
308 metadata$range <- apply(apply(matrix, 2, range), 2, diff)
309 metadata$minRange <- apply(matrix, 2, min)
310 metadata$maxRange <- apply(matrix, 2, max)
311 flowframe <- new("flowFrame",
312 exprs = matrix,
313 parameters = AnnotatedDataFrame(metadata))
314 fsom <-
315 FlowsOM(
316 flowframe,
317 colsToUse = c(1:2),
318 nclus = 10,
319 seed = overall_seed
320)
321 FlowsOM.clusters.umap <-
322 as.matrix(fsom[[2]][fsom[[1]]mapmapping[, 1]])
323
324 ggplot(UMAP.plot) + coord_fixed(ratio=graphical.ratio.umap) +
325 geom_point(aes(x=x, y=y, color=FlowsOM.clusters.umap), cex = 0.5) +
326 labs(x = "UMAP 1", y = "UMAP 2", title = "FlowSOM Clustering on UMAP Axes",
327 color = "cluster") + theme_bw() +
328 guides(colour = guide_legend(override.aes = list(size=4)))+
329 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(),
332 panel.grid.minor = element_blank())
333 ...
```

This section performs FlowSOM clustering on the UMAP results

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



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

Build Graphs

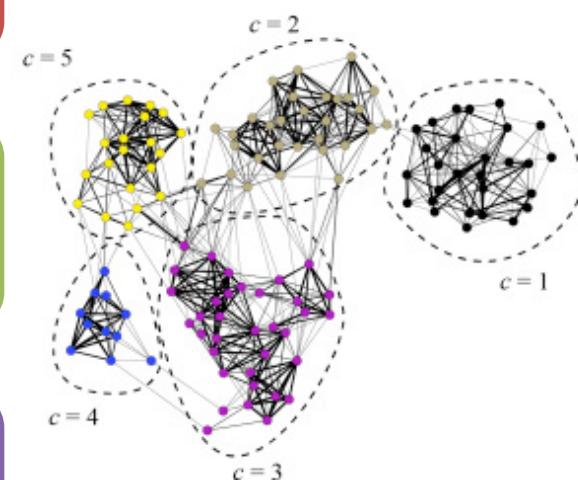
- Build a single-cell graph for each sample or dataset
  - Each cell is a node and is connected by edges to nearest neighbors

Cluster

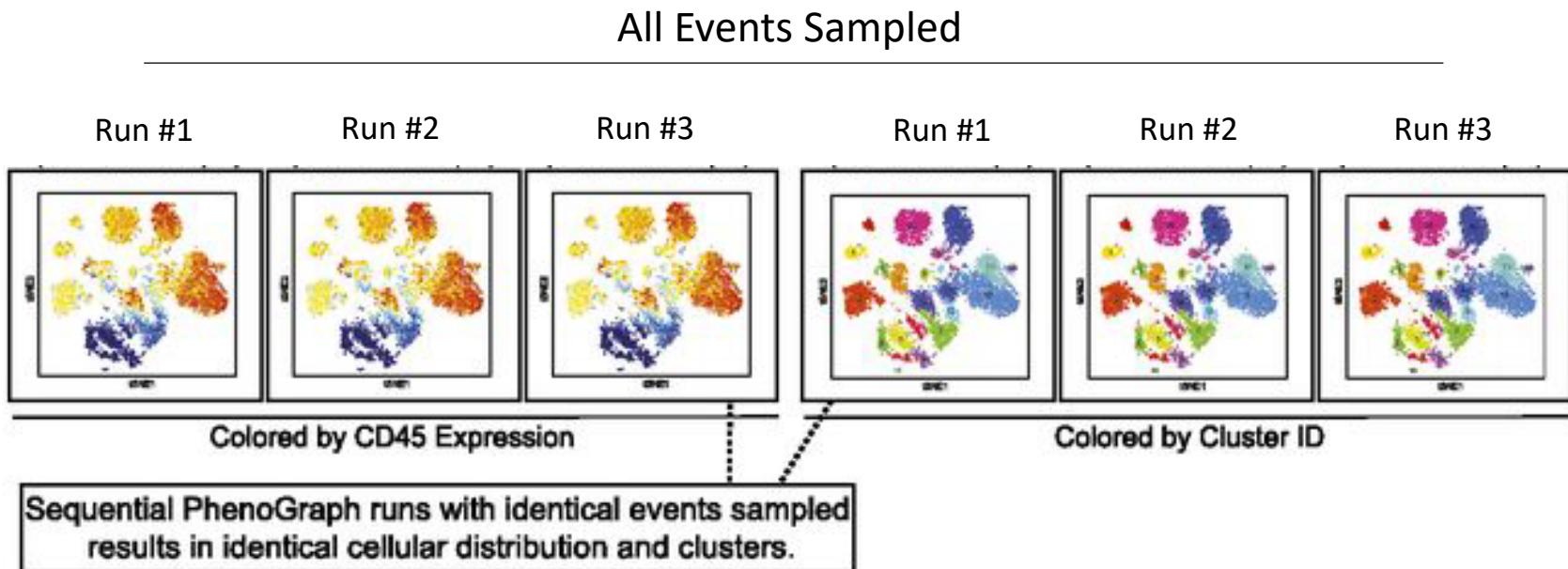
- Partition each graph into distinct subpopulations
  - Based on density/interconnected nodes

Define

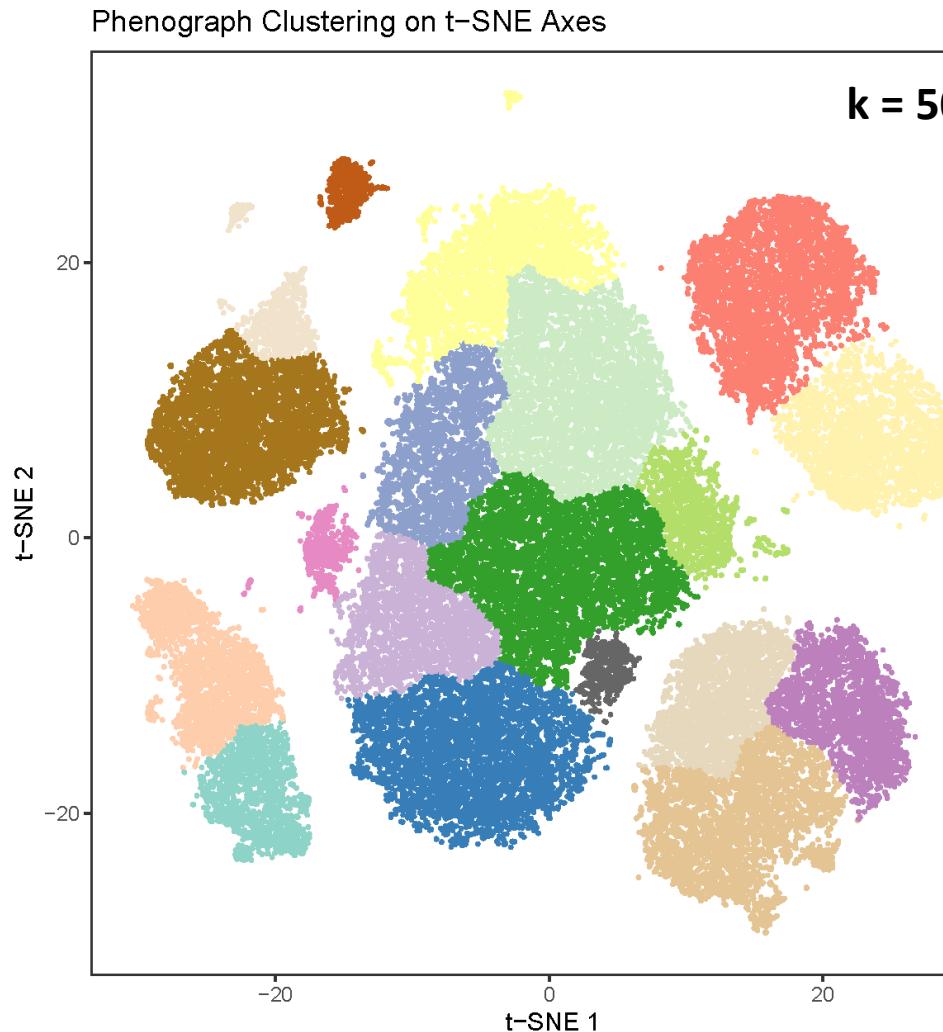
- Extract surface and signaling features for each subpopulation



# Phenograph is Deterministic

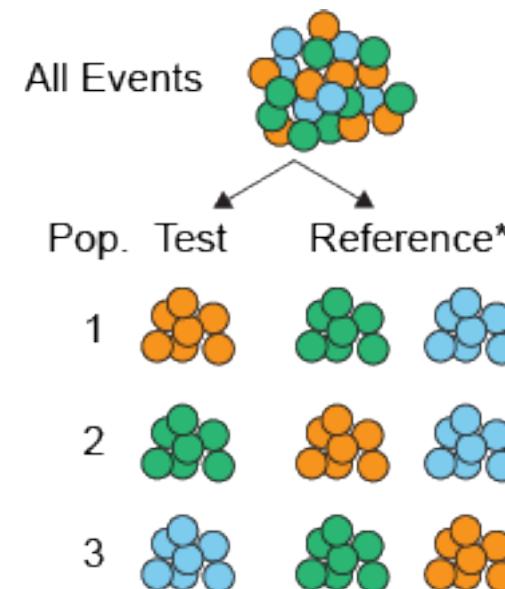
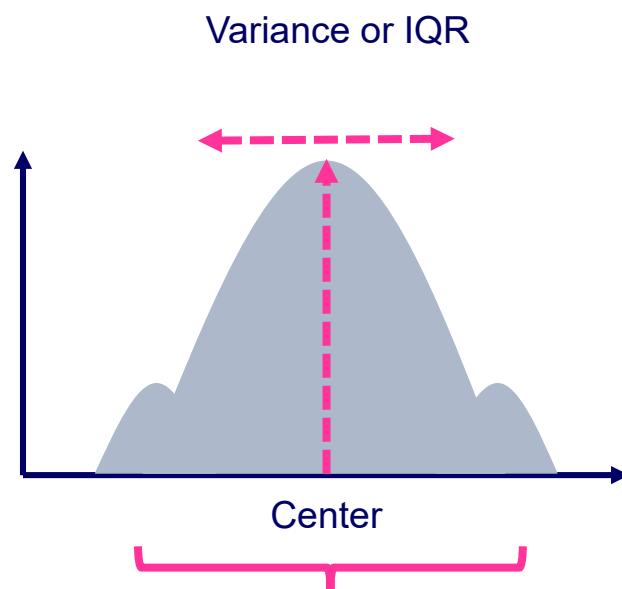


# K Nearest Neighbor Settings Determine the Number of Clusters Identified by Phenograph

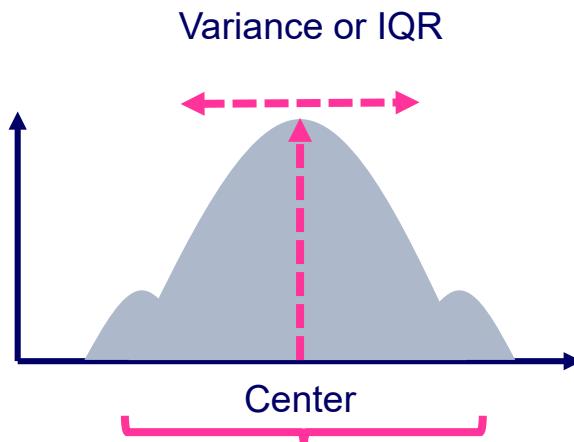


# Marker Enrichment Modeling Analysis Identifies Markers that are Specifically Expressed or Lacking on Populations

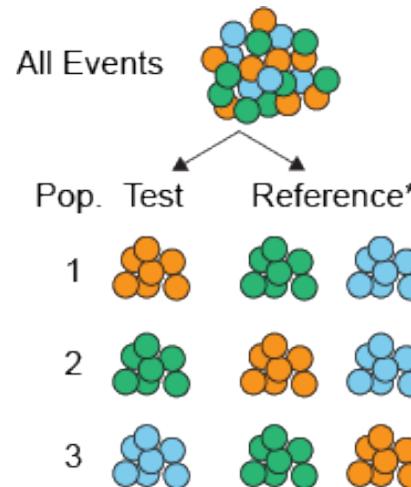
MEM accounts for variance and median of markers to identify enriched features on subsets of cells



# MEM Quantifies Relative Enrichment by Combining Magnitude and Interquartile Range



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



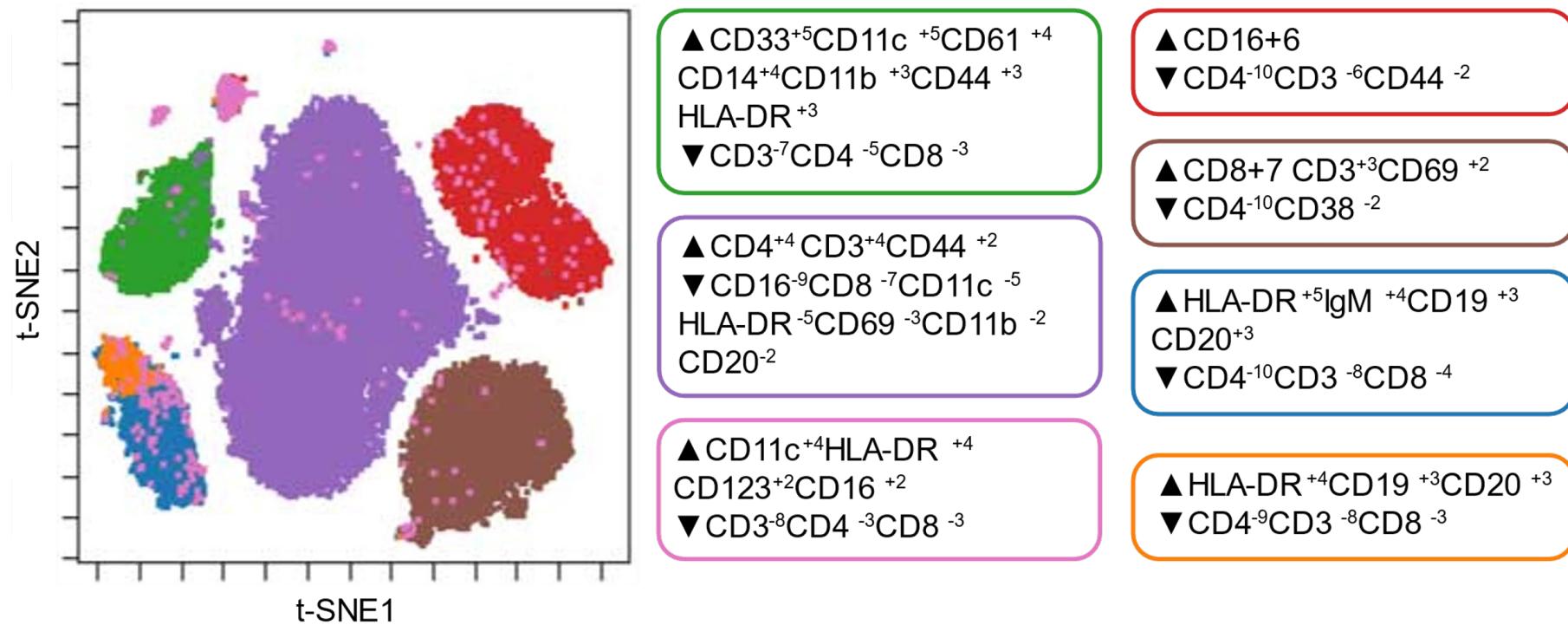
MEM label

▲ 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>

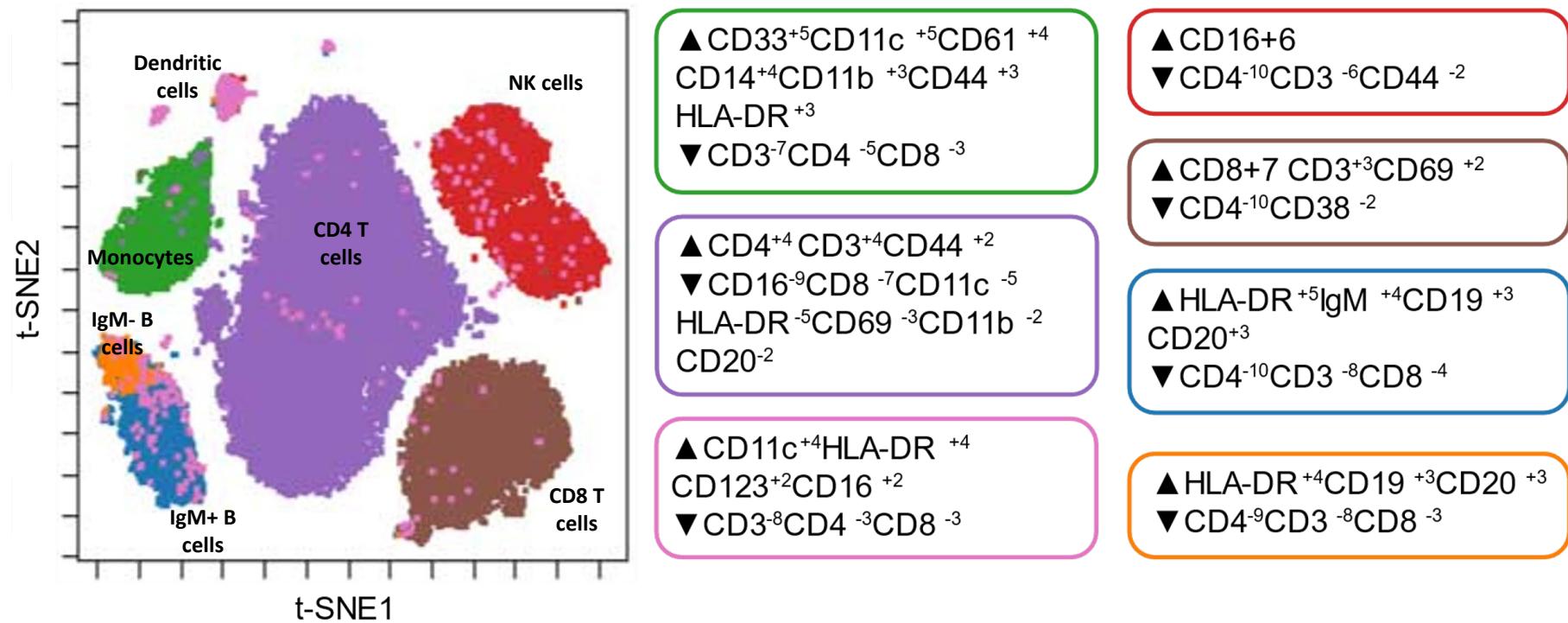
Linear transformation to -10 to +10

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

# MEM Quantifies Relative Enrichment by Combining Magnitude and Interquartile Range



# MEM Quantifies Relative Enrichment by Combining Magnitude and Interquartile Range



# 01\_PBMC\_extended\_workflow\_example.rmd

## Run MEM

```
335 ````{r run_MEM_on_FlowSOM_on_t-SNE}
336 # Time ~30 sec
337
338 # Run MEM on the FlowSOM clusters found from using t-SNE axes
339 cluster = as.numeric(as.vector((FlowSOM.clusters.tsne)))
340 MEM.data = cbind(transformed.chosen.markers, cluster)
341
342 MEM.values.tf = MEM(
343 MEM.data, # input data (last column must contain cluster values)
344
345 transform = FALSE, # data is already scaled in this case
346 cofactor = 1,
347 choose.markers = FALSE,
348 markers = "all", # use all transformed, chosen markers from pre-
349 # selection
350
351 choose.ref = FALSE, # reference will be all other cells
352 zero.ref = FALSE,
353 rename.markers = FALSE,
354 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
355 file.is.clust = FALSE,
356 add.fileID = FALSE,
357 IQR.thresh = NULL
358)
359
360 # build MEM heatmap and output enrichment scores
361 build.heatmaps(
362 MEM.values.tf, # input MEM values
363
364 cluster.MEM = "both", # dendrogram for columns and rows
365
366 display.thresh = 2, # display threshold for MEM scores
367 newwindow.heatmaps = FALSE,
368 output.files = TRUE, # makes txt and PDF files for heatmap and MEM
369 # scores
370
371 labels = TRUE, # include labels in heatmap
372 only.MEMheatmap = FALSE
373)
````
```

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

```
376 ````{r run_MEM_on_FlowsOM_on_OG}
377 # Time ~30 sec
378
379 cluster = as.numeric(as.vector(FlowsOM.clusters.og)))
380 MEM.data = cbind(transformed.chosen.markers, cluster)
381
382 MEM.values.ogf = MEM(
383   MEM.data,
384   transform = FALSE,
385   cofactor = 1,
386   choose.markers = FALSE,
387   markers = "all",
388   choose.ref = FALSE,
389   zero.ref = FALSE,
390   rename.markers = FALSE,
391   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
392   labels
393   file.is.clust = FALSE,
394   add.fileID = FALSE,
395   IQR.thresh = NULL
396 )
397
398 build.heatmaps(
399   MEM.values.ogf,
400   cluster.MEM = "both",
401   display.thresh = 2,
402   newwindow.heatmaps = FALSE,
403   output.files = TRUE,
404   labels = TRUE,
405   only.MEMheatmap = FALSE
406 )
407
408 ````{r run_MEM_on_FlowsOM_on_UMAP}
409 # Time ~30 sec
410
411 cluster = as.numeric(as.vector(FlowsOM.clusters.umap)))
412 MEM.data = cbind(transformed.chosen.markers, cluster)
413
```

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

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

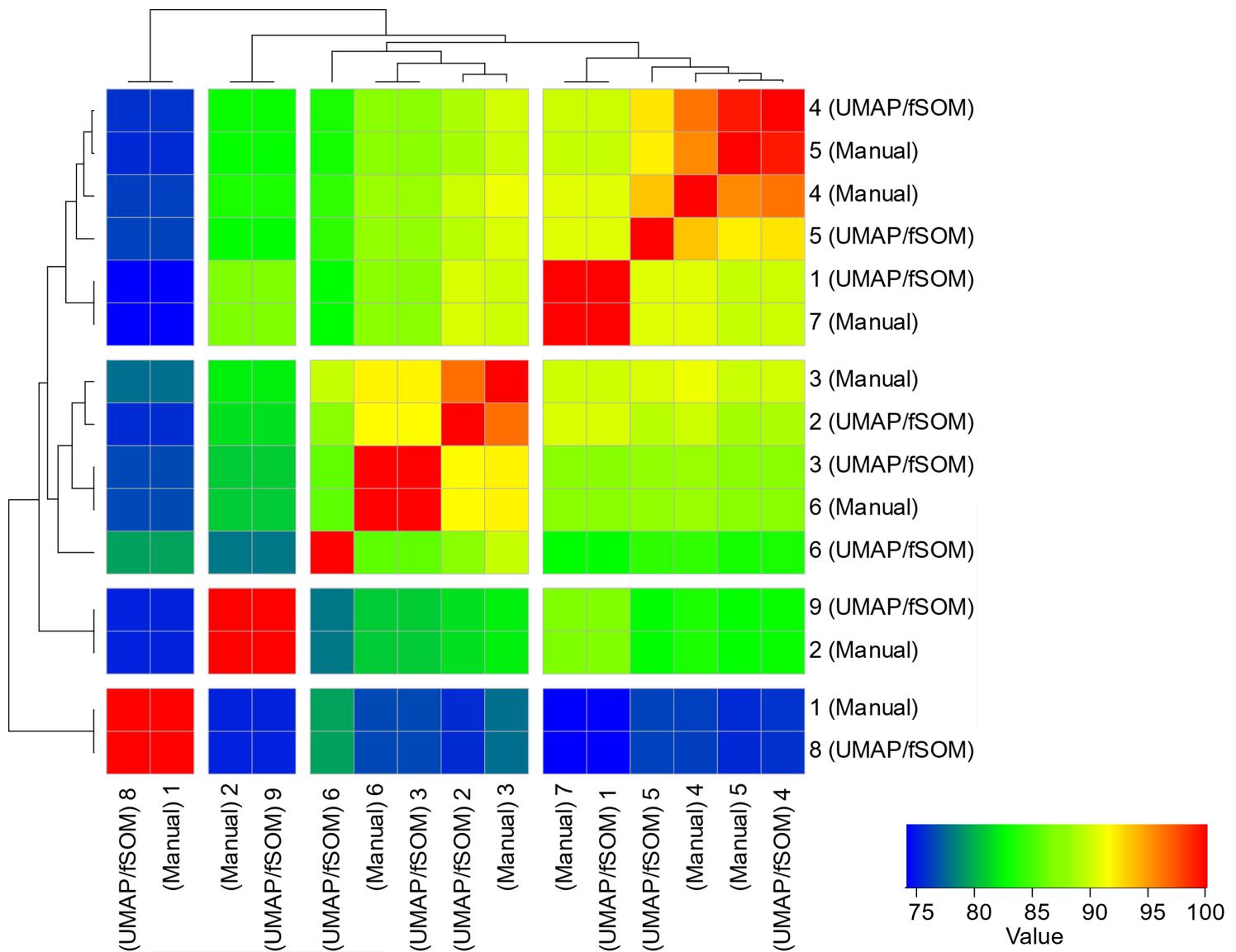
01_PBMC_extended_workflow_example.rmd

Run MEM

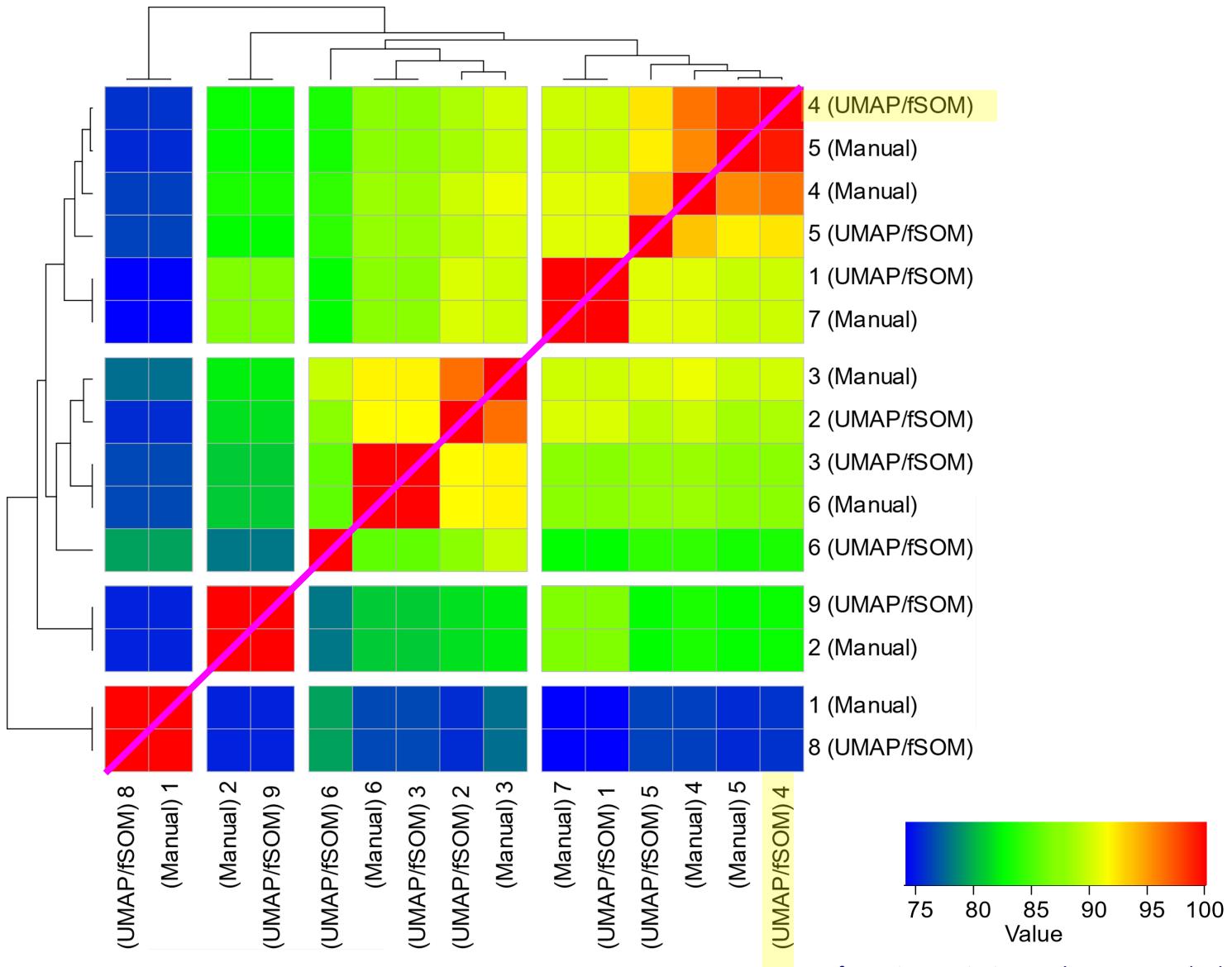
```
440 ````{r run_MEM_on_manually_gated_pops}
441 # Time ~30 sec
442
443 MEM.values.orig = MEM(
444   combined.data,
445   transform = TRUE,
446   cofactor = 15,
447   choose.markers = FALSE,
448   markers = "12:20,22:23,25:33,35:36,38:40",
449   choose.ref = FALSE,
450   zero.ref = FALSE,
451   rename.markers = FALSE,
452   new.marker.names = "CD19,CD117,CD11b,CD4,CD8,CD20,CD34,CD61,CD123,CD45RA,CD45,CD10,CD3
453 3,CD11c,CD14,CD69,CD15,CD16,CD44,CD38,CD25,CD3,IgM,HLA-DR,CD56",
454   file.is.clust = FALSE,
455   add.fileID = FALSE,
456   IQR.thresh = NULL
457 )
458 build.heatmaps(
459   MEM.values.orig,
460   cluster.MEM = "both",
461   display.thresh = 2,
462   newWindow.heatmaps = FALSE,
463   output.files = TRUE,
464   labels = TRUE,
465   only.MEMheatmap = FALSE
466 )```
467
```

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

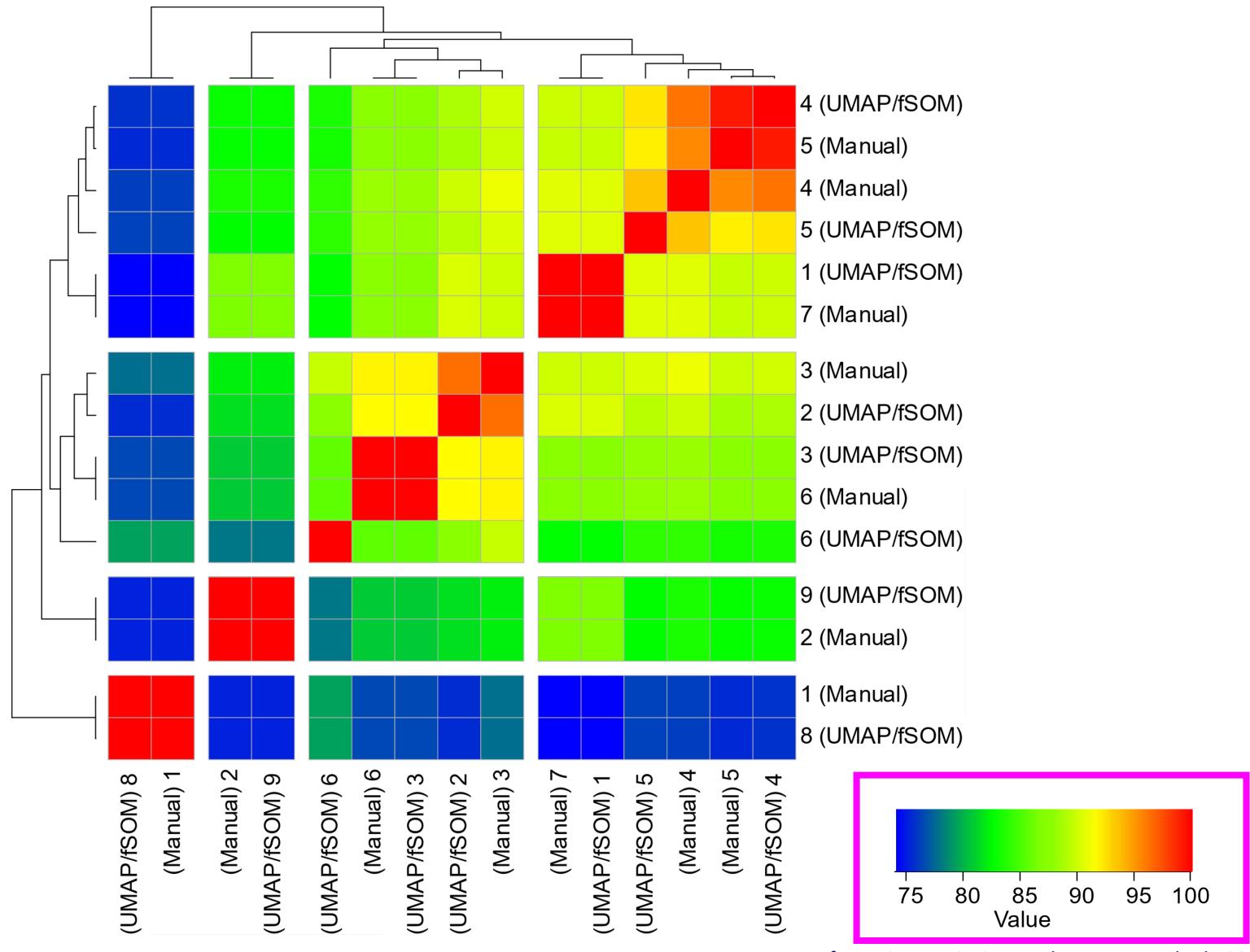


RMSD Heatmap Output is Reflected Over the Diagonal Axis



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

Heat Indicates Similarity in MEM Scores



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

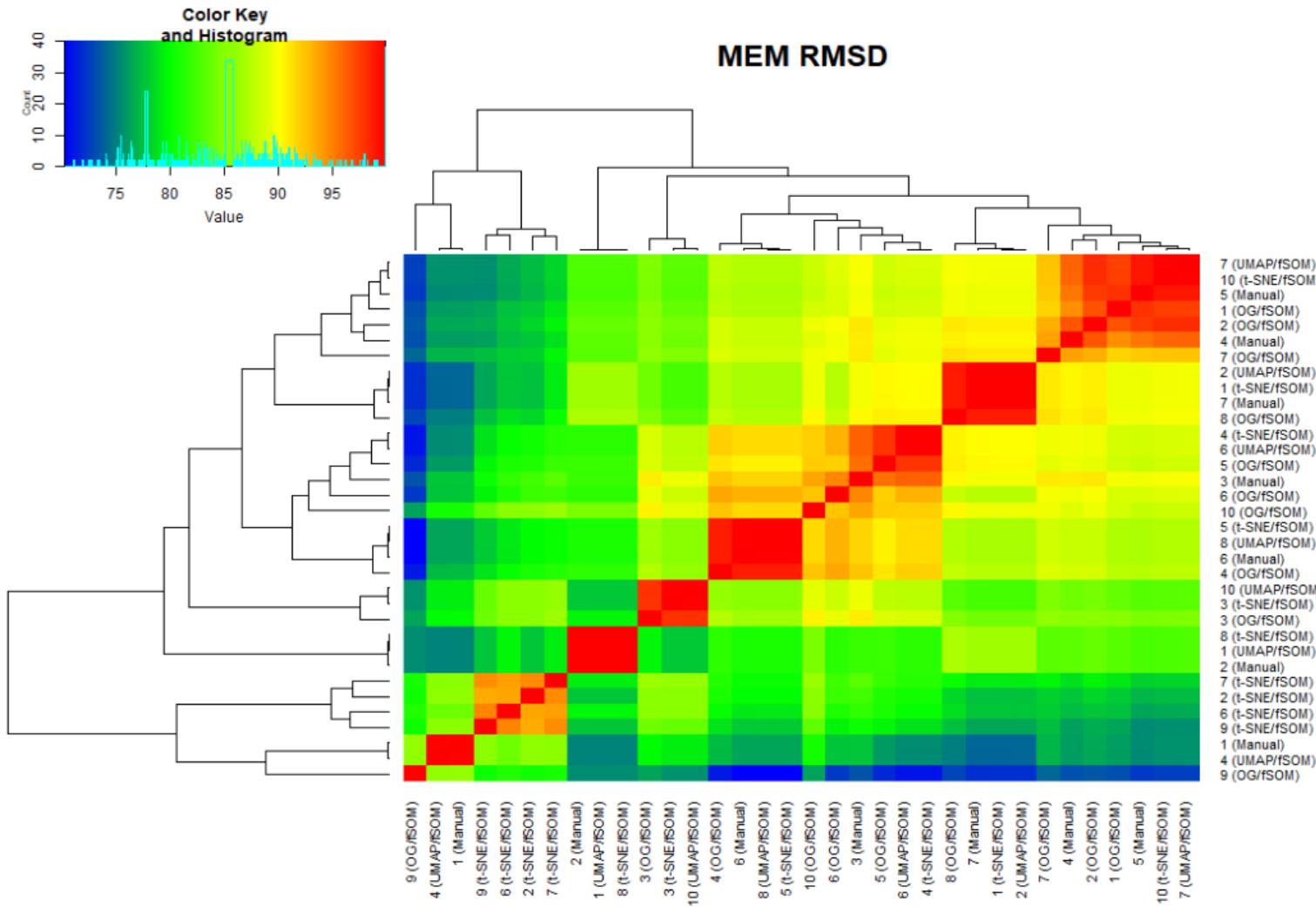
01_PBMC_extended_workflow_example.rmd

Run RMSD

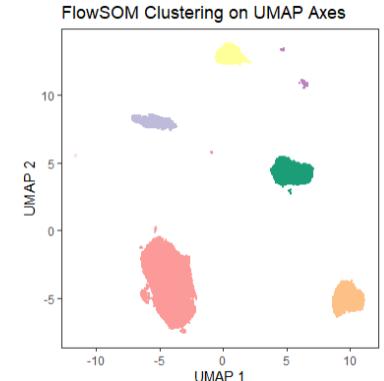
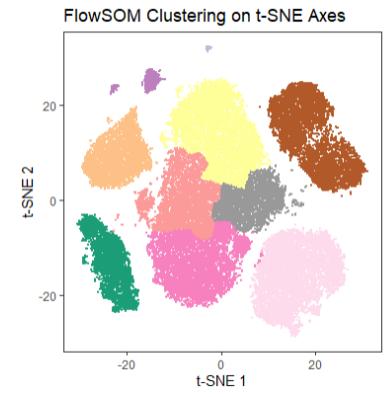
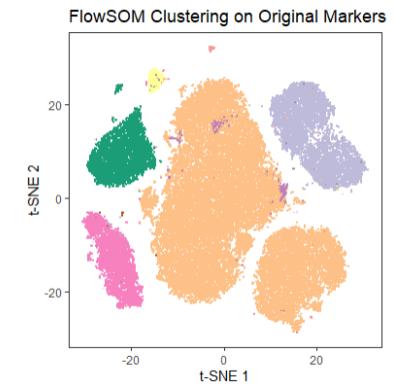
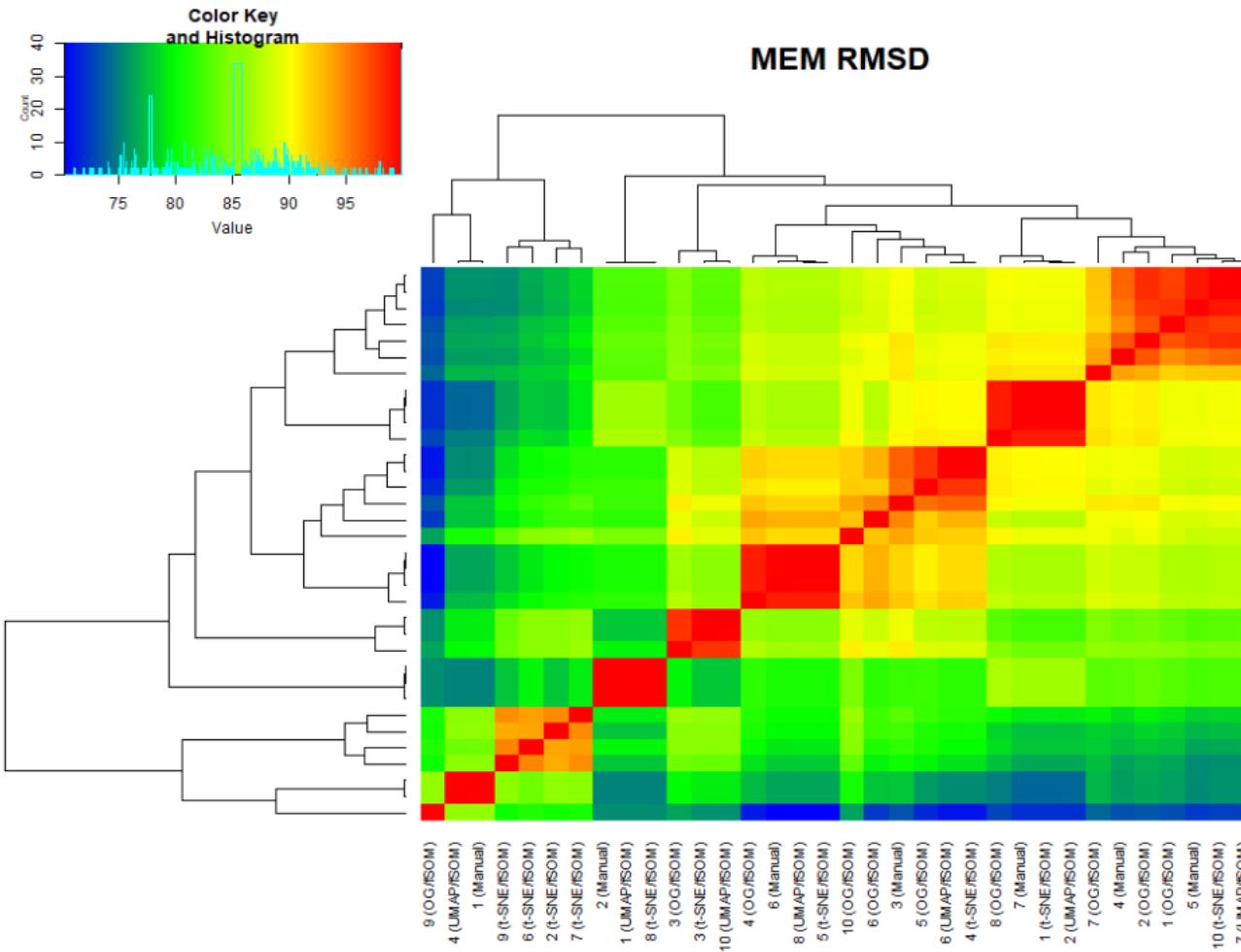
```
469 `r RMSD for All clusters}
470 # RMSD to compare labels from all populations (FlowSOM clusters vs. manually
471 # gated populations)
472
473 orig.MEM.scores = as.data.frame(MEM.values.orig[[5]])
474 rownames(orig.MEM.scores) = paste0(rownames(orig.MEM.scores), " (o")
475
476 ogf.MEM.scores = as.data.frame(MEM.values.ogf[[5]])
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,
486 tf.MEM.scores))
487
488 RMSD_vals <-
489   MEM_RMSD(
490     all.MEM.values,                      # input all MEM values from clustering and
491                                         # expert gating
492     format = NULL,
493     newwindow.heatmaps = FALSE,
494     output.matrix = TRUE
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)

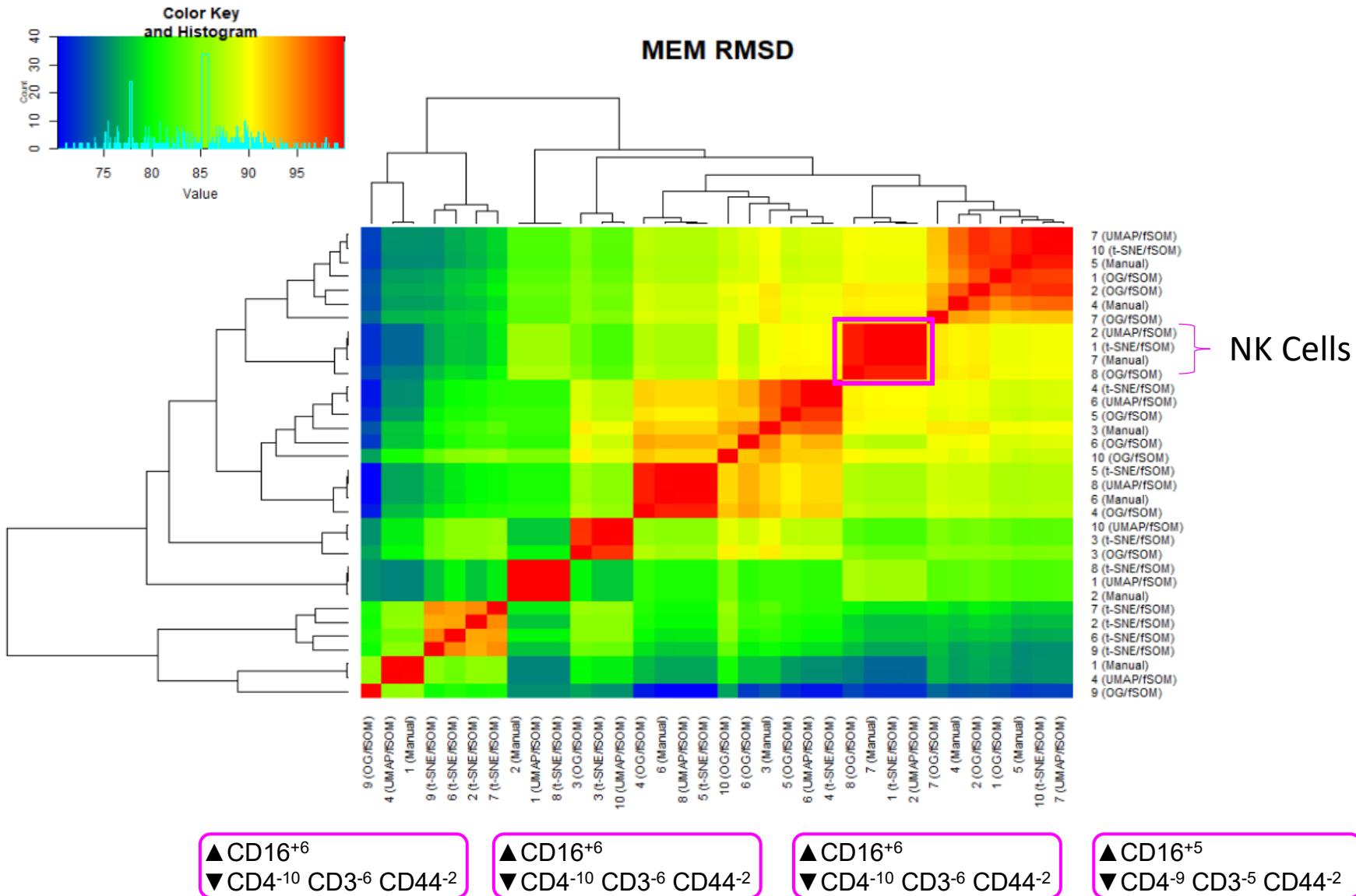
R Output for RMSD Demonstrates that Clusters are More Similar to Each other than Methods of Deriving Them



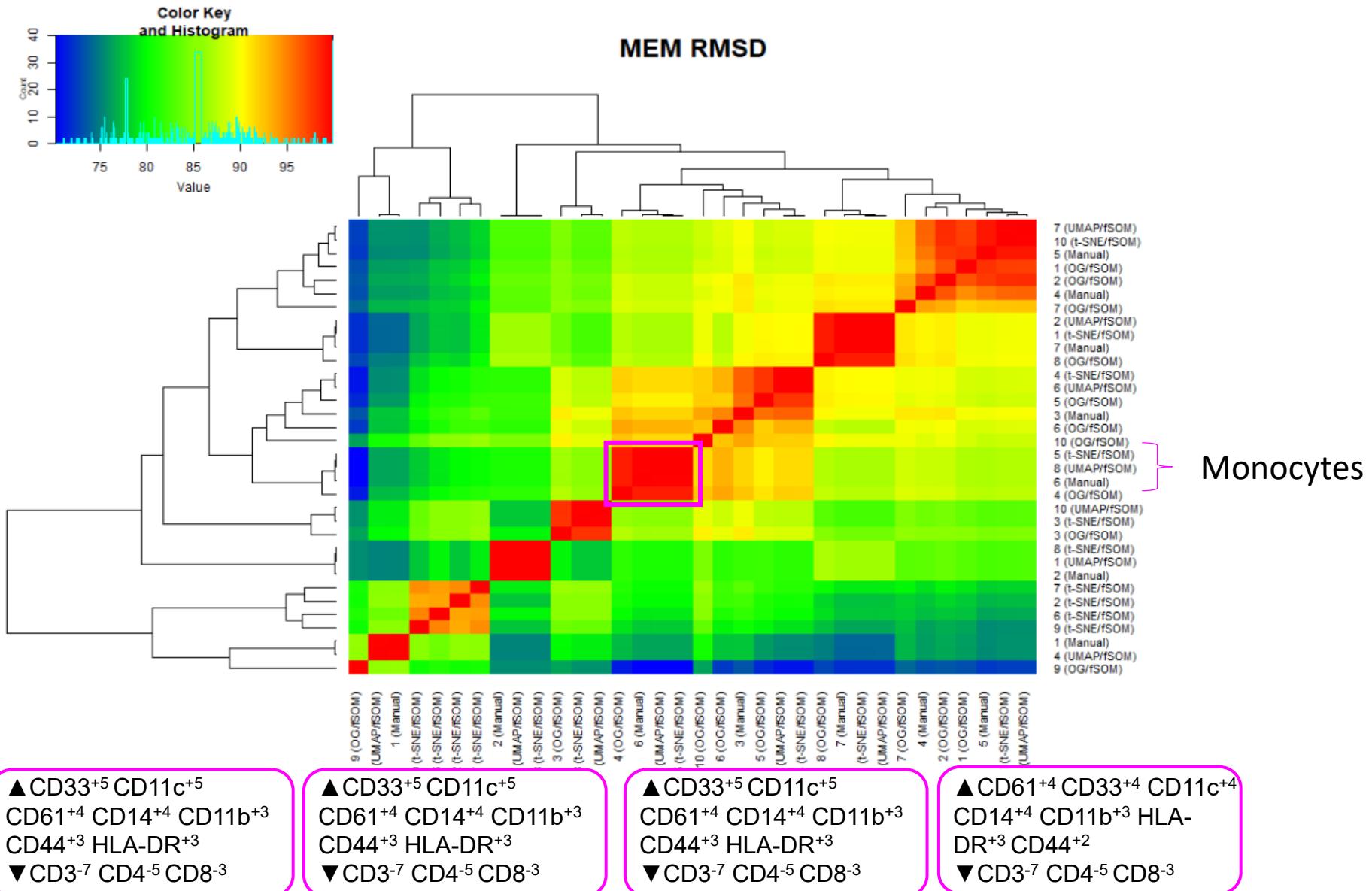
R Output for RMSD Demonstrates that Clusters are More Similar to Each other than Methods of Deriving Them



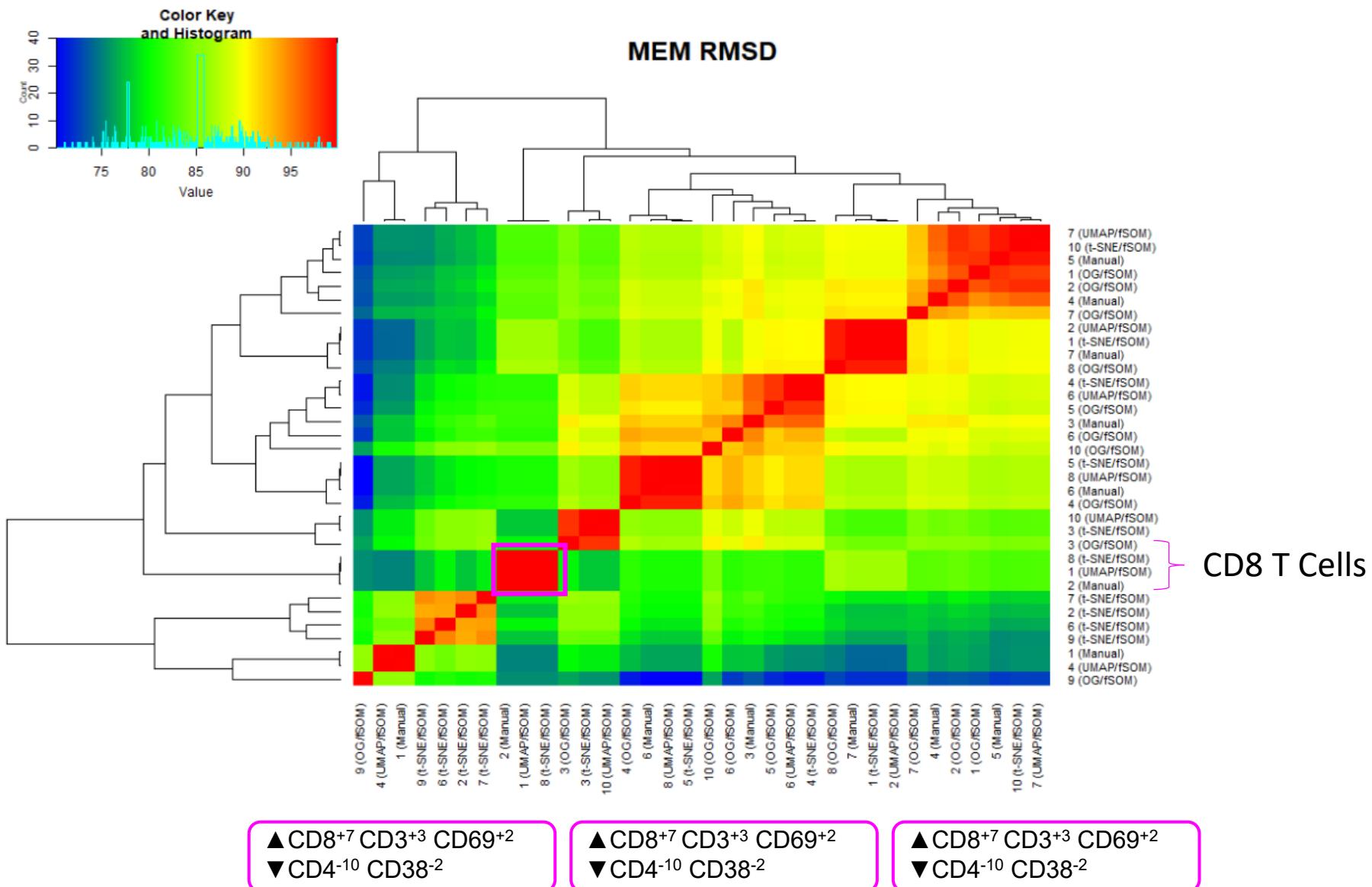
R Output for RMSD Demonstrates that Clusters are More Similar to Each other than Methods of Deriving Them



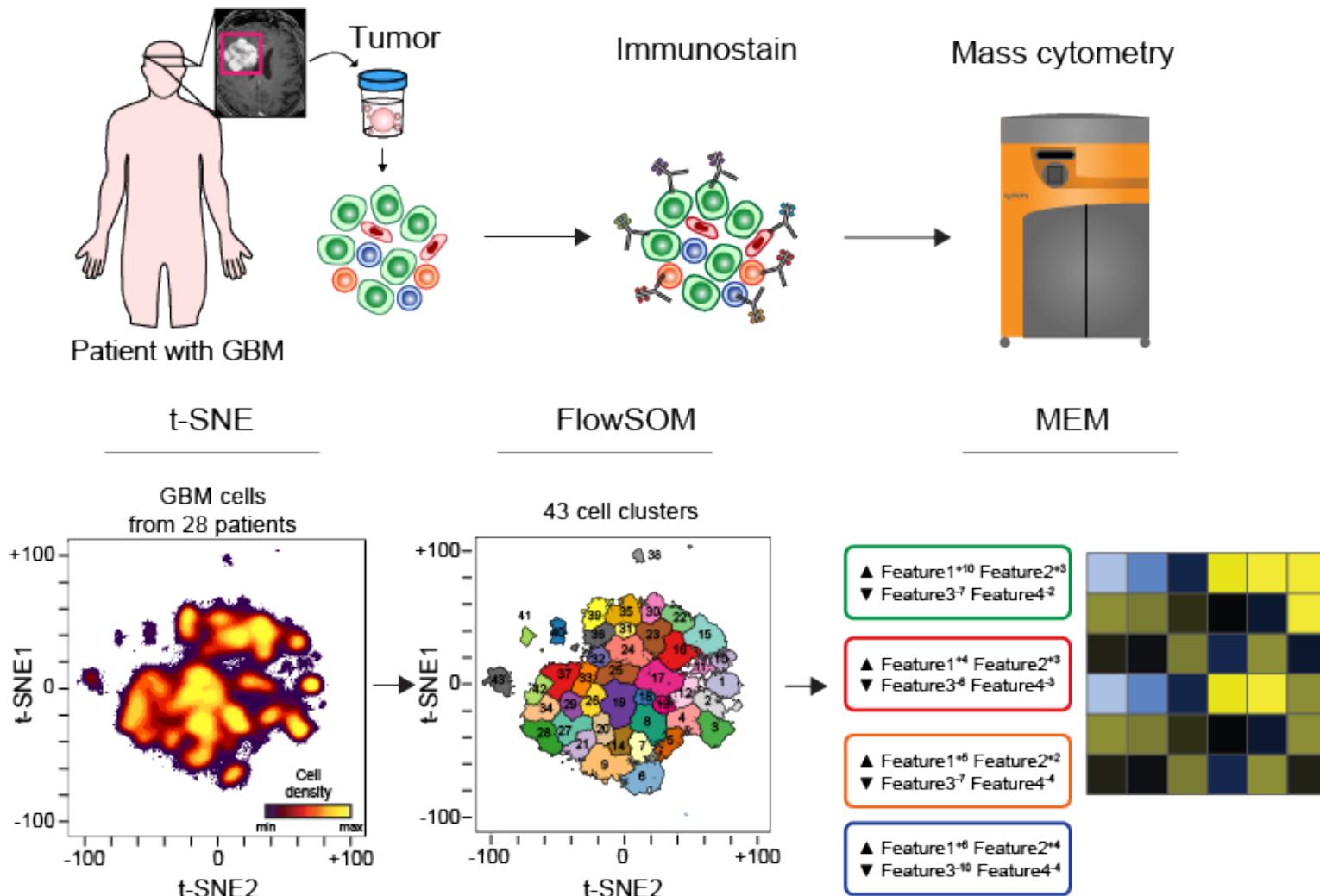
R Output for RMSD Demonstrates that Clusters are More Similar to Each other than Methods of Deriving Them



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



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

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>