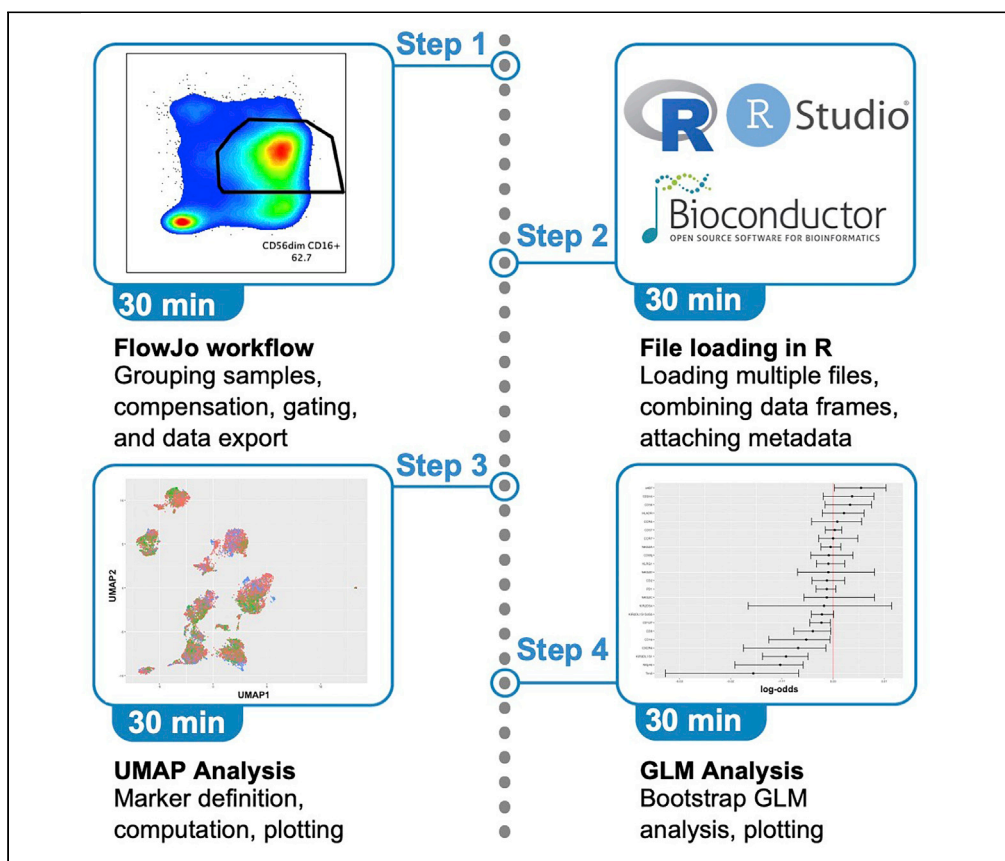


## Protocol

# Protocol for identification and computational analysis of human natural killer cells using flow cytometry and R



Kyle Kroll, R. Keith Reeves

kyle.kroll@duke.edu (K.K.)  
 keith.reeves@duke.edu (R.K.R.)

### Highlights

Computational analyses to examine the intersection of NK cells, HIV, and aging

Utilize standard flow cytometry analysis workflows that can be used with existing data

Easy application of computational analyses in R to flow cytometry data

Lightweight code that can be run on local hardware, no need for cloud computing

Identifying differential protein expression is routinely used to delineate natural killer (NK) cells from various sample cohorts. This protocol describes key steps for NK cell analysis: identifying human NK cells using flow gating, data export from FlowJo, data loading in R, dimensionality reduction and visualization with Uniform Manifold Approximation and Projection, and generalized linear modeling with CyotGLMM. These analyses can help generate potential biomarkers of interest to identify NK cells across aging, treatment groups, and others.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Kroll & Reeves, STAR Protocols  
 4, 102044  
 March 17, 2023 © 2023 The Authors.  
<https://doi.org/10.1016/j.xpro.2023.102044>



## Protocol

## Protocol for identification and computational analysis of human natural killer cells using flow cytometry and R

Kyle Kroll<sup>1,2,\*</sup> and R. Keith Reeves<sup>1,3,\*</sup><sup>1</sup>Division of Innate and Comparative Immunology, Surgical Sciences, Duke University, Durham, NC, USA<sup>2</sup>Technical contact<sup>3</sup>Lead contact\*Correspondence: [kyle.kroll@duke.edu](mailto:kyle.kroll@duke.edu) (K.K.), [keith.reeves@duke.edu](mailto:keith.reeves@duke.edu) (R.K.R.)  
<https://doi.org/10.1016/j.xpro.2023.102044>

## SUMMARY

Identifying differential protein expression is routinely used to delineate natural killer (NK) cells from various sample cohorts. This protocol describes key steps for NK cell analysis: identifying human NK cells using flow gating, data export from FlowJo, data loading in R, dimensionality reduction and visualization with Uniform Manifold Approximation and Projection, and generalized linear modeling with CytoGLMM. These analyses can help generate potential biomarkers of interest to identify NK cells across aging, treatment groups, and others.

For complete details on the use and execution of this protocol, please refer to Kroll et al. (2022).<sup>1</sup>

## BEFORE YOU BEGIN

The protocol below focuses primarily on human CD56<sup>dim</sup>CD16<sup>+</sup> NK cells. However, we have also applied this protocol to bulk cellular data, other NK subpopulations, other immune cells (T and B cells, etc.), and other species such as Rhesus (*Macaca mulatta*) and Cynomolgus macaques (*Macaca fascicularis*). This protocol assumes that single cell suspensions were isolated, stained, and analyzed on a flow cytometry platform. Prior experience and knowledge of flow cytometry gating is required. Technical explanations for justifying flow gating are outside the scope of this protocol. Prior to beginning the protocol, ensure that the computational environment is set up by installing FlowJo, R, RStudio, dplyr, ggplot2, uwot, and CytoGLMM (key resources table). The protocol described has been generalized to accept any input data that is in tabular format with cells as rows and features as columns.

## Institutional permissions

Cryopreserved human peripheral blood mononuclear cells (PBMC) were obtained from the Hawai'i Aging with HIV-1 Cohort (HAHC) at University of Hawai'i. Details of the HAHC study enrollment and clinical characterization were previously published<sup>2</sup> and approved by the University of Hawai'i Institutional Review Board (IRB) under protocol #10675. All participants signed IRB-approved informed consent forms prior to participation. Please ensure that you have the applicable institutional, local, and federal approval before beginning work.

## KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
R v4.1.2	R Core Team <sup>3</sup>	<a href="https://www.R-project.org/">https://www.R-project.org/</a>
RStudio	Posit	<a href="https://posit.co/">https://posit.co/</a>

(Continued on next page)



<i>Continued</i>		
REAGENT or RESOURCE	SOURCE	IDENTIFIER
Ggplot2	Wickham <sup>4</sup>	<a href="https://CRAN.R-project.org/package=ggplot2">https://CRAN.R-project.org/package=ggplot2</a>
Dplyr	Wickham et al. <sup>5</sup>	<a href="https://CRAN.R-project.org/package=dplyr">https://CRAN.R-project.org/package=dplyr</a>
Uwot	Melville <sup>6</sup>	<a href="https://CRAN.R-project.org/package=uwot">https://CRAN.R-project.org/package=uwot</a>
CytoGLMM	Seiler et al. <sup>7</sup>	<a href="https://www.bioconductor.org/packages/release/bioc/html/CytoGLMM.html">https://www.bioconductor.org/packages/release/bioc/html/CytoGLMM.html</a>
FlowJo	BD Biosciences	<a href="https://flowjo.com">https://flowjo.com</a>
<i>Deposited data</i>		
The data used to generate the data and figures for this protocol	This study	<a href="https://doi.org/10.6084/m9.figshare.21648488.v1">https://doi.org/10.6084/m9.figshare.21648488.v1</a>

## STEP-BY-STEP METHOD DETAILS

### Loading and performing compensation of FCS files

⌚ Timing: ~30 min (for step 1)

This section's goal is to load and perform compensation on instrument-exported flow cytometry standard (FCS) files. After completion of this section, the user will have properly compensated data that is subsequently read for downstream analyses.

1. Load FCS files for samples and single compensation controls into FlowJo ([key resources table](#); other software may be used, but these protocol steps specifically refer to FlowJo).
2. Create groups for samples, if needed ([Figure 1](#)).
3. Create a compensation matrix in FlowJo ([Figure 2](#)).
  - a. Select the "Compensation" group.
  - b. Using the navigation ribbon, go to "Workspace" and then "Compensation" ([Figure 2A](#)).
4. Ensure the proper negative and positive peaks for each parameter are set ([Figure 2B](#)).

**Note:** For projects using a universal negative control, ensure that the negative gate on the marker histogram captures the entire peak that is at or near 0.

- a. Then for each compensation sample, select the brightest peak on the histogram.

**Note:** For projects using per-tube negative controls, ensure that both a positive and negative gate are set such that the positive gate captures the brightest peak, and the negative gate captures the peak at or near 0.

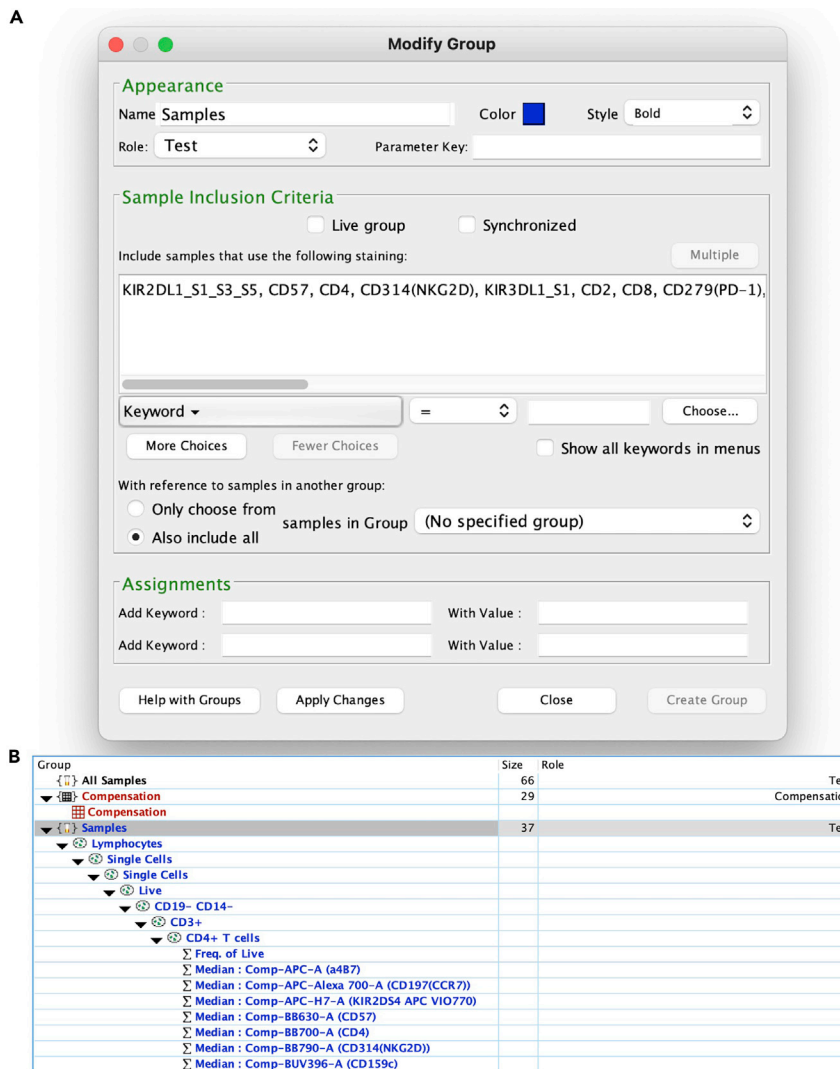
5. Apply the newly created compensation matrix to sample groups by clicking "Apply to group", which is found in the top left of the "Compensation" window.

**Note:** Ensure that compensation is applied to the groups that contain samples.

6. Open the compensation matrix and modify the matrix as needed by clicking "Edit Compensation Matrix" to ensure accurate compensation ([Figure 2C](#)).
  - a. For more detail on compensation errors and how to correct them, see the [troubleshooting](#) section.

### Gating NK cells

⌚ Timing: ~30 min (for step 7)

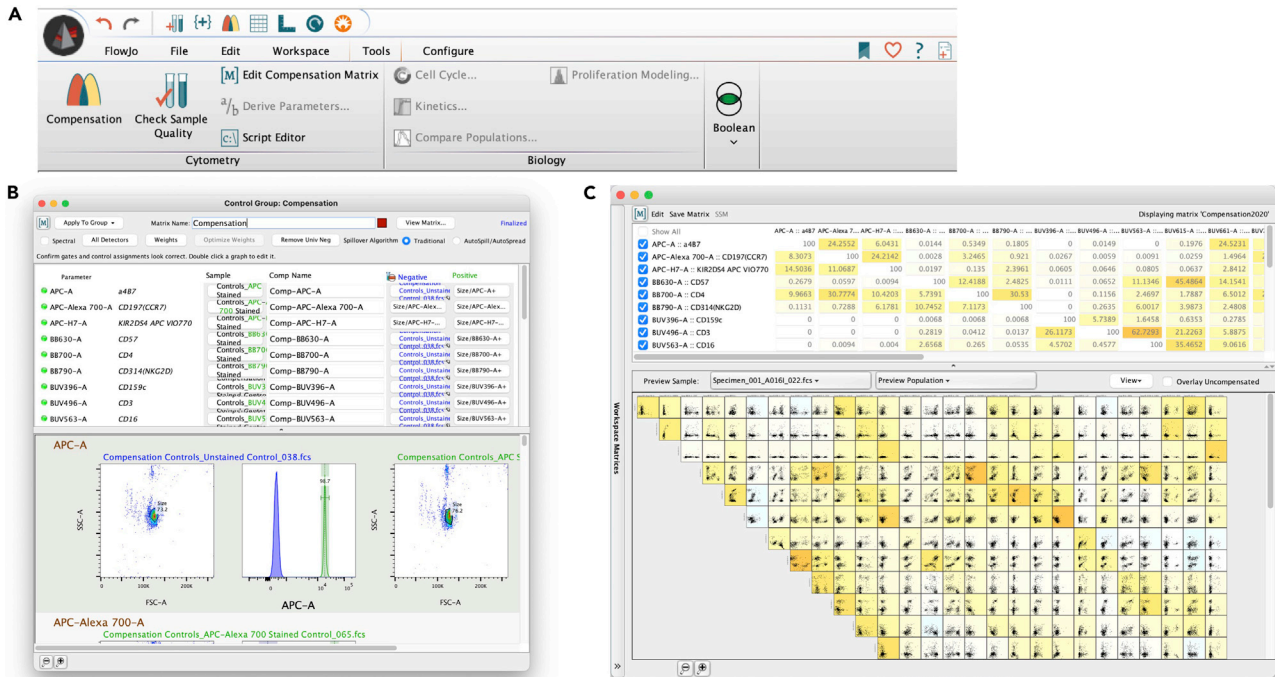


**Figure 1. Window configuration for creating and setting sample groups**

(A) Create or modify group settings panel, includes group name, role, staining, etc.  
(B) Main window showing groups and gating trees for the group.

The goals for this section are to properly identify NK cells using accepted gating strategies for human NK cells. These NK cells are defined as lineage (CD3, CD14, CD19) negative, CD56<sup>+</sup>, CD16<sup>+/-</sup> lymphocytes. Depending on the user's population of interest, these initial gating steps may differ. Initial cleanup gating is utilized to exclude non-lymphocytes, doublets, and non-viable cells. After completion of this section, the user will have properly identified CD56<sup>dim</sup>CD16<sup>+</sup> NK cells.

7. Open a sample file and select mononuclear cells based on initial gating of FSC-A and SSC-A (Figure 3; P1).
8. Open mononuclear cells and gate for single cells only using sequential gating for both FSC-A and FSC-H, and SSC-A and SSC-H (Figure 3; P2; P3).
9. Gate for live cells by gating on the negative population of the chosen viability dye (Figure 3; P4).
  - a. We recommend being conservative with gating for viable cells. Dead cells can have a great impact on downstream marker expression profiles.
10. For identifying NK cells, use a general signature of CD3<sup>-</sup> CD14<sup>-</sup> CD19<sup>-</sup> CD56<sup>+</sup>.



**Figure 2. Workflow for creating and modifying compensation matrices**

(A) Main tool ribbon showing compensation tools including creating compensation (left-most “Compensation” tool) and “Edit Compensation Matrix.” (B) Compensation creation window including definitions for negative and positive peaks. (C) Compensation matrix editor view including NxN displays for each channel and cells to directly modify compensation values.

11. First, plot CD14 and CD19 and gate the double negative population (Figure 3; P5).

**Note:** It is critical to ensure that the population gated is negative for both of these markers. Monocytes and B cells have very different expression profiles from NK cells and can skew downstream results if a substantial number of contaminating cells are present.

12. Next, plot CD4 and CD3 and gate for the double negative population (Figure 3; P6).

**Note:** It is critical that no contaminating T cells are present in downstream analysis. T cells, which can be similar to NK (especially CD8<sup>+</sup> T Cells), can greatly skew downstream analysis.

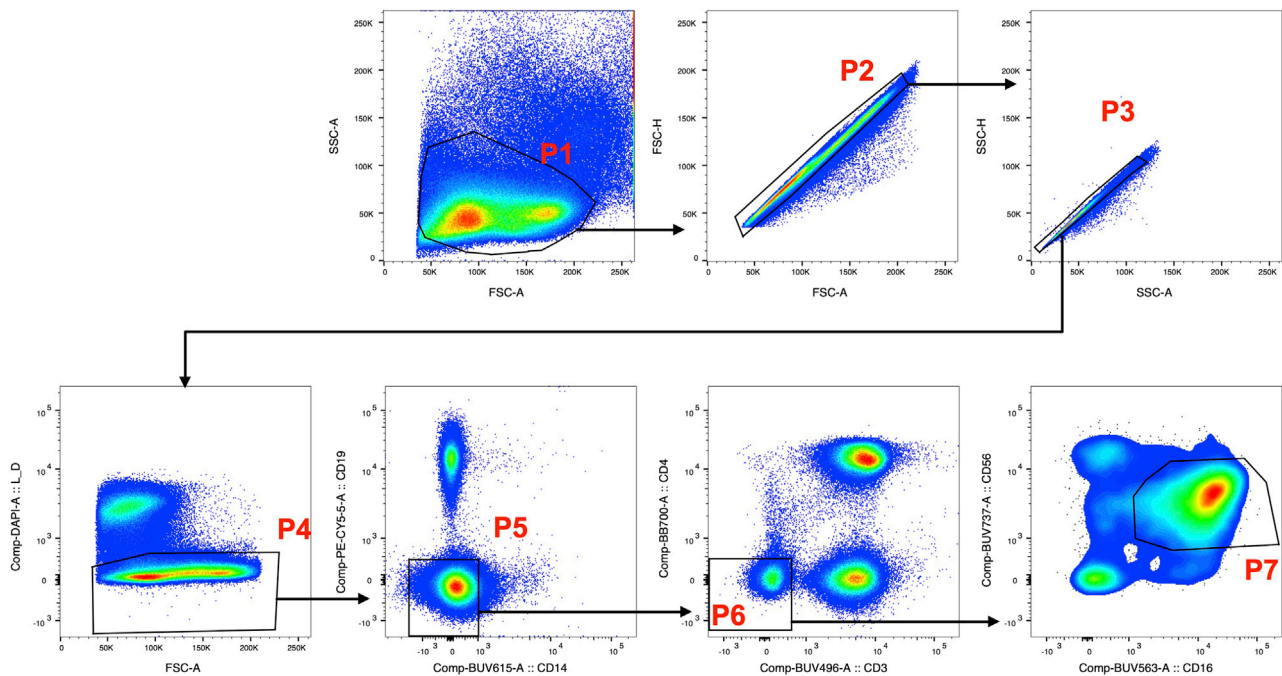
13. Plot CD56 and CD16 on CD14<sup>-</sup> CD19<sup>-</sup> CD4<sup>-</sup> CD3<sup>-</sup> population and gate for CD56<sup>dim</sup>CD16<sup>+</sup> cells. These are the NK cells of interest (Figure 3; P7).

**Note:** This selection can be tailored to the user’s goals. We focus on the cytotoxic CD56<sup>dim</sup>CD16<sup>+</sup> population, but the immunoregulatory CD56<sup>bright</sup>CD16<sup>-</sup> population can be of interest as well.

14. Select the entire gating path and drag it to the group folder to apply this gating to all samples.  
15. Check each sample to ensure accurate gating and modify as needed.

### Exporting NK populations as CSV files

⌚ Timing: ~30 min (for step 16)



**Figure 3. Gating strategy to define NK cells in humans**

Arrows denote sequential gating steps starting with P1 – mononuclear cells, P2 – single cells based on forward scatter, P3 – single cells based on side scatter, P4 – live cells based on negative staining of amine dye, P5 – cells negative for CD19 (B cell) and CD14 (monocyte) lineage markers, P6 – cells negative for CD3 (T cell lineage) and CD4 (other cell types), and P7 – NK cells based on expression of CD56<sup>dim</sup>CD16<sup>+</sup> cells.

In this following section, we will be exporting the compensated and gated NK populations to the generic comma-separated values (CSV) file format. This file format can be utilized in a wide arrange of analyses and software.

16. Open the group folder and on the first sample select the “NK cell” population.
17. Go to the navigation ribbon and select “Edit” and “Select Equivalent Nodes” to select all NK cells (Figure 4A).
18. Export files using Command + E on Mac or Ctrl + E on Windows, or by going to “File” -> “Export/Concatenate” (Figure 4B).
19. In the export window at the file type dropdown, select “CSV – Channel Values” (Figure 4C).
20. Under “Parameters” select “All compensated parameters” (Figure 4C).
21. Select an export location and export all files.

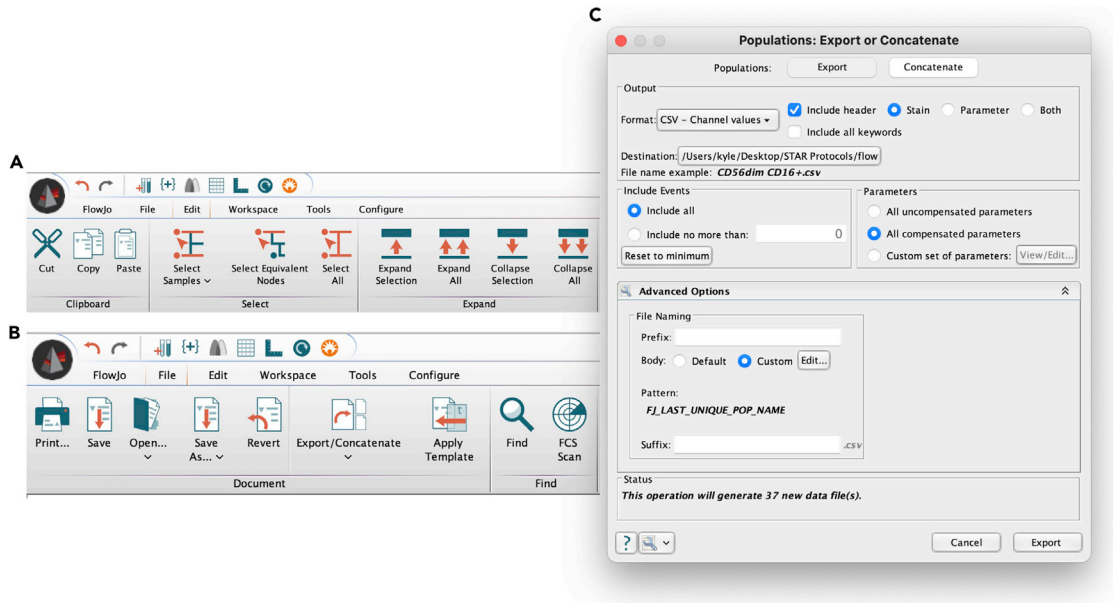
### Creation of metadata file

⌚ Timing: ~30 min (for step 22)

Metadata is a critical component of every project. This file will serve as the guide for associating files to their appropriate sample IDs, group status, and any other pertinent information needed for downstream analyses. There are many ways of both manual and automated methods of creating these types of files, but for the purpose of this protocol the user will be instructed to perform this step manually.

22. Create a new file in Microsoft Excel or other tabular editing software.
23. Create at least two columns with the headers “Sample” and “Group”.
24. For each sample exported in the previous section, create a row with the sample identifier in “Sample” and relevant grouping factor in “Group”.





**Figure 4. Data exporting in FlowJo**

- (A) Tool ribbon displaying population selection tools.  
 (B) Tool ribbon displaying various exportation tools including “Export/Concatenate”.  
 (C) Export/Concatenate dialog window including output format, file destination, parameter selection, and naming patterns.

- a. The sample identifier can be any combination of sample characteristics or sequential identifiers. For example, here we use a prefix “Sample” followed by sequential numbering to assign a unique ID to each sample.

25. Save file as a CSV in a convenient location.

**Note:** Make sure to save as CSV and not an Excel file since reading into R requires a CSV file. Excel files require a different method to read into R. This can be done but will require user to use their own code.

### Loading data into R and creating sample data.frame

⌚ Timing: ~30 min (for step 26)

The first step of computational analyses in R is loading data and getting it into a format suitable for downstream steps. In this section, we load all of the exported CSV files, combine them into a single R data.frame, and finally read in the metadata and add that information to the data.frame. This step is an important step, so care should be taken to examine the data as its loaded to ensure that it is loaded correctly.

26. Create a new Rscript file in RStudio.
27. Set the working directory to the folder that exported CSV files were placed in steps 21 and 25.

```
>setwd ("<PATH>/<TO>/<PROJECT>")
```

28. Install the required packages and load them.

```
>if (!require("BiocManager", quietly = TRUE)) {  
  install.packages("BiocManager")  
}  
>BiocManager::install("CytoGLMM")  
>install.packages(c("ggplot2", "uwot", "dplyr"))  
>library(ggplot2)  
>library(uwot)  
>library(dplyr)  
>library(CytoGLMM)
```

29. Iterate through each CSV of data and read it into memory. Then merge them into a single data frame.

**Note:** For this protocol, the folder structure used is a parent directory that contains a subdirectory named "flow" which holds the exported FlowJo CSV files. The metadata file is found in a subdirectory of the parent project folder named "meta." The structure should be as follows:

PROJECT FOLDER:

⇒ meta/

    metadata.csv.

⇒ flow/

    All sample CSV files.

```
>csv.list <- list()  
># Iterate through each CSV  
># Adding the sample ID and adding it to our `csv.list`  
>csv.files <- list.files(paste0(getwd(), "/flow/"), pattern = ".csv")  
># For each file, read it in and add the sample ID which is in the filename  
>for(i in 1:length(csv.files)) {  
  > sample.id <- strsplit(csv.files[i], "_")[[1]][4] # Split filename and take 3rd element  
  > tmp <- read.csv(csv.files[i], check.names = F) # read in the CSV  
  > try({ # Some samples had no viable cells. Handle that exception here  
    > tmp$Sample <- sample.id # Create a new column for the sample ID  
    > tmp <- as.data.frame(tmp)  
  })  
  > csv.list[[i]] <- tmp # Append the CSV as a dataframe to the list  
>}  
># Now let's combine each of the csv files in csv.list  
>df <- do.call("rbind", csv.list)
```



30. Next, read in the newly created metadata file and match metadata information to the data-frame. You will need to set the path to your respective metadata file. See [Table 1](#) for example data.frame output.

```
>metadata <- read.csv("../meta/metadata.csv", check.names = F)
># Add in metadata information
>df <- merge(df, metadata[, c("Sample", "Group", "Age")], by = "Sample", all=T)
># Reorder the data frame (personal preference)
># Change this for your specific dataset
>df <- df[, c("Sample", "Group", "Age", colnames(df)[2:29])]
># Drop samples that were NA or had no cells
>df <- df[complete.cases(df),]
```

⚠ **CRITICAL:** When combining data from several files, always examine the resulting data-frame to ensure that all data were loaded correctly. You may wish to check data columns for the correct class of variable.

### Uniform Manifold Approximation and Projection (UMAP)

⌚ **Timing:** ~30 min (for step 31)

One of the first steps in many computational analyses is dimensionality reduction and visualization. These steps serve several purposes, including identifying distinct biological clustering of the data or identifying potential biases in the data (sample, technical, etc.). This section utilizes the UMAP for its speed and preservation of local relationships, but many other algorithms can be used, such as t-Distributed Stochastic Neighbor Embedding (t-SNE), Principal Component Analysis (PCA), etc.

31. Before performing UMAP analysis, the phenotypic markers of interest need to be defined. These will be specific to your dataset. While here, we will also rename the columns for downstream CytoGLMM analysis.

**Note:** Due to the complexity of high dimensional flow cytometry data, traditional gating workflows utilizing sequential biplots becomes extremely time intensive and prone to user bias. In an attempt to overcome these challenges, we utilize UMAP to create a two-dimensional embedding of the data to identify clusters of phenotypically similar cells. These clusters can then be labeled and interrogated in finer detail.

```
>colnames(df) <- c("Sample", "Group", "Age", "a4B7", "CCR7", "KIR2DS4", "CD57",
"CD4", "NKG2D", "NKG2C", "CD3", "CD16", "CD14", "HLADR", "CD56", "CD127", "KIR3DL1S1", "CD2",
"CD8", "PD1", "CD244", "NKp46", "Tim3", "CXCR6", "LD", "KIR2DL1S1S3S5", "CD85j", "KLRG1",
"CD19", "CCR5", "NKG2A")
>markers <- c("a4B7", "CCR7", "KIR2DS4", "CD57", "NKG2D", "NKG2C", "CD16", "HLADR", "CD56",
"CD127", "KIR3DL1S1", "CD2", "CD8", "PD1", "CD244", "NKp46", "Tim3", "CXCR6", "KIR2DL1S1S3S5",
"CD85j", "KLRG1", "CCR5", "NKG2A")
```

**Table 1. Example data.frame output**

	Sample	Group	Age	a4B7	CCR7
1	Sample1	Viremic	31.91	305	258
2	Sample1	Viremic	31.91	239	223
3	Sample1	Viremic	31.91	270	231
4	Sample1	Viremic	31.91	342	240
5	Sample1	Viremic	31.91	354	217
6	Sample1	Viremic	31.91	271	191

32. Next, call the UMAP function with default parameters.

- a. Optionally, you can run several iterations with differing parameters, but default parameters work for most use cases.

```
>um.res <- umap(scale(df[, markers], center=T, scale=T), verbose=T)
```

33. Rename the um.res columns to something more readable and combine the original data.frame into UMAP results.

```
>colnames(um.res) <- c("UMAP1", "UMAP2")
>um.df <- cbind(df, um.res)
```

34. Plot the UMAP results using ggplot2.

**Note:** You can modify the aes() settings to change which metadata or fluorescent values are used for coloring points.

**Note:** Some datasets can contain hundreds of thousands or millions of events. One method to speed up graphic plotting is to subset the data. Here for each sample, a random sample of 300 cells or all cells are kept.

**Note:** Additionally, due to how ggplot2 plots data points, we need to mix the data frame to ensure that samples read first do not get hidden by samples read in later.

```
>um.df.small <- um.df %>%
  group_by(Sample) %>%
  > sample_n(min(n(), 300))
>um.df.small <- as.data.frame(um.df.small[sample(1:nrow(um.df.small)), ])
>ggplot(um.df.small,
  aes(x = UMAP1,
    y = UMAP2,
    color = Group)) +
  geom_point(size = 0.5)
```

35. Additionally, iterate through markers and plot a relative expression graph for each marker.

```
>for(marker in markers) {
>  p <- ggplot(um.df.small,
              aes(x = UMAP1,
                  y = UMAP2,
                  color = um.df.small[, marker])) +
  geom_point(size = 0.5) +
  scale_color_gradientn(colors = c("blue", "white", "red")) +
  labs(color=marker)
> ggsave(paste0("~/Desktop/STAR Protocols/figures/UMAP/", marker, ".pdf"), p, width = 11,
         height = 8.5, units = "in", device = "pdf")
>}
```

**△ CRITICAL:** Depending on your dataset, you may need to perform a normalization step for each marker. Typically, either quantile clamping or z-scores can be used. For example, if z-score scaling is required, on the `ggplot()` call, replace `>color = um.df.small[, marker]` with `>color = scale(um.df.small[, marker], center=T, scale=T)`

### Generalized linear modeling using CytoGLMM

⌚ Timing: ~30 min (dependent on number of cells and samples)

In this section, we apply generalized linear modeling (GLM) on the data to identify potential biomarkers of the chosen group comparisons. The CytoGLMM package was developed for applying GLM to flow cytometry data. Flow cytometry data contains thousands of cells from a single sample which presents a unique challenge in that these cells are independent yet representative of a single source.

The following steps are adapted from the CytoGLMM vignettes (<https://christofseiler.github.io/CytoGLMM/articles/CytoGLMM.html>).

36. Select two comparisons to make. For this example, select the healthy donor (HD) group and compare the under 45 to over 45 groups.

**Note:** The purpose of these analyses is to identify markers that are predictive of each group. For example, these could be biomarkers useful for identifying diseased samples or identifying markers of the treatment group. Flow cytometry data generates millions of data points but are confined to dozens of samples. CytoGLMM leverages these large data sets to report log-odds ratios of marker expression defining group membership.

- a. First, create a categorical group name for age groups.
- b. Set samples, groups, and age groups to factors.
- c. This step is time-dependent on the number of cells. Optionally, downsample to reduce computation time.

```
>df$Age.Group <- ifelse(df$Age <= 45, "Under45", "Over45")
>df$Sample <- as.factor(df$Sample)
>df$Group <- factor(df$Group, levels = c("HD", "cART", "Viremic"))
># Can only compare two groups so we'll compare HD ages
# also need to downsample to 1000 events/sample since this step can take a long time
>glm.df <- df %>% filter(Group == "HD") %>% group_by(Sample, Group) %>% sample_n(if(n() < 1000)
n() else 1000)
>glm.df$Age.Group <- factor(glm.df$Age.Group, levels = c("Under45", "Over45"))
>glm.df[, markers] <- scale(glm.df[, markers], center=T, scale=T)
```

37. Perform GLM analysis with the 'cytoglm' function.

- a. Optionally, change the 'num\_cores' to increase how many CPU cores are utilized.

```
>glm_fit = CytoGLMM::cytoglm(glm.df,
  num_boot = 1000,
  protein_names = markers,
  condition = "Age.Group",
  group = "Sample",
  num_cores = 1)
```

38. Plot the results.

```
>plot(glm_fit, order=T)
```

△ **CRITICAL:** Grouping variables will differ between projects. Ensure proper grouping is set up for the specific dataset being examined.

## EXPECTED OUTCOMES

Expected outcomes include a data.frame of sample and protein expressions (Table 1). UMAP relative marker expression plots for all markers defined, as in Figure 5 (not all markers are shown). In addition, GLM log-odds ratio plots are generated for comparisons of choice as seen in Figure 6.

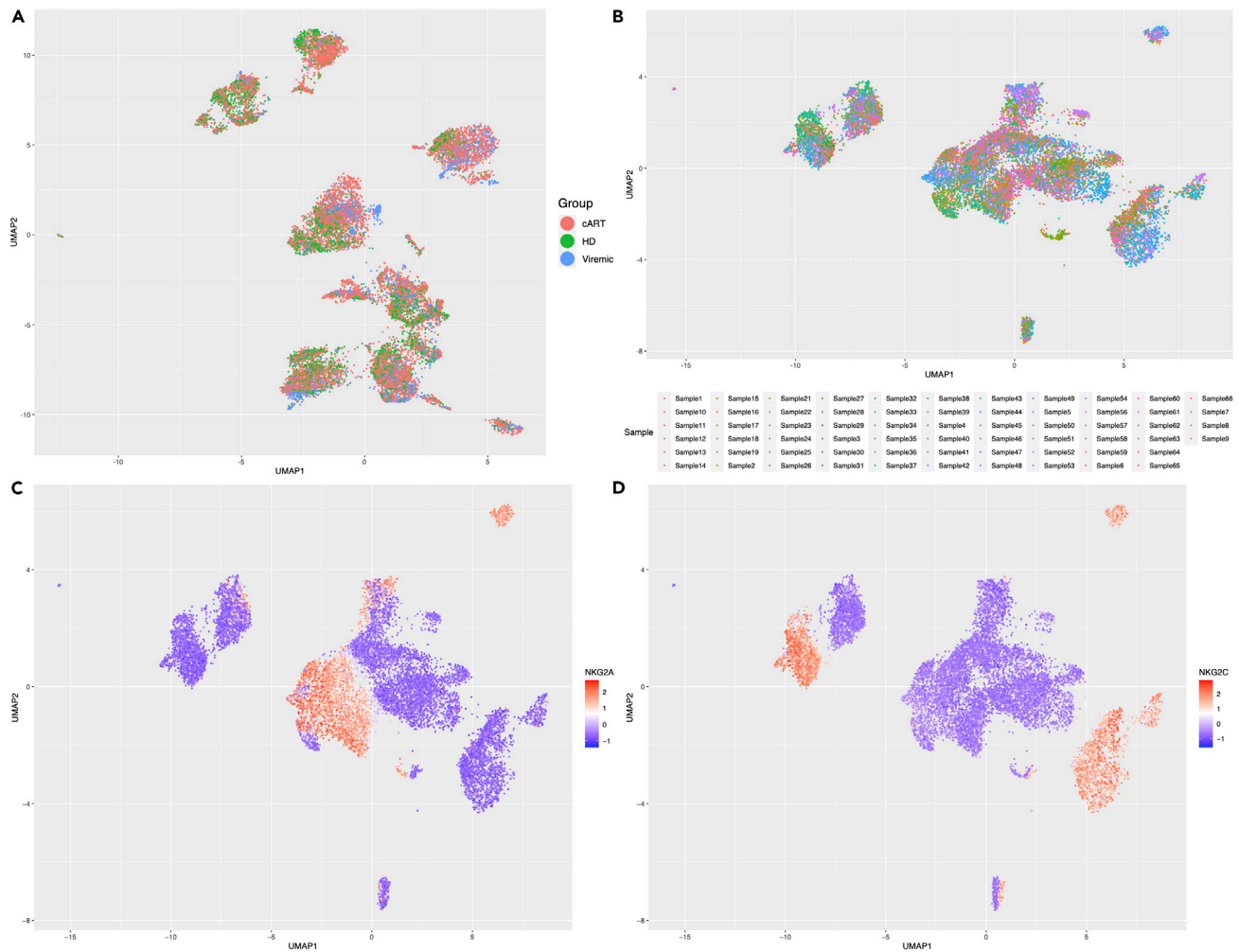
## LIMITATIONS

While this protocol should, in theory, be applicable to any cytometry platform (flow, mass, spectral, etc.), it has only been tested and utilized using flow cytometry with BD Biosciences cytometers. In addition, only FlowJo software has been used and tested for initial gating and data exportation, but this protocol should also be applicable to other analysis platforms (FCS Express, CytoBank, etc.).

## TROUBLESHOOTING

### Problem 1

When examining the compensation matrix there are one or more fluorophores that appear over- or undercompensated (Figure 7).



**Figure 5. UMAP plots of NK cells from different sample groups**

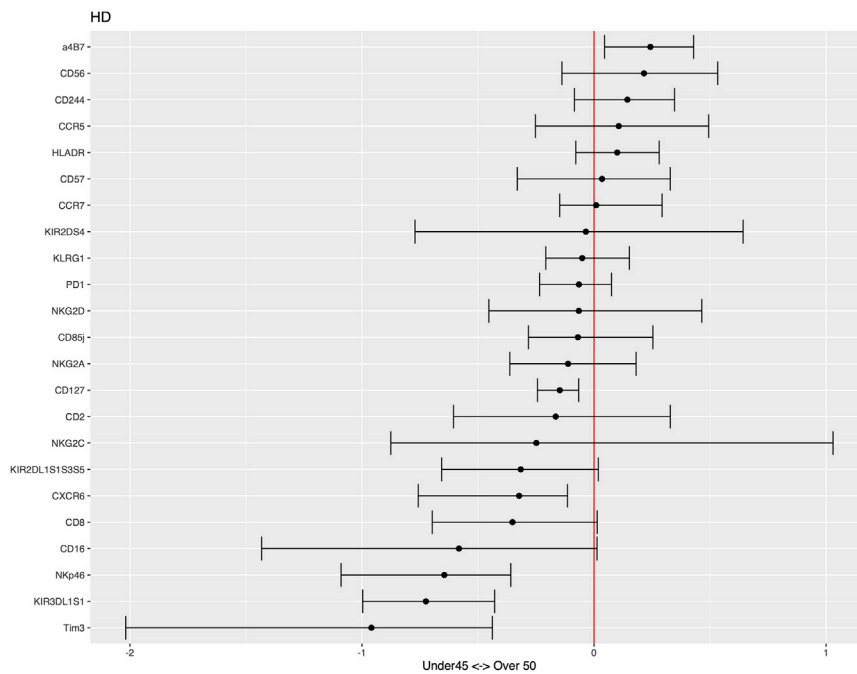
- (A) UMAP plot showing color coded distribution of NK cells from HD, cART, and Viremic groups.  
 (B) UMAP plot showing color coded distribution of NK cells from sample source.  
 (C) Relative marker expression showing z-score scaled NKG2A expression.  
 (D) Same as (C) but showing z-score scaled NKG2C relative expression.

### Potential solution

Create a copy of the current compensation matrix by clicking the “Edit” button in the “Matrix Editor” window. Next, determine if the parameter is over- or undercompensated. Overcompensated parameters will be pushed onto the axis limits. These are corrected by finding the corresponding cell in the matrix and lowering the value. Undercompensated parameters will be pushed towards the opposite axis (for example, if the undercompensated parameter is on the x-axis, the population will be pushed towards the top of the graph). Undercompensated parameters can be fixed by finding the corresponding cell in the matrix and lowering the cell value. See below image for example of under-, over-, and correctly compensated parameters.

### Problem 2

Error in R when combining input CSV files in step 30.



**Figure 6. CytoGLMM log-odds ratio plot**

Plot displays the log-odds ratio for each marker being a predictor for Under 45 or Over 45 samples. Dot represents mean log-odds ratio of bootstrap runs and bars represent the 95% confidence intervals.

### Potential solution

Ensure that stain names are consistent between all workspaces. The usage of analysis templates can help alleviate this issue.

### Problem 3

Error in R when performing GLM in step 39 that may look similar to the following:

```
Error: BiocParallel errors
  1 remote errors, element index: 1
  999 unevaluated and other errors
  first remote error: y values must be 0 <= y <= 1
```

### Potential solution

Ensure grouping variables are set to factors by either calling:

```
>as.factor(df$`VARNAME`)
```

Or

```
>factor(df$`VARNAME`, levels = c("Level101", "Level102"))
```

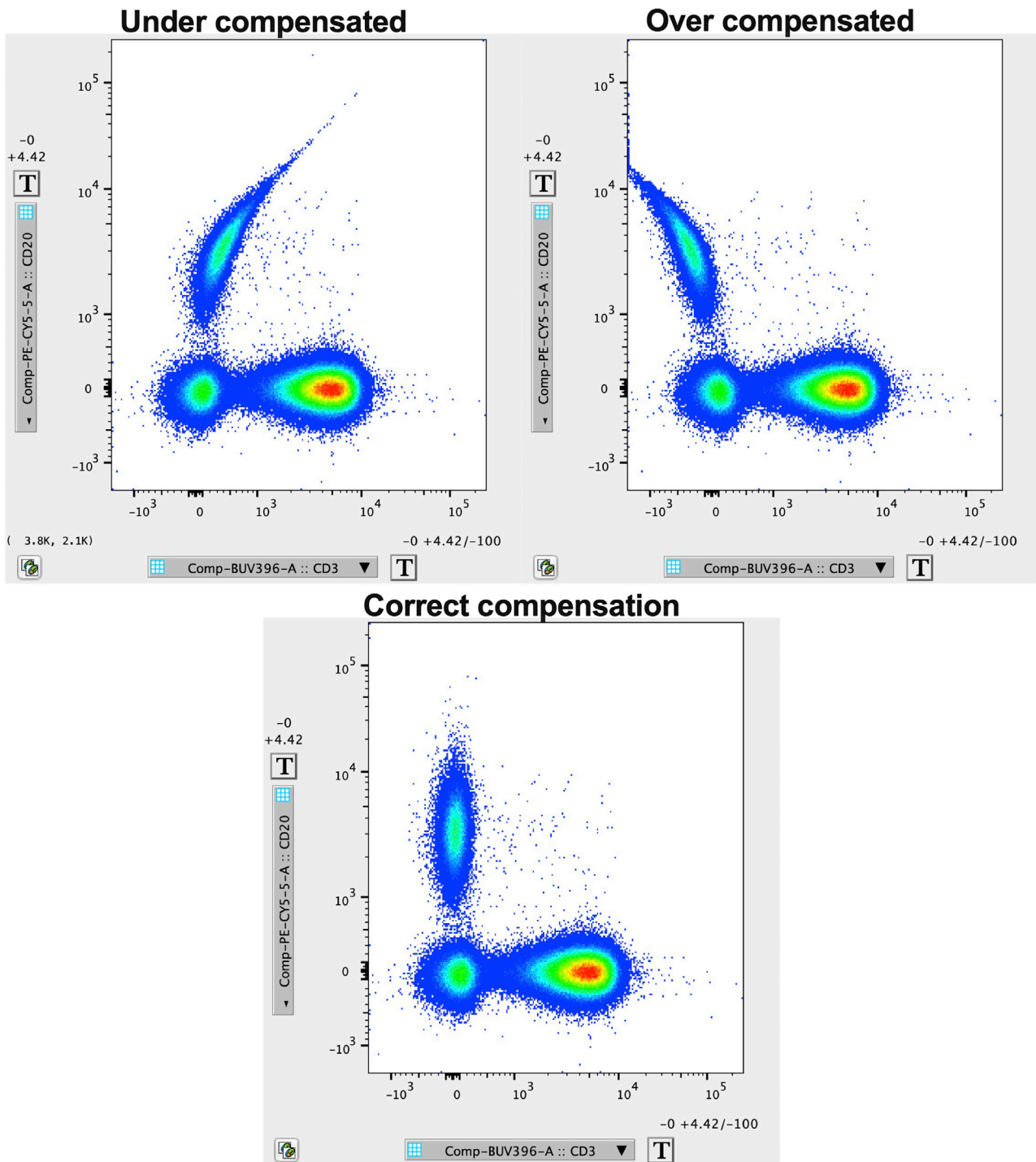


Figure 7. Representative plot of troubleshooting compensation

## RESOURCE AVAILABILITY

### Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, R. Keith Reeves ([keith.reeves@duke.edu](mailto:keith.reeves@duke.edu)).



### Materials availability

This study did not generate new unique reagents.

### Data and code availability

The data used to generate the data and figures for this protocol are freely available from <https://doi.org/10.6084/m9.figshare.21648488.v1>.

### ACKNOWLEDGMENTS

The authors thank Thomas A. Premeaux, Michael J. Corley, Scott Bowler, and Lishomwa C. Ndhlovu of Cornell University for the samples used for this protocol and original study. The authors also thank Cecilia M. Shikuma at the University of Hawai'i. This research was supported by National Institutes of Health (NIH) grants R01AI120828, R01AI143457, and R01AI161010 (to R.K.R.). We also acknowledge support from the CVVR Flow Cytometry and Harvard University Center for AIDS Research Advanced Laboratory Technologies Core (P30 AI060354). The authors thank Michelle Lifton for technical assistance and panel design in flow cytometry.

### AUTHOR CONTRIBUTIONS

K.K. and R.K.R. designed the study. K.K. ran assays, analyzed data, wrote code, and wrote the manuscript. Both authors edited the final document.

### DECLARATION OF INTERESTS

The authors declare no competing interests.

### REFERENCES

1. Kroll, K.W., Shah, S.V., Lucar, O.A., Premeaux, T.A., Shikuma, C.M., Corley, M.J., Mosher, M., Woolley, G., Bowler, S., Ndhlovu, L.C., and Reeves, R.K. (2022). Mucosal-homing natural killer cells are associated with aging in persons living with HIV. *Cell Rep. Med.* 3, 100773. <https://doi.org/10.1016/j.xcrm.2022.100773>.
2. Valcour, V., Shikuma, C., Shiramizu, B., Watters, M., Poff, P., Selnes, O., Holck, P., Grove, J., and Sacktor, N. (2004). Higher frequency of dementia in older HIV-1 individuals: the Hawaii Aging with HIV-1 Cohort. *Neurology* 63, 822–827. <https://doi.org/10.1212/01.wnl.0000134665.58343.8d>.
3. R Core Team (2021). R: A Language and Environment for Statistical Computing (R Foundation for Statistical Computing). <https://www.R-project.org/>.
4. Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis* (Springer-Verlag).
5. Wickham, H., François, R., Henry, L., and Müller, K. (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.10. <https://CRAN.R-project.org/package=dplyr>.
6. Melville, J. (2022). uwot: The Uniform Manifold Approximation and Projection (UMAP) Method for Dimensionality Reduction. R package version 0.1.14. <https://CRAN.R-project.org/package=uwot>.
7. Seiler, C., Ferreira, A.-M., Kronstad, L.M., Simpson, L.J., Le Gars, M., Vendrame, E., Blish, C.A., and Holmes, S. (2021). CytoGLMM: conditional differential analysis for flow and mass cytometry experiments. *BMC Bioinf.* 22, 137. <https://doi.org/10.1186/s12859-021-04067-x>.