CITE-seq analysis proof-of-concept

Patrick Cherry

2024-06-20

```
cat(readLines("CITE-seq_data_fetch.R"), sep = "\n")
```

```
if(!require(GEOquery)){BiocManager::install("GEOquery")}
library(GEOquery)
library(fs)

# https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE100866
# https://www.nature.com/articles/nmeth.4380#article-info

gse_id <- "GSE100866"

gse <- getGEO(gse_id, GSEMatrix = TRUE)
show(gse)

filePaths <- getGEOSuppFiles(gse_id)

fs::file_delete(fs::path("data", gse_id, fs::path_file(fs::dir_ls(gse_id))))
fs::file_move(fs::path(gse_id), "data")
```

```
fs::dir_ls(data_dir) |> fs::path_file()
```

```
[1] "GSE100866_CBMC_8K_13AB_10X-ADT_clr-transformed.csv.gz"
[2] "GSE100866_CBMC_8K_13AB_10X-ADT_umi.csv.gz"
[3] "GSE100866_CBMC_8K_13AB_10X-RNA_umi.csv.gz"
[4] "GSE100866_CD8_merged-ADT_clr-transformed.csv.gz"
[5] "GSE100866_CD8_merged-ADT_umi.csv.gz"
[6] "GSE100866_CD8_merged-RNA_umi.csv.gz"
[7] "GSE100866_PBMC_vs_flow_10X-ADT_clr-transformed.csv.gz"
[8] "GSE100866_PBMC_vs_flow_10X-ADT_umi.csv.gz"
[9] "GSE100866_PBMC_vs_flow_10X-RNA_umi.csv.gz"
```

Read in data

count data

```
cbmc_rna <-
  as.sparse(
    read.csv(
      path(data_dir, "GSE100866_CBMC_8K_13AB_10X-RNA_umi.csv.gz"),
      sep = ",", header = TRUE, row.names = 1))
```

```
dim(cbmc_rna)
```

```
[1] 36280  8617
```

`dim()` outputs (columns, rows); the CBMC matrix (cord blood mono-nuclear cells) contains 36280 features and 8617 samples (single cell droplets).

```
cbmc_rna <- CollapseSpeciesExpressionMatrix(cbmc_rna,
                                            prefix = "HUMAN_", controls = "MOUSE_",
                                            ncontrols = 100)
```

```
dim(cbmc_rna)
```

```
[1] 20501  8617
```

`CollapseSpeciesExpressionMatrix()` is a convenience function for slimming down a multi-species expression matrix, when only one species is primarily of interest. Given the default parameter of `ncontrols = 100`, this command keeps only the top 100 features detected from each species in each sample. This matrix went from 36280 to 20501 features, which is a 43% reduction.

ADT UMI matrix

```
cbmc_adt <-
  as.sparse(
    read.csv(
      path(data_dir, "GSE100866_CBMC_8K_13AB_10X-ADT_umi.csv.gz"),
      sep = ",", header = TRUE, row.names = 1))
```

```
dim(cbmc_adt)
```

```
[1]   13 8617
```

Quick matrix QC

```
testthat::expect_equal(dim(cbmc_rna)[2] == dim(cbmc_adt)[2],
                       TRUE)
```

The number of rows (samples / UMIs) matches the RNA counts matrix; we have corresponding sample data.

```
length(intersect(colnames(cbmc_rna), colnames(cbmc_adt))) /
  length(union(colnames(cbmc_rna), colnames(cbmc_adt)))
```

```
[1] 1
```

And the names of the samples all match.

Seurat object & cluster

```
cbmc <- CreateSeuratObject(counts = cbmc_rna)
```

```
Warning: Feature names cannot have underscores ('_'), replacing with dashes
('-')
```

```
cbmc <- NormalizeData(cbmc)
```

```
Normalizing layer: counts
```

```
cbmc <- FindVariableFeatures(cbmc)
```

```
Finding variable features for layer counts
```

```
cbmc <- ScaleData(cbmc)
```

```
Centering and scaling data matrix
```

```
cbmc <- RunPCA(cbmc, verbose = FALSE)
ElbowPlot(cbmc, ndims = 50)
```

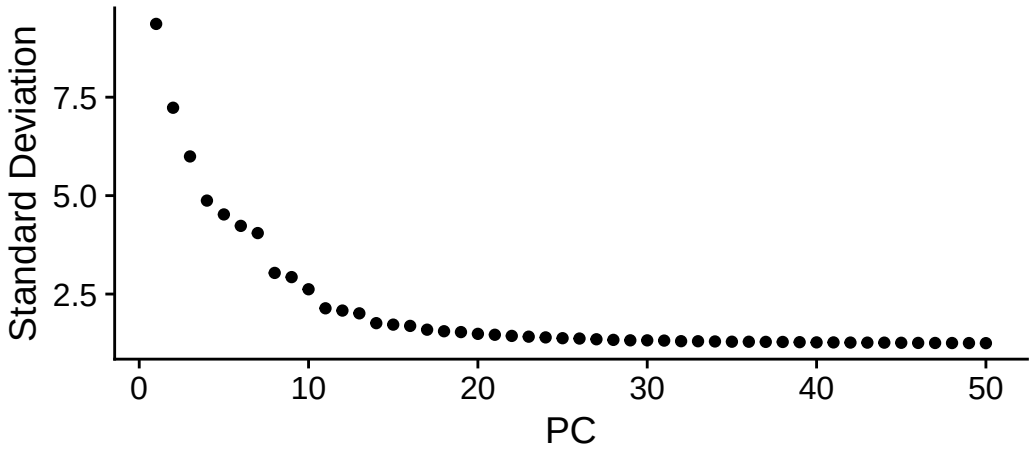


Figure 1: Elbow plot or scree plot of principle components computed from scRNA-seq counts

The elbow plot above shows some interesting PC influence behavior. There are some clusters PCs (like 4-7, 8-10, and 11-13) that make it less clear where the "elbow" of influence trend is. To be very safe, we can keep up to PC 25, where the trend approaches a horizontal line.

```
cbmc <- FindNeighbors(cbmc, dims = 1:25)
```

```
Computing nearest neighbor graph
```

```
Computing SNN
```

```
cbmc <- FindClusters(cbmc, resolution = 0.8)
```

```
Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck

Number of nodes: 8617
Number of edges: 343912

Running Louvain algorithm...
Maximum modularity in 10 random starts: 0.8613
Number of communities: 19
Elapsed time: 0 seconds
```

```
cbmc <- RunTSNE(cbmc, dims = 1:25, method = "FIt-SNE")
```

```
cbmc_rna_markers <-
  FindAllMarkers(cbmc,
                 max.cells.per.ident = 100, logfc.threshold = log(2),
                 only.pos = TRUE, min.diff.pct = 0.3, verbose = FALSE)
```

The following cluster identities are provided for us from the authors of the paper. A cell-type classifier would need to be run on the data to label the bar-codes (droplets) with their identifiers.

```
new.cluster.ids <- c("Memory CD4 T", "CD14+ Mono", "Naive CD4 T", "NK", "CD14+ Mono",
                     "Mouse", "B", "CD8 T", "CD16+ Mono", "T/Mono doublets", "NK", "CD34+",
                     "Multiplets", "Mouse", "Eryth", "Mk", "Mouse", "DC", "pDCs")
names(new.cluster.ids) <- levels(cbmc)
cbmc <- RenameIdents(cbmc, new.cluster.ids)
```

Clustering t-SNE plot

```
DimPlot(cbmc, label = TRUE, reduction = "tsne")
```

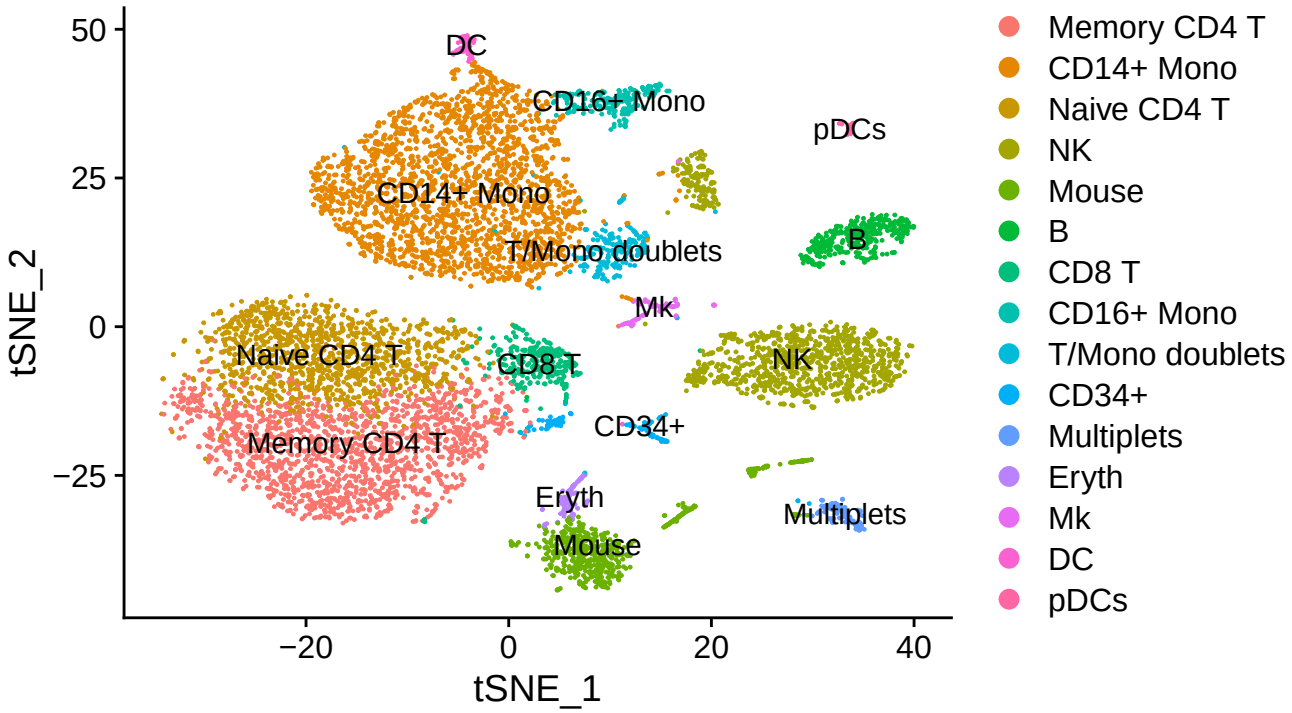


Figure 2: tSNE plot of clustered and classified scRNA-seq data. The 18 cluster ids provided from authors of paper.

Incorporate protein expression (antibody barcodes) to the Seurat object

```
cbmc[["ADT"]] <- CreateAssayObject(counts = cbmc_adt)
```

The above code adds a new assay called "ADT" to the Seurat object `cmbc`. We can confirm it's added with the following `GetAssayData()` command.

```
GetAssayData(cbmc, layer = "counts", assay = "ADT")[1:3,1:3]
```

```
3 x 3 sparse Matrix of class "dgCMatrix"
    CTGTTTACACCGCTAG CTCTACGGTGTGGCTC AGCAGCCAGGCTCATT
CD3               60                52                89
CD4               72                49               112
CD8               76                59                61
```

```
rownames(cbmc_adt)
```

```
 [1] "CD3"    "CD4"    "CD8"    "CD45RA" "CD56"   "CD16"   "CD10"   "CD11c"
 [9] "CD14"   "CD19"   "CD34"   "CCR5"   "CCR7"
```

Now we can repeat the pre-processing (normalization and scaling) steps that we typically run with RNA, but modifying the 'assay' argument.

(For CITE-seq data, the Broad does not recommend typical Log-normalization. Instead, they use a centered log-ratio (CLR) normalization, computed independently for each feature. This is a slightly improved procedure from the original publication.)

```
cbmc <- NormalizeData(cbmc, assay = "ADT", normalization.method = "CLR")
```

```
Normalizing across features
```

```
cbmc <- ScaleData(cbmc, assay = "ADT")
```

```
Centering and scaling data matrix
```

```
DefaultAssay(cbmc) <- "RNA"
```

Visualize protein levels on RNA clusters

```
FeaturePlot(cbmc, features = c("adt_CD3", "adt_CD11c", "adt_CD8", "adt_CD16",
                               "CD3E", "ITGAX", "CD8A", "FCGR3A"),
            min.cutoff = "q05", max.cutoff = "q95", ncol = 4)
```

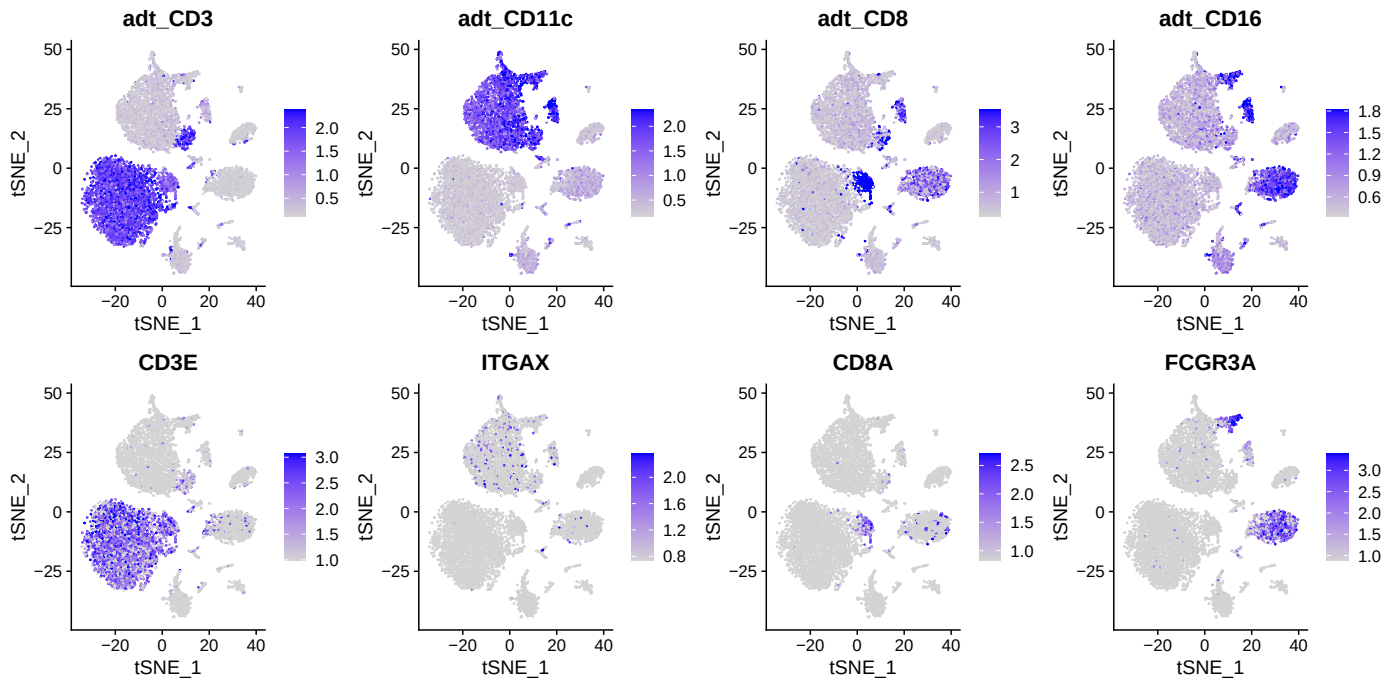


Figure 3: Set I: Juxtaposed heat map of cell-surface antibody-derived signal (top row) versus mRNA-seq (bottom row) of correspoding proteins and their mRNAs

```
# Compare gene and protein expression levels for the other 6 antibodies.
FeaturePlot(cbmc,
            features = c("adt_CD4", "adt_CD45RA", "adt_CD56",
```

```
                    "adt_CD14", "adt_CD19", "adt_CD34",
                    "CD4", "PTPRC", "NCAM1",
                    "CD14", "CD19", "CD34"),
        min.cutoff = "q05", max.cutoff = "q95", ncol = 6)
```

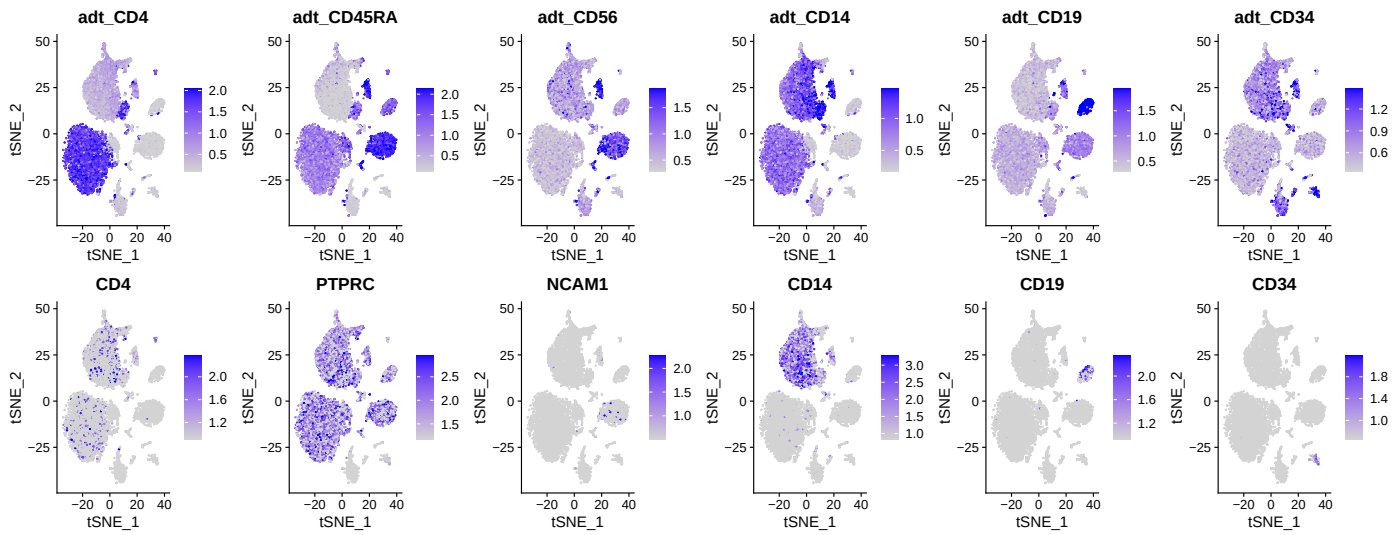


Figure 4: Set II: Juxtaposed heat map of cell-surface antibody-derived signal (top row) versus mRNA-seq (bottom row) of correspoding proteins and their mRNAs

As we can see from the above figures Figure 3 and Figure 4, in some cases, the same clusters / tSNE neighborhoods have high correspondence for the mRNA and the cell surface protein (*e.g.* CD3, CD4), but the penetrance (proportion of the cells) showing both antibody and mRNA feature may not be 100%; in other cases, some clusters expressing the cell-surface marker do not show much mRNA level (*e.g.* CD11, CD56, CD19, CD34), and vice-versa (*e.g.* CD45RA).

These side-by-side comparison illustrate the strength of the evidence added to the experiment by analyzing for both mRNA and protein.

```
RidgePlot(cbmc, features = c("adt_CD3", "adt_CD11c", "adt_CD8", "adt_CD16"), ncol = 2)
```

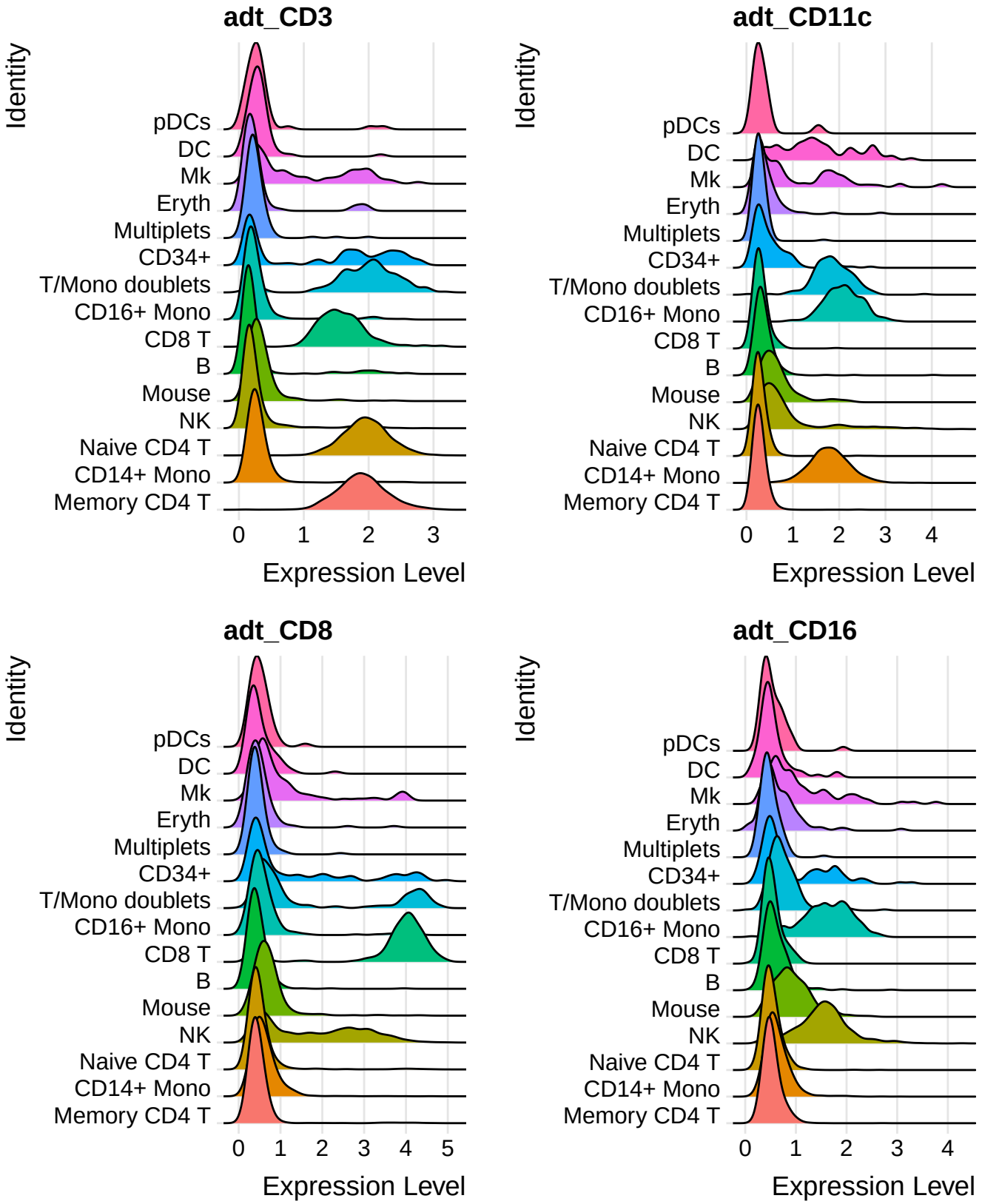


Figure 5: Stacked density plots showing this distribution of antibody-derived tag (ADT) signal in each scRNA-seq cell class for four cell-sirface markers: CD3, CD11c, CD8, and CD16.

Why may these differences occur?

One reason the "levels" between protein and mRNA could differ is that one of them is samples a very low count, and the normalization is concealing that fact.

```
norm_joy_plot_cd3 <- RidgePlot(cbmc, features = c("adt_CD3", "rna_CD3E"), ncol = 2) +
  patchwork::plot_annotation(subtitle = "Normalized expression levels")
count_joy_plot_cd3 <- RidgePlot(cbmc, features = c("adt_CD3", "rna_CD3E"), ncol = 2, layer = "counts
  patchwork::plot_annotation(subtitle = "Un-normalized NGS counts")

patchwork_plot_cd3 <- norm_joy_plot_cd3 / count_joy_plot_cd3

(patchwork_plot_cd3 <- patchwork_plot_cd3 +
  patchwork::plot_annotation(title = "Comparison of normalized and count data",
                             tag_levels = "a") &
    theme(axis.text = element_text(size = 5))
  )
```

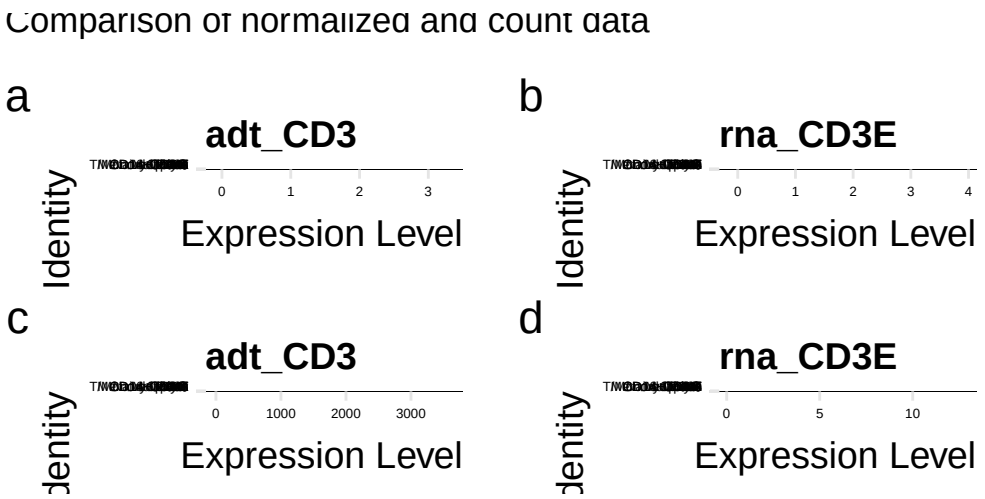


Figure 6: One gene, CD3, displayed as either antibody-derived tags (left side) or mRNA (right side), with x-axis in units of normalized expression (top half) or raw, un-normalized counts (bottom) for all cell classes.

As Figure 6 shows, the raw counts (which are UMI-collapsed) of the antibody-derived tags has a very large dynamic range with large values (around 1000 to 2000 counts) for positive cells, whereas the mRNA raw counts shows discretization close to 0, with the plurality of cells in each classification having 0 counts, followed by 1, then 2, then 3… counts. This shows that a low number of mRNA molecules either present in the cells (or incorporated into the single cell NGS library) is limiting the *sensitivity* of the scRNA-seq method at identifying key transcripts in the cells; ADTs have far less of this problem, where there is a large difference (from ~ 0 to about 1000 counts) between negative and positive cells, suggesting the antibody method is less susceptible to Poisson / binomial sampling noise and uncertainty.

```
FeatureScatter(cbmc, feature1 = "adt_CD19", feature2 = "adt_CD3", pt.size = 0.5, shuffle = TRUE)
```

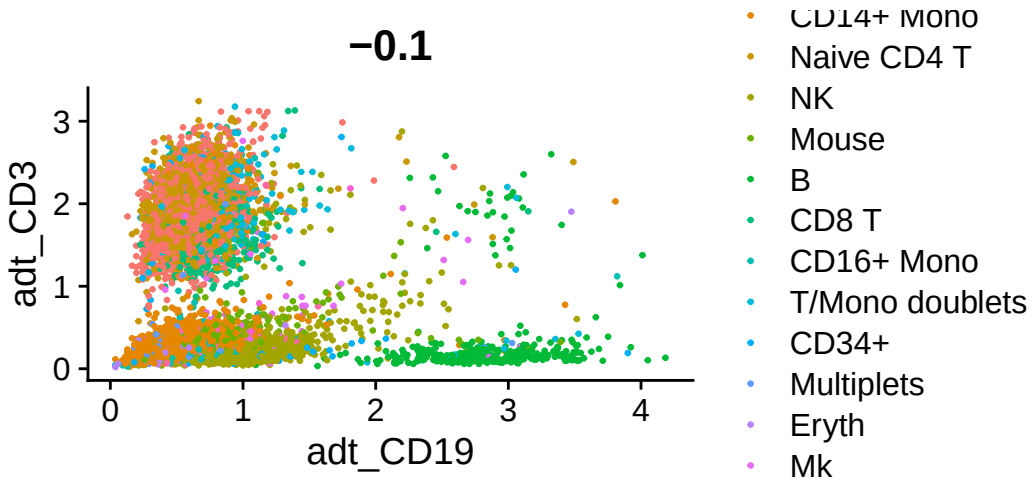


Figure 7: Scatter plot showing the correlation (in title) of CD3 vs CD19 antibody-derived tag (ADT) signal across ovserved cells.

Figure 7 shows that CD3 and CD19 are poorly correlated, with CD19 displaying high antibody signal in B cells, and CD3 showing higher signals in the T-cell classes (memory CD4+ T cells, naive CD4 T cells, and some T cell - monocyte doublets). This scatter plot of ADTs could function similarly to a flow plot, complete with gates.

```
patchwork_cor_plots <- (
  FeatureScatter(cbmc, feature1 = "adt_CD3", feature2 = "rna_CD3E", pt.size = 0.5, shuffle = TRUE, c
  FeatureScatter(cbmc, slot = "counts", feature1 = "adt_CD3", feature2 = "rna_CD3E",
                                        pt.size = 0.5, jitter = TRUE, combine = TRUE)
  ) | (
    FeatureScatter(cbmc, feature1 = "adt_CD4", feature2 = "rna_CD4", pt.size = 0.5, shuffle = TRUE,
    FeatureScatter(cbmc, slot = "counts", feature1 = "adt_CD4", feature2 = "rna_CD4", pt.size = 0.5,
    )
(patchwork_cor_plots <- patchwork_cor_plots +
    patchwork::plot_annotation(title = "Comparison of normalized and count data in scatter plot with
                               tag_levels = "a") +
    plot_layout(guides = "collect"))
```

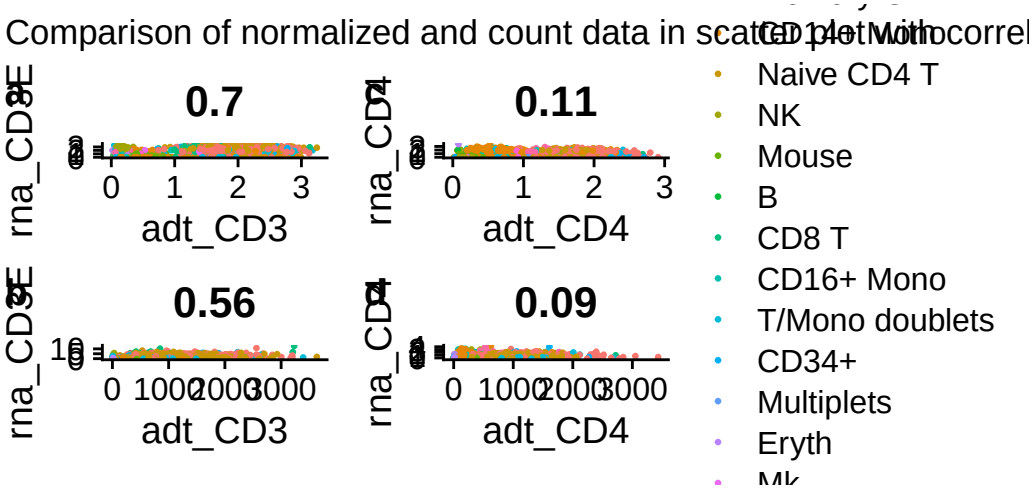


Figure 8: Scatter plot showing the correlation (in title) of ADT to mRNA expression for CD3 and CD4 using both normalized data and direct count data

```
tcells <- subset(cbmc, idents = c("Naive CD4 T", "Memory CD4 T", "CD8 T"))

(FeatureScatter(tcells, feature1 = "adt_CD4", feature2 = "adt_CD8") |
  FeatureScatter(tcells, feature1 = "adt_CD4", feature2 = "adt_CD8", slot = "counts") ) +
  patchwork::plot_annotation(title = "Comparison of normalized and count data",
                             tag_levels = "a") +
    plot_layout(guides = "collect")
```

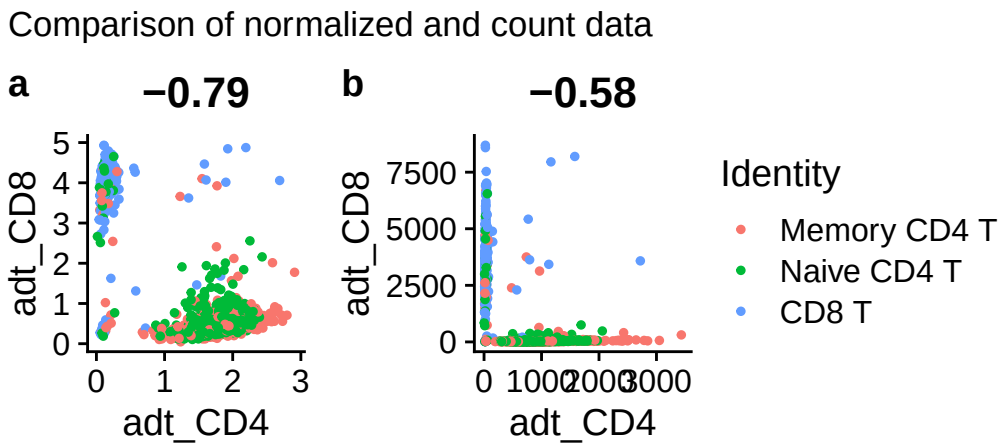


Figure 9: Scatter plot of antibody-derived tags (ADT) for CD4 and CD8 receptors, with normalized data in (a) and raw counts in (b)

```
ncol(subset(tcells, subset = adt_CD4 < 1 & adt_CD8 < 1)) / ncol(tcells)
```

```
[1] 0.009978833
```

In Figure 9, the pearson correlation of CD4 and CD8 antibody CITE-seq signal is -0.79, indicating these are signals are significantly anti-correlated, which is consistent with the immunology of T cells vs B cells. While the normalized data (a) look pretty separate / disjoint, the count data (b) are even more orthogonal, though there is less space separating the clusters in b than in a.

```
(FeatureScatter(tcells, feature1 = "rna_CD4", feature2 = "rna_CD8A", jitter = TRUE) |
   FeatureScatter(tcells, feature1 = "rna_CD4", feature2 = "rna_CD8A", slot = "counts", jitter = TRU
    patchwork::plot_annotation(title = "Comparison of normalized (a) and count data (b) of mRNA expr
                             tag_levels = "a") +
    plot_layout(guides = "collect")
```

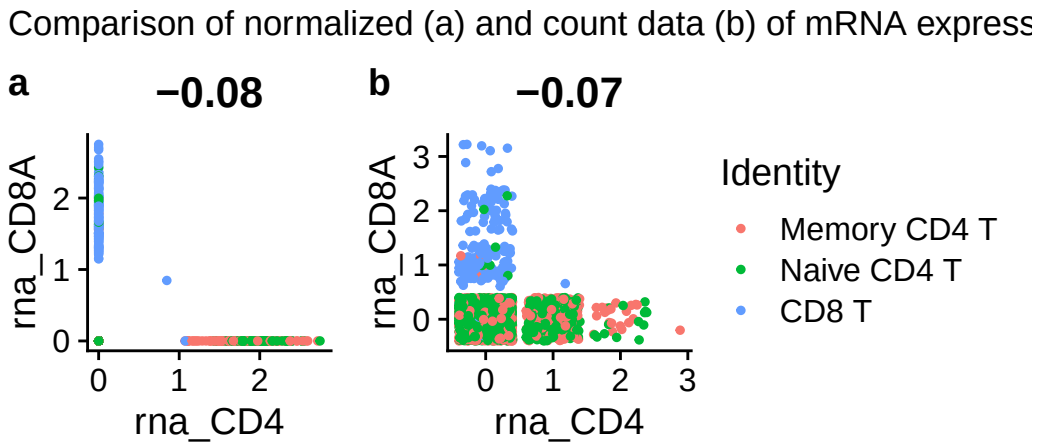


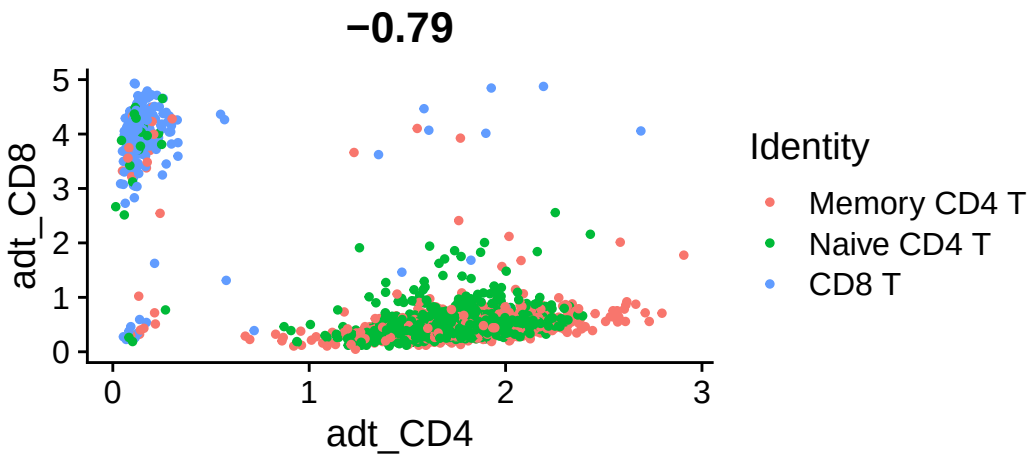
Figure 10: Scatter plot of mRNA expression for CD4 and CD8, with normalized data in (a) and raw counts in (b)

```
ncol(subset(tcells, subset = rna_CD4 == 0 & rna_CD8A == 0)) / ncol(tcells)
```

```
[1] 0.8285455
```

The above plot Figure 10 shows the weakness of attempting to classify immune cells by rarely-expressed mRNAs alone (despite the corresponding gene product being definitional to the cell class) when measured by RNA expression; As high as 83% of the T-cells are double-negative for CD4 and CD8.

```
DefaultAssay(tcells) <- "ADT"  # work with ADT count matrix
FeatureScatter(tcells, feature1 = "adt_CD4", feature2 = "adt_CD8")
```




```
ncol(subset(tcells, subset = adt_CD4 < 1 & adt_CD8 < 1)) / ncol(tcells)
```

```
[1] 0.009978833
```

However, for surface antigen detection in CITE-seq, only 0.997% are double negative for CD4 protein and CD8 protein.

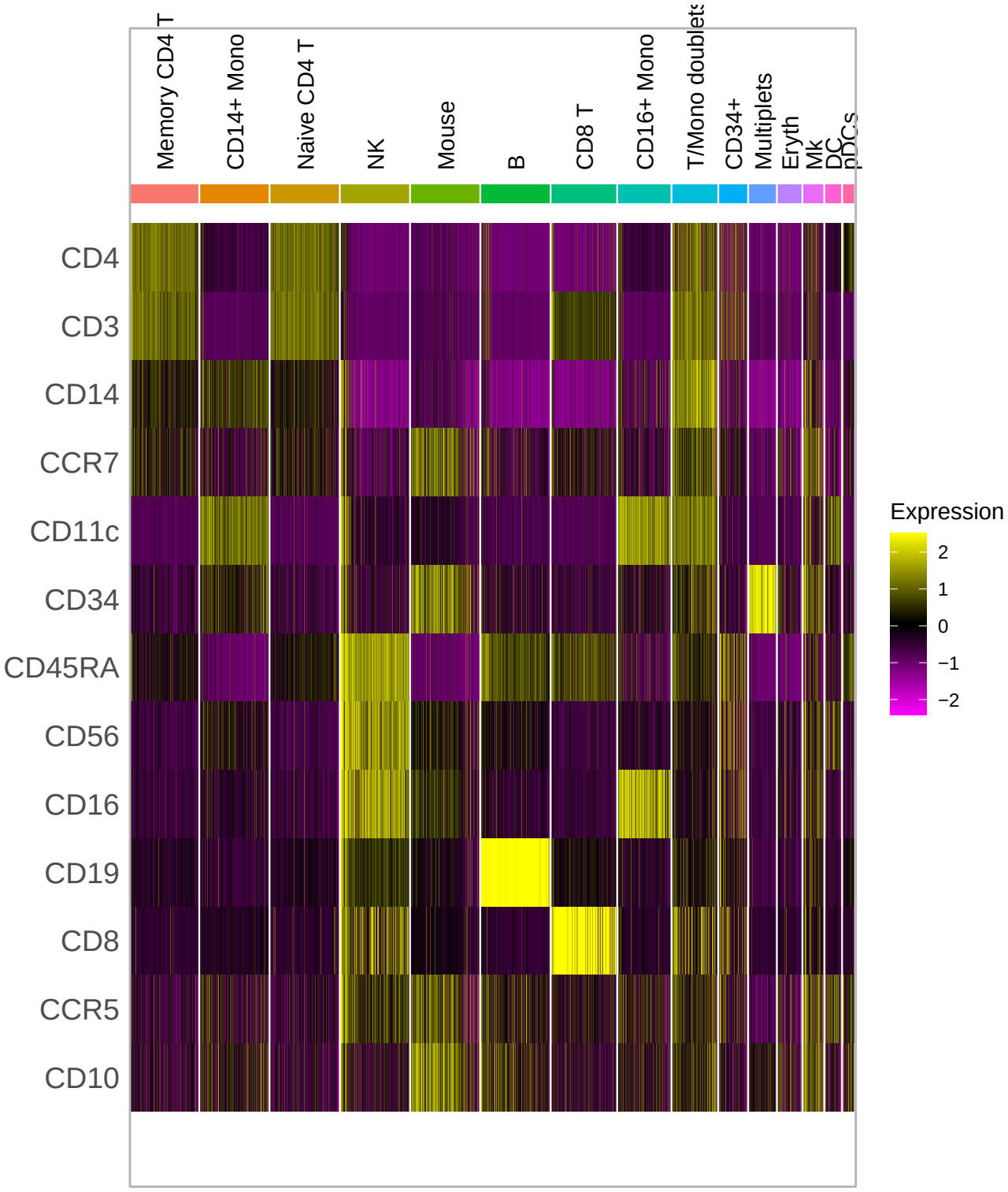
Differential protein levels between clusters

Here, I sample 300 cells per cluster to enhance visualization.

```
cbmc_subset <- subset(cbmc, downsample = 300)

# Find protein markers for all clusters, and draw a heatmap
adt_markers <- FindAllMarkers(cbmc_subset, assay = "ADT", only.pos = TRUE)
```

```
Calculating cluster Memory CD4 T

Calculating cluster CD14+ Mono

Calculating cluster Naive CD4 T

Calculating cluster NK

Calculating cluster Mouse

Calculating cluster B

Calculating cluster CD8 T

Calculating cluster CD16+ Mono

Calculating cluster T/Mono doublets

Calculating cluster CD34+

Calculating cluster Multiplets

Calculating cluster Eryth

Warning in FindMarkers.default(object = data.use, cells.1 = cells.1, cells.2 =
cells.2, : No features pass logfc.threshold threshold; returning empty
data.frame

Calculating cluster Mk

Calculating cluster DC

Calculating cluster pDCs
```

```
DoHeatmap(cbmc_subset,
          features = unique(adt_markers$gene),
          assay = "ADT", angle = 90, size = 4) +
  guides(color = "none") +
  theme(axis.text.y = element_text(size = 14),
        strip.text = element_text(size = 2))
```



The unknown cells co-express both myeloid and lymphoid markers (true at the RNA level as well). They are

likely cell clumps / multiplets that should be discarded.

Cluster directly on protein levels

Keeping human cells only:

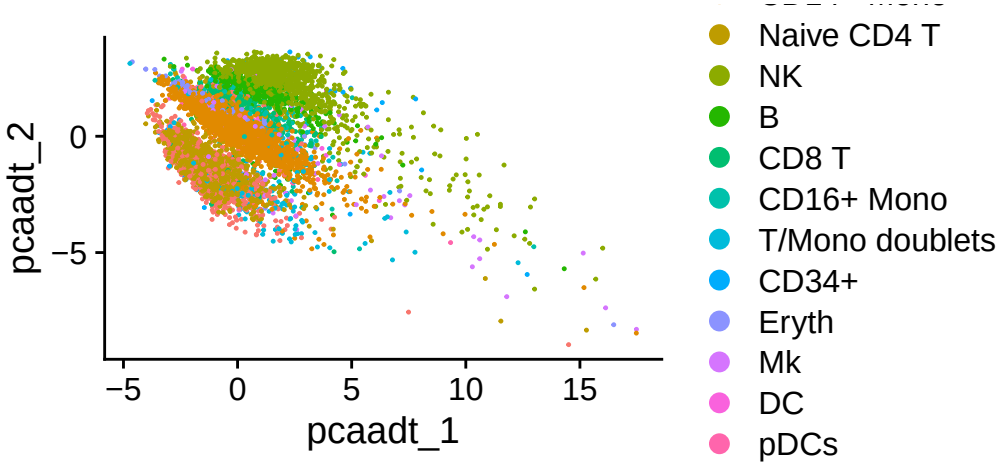
```
cbmc <- subset(cbmc, idents = c("Multiplets", "Mouse"), invert = TRUE)
```

```
DefaultAssay(cbmc) <- "ADT"
cbmc <-
  RunPCA(cbmc,
         features = rownames(cbmc),
         reduction.name = "pca_adt", reduction.key = "pcaadt_",
         verbose = FALSE)
```

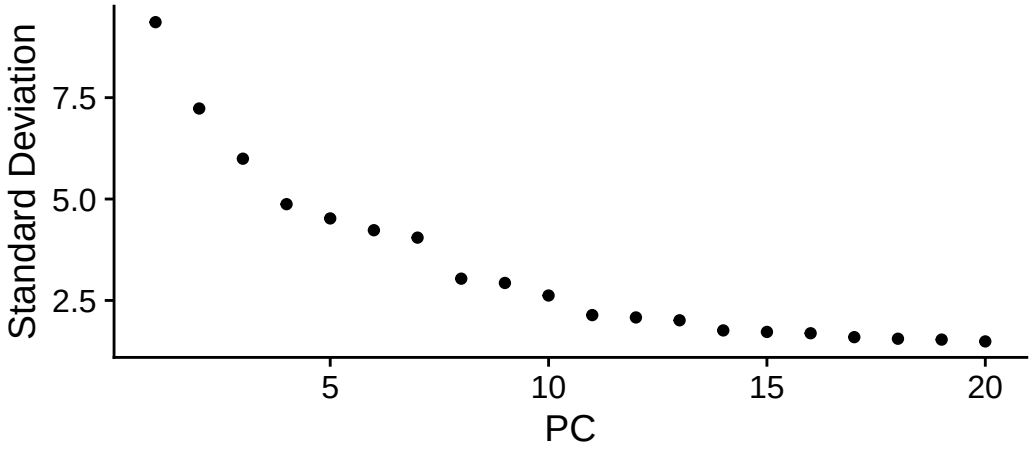
```
Warning in irlba(A = t(x = object), nv = npcs, ...): You're computing too large
a percentage of total singular values, use a standard svd instead.
```

I'm using `reduction.name` and `reduction.key`, because this is the second PCA being run on this multi-modal Seurat object, and I don't want the names to collide with the scRNA-seq PCA.

```
DimPlot(cbmc, reduction = "pca_adt")
```




```
ElbowPlot(cbmc)
```




```
adt_data <- GetAssayData(cbmc, layer = "data")
adt_dist <- dist(t(adt_data))

cbmc[["rnaClusterID"]] <- Idents(cbmc)
```

```
cbmc[["tsne_adt"]] <- RunTSNE(adt_dist, assay = "ADT", reduction.key = "adtTSNE_")
cbmc[["adt_snn"]] <- FindNeighbors(adt_dist)$snn
```

```
Building SNN based on a provided distance matrix

Computing SNN
```

```
cbmc <- FindClusters(cbmc, resolution = 0.2, graph.name = "adt_snn")
```

```
Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck

Number of nodes: 7891
Number of edges: 274115

Running Louvain algorithm...
Maximum modularity in 10 random starts: 0.9482
Number of communities: 12
Elapsed time: 0 seconds

Warning: Adding a command log without an assay associated with it
```

```
( clustering_table <- table(Idents(cbmc), cbmc$rnaClusterID) )
```

```
   Memory CD4 T CD14+ Mono Naive CD4 T   NK    B CD8 T CD16+ Mono
0          1416          0        1071    3    0    18          0
1             1       2198           0    5    0     0         36
2             6          0           3  887    2    10          0
3           273          0         194   26    0     6          0
```

```
  4              0          4           0    3  313     0          1
  5             23          0          18    4    1   247          0
  6              1         23           3  153    2     2          9
  7              3         59           4    0    0     0          9
  8              0          7           0    4    0     0        175
  9              3          4           0    1    0     1          0
  10             0          0           1    0    0     0          0
  11             1          0           2    0   24     0          0

     T/Mono doublets CD34+ Eryth   Mk   DC pDCs
  0                3    45     2    8    0    1
  1                3     0     3   24   55    1
  2                0    45     2    7    2    1
  3                2     7     4   16    1    1
  4                7     2     0    3    0    0
  5                0    10     0    2    0    0
  6               59     7     1    9    6    2
  7              118     2     0    1    0    0
  8                0     0     0    1    0    0
  9                2     5    92   17    5    1
  10               0     0     0    0    1   42
  11               3     0     0    0    0    0
```

```
new_cluster_ids <- levels(unique(cbmc$rnaClusterID))

names(new_cluster_ids) <- levels(cbmc)
```

```
levels(unique(cbmc$rnaClusterID))
```

```
 [1] "Memory CD4 T"    "CD14+ Mono"      "Naive CD4 T"     "NK"
 [5] "B"               "CD8 T"           "CD16+ Mono"      "T/Mono doublets"
 [9] "CD34+"           "Eryth"           "Mk"              "DC"
[13] "pDCs"
```

```
names(new_cluster_ids)
```

```
 [1] "0"  "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" NA
```

```
levels(cbmc)
```

```
 [1] "0"  "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11"
```

```
cbmc <- RenameIdents(cbmc, new_cluster_ids)
```

```
Warning: Cannot find identity NA
```

```
tsne_rnaClusters <- DimPlot(cbmc, reduction = "tsne_adt", group.by = "rnaClusterID", pt.size = 0.5)
  NoLegend() +
  ggtitle("Classification based on scRNA-seq") +
  theme(plot.title = element_text(size = 12, hjust = 0.5))

tsne_rnaClusters <- LabelClusters(plot = tsne_rnaClusters, id = "rnaClusterID", size = 4)

tsne_adtClusters <- DimPlot(cbmc, reduction = "tsne_adt", pt.size = 0.5) +
  NoLegend() +
  ggtitle("Classification based on ADT signal") +
  theme(plot.title = element_text(size = 12, hjust = 0.5))

tsne_adtClusters <- LabelClusters(plot = tsne_adtClusters, id = "ident", size = 4)

# Note: for this comparison, both the RNA and protein clustering are visualized on a tSNE
# generated using the ADT distance matrix.
( tsne_rna_adtClusters <- patchwork::wrap_plots(list(tsne_rnaClusters, tsne_adtClusters), ncol = 2)
  plot_annotation(tag_levels = 'a') )
```

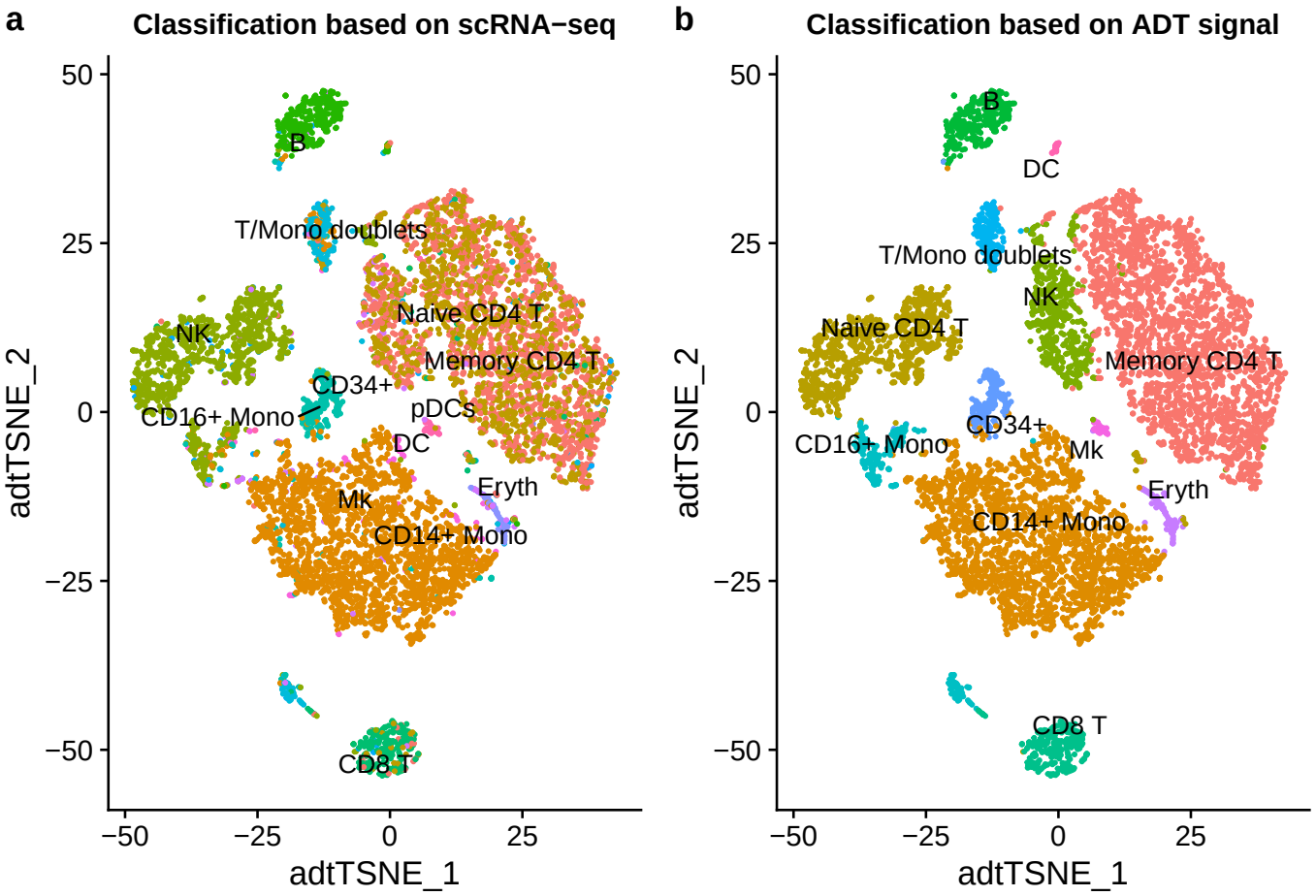


Figure 11: Juxtaposed tSNE plots of ADT (antibody) signal, colored and labeled by the data source indicated in the title (classification based on scRNA-seq in a; classification based on ADT signal in b).

The tSNE clustering in Figure 11 above is based on the distance matrix ADT (antibody) signal, whereas the

coloring and cluster labels are, on the scRNA-seq data.

Overall, the ADT-driven clustering yields similar results. The compare / contrast conclusions are:

- ADT clustering improves CD4/CD8 T cell group distinction, based on the robust, high-count ADT data for CD4, CD8, CD14, and CD45RA.
- However, ADT-based clustering is worse for the Mk/Ery/DC cell-surface markers, and scRNA-seq distinguishes these populations better.
- Some of the clusters are likely doublets, which have low confidence classifier calls in both the scRNA-seq and ADT methods. (However, scRNA-seq could have more features for more confident doublet identification and removal.)