scRNA-seq Workshop Part 2

Applied Bioinformatics for Genomics

JENNIFER A. FOLTZ, PHD

ASSISTANT PROFESSOR, SECTION OF COMPUTATIONAL BIOLOGY

JENNIFER.A.FOLTZ@WUSTL.EDU

With content contributions from Dr. Petti

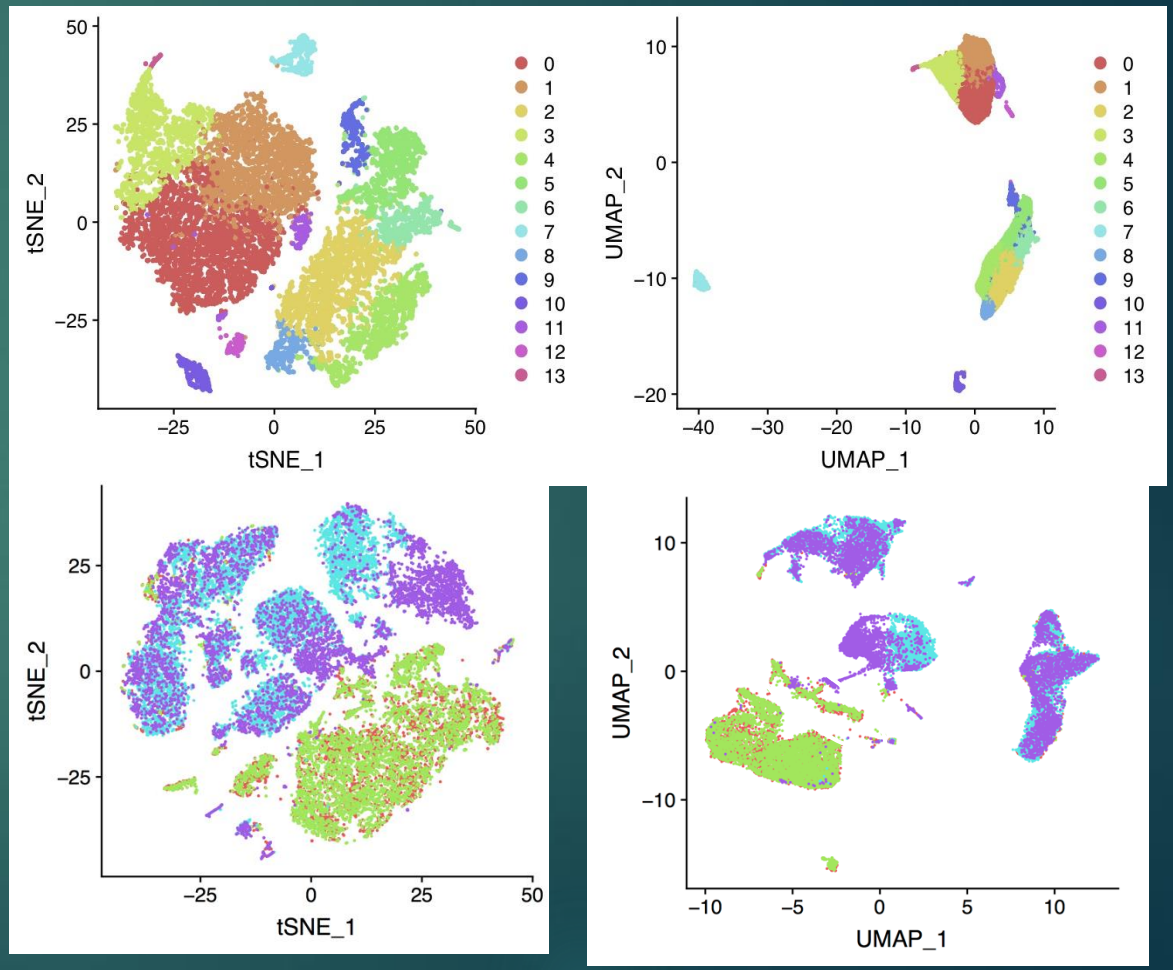
Plotting using t-SNE/UMAP

t-SNE = *t*-distributed Stochastic Neighbor Embedding
UMAP = Uniform Manifold Approximation and Projection

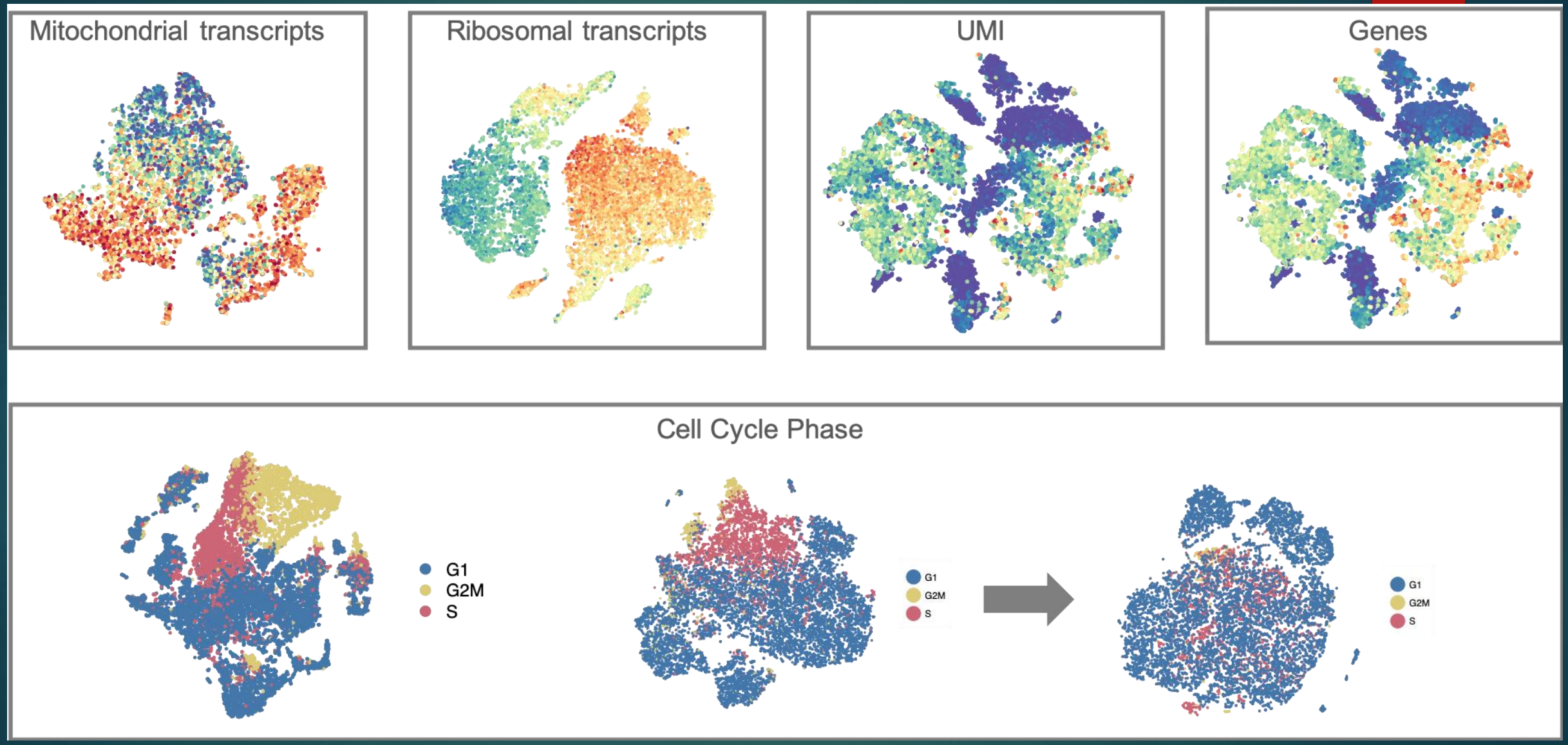
Goal: Embed high-dimensional data in low-dimensional space

End product: 2D plot where cells are positioned near each other if they have similar gene expression profiles. “Units” are relative and data-dependent.

- Expression “distances” between points (ie cells) in high-dimensional space are modeled using a gaussian distribution.
- Operates in “PCA space”
- “clusters” are not clusters. This is not clustering.
- t-SNE preserves local structure only.
- UMAP preserves local AND global structure.
- Implication: In UMAP, distances between points *and* clusters are ***more*** interpretable in terms of expression distances/similarity.

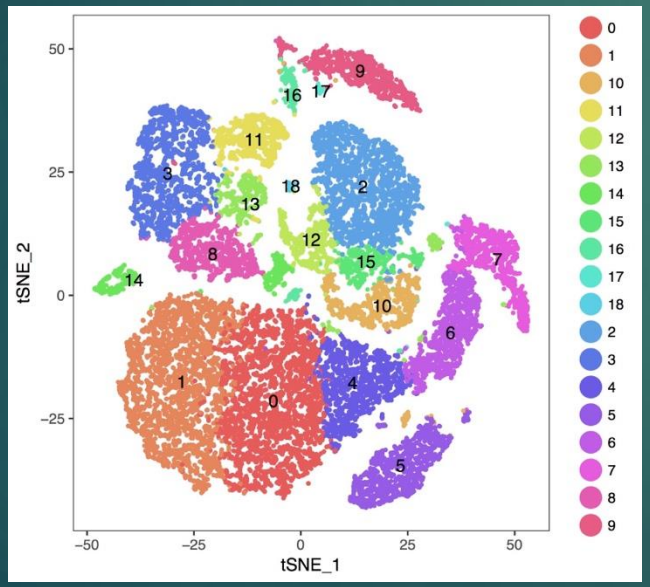


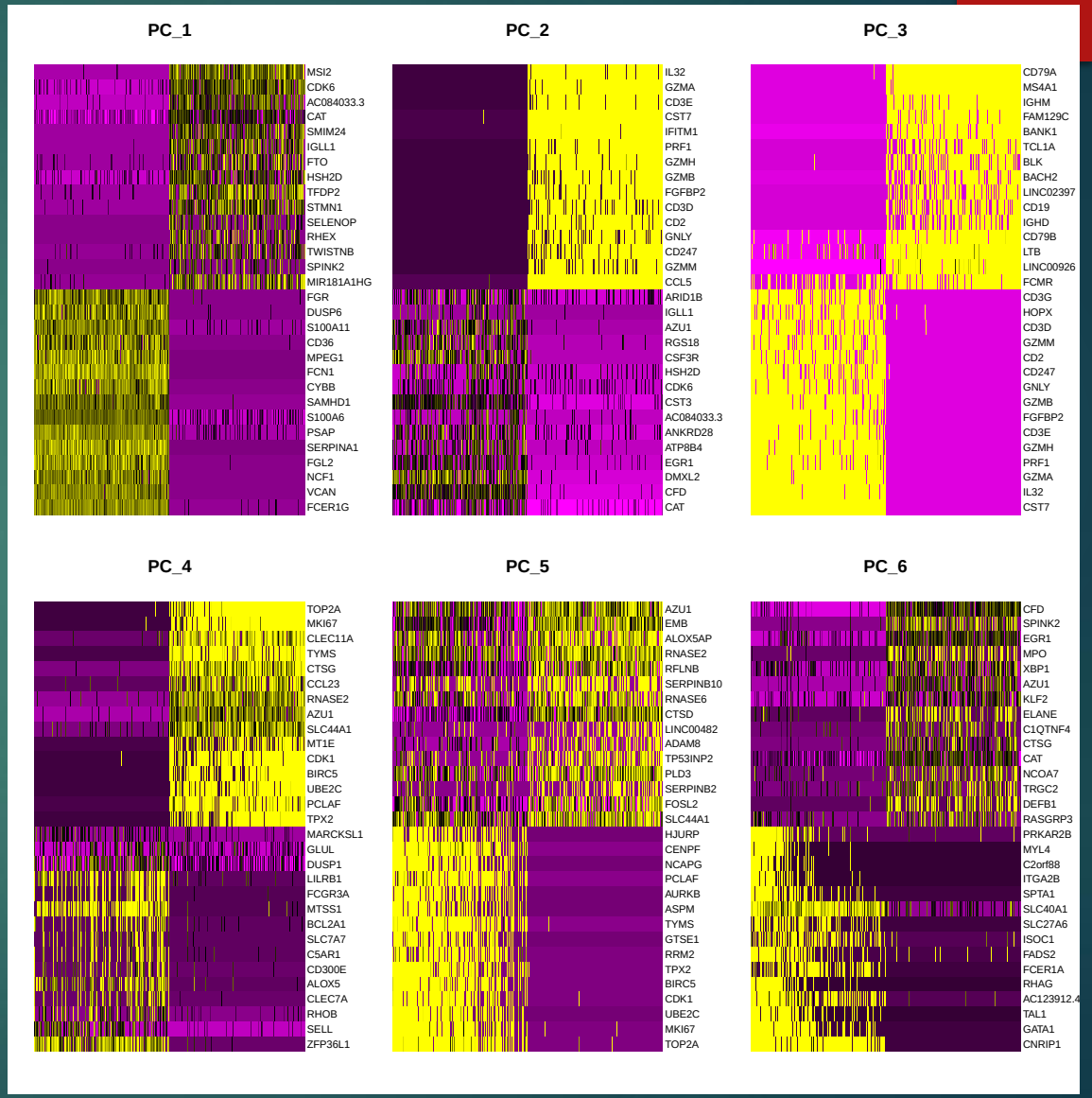
Interpreting the t-SNE/UMAP, Part I: Potentially misleading sources of variation



Interpreting the t-SNE/UMAP, Part II: Systematic analysis of variation

- What is driving the t-SNE/UMAP layout?
- Find genes that vary:
  - Principal components
  - Individual cluster-specific genes
- Examine across clusters/t-SNE/UMAP
- **http://bioconductor.org/books/3.15/OSCA.basic/dimensionality-reduction.html**

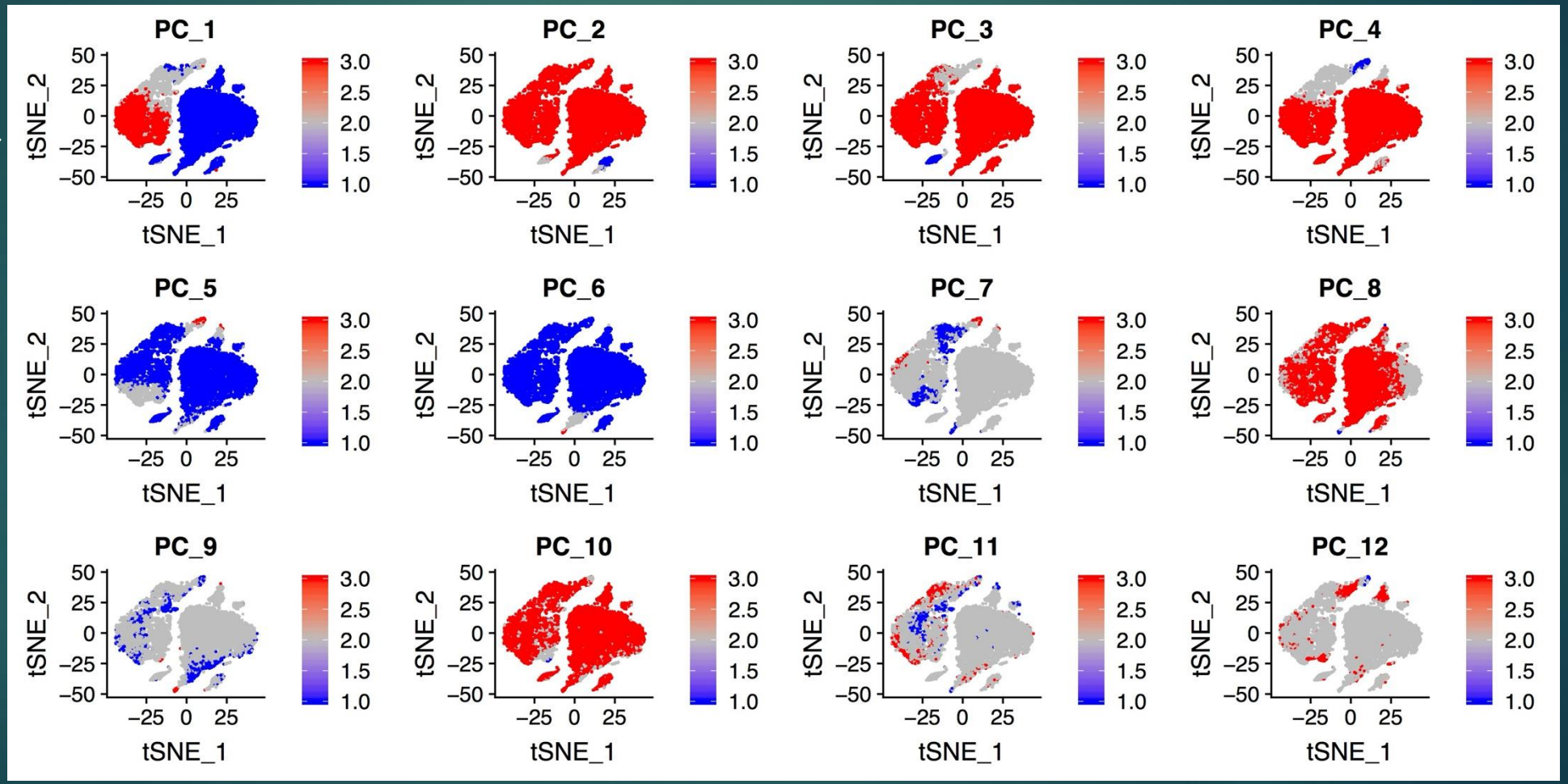




Part II, cont'd: Visualizing sources of variation

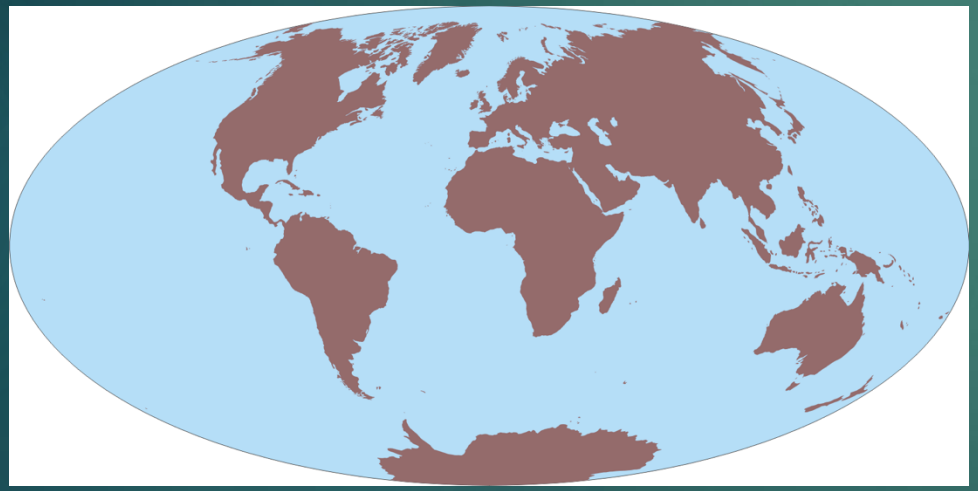
Small but distinct cell clusters may contribute heavily to the layout (may need to be removed)

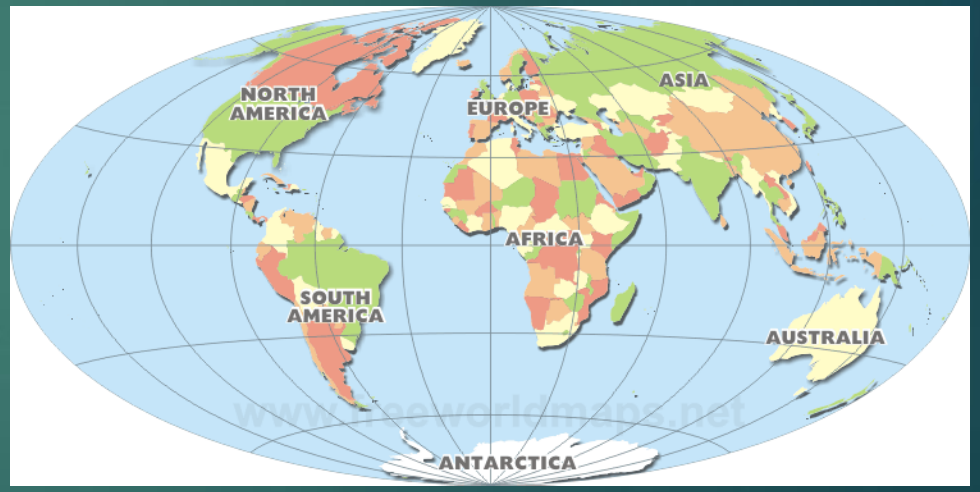
PC #1 captures the biggest source of variation



2-D layout vs. Clustering

- tSNE and UMAP reflects natural organization of data by approximating high-dimensional relationships in low-dimensional space
- Clustering imposes structure by assigning cells to non-overlapping groups based on relative expression similarity





Clustering 101

- Clustering largely helps with being able to run stats
  - Very few DE packages out there that can work without calling clusters
    - Single cell haystack
  - Multiple ways to go about these:
    - K clustering- you decide how many clusters (k) you want the algorithm to decide
    - You decide everything: e.g. lasso clustering, circling cells you want to group together
    - Threshold clustering: by expression of a gene or subset of genes:
      - E.g. subset(object, CD3D <0 & NCAM-1 > 0) for NK cells (just an example!)
      - Challenging due to the noise of the data- do you subset on raw counts, normalized data, etc? What about cells that should be positive but aren't?
    - "unsupervised" graph-based clustering- can be Louvain, leiden, smart local moving-different methods called under the hood to get to the same goal
      - Users specifies a "resolution"- higher = more clusters, lower = less clusters: usually higher needed with more cells and/or more heterogeneity; and the inverse is also true

Coming to terms with your clusters

- ”unsupervised” graph-based clustering- can be Louvain, leiden, different methods called under the hood to get to the same goal
  - Users specifies a “resolution”- higher = more clusters, lower = less clusters: usually higher needed with more cells and/or more heterogeneity; and the inverse is also true
  - You can always have more or less clusters:
  - Justification:

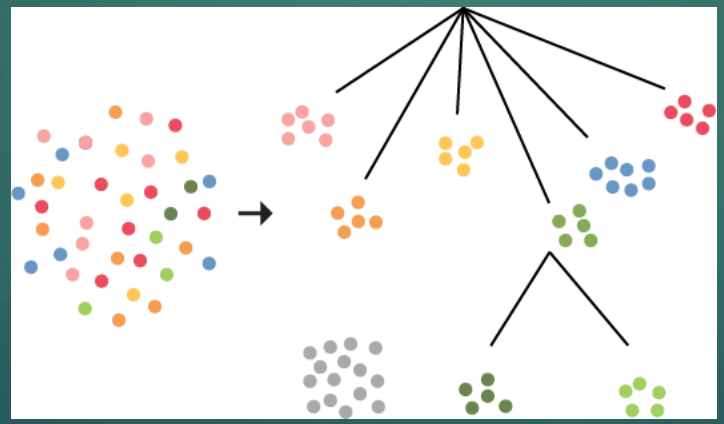
High-Dimensional Single-Cell Analysis Identifies Organ-Specific Signatures and Conserved NK Cell Subsets in Humans and Mice

Abstract

Natural killer (NK) cells are innate lymphoid cells (ILCs) involved in antimicrobial and antitumoral responses. Several NK cell subsets have been reported in humans and mice, but their heterogeneity across organs and species remains poorly characterized. We assessed the diversity of human and mouse NK cells by single-cell RNA sequencing on thousands of individual cells isolated from spleen and blood. Unbiased transcriptional clustering revealed two distinct signatures differentiating between splenic and blood NK cells. This analysis at single-cell resolution identified three subpopulations in mouse spleen and four in human spleen, and two subsets each in mouse and human blood. A comparison of transcriptomic profiles within and between species highlighted the similarity of the two major subsets, NK1 and NK2, across organs and species. This unbiased approach provides insight into the biology of NK cells and establishes a rationale for the translation of mouse studies to human physiology and disease.

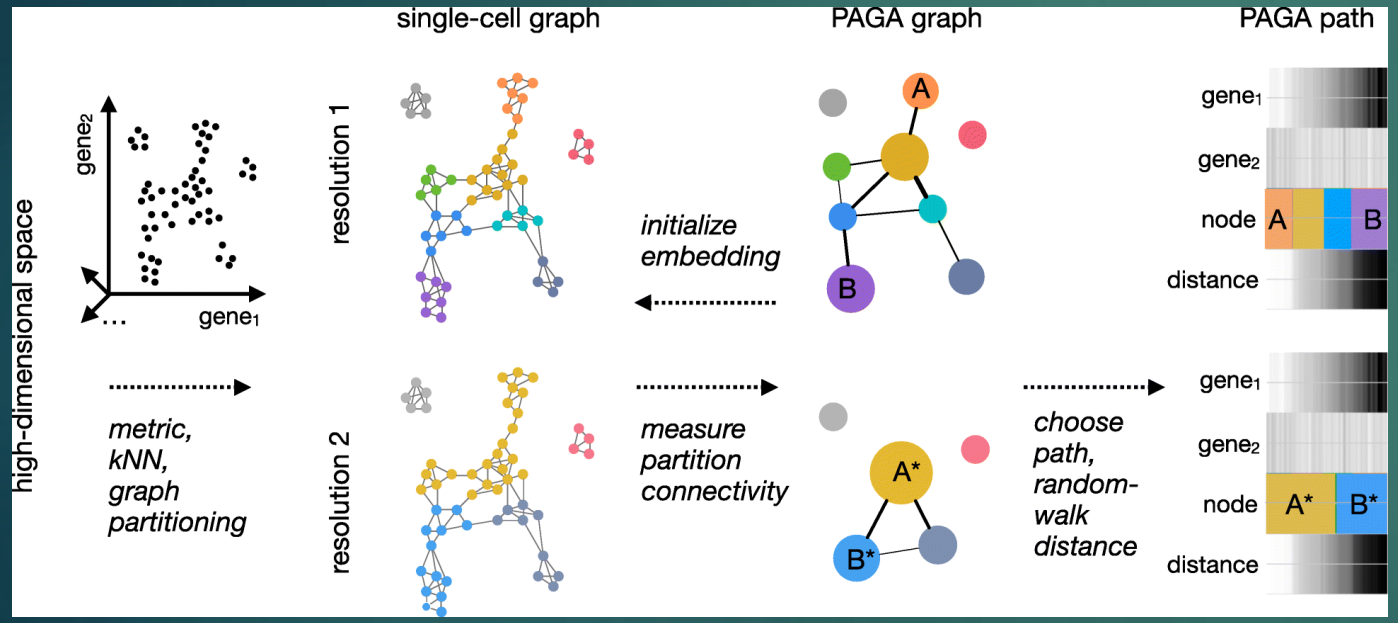
Cluster Assignments Continued

- Variety of packages to assign cells to a cell type for you (e.g. a reference)
  - Ex. SingleR, Garnett, Seurat TransferData
  - Some care about your clustering while others are cluster agnostic
  - Cluster ”extend” decreases noise
  - Cluster agnostic pro is no need to cluster prior to using the reference



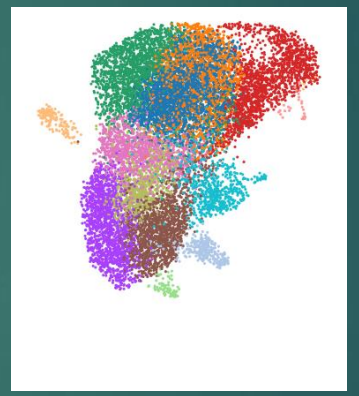
garnett

Partition-Associated Graph Abstraction (PAGA)

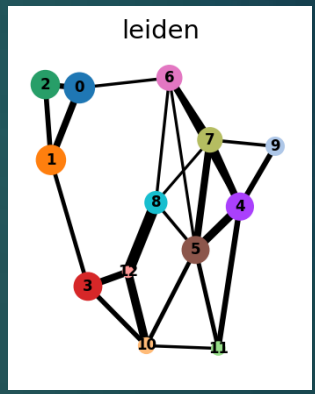


Example:

leiden



PAGA



Available in R through: https://github.com/zhanghao-njmu/SCP

Wolf, F.A., Hamey, F.K., Plass, M. *et al.* PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol* **20**, 59 (2019). https://doi.org/10.1186/s13059-019-1663-x

Characterizing cells using differential gene expression

Other Differential Expression Tests to Consider:

- AUC/ROC
- DeSeq2
- MAST/Logistic Regression
- Scran pairwiseWilcox() by blocking
- Wilcoxon rank-sum test
- Pseudo-bulking

Things to Consider:

- Pairwise differential gene expression
- What fraction of cells in each sample express a given gene?
- Of the cells in each sample that express a given gene, does the mean expression in those cells differ?
- Does the distribution of cell types differ between samples?
- Do the samples exhibit cell-type-specific differential gene expression?
- Input into gene ontology algorithms (e.g. do you need a universe/background?)

Characterizing cells using differential gene expression

- Data has zero-inflated negative binomial distribution (lots of zeros, overdispersed) so can't use bulk methods
  - Default in Seurat: Wilcoxon rank-sum test
  - Nonparametric version of t-test
  - For two clusters (A and B), and one gene, rank each cell in each cluster according to expression
  - Determine whether sum-of-ranks for cluster A is significantly different than sum-of-ranks for cluster B
  - Clear explanation of Wilcoxon rank-sum test: http://statweb.stanford.edu/~susan/courses/s141/hononpara.pdf
  - ***Statistical power is connected to the number of cells in a group, and can be driven by outliers***
  - Numerous other tests in Seurat and other packages

    Fold-changes are lower due to noise and low- detection

    Generally accepted to set a minimum detection rate to decrease noise & power

        Most commonly 25% of cells must express the gene for it be detected but can do lower or higher depending on question

Analyzing Multiple Samples

- Merging or batch-correction?
- Avoid batch correction unless absolutely necessary
  - Correct for different technologies (e.g. 3’ and 5’)
  - Correct for different batches
  - Discover conserved biology by finding corresponding cells across different data sets
- Cellranger does faux batch-correction (corrected values are discarded), but batch-corrected tSNE can be visualized in the loupe browser.

What Not to Do

- CRITICAL: Experimental Design Considerations
  - Submission Date
  - What is your hypothesis?
  - How do you envision doing differential expression downstream?
- DO:
  - Treat experimental groups as similar as possible:
    - E.g. if you need to sort one group, sort the other group even if it is technically not needed
  - Include control cell populations that can be used to assess how well a technique is working
- DON'T:
  - Submit control & experimental groups on separate days, technologies, etc.
  - Batch-correct on your experimental question
    - E.g. batch-correct on drug treatment and expect to find clusters that are different with drug treatment

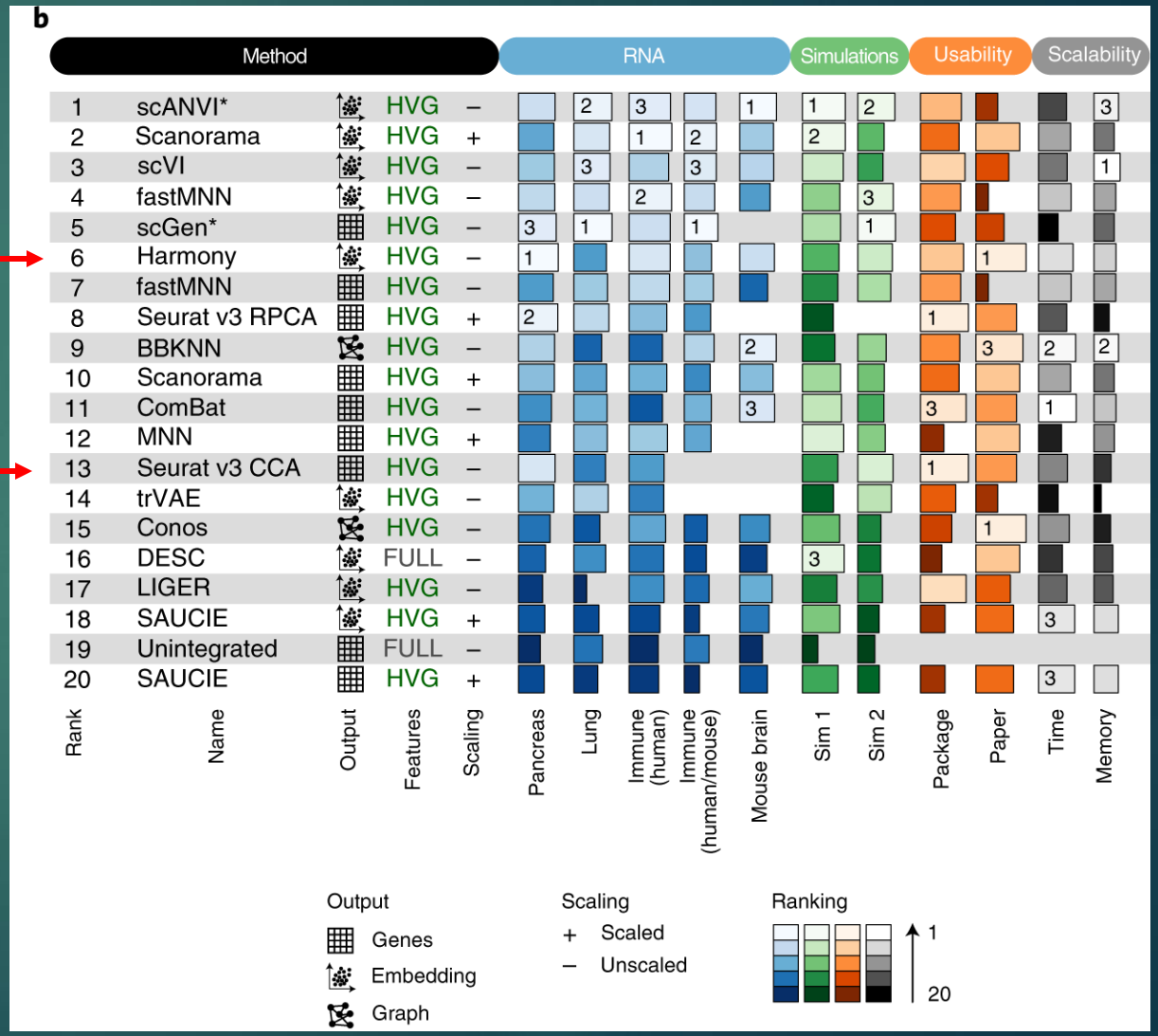
Analyzing Multiple Samples

Luecken, et al., 2022

Table 1 Description of the 14 batch-effect correction methods

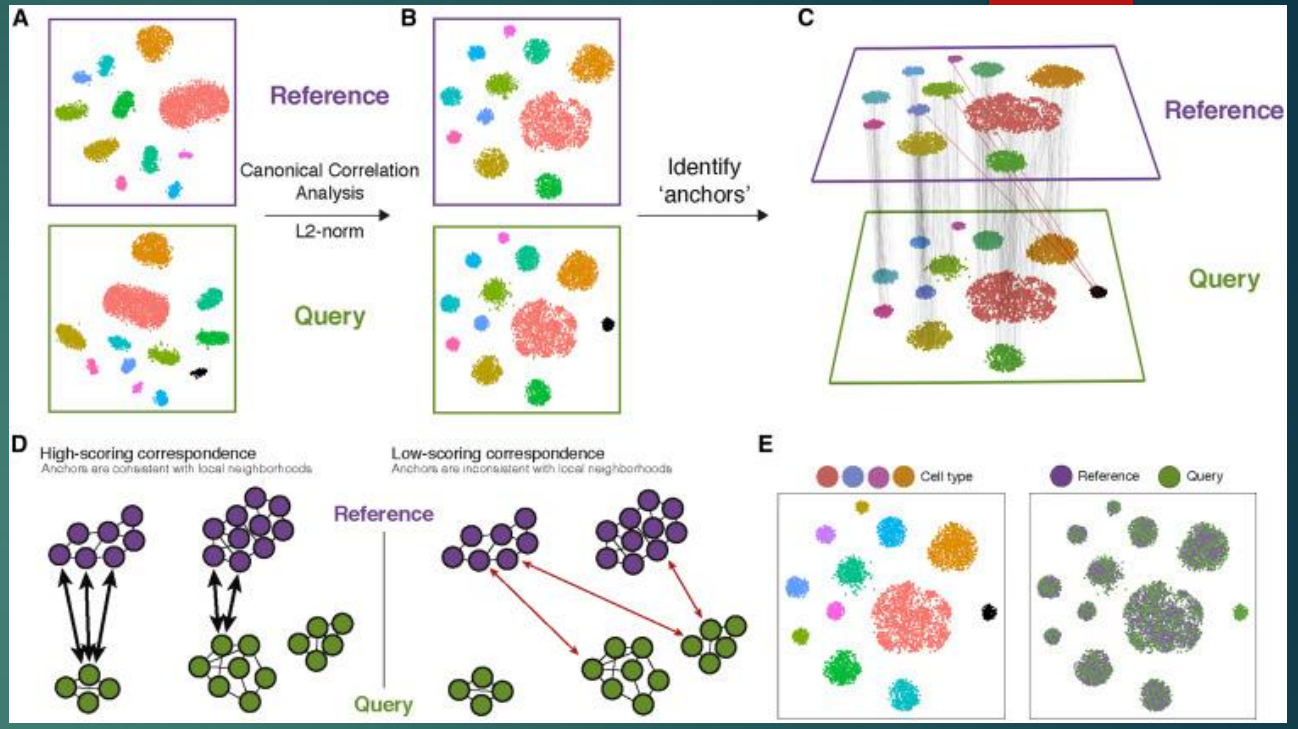
From: A benchmark of batch-effect correction methods for single-cell RNA sequencing data

Tools	Programming language	Batch-effect-corrected output	Methods	Reference package version
Seurat 2 (CCA, MultiCCA)	R	Normalized canonical components (CCs)	Canonical correlation analysis and dynamic time warping	Butler et al. [4], Seurat package version 2.3.4
Seurat 3 (Integration)	R	Normalized gene expression matrix	Canonical correlation analysis and mutual nearest neighbors-anchors	Stuart et al. [12], Seurat package version 3.0.1
Harmony	R	Normalized feature reduction vectors (Harmony)	Iterative clustering in dimensionally reduced space	Korsunsky et al. [13], Harmony version 0.99.9
MNN Correct	R	Normalized gene expression matrix	Mutual nearest neighbor in gene expression space	Haghverdi et al. [5], Scran package version 1.12.0
fastMNN	R	Normalized principal components	Mutual nearest neighbor in dimensionally reduced space	Haghverdi et al. [5], Lun ATL [7], Scran package version 1.12.0
ComBat	R	Normalized gene expression matrix	Adjusts for known batches using an empirical Bayesian framework	Johnson et al. [1]
limma	R	Normalized gene expression matrix	Linear model/empirical Bayes model	Smyth et al. [2], limma version 3.38.3
scGen	Python	Normalized gene expression matrix	Variational auto-encoders neural network model and latent space	Lotfollahi et al. [16], 2019, scGen version 1.0.0
Scanorama	Python/R	Normalized gene expression matrix	Mutual nearest neighbor and panoramic stitching	Hie et al. [9], Scanorama version 1.4.
MND-ResNet	Python	Normalized principal components	Residual neural network for calibration	Shaham et al. [15] updated code to Python 3
ZINB-WaVE	R	Normalized feature reduction vectors (ZINB-WaVE)/normalized gene expression matrix	Zero-inflated negative binomial model, extension of RUV model	Risso et al. [6], ZINB-WaVE version 1.6.0
scMerge	R	Normalized gene expression matrix	Stably expressed genes (scSEGs) and RUVIII model	Lin et al. [18], scMerge version 1.1.3
LIGER	R	Normalized feature reduction vectors (LIGER)	Integrative non-negative matrix factorization (iNMF) and joint clustering + quantile alignment	Welch et al. [14], liger version 1.0
BBKNN	Python/R	Connectivity graph and normalized dimension reduction vectors (UMAP)	Batch balanced *k*-nearest neighbors	Polański et al. [10], bioRxiv. BBKNN version 1.3.2



Option 1: Integration

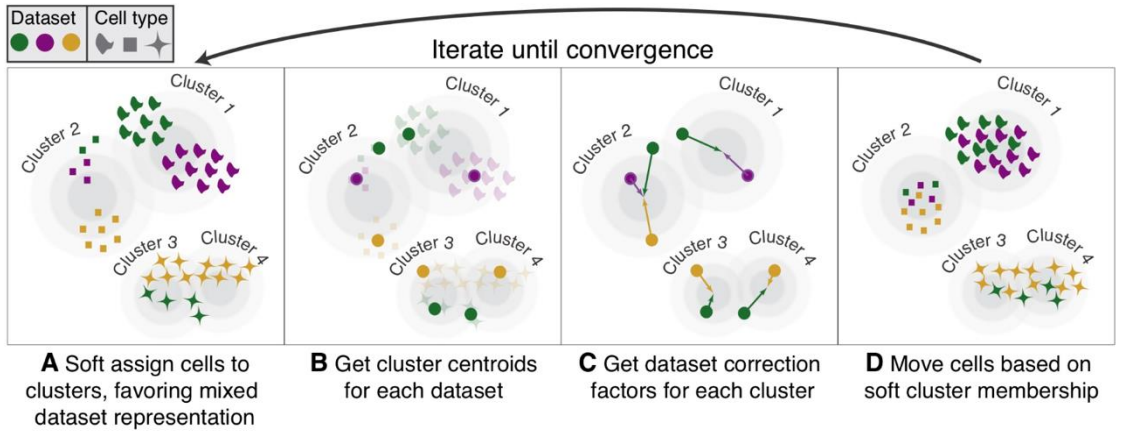
```
anchors <- FindIntegrationAnchors(object.list = scrna.list, di
ms = 1:30) # find anchors

scrna.int <- IntegrateData(anchorset = anchors, dims = 1:30
) # Integrate data
```



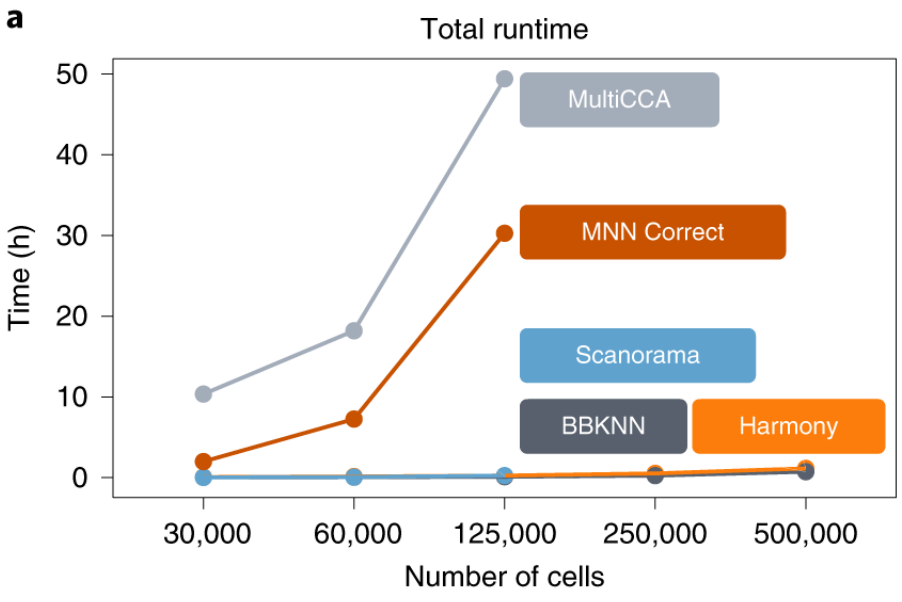
Stuart et al. 2019

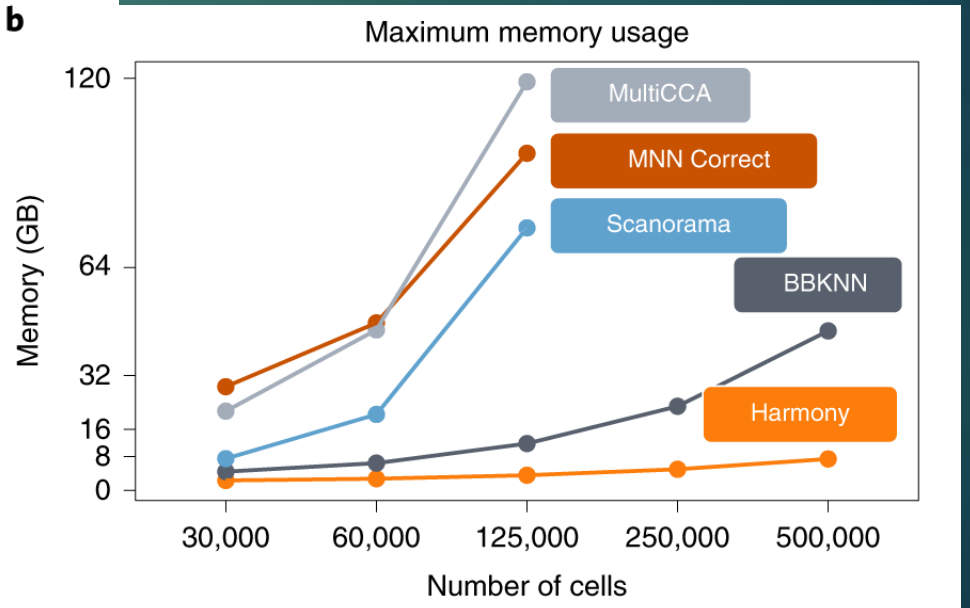
- Integrated values not intended for use with differential expression calculations.
  - We recommend running your differential expression tests on the “unintegrated” data. By default this is stored in the “RNA” Assay. There are several reasons for this.
  - The integration procedure inherently introduces dependencies between data points. This violates the assumptions of the statistical tests used for differential expression.
- TransferData function uses data integration to classify cells based on a reference data set.

Option 2: Harmony Batch Correction



- Ability to batch correct on multiple classes
- Minimal increase in the saved object file size

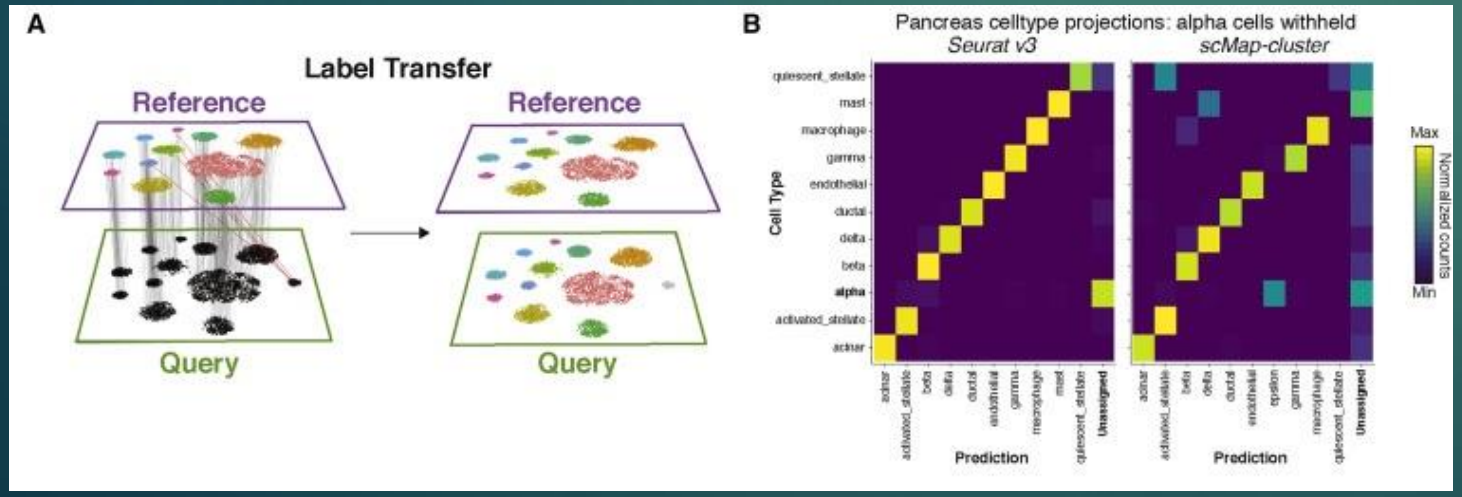




Korsunsky et al., 2019

Label Harmonization/Transfer Approaches

- In lieu of batch correction, can “integrate” data by aligning the sample annotations
  - e.g. look for gene changing between the same cell populations across samples
- Pros: No batch correction required, smaller datasets easier to work with computationally
- Cons: No increase in power by combining samples, annotations may not be 100% consistent

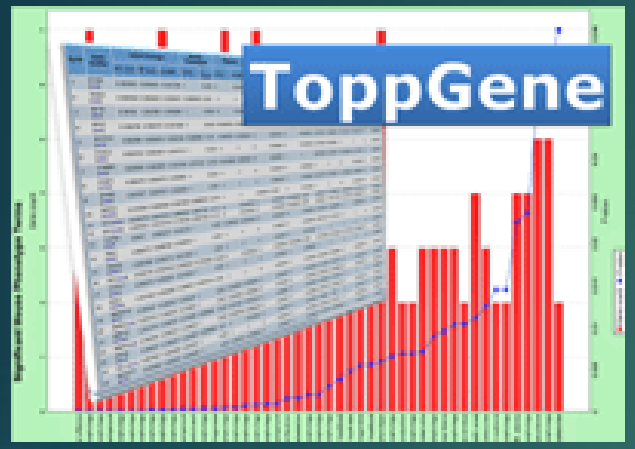


Stuart et al. 2019

Re-clustering of Data

- Need to rerun FindVariableFeatures (in order to increase chance of finding additional heterogeneity or cell populations)
- Rerun ScaleData
- Rerun dimensionality reduction tests, and batch correction (if using Harmony)

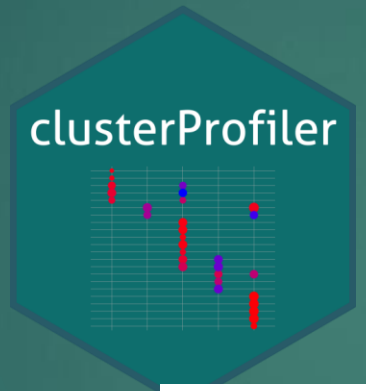
Pathway & Gene Set Analysis

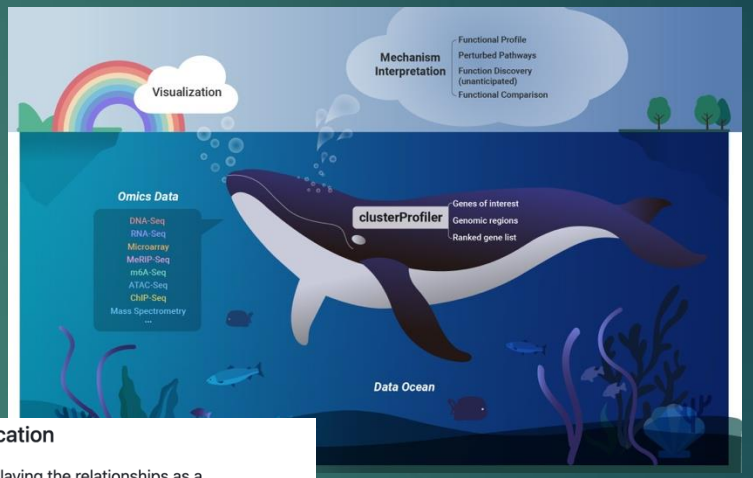


https://toppgene.cchmc.org/enrichment.jsp

ToppFun

Clusterprofiler
https://yulab-smu.top/biomedical-knowledge-mining-book/





15.4 **Heatmap-like functional classification**

The `heatplot` is similar to `cnetplot`, while displaying the relationships as a heatmap. The gene-concept network may become too complicated if user want to show a large number significant terms. The `heatplot` can simplify the result and more easy to identify expression patterns.

```
p1 <- heatplot(edox, showCategory=5)
p2 <- heatplot(edox, foldChange=geneList, showCategory=5)
cowplot::plot_grid(p1, p2, ncol=1, labels=LETTERS[1:2])
```

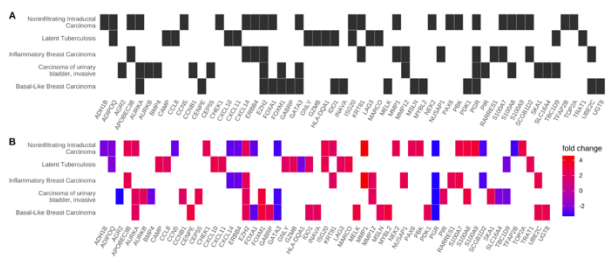
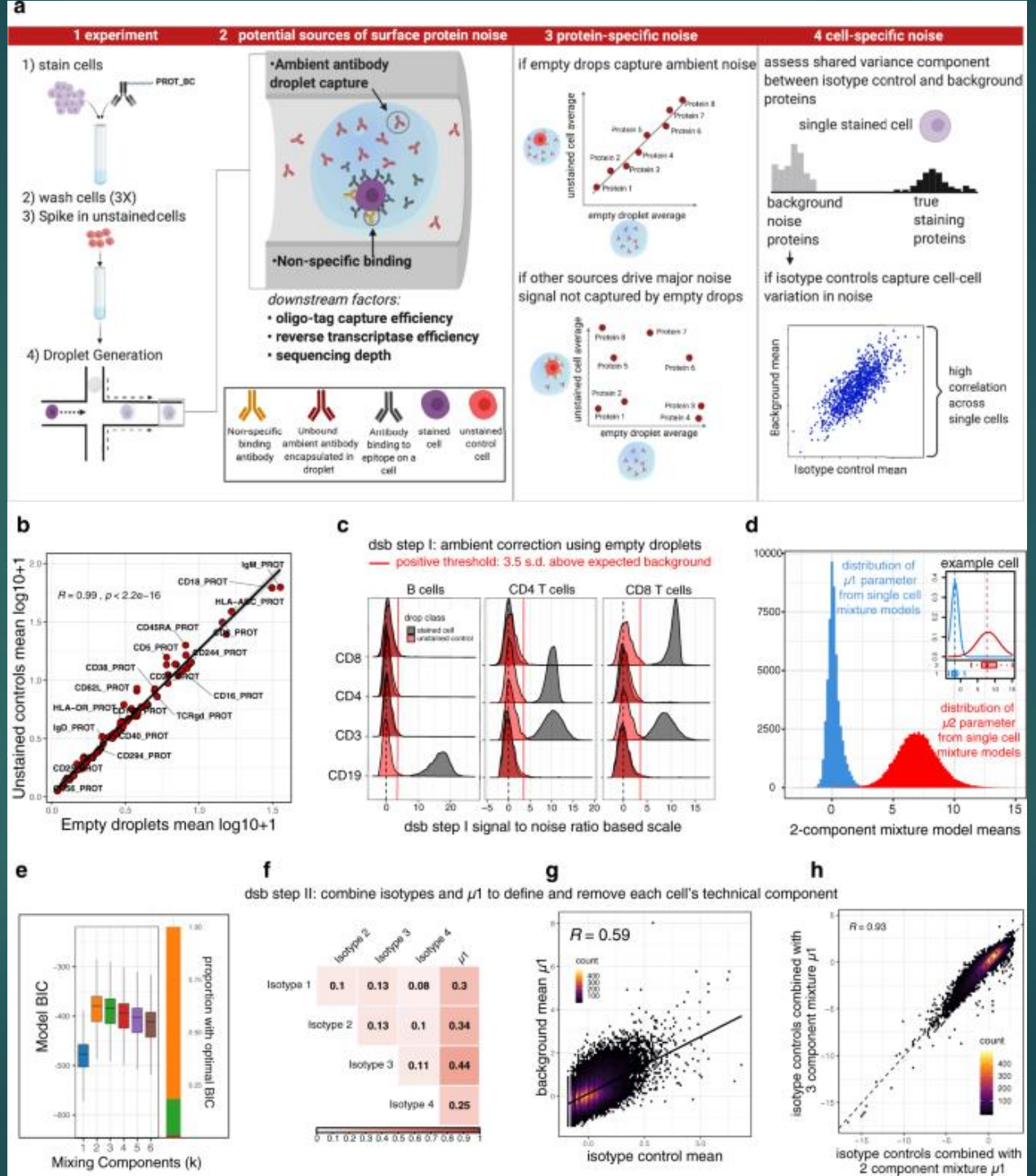


Figure 15.6: **Heatmap plot of enriched terms.** default (A), `foldChange=geneList` (B)

CITE-seq Data

- 3 main types of normalization:
  - 1. CLR (centered log ratio) in Seurat, with margin=1, plots low to high for each features
  - CLR in Seurat, with margin 2, requires the assumption that each cell is stained with roughly the same amount of antibodies- normalizes per cell
  - Denoised & Scaled by Background (dsb) (separate package)
    - This requires the raw output with empty droplets from 10x to specify a background
    - Recommended to have isotype controls as well

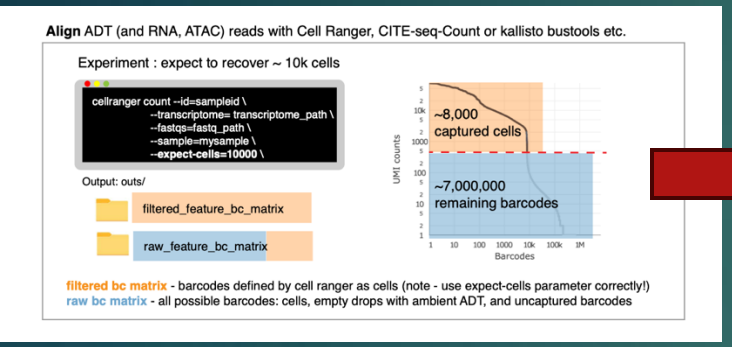
DSB

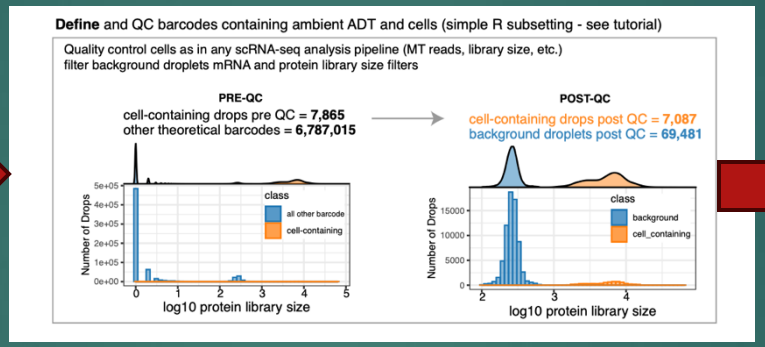


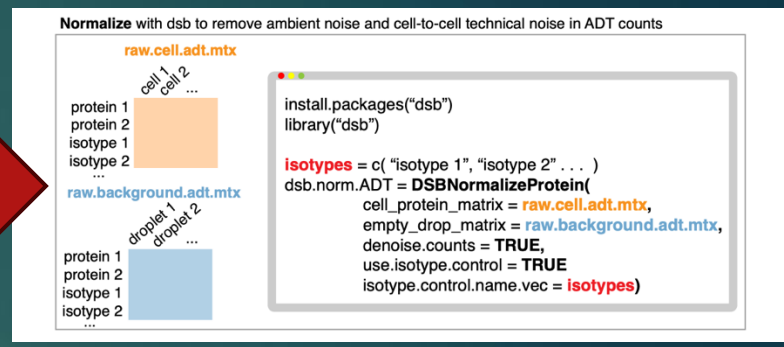
https://github.com/niaid/dsb

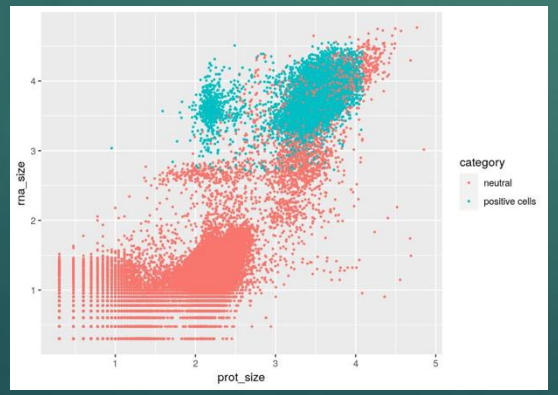
Mule et al., 2022

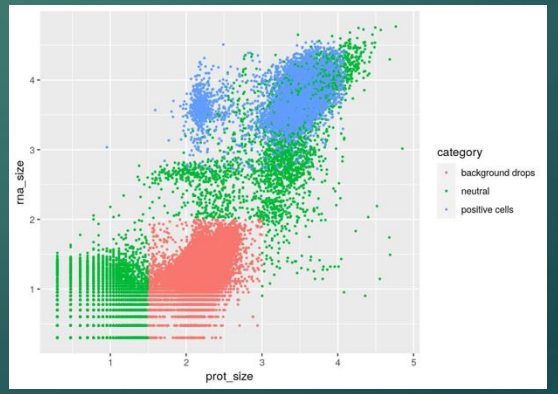
DSB Workflow





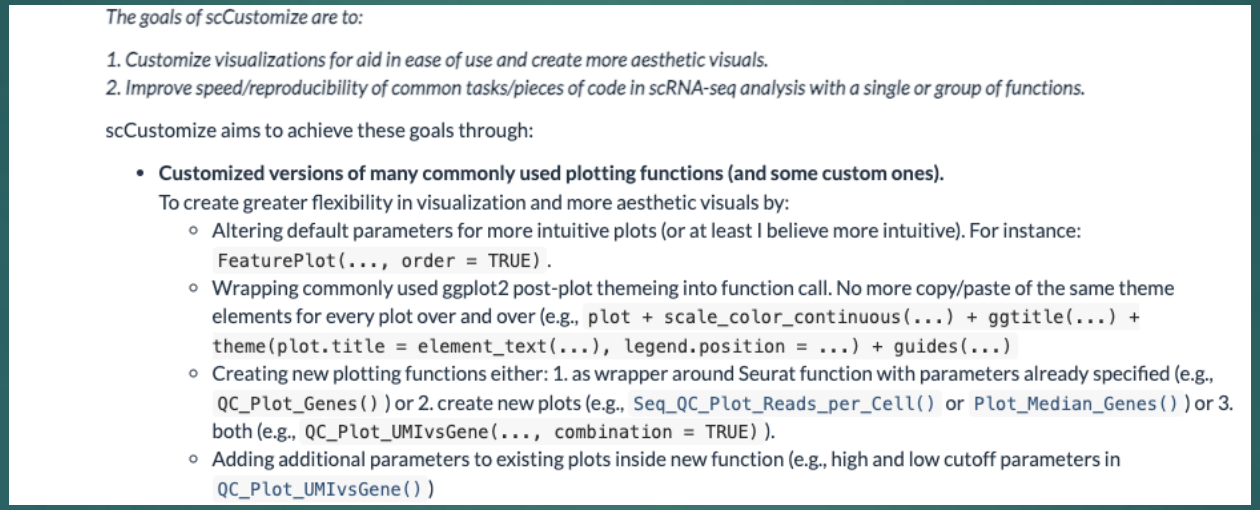






Pretty Plots and More (scCustomize)

- https://samuel-marsh.github.io/scCustomize/index.html



The goals of scCustomize are to:

1. Customize visualizations for aid in ease of use and create more aesthetic visuals.
2. Improve speed/reproducibility of common tasks/pieces of code in scRNA-seq analysis with a single or group of functions.

scCustomize aims to achieve these goals through:

- **Customized versions of many commonly used plotting functions (and some custom ones).**
  To create greater flexibility in visualization and more aesthetic visuals by:
  - Altering default parameters for more intuitive plots (or at least I believe more intuitive). For instance: `FeaturePlot(..., order = TRUE)`.
  - Wrapping commonly used ggplot2 post-plot themeing into function call. No more copy/paste of the same theme elements for every plot over and over (e.g., `plot + scale_color_continuous(...) + ggtitle(...) + theme(plot.title = element_text(...), legend.position = ...) + guides(...)`
  - Creating new plotting functions either: 1. as wrapper around Seurat function with parameters already specified (e.g., `QC_Plot_Genes()`) or 2. create new plots (e.g., `Seq_QC_Plot_Reads_per_Cell()` or `Plot_Median_Genes()`) or 3. both (e.g., `QC_Plot_UMIvsGene(..., combination = TRUE)`).
  - Adding additional parameters to existing plots inside new function (e.g., high and low cutoff parameters in `QC_Plot_UMIvsGene()`)

Homework

- It’s Friday, 10 am, and your PI tells you they have a grant due at 4 pm that same day.
- The PI requests a figure for the single-cell data you received a couple weeks ago, and asks for a figure to demonstrate that you can perform single-cell analysis and identify T cells within your dataset.

- To do: Make a figure fit for a grant, with no more than 3 panels.

Final Thoughts

- Don’t be afraid to not know
- Everyone started at ground zero sometime
- Think of coding as learning how to use a computer- each software/package has its’ own set of quirks
- Not everything published works in real life

Resources

- https://rnabio.org/module-08-scrna/0008/02/01/scRNA/
- http://bioconductor.org/books/release/OSCA/
- https://hbctraining.github.io/scRNA-seq_online/lessons/01_intro_to_scRNA-seq.html
- PCA: https://www.youtube.com/watch?v=FgakZw6K1QQ&t=0s
- https://www.sc-best-practices.org/preamble.html