Comp790-166: Computational Biology

Lecture 18

March 20, 2022

Good Morning Question

1. Who can remind us about the trajectory inference problem that we discussed before the week of project presentations and spring break?

Today

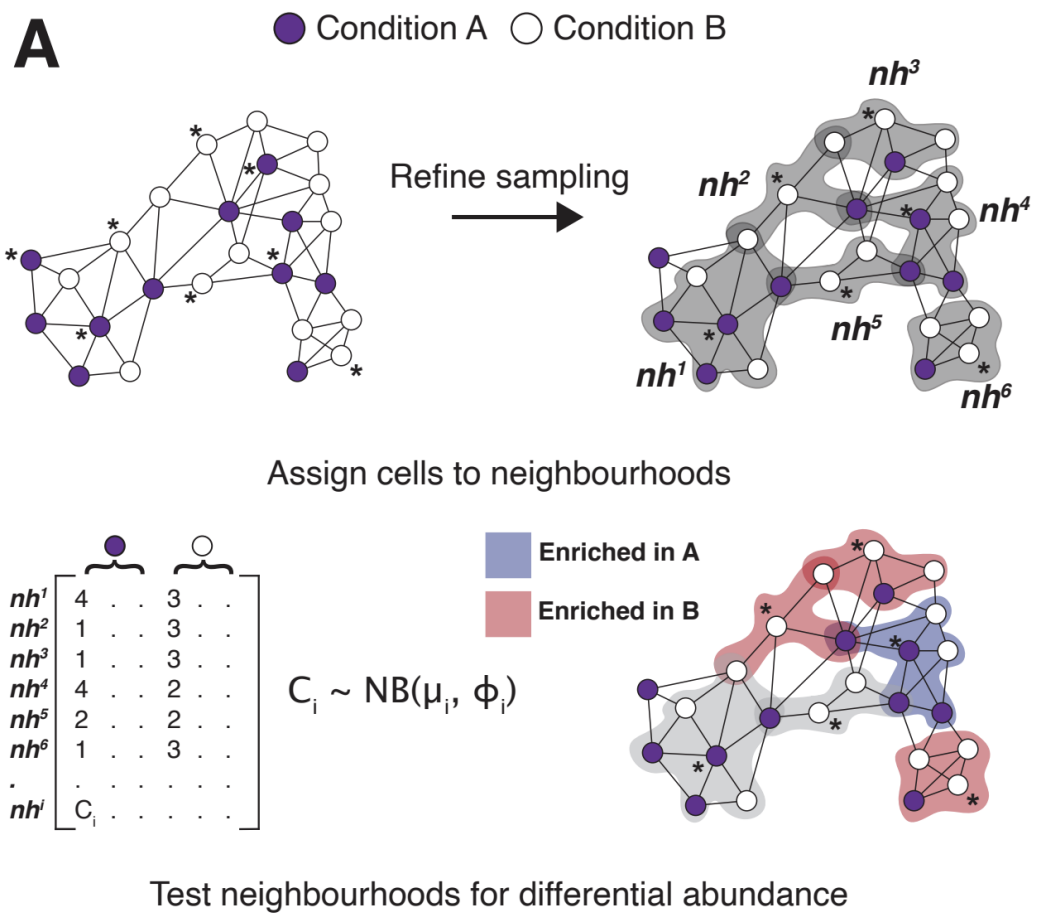
- Milo as a graph-based method for differential abundance.
- Contrastive PCA for dealing with background data.

Intermission for Announcements

- Homework to be returned sometime today by email.
- Reading summaries - don't forget about those!

Milo

Milo is effectively a graph-based version of Cydar.
`https://www.nature.com/articles/s41587-021-01033-z`.



General Overview of Milo

- Build k-NN graph of cells
- Define a representative set of nodes to serve as the ‘center‘ of neighborhoods across the graph
- Define the neighborhood of a node, j, as the collection of cells that are connected to node j by an edge.
- Count cells in each neighborhood. You end up with a matrix of samples $\times$ counts of cells across neighborhoods.
- Test for differential abundance in neighborhoods Spatial FDR again to control for the proportion of neighborhoods that are false-positive.

How Does Milo Do?

MCC (Matthews Correlation Coefficient) is a performance metric that measures performance from integrating multiple performance metrics.

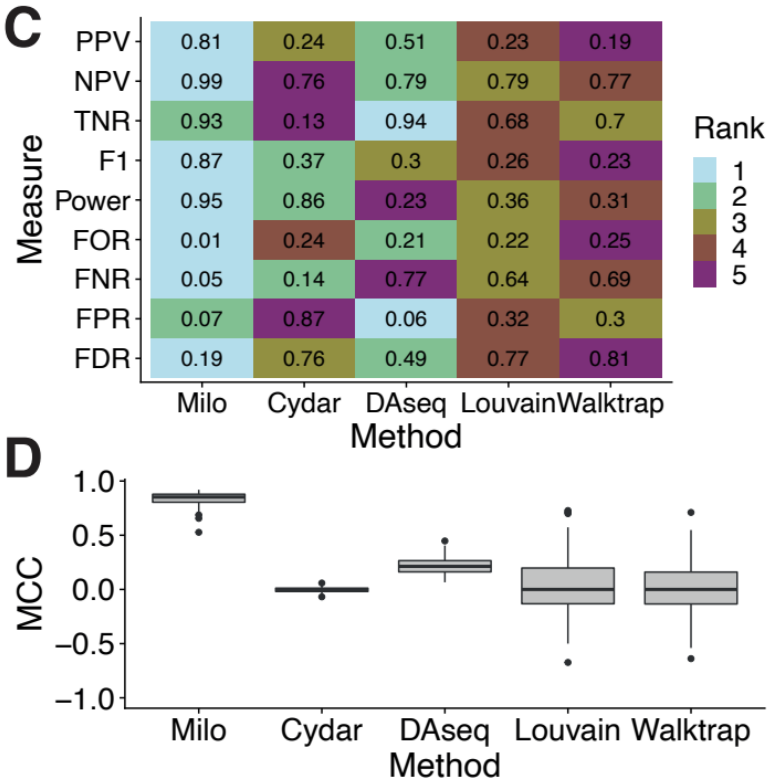


Figure: from Dann *et al.* 2020. BioArXiv

Resulting Visualization of Milo

Milo visualizes the graph between the subsets of selected nodes that were used to form neighborhoods.

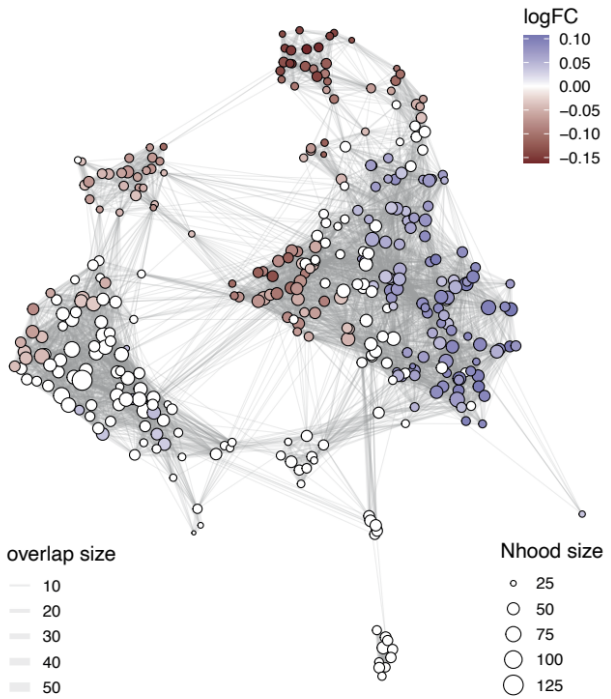


Figure: from Dann *et al.* 2020. BioArXiv. Here the data are single thymic epithelial cells sampled from mice from age 1 to 52 weeks.

Connecting to Ground Truth Labels of the Cells

Each cell as an 'age' associated with it. We can see that cells belonging to neighborhoods that had an increased abundance of cells with age are colored blue.

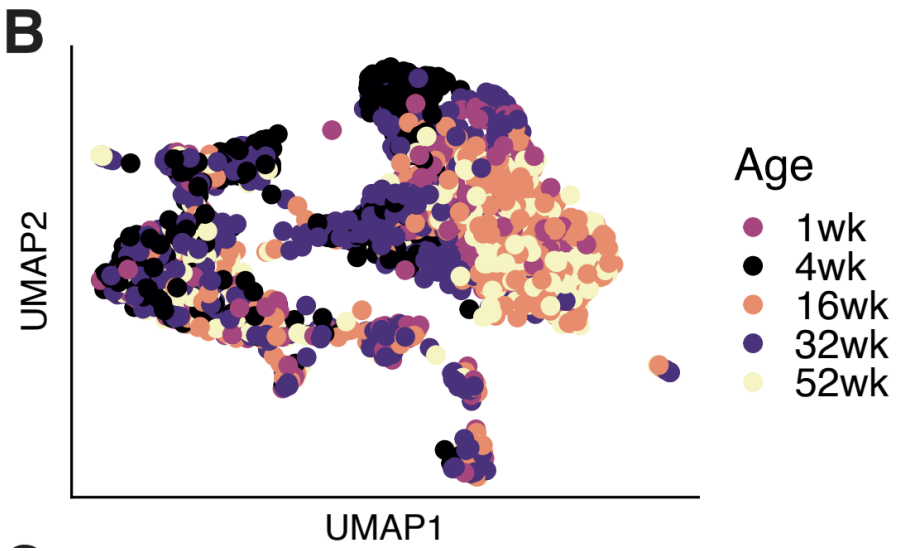


Figure: from Dann *et al.* 2020. BioArXiv. Cells are colored by the age of the mouse that they came from.

Thoughts on Milo + Comparison with Meld

- **Neighborhoods Initialized Randomly.** It seems that we can really do better than choosing nodes at random to serve as the centers of the neighborhoods.
- This problem of choosing seeds on a graph is actually hard. How do you choose seeds that are sufficiently equidistant from other seeds. (For example, this would be easier on a grid)

What if we thinking about a set of control samples as a *background* that we can compare samples from our experiments to?

Switching Gears - Dealing with Background Populations and Data

- Consider high-dimensional gene expression measurements collected from people from all over the world.
- Suppose these patient samples also correspond to healthy and cancer patients.
- If the question is to find gene expression patterns associated with cancer subtypes, PCA on our samples may mostly reflect demographic variation between patients, rather than biological variation related to cancer subtypes.

Intro to Contrastive PCA

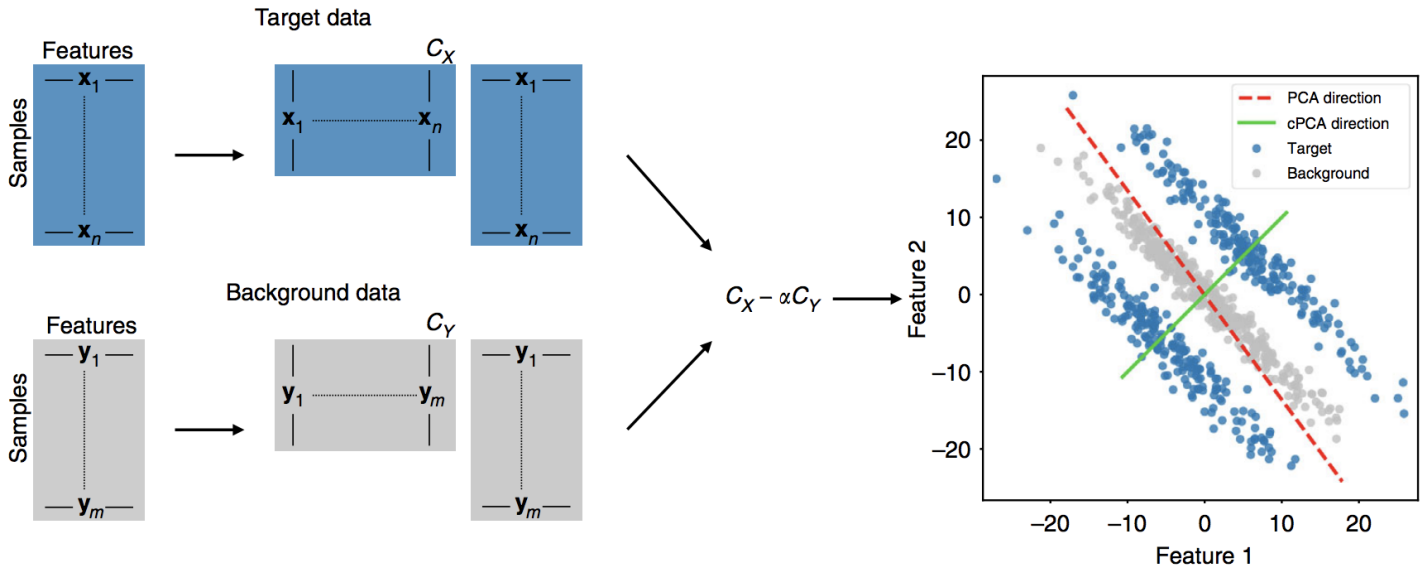


Figure: from Abid *et al.* Nature Communications. 2018. When projecting the data, the goal is to find the target direction that has the highest variance in the target data in comparison to the background data.

Thinking About Background Data

- Given two groups of datapoints (e.g. patient measurements), you can imagine there is variance common to both datasets and variance characteristic of each one.
- For example, thinking about a control group and a disease group, both have population-level variation, but the disease group has particular disease subtypes.
- As another example, consider time series data when you want to decouple variation from a particular timepoint from variation across the entire time series.
- Choice of background dataset is important here and should ideally contain ‘structure‘ that we would like to remove form the target data.

Motivating Biological Examples

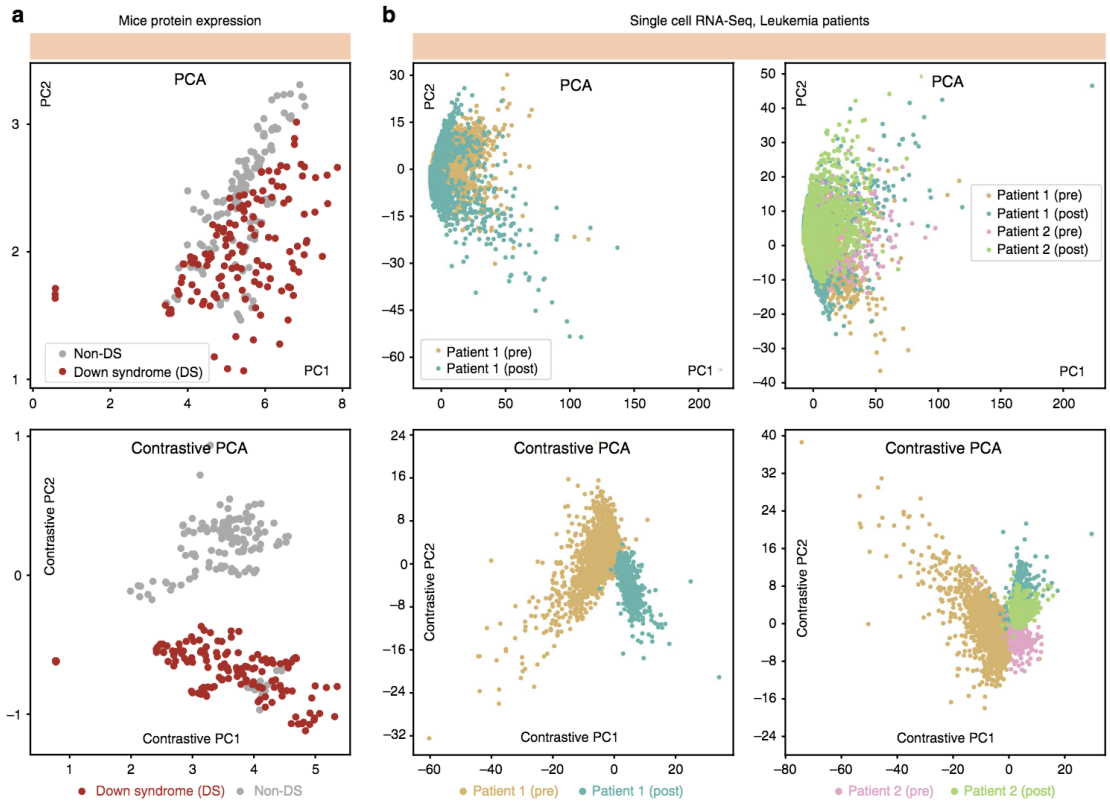


Figure: from Abid *et al.* Nature Communications. 2018. (Left) : Protein expression in Down Syndrome vs Non Down Syndrome Mice. Single cell data pre and post transplant.

cPCA Problem Setup

- Assuming we start with d-dimensional target data $\{\mathbf{x}_i \in \mathbb{R}^d\}$ background data $\{\mathbf{y}_i \in \mathbb{R}^d\}$

For some direction vector, $\mathbf{v} \in \mathbb{R}^d_{\text{unit}}$ the variance it accounts for in the target and background data can be expressed as,

$$\text{Target data variance} : \lambda_X(\mathbf{v}) \stackrel{\text{def}}{=} \mathbf{v}^T \mathbf{C}_X \mathbf{v}$$

$$\text{Background data variance} : \lambda_Y(\mathbf{v}) \stackrel{\text{def}}{=} \mathbf{v}^T \mathbf{C}_Y \mathbf{v}$$

What is happening here and what does this remind you of?

Given a contrast parameter $\alpha \geq 0$ that quantifies the trade-off between having high target variance and low background variance, cPCA computes the contrastive direction $\mathbf{v}^*$ by optimizing

$$\mathbf{v}^* = \text{argmax}_{\mathbf{v} \in \mathbb{R}^d_{\text{unit}}} \; \lambda_X(\mathbf{v}) - \alpha \lambda_Y(\mathbf{v})$$

This problem can be rewritten as

$$\mathbf{v}^* = \text{argmax}_{\mathbf{v} \in \mathbb{R}^d_{unit}} \; \mathbf{v}^T \left(C_X - \alpha C_Y \right) \mathbf{v}$$

cPCA is Quite Simple!

Algorithm 1 cPCA for a Given α

Inputs: target data $\{\mathbf{x}_i\}_{i=1}^n$; background data $\{\mathbf{y}_i\}_{i=1}^m$; contrast parameter α; the number of components k.

Centering the data $\{\mathbf{x}_i\}_{i=1}^n$, $\{\mathbf{y}_i\}_{i=1}^m$.
Calculate the empirical covariance matrices:

$$C_X = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T, C_Y = \frac{1}{m}\sum_{i=1}^{m} \mathbf{y}_i \mathbf{y}_i^T.$$

Perform eigenvalue decomposition on

$$C = (C_X - \alpha C_Y).$$

Compute the the subspace $V \in \mathbb{R}^k$ spanned by the top k eigenvectors of C.
Return: the subspace V.

Figure: Just do eigendecomposition on **C** and consider the eigenvectors corresponding to the top k eigenvalues of C.

Effect of Varying α

For $\alpha = 0$, cPCA will create directions that maximize the target variance. For higher α, directions with smaller background variance become more important.

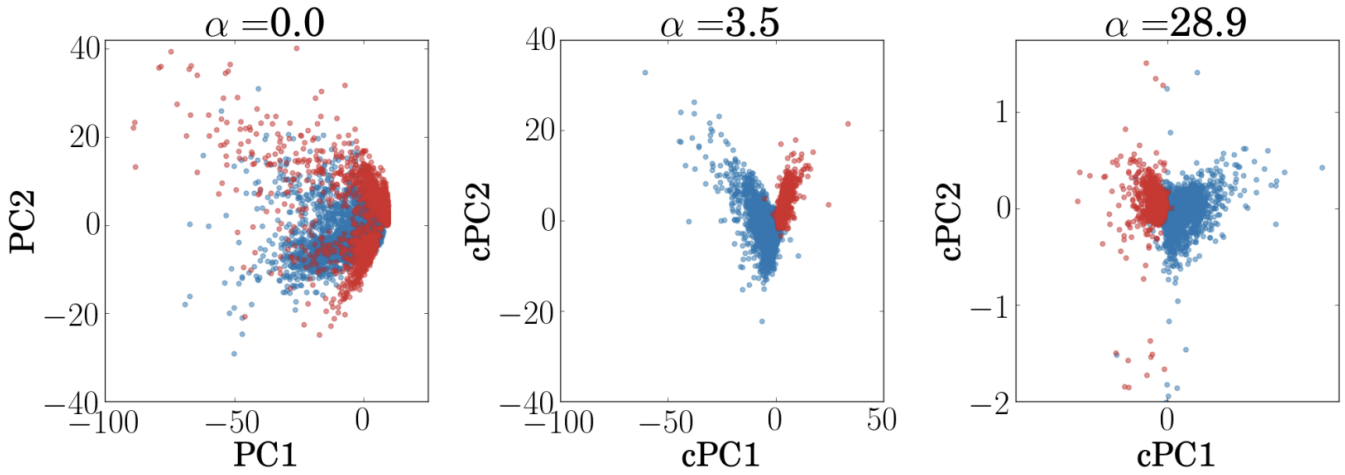


Figure: from Abid *et al.* Nature Communications. 2018. This dataset is visualizing cells from two different samples.

Recap

We have now seen Meld vs Cydar vs Milo vs cPCA (if we have a background). Thoughts? What would you do to compare your samples from disparate clinical or experimental groups?

Combining Multiple Single-Cell Datasets

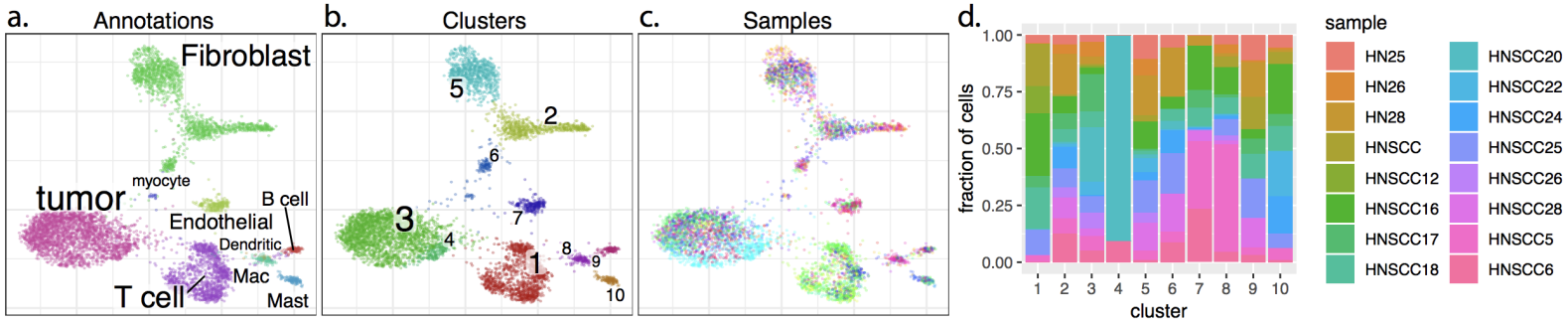


Figure: from Barkas *et al.* Nature Methods. 2019. Conos looks at how to integrate cells from multiple datasets (patients, tissues, etc.)

- The problem is a bit different from batch effect effect correction where you can identify technical artifacts and get rid of them. Cell-populations might be completely missing from particular datasets.

Conos Overview: Construct a Joint Between-Cell Graph

The goal is to establish a unified graph representation of the multiple single-cell datasets. Specifically, to infer cell-populations across all datasets, Conos seeks to infer inter-cell edges between datasets.

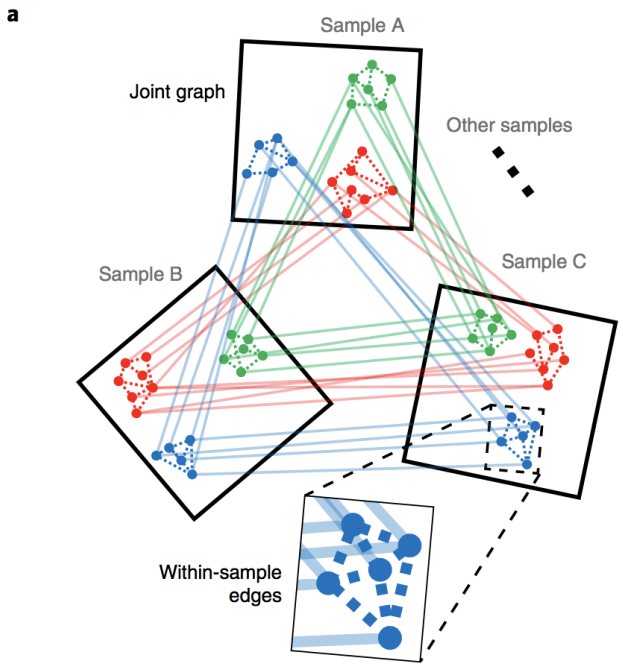


Figure: from Barkas *et al.* Nature Methods. 2019.

Pairwise Dataset Alignment

- As a pre-processing step, choose a set of high-variance genes. (The authors use 2,000).
- For a pair of datasets, i and j, let G_i and G_j denote their corresponding set of features measured per cell. Then consider only features that are measured in both datasets (so $G_i \cap G_j$)
- The similarity between cells K and l in datasets i and j is $w_{kl} = \exp(-\frac{||M_k^i - M_l^j||}{\sigma})$

Creating the Joint Graph

- Use w_{kl} for k-NN gaphs
- For **inter-sample edges**, connect each cell to its 15 nearest neighbors by default
- For **intra-sample edges**, connect each cell to its 5 nearest neighbors.
- Create joint clusters by clustering the graph using a graph-based community detection method.

Controlling Mixing Between Datasets

- Add a k_1 parameter or mixing parameter that allows for an increase of the nearest neighbor search radii, k. Control k_1 with an alignment strength parameter, $k_1 = \alpha^2 K_{\text{max}}$ (K_{max} is the maximum number of total cells across samples in the panel.

α ranges between 0 and 1 and 0 corresponds to alignment with no addition edges, and 1 corresponds to a full alignment.

This is followed by a pruning step...

They have a little strategy to reduce maximal degree closer to k and to make the graph less dense.

- Order nodes from highest to lowest degree
- For each node, order edges by the degree of target vertices (high to low)
- Algorithm goes through nodes and corresponding edges and removes an edge if the degrees of both incident nodes are larger than a specific cutoff, k_0

Rebalance Edge Weights

- Since samples are often collected across conditions, the authors wanted to provide flexibility to control how likely pairs of cell populations are to be mapped to each other, between conditions. Specifically, balance edge weights between cells connected between the same or different values of a factor.

The solution is to minimize the following,

$$\sum_{l=1}^{N_{\text{factors}}} \sum_{s=1}^{N_{\text{cells}}} \left| \frac{\sum_{t \in \text{adj}_l(s)} w_{st}}{\sum_{t \in \text{adj}(s)} w_{st}} - \frac{1}{N^s_{\text{factors}}} \right|$$

Unpacking...

$$\sum_{l=1}^{N_{\text{factors}}} \sum_{s=1}^{N_{\text{cells}}} \left| \frac{\sum_{t \in \text{adj}_l(s)} w_{st}}{\sum_{t \in \text{adj}(s)} w_{st}} - \frac{1}{N^s_{\text{factors}}} \right|$$

- N_{factors} is the total number of factor levels
- N_{cells} is the total number of cells.
- $adj(s)$ is the set of cells adjacent to cell s.
- $adj_l(s)$ is the set of cells adjacent to s and belong to factor level l.
- w_{st} is the weight of the edge between cells s and t
- N^s_{factors} is the number of different factors of cells connected to s.

Imbalance Between Factor l and cell s

For their minimization they first estimate the imbalance ratio for a cell s and a factor level, l as,

$$u_{sl} = N^s_{\text{factors}} \frac{\sum_{t \in \text{adj}_l(s)} w_{st}}{\sum_{t \in \text{adj}(s)} w_{st}}$$

Using Imbalance to Update Edge Weights

Edge weights are updated using the imbalance computed in the previous slide as,

$$w_{st} = \frac{w_{st}}{\sqrt{u_{sl}\, u_{tl_s}}}.$$

- Here l_c denotes the factor level of cell c.
- This process is repeated 50 times.

Effect of Alignment Strength

Here is an example varying alignment strength on a dataset containing cells from multiple technologies.

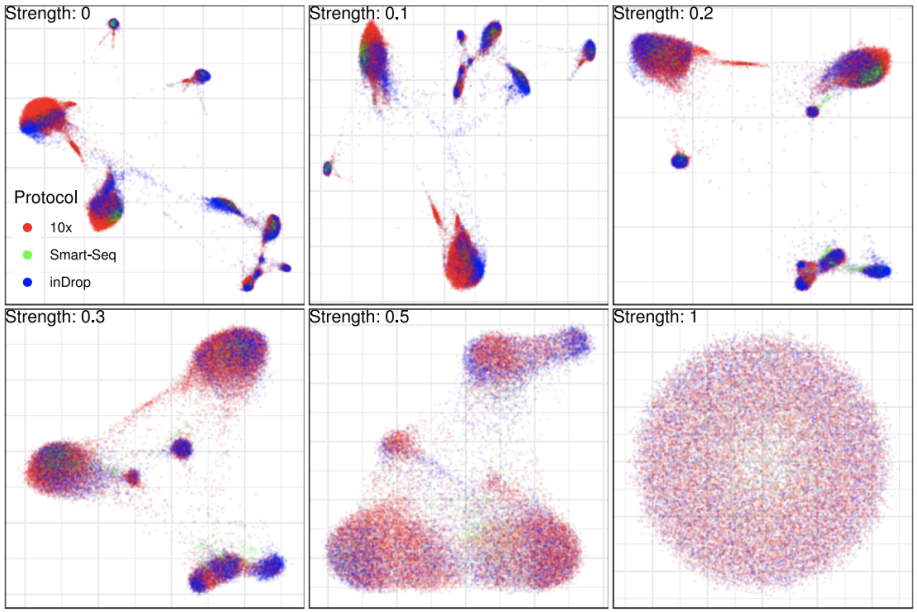


Figure: from Barkas *et al.* Nature Methods. 2019.

Example 1: Bone Marrow and Chord Blood

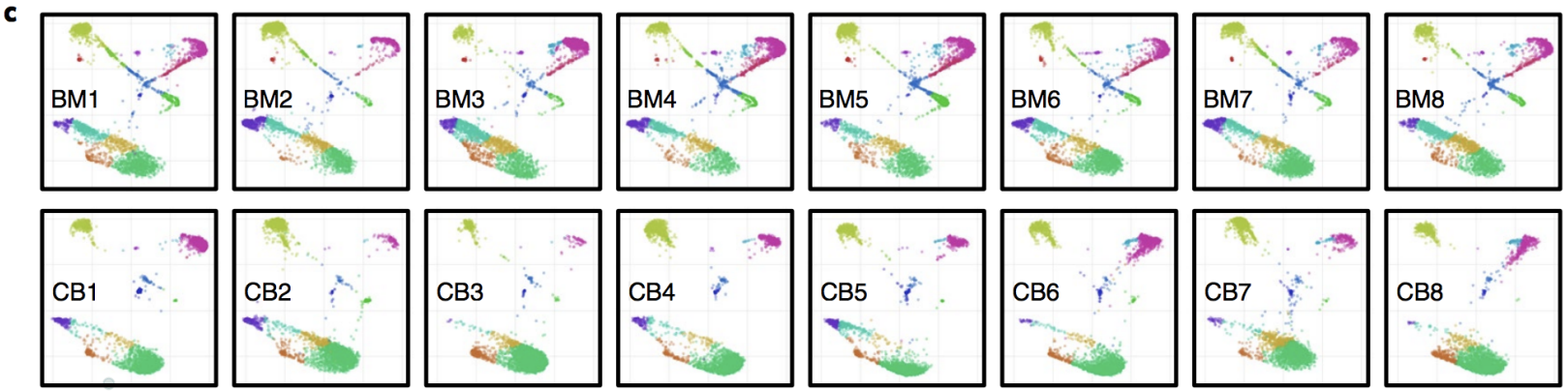


Figure: from Barkas *et al.* Nature Methods. 2019. You can see similarities and differences between cell-populations in each dataset.

Experiment 1: Adding Noise and Recovering Clusters

Noise was added to increase heterogeneity and decrease signal between samples.

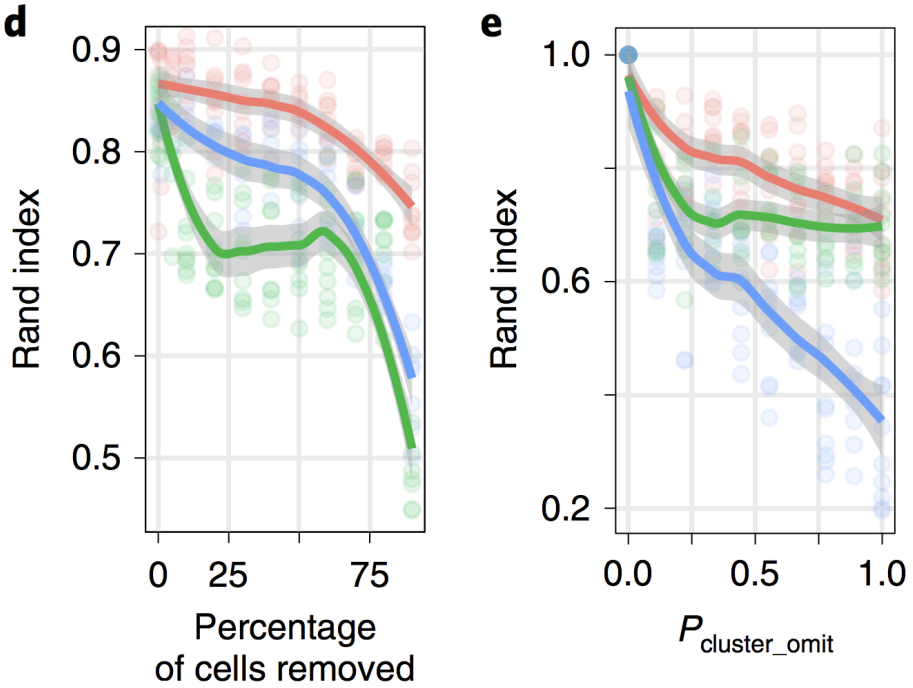


Figure: from Barkas *et al.* Nature Methods. 2019. They made perturbations by decreasing the number of cells or by decreasing the magnitude of expression-specific signatures.

Evaluating Cluster Entropy

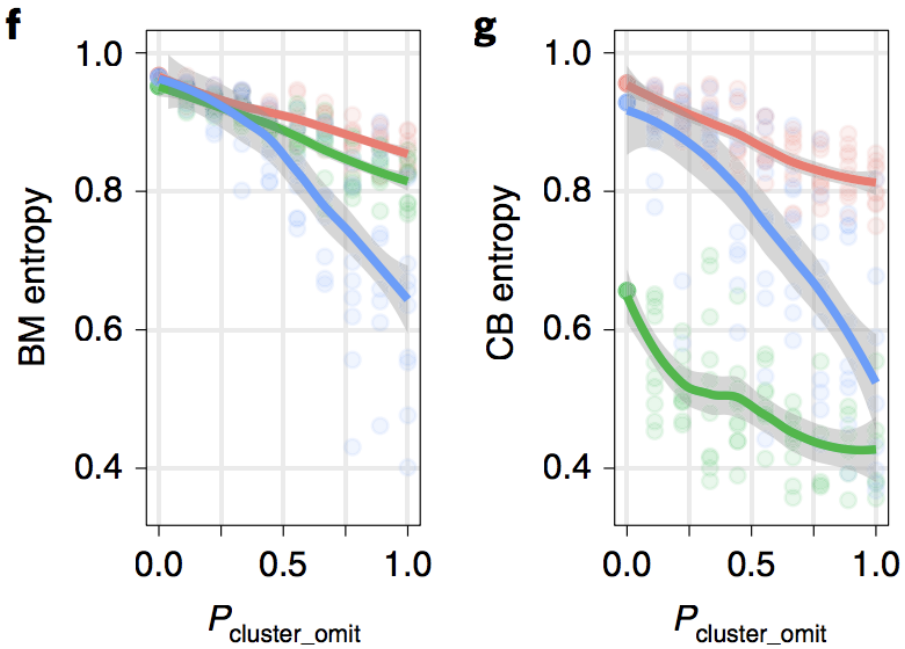


Figure: from Barkas *et al.* Nature Methods. 2019. Conos was able to maintain high entropy within clusters, or ensuring that clusters contained cells across all samples.

Visualizing the Joint Graph

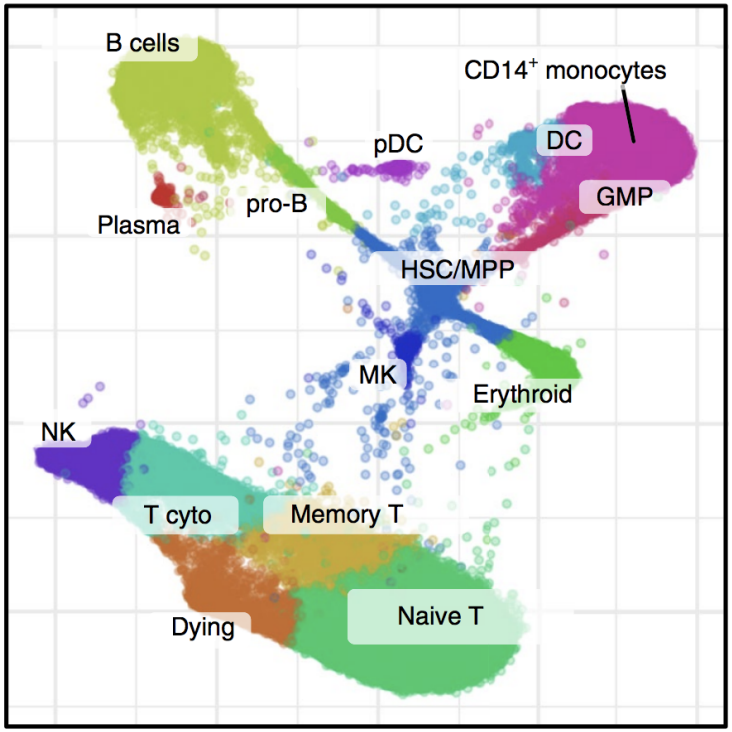


Figure: from Barkas *et al.* Nature Methods. 2019. The layout of the joint graph is determined by LargeVis.

Ensuring to Group Cells by Phenotype, not State

- Suppose you had CD4+ T cells in a cancer sample and CD4+ in a healthy sample. Because their function in the cancer sample is disrupted, it might not cause cells to cluster by phenotype. What we really want is a unified cluster of CD4+ across all samples.

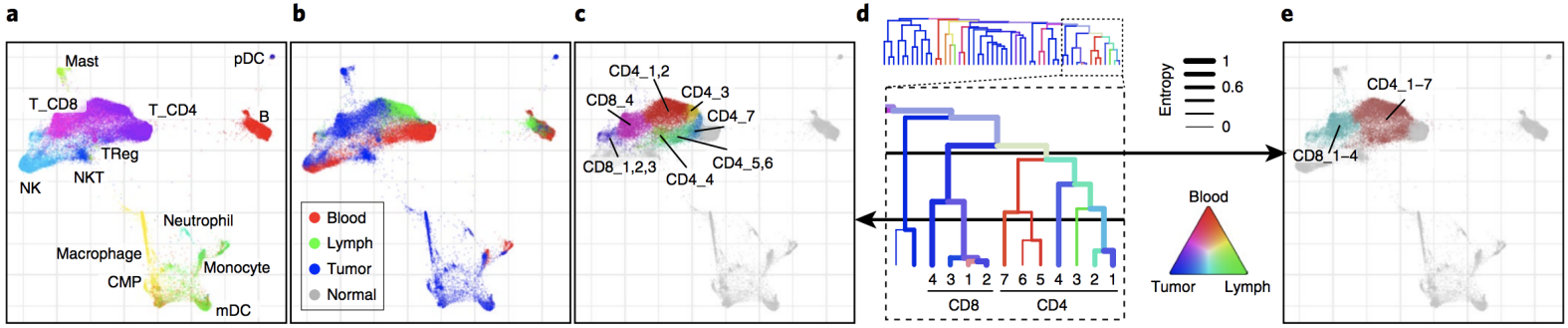


Figure: from Barkas *et al.* Nature Methods. 2019.

Predicting a Cell's Label from the Graph Structure

Label propagation can be used to predict a cell's label based on the labels of neighboring cells.

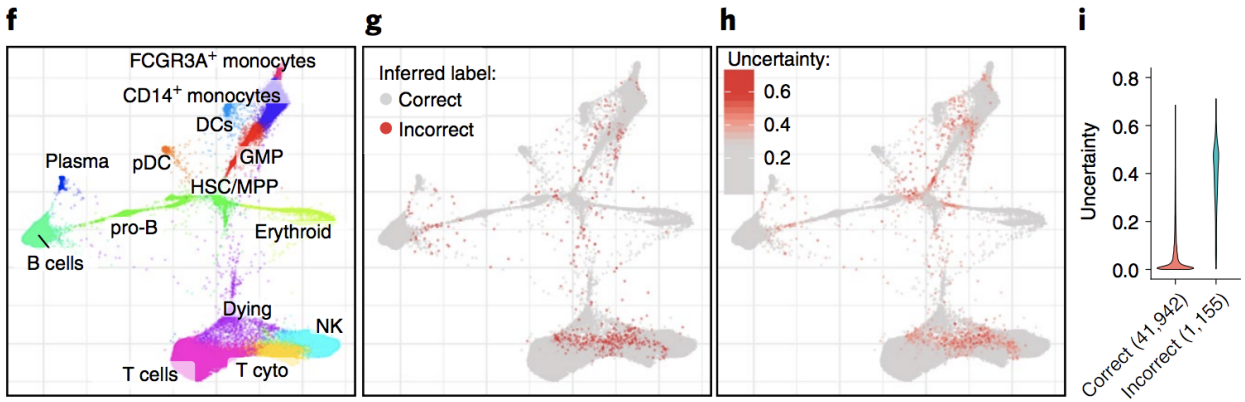


Figure: from Barkas *et al.* Nature Methods. 2019. Cells colored red represent those with incorrect predictions.

Run-time Based on Cells and Datasets

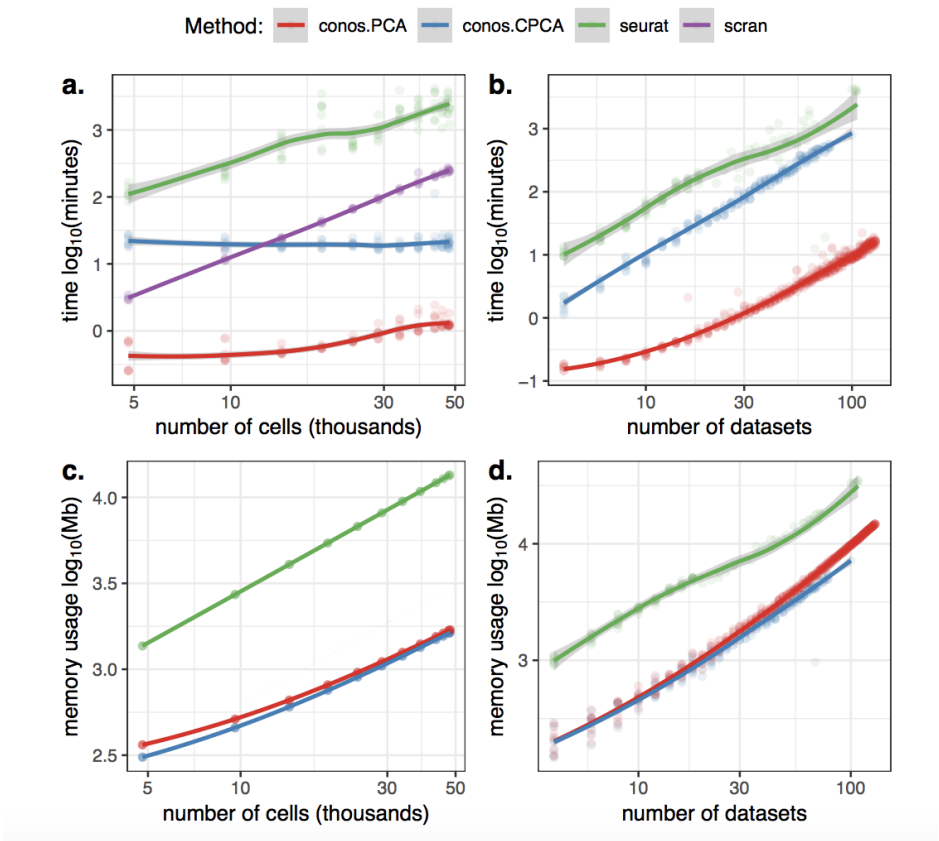


Figure: from Barkas *et al.* Nature Methods. 2019.