Comp790-166: Computational Biology

Lecture 19

March 22, 2022

Good Morning Question

1. Who can give an example of target and background datasets that could be used by cPCA

Today

- Conos for merging cells from multiple samples and conditions
- SLICER and trajectories

Intermission for Announcements

- Homework returned.
- Next homework to be assigned $\sim$ April 6

Combining Multiple Single-Cell Datasets

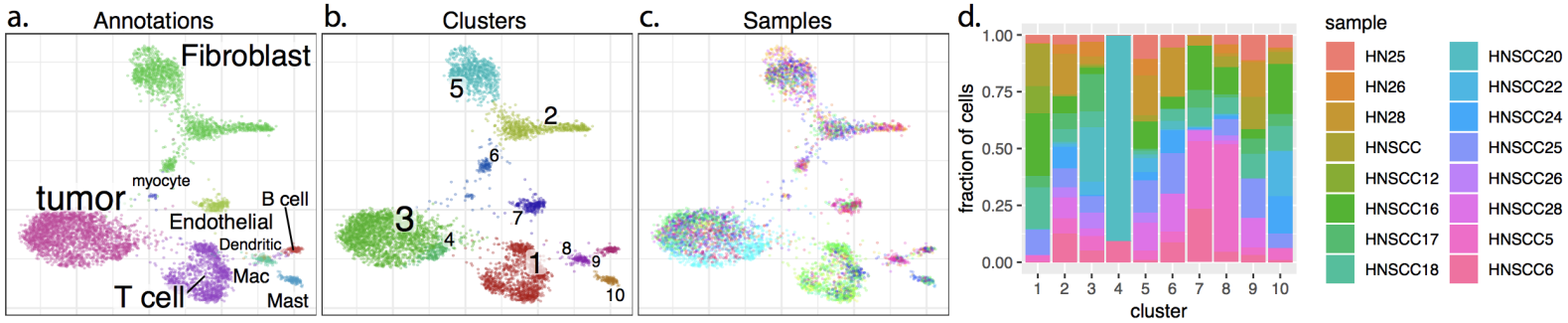


Figure: from Barkas *et al.* Nature Methods. 2019. Conos looks at how to integrate cells from multiple datasets (patients, tissues, etc.)

- 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.

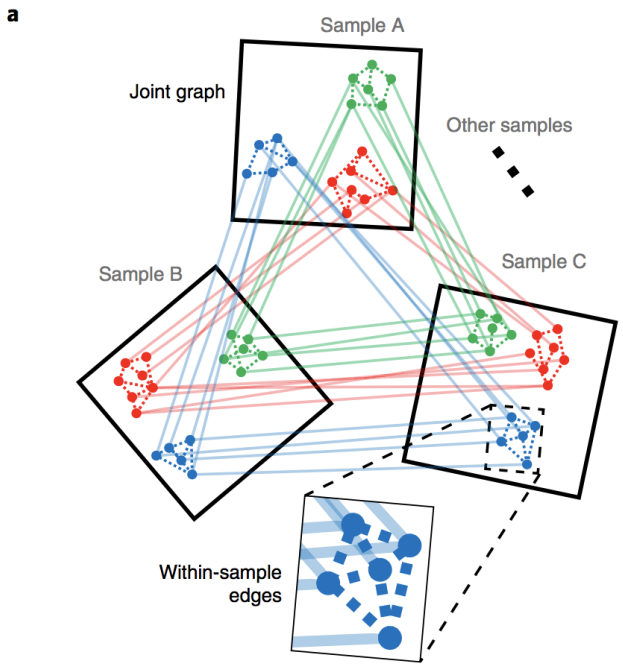


Figure: from Barkas *et al.* Nature Methods. 2019.

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.

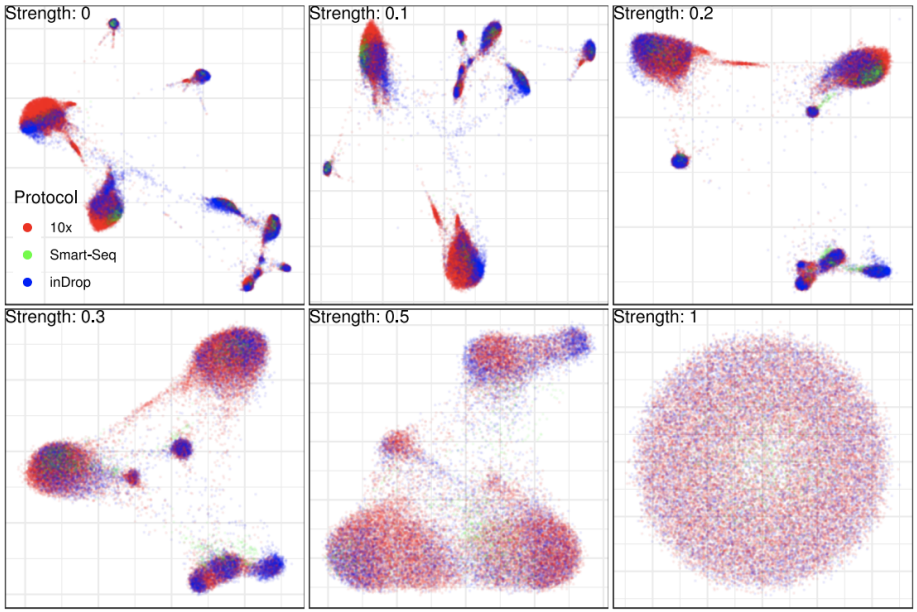


Figure: from Barkas *et al.* Nature Methods. 2019.

Example 1: Bone Marrow and Chord Blood

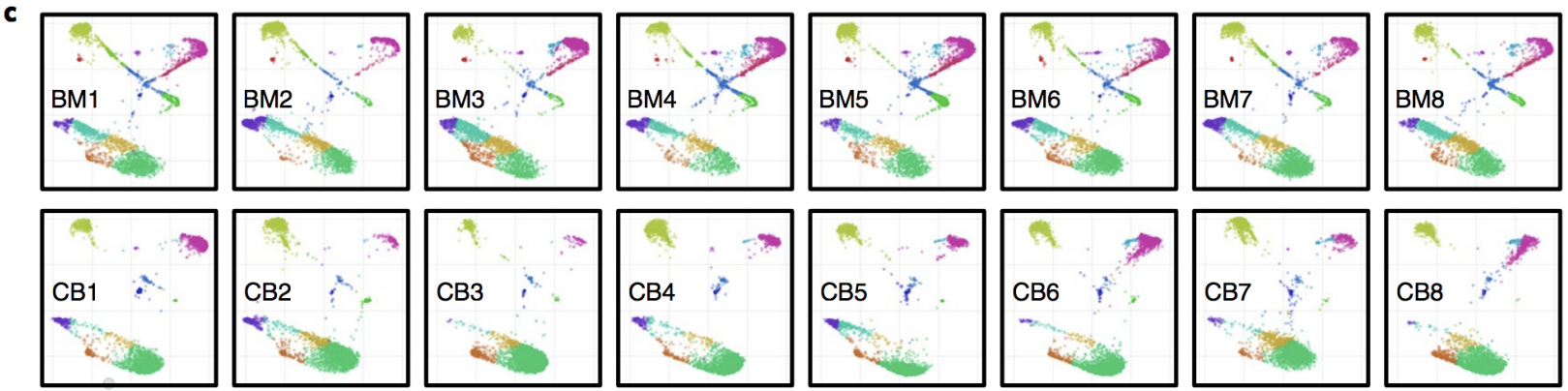


Figure: from Barkas *et al.* Nature Methods. 2019. You can see similarities and differences between cell-populations in each dataset.

Visualizing the Joint Graph

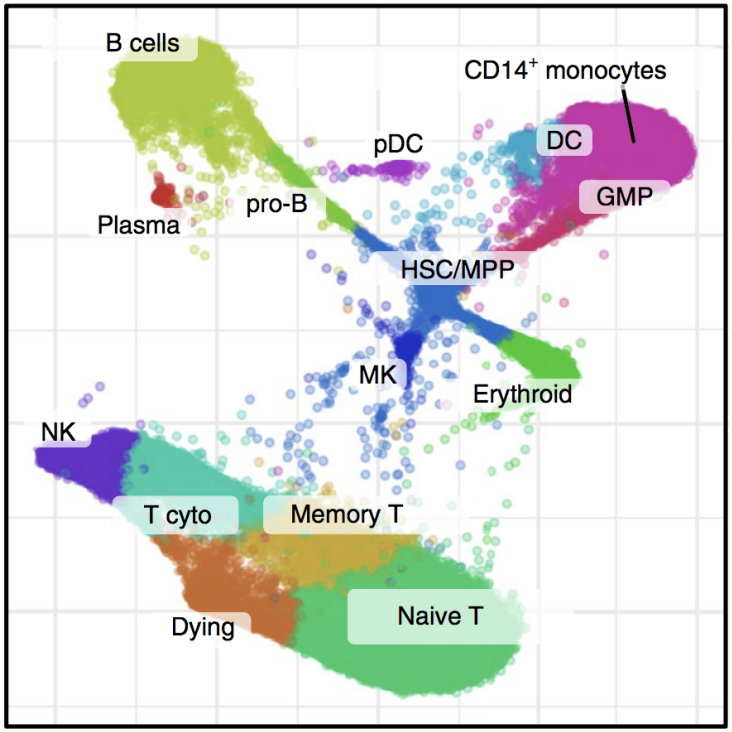


Figure: from Barkas *et al.* Nature Methods. 2019. The layout of the joint graph is determined by LargeVis.

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.

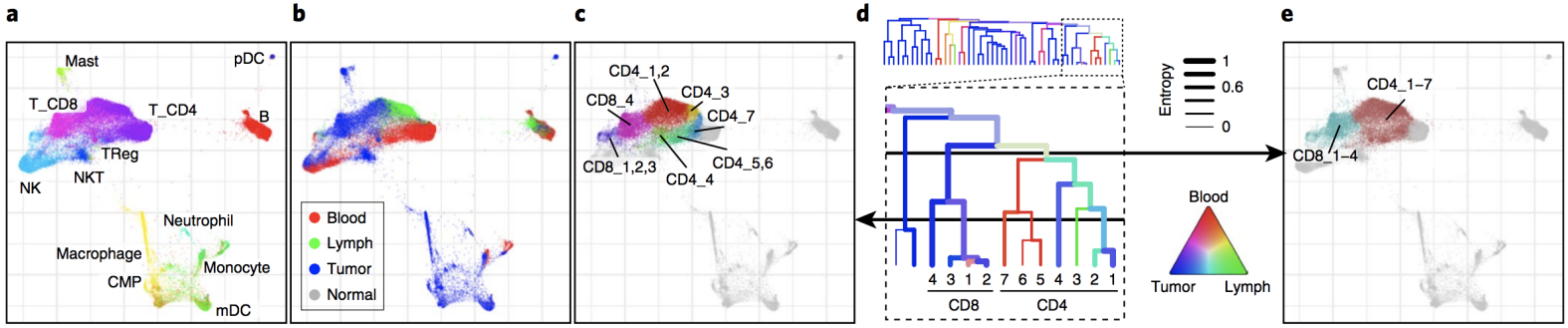


Figure: from Barkas *et al.* Nature Methods. 2019.

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.

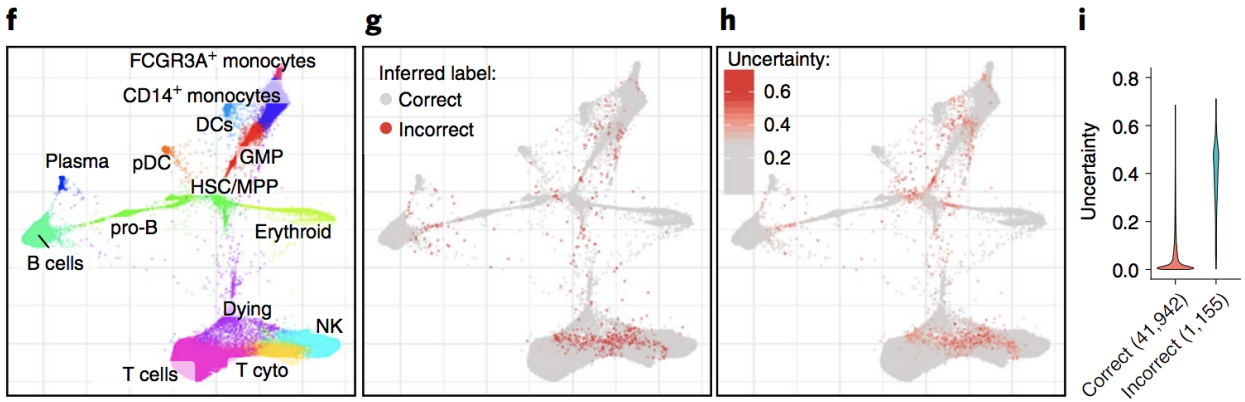


Figure: from Barkas *et al.* Nature Methods. 2019. Cells colored red represent those with incorrect predictions.

Recap

- We just saw a couple of ways that we can combine multiple datasets single-cell datasets.
- These multiple datasets can correspond to **batches, technologies, or different tissue samples.**
- Conos tries to define a graph such that its structure (in this case, communities) contain cells that are mixed across datasets
- When integrating multiple datasets, we must be careful that cells are grouping together because of phenotype, not because of the function that might have been perturbed under a particular condition.

Welcome SLICER

SLICER builds on and expands the very early Diffusion based techniques through the following

- Automatically select genes to use for building the trajectory (or in establishing the ordering between cells)
- Use locally linear embedding to capture non-linear relationships between gene expression levels and progression through a process
- Define ‘geodesic entropy‘ and use it to define branches
- Capture unique trajectory patterns such as bubbles.

SLICER Overview

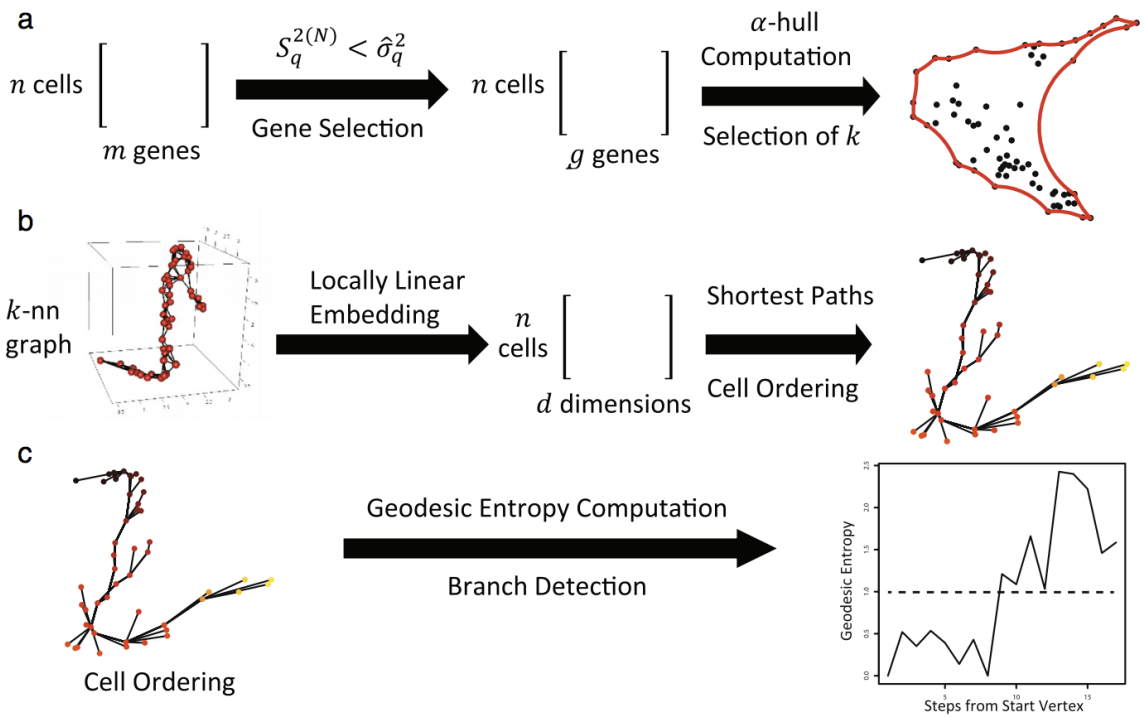


Figure: from Welch *et al.* Genome Biology. 2016

Step 1: Selecting Features to Use (Intuition)

Establishing some intuition about what makes a good ‘trajectory feature‘

- If a feature is involved in progression along a trajectory, expect gradual change in that feature along the trajectory
- A feature not involved should not fluctuate along the trajectory.
- In real life, we have no idea what is happening with this trajectory. Use similarity within neighborhoods to study ‘segments‘ of a trajectory.

Neighborhood Variance

Interesting features are those whose variance is greater than some level of neighborhood variance. Specifically, for the gth feature, we can compute its variance (σ_g^2) across samples and compare it (making sure it is at least as large as) to this defined neighborhood variance.
The neighborhood variance is defined as,

$$S_g^{2(N)} = \frac{1}{nk_c - 1} \sum_{i=1}^{n} \sum_{j=1}^{k_c} \left(e_{ig} - e_{N(i,j)g} \right)^2$$

- k_c is the number of nearest neighbors needed for each node for the graph to be connected.
- Each e_{ig} is representing the value of feature g in cell i.
- $e_{N(i,j)g}$ is representing the feature value of the jth nearest neighbor in cell i.

Local Linear Embedding ($d = 2$)

Step 1: Find the weights (w_{ij}s) that can best reconstruct the original data (e.g. the Es cell $\times$ feature) in terms of k nearest neighbors as,

$$W = \text{argmin}_W \sum_{i=1}^{n} \left| E_i - \sum_{j=1}^{k} w_{ij} E_j \right|_2^2$$

Step 2: Find optimal d-dimensional embedding, so in this case, L

$$L = \text{argmin}_L \sum_{i=1}^{n} \left| L_i - \sum_{j=1}^{k} w_{ij} L_j \right|_2^2$$

k-NN graph and shortest path

- Compute k-nearest neighbor graph between cells in terms of the LLE-determined coordinates.
- Specify a starting point (like a stem cell), and use a shortest path algorithm like Dijkstra to find the shortest path to some cell of interest.

Detecting Branches with Geodesic Entropy Measure

- Let $t_i = \{s = v_1, \ldots, v_k, \ldots, v_l = i\}$ be the shortest path from the starting point s to some cell, i.
- Denote the kth node on the shortest path from s to i by $t_i(k)$.
- Define f_{jk} as the number of paths passing through point j at distance k, $f_{jk} = \sum_i^n I[t_i(k) = j]$
- Then compute the fraction of all paths in S that pass though node j at distance k, $p_{jk} = \frac{f_{jk}}{\sum_{i=1}^n f_{ik}}$
- $H_k = -\sum_{i=1}^n p_{ik} \log_2 p_{ik} \rightarrow$ look at high entropy

SLICER Applied to Synthetic Data

Studying geodesic entropy over k. Higher entropy in terms of steps corresponds to the 'bubbles' in the data.

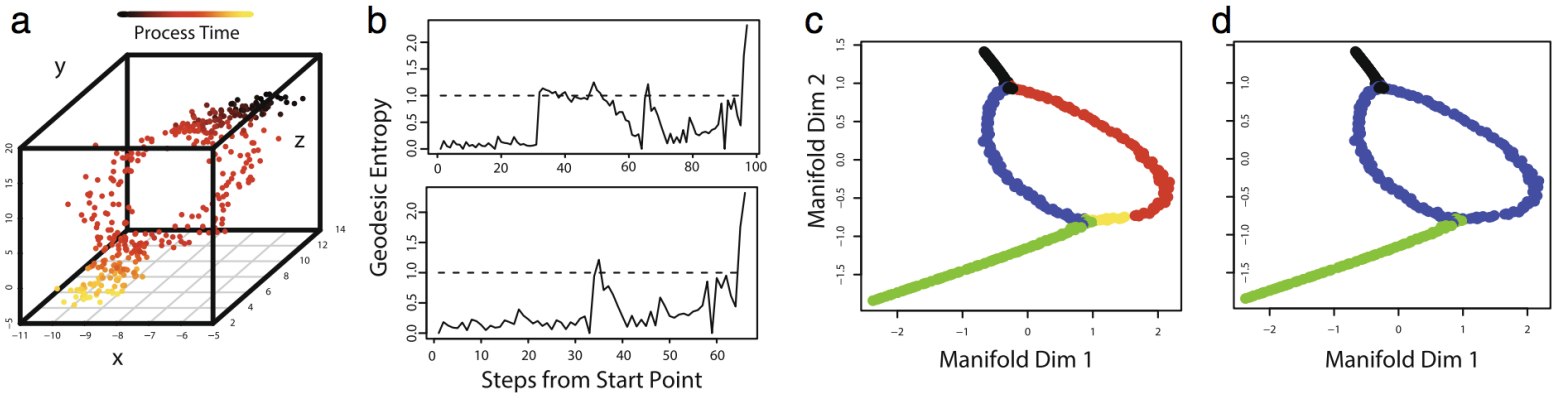


Figure: from Welch *et al.* Genome Biology. 2016.

Neural Stem-Cell Differentiation Data

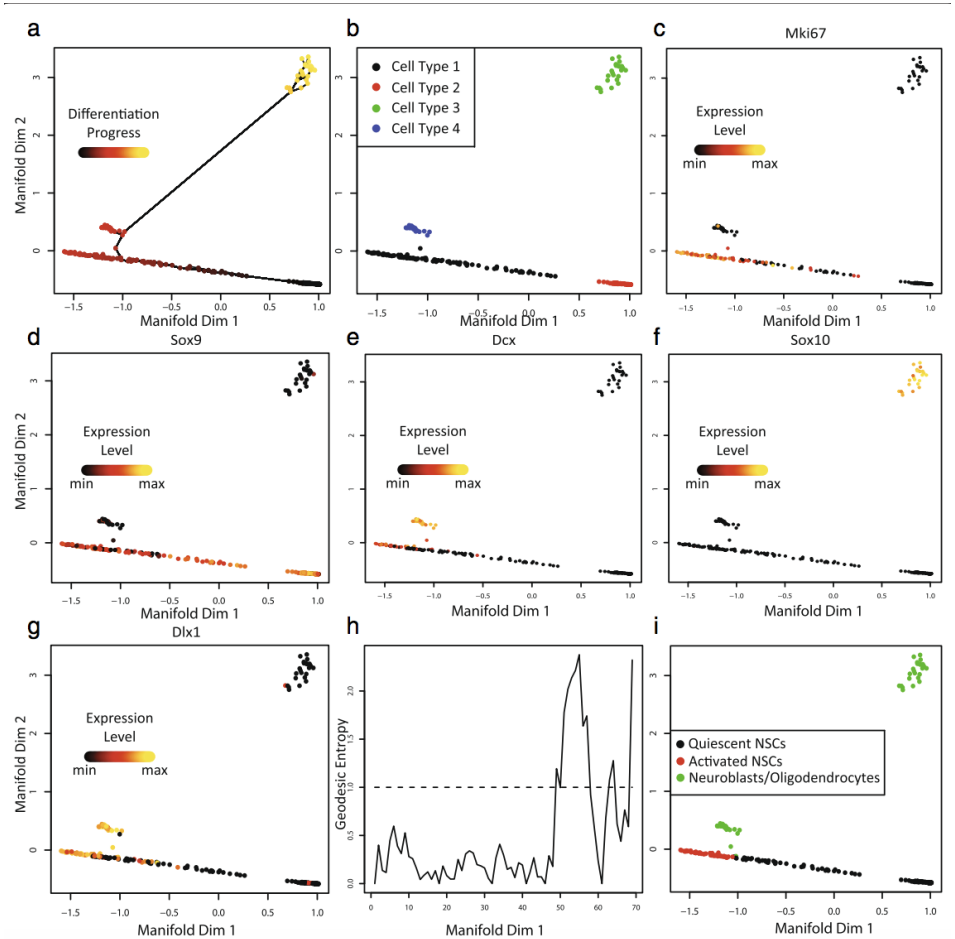
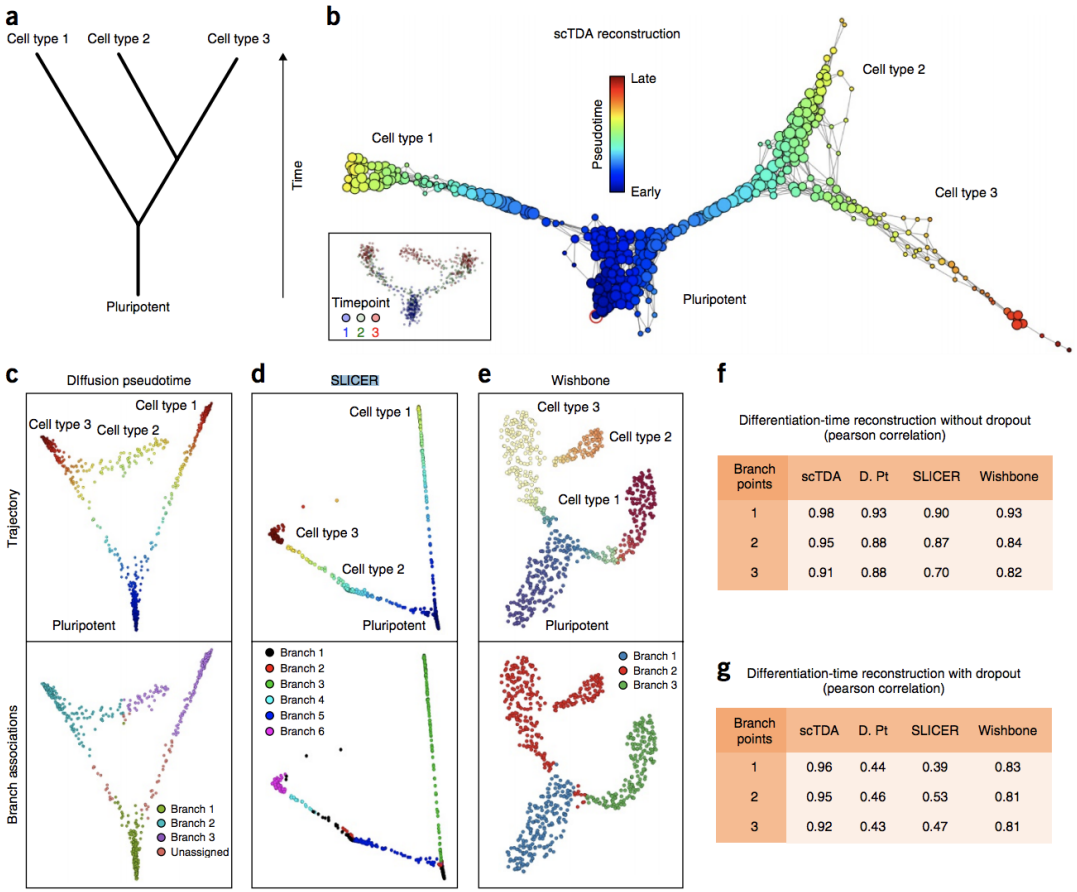


Figure: from Welch *et al.* Genome Biology. 2016.

SLICER Compared



Branch points	scTDA	D. Pt	SLICER	Wishbone
1	0.98	0.93	0.90	0.93
2	0.95	0.88	0.87	0.84
3	0.91	0.88	0.70	0.82

Branch points	scTDA	D. Pt	SLICER	Wishbone
1	0.96	0.44	0.39	0.83
2	0.95	0.46	0.53	0.81
3	0.92	0.43	0.47	0.81

Figure: from Rizvi *et al.* Nature Biotechnology. 2016.

Where we are going now...

The Overall Problem: Combining Multiple Sets of Features

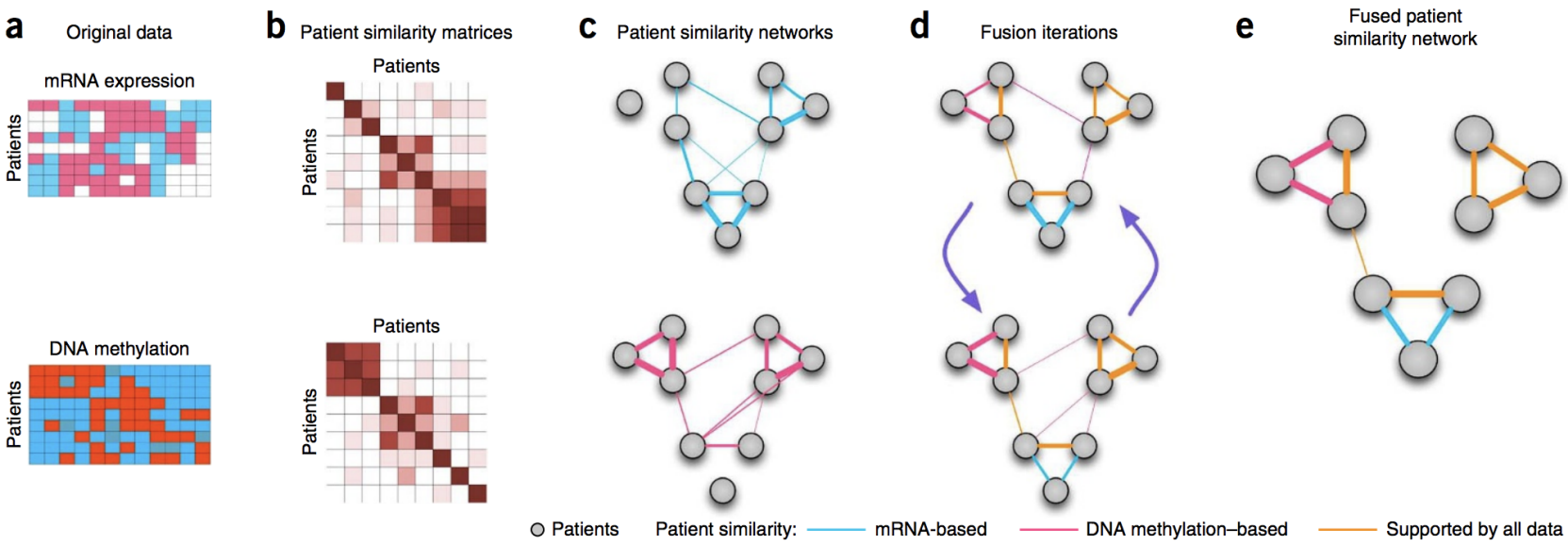


Figure: from Wang *et al.* Nature Methods 2014. The problem is to learn a joint representation of all patients that respects each modality.

The Cancer Genome Atlas (TCGA)

The focus on merging multiple datasets was inspired by The Cancer Genome Atlas, an effort to profile large patient cohorts of patients with various cancer types, with several modalities.

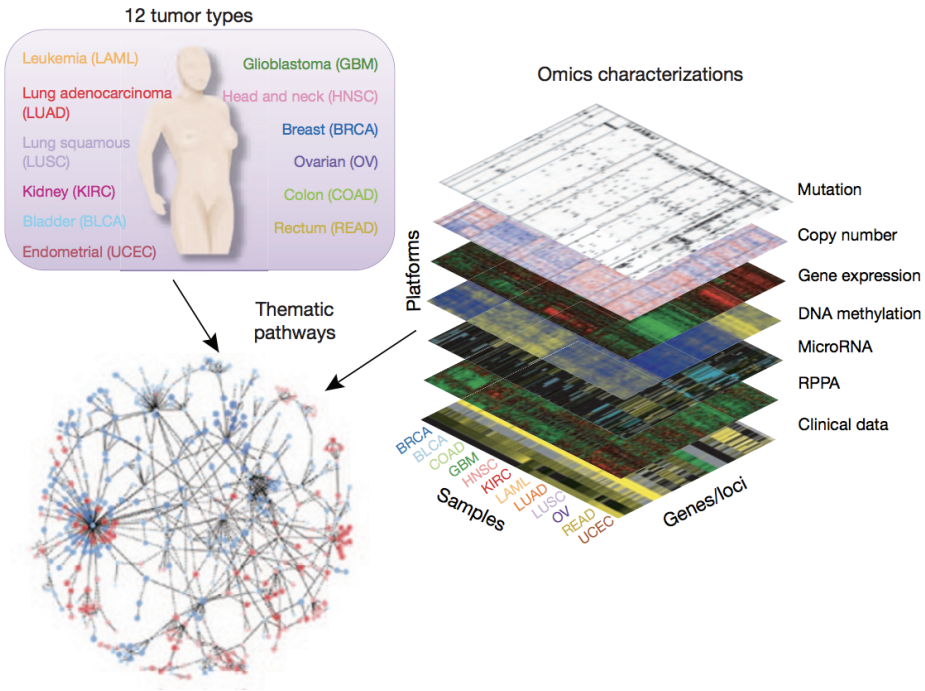


Figure: from TCGA, Nature Genetics. 2013.