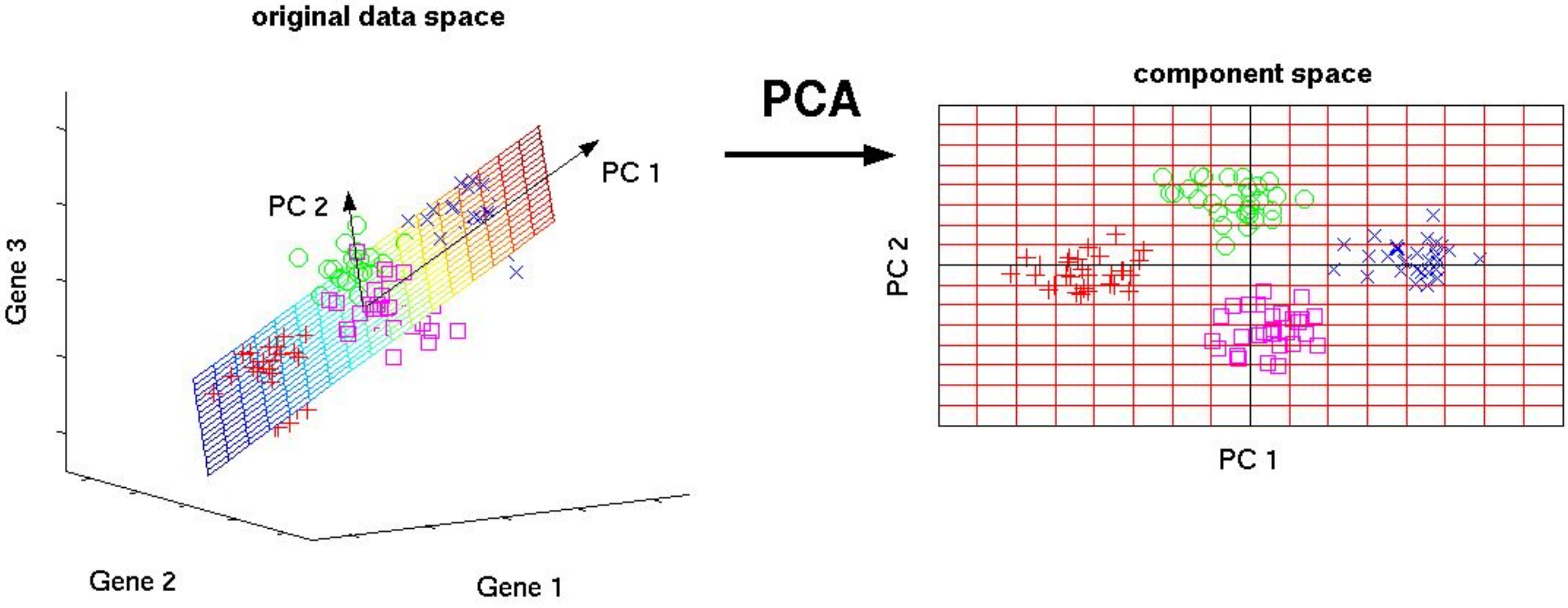
Dimensionality Reduction

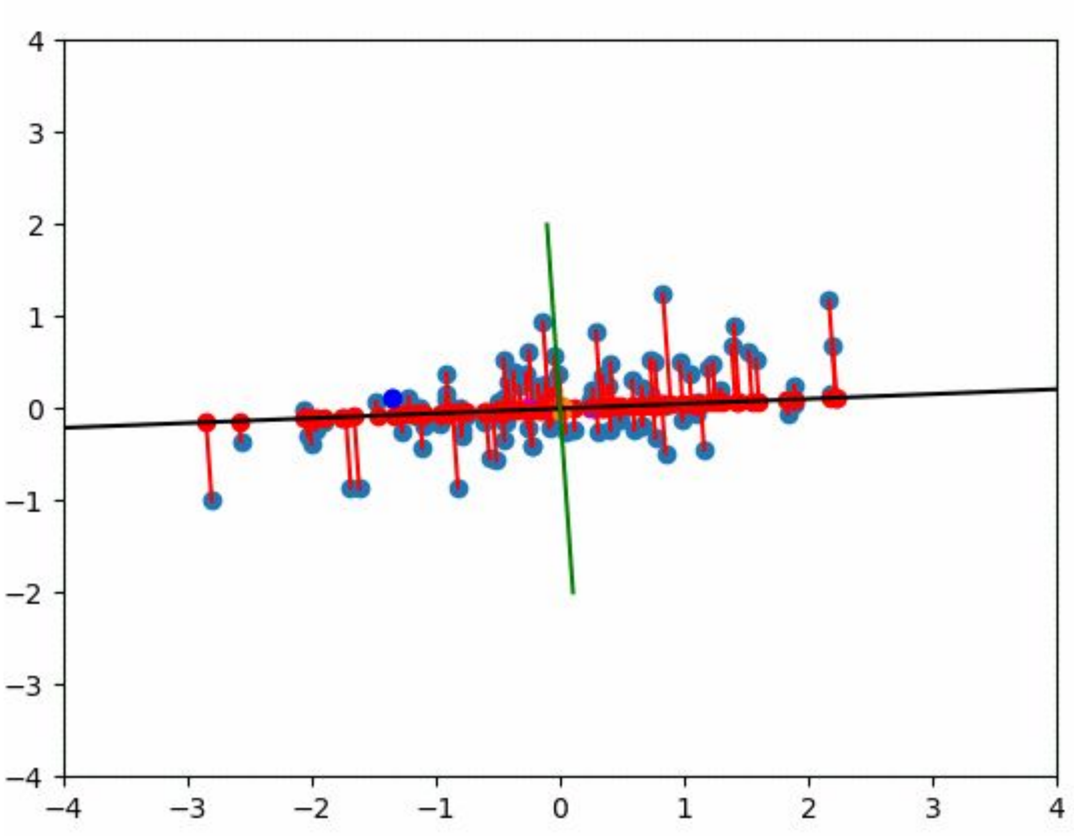
- Transcriptome data is highly multidimensional
  - Each gene's expression measurement is a separate dimension
  - Expression is often correlated among genes

- We'd like to find a representation of the expression data with fewer dimensions
  - Remove redundant information
  - Speed downstream calculations
  - Reduce "noise"
  - Allow us to make visualizations that capture the important variation in the data

Principal Components Analysis (PCA)



http://www.nlpca.org/pca_principal_component_analysis.html



https://medium.com/x8-the-ai-community/principal-component-analysis-a-brief-introduction-dc8cf3e03c71

Assumptions/Limitations of PCA

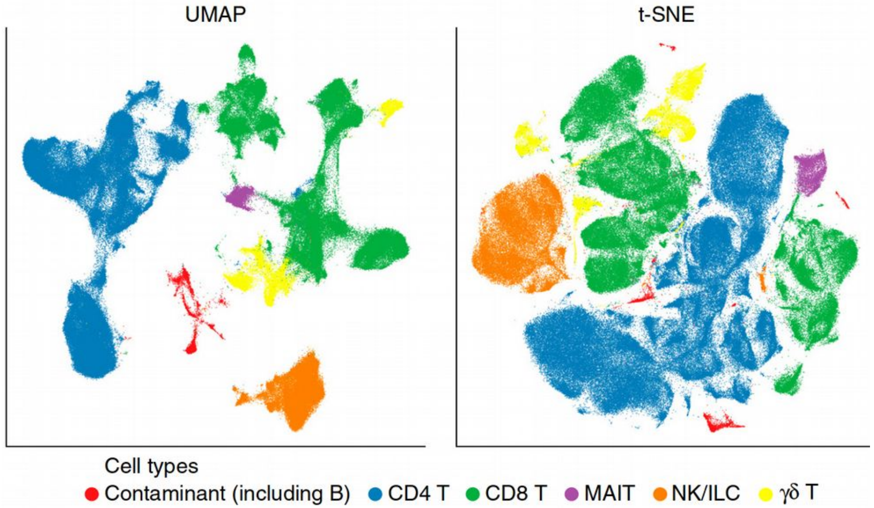
- PCA is a linear transformation of the input data
  - Fast!
  - Reversible if we keep all dimensions
  - Usually we don't keep everything... removing higher dimensions reduces effects of noise
- Assumes ~ normal distributions for error
  - For scRNA-seq count data, this can be approximated with log-scale normalization
- Sensitive to outliers

- GLM-PCA may solve many of these limitations, but is not in wide use: (Townes *et al.* 2019 https://doi.org/10.1186/s13059-019-1861-6)

UMAP and tSNE

Machine learning methods for dimensionality reduction

Details are beyond the scope of this course, but the basic steps are these:



https://doi.org/10.1038/nbt.4314

- Calculate the similarity between pairs of data points
- Find a representation in low dimensionality space (mapping) that recapitulates the similarity matrix
    - How? Start with a mapping then progressively update it by how well the distances in the low dimension space match the original distances

A nice visualization/playground for tSNE: https://distill.pub/2016/misread-tsne/

Assumptions/Limitations of UMAP & tSNE

- No assumptions about shape of data
    - Performs better when structures may not have “normal” distributions
- Tends to produce more visually distinct clustering

- Local structure is more reliable than global
- Non-reversible (can’t infer original data from mapping)
    - Don’t use the resulting coordinates for analysis!
- Can be slow
    - Common to use PCA first for partial dimension reduction, then UMAP/tSNE on that
    - UMAP is faster

To the notebooks, Batman!

Clustering Cells

Dimensionality reduction often results in visible “clusters”, but how do we define those?

Many methods!

- hierarchical clustering
    - join closest points/groups recursively
- k-means clustering
    - pick a number k, then find the “best” way to divide cells into that many groups
    - assumes clusters are “spherical”
- graph-based clustering
    - Connect cells to other cells with similar expression, then divide up the graph into clusters

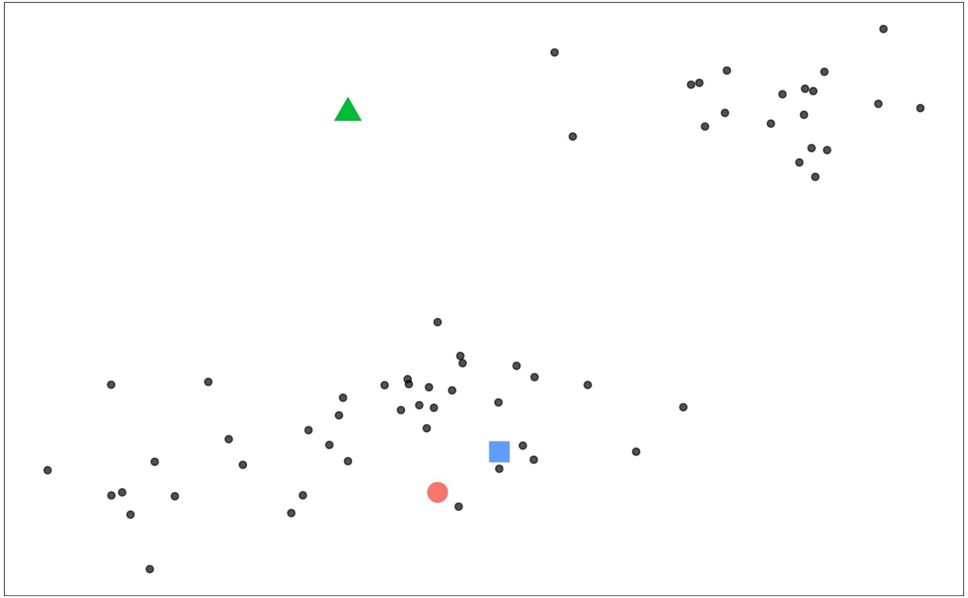
k-means clustering

Step 1: Pick **k** random centers

Step 2: Assign points to clusters by which center is closest

Step 3: Find new centers as the mean locations of all points in a cluster

Repeat Steps 2 and 3 until the clusters are stable



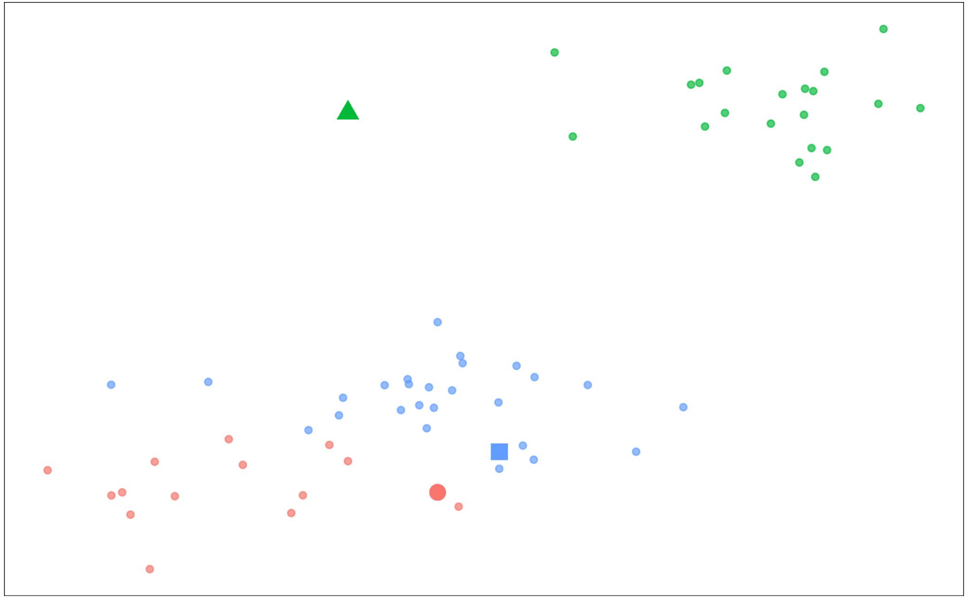
k-means clustering

Step 1: Pick **k** random centers

Step 2: Assign points to clusters by which center is closest

Step 3: Find new centers as the mean locations of all points in a cluster

Repeat Steps 2 and 3 until the clusters are stable



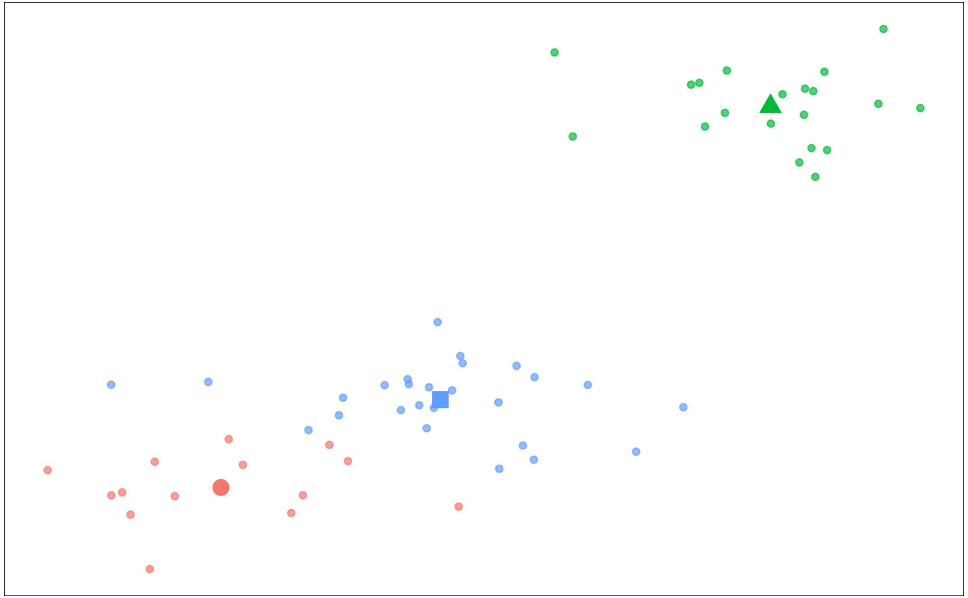
k-means clustering

Step 1: Pick **k** random centers

Step 2: Assign points to clusters by which center is closest

Step 3: Find new centers as the mean locations of all points in a cluster

Repeat Steps 2 and 3 until the clusters are stable



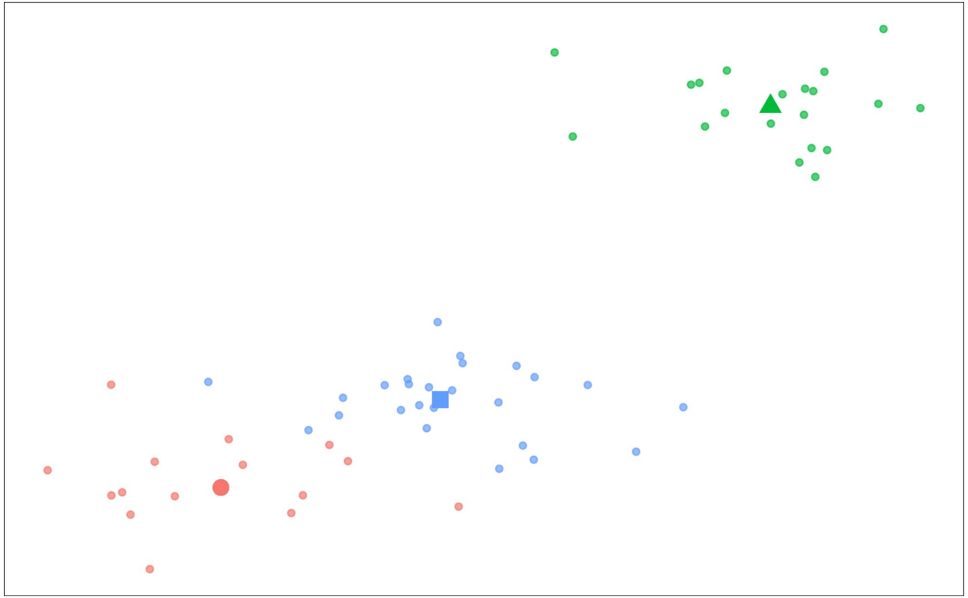
k-means clustering

Step 1: Pick **k** random centers

Step 2: Assign points to clusters by which center is closest

Step 3: Find new centers as the mean locations of all points in a cluster

Repeat Steps 2 and 3 until the clusters are stable



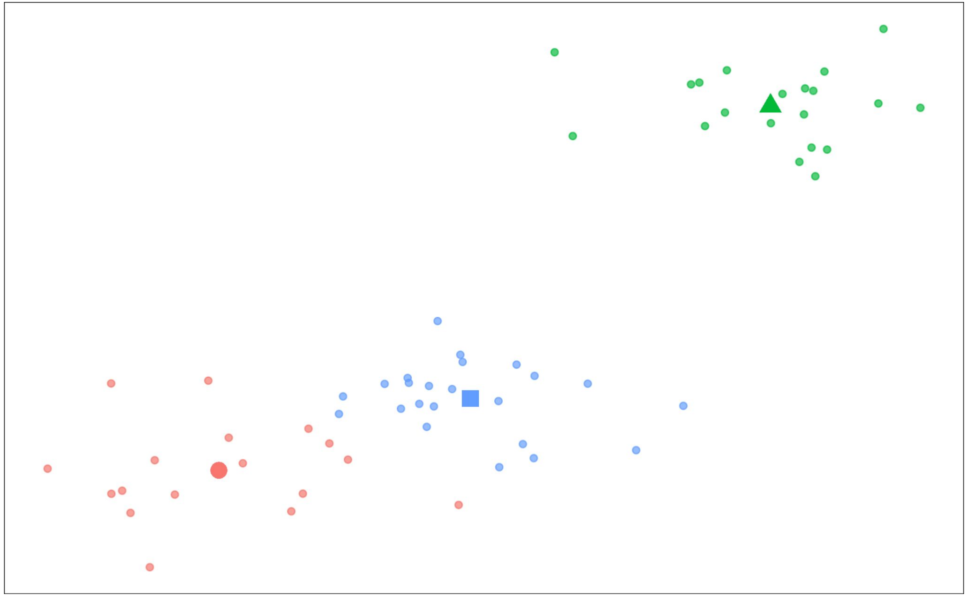
k-means clustering

Step 1: Pick **k** random centers

Step 2: Assign points to clusters by which center is closest

Step 3: Find new centers as the mean locations of all points in a cluster

Repeat Steps 2 and 3 until the clusters are stable



Graph-based Clustering

Step 1: Calculate similarity matrix among points

Step 2: Build a weighted network graph connecting points to their neighbors

Step 3: Divide network graph into “neighborhoods” based on connection patterns

Many options at each step! The algorithms can determine how many clusters to assign.

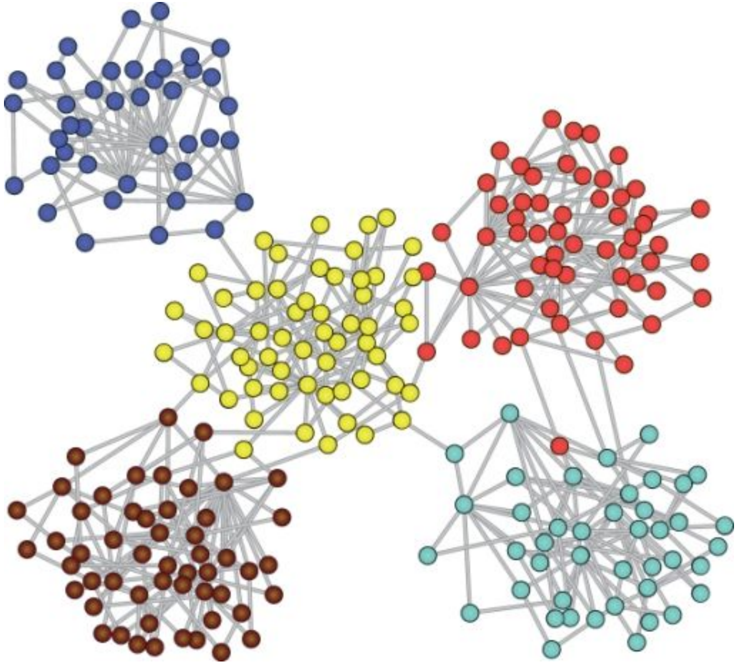


Image from:
https://github.com/benedekrozemberczki/awesome-community-detection

What do the clusters represent?

- Groups of cells with distinct gene expression patterns
- What does that mean?
  - maybe cell types?
  - sometimes cell states?
  - perhaps perturbations?
- Interpretation will vary based on the sample you are using!
  - Do not expect a simple mapping of clusters to cell types

- Clustering is usually somewhat stochastic
  - parameter choice and random seeds will affect clusters
  - Use caution when interpreting clustering results!