1 Normalization

RPKM is a very important normalization tool.
not too sure about tpm, need to double check on that

2 Single Cell

Goals:

1. cells differentiates into sub-celltypes
2. unknown celltype discovery

2.1 Dimensionality reduction

Motivation:

1. high dim data often has lower dim representation w/o much reconstruction error
2. lower dim representation can often represent info about high dim pairwise dist.

Types of dim-red:

- Global methods
  1. all pairwise dist equally impt
  2. lower dim pairwise dist fit high-dim ones
  3. often use magnitude or rank order
- Local methods
  1. only local dist reliable in high dim
  2. more weight on modelling local dist correctly

Methods:

- PCA
  - finds directions with largest variance
  - minimize squared reconstruction error
  - equiv to liner autoencoders
  - Steps of PCA
    1. $\overline{X}$: mean of all samples(usually rows), adjust $X \to X' = X - \overline{X}$
    2. covar matrix $C = X'^T X'$
    3. find eigenvectors and eigenvalues of C, ie. all pair of $\vec{v}, \lambda$ st. $C\vec{v} = \lambda\vec{v}$
    4. eigenvalues can be used to calculate percentage of total variance for each component

$$v_j = 100\frac{\lambda_j}{total \quad eigenvalue}$$

  This is non-parametric method, do not insist on a parametric encoding function

- Multi-Dimensional Scaling
  - arrange low dim points to minimize diff between pairwise distances in the high and low D space
  - a possible approach: start w a random vector, perform gradient decent
  - is there something to do with PCA? then we don't need iterative method

- Sammon (non-linear autoencoder)
  - with extra layers, much more powerful than PCA, but can be slow to converge, and can get stuck on local optima
  - Multi-Dimensional Scaling(MDS) can be made non-linear by giving higher weights to smaller distances, a popular formula is

$$cost = \sum_{ij} \left(\frac{|x_i - x_j| - |y_i - y_j|}{|x_i - x_j|} \right)^2$$

    where x is high-dim dist, and y is low-dim dist
  - still slow and get stuck on local optima

2.2 Graph-basd method

- address uniform circularity
- Isomap is a dim-red technique based on graphs
  - each datapoint is connected to k nearest neighbor in high-dim
  - edge weights = euclidean dist
  - approx of distance = shortest path in contracted graph
- Probabilistic local MDS
  - local distances are more impt than non-local ones
  - in this way all local distances are given equal importance
- stochastic neighbor embedding(SNE) has a probabilistic way to decide if a distance is local
  - convert global distances into probability of one datapoint picking another datapoint as its neighbor (what defines a neighbor tho) - still about isomaps?
  - each point in high-dim has a conditional probability of picking any other point as its neighbor
  - distribution (some sort of Gaussian) is over high-dim distances (if high-dim coords unavailable, a similarity / dissimilarity matrix may be used)
    $p_{j|i}$ is the prob. of picking j given starting at i in high-dim.

$$P_{j|i} = \frac{e^{-2d_{ij}^2/2\sigma_i^2}}{\sum_k e^{-d_{ik}^2/2\sigma_i^2}}$$

  - having the probabilities potentially allow us to throw away the raw high-dimensional data
  - evaluation done using pairwise distance in low dimensional map (shows how well the lower dim representation models high-dim data ig)
    $q_{j|i}$ is the prob of picking j given starting at i in low-dim
  - compute the Kullback-Leibler divergence between prob. in the high-dim and low-dim spaces (why not just use dist in high dim?) - more space / time efficient
  - nearby pts in high-dim should be close in low-dim
- picking σ used to compute p - the radius of the gaussian
  - different radius is needed in different parts of the space to keep the no. of neighbors constant
  - big radius → high entropy for distribution over i's neighbors
  - small radius → low entropy
- Symmetric SNE

  - simpler than stochastic
  - works best if different procedures are used for computing p's and q's.
  - compromise: no longer guarantees that if using same dimension will produce optimal solution
  - turn conditional prob into symmetric pairwise probabilities
- Optimization methods for SNE

2.3 Supervised dim-red: Neural networks

- last few layers have much fewer values than inputs
- use intermediate layers as lower-dim representations
- can easily add prior biological knowledge, such as protein interactions or transcription factors
- essentially, some nodes in the hidden layer are same as before, others are based on biological info

2.3.1 Additional NN architecture: Siamese

- supervised, but not trying to maximize training accuracy
- input: whether each pair is similar
- output: binary label of similar / not similar
- thus directly optimize dim-red layer for KNN

3 ChIP-Seq

4 cis-regulatory motif