Single-Cell RNA-seq Analysis
Methods Overview

Horacio Gómez-Acevedo
Department of Biomedical Informatics
University of Arkansas for Medical Sciences

August 16, 2021

UAMS

sc-RNA seq

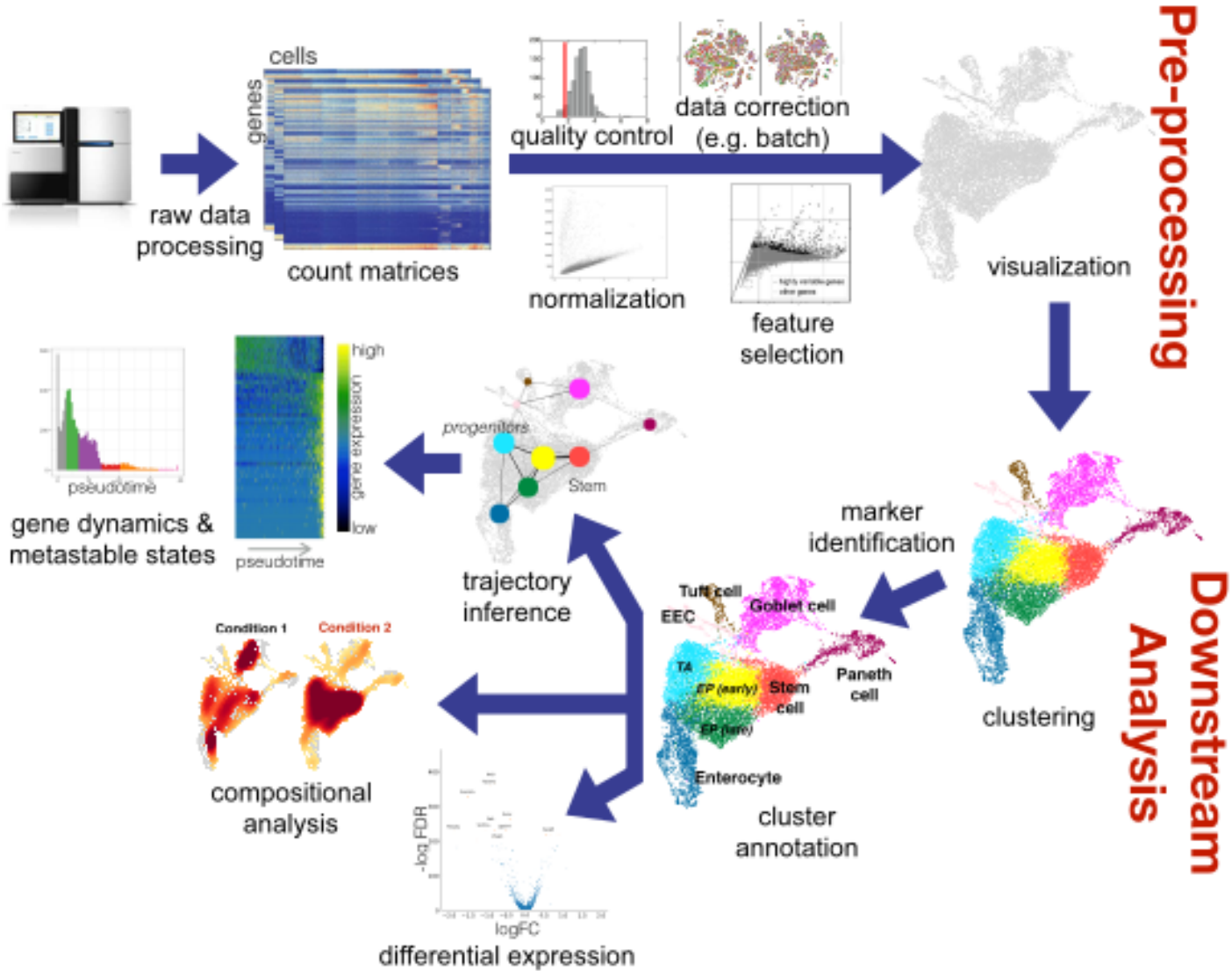
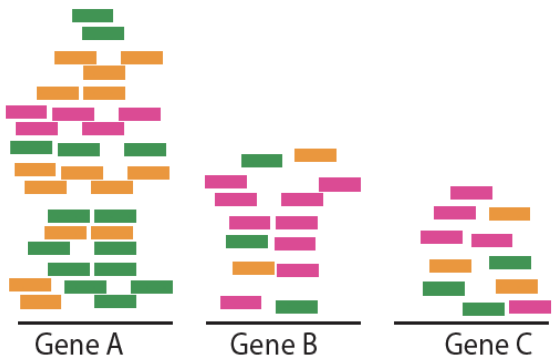
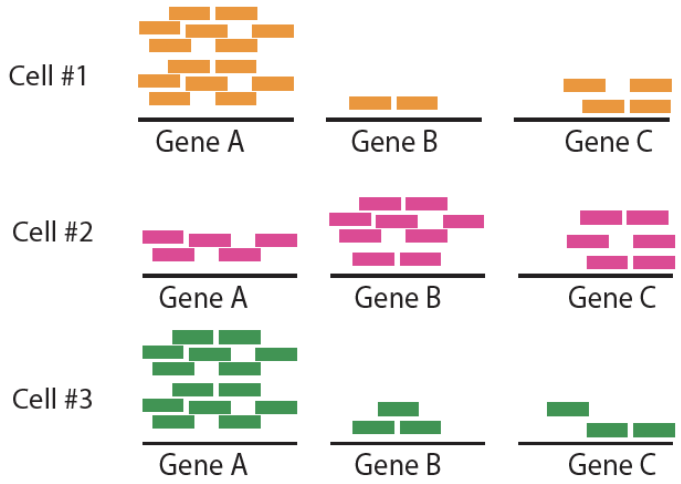


Figure: Overview

Bulk vs Single-cell RNAseq



(a) Bulk RNAseq



(b) Sc-RNAseq

Space: the final frontier

The mathematical representation of the data is given by placing the readings (RPKM) of each individual gene in an "axis" of an n dimensional space. In this case n represents a large number of genes.

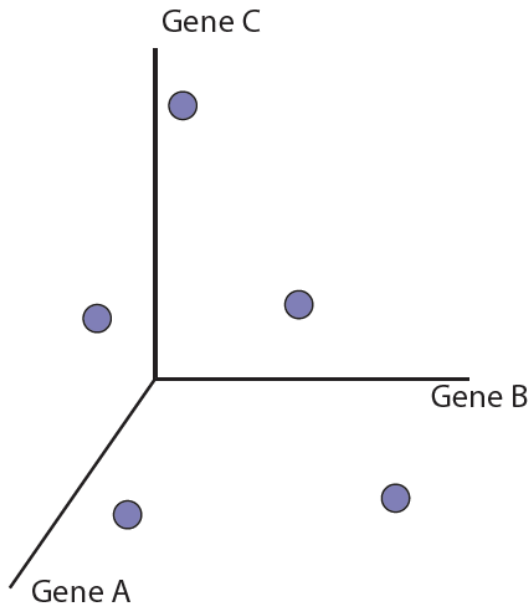


Figure: Spatial representation of scRNAseq data

PCA interlude

Principal Component Analysis is one of the most commonly used methodologies to "inspect" high dimensional data.

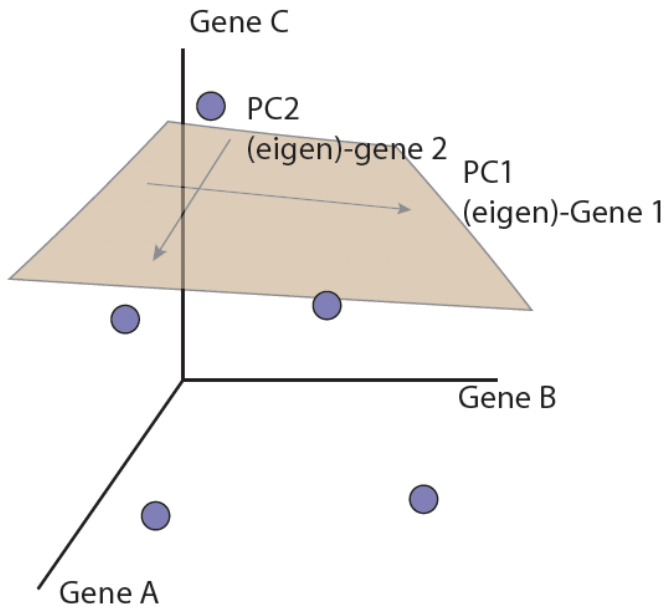


Figure: Plane projection of data

PCA projection

The main goal of PCA is to find the representation of our data in a lower dimensional space (mostly a plane). But the selection of such plane should preserve original data variability.
Formally, the two directions represent the directions of the maximal variability of our data.

RGB example

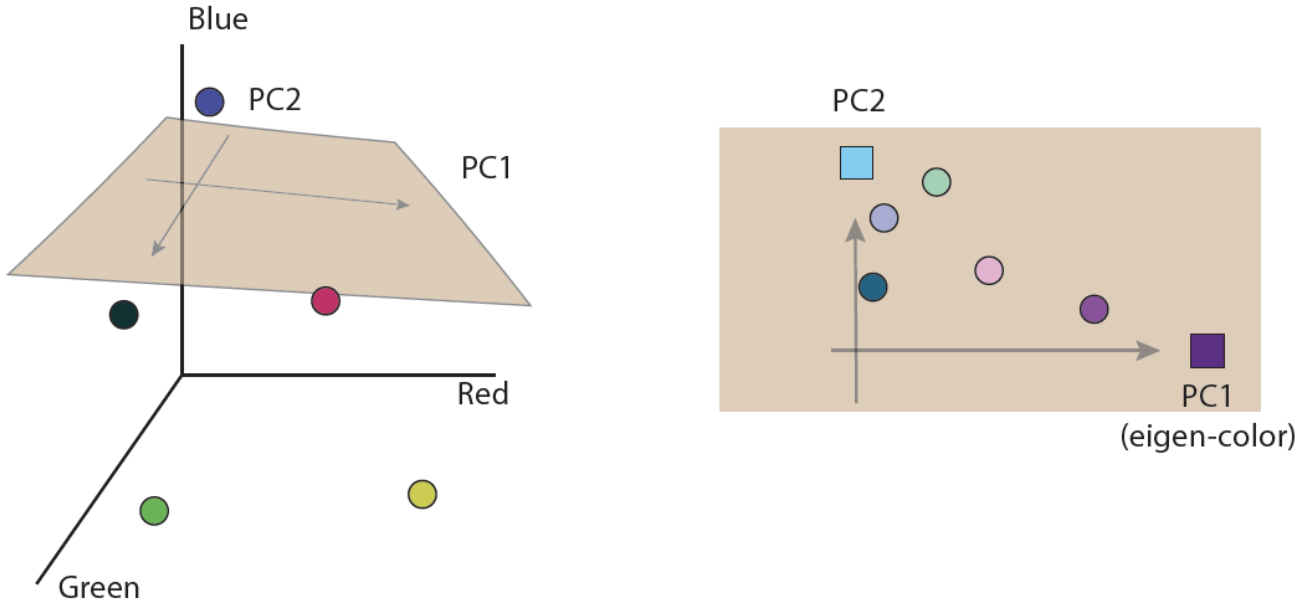


Figure: Plane projection of RGB data

Clustering

Clustering refers to a set of computational techniques for finding subsets or *clusters* in a data set. Clustering is among the so-called **unsupervised learning methodologies** .

$$\text{Unsupervised} \neq \text{Automatic}$$

Thus, the main goal of clustering is to find homogeneous subgroups among the data.

Clustering terminology

Let's define a distance between two points (in a plane), say d.
Also, the *centroid* of a cluster is the mean of their observations.

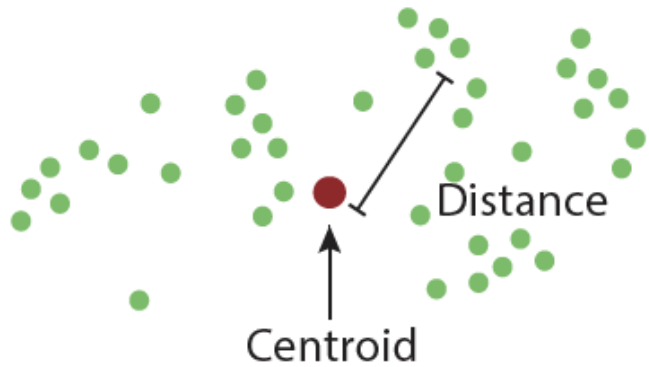


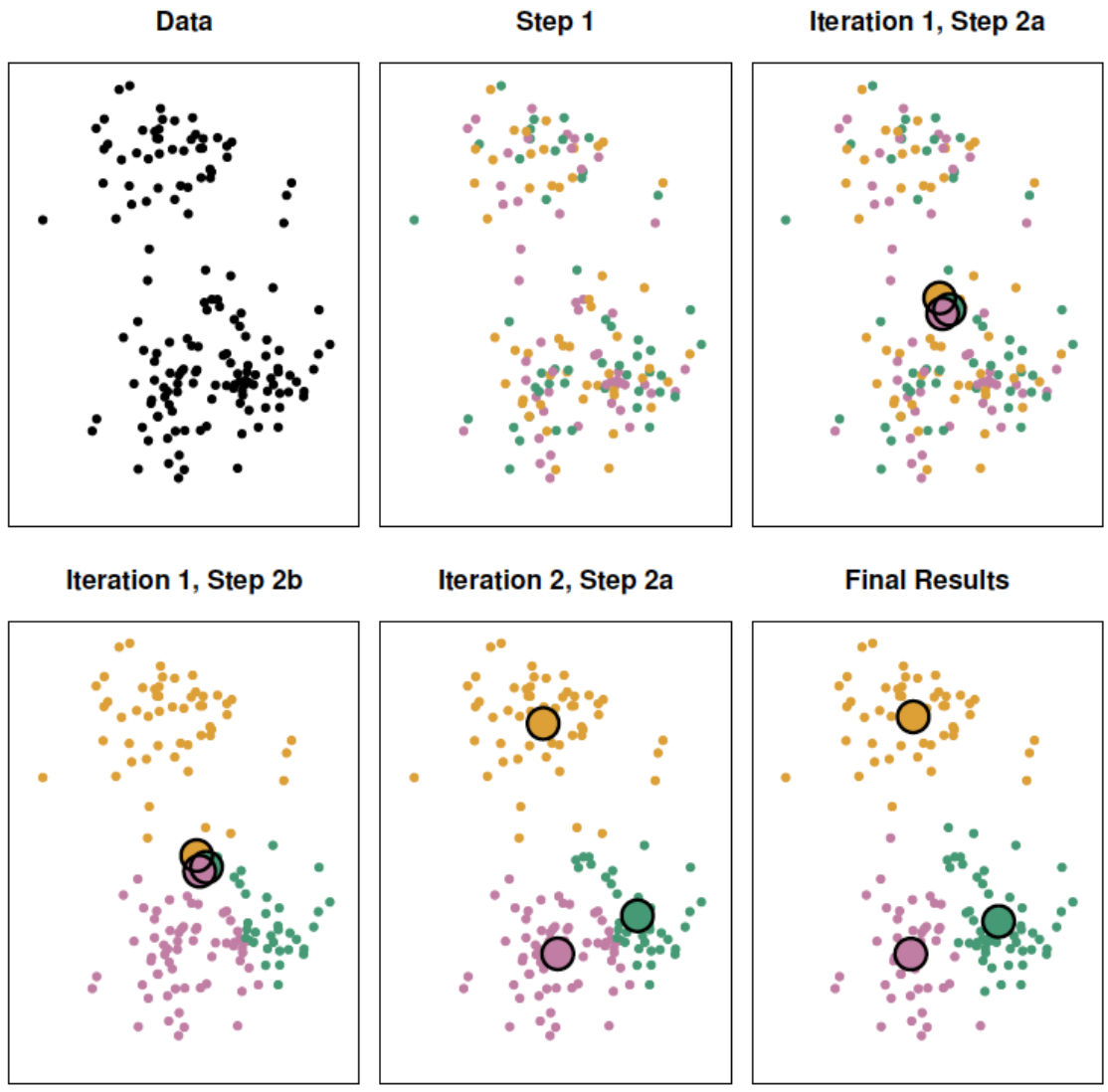
Figure: Distance and Centroid of a cluster

K-means clustering

One of the simplest methods for clustering is K-means clustering. We begin with a data set and a value K fixed by a human (say $K = 3$).

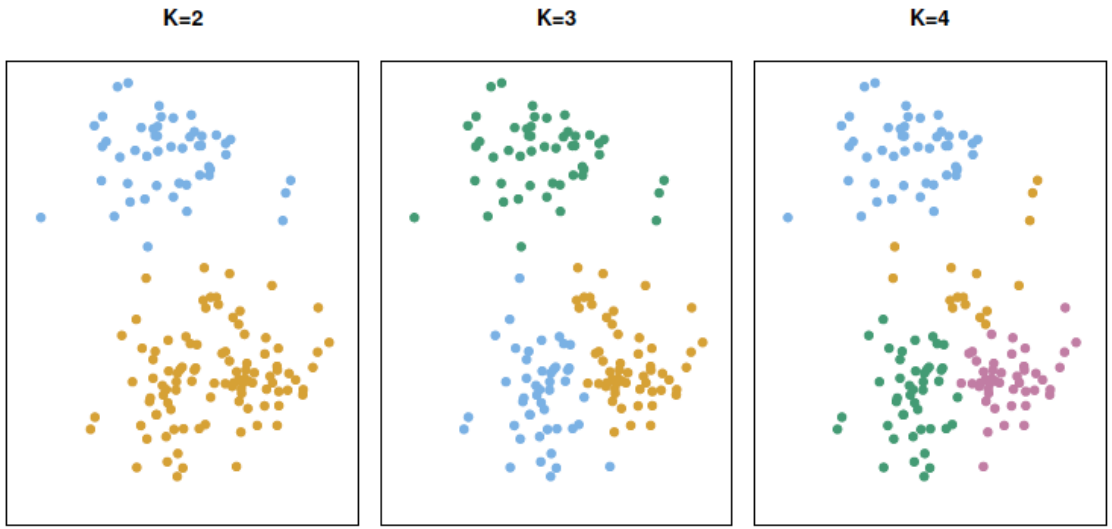
1. Assign randomly a value between 1 and K to each data point.
2. Iterate the following procedure until the clusters assignments stop changing
   - 2.1 Find the centroid for each of the K clusters.
   - 2.2 Each point will be assigned to the cluster K whose distance is the smallest. If two or more are equidistant, select randomly the cluster among the equidistant clusters.

K-means clustering picture



Problems with K-means clustering

- Selection of the distance function d (Euclidean, Pearson correlation, arccos, etc.)
- Selection of K.



An Introduction to Statistical Learning (Chapter 10)

t Stochastic Neighbor Embedding

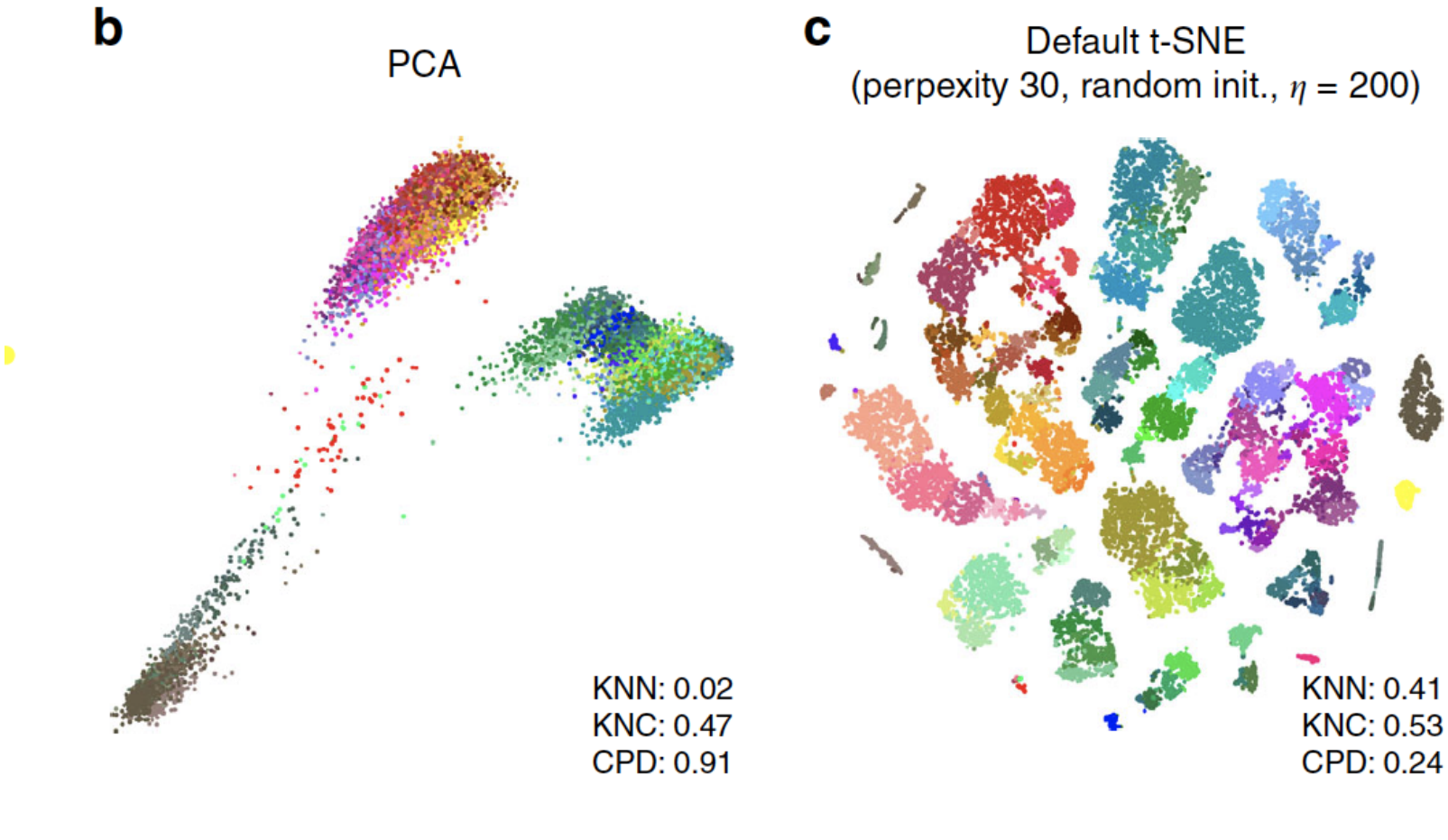
t-SNE maps a set of high-dimensional points to a plane, such that ideally, close neighbors remain close and distant points remain distant.
Informally, the algorithm places all points on the 2D plane, initially at random positions, and lets them interact as if they were physical particles.
The interaction is governed by two laws:

- all points are repelled from each other
- each point is attracted to its nearest neighbors

This methodology is governed by a parameter called **perplexity**.

tSNE vs PCA

A visual comparison of PCA vs tSNE



Out-of-Bag Observations

The main idea of the bootstrap is that from m observations, we select a sample with replacement m observations.
What is the probability of **not** selecting sample 1 ?
The probability of picking sample different from 1 would be $(1 - \frac{1}{m})$. Since we are repeating the experiment with replacement, the probability that a bootstrap sample does not contain sample 1 is

$$\left(1 - \frac{1}{m}\right) \cdot \left(1 - \frac{1}{m}\right) \cdots \left(1 - \frac{1}{m}\right) = \left(1 - \frac{1}{m}\right)^m$$

A little bit of calculus shows that

$$\lim_{m \to \infty} \left(1 - \frac{1}{m}\right)^m = \exp(-1) \approx 36.79\%$$

Thus, bootstrapping will not touch about $1/3$ of the observations! and those observations are referred to as **Out-of-Bag (OOB)**.

OOB Error Estimation

We can exploit the OOB observations to estimate the test error in the bagging process without the need of cross-validation or even a split of the data in training and testing.
Once we have obtained our $\hat{f}_{bag}(x)$, we can use the OOB observations (i.e., observations not used for the bagging estimation) to determine predictions.
More precisely, we obtain $\hat{f}^i_{\text{oob}}(x)$ based on the OOB observations $\{x_{i1}, \ldots, x_{iK}\}$, where $K \approx B/3$ for B big enough.

$$\hat{f}_{OOB}(x) = \frac{1}{K} \sum_{i=1}^{K} \hat{f}^i_{\text{oob}}(x)$$

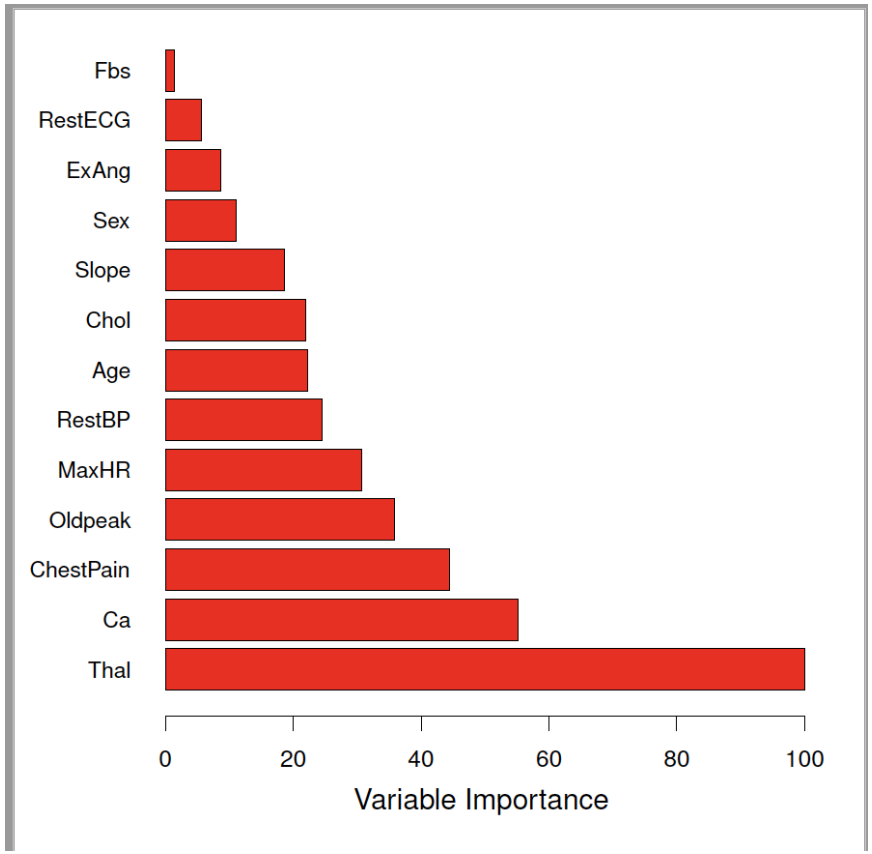
This procedure leads to the calculation of the test MSE that is a valid estimate since the response is derived from trees that were not involved in the bagged model.
A similar expression is valid for classification, but instead of the average we can use the majority vote and purity metrics instead of MSE or RSS.

Variable Importance Measures

We know that bagging improves the accuracy in our predictions, at the expense of making our models harder to interpret.
For bagging regression trees, we use the **Variable Importance Measure (VIM)** that is defined as the total amount that the RSS is decreased due to splits over the given predictor, averaged over all B trees. The larger the VIM, the more "relevant" is that predictor.
For bagging classification trees, we can define VIM as the total amount that the Gini index (or cross-entropy) is decreased by splits over a given predictor, averaged over all B trees.

VIM example

The `Heart` data set VIM plot with a mean decrease of GIni index and normalized VIM is shown below.

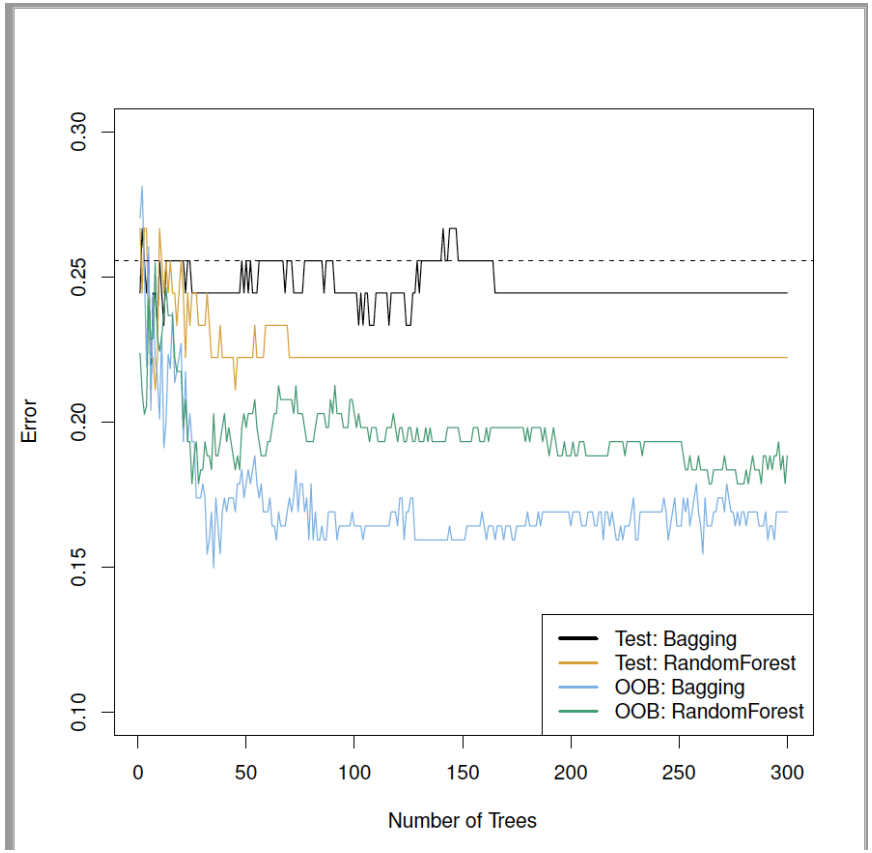


Random Forest

It follows similar rationale as in bagging but with an interesting random twist.
We build a number of decision trees on bootstrapped training samples. But when building these decision trees, each time a split in a tree is considered, a *random sample of m predictors* is chosen as split candidates from the full set of p predictors. We normally set $m \approx \sqrt{p}$.
What is the advantage of random forest over bagging?
When we have a strong predictor, bagging trees will consider that predictor frequently, thus bagging trees will look alike. By having a random choice on the predictors, we may generate "different" trees that otherwise we would not have explored. This process is referred to as *decorrelating trees*.

Random Forest vs Bagging

The test errors from the `Heart` data are depicted below



Boosting

Boosting is another general methodology to improve the predictions from a decision tree. In this case trees are grown sequentially as they gather information from previously generated trees.
Boosting does not require bootstrap sampling as each tree is fit on a modified version of the original data set.

Boosting Algorithm

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.
2. for $b = 1, \ldots, B$ repeat:
   2.1 Fit a tree $\hat{f}^b$ with d splits to the training data (X, r).
   2.2 Update $\hat{f}$ by adding in a shrunken version of the new tree:

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$$

   2.3 Update the residuals,

$$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i)$$

3. Output the boosted model

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x) \tag{1}$$

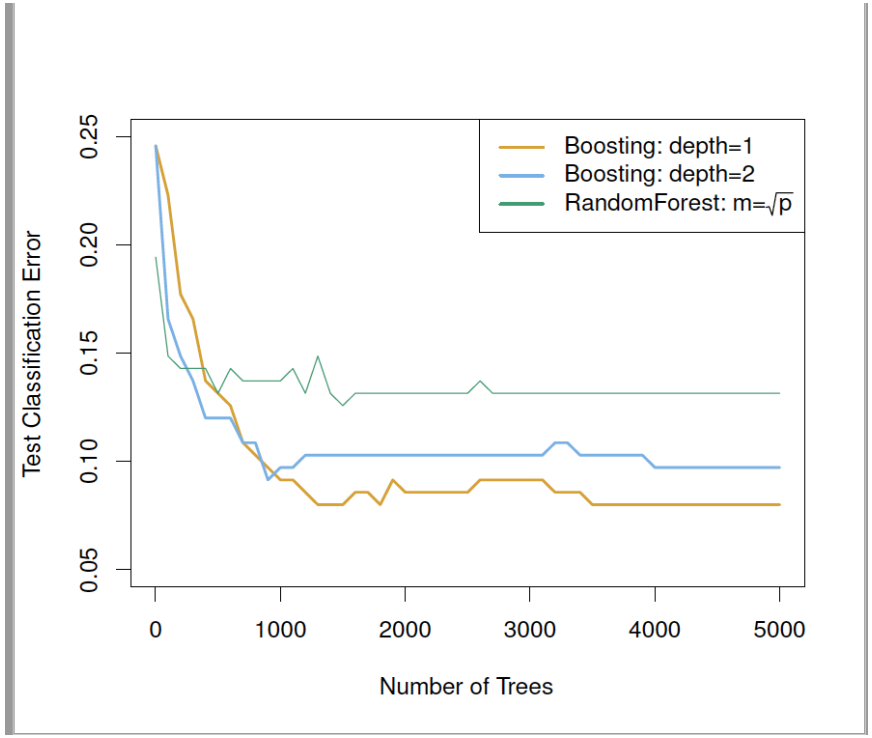
Boosting

Given a current model, we fit a decision tree to the residuals from the model rather than the outcome Y as the response. And we add these residuals to a new decision tree and update again the residuals.
Boosting has the following tuning parameters:

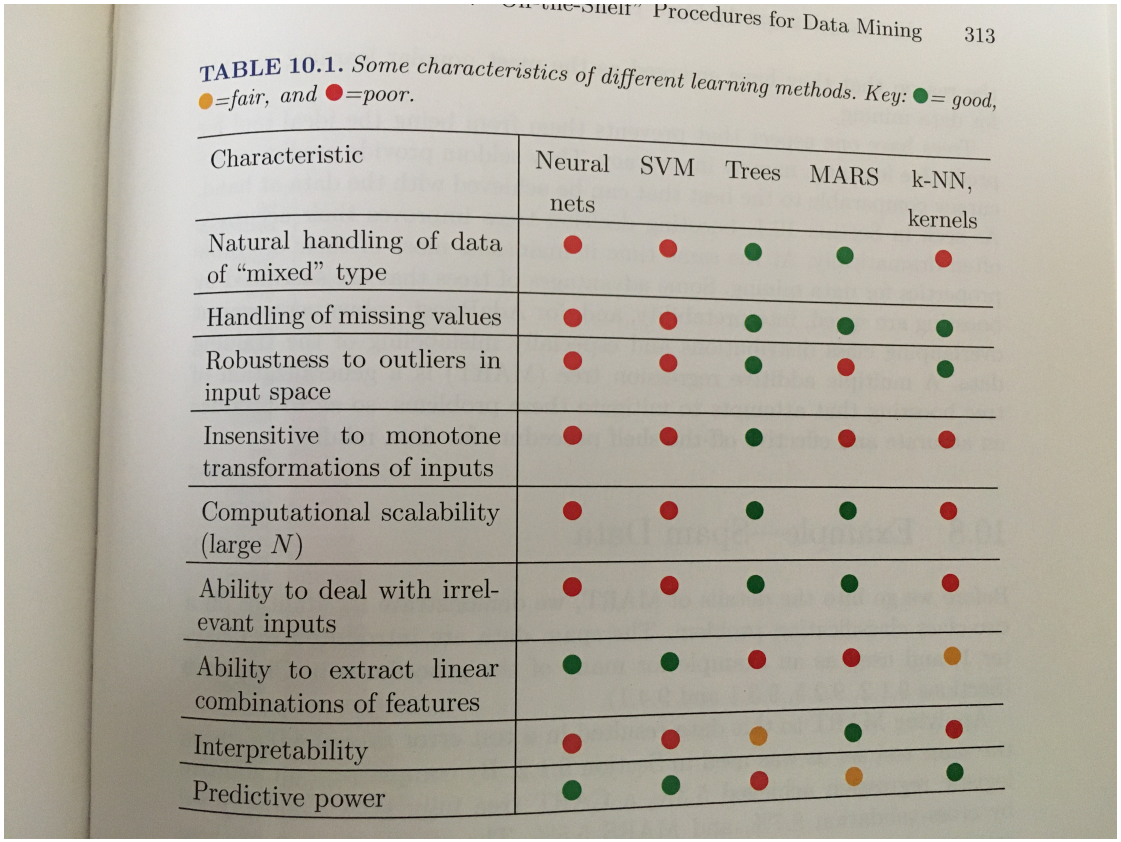
- B that represents the number of trees. Do not take very large values of B as boosting tends to overfit. We can use cross-validation to determine a good candidate for B.
- The shrinkage parameter λ controls the learning rate.
- The number of splits d controls the complexity of the boosted ensemble. Sometimes $d = 1$ works well.

Boosting

Boosting and random forest comparison in a 15-class gene expression data set to predict cancer.



Final Thoughts



...the-Shelf" Procedures for Data Mining 313

TABLE 10.1. *Some characteristics of different learning methods. Key:* ● *= good,* ● *=fair, and* ● *=poor.*

Characteristic	Neural nets	SVM	Trees	MARS	k-NN, kernels
Natural handling of data of "mixed" type	●	●	●	●	●
Handling of missing values	●	●	●	●	●
Robustness to outliers in input space	●	●	●	●	●
Insensitive to monotone transformations of inputs	●	●	●	●	●
Computational scalability (large N)	●	●	●	●	●
Ability to deal with irrelevant inputs	●	●	●	●	●
Ability to extract linear combinations of features	●	●	●	●	●
Interpretability	●	●	●	●	●
Predictive power	●	●	●	●	●

References

Materials and some of the pictures are from (1),(2), and (3).

1. Gareth James et al. *An Introduction to Statistical Learning with applications in R*. Springer (2015)
2. Trevor Hastie et al. *The Elements of Statistical Learning* Springer (2001).
3. Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn & TensorFlow* O'Relly (2017)

I have used some of the graphs by hacking TiKz code from StakExchange, Inkscape for more aesthetic plots and other old tricks of TEX