Fast Alignment Heuristics

COMP462/561: Computational Biology Methods

Fall 2016

M & W: 10:00 am – 11:30 am

*Based on Course Notes by Dr. Mathieu Blanchette

Reminder!

- Office Hours:

  David Becerra – Thursday's 11:30am-1:00 pm (Trottier 3110)
  Mathieu Blanchette – Monday's 11:30am-1:00 pm (Trottier 3107)
  Christopher Cameron – Friday's 10:00-11:30am (Trottier 3110)

- Assignment #1 will be out this week
  - Don't wait until the last minute to get help…

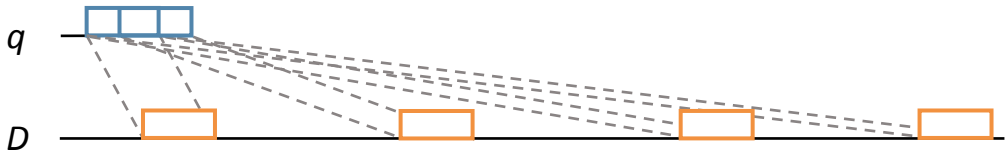
2

Motivation

Problem:

Given a query sequence, *q*, of length *m* (small, ~1000 nucleotides) and a large database (target), *D*, of size *n* (billions of nucleotides)

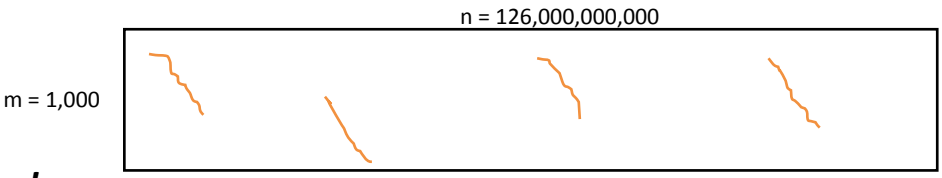
Find all local alignments of *q* within *D* that have a score above threshold, *T*



3

Smith-Waterman?

- SW is **too slow…**would take O(mn + m*hits)
- How?
  - Trace back all entries of a dynamic programming matrix with a score>*T*



- ***Too slow!***
  - For example, NCBI is a database containing 1000's of genes
    - NCBI webserver needs the ability to satisfy many queries per second

4

Smith-Waterman: Local Alignment (1981)

Problem:

Given two sequences, A and B, of lengths *m* and *n*, find the optimum alignment of all possible lengths.

Steps:

1. Initialization matrix
2. Fill matrix with appropriate alignment scores
3. Trace back from highest scoring cell(s) to find best alignment(s)

5

SW Initialization

A = GCTGAATCAT

B = GCTTAC

For two sequences, A and B, a **pair-wise matrix**, H, is built such that:

$H(i,0)=0,\ 0\le i\le m$

$H(0,j)=0,\ 0\le i\le n$

		C	G	T	G	A	A	T	T	C	A	T
-	0	0	0	0	0	0	0	0	0	0	0	0
G	0											
C	0											
T	0											
T	0											
A	0											
C	0											

6

SW Matrix Filling

Similar to **Needleman-Wunsch (NW)**, fill in the matrix such that:

$$H(i,j)=max\{\blacksquare 0\&\&@H(i-1,j-1)+s(a\downarrow i,b\downarrow j)\&Match|Mismatch@max\downarrow k\geq 1\{H(i-k,j)+g\}\&Deletion@max\downarrow l>1\{H(i,j-$$

	-	C	G	T	G	A	A	T	T	C	A	T
-	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	2	1	2	1	0	0	0	0	0	0
C	0	2	1	0	0	1	0	0	0	2	1	0
T	0	1	0	3	2	1	0	2	1	1	1	3
T	0	0	0	2	2	1	0	2	4	3	2	2
A	0	0	0	1	1	4	3	2	3	3	5	4
C	0	2	1	0	0	3	3	2	2	5	4	4

With a match score of +2 and a mismatch & indel score equal to -1.

SW Trace Back

- With NW, we trace back from the bottom right-most cell of the matrix
  - Slightly different with SW. **How?**

Local Alignment #1

Local Alignment #2

	-	C	G	T	G	A	A	T	T	C	A	T
-	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	2	1	2	1	0	0	0	0	0	0
C	0	2	1	0	0	1	0	0	0	2	1	0
T	0	1	0	3	2	1	0	2	1	1	1	3
T	0	0	0	2	2	1	0	2	4	3	2	2
A	0	0	0	1	1	4	3	2	3	3	5	4
C	0	2	1	0	0	3	3	2	2	5	4	4

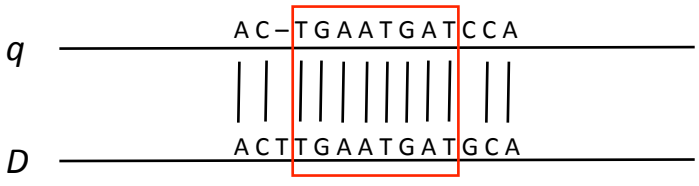
Basic Local Alignment Search Tool Idea

- Give up on (guaranteed) optimality
  - *Heuristic* approach
  - Search only for local-alignments with **high-scoring gapless alignments (HSPs)**
- Pre-process the database, *D*, so that queries can be answered in constant time with respect to *n*
- **BLAST** was published in 1990
  - cited by more than 10^5 papers

9

Gapless Alignments

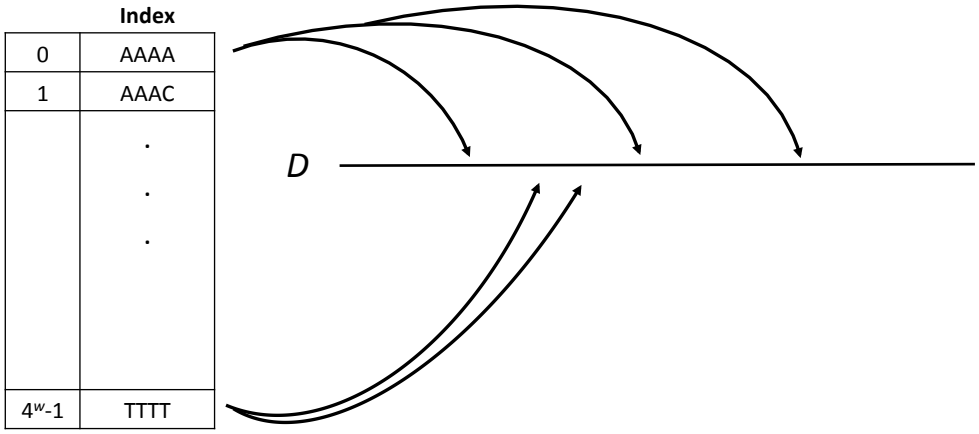
- If *q* has a good alignment, *X*, somewhere in *D*



- Then *X* is likely to contain a HSP

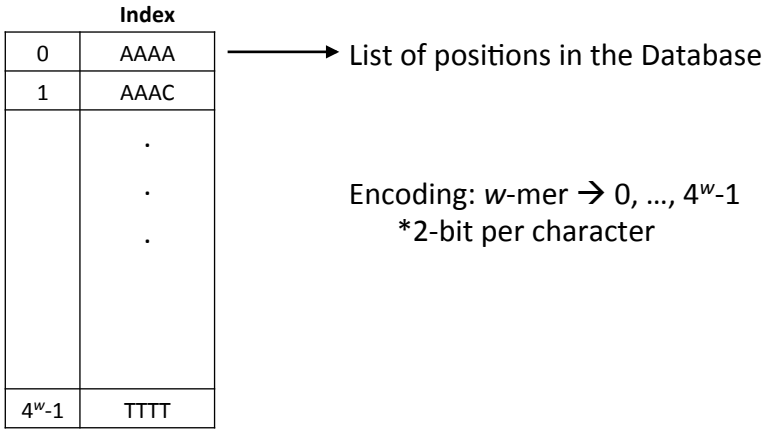
10

Preprocess Database to Build Indices



11

Indexing the Database



Encoding: *w*-mer → 0, …, 4^w-1
*2-bit per character

12

Scanning for Hits in *D*

- Given a query, *q*

```
For each w-mer in q     O(|q|)
    Find index of D     O(w)
    Consider all hits
```

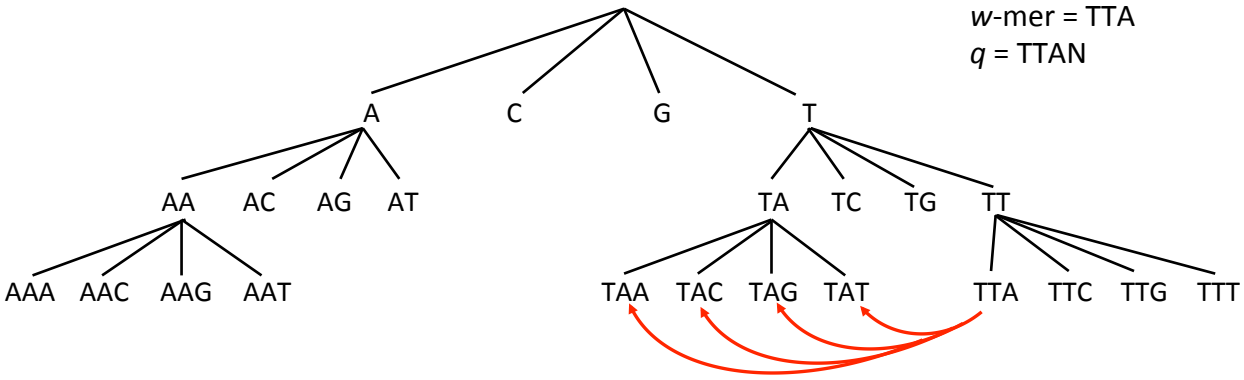
How many hits do we expect for a *w*-mer of size 11?

$$3 \times 10^9 / 4^{11} = 1000$$

13

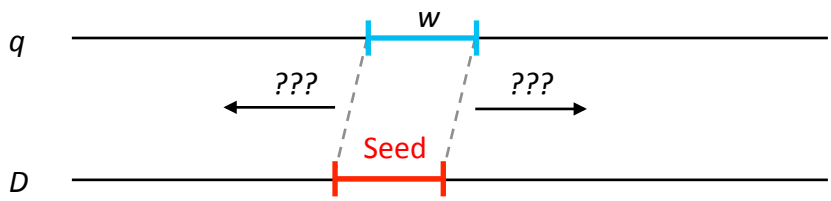
Improving Scan Times

- Encode database indices in a **trie**



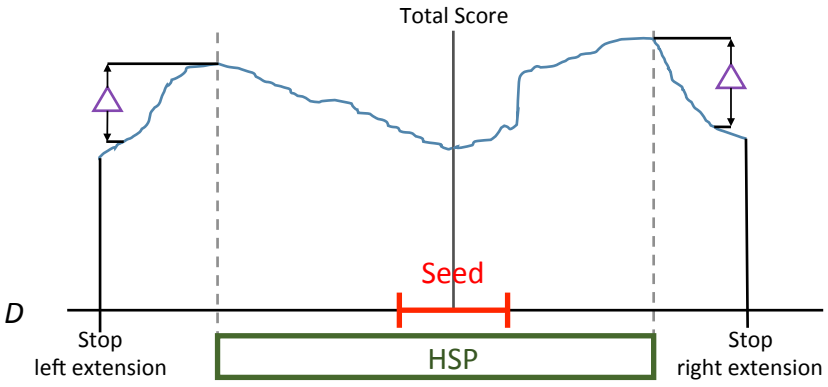
14

Extending hits to find HSP



15

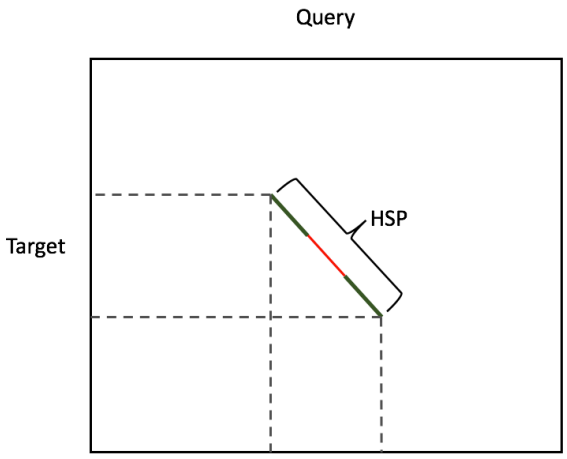
Ungapped Extension Phase



Time? Linear in size of extension

16

BLAST HSP

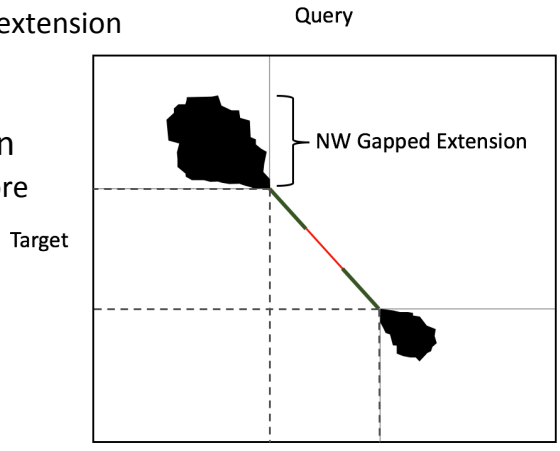


How can we improve the hit further?

17

Gapped Extension

- If the HSP's alignment score is greater than some threshold, *T*
  - Do a more expensive gapped extension
    - Using NW
- Perform NW in each direction
  - Consider only entries with score greater than "best so far"



18

Statistics of Local Alignments

- Even if *D* was completely random, we would expect to observe some pretty high scoring HSPs
  - How do we know when we should get excited?

- **E-value** (score(HSP))
  - The expected number of local alignments with a score greater or equal to HSP's that would be found in a random *D*

19

Karlin-Attschul (1990)

$$E(S) = Kmne^{-\lambda S}$$

- *S* is the score of the ungapped HSP alignment
- *K* and λ depend on the scoring scheme and background probabilities
  - λ scales scores scheme

- A low E-value (10^{-1} - 10^{-100}) is a good match
  - Low chance of observing HSP given random chance alone

20

Choosing the *w* Size

Small (<=11)

- High probability of finding exact *w*-mer in HSP
- Lots of false positive seeds
- High sensitivity
- Slow

Large (>12)

- Miss many HSPs
- Few false positives
- Low sensitivity
- Fast

21

Variants

For proteins: inexact matches are considered

- Based on a **point accepted matrix (PAM)**

Query	Target	BLAST variant
DNA	DNA	blastn
Protein	Protein	blastp
Protein	DNA	tblastn
DNA	protein	blastx

22

Optimizations

- **Dealing with repeats in q or D**
- **Two-Hit method**
  - Lower *T* to allow more hits, but only extend if two hits fall on the same diagonal
    - Within a window of fixed length
    - *Increases* hits and *lowers* extensions
- **Gapped seeds**

23

Upcoming Topics

- Wednesday – **multiple sequence alignment (MSA)**
  - Dr. Blanchette will return!
- End of the semester – **Burrows-Wheeler Transform (BWT)**
  - https://en.wikipedia.org/wiki/Burrows%E2%80%93Wheeler_transform
  - In pattern matching: https://www.youtube.com/watch?v=z5EDLODQPtg

24

Clustering

COMP462/561: Computational Biology Methods

Fall 2016

M & W: 10:00 am – 11:30 am

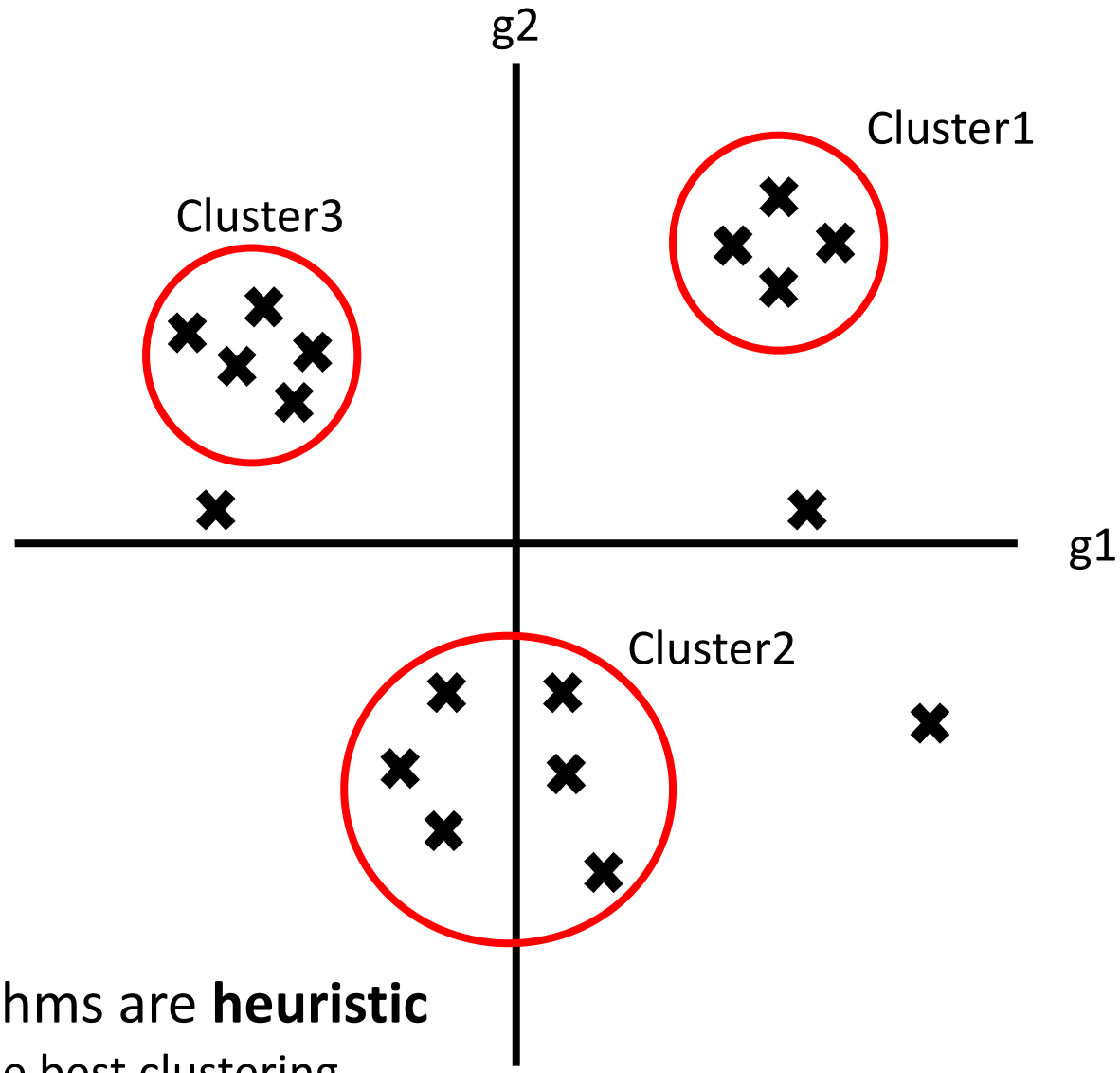
*Based on Course Notes by Dr. Mathieu Blanchette

Motivation

Given: A collection of unlabeled samples $X_i \ldots X_n$, where X_i represents the data for sample i

Goal: Partition samples into groups that are similar within themselves but dissimilar between

	X_1	…	X_n
gene1			
gene2			
gene3			
…			
$gene_{k-1}$			
$gene_k$			

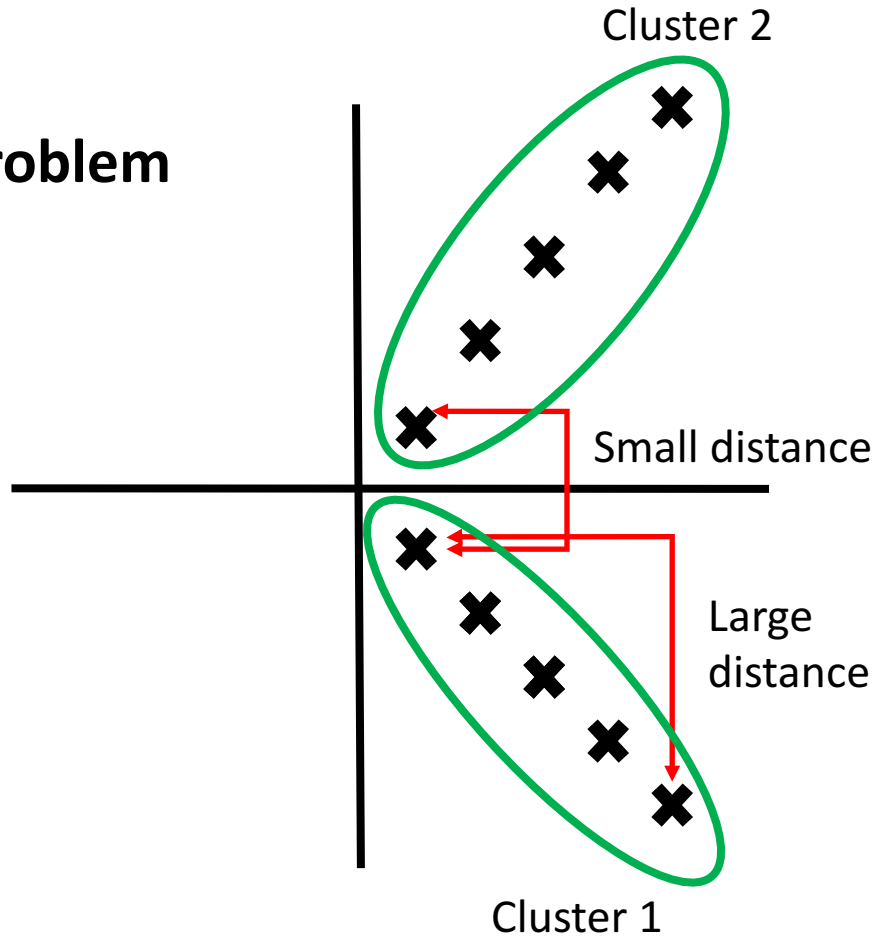


- All the clustering algorithms are **heuristic**
  - They don’t guarantee the best clustering

Similarity (or Distance) Measures

Given: Two expression profiles, X_i and X_j

Problem



Euclidean Distance

$$d_E(X_i, X_j) = \sqrt{\sum_{g=1\ldots k} (X_{i,g} - X_{j,g})^2}$$

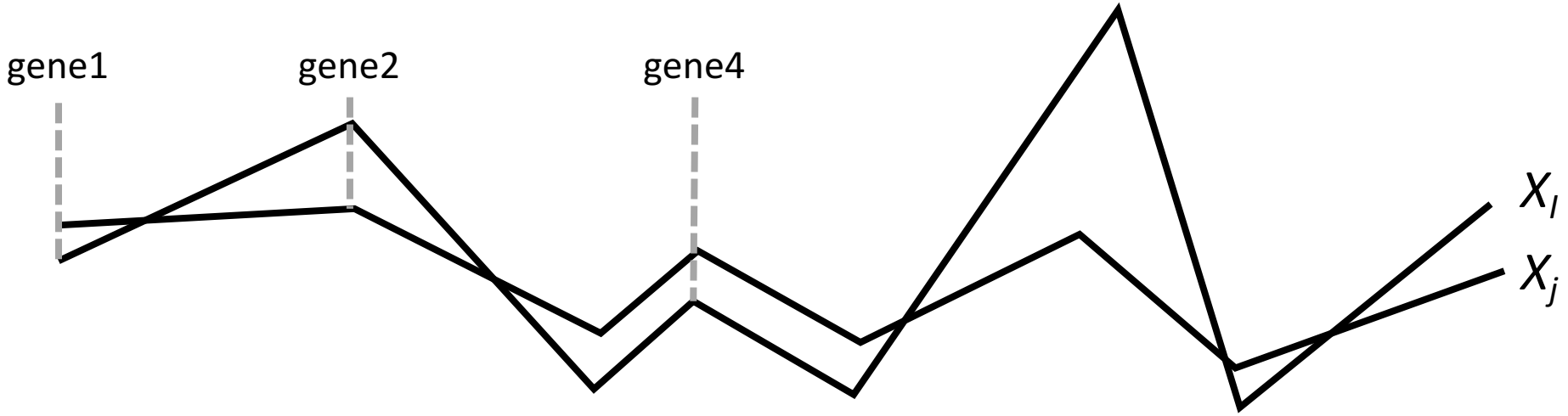
Pearson Correlation Coefficient

Similarity Measure

$$Sim(X_i, X_j) = \frac{Cov(X_i, X_j)}{\sqrt{Var(X_i) \times Var(X_j)}}$$

$$= \frac{\sum (X_i(g) - \bar{X}_i)(X_j(g) - \bar{X}_j)}{\sqrt{(\sum (X_i(g) - \bar{X}_i)^2) \times (\sum (X_j(g) - \bar{X}_j)^2)}}$$

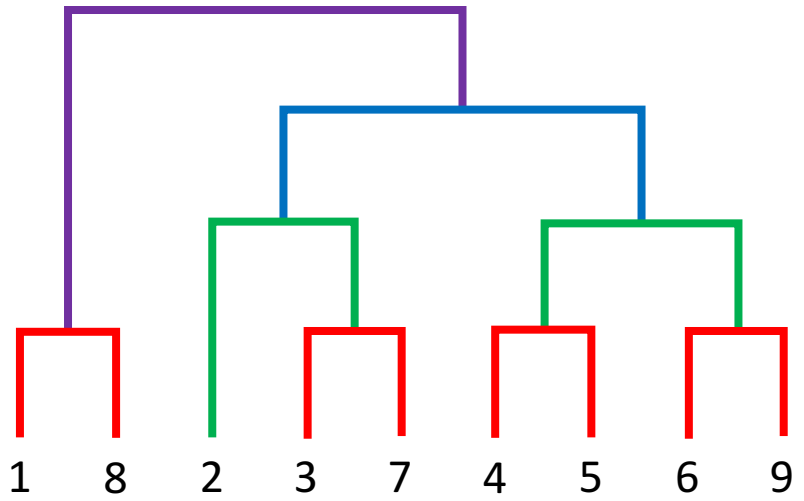
Pearson Correlation Coefficient Cont'd



- Different expression level
  - But always goes in the same direction

Hierarchical Clustering

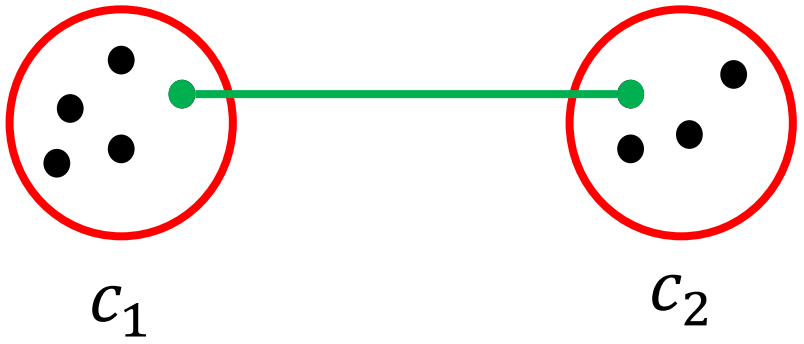
1. Start with each data point in its own cluster
2. Find the two clusters that are the closest and merge them
3. Repeat step two until all data points belong to a single cluster



Measuring Similarity Between Clusters

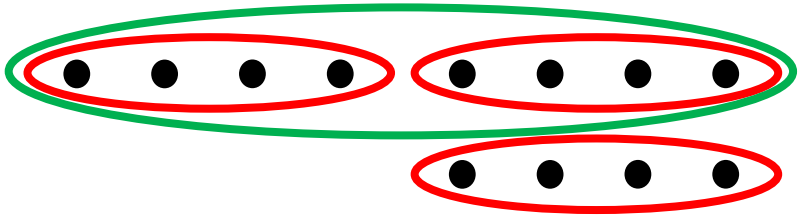
1) Single Linkage approach

$$Sim(c_1, c_2) = max_{x \in c_1, y \in c_2} \{sim(x, y)\}$$



Problem

- Given the following data points:

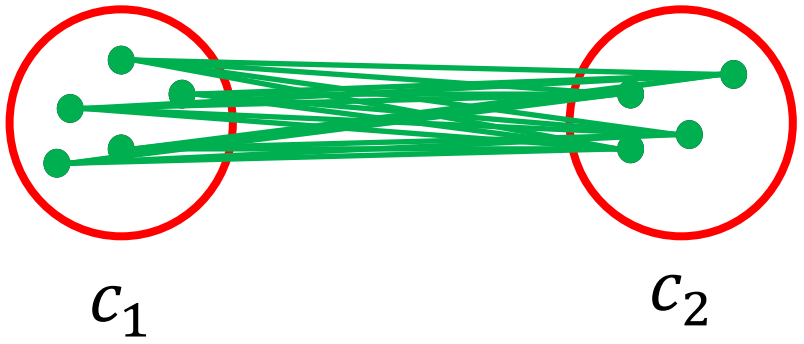


- Apply single linkage approach to clustering
- Get long and skinny clusters by having one point near the others
  - Shouldn't the two clusters on the right pair better together?

Measuring Similarity Between Clusters

2) Average linkage

$$Sim(c_1, c_2) = \frac{1}{|c_1| \cdot |c_2|} \sum_{x \in c_1, y \in c_2} Sim(x, y)$$



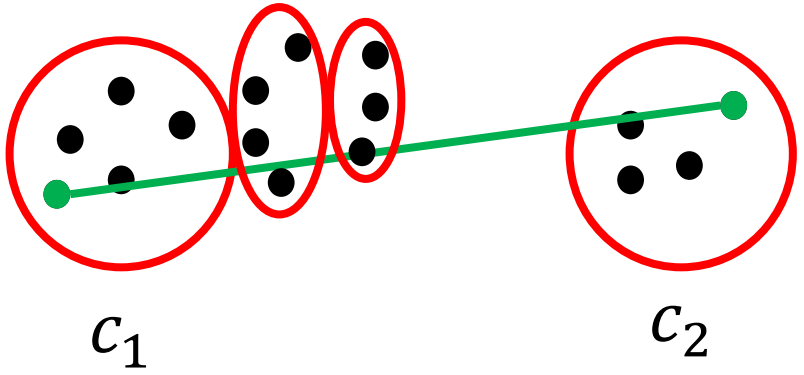
Take all pairs!

Measuring Similarity Between Clusters

3) Complete linkage

Makes very compact clusters

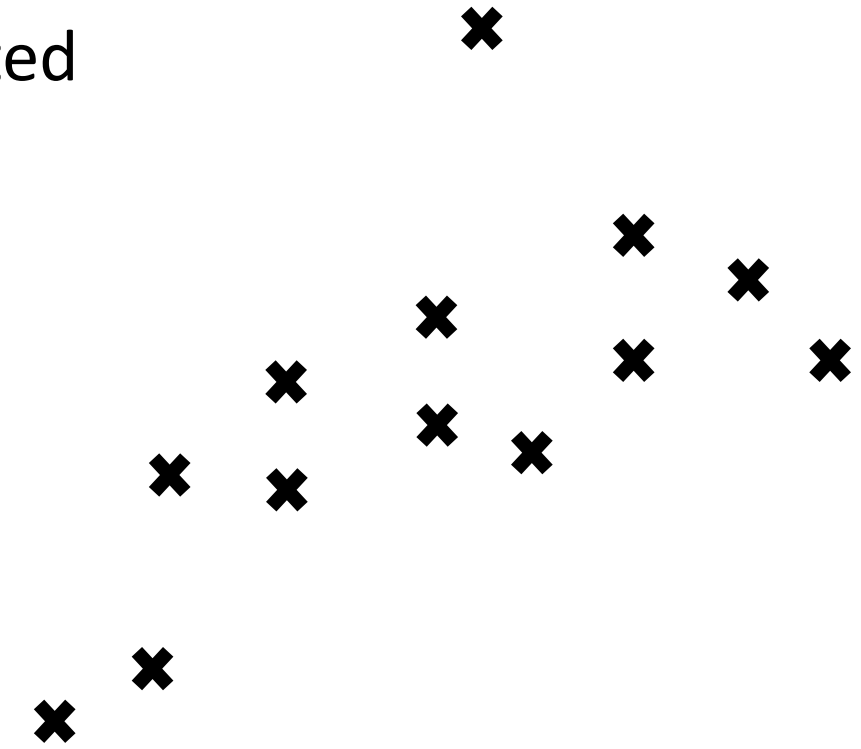
$$Sim(c_1, c_2) = min_{x \in c_1, y \in c_2}\ sim(x, y)$$



K-Means Algorithm

- '*k*' is the number of clusters desired / expected
- Each cluster has a centroid

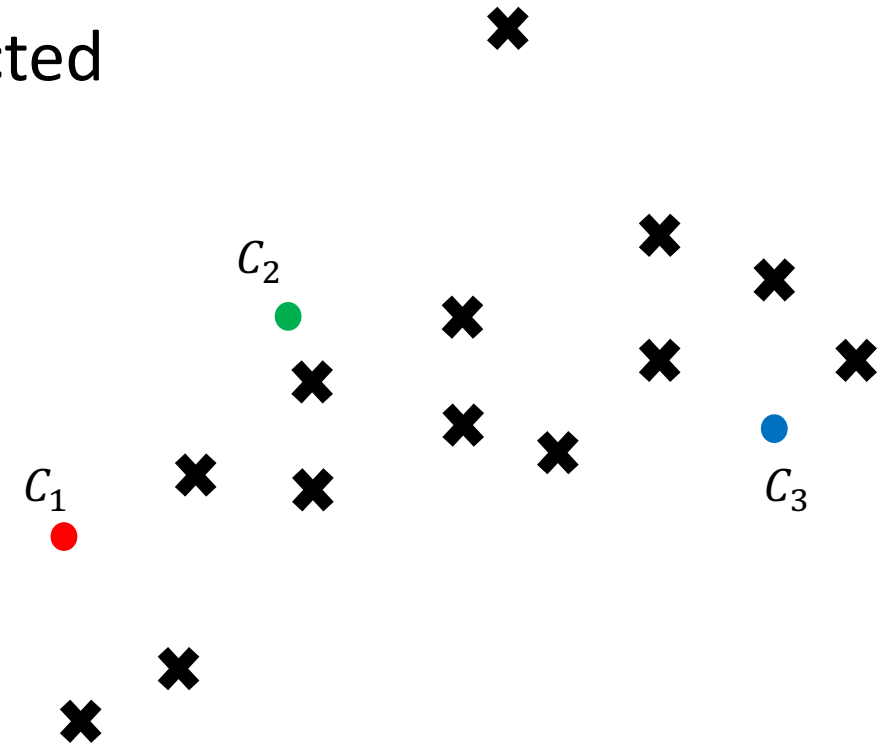
1. Randomly choose k centroids
2. Assign data points to nearest centroid
3. Move centroid to center of cluster
4. Repeat 2-4. Stop when no change to data point assignment



K-Means Algorithm

- '*k*' is the number of clusters desired / expected
- Each cluster has a centroid

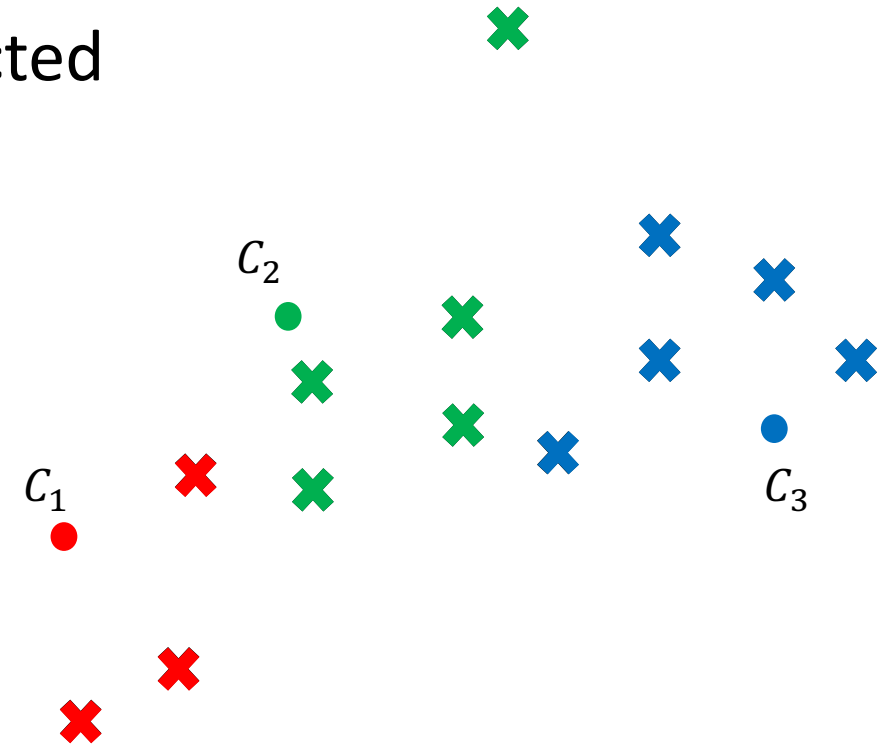
1. Randomly choose k centroids
2. Assign data points to nearest centroid
3. Move centroid to center of cluster
4. Repeat 2-4. Stop when no change to data point assignment



K-Means Algorithm

- '*k*' is the number of clusters desired / expected
- Each cluster has a centroid

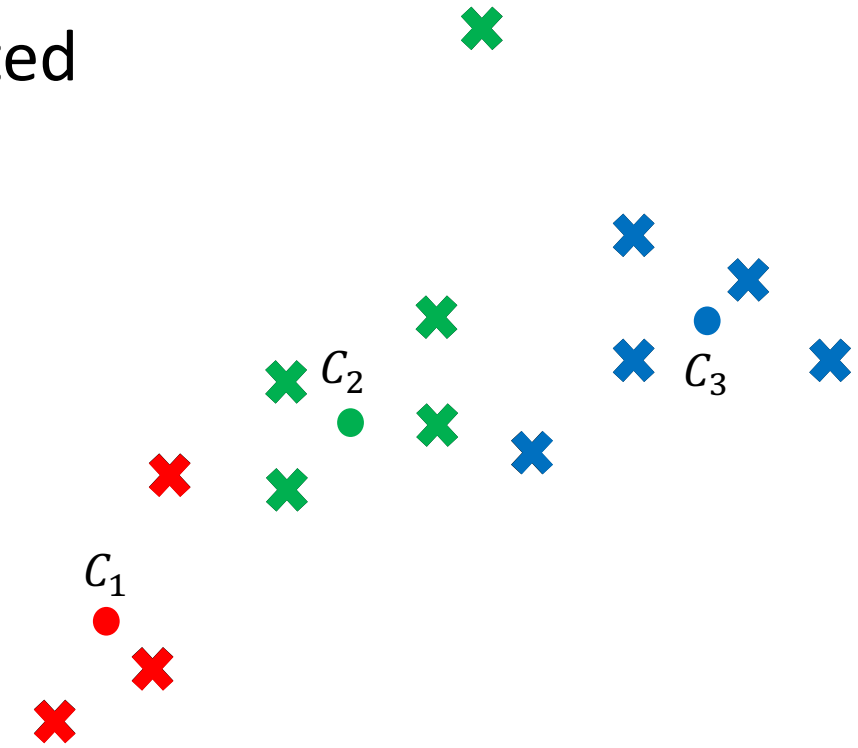
1. Randomly choose k centroids
2. Assign data points to nearest centroid
3. Move centroid to center of cluster
4. Repeat 2-4. Stop when no change to data point assignment



K-Means Algorithm

- '*k*' is the number of clusters desired / expected
- Each cluster has a centroid

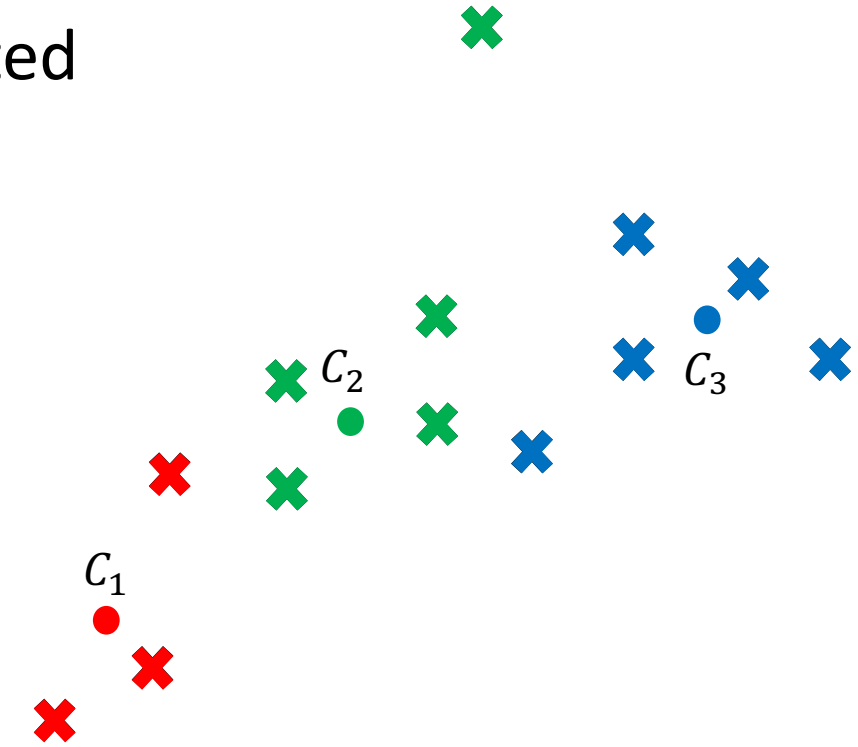
1. Randomly choose k centroids
2. Assign data points to nearest centroid
3. Move centroid to center of cluster
4. Repeat 2-4. Stop when no change to data point assignment



K-Means Algorithm

- '*k*' is the number of clusters desired / expected
- Each cluster has a centroid

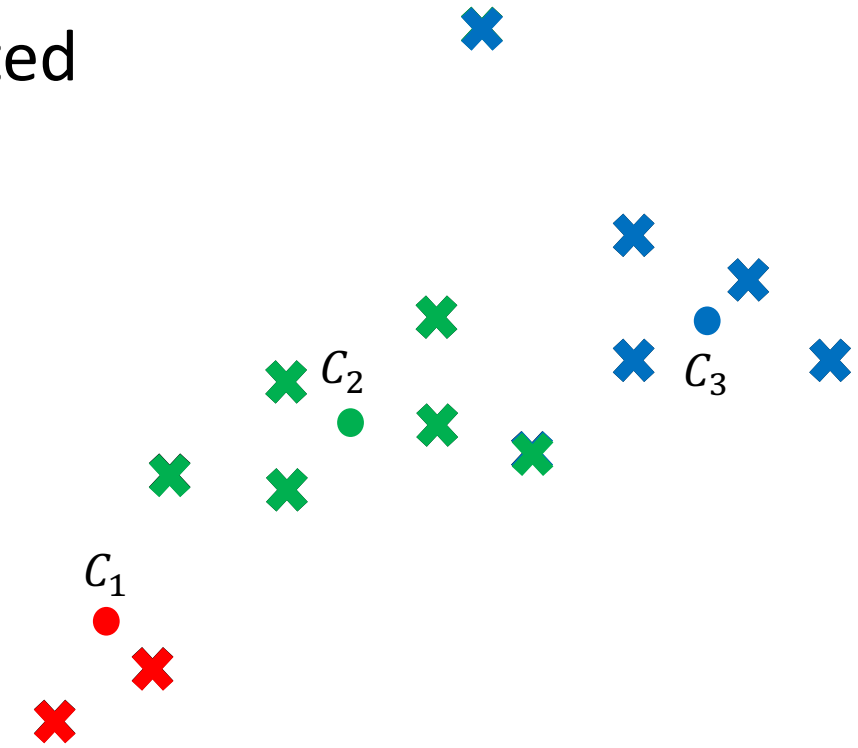
1. Randomly choose k centroids
2. Assign data points to nearest centroid
3. Move centroid to center of cluster
4. Repeat 2-4. Stop when no change to data point assignment



K-Means Algorithm

- '*k*' is the number of clusters desired / expected
- Each cluster has a centroid

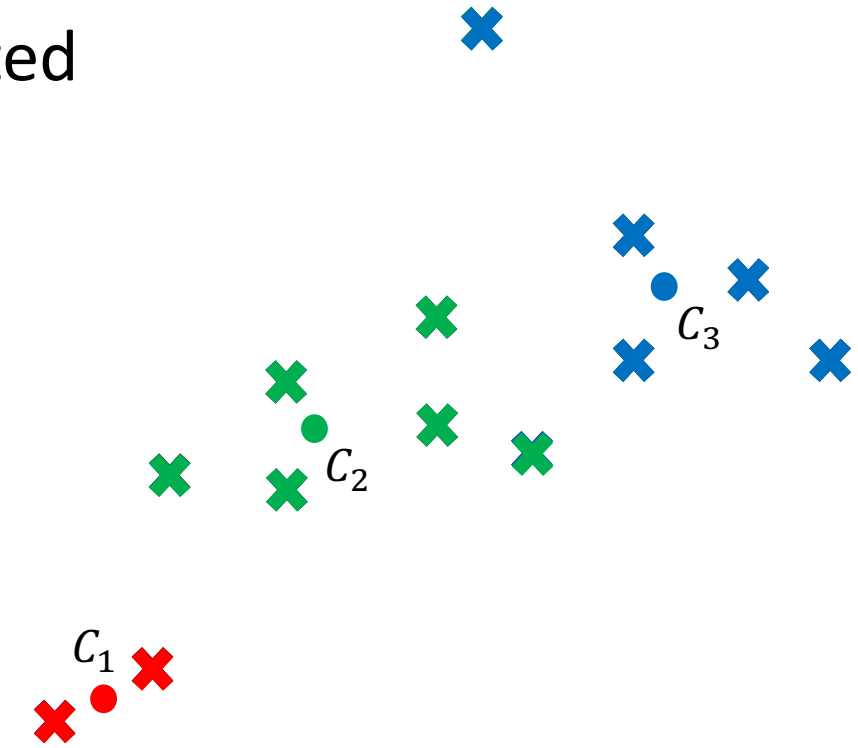
1. Randomly choose k centroids
2. Assign data points to nearest centroid
3. Move centroid to center of cluster
4. Repeat 2-4. Stop when no change to data point assignment



K-Means Algorithm

- '*k*' is the number of clusters desired / expected
- Each cluster has a centroid

1. Randomly choose k centroids
2. Assign data points to nearest centroid
3. Move centroid to center of cluster
4. Repeat 2-4. Stop when no change to data point assignment



Cluster Validation

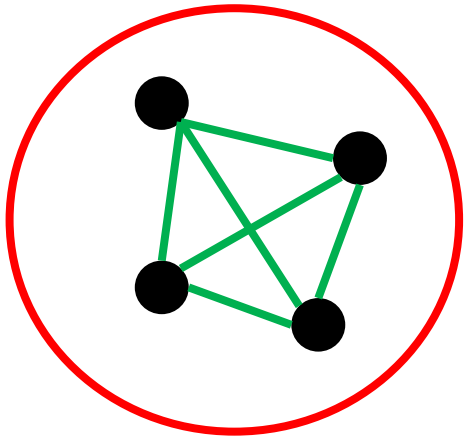
- **Cohesion:** measures how closely related data points in a cluster are (i.e., within cluster Sum of Squares [WSS])

$$WSS = \sum_{i} \sum_{x \in c_i} \|x - m_i\|^2$$

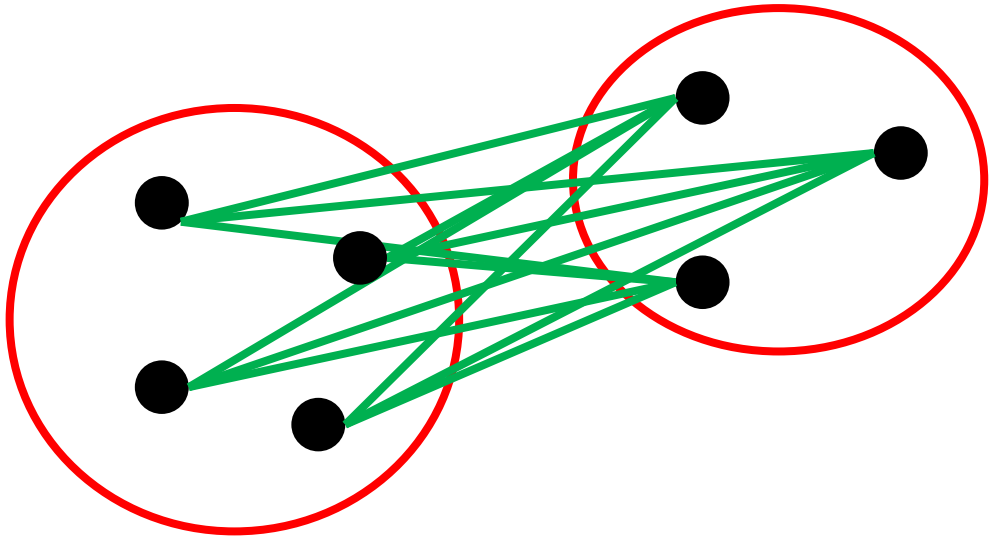
- **Separation:** measures how distinct or well-separated a cluster is from others (i.e., between cluster Sum of Squares [BSS])

$$BSS = \sum_{i} \sum_{j} |c_i| \cdot |c_j| \cdot \|m_i - m_j\|^2$$

Cohesion and Separation



Cohesion



Separation