BMI/CS 576 – Day 4

- Today
  - Fragment assembly
  - Graphs
- Next week
  - Spectral assembly
  - Assembly in practice

Submitting notebooks

- Please submit as you go!
- I recommend submitting after you complete each problem
- You are allowed to submit as many times as you like
  - Last submission is used for grading
- Advantages of submitting as you go:
  - You have partial work submitted in case something comes up before the deadline
  - Instructors can see how far along the class is in completing the problems (can adjust accordlingly)

Quiz

- The shortest superstring for the set of input strings {ATAG, CATA, TAAT} is

```
CATA
   TAAT
      ATAG
CATAATAG
```

Muddiest points

- Optional from now on

Shortest superstring task

- Greedy fragment assembly algorithm is not guaranteed to give the shortest superstring
- Shortest superstring is not necessarily the true genome
- These two facts are not encouraging
- Why bother with this approach?

Objective function-based approaches

- In science, we attempt to model/estimate the truth
  - In assembly, truth = true genome sequence
  - We can never know the truth, we can only collect data
- Common approach: objective functions
  - Estimate the truth by optimizing (minimizing or maximizing) an objective function, which is a function of the data
  - e.g., length of superstring is our current objective function
- Objection functions vary in
  - how close they come in their estimates of the truth
  - how computationally complex it is to optimize them

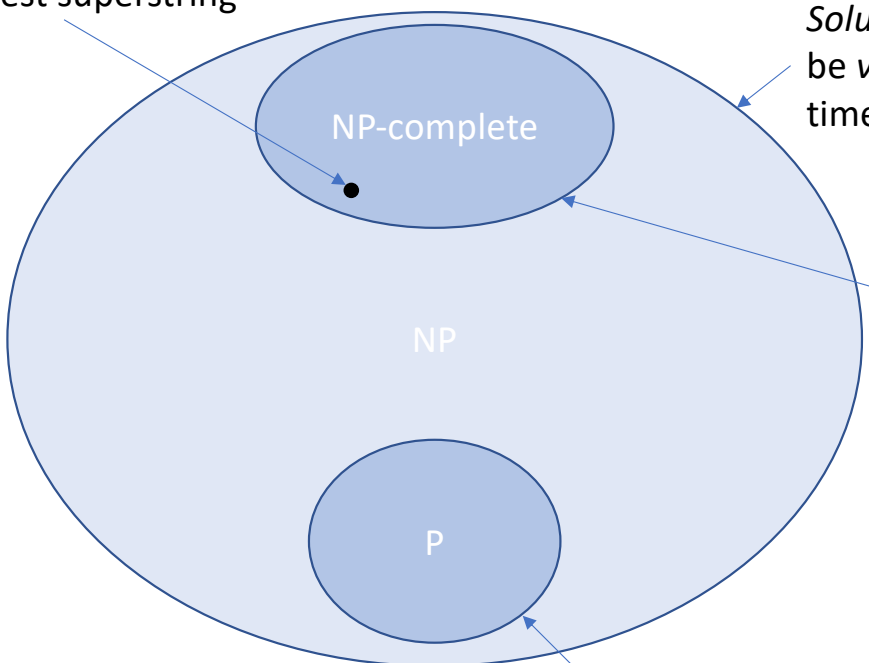
Spectral assembly

- Topic for next week (Tuesday)
- Hold off on questions until then
- Sneak preview
  - Uses a different objective function that
    - is not as good as shortest superstring in estimating the truth
    - is easy to optimize!

In practice

- Topic for next week (Thursday)
- There are numerous academic and commercial assembly method available
  - Vast majority are graph-based
- Assembly methods evaluated by
  - Simulation (truth is known)
  - Real sequencing data generated from genome for which we already ”know” the sequence

Computational Complexity



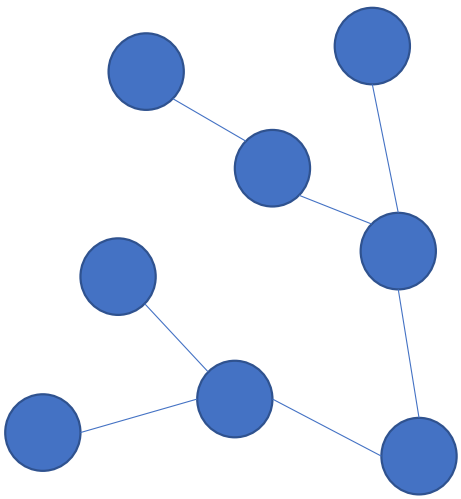
Solutions to problems in this class can be *verified* by a known polynomial time algorithm

Any problem in NP can be reduced (in polynomial time) to a problem in NP-complete

Is it possible that P = NP, however most theoreticians believe that this is highly unlikely

Problems in this class have known polynomial time algorithms for their solution

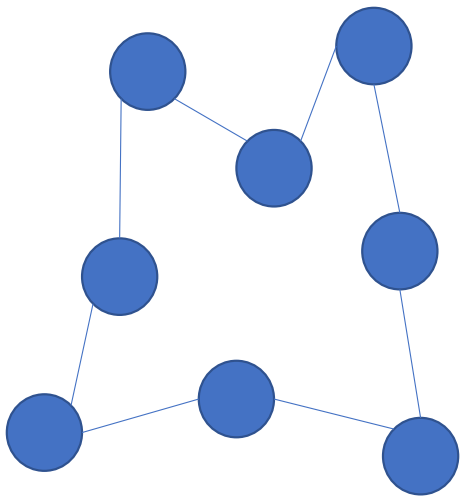
TSP and MST



Spanning tree

A minimum spanning tree (MST) minimizes the sum of the weights of the edges in the tree

A polynomial time greedy algorithm (e.g., Kruskal) gives optimal solution



Hamiltonian cycle

A solution to the Traveling Salesman Problem (TSP) is a Hamiltonian cycle that minimizes the sum of the edges in the cycle

Simple greedy algorithm is not guaranteed to give optimal solution