

BMI/CS 576 – Day 5

- Today
 - Spectral assembly
 - Multiple inheritance in Python
- Thursday
 - Assembly in practice

Notebook grading

- Lowest four notebook grades will be dropped across the semester
- Primarily to accommodate unexpected events
 - Illness
 - Emergencies

HW1

- Posted to Canvas
- Sequence assembly
- Due Mon, Oct 7

HW late days

- 4 “free” late days allowed across semester
- Primarily to accommodate unexpected events
 - Illness
 - Emergencies

UW-Madison Multi-factor authentication

- Use secondary device (e.g., phone) to prove identity when logging in with NetID
- Required for students after Oct 31
- Enroll as soon as possible to avoid problems
- Won't be able to log in to any wisc services (including Canvas) without it!
- <https://it.wisc.edu/learn/guides/getting-started-multi-factor-authentication-students/>

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 accordingly)

How is spectral assembly used in practice?

- de Bruijn approach
 - use shotgun sequencing read data
 - obtain k-mer spectrum from union of spectra of reads
 - apply Eulerian path approach to computing assembly
- Relative to the shortest superstring approach, the de Bruijn approach:
 - Changes the objective function
 - to a function that is less accurate
 - Make optimizing the objective function feasible
 - Polynomial time algorithm

Eulerian path algorithm

- Adding the “missing edge”
 - Only if the graph is not already balanced
 - If not balanced, there should be
 - exactly one vertex (s) with $\text{outdegree} - \text{indegree} = 1$
 - exactly one vertex (t) with $\text{indegree} - \text{outdegree} = 1$
 - Add an edge from t to s to balance the graph

Eulerian path algorithm

- Choosing a vertex to start from to find Eulerian cycle
 - Can be done arbitrarily, since we can start tracing a cycle from any vertex along it
 - Different start vertices lead to different solutions?
 - There may be multiple possible Eulerian cycles
 - -> multiple possible superstrings
 - If only one Eulerian cycle exists, doesn't matter which vertex you start at
- Breaking the cycle to form a path
 - If a “missing edge” was added, this edge is removed
 - Otherwise, can break at any vertex
 - Multiple possible paths