Agenda

- Assignment 3 Wrap Up
- Assignment 4
Assignment 3 Wrap Up
Edge Case

CDS

1..1356
@gene=dnaA"

<table>
<thead>
<tr>
<th>LOCUS</th>
<th>NZ_LS483333</th>
<th>1746380 bp</th>
<th>DNA</th>
<th>circular CON 25-DEC-2022</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFINITION</td>
<td>Streptococcus pyogenes strain NCTC12064 chromosome 1, complete sequence.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. Parse a genbank file (.gbff) and...
   a. Extract all CDS features
   b. Read in the sequence
2. Build a site model for translation start sites (TSS)
   a. Use CDS features to get nucleotide frequencies +/- 10bp around all TSS (21bp total including TSS)
   b. Use sequence to get nucleotide frequencies throughout the genome on both strands
   c. Compute the weights using the log2 ratios of the frequencies
3. Use the site model to compute scores at
   a. Every annotated TSS
   b. The entire genome (21bp window) on both strands
Assignment 4
Overview

Part 1: Write a program to find the highest-weight path in a directed acyclic graph using dynamic programming

Part 2: Run your program on a linked list created from DNA sequence
Program 1: Highest weight path

1. Convert graph to text file of **vertices** and **edges** by hand
2. Use dynamic programming to find the max weight path through the graph (Lectures 7/8)
   a. Overall
   b. With constraints (START/END)
3. Output
   a. Path Score
   b. The start/end vertex on the path
   c. Labels for all the edges on path (in order)

Example:
V vii START
V vi
V v
...
E A ii i -1
E B iii i 5

Part 1
Score: 8.0
Begin: vi
End: ii
Path: ID

Part 2
Score: 4.0
Begin: vii
End: i
Path: LIDA
Example - Unconstrained

my_graph.txt:

V I
V II
V III
V IV
V V
E A I II -1
E B I III 5
E C II III 7
E D II IV 4
E E III V 8
Example - Unconstrained

my_graph.txt:

Score: 15
Begin: II
End: V
Path: CE
Example - Constrained

my_graph_constrained.txt:

V I START
V II
V III
V IV
V V END
E A I II -1
E B I III 5
E C I II III 7
E D II IV 4
E E III V 8
Example - Constrained

```
my_graph_constrained.txt:

V I START
V II
V III
V IV
V V END
E A I II -1
E B I III 5
E C II III 7
E D II IV 4
E E III V 8

Score: 14
Begin: I
End: V
Path: ACE
```
Assume that graph file is depth ordered
Vertex I has no parents so points to itself

<table>
<thead>
<tr>
<th>Vertex</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highest Weight Parent</td>
<td>I</td>
<td>II</td>
<td>III</td>
<td>IV</td>
<td>V</td>
</tr>
<tr>
<td>w(v) (Vertex weight)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Best Path Start: I
**Example - Dynamic Programming**

**my_graph.txt**:

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Highest Weight Parent</th>
<th>w(v) (Vertex weight)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>II</td>
<td>0</td>
</tr>
<tr>
<td>II</td>
<td>I</td>
<td>5</td>
</tr>
<tr>
<td>III</td>
<td>I</td>
<td>5</td>
</tr>
<tr>
<td>IV</td>
<td>IV</td>
<td>0</td>
</tr>
<tr>
<td>V</td>
<td>III</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Start</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>III</td>
</tr>
</tbody>
</table>

**Diagram**:

- Vertex I
- Highest Weight Parent: II
- w(v) (Vertex weight): 0
- Start: III
Example - Dynamic Programming

**my_graph.txt:**

```
I
A, -1
II
B, 5
III
D, 4
C, 7
IV
7
V
E, 8
```

<table>
<thead>
<tr>
<th>Vertex</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highest Weight Parent</td>
<td>I</td>
<td>II</td>
<td>II</td>
<td>II</td>
<td>V</td>
</tr>
<tr>
<td>w(v) (Vertex weight)</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Start</td>
<td>III</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Graph Details

- **Vertex I**
  - Weight: 0
  - Parent: A

- **Vertex II**
  - Weight: 5
  - Parent: B

- **Vertex III**
  - Weight: 7
  - Parent: C

- **Vertex IV**
  - Weight: 4
  - Parent: D

- **Vertex V**
  - Weight: 8
  - Parent: E
**Example - Dynamic Programming**

*my_graph.txt:*

- **V I**
- **V II**
- **V III**
- **V IV**
- **V V**

**Vertices and Weights:***

- **A, -1**
- **B, 5**
- **C, 7**
- **D, 4**
- **E, 8**

**Highest Weight Parent:***

- **I → II**
- **II → III**
- **III → IV**
- **III → V**

**Vertex w(v) (Vertex weight):***

- **I: 0**
- **II: 0**
- **III: 7**
- **IV: 4**
- **V: 15**

**Best Path Start:***

- **V**
Example - Dynamic Programming

my_graph.txt:

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>VI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VII</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VIII</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VII</td>
<td></td>
<td>B, 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VII</td>
<td></td>
<td>C, 7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VII</td>
<td></td>
<td></td>
<td></td>
<td>D, 4</td>
<td></td>
</tr>
<tr>
<td>VII</td>
<td></td>
<td></td>
<td></td>
<td>E, 8</td>
<td>15</td>
</tr>
</tbody>
</table>

- Now traceback to find highest weight path
Program 2: DNA Linked List

1. Create a linked list from a DNA sequence and a scoring scheme
   a. Positions are vertices
   b. Bases are edges
2. Run your program from part 1 on the graph

Example:

Scores
A = -1.49
T = -1.49
G = .74
C = .74

Sequence: AGCT
Graph:
V 0
V 1
V 2
V 3
V 4
E A -1.49
E G .74
E C .74
E T -1.49
**Program 1:**
Use dynamic programming to find the highest weight path in an arbitrary WDAG

**Program 2:**
Make a linked list from a fasta and run program 1 on it
Reminders

- HW4 due this Sunday, 11:59pm
- Please have your name in the filename of your homework assignment and match the template