CS 3704 Project Milestone 3

Group Name: Sponsored by PaxHistoria

Group Members: Eli Bullock-Papa, Connor Brodish, Kishitij Kaushal, and Ryan Zhang

Process Deliverable II

The submission for this deliverable will depend on the specific SE process model your team plans to use to complete the group project (as described in your project proposal).

Scrum: submit notes (include each teammate) from at least weekly scrum meetings.

Meeting 6 (Nov 4 – 8)

1. Member: Connor Brodish
    - What I did? Reviewed project milestone 3 spec
    - What I need to do next? Need a break, feeling mentally and physically exhausted
    - What is blocking me? Multiple coding/project/homework assignments to balance, as well as a comparative languages midterm
2. Member: Kishitij Kaushal
    - What I did? Looked at Project Milestone 3
    - What I need to do next? Prepare for Discussion presentation
    - What is blocking me? Assignments
3. Ryan Zhang:
    - What I did? Prepared documentation for the project.
    - What I need to do next? Conduct some interviews with classmates and get feedback.
    - What is blocking me? Biomedical Engineering midterm
4. Eli Bullock-Papa:
    - What I did? Searched for libraries for 3D visualization.
    - What I need to do next? Figure out how to integrate it with our project.
    - What is blocking me? I had some assignments due this week.

Meeting 7 (Nov 11 – 15)

1. Member: Connor Brodish
    - What I did? Took a wellness week, thought I deserved a break
    - What I need to do next? Begin working towards milestone 3 (create document, outline, etc.)
    - What is blocking me? Nothing now, feeling better and ready to work
2. Member: Kishitij Kaushal
    - What I did? Discussion Presentation.
    - What I need to do next? Meet with teammates to discuss the project
    - What is blocking me? Systems Project

3. Ryan Zhang:
    - What I did? Interviewed classmates regarding our project.
    - What I need to do next? Analyze the interviews and recommend to groupmates the changes we should make.
    - What is blocking me? I have a Systems Project due next week.
4. Eli Bullock-Papa:
    - What I did? Took a break because I was sick
    - What I need to do next? Integrate the library with our code.
    - What is blocking me? Multiple assignments

Meeting 8 (Nov 18 – 22)

1. Member: Connor Brodish
    - What I did? Created this document, notified groupmates of upcoming deadline
    - What I need to do next? Wait for other groupmates to delegate any tasks to me
    - What is blocking me? Assignments to complete before leaving for Thanksgiving (due to procrastination and less time)

2. Member: Kishitij Kaushal
    - What I did? Worked on PM3 after being notified
    - What I need to do next? Start coding and get the project working.
    - What is blocking me? Systems Project
3. Ryan Zhang:
    - What I did? I recommended the team to make changes on the project
    - What I need to do next? Work on the documentation of the project and figure out how to deploy.
    - What is blockin me? I have a Systems Project due this week.

4. Eli Bullock-Papa:
    - What I did? Made some changes to the UI to enhance user experience.
    - What I need to do next? Enjoy the thanksgiving break.
    - What is blocking me? I have another Project due this week.

High-level Design

The combination of event-based and layered architecture is ideal for Comp Bio Helpers as it is well suited for a nextJS application like ours. The event-based approach is standard for React applications, and allows for dynamic, real-time updates across the user interface, such as recalculating alignments or updating 3D visualizations based on user input, ensuring a responsive and interactive experience.

Meanwhile, the layered architecture organizes the backend into distinct layers (UI, and API/data layers), promoting separation of concerns. This structure simplifies the management of complex algorithms (e.g., global alignment and BLOSUM computations) while maintaining a clean interface between frontend and backend.

Low-level Design

Discuss which design pattern family might be helpful for implementing a specific subtask for this project. Justify your answer, providing a code or pseudocode representation *and* an informal class diagram.

Design Pattern Families:

We chose the subtask of Updating and Displaying the 3D Visualization Dynamically Based on User Input (RNA Sequences and Scoring Parameters).

State Pattern:

- The state of the application (RNA sequences, match/mismatch/gap scores) directly controls the behavior of the app.
- Changes in the state will trigger updates in multiple components, such as the 3D visualization and statistical displays.

Observer Pattern:

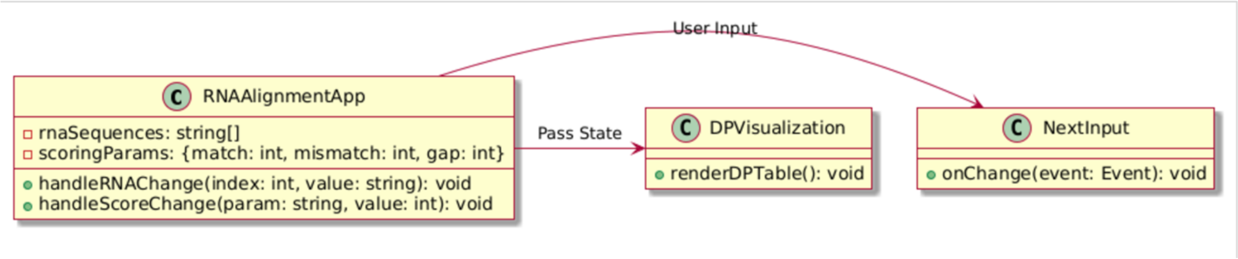
- React's state mechanism automatically updates dependent components when the state changes, implementing the Observer pattern natively.
- This ensures that components stay in sync with the latest input values without requiring manual updates.

Pseudocode Representation:

```
import React, { useState } from "react";

// Main App Component
const RNAAlignmentApp = () => {
  // State: RNA sequences and scoring parameters
  const [rnaSequences, setRNASequences] = useState(["", "", ""]);
  const [scoringParams, setScoringParams] = useState({
    match: 1,
    mismatch: -1,
    gap: -2,
  });

  // Handlers for input changes
  const handleRNAChange = (index, value) => {
    const newSequences = [...rnaSequences];
    newSequences[index] = value;
    setRNASequences(newSequences);
  };

  const handleScoreChange = (param, value) => {
    setScoringParams({ ...scoringParams, [param]: parseInt(value, 10) });
  };

  return (
    <div>
```

```
      <h1>RNA Alignment Tool</h1>

      {/* Input Section */}
      <div>
        <h2>Input RNA Sequences</h2>
        {rnaSequences.map((seq, index) => (
          <NextInput
            key={index}
            value={seq}
            placeholder={`RNA Sequence ${index + 1}`}
            onChange={(e) => handleRNAChange(index, e.target.value)}
          />
        ))}

        <h2>Scoring Parameters</h2>
        <NextInput
          value={scoringParams.match}
          placeholder="Match Score"
          onChange={(e) => handleScoreChange("match", e.target.value)}
        />
        <NextInput
          value={scoringParams.mismatch}
          placeholder="Mismatch Penalty"
          onChange={(e) => handleScoreChange("mismatch", e.target.value)}
        />
        <NextInput
          value={scoringParams.gap}
          placeholder="Gap Penalty"
          onChange={(e) => handleScoreChange("gap", e.target.value)}
        />
      </div>

      {/* Visualization Section */}
      <DPVisualization rnaSequences={rnaSequences} scoringParams={scoringParams} />
    </div>
  );
};

// Dynamic 3D Visualization Component
const DPVisualization = ({ rnaSequences, scoringParams }) => {
  // Simulate rendering of 3D DP Table
  const renderDPTable = () => {
    console.log("Rendering DP table with RNA sequences:", rnaSequences);
    console.log("Using scoring parameters:", scoringParams);
    // Add visualization logic (e.g., using Three.js)
  };

  // Trigger render whenever props change
  React.useEffect(() => {
    renderDPTable();
  }, [rnaSequences, scoringParams]);

  return <div>3D Visualization Placeholder</div>;
};

export default RNAAlignmentApp;

```

Class Diagram:


RNAAlignmentApp
rnaSequences: string[]
scoringParams: {match: int, mismatch: int, gap: int}
handleRNAChange(index: int, value: string): void
handleScoreChange(param: string, value: int): void
User Input
Pass State
DPVisualization
renderDPTable(): void
NextInput
onChange(event: Event): void

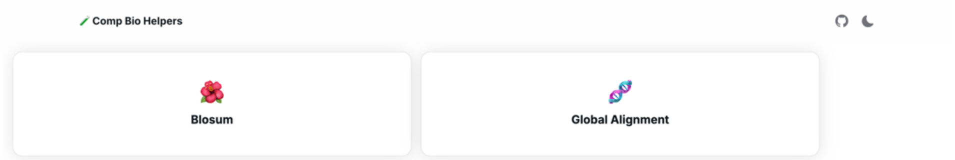
Design Sketch

Design sketches provide a visual overview of the look and feel of your project. This may include but is not limited to one of the following:

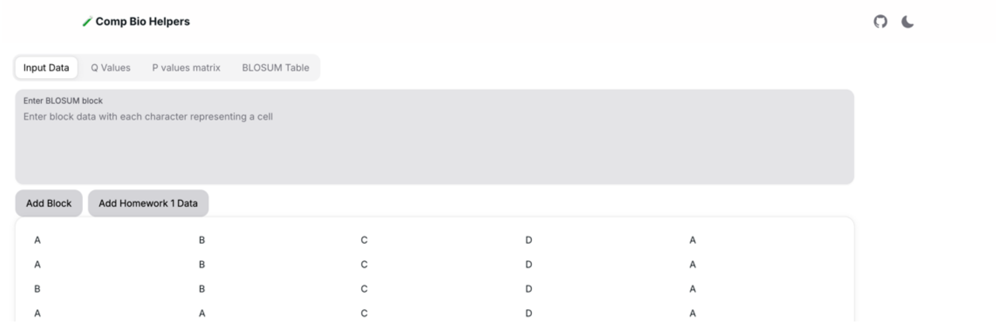
- Creating a wireframe mockup of your project user interface in action.
- Creating a storyboard that illustrates a primary task that a user would complete with your project.

Provide a brief rationale (no more than 1 paragraph) explaining some of the design decisions for your program based on the provided sketch. Use concepts discussed in class to justify your design.

Landing Page



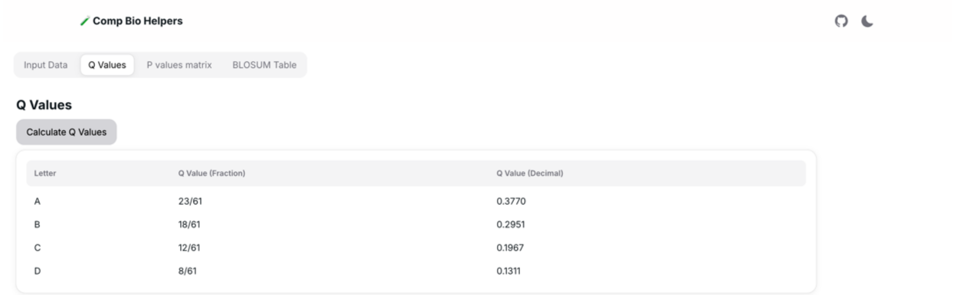
Blosum Table



Enter BLOSUM block

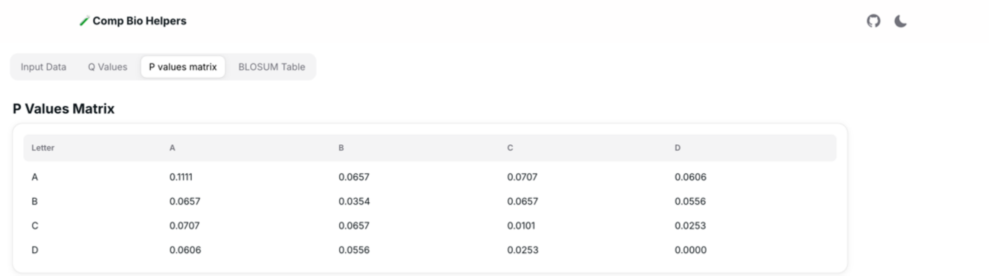
Enter block data with each character representing a cell

A	B	C	D	A
A	B	C	D	A
B	B	C	D	A
A	A	C	D	A



Q Values

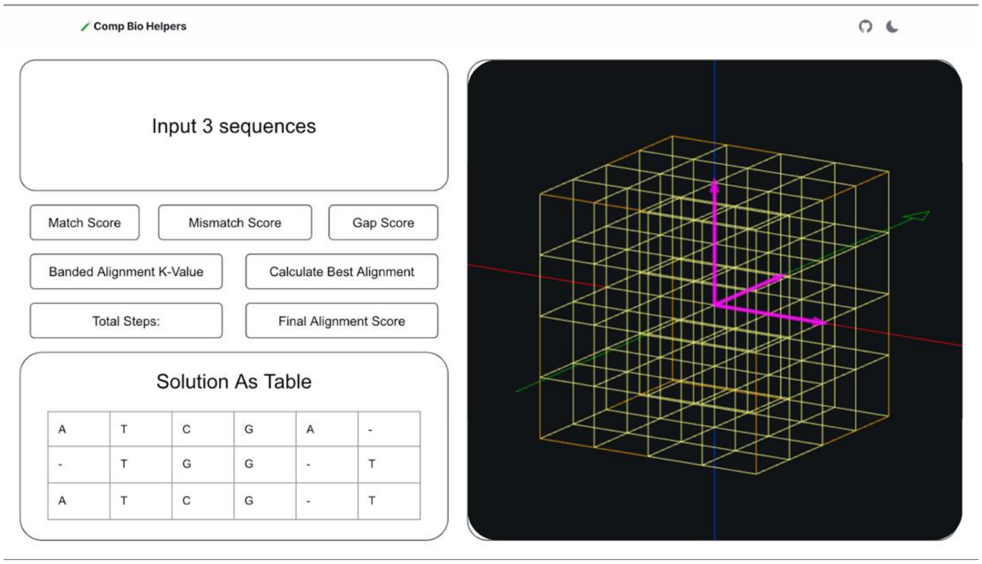
Letter	Q Value (Fraction)	Q Value (Decimal)
A	23/61	0.3770
B	18/61	0.2951
C	12/61	0.1967
D	8/61	0.1311



P Values Matrix

Letter	A	B	C	D
A	0.1111	0.0657	0.0707	0.0606
B	0.0657	0.0354	0.0657	0.0556
C	0.0707	0.0657	0.0101	0.0253
D	0.0606	0.0556	0.0253	0.0000

3D Global Alignment



Solution As Table

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

Rationale For Design Decisions

By grouping input parameters (e.g., Match Score, Mismatch Score, Gap Score) and the “Calculate Best Alignment” button together, the interface places users in control of the alignment process and provides a clear workflow. The dynamic 3D visualization of the DP table reduces users’ memory load by making complex algorithmic steps tangible and easier to understand, aligning with the principle of recognition over recall. Consistent placement of input fields, outputs, and visualization ensures predictability, enhancing usability and aligning with the heuristic of consistency and standards.