Contributions table:

Group members	Net ids	roles	Milestone 1	Milestone 2	Other contributions

Thomas Rendon	trendon2	Facilitator	mendels_law(hom, het, rec)
			fibonacci_rabbits(n,k)
			hamming_dist(dna1, dna2)
				get_edges(dna_dict)
				rev_palindrome(dna)
					Setup meeting times and made sure everyone was available.

Alexander Spath	aspath2	Integrator	reverse_complement(dna)
			gc_content(dna_list)
			rna2codon(rna)
			splice_rna(dna, intron_list)
				perfect_match(rna)
				find_splice(dna_motif, dna)
					Submitted assignments and did last minute fixes to some functions/comments

Andrew Sprenger	aws10	Strategist	source_rna(protein)
			locate_substring(dna_snippet, dna)
			dna2rna(dna)
			count_dom_phenotype(genotypes)
				shared_motif(dna_list)
				random_genome(dna,gc_content)
					Worked on creating task list and monitoring workload

Introduction:

In our CS 101 final project we were tasked with three milestone projects composed of varying numbers of functions as well as a final report. The central theme of the project was sequencing and processing DNA; in the varied functions we developed tools to assist in related processes. The final report, as mentioned above, requires a table describing group member contributions (also above), three technical analyses (one for each member) each detailing a single function in each of the first two milestones, an introduction to the project (you're reading it), three body paragraphs of literary analysis of scholarly articles which form a cohesive argument (for an improved final project) that tie into the thesis (below this description), and a conclusion which encapsulates central takeaways and the central flow of the argument. Throughout the three literature reviews below we will describe improved methods for the injection of interactive and collaborative elements into this project with the implementation of lessened abstraction and workload distributions.

Tech analysis 1: Andrew Sprenger

A function I helped work on in Milestone 1 was reverse_complement(dna). The purpose of this function is to take a DNA string and return the reverse complement. Although the function is pretty straightforward and Lab 3 helped a little, I struggled getting started. I think I was overthinking it and making it way too complicated. First, I tried to reverse the string and take the complement in one line of code within the loop, but I could not get it to work. Finally I realized that it would be easier to just reverse the string first so I did that. The class material that I used to complete this function was Lab 3. It provided the basic outline for the functions s(dna) and dna2rna(dna) which I applied to reverse_complement. I also could not remember how to create an empty string and return the answer as a string at first. I went back into the lessons and figured that out relatively quickly.

A function that I worked on in Milestone 2 was get_edges(dna_dict). The purpose of this function is to take a dictionary with DNA strings and return a list of their Rosalind identifiers. I had a lot of trouble figuring out how to start this function. The project page had a lot of information about graph theory, and I was confused how to apply that to the function. However, it did get easier as I started to think it through. The trouble I has was with the double for loops. I understood how to match the prefixes with the suffixes, but it took me a while to figure out how exactly to do that with two loops. The class material that helped me complete this function was the lessons. I do not remember which one specifically, but there was one that had an example of how to use two for loops. I also went back to the lessons to refamiliarize myself on how to use dictionaries. These helped me understand how to create this function.

Tech analysis 2: Thomas Rendon

One of the functions I helped work on in Milestone 1 was mendels_law(hom, het, rec). The purpose of the mendels_law function was to determine the likelihood of traits in different organisms. There are three possible traits that an organism may have. These possibilities are homozygous dominant being AA, heterozygous being Aa, and homozygous recessive being aa. The limitations I faced on this code were the two equations I was able to use to calculate the likelihood of traits. The main difficulty I faced while completing this code was the math behind calculating the likelihoods of traits. I had to figure out the math behind the likelihood on paper first before figuring it out on code. In order to fix the code, I had to make sure I had all of the math figured out and had to match the variables correctly with the mathematics.

One of the functions I helped work on in Milestone 2 was rev_palindrome(dna). The purpose of the rev_palindrome function is to turn the reverse complement of a string with length of 4 to 12 and turn it into lists of tuples (position,length). Limitations that I faced while coding this function was the length having to be from 4 to 12. If the string of DNA was shorter or longer than that specific length then it wouldn't return a tuple back. I noticed that part of the coding process would be difficult when trying to code the iterations with the correct length. A way to debug this problem would be introducing another for loop of j.

Tech analysis 3: Alexander Spath

In milestone one I worked on gc_content(dna_list), its purpose was to find the highest GC (nucleotides) content (as a percentage) from among a list of DNA strings and return it as a tuple containing the maximum and the index of the dna string with that maximum value. Getting started was not too difficult as I immediately knew I needed an empty list to fill with the GC percentage values of each dna string. I began with a for loop that would look at each string in the dna_list and append the empty list with the sum of all its G and C nucleotides divided by the total number of its nucleotides. Here I had a list of GC nucleotide percentages for every string in dna_list and all that was left was to return the index of the string with the maximum and the maximum value as a tuple (ie. (indexmax, max).

In milestone two I worked on perfect_match(rna). With this function I was tasked with finding the total number of perfect matchings for a given rna string, which seemed rather difficult, for a while I was stumped until I remembered the sections of statistics that were covered in my thermal physics course and knew there had to be some simple mathematical relationship between nucleotide counts and perfect matchings (I visualized it as something similar to unique states). I began with finding the nucleotide counts for A,C,G,U within rna and then moved on to an if else statement that checked for the presence of G and C nucleotides if there weren't any then the math (imported math and used math.factorial) would only use the counts of A and U nucleotides (otherwise it would consider G and C) I tested out a few simple relationships and found that (G count) factorial*(C count) factorial yielded the desired result (alternatively A and U). I also noticed that there is the case of a nucleotide not having enough matching pairs to yield any perfect matchings thus I accounted for this in an if statement which returned 0.

Lit analysis 1: Andrew Sprenger

In order to improve the final project for this class, it needs to take a more interactive and collaborative approach. "Impact of Interactive Education on the Learning Outcomes and Quality Assurance," [1] an article written by Ihab Ali El-Qirem,Ayman Abdalmajeed Alsmadi, and Enas Al-lozi, discusses the benefits of interactive learning and how to implement it into the education system. The article mentions accomplishing this through the "implementation of interactive tools such as communication tools and other tools, presentation tools, multimedia recordings of interactive sections, and other techniques and software" [1]. One aspect that I believe is the most important is the multimedia recordings of interactive sections. If you miss a lecture in this class, the only other way to gain the information you missed is through reading the lessons. However, many students may find it difficult to learn something as complex as coding just through reading. Having recordings of the lectures in some form may make it easier for students to retain information and apply it to the project. Even if it is not the full lecture, short walkthroughs of important concepts and ideas would be extremely beneficial to students to complete the project. Another point brought up in the article is that aspects of collaborative and interactive learning "such as visual presentation of field models and practical demonstration of ideas are vital"[1] to students and their future. While there are practical demonstrations happening in lecture, I felt that it is still not enough. The project page in relate mentioned that there would be dedicated days to work on the project in lab. However, there were none of these days. A way to improve the project would be to dedicate some of those labs to have an interactive learning approach between the students and TA. The functions would be a lot more clear with an explanation from a TA in person and the project would be a lot more manageable overall.

Lit analysis 2: Thomas Rendon

According to an article by Anuradha A. Gokhale created a study on 48 students based on collaborative learning and how it applies to critical thinking. The study was conducted on two groups of students. The selected groups of study were based on individual students studying material and a group working together studying material. The groups were given a lecture, worked on a worksheet and then took a pretest and posttest. The scores from each group were then analyzed. According to the study, "...students who participated in collaborative learning had performed significantly better on the critical- thinking test than students who studied individually" [2]. To be exact on the difference of scoring, the collaborative group earned an average of 12.21 on the critical thinking questions and the individual group earned an average of 8.63 on the critical thinking questions. This can be found as a significant difference in scoring per group. This study proves that collaborative group studies perform better and have better critical thinking skills than individual students studies. This can be concluded as true because groups can possess greater knowledge in learning material. Groups are able to share their ideas in ideas such as their initial knowledge and what they learned on problems. Whereas an individual is only the knowledge they have themselves and what they learn without the help of others. This study correlates with the CS101 group project. Although we were in groups for this project, collaborative work never panned out the way it should've. As we were individuals in college with different schedules every single day, we found it hard to set up meetings and work together. A majority of our project was completed through splitting up work and working on it individually. As an introductory coding class, I believe the workload was too much for students who intend to focus on other classes relating more to their major and more important for their future. A lesser workload would allow students to work together. This would support the study on collaborative learning and critical thinking skills being applied.

Lit analysis 3: Alexander Spath

Herein we have demonstrated that collaborative and interactive learning provides students with creative avenues and partnerships that allow for a more effective learning environment; we ought to discuss how we can improve such structures within our project with an accessible and intuitive workspace, as well as, some other improvements. A study looking into the infusion of collaborative and interactive learning into an introductory computer science course seems to provide a satisfactory solution. In [3], repl.it, "an interactive computer programming and development environment", was used to give students the space to "focus on coding, collaborate, and build" while "minimizing the time spent acquiring, setting up, supporting, debugging, or adding a third-party package to an IDE."[3]. In our case we used google colab which was fairly well suited for our needs however we kept running into issues with failed saves, which occasionally led to us losing entire functions we had spent hours on. If we had used repl.it we would not have had to worry about setting up a shared folder and unwanted deletions, it would have also granted a standard environment for all students as “everyone shares the same compute infrastructure”[3] thus “they all see the same errors and the same output. [3]”. In [3] the project was also split into six parts rather than three. In our case such a distribution would allow students to split larger meetings into smaller ones at more convenient times throughout the semester. As was previously discussed, throughout both milestone one and two I noticed that we were spending more time individually tackling the assigned functions in a very disinterested state where we simply were struggling to properly connect our knowledge through the abstractions of synthesized dna/mathematics. A lessened emphasis on such abstractions would allow students to focus more on the logic of functions which would facilitate more open discussion. [3] concluded that students showed improvement in both opinion of computer science courses and in academic performance; a sentiment that could be echoed by CS 101 students with the implementation of a standardized work environment, lighter workload distribution, and variance of abstraction (ie. not only dna).

Conclusion:

The best way to improve the final project in this course would be to add elements of interactive and collaborative learning. Doing this would help lessen the workload and abstraction overall in the project. This is supported by multiple scholars. In the first article [1], the authors discuss how tools such as multimedia recordings (an aspect of interactive learning) can lead a student down a path of success. The second article [2] discusses a study on the effects of collaborative learning and critical thinking. There was significant evidence to suggest that the two are positively correlated. Finally, the third article [3] focuses specifically on the effectiveness of collaborative and interactive learning and how they relate to computer science. The results showed that this style of learning had great benefits to the students involved. After reading and analyzing these articles, it is clear that improving interactive and collaborative learning in this course would lead to more success for the students, thus improving the project.

References:

[1] El-Qirem IA, Alsmadi AA, Al-lozi E. “Impact of Interactive Education on the Learning Outcomes and Quality Assurance.” *Journal of Higher Education Theory & Practice*, https://search-ebscohost-com.proxy2.library.illinois.edu/login.aspx?direct=true&db=eft&AN=157965351&site=eds-live&scope=site

[2] Gokhale, Anuradha A. “Collaborative Learning Enhances Critical Thinking.” *Journal of Technology Education*, vol. 7, no. 1, 1995, scholar.lib.vt.edu/ejournals/JTE/v7n1/gokhale.jte-v7n1.html?ref=.

[3] M. M. Rahman, M. H. Sharker and R. Paudel, "Active and Collaborative Learning Based Dynamic Instructional Approach in Teaching Introductory Computer Science Course with Python Programming," 2020 IEEE Integrated STEM Education Conference (ISEC), 2020, pp. 1-7, doi: 10.1109/ISEC49744.2020.9280598.