University of Massachusetts Dartmouth

Department of Computer and Information Science

nucleoSLIDE: A Citizen Science Game for the Motif Finding Problem

A Thesis in

Computer Science

by

Allison A. Poh

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

May 2020

We approve the thesis of Allison A. Poh

Date of Signature

Firas Khatib
Assistant Professor, Department of Computer and Information Science
Thesis Advisor

Iren Valova
Professor, Department of Computer and Information Science
Thesis Committee

Clinton Rogers
Full Time Lecturer, Department of Computer and Information Science
Thesis Committee

Xiaoqin Zhang
Graduate Program Director, Computer Science

Haiping Xu
Chairperson, Department of Computer and Information Science

Jean VanderGheynst
Dean, College of Engineering

Tesfay Meressi
Associate Provost for Graduate Studies

Abstract

nucleoSLIDE: A Citizen Science Game for the Motif Finding Problem

by Allison A. Poh

Citizen science is the collaboration of public participation with scientific research in efforts to advance scientific knowledge, address societal needs, and increase science literacy and attitudes in the public. Volunteers collect, process, and analyze data that would otherwise be too time consuming and costly for researchers to handle. One genre of citizen science projects are citizen science games, or games with a purpose. Through gamification, the complexity of problems can be reduced to that of which the general public can understand. *Foldit*, *EyeWire*, and *Phylo* are examples of gamifying the complex problems of folding proteins, mapping neurons, and aligning genome sequences, respectively. In this thesis, a new citizen science game, nucleoSLIDE, is developed to help solve the motif finding problem, where a motif can be defined as some meaningful, unknown pattern of length k hidden in a DNA sequence. The motif finding problem is important for genetic research and involves finding a collection of motifs, one from each sequence, in which the variation in motifs is minimal. This problem is NP-complete, meaning there exists no algorithm with a guaranteed optimal solution at this time. Furthermore, the size of sequences and lack of conservation of motifs in real-world data further complicates the problem. Like other citizen science games, nucleoSLIDE takes advantage of natural human skills, like pattern recognition, to overcome these complications. This thesis demonstrates how the complexity of the motif finding problem, a non-game context, can be reduced through the use of common game design elements, including but not limited to goals, rules, play, and user interface.

Acknowledgments

Foremost, I would like to thank Dr. Firas Khatib, thesis advisor and source of inspiration, for all of his guidance and support. It was his teachings of bioinformatics that led me down this journey. There could not possibly be a better guide for this thesis than him.

I am grateful for Paul Naylor, professional technician of the Department of Computer and Information Science (CIS) at the University of Massachusetts Dartmouth (UMass Dartmouth), for his aid in transferring my thesis to a web server.

My appreciation goes out to all members of the Department of CIS at UMass Dartmouth that have helped me during my undergraduate and graduate studies. I am especially indebted to the help and support of Professor Clinton Rogers and Dr. Iren Valova, both committee members of my thesis.

Table of Contents

List of Figures

List of Tables

Abbreviations

ABC	Artificial Bee Colony
AI	Artificial Intelligence
CIS	Computer and Information Science
CLO	Cornell Lab of Ornithology
CLT	Cognitive Load Theory
DNA	Deoxyribonucleic Acid
EM	Expectation-Maximization
GA	Genetic Algorithm
GEQ	Game Engagement Questionnaire
HCI	Human-Computer Interaction
IEQ	Immersive Experience Questionnaire
IoT	Internet of Things
MFP	Motif Finding Problem
MSA	Multiple Sequence Alignment
NIH	National Institutes of Health
NP	Nondeterministic Polynomial Time
PENS	Player Experience of Need Satisfaction
PSO	Particle Swarm Optimization
SDT	Self Determination Theory
TF	Transcription Factor
TFBS	Transcription Factor Binding Site
UI	User Interface
UMass Dartmouth	University of Massachusetts Dartmouth
UX	User Experience

Chapter 1: Introduction

Citizen science is a powerful tool for enhancing scientific research. By including public participation, citizen science minimizes the efforts and time needed for data collection, analysis, and processing. Often times, algorithmic solutions to scientific problems are too computationally complex or time consuming for researchers to handle alone. Citizen science projects may request participants to give spare computer processing power to researchers, or ask participants to do some active task on or for data.

One seemingly effective approach to encouraging participation in citizen science is through gamification. Gamification is the process of incorporating game elements into some non-game context. Citizen science games present puzzles or challenges that are based on real-world data and science, but are designed in such a way that no scientific background is needed for play.

The motif finding problem (MFP) is one example of a computationally complex problem that citizen science may be able to enhance. Although the problem has been around for some time, it is still one of the most challenging problems faced by biologists. In brief, MFP is the problem of finding some motif, or pattern, of a defined length within multiple DNA sequences. The complexity of the problem lies in the fact that motifs are not conserved, meaning an identical match in each sequence is unlikely. Furthermore, DNA sequences are extremely long and enumeration of all possible motifs is unrealistic. There exists several algorithms for MFP, however the problem is NP-complete.

The aim of this thesis is to enhance MFP through gamification. By creating a citizen science game for MFP, the goal is to utilize natural human pattern recognition skills to enhance motif finding and ultimately genome understanding. When a problem is structured into goals, rules, and play, competition and motivation arises and leads to sustained participation. Through the use of a thoughtfully designed UI, participants will find it easy

and inviting to help research. Additionally, gamifying MFP will increase public literacy in the topic.

Chapter 2 defines citizen science and examines current citizen science games to form a list of game elements for gamifying scientific research. Chapter 3 introduces MFP and examines some existing algorithms. Chapter 4 presents nucleoSLIDE, a citizen science game for MFP, and examines its front-end and back-end components, MFP algorithm, implementation, and evaluation of design decisions. The final chapter provides a conclusion and states potential future work.

Chapter 2: Citizen Science

Citizen science, as defined by the *Crowdsourcing and Citizen Science Act*, is "a form of open collaboration in which individuals or organizations participate voluntarily in the scientific process [1]." The participants comprise of individuals (or other entities) who may or may not have any scientific training. Their contributions can include formulating research questions, conducting experiments, collecting and analyzing data, interpreting results, making discoveries, developing technologies and applications, and solving complex problems. The goals of citizen science are to advance scientific research, address societal needs, and increase STEM literacy in the public [2].

Citizen science games allow participants to solve scientific problems through puzzles and other game-like activities. They provide a "fun" learning environment for the participants while also giving scientists the opportunity to make new discoveries and answer open research questions without exhaustive efforts. They are essentially "framework[s] for harnessing the computing power of mankind [3]." Through gamification and crowdsourcing, citizen science games have led to significant scientific findings. Games like *Foldit* [4], *EyeWire* [5], and *Phylo* [3] are some examples of successful citizen science games.

Section 2.1 discusses the rise of citizen science and its influence on scientific research so far. Section 2.2 specifically addresses citizen science games and how they are used to engage the public in scientific research. Current citizen science games are presented and their core game elements are discussed here.

2.1 The Citizen Science Movement

Citizen science projects have been around since at least the 1800s when William C. Redfield asked a network of observers to record the direction of the wind at a specific time during a storm. In doing so, he proved the existence of cyclonic storms, a concept that was

rejected eight years prior due to his inability to be at multiple places at once [2]. Since then, scientists all over the world have been replicating Redfield's crowdsourcing approach for data collection. Within the past 20 years, the rise of the Internet of Things (IoT) brought about a surge in citizen science projects and ultimately what some consider to be the citizen science movement. IoT has greatly enhanced the ability to share project information, participation protocols, and data collection methods. Participants can access projects through apps and desktop applications. IoT has allowed these projects to expand to disciplines that were once too advanced and difficult to share with the public, such as microbiology and genetics [6].

Bonney *et al*. categorized citizen science projects into four generalized genres based on their goals and participation tasks: data collection, data processing, curriculum-based, and community science [7]. Projects under data collection ask participants to gather information about a particular subject in order to answer a specific question or monitor the subject's environment. Typically, the subjects under analysis are wildlife or vegetation. It has been estimated that there are between 1.36 and 2.28 million participants each year for data collection projects, having a minimum estimated economic worth of $667 million to $2.5 billion [7], [8]. Data processing projects request participants to transcript, categorize, manage, and interpret data that would otherwise be too excessive for scientists to process on their own. Examples of datasets used in these projects include the mapping of neurons and the surface of the moon. In a study of seven data processing projects hosted by *Zooniverse*, a citizen science platform [9], it was found that the projects led to a large amount of work completed at a low cost, estimating total contributions to be $1.5 million in the first 180 days [7], [10]. Curriculum-based projects are different from data collecting and data processing projects as their target audience is primarily K-12 students. Their objective is to meet curriculum goals while maintaining the interests of students. Although the participants perform tasks like data collecting and data processing, the enhancements to scientific research are less powerful in curriculum-based projects than those in the previous categories because their effectiveness relies on the classroom context [7], [11]. The final group of projects are called community science projects and target local

audiences. In contrast to all other genres, community science projects are typically initiated by a citizen or citizens seeking help from scientists. Resultantly, these projects involve citizens in multiple stages of the scientific process and produce greater local impacts. An example of a community service project is the collection of water samples by citizens and the testing of those samples by local scientists [7]. A single citizen science project may fall into multiple categories.

The Cornell Lab of Ornithology (CLO) is known for their bird-related citizen science programs and, based on their successes, have established a model for developing citizen science projects. The idea of the model is to ensure proper protocols are followed by all stakeholders and that the results of the project warrant the need for the project. In the initial stages of development, a scientific question is formulated and a development team of scientists, educators, technologists, and evaluators is assembled. The team then develops the project. Protocols, forms, and supplemental materials are created, tested, and refined at this step to ensure high data quality during participant data collection and submission. Once project development is completed, participants are recruited and trained. The data from the project is collected and edited for scientists and the public, and analyzed and interpreted by the researchers. The final step of the model is to measure the project's impact on science and the public. Most time-independent citizen science projects go through iterations of the CLO model, improving protocols and data handling methods based on feedback from stakeholders and analysis of results [12], [13].

2.2 Citizen Science Games

The goal of gamifying citizen science projects is to engage citizens in data collection, processing, and analysis of less-interesting or complex datasets by satisfying their natural psychological needs of competence, autonomy, and relatedness [14]. Self Determination Theory (SDT) theorizes the motivations of individuals and, in regards to games, hypothesizes that "games provide intrinsic motivation … by giving the players competence in the game mechanisms, autonomy in the execution of their actions, and providing a sense of relatedness to others [15]." As players are presented with new puzzles or challenges,

they acquire new skillsets. With competence comes a natural desire to test and show-off skills, resulting in extended and successive play. Players are also driven simply by the goal of these games, having a moral desire to help scientific research. The connectedness citizen science games create between players and science increases the players desire to learn about the research and could enhance their outlook on science and learning in general. A good citizen science project is one that improves scientific research and the public, both of which are enhanced through gamification [16].

The Player Experience of Need Satisfaction (PENS) model focuses on the execution of SDT principles in games. Through a nineteen-item questionnaire, players evaluate their experience by rating the games ability to provide competence, autonomy, and relatedness. For example, for competence, the player is asked about the intuitiveness of the controls [15]. Other forms of measuring player experience include the Immersive Experience Questionnaire (IEQ) and the Game Engagement Questionnaire (GEQ). IEQ focuses on cognitive development, emotional involvement, real world dissociation, challenge, and control, while GEQ focuses on absorption, flow, presence, and immersion. These questionnaires share many overlaps [17]. Although citizen science games are classified differently than typical video games, these games with purposes must satisfy the players' needs in order to keep engagement.

The complexity of data for some domains makes citizen science tasks difficult to develop for common people. Through games, however, the cognitive skills needed for handling such data can be learned at the players' own pace and desire. Cognitive Load Theory (CLT) describes how people process information. Cognitive load is the amount of effort needed to understand and perform a task. Causing cognitive overload in a player will impact play negatively and could result in their discontinuation of the game. To avoid this, CLT suggests breaking individual elements into cognitive schemas. The idea is that, when focusing on one schema at a time, the information will be stored in long-term memory and allow for fast retrieval [15]. In citizen science games, cognitive schemas can be seen in tutorials and level designs. Interactions are introduced as the player progresses to ensure

understanding of the game's interface and mechanics. Often times, citizen science games will use artificial data in tutorials to ensure the player learns skills and information in a reasonable environment.

Gamification also improves the sustainment of participation in citizen science projects. The competitive aspect of games allows for players to be recognized through scores and leaderboards. Furthermore, smaller goals like completing levels and receiving in-game rewards encourages players to progress. Recognition and a drive to progress encourages repetitive play. Additionally, some games offer team play where players collaborate to solve puzzles. Teams bond and members become dependent on each other, leading to sustainment of groups of players [18].

2.2.1 Impactful Citizen Science Games

Foldit [4] is a 3-dimensional puzzle game for predicting and designing protein structures. Players can move the structure and its components around, as well as apply actions to it like "wiggle" and "shake". The inspiration behind *Foldit* came from another crowdsource project, *Rosetta@home*, when a team at the University of Washington asked people to donate the spare computational capacity of their personal computers. *Rosetta@home* offered a screensaver to visualize the output of its protein folding algorithm, called *Rosetta*, running on participants' computers. Based on the visualizations, people started suggesting better ways to fold the proteins. As a result, *Foldit* launched in May 2008, challenging people to use their spatial reasoning skills to better fold protein structures. Within a few months, it was found that people were not only able to match the algorithm's results, but also outperform *Rosetta* on some puzzles. Within one year, there were 200,000 active players [19], [20]. And, within two years, players began folding previously unsolvable puzzles. The crystal structure of a monomeric retroviral protease, an enzyme found in an AIDS-like virus in rhesus monkeys, was solved in less than ten days [21]. The quick success of *Foldit* proved how powerful and effective citizen science games can be at enhancing scientific research. Screenshots of *Foldit* can be found in Appendix A.1.

EyeWire [5] is another 3-dimensional citizen science puzzle game. The goal is to enhance connectomics, the study of "the network architecture and dynamics of neurons and other cells in the brain [5]." Players color branches of the neuron in a cube by examining its cross-sections. Before launching in 2012, scientists solely relied on artificial intelligence (AI) to generate and analyze neuron data, a multiyear process. Since its release, *EyeWire* players have discovered six new types of neurons in the retina. They also reconstructed previously unknown circuits that could be beneficial to curing vision-related disorders [5]. *EyeWire* has over 200,000 players from 150 countries [22]. The success of *EyeWire* has inspired *Cortex,* a similar game but for mapping neurons and finding synapses in the brain, that is expected to launch by 2022 [5]. Screenshots of *EyeWire* can be found in Appendix A.2.

Phylo [3] is a 2-dimensional puzzle game, launched on November 29, 2010, designed to better the multiple sequence alignment (MSA) problem. MSA is the task of "find[ing] the highest-scoring alignment between multiple strings under a given scoring matrix [23]." The multiple strings are either DNA, RNA, or protein sequences, and the highest-scoring alignment is the arrangement of nucleotides or amino acids that best identify regions of similarity. In gameplay, players must arrange colored dots (representing the nucleotides or amino acids) horizontally, without changing the order of the dots. Gaps and mismatches between dots are allowed but penalized, as they would be when scoring alignments algorithmically [3]. Like structuring proteins or mapping neurons, MSA is computationally complex and no algorithm guarantees an optimal solution. Within two years, 350,000 solutions were submitted by 12,000 users with up to a 70% accuracy improvement on the alignments than solutions produced by the algorithm [24]. Screenshots of *Phylo* can be found in Appendix A.3.

2.2.2 Game Design Elements

Gamification is "the use of game design elements in non-game contexts … to motivate and increase user activity and retention [25]." This section defines common game design elements and uses *Foldit* [4], *EyeWire* [5], and *Phylo* [3] to demonstrate how they can be

incorporated in non-game contexts. All references to figures can be found in Appendix A. A table summarizing the game design elements can be found in Appendix B.

Aesthetics and graphics refers to "the audiovisual language (…) conceptualized, chosen and developed by the designers for the visualization, and the display of the elements involved in the game [26]." Audiovisual components are anything that can be seen or heard and judged aesthetically, such as media, imagery, style, and movements of graphics and components. Aesthetics also describe the desired emotional responses aroused in players. Graphics are visual objects and designs in a game world and user interface (UI) [26], [27]. Since *Foldit*'s puzzles are visually complex, other graphical components are designed with minimum colors. The background is a light, solid color and all UI features are grey and collapsible. The puzzle itself is made up of 3-dimensional graphics, each having fluid movements (Figures 16-17). *EyeWire* is known for its aesthetically pleasing puzzles, with bright strands of color on a dark background, and even has a museum to showcase puzzle "art" [28] (Figures 18-20). In contrast, *Phylo* uses bright imagery and cartoonish graphics to help tell its story (Figures 21-22).

Competition and cooperation engage multiple participants at once. Competition results when players work against each other towards mutually exclusive goals, increasing the feeling of social superiority for the "winning" players. Cooperation is when players work together towards the same or related goals, fostering social relatedness. Both can be symmetric (players or teams perform the same tasks) or asymmetric (players or teams perform different tasks) [27], [29], [30]. *Foldit* is especially known for its competitiveness and cooperation capabilities. *Foldit* players compete against each other to reach the highest score and be placed on the leaderboard. They can compete to be the "Top Evolver" or "Top Soloist" [4]. *Foldit* also allows players to form teams, where members can progress on each other's work. In one study of citizen science games, players expressed that they would only play a game that allowed team play [18].

Framing elements according to the target audience means making sure the play literacy is appropriate for intended players. This includes clear and appropriate challenges, controls, UI, and required skillset. Especially in citizen science games, this includes differentiating between fictitious and factual elements, and game and real-world objectives [26]. Citizen science projects in general require framing the science behind the project in such a way that can maximize participation. *Foldit* and *EyeWire* tackle conceptually hard problems, especially for people with no background in their targeted subjects. However, through the use of colors and shapes, the player is able to differentiate between "good" actions and "bad" actions, minimizing the game's complexity. For example, *Foldit* uses red spiky balls to alert the user of intersecting sidechains, something that causes score deductions (Figure 17). *Phylo* overrides science complexity by styling the game with bright colors, identifiable shapes, and "fun" graphics. It is framed to include younger audiences than *Foldit* and *EyeWire*. All three games use click-and-drag controls and have clear UIs (Figures 16, 18, and 21). Furthermore, any possibility of confusion in the controls or UI are alleviated in their tutorials.

Gameplay defines what a player can do in a game. It is comprised of challenges and the permitted actions to address those challenges. Challenges are nontrivial objectives, direct or optional, that are defined by the rules of the game. The ramification of actions toward challenges, both in-game and emotionally, are defined by the gameplay [25], [27], [31]. For example, in *EyeWire*, the challenge is to trace a neuron from each end of a cube to complete the puzzle by highlighting the correct connecting neuron branches. The actions permitted are rotating the cube and coloring branches in a cross-section of the cube (Figures 18-19).

Goals are essential to all games because they give the player a reason to play. They are defined by the rules and must be clear and attainable [30]. With *Foldit*, *EyeWire*, and *Phylo* being puzzle games, their goals are all the same: complete each puzzle. The rules then define their specific goals. An additional game goal is to reach the highest score of each

puzzle. Beyond the goals defined by the rules are the objective of the citizen science projects, including improving scientific research and enhancing public literacy.

Learning is the process of acquiring knowledge about the game [27]. In citizen science games, this also includes learning about the scientific background of the project. Citizen science games are designed to maximize the amount of participants and therefore do not require prior knowledge of the subject. The data and elements are designed in a way that limits complexity. However, not all parts of a problem can be reduced to no complexity. Furthermore, players tend to desire knowledge of what they are doing. Learning is essential to citizen science games in particular because they inform and please the players [31]. *Foldit*, *EyeWire*, and *Phylo* all provide an About Page that explains the science behind the game through explanations, videos, and references to published works. In terms of learning the games, all three games provide tutorials to teach the terminology and skillsets needed, and puzzle complexity grows as a player's skillset grows.

Novelty is added to games through new, original, or unusual gameplay and features [27]. Elements from traditional games are important to incorporate in citizen science games to maximize play literacy at a faster rate. However, citizen science games are highly novel as they each address specific real-world problems. Folding proteins, mapping neurons, and aligning DNA are all novel game concepts. "Wiggling" and "shaking" a 3-dimensional object, essential *Foldit* actions, is unusual gameplay (Figure 16). As well as scanning a cube and coloring its cross-sections as in *EyeWire* (Figures 18-19).

Play is another essential element of games. Play is a player's participation, including their "freedom to act and freedom to choose how to act [27]." Fizek and Dippel call CS games "laborious playgrounds," noting the blur they create between work and play [32]. *Foldit*, *EyeWire*, and *Phylo* allow the player to experiment as they wish in accordance to the rules. The players must reach a certain score to complete the puzzle, but how they choose to get there is up to the player.

Pretending involves creating a magic circle, or pretended reality, where a boundary is put into place between game concepts, situations, and events and real-world concepts, situations, and events [27]. In the case of citizen science games, the magic circle is sometimes blurred because actions in the game are meaningful in the real-world. However, citizen science games that extract enough science from game objects have a more defined magic circle [32]. For example, in *Foldit*, "wiggling" a protein in the game means the same thing as "wiggling" a protein in the real-world (Figure 17). A defined magic circle example for *Foldit* is the solving of a real-world problem is equivalent to completing a level in the magic circle. In *Phylo*, the player aligns rows of dots but, in the real-world, a scientist aligns nucleotides of DNA (Figure 21).

Risks, rewards, and recognition increases excitement for players. When faced with uncertainty, players must choose to make an action. If the action is correct, then a reward is received (or at least the player is closer to receiving a reward) [27]. Recognition for rewards is a driving factor of continued engagement in citizen science games [18]. Players want to be credited for their work through points, badges, leaderboards, and performance graphs [29]. Puzzles require testing of ideas and therefore naturally come with risks and rewards. *Foldit* dynamically displays scores during actions, giving immediate feedback to whether the reward is worth the risk (Figure 16). Sometimes, an action that would reduce the score is needed in order to perform a much more effective action. *Phylo* puzzles may require a player to slide multiple dots at once even if the intention is to align a single dot (Figure 21). In all games, if the risk was not rewarding, the player is permitted to undo their actions. When the player reaches an accomplishment, *EyeWire* automatically notifies everyone in the chat of that player's achievements (Figure 20). Additionally, recognition can be given to players who have made significant achievements by including them in published papers, such as by *Foldit*, *EyeWire*, and *Phylo* in [21], [33], and [24], respectively.

Rules are part of every game and are defined as "definitions and instructions that the players agree to accept for the duration of the game [27]." They establish goals and meaning to

game concepts, situations, and events. They also define game symbols, gameplay, sequence of play, termination conditions, and metarules [27]. Rules are generally explained through tutorials or in intuitive beginning levels. *Foldit*'s and *Phylo's* tutorials increase the rules as the player progresses, making new actions available at different points (Figure 17-22). In *Foldit*, there are many definitions and instructions for the player to follow and, resultantly, the complexity of puzzles increases fast. *Phylo*, on the other hand, is less complex: a player can move dots horizontally, snapping them into predefined spots, without affecting the order of the dots or exceeding the permitted bounds.

Setting the pace is the player's ability to progress through the game at their own speed [27]. *Foldit*, *EyeWire*, and *Phylo* do not limit the amount of time a player can have on a level and time is not a factor when scoring solutions. During the tutorial, *Foldit* allows the player to repeat instructions numerous times to ensure the needed skills are learned. Instructions do not go away until the user clicks the "ok" button (Figure 17). *Phylo* only adds on to puzzles once the player has completed a row to avoid overloading the player (Figure 21). All three games allow the player to undo or restart a puzzle without repercussions.

Socializing is the act of communicating with players through forums, chats, or other technologies. For complex games, socializing allows players to support one another and build a sense of community. In cooperative games, it allows players to strategize and communicate about task distribution [27]. *Foldit* has three chat options during gameplay: group, puzzle, and global (Figure 16). *EyeWire* also provides a chat and encourages players to use each other for help (Figure 20). In a study of participant motivations in citizen science games, it was found that forums and chats are a main motivation for play [18].

Storytelling is sharing background information, rules, and goals to players through narration or other expressive means. It boosts entertainment, clarifies complex actions or events, and increases immersion [27]. There are two levels of storytelling: the story designed into the game and the story created by the player [31]. For citizen science games, storytelling designed into the game explains to the player the setting, back story, and

problem that their gameplay will help solve. It provides context to challenges and actions beyond the goal of obtaining high scores or leaderboard recognition [26], [29]. The player's story is the player's adaptation of the designed story, built off their actions and interactions with the game [31]. *Phylo* has a story mode that uses text-narration and graphics to portray the goal of the game in a user-friendly manner. The story is that the player is on a spaceship, flying around Earth to compare DNA sequences of various animals. After completing levels, the spaceship is shown flying to the next level, symbolizing progression (Figure 22).

User experience (UX) is the player's emotions and attitudes to game design decisions. Some argue that the principle goal of game design is to reach ultimate UX through entertaining gameplay [31]. As mentioned earlier, this can be measured through questionnaires like PENS, IEQ, and GEQ.

UI "encompasses everything the user sees, hears, and interacts with and how that interaction happens [31]." It should not distract the player from gameplay, storytelling, or learning, but rather make entertainment accessible. *Foldit*, *EyeWire*, and *Phylo* are similar in that their UIs display scores, progress, and the puzzle. Additionally, *EyeWire* features include a leaderboard, activity feed, real-time chat, and game commands [34] (Figure 20). *Foldit* also displays a real-time chat and leaderboard in addition to a panel of permitted actions, undos, and modes (Figure 16).

Chapter 3: The Motif Finding Problem

In simplest terms, motif finding is the "problem of discovering some 'hidden message' shared by a collection of strings [35]." Biologists face this problem when studying the regulation of gene expressions. Within DNA sequences are short DNA intervals called regulatory motifs that signal to transcription factors (TFs), or regulatory proteins, to bind there. Once bound, the TFs can regulate the genes by turning them on and off. Discovering regulatory motifs tells biologists where the transcription factor binding sites (TFBSs) are located, and is a primary step for studying how genes function [35], [36].

Due to the enormous length of DNA sequences, biologists rely on algorithms to find these binding sites. However, MFP is a challenging problem even for algorithms. Biologists are able to provide DNA sequences and an estimated motif length (although this sometimes requires trial-and-error [37]), but aren't able to tell the algorithm what the motifs look like in advance. Furthermore, due to mutations and other errors, regulatory motifs are not completely conserved and can vary at some positions in different sequences [35]. The search for "hidden messages" gets harder when there are variations of the message.

Chapter 3 explains MFP from a bioinformatics standpoint. Biology definitions and examples will be provided as needed, but the information presented here is intended to clarify the inputs and outputs for algorithmic understanding. Section 3.1 shows how this biology problem can be transformed into a computational problem. Section 3.2 discusses common algorithms studied in previous research. The chapter concludes with discoveries resulting from motif finding.

3.1 Computational Breakdown

MFP is defined as: given a collection of strings and some integer k, find an unknown pattern of length k that occurs in each string with minimal variations [35]. If the pattern appears exactly in every string, then the output is simply that pattern. If there is no one pattern that

appears in every string, then a collection of patterns, one from each string, is found and reduced to a consensus pattern. The latter is the more common outcome when performing MFP on DNA sequences.

How does this definition translate to a biology problem? Consider a DNA sequence to be a string of nucleotides, where a nucleotide is some character of value ‘A’, ‘C’, ‘G’, or ‘T’. Each DNA sequence has a different order of nucleotides and has no significance in ordering (from a computational perspective) except for some hidden motif of length *k*, where *k* typically ranges from 5 to 20 [38]. A motif thus can be defined as a short pattern of nucleotides that has biological significance. Each motif within a collection of motifs is called a *k*-mer. MFP can now be defined as: given a collection of DNA sequences and some integer *k*, find a collection of *k*-mers, one from each sequence, in which variation is minimal [35].

When each sequence has the same motif, enumeration of all possible *k*-mers within the sequences will lead to a correct output (but at a slow speed especially for large *k* values) [35], [38]. Yet, when looking for a collection of motifs, there needs to be a way to determine which *k*-mers have the least variation. This can be done with motif scoring.

Scoring motifs involves “select[ing] a *k*-mer from each string and [scoring] these *k*-mers depending on how similar they are to each other [35].” Imagine placing the *k*-mers into a table, where each cell of the *i*th row consists of one nucleotide of the *i*th *k*-mer. This table is referred to as a motif matrix. A consensus string (or consensus pattern [39]) can be formed by selecting the nucleotide with the highest count per each column. The consensus string represents a possible motif for binding and has a length of *k*. The score of the collection of *k*-mers is the addition of all nucleotides from the *k*-mers that do not match the consensus string. For example, if the first column consists mostly of ‘A’s, then the first nucleotide of the consensus string is ‘A’ and the score is the amount of *k*-mers that do not have ‘A’ in their first position, or the amount of cells with a ‘C’, ‘G’, or ‘T’ in the first column. Counting nucleotides is done for each column, adding the most frequent

nucleotide to the consensus string and adding the count of all other nucleotides to the score. The consensus string with the smallest score is the best solution [35]. Figure 1 shows how to calculate the score for ten 10-mers column-by-column. Highlighted characters represent the nucleotides with the highest frequency for each column. Alternatively, if a possible consensus string is known beforehand, the score can be counted by comparing each *k*-mer to the consensus string and counting the mismatches. This is equivalent to counting mismatched nucleotides row-by-row rather than column-by-column. The difference between a *k*-mer and a consensus string is known as the Hamming distance, and the sum of Hamming distances within a collection of *k*-mers is equal to the score [35].

Motifs		A	C	T	T	G	C	a	A	A	a
		A	t	T	T	G	C	T	g	A	C
		A	C	T	g	G	C	T	A	A	C
		c	g	T	T	G	C	T	c	t	C
		A	C	T	T	G	C	g	g	A	C
		t	C	T	a	G	C	T	A	t	C
		A	C	T	T	c	C	T	c	A	C
		A	a	T	T	c	C	g	A	t	t
		A	C	T	T	G	C	a	A	A	t
		t	C	T	T	G	C	g	A	t	C
Score(Motifs)		3 +	3 +	0 +	2 +	2 +	0 +	5 +	4 +	4 +	3 = 26
Profile(Motifs)	*A:*	0.7	0.1	0.0	0.1	0.0	0.0	0.2	0.6	0.6	0.1
	C:	0.1	0.7	0.0	0.0	0.2	1.0	0.0	0.2	0.0	0.7
	G:	0.0	0.1	0.0	0.1	0.8	0.0	0.3	0.2	0.0	0.0
	T:	0.2	0.1	1.0	0.8	0.0	0.0	0.5	0.0	0.4	0.2
Consensus(Motifs)		A	C	T	T	G	C	T	A	A	C

Figure 1: Example of calculating the score, profile matrix, and consensus string for a collection of 10 motifs. Highlighted characters represent the most frequent nucleotide for each column. An alternative way to calculate the score is to count the mismatches row-by-row instead of column-by-column. Both ways will result in a score of 26. Figure created by author of thesis.

The frequency of each nucleotide can also be formed into a matrix. For each column, the number of 'A's, 'C's, 'G's, and 'T's are individually tallied and divided by the total number of *k*-mers. The results are put into the first column of a 4x*k* profile matrix (or positional weight matrix [36], [39]). (Note: This technique goes against Cromwell's Rule that probabilities of zero and one should be avoided. For simplicity, this is overlooked in this

section. See section 4.2.2 for profile matrices with pseudocounts). The nucleotide associated with the highest value for each column is the nucleotide added to the consensus string [35]. Figure 1 gives an example of a profile matrix for ten 10-mers.

The above scoring system does not differentiate between *k*-mers with identical consensus strings and scores, but different variations of nucleotides. For example, in Figure 1, the nucleotides in columns 8 and 9 have identical outcomes but not identical nucleotide variations. Column 8 has three possible nucleotides, (A', 'C', or 'G') while column 9 has only two possible nucleotides ('A' or 'T'). A modified approach to scoring is to include the measure of uncertainty, called entropy, for each column. In the given example, column 9 would have a lower entropy than column 8 because there is less uncertainty that the consensus nucleotide should be 'A'. Under this approach, the score is equal to the sum of entropies [35].

The process of finding *k*-mers and determining their consensus strings and scores requires an iterative approach. There are several algorithms for finding motifs and the most common are explored in the next section. However, it is important to note that MFP is NP-complete and lacks an efficient known solution. The existing algorithms do not consistently find optimal solutions, but are good approximations [39].

3.2 Motif Finding Algorithms

Hashim *et al*. classifies existing motif finding algorithms into four categories: enumerative, probabilistic, nature-inspired, and combinatorial [36]. Enumerative algorithms search the whole search space for all, or some calculated amount, of possible solutions. Probabilistic algorithms, on the other hand, utilize matrices to weigh solutions. Nature-inspired algorithms are newer and follow an evolutionary approach, and combinatorial algorithms are a combination of multiple algorithms. Most existing algorithms fall in either the enumerative or probabilistic categories.

3.2.1 Enumerative Approaches

Enumerative approaches run in exponential time. The most conceptually basic motif finding approach is to search the whole search space for all possible *k*-mers. Known as the brute force approach, every possible collection of *k*-mers within the sequences is scored and the collection of motifs with the smallest score is returned. If there are *t* sequences with *n* nucleotides each, the overall running time is $O(n^t \cdot k \cdot t)$ [35]. Not only is this brute force approach inefficient, but also unrealistic to use on actual DNA sequences due to their size and variability [36].

A second brute force approach is to search for all possible consensus strings within the sequences. The idea is to find a consensus string (called a median string here) that has the minimum Hamming distance among all possible *k*-mers, where the Hamming distance is equivalent to the sum of mismatches. The overall running time for searching for the best median string is $O(4^k \cdot n \cdot k \cdot t)$. Since *k* is always a small value (no larger than 20 [38]) and *t* is always large (at least in the thousands), this brute force approach is more efficient than the previously mentioned one [35].

Other enumerative approaches increase efficiency by preprocessing the data into search data structures. A suffix tree can be used to spell out every possible *k*-mer within all sequences, where a unique path from root to leaf represents a possible *k*-mer and has a depth of *k*. Each node additionally keeps track of the amount of times that node has been reached, and paths with leaves with the highest values are checked as potential solutions [36], [40]. Similarly, graphs can be used to represent all possible *k*-mers, where each vertex is one *k*-mer. If two vertexes share an edge, the *k*-mers are similar within some acceptable difference. Each *k*-clique represents a collection of *k*-mers that could have the minimum score [36], [37]. Hashing can also be used to sift through *k*-mers by hashing possible *k*-mers into buckets and limiting the need of scoring to only those buckets with large quantities [41].

3.2.2 Probabilistic Approaches

Probabilistic approaches use profile matrices [35] or positional weight matrices [36], [39] to separate valid motifs from invalid motifs. The most general category of probabilistic approaches are those based on the expectation-maximization (EM) algorithm. EM is an iterative algorithm for refining parameters. For MFP, EM requires two steps: (1) estimate scores based off the current matrix, and (2) refine the matrix using those estimated scores. The algorithm ends when the method converges to a minimum scored collection of *k*-mers [36], [42]. A greedy motif search algorithm iteratively compares *k*-mers from the first DNA sequence to *k*-mers from all other sequences and chooses the best collection at each iteration. For each *k*-mer of the first sequence, a profile matrix is formed for each other sequence based on their *Profile*-most probable *k*-mer, where *Profile* refers to the profile matrix. The *Profile*-most probable *k*-mer is the *k*-mer that was most likely created by the profile matrix. If the score of the matrix using the *Profile*-most probable *k*-mer is better than the score of the previous profile matrix, the new one replaces the previous one. That is, the previous profile matrix is maximized based on the expected score of the *k*-mers [35].

The greedy motif search algorithm can be changed into a random motif search algorithm by choosing *k*-mers randomly. The randomized motif search algorithm is a Monte Carlo algorithm and allows for faster speed but at the cost of accuracy [35], [36]. One hindrance to accuracy is the allowance of changing multiple *k*-mers in a single iteration. Gibbs sampling, on the other hand, is an iterative algorithm similar to the randomized motif search but only allows for one *k*-mer to change at a time. However, accuracy is still not optimal. Randomized motif search and Gibbs sampling both require many runs to find the best scoring collection of *k*-mers [35].

3.2.3 Nature-Inspired Approaches

Nature-inspired approaches use evolutionary algorithms inspired by nature. Genetic algorithms (GAs), inspired by Charles Darwin's theory of natural evolution, reduces the amount of searching by discarding any "offspring" *k*-mers that are not similar enough to

their "parent" [36]. Particle swarm optimization (PSO) algorithms are based on the behaviors of animals in flocks, where each possible solution is represented as a point in a multi-dimensional space. "Particles" adjust their velocity each time they approach a better point [36], [43]. An artificial bee colony (ABC) algorithm is based on a bee swarm, dividing the workload into groups of "bees" (employed, onlookers, and scouts) in order to increase efficiency [36], [44]. The common theme of nature-inspired algorithms is that they are flexible. GA, PSO, and ABC are just a few of the new approaches to solve MFP that are inspired by biology themselves.

3.3 Significance

MFP is one of the greatest challenges and most widely studied problems in bioinformatics. Motif discovery is significant because it allows for an understanding of the regulation of gene expression. The discovery and understanding of the *lac* operon arguably marked the beginning of motif finding studies. The *lac* operon is a group of genes in *E. coli* with a single promoter. A promoter region is the portion of DNA that proceeds the binding site. It was observed that the genes produced a significant amount of enzymes according to environmental factors. After looking at *E. coli*'s sequences, it was found that genes were regulated when there was binding at their TFBSs. DNA sequencing and motif finding methods became significant studies after this finding [42], [45].

Another major discovery based on motif finding is the discovery of circadian clocks. Most living beings have regulatory genes that work in solar time. These genes' TFs bind to TFBSs of other genes to turn them on or off. Discovering TFBSs of circadian regulatory genes have helped scientists understand the circadian clocks of plants, animals, and bacteria. We now understand when plants undergo photosynthesis or begin flowering. We also understand health related illnesses like delayed sleep phase disorder in humans [35].

Chapter 4:
nucleoSLIDE: A Citizen Science Game for the Motif Finding Problem

The proposed solution to enhancing genomic research and public knowledge of MFP is nucleoSLIDE, a 2-dimensional citizen science puzzle game that utilizes the human's natural pattern recognition skills to find motifs in DNA sequences. Players are tasked with finding some common pattern of a defined length in each sequence with minimal variation, and sliding this pattern into the center of the puzzle. A perfect solution is one that matches (or perhaps exceeds) the algorithm's solution. Puzzles are created using real DNA sequences from the National Institutes of Health (NIH) genetic sequence database, GenBank [46], and solutions represent potential TFBSs for those sequences. nucleoSLIDE was developed for the web.

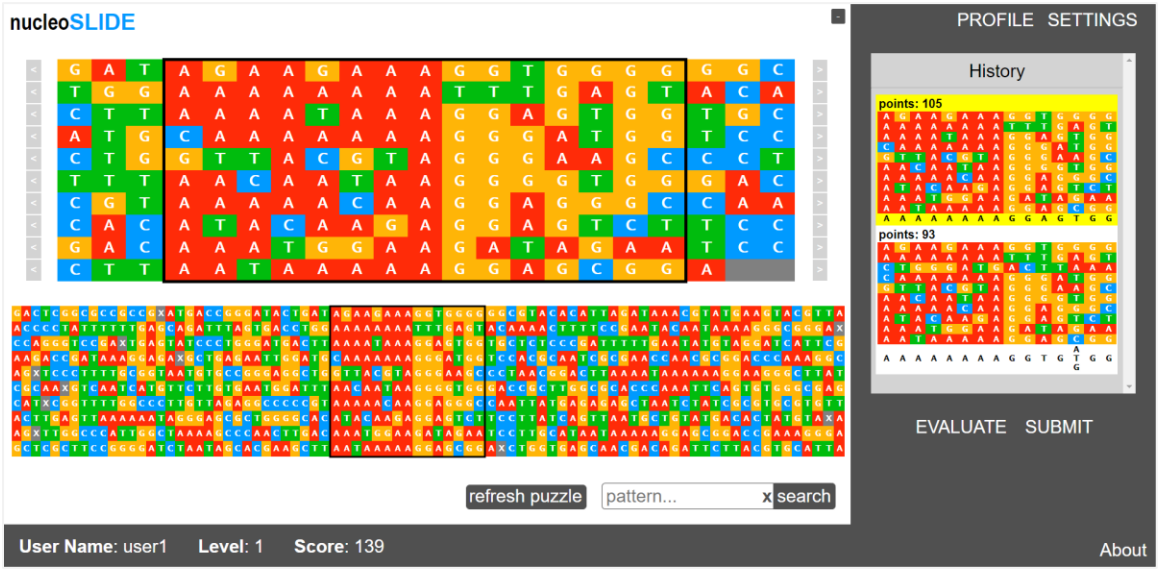

Figure 2: nucleoSLIDE, a citizen science game for the motif finding problem.

4.1 Frontend

nucleoSLIDE consists of three web pages: Login/Sign Up Page, About Page, and Game Page. The UI components of each page are examined in this section.

4.1.1 Login/Sign Up

Standard login and sign up UI designs are used. When the player arrives, they are prompted for their username or email address and password (Figure 3). If they do not have an account, they can select “sign up” and enter new user credentials. Additionally, the player has the ability to visit the game’s About Page before creating an account (see section 4.1.2 for About Page).

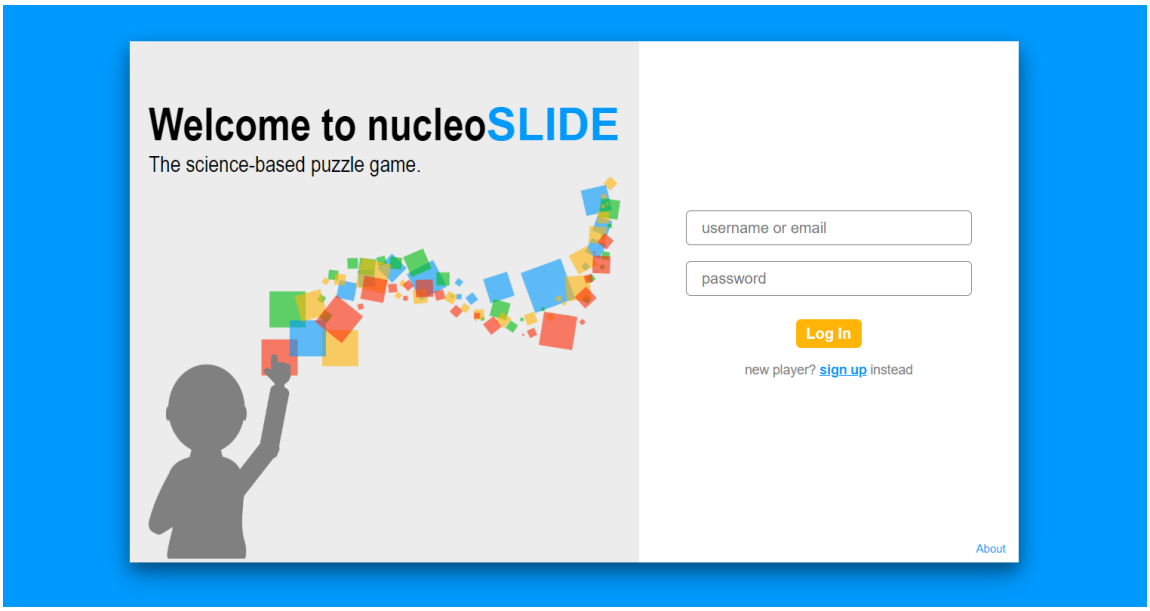

Figure 3: nucleoSLIDE’s Login Page.

4.1.2 About

The purpose of the About Page is to inform the player, or potential players, about the game’s objective. Although no science background is required for play, this page explains

MFP, the challenges it poses, and the potential impact of nucleoSLIDE on scientific discovery. Figures 4 and 5 show the introduction and MFP sections of the About Page, respectively.

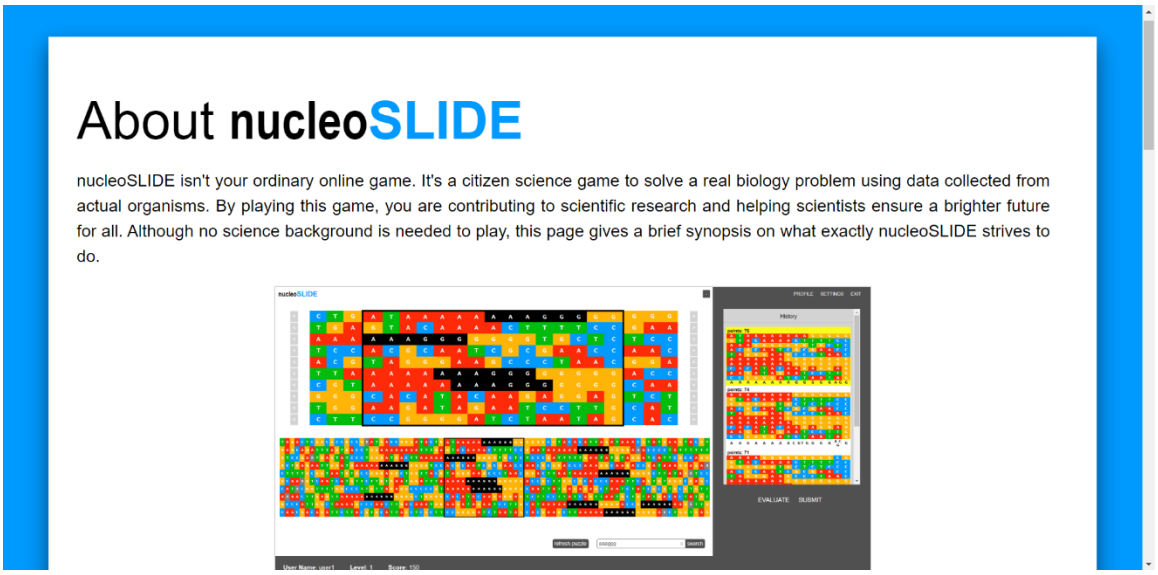

Figure 4: Introduction section of nucleoSLIDE's About Page.

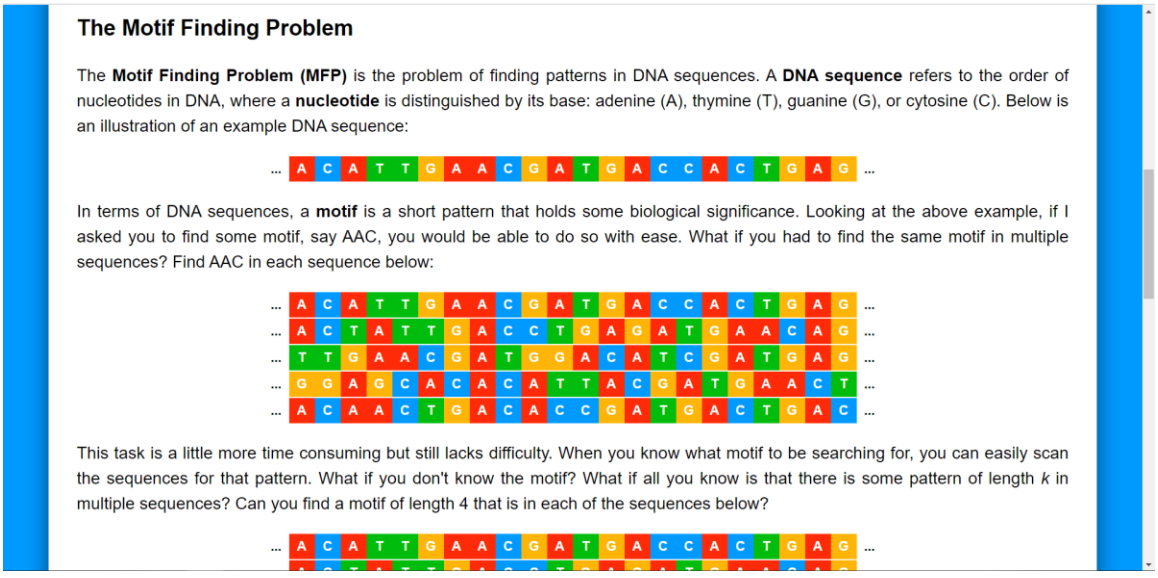

Figure 5: The Motif Finding Problem section of nucleoSLIDE's About Page.

4.1.3 Game

nucleoSLIDE's game UI consists of three core sections: workspace, status bar, and side panel. The workspace is where the player can actively interact with the sequences. The status bar provides information about the current level and the player's score. The side panel gives capabilities for evaluating, submitting, and saving or revisiting workspaces. Descriptions of each section are provided below.

4.1.3.1 Workspace

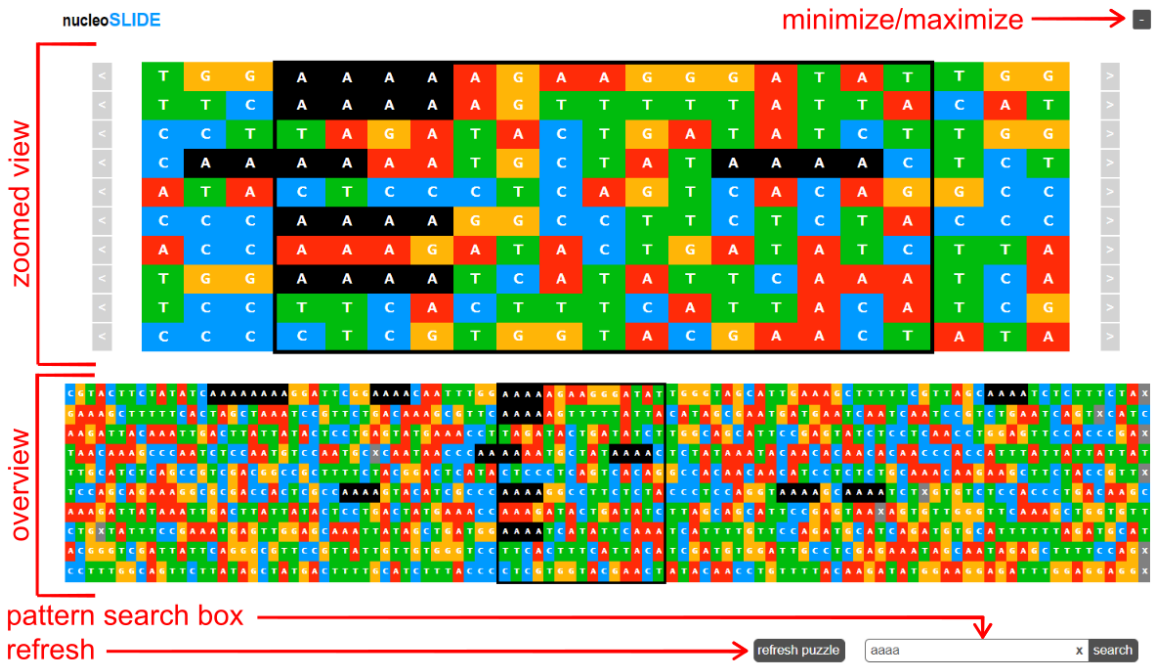

Figure 6: Breakdown of the workspace section of nucleoSLIDE's Game Page.

The purpose of the workspace is to present the puzzle and provide actions for the player to complete the puzzle. The totality of the puzzle is presented in the overview section. Each row represents a sequence, and each block a nucleotide (except for a single block in each row that represents the end of the sequence). The overview section of the puzzle in Figure 6 shows that the puzzle consists of ten sequences, each with a length of 100 nucleotides.

The color scheme for the nucleotides are red for adenosine, blue for cytosine, yellow for guanine, and green for thymine (color schemes can be changed through settings; see 4.1.3.3 for settings), and the labeling of nucleotides is ‘A’, ‘C’, ‘G’, and ‘T’, respectively. The end-of-sequence blocks are colored grey and labeled ‘x’. Blocks colored black represent found patterns and are explained below.

In the middle of the overview is a boxed window, represented with bold, black borders and encompassing a portion of each sequence. This is the pattern-matching window. The goal is to find a pattern amongst all sequences that has the length of the pattern-matching window. However, in most cases, there will not be an identical pattern in each sequence. Therefore, the goal becomes finding a consensus pattern that least varies from sequence to sequence. For example, Figure 7 shows the overview of a solved puzzle. There is no pattern of length 15 (the length of the pattern-matching window) that appears in each sequence exactly once. However, there can be found some variation of a consensus pattern, ‘AAAAAAAAGGGGGGG’, in each sequence. The solution is the collection of closest variations to the consensus pattern found in each sequence, and each variation is placed into the pattern-matching window for scoring (see 4.1.3.3 for scoring).

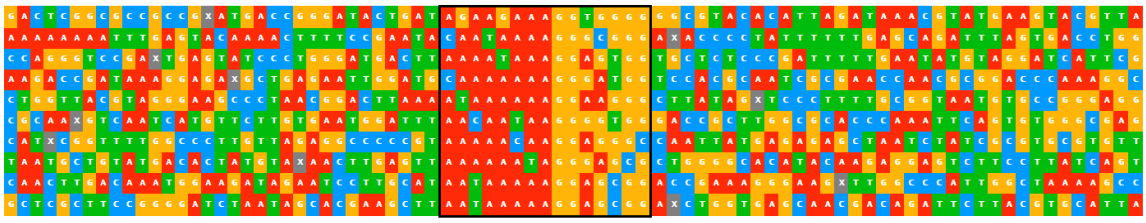

Figure 7: Overview of a solved nucleoSLIDE puzzle. Consensus pattern is ‘AAAAAAAAGGGGGGG’.

The zoomed view shows identical information as the overview, just focused on the pattern-matching window. The zoomed view is where the player performs sequence sliding interactions. In Figure 6, arrow buttons can be seen to the left and right of each sequence. Clicking and holding these buttons will slide the sequence in its respective direction (see

4.1.3.3 for more interactions regarding sequence sliding). The overview is dynamically updated with the zoomed view.

The minimize/maximize button allows the workspace to expand and contract by toggling visibility of the side panel (see 4.1.3.3 for side panel). This is especially useful for puzzles with large sequence lengths. Figure 8 shows a comparison of a regular-size overview and an expanded overview for a puzzle with large sequence lengths.

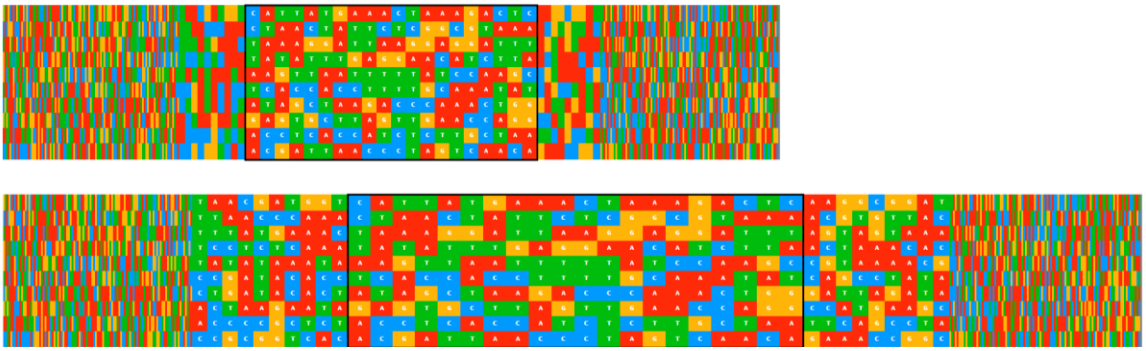

Figure 8: Regular-sized overview (top) versus expanded overview (bottom) of a puzzle with sequences of length 250.

The pattern search box enhances the player's ability to find patterns, or subpatterns, speedily. The player can type a string of nucleotides (where inputted string *w* belongs to alphabet Σ ={a, A, c, C, g, G, t, T}) into the search box. Upon clicking 'search,' all blocks forming the inputted pattern are highlighted in black. For example, in Figure 6 the player inputted "aaaa". Any block within a succession of four blocks labeled 'A' are highlighted black. The player can select any highlighted box to slide its row so that the selected pattern is inside the pattern-matching window. Selected patterns are placed adjacent to the left wall of the window. The search box includes a clear button labeled 'x' to restore appropriate coloring of each highlighted block.

The last component of the workspace UI is the refresh button. The refresh button resets the puzzle, sliding all rows into their original positions and restoring block colors. Refresh does not clear saved workspaces in the history display (see 4.1.3.3 for history display).

4.1.3.2 Status Bar

The status bar is a simple section of the UI and lies below the workspace. It presents the player's user name, level number, and total score of all points earned in previous levels. The source of the sequences is also provided, in simple terms and scientific terms, to give the player some reference as to what they are working on (though this information is not necessary for gameplay). Figure 9 depicts the status bar for *newuser*, who is on level four with a score of 412.

Figure 9: Status bar of nucleoSLIDE's Game Page.

4.1.3.3 Side Panel

The side panel lies on the right side of the Game Page. It consists of profile and settings dropdowns, a history display, evaluate and submit options, and a link to the About Page. All components of the side panel are described in this section. Figure 10 illustrates the side panel and associated windows.

The profile window displays information about the user's accomplishments. Levels cleared refers to the number of levels the user has completed, including the tutorial (or level 0). When a player completes a level, if their score is high enough, they can receive a badge (see section 4.1.3.4 for badges). The count of badges earned and the number of them being Gene-ius badges (the highest badge) are displayed in the profile, in addition to the player's rank.

Figure 10: Side panel of nucleoSLIDE's Game Page. Profile and settings windows expand on click.

The settings window allows the player to customize gameplay. Interaction refers to the actions permitted to slide sequences. These include 'click', 'arrow keys', and 'drag'. Figure 10 shows how the interactions affect the display of the overview in the workspace. Sensitivity refers to how much effort is needed to perform interactions. These include 'low', 'normal', and 'high'. For example, an interaction set to 'click' and a sensitivity set to 'high' would result in a fast slide of sequences when the player clicks and holds an arrow button. Lastly, color refers to the coloring of the blocks. Available color schemes are based

on color schemes used by existing DNA sequence software [47]. Figure 12 shows each available color scheme offered by nucleoSLIDE.

Figure 11: Interactions offered by nucleoSLIDE: ‘click’ (left), ‘arrow keys’ (middle), and ‘drag’ (right).

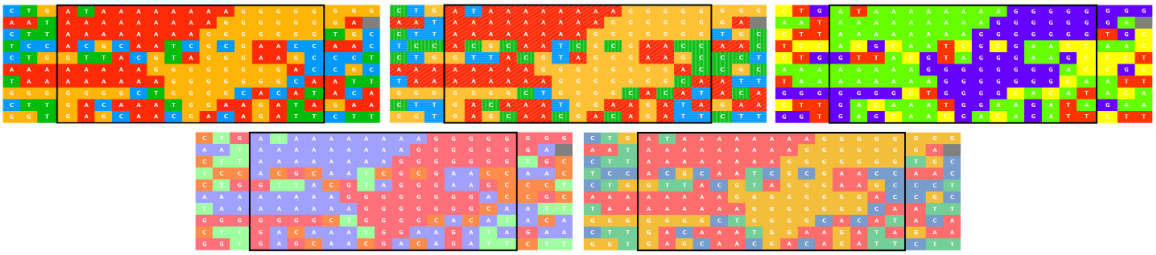

Figure 12: Color schemes offered by nucleoSLIDE.

The history display showcases all evaluated workspaces. The workspaces are ordered chronologically, with the first being the most recently evaluated workspace. The workspace with the highest points (see section 4.2.2 for points and scoring) is highlighted in yellow. Each workspace displayed is focused on its pattern-matching window, as this is the only information that affects the score. Additionally, the consensus pattern for each workspace is presented here. If a player selects a saved workspace, the player’s current workspace is updated to the saved workspace. Figure 10 shows two workspaces in the history display, with the top workspace being the most recently added and the one with the highest point value.

Below the history display are options to evaluate the current workspace and submit the current workspace. Selecting evaluate will store the current workspace at the top of the history display and determine its point value and consensus string. Selecting submit will

also evaluate the workspace and compare it to the algorithm's solution (see section 4.2.2 for nucleoSLIDE's MFP algorithm). If the workspace's point value is within range of the algorithm's, the player has the option to move on to the next level. If not, or if the player chooses to continue working on the puzzle for a solution with a higher point value, the workspace is added to the history display.

Lastly, a link to the About Page is provided in the side panel to give the player information about MFP if they become interested during gameplay.

4.1.3.4 Miscellaneous

There are a number of UI components of the Game Page that do not comfortably fall into any of the main sections described above, as they either incorporate all three sections or overlay the page. For instance, when the user submits their current workspace, one of four pop-ups appear based on the workspace's point value. If the player submits a workspace that is not within 85% of the algorithm's solution, they receive a "You're not quite there yet" overlaid pop-up and must continue working on the puzzle before progressing to the next level. When their solution is within range, the player receives a "Great work!" overlaid pop-up and has the option to continue working on the puzzle or move on to the next puzzle. If the point value is within 90%, the player receives a Motif Finder Badge and, if the player's solution matches (or perhaps exceeds) the algorithm's solution, they receive a Gene-ius Badge. All four overlaid pop-ups can be seen in Figure 13.

The tutorial level, or level 0, uses pop-ups with and without overlays to communicate information about the game and instructions on how to play. Before letting the player interact with any Game Page features, they must read a series of brief pop-ups. They are then walked through an interactive tutorial on the features of the Game Page and how those features can help with game play. Examples are shown in Figure 14.

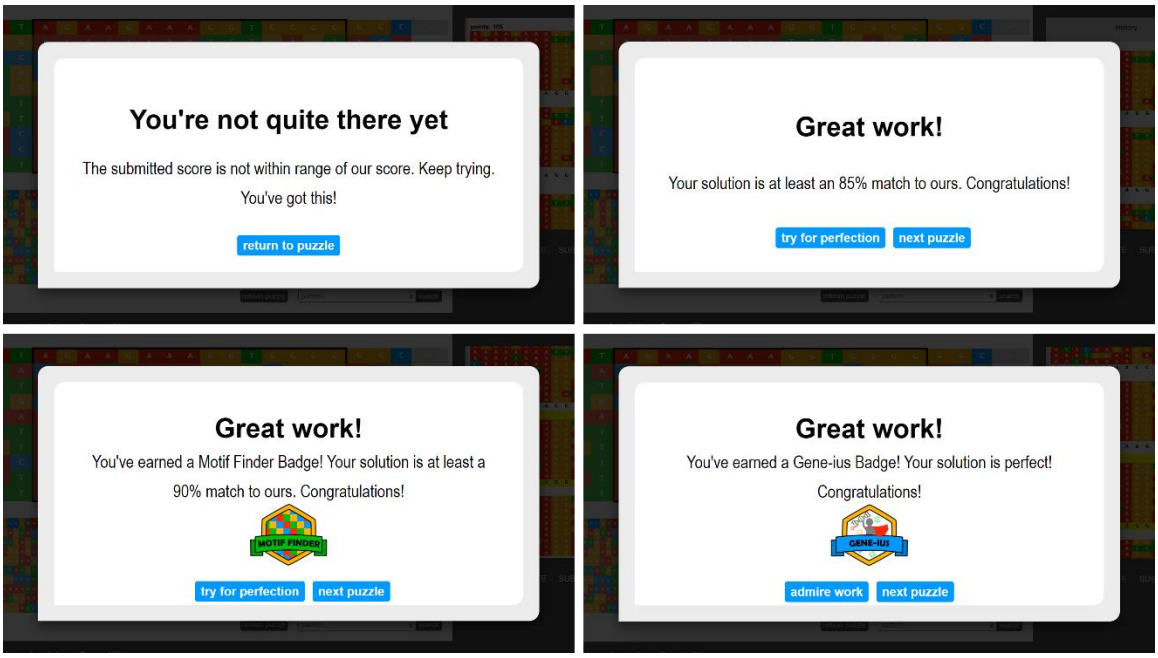

Figure 13: Overlaid messages when player attempts workspace submission.

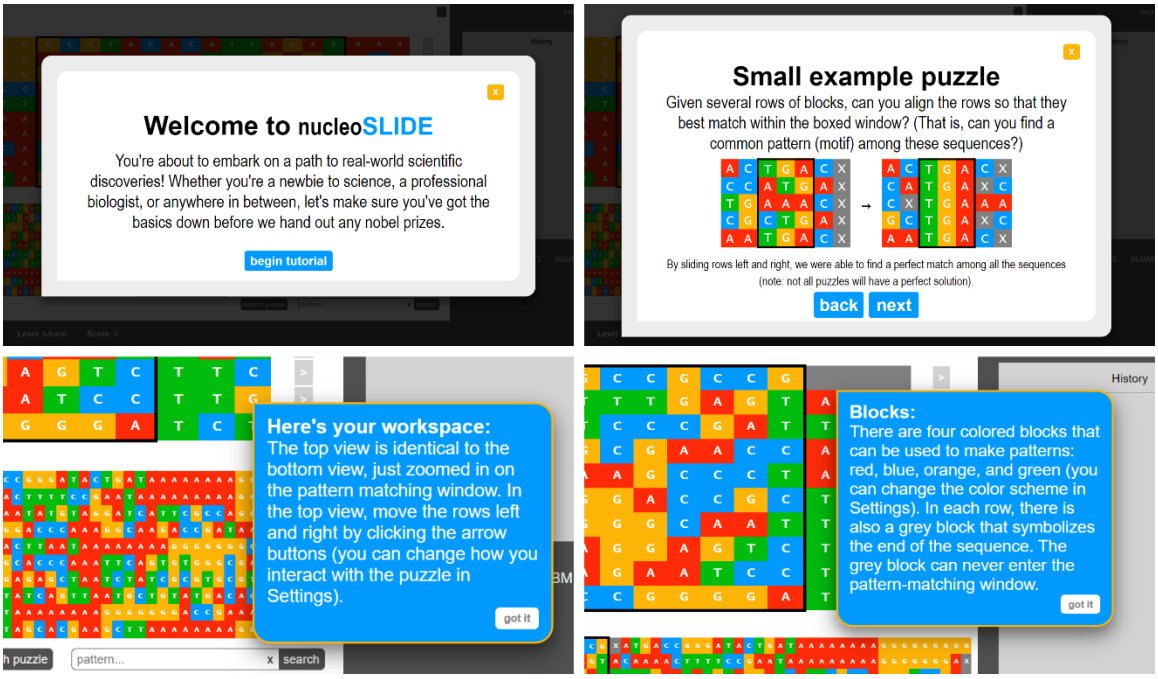

Figure 14: nucleoSLIDE's tutorial. Overlaid informational pop-ups at the start of the tutorial (top) and interactive pop-ups to teach game play (bottom).

4.2 Backend

The backend of nucleoSLIDE includes all interactions with the database and the running of the MFP algorithm. Since submitted workspaces are scored following a similar scoring process as the algorithm, describing the backend process to calculate workspace point values would be redundant.

4.2.1 Database

nucleoSLIDE uses an SQLite database located on the web server. Queries are made when a player logins or signs up, moves on to a new level, and exits the Game Page. The database consists of three tables: results, sequences, and users. Table results is only written to and contains submitted solutions, including the level number, score, amount of attempts, solution, and userid. Table sequences is only read from and is filled with real DNA sequences from GenBank [46]. Table users contains each player's username, email, and password, as well as their current level, score, settings, state of workspace, history, number of badges, and number of Gene-ius badges. Table users is read from when a player logins and wrote to when a player moves on to a new level or exits the game.

4.2.2 Algorithm: Greedy Motif Search with Pseudocounts

nucleoSLIDE runs a greedy motif finding algorithm called Greedy Motif Search with Pseudocounts. The idea of a greedy motif search algorithm was briefly introduced in section 3.2.2. This section will give a more in-depth explanation of the algorithm and explain how it is used in nucleoSLIDE. All references to lines of code are referring to the algorithm's pseudocode provided in Figure 15.

Before addressing Greedy Motif Search with Pseudocounts as a whole, it is important to define the inputs, outputs, and any references to other algorithms or rules. The inputs include a collection of strings *Dna*, integer *k*, and integer *t*, where *k* is the motif length and *t* is the number of strings in *Dna*. A *k*-mer is a (*k*,*d*)-motif, where *d* is the maximum number of mismatches the *k*-mer has to all other *k*-mers in a collection of motifs. *Motif* is a single

motif, where $Motif_i$ represents a motif in Dna_i. *Motifs* is a *t*x*k* motif matrix that represents a collection of motifs, where each row represents one *k*-mer and each cell one nucleotide. *Profile* is a 4x*k* profile matrix, where each row represents a nucleotide and each column a probability distribution of the nucleotides based on the motif matrix. Under Laplace's Rule of Succession, probabilities are adjusted using pseudocounts to avoid probabilities of zero because, according to Cromwell's rule, extremely unlikely events should not be oversimplified and ruled out. Therefore, each count of nucleotides in each cell of the profile matrix is increased by one, and the value of possible outcomes is adjusted to eight (rather than four) to account for the addition of four one's per each column (this is not depicted in Figure 1 of section 3.2.2). A *Profile*-most probable *k*-mer is a *k*-mer that was most likely to have been generated by *Profile* based on the probability distributions.

BestMotifs is a motif matrix of the first *k*-mers in each string from *Dna* (Figure 15: line 2). It is this motif matrix that is updated with each iteration (line 10) and returned in the end (line 11). The algorithm iterates over Dna_1, trying each possible *k*-mer as the first motif (lines 3-4). For each successive string Dna_i, the *Profile*-most probable *k*-mer of Dna_i is found and stored into $Motif_i$ (lines 5-7). *Motifs* becomes the collection of *Profile*-most probable *k*-mers and its score is compared to the score of *BestMotifs*. If the score is higher, *BestMotifs* is replaced. Iterations continue for all *k*-mers from Dna_1. Once returned, *BestMotifs* can be used to quickly determine the consensus string and score of a puzzle.

```
1  GREEDYMOTIFSEARCHWITHPSEUDOCOUNTS(Dna, k, t)
2      BestMotifs ← motif matrix formed by first k-mers in each string from Dna
3      for each k-mer Motif in the first string from Dna
4          Motif_1 ← Motif
5          for i = 2 to t
6              apply Laplace's Rule of Succession to form Profile from motifs Motif_1, ... Motif_i-1
7              Motif_i ← Profile-most probable k-mer in the i-th string in Dna
8          Motifs ← (Motif_1, ..., Motif_t)
9          if SCORE(Motifs) < SCORE(BestMotifs)
10             BestMotifs ← Motifs
11     return BestMotifs
```

Figure 15: Pseudocode for Greedy Motif Search with Pseudocounts. Pseudocode from [35].

To score *Motifs* and *BestMotifs* (line 9), the Hamming distance of each row of the motif matrix are calculated and summed. The hamming distance is the number of mismatches between each row and *Motifs*. The sum of hamming distances represents the score, where a smaller hamming distance means better conserved motifs. The algorithm and submitted workspaces are scored identically, however the point value that the player sees is not this same value. Rather than telling the player to shoot for the least amount of points, which is counter intuitive in games, the player is told to find a solution with the most amount of points. In this case, points is calculated by counting the number of matches in the motif matrix.

Greedy Motif Search with Pseudocounts was selected for nucleoSLIDE for several reasons. Firstly, a greedy algorithm is fast as it always chooses the optimal choice at each step instead of processing every choice and their derivatives. Speedy feedback is important in games to increase UX. With increased speed, however, comes a loss of accuracy. The addition of pseudocounts improves the accuracy by not ruling out *k*-mers too soon. Furthermore, since MFP is NP-complete, their does not exist an algorithm that consistently finds an optimal solution every time in polynomial time. There has to be some tradeoff between accuracy and time. Random algorithms have been proven to be accurate and fast, however they require multiple runs to find the best solution. This is not ideal for a game environment. Additionally, the puzzle sizes are limited to the screen size. Puzzles are built on subsequences of DNA rather than whole sequences because there is a limit to how many blocks can fit on the screen and how small blocks can be until they are undistinguishable to the player. Greedy Motif Search with Pseudocounts works well with smaller inputs. Lastly, this algorithm will never under-score a puzzle. That is, if the algorithm's solution to a puzzle is incorrect, it will only be a larger score than what the optimal solution should be. In terms of points, solutions to puzzles will never have a larger point value than the optimal solution, so a player will never be told they are wrong for beating the algorithm.

4.3 Implementation

The following languages and technologies were used to build nucleoSLIDE: HTML, CSS, PHP, SQLite, and JavaScript. HTML was used to create the general layout of the pages and CSS to style the pages. The majority of the UI and all functionalities were built in pure JavaScript, including the totality of the puzzles. SQLite queries through PHP were used to communicate to the database, and PHP was used to pass retrieved data between pages. Additionally, a Java program was created to parse data from GenBank.

4.4 Design Elements

This section defends the design decisions made for nucleoSLIDE based on elements described in section 2.2.2. Elements included in nucleoSLIDE are aesthetics and graphics, competition, framing, gameplay, goals, learning, novelty, play, pretending, rules, setting the pace, storytelling, and UI.

The *aesthetics and graphics* of nucleoSLIDE are simple and functional and attempt to evoke discovery and challenge in players. All graphics use simple colors from a defined color scheme and simple shapes. Since the game uses colors to aid the player, the colors of the UI are grey and white to not draw attention away from the puzzles. Any information needed to be communicated to the player is done so through overlaid pop-ups. Darkening the game allows the ability to introduce more complex graphics, like badges, without overloading the player. The intentions of the game are to evoke a desire for discovery and challenge. The actions for interacting with the puzzle are smooth to allow the player to experiment with the data without feeling limited. Additionally, the complexity of puzzles should pose challenging to players and stimulate their need to overcome the challenges.

Competition is created through use of scores, points, and badges. Players work against each other (and the algorithm) to receive the highest ranking. Rank calculations take into consideration a player’s score, completed levels, and badges.

nucleoSLIDE is *framed* in a way that maximizes the amount of participation. No knowledge of genomes or MFP is required to play. By building the game as a table, players do not have to learn about new shapes or structures. Through settings, players can select their preferred interaction style and color scheme, increasing literacy of controls before play and improving accessibility for color blindness. The initial interaction setting is 'click' because, through the use of buttons, it emphasizes that only horizontal movement is permitted. Furthermore, the tutorial uses encouraging language to emphasize that all players are welcomed.

Gameplay includes the challenges faced by the player and the actions allowed to address those challenges. The challenge of nucleoSLIDE is to solve each puzzle by finding the most common pattern for all sequences with minimal variations. To address this challenge, the player must slide sequences left and right, aligning their solution in the pattern-matching window. Permitted sliding actions include button clicks, arrow keys, or drag. Players are permitted to save and revisit workspaces. Players are also allowed to search for patterns, select searched patterns, and reset the puzzle.

The *goal* for nucleoSLIDE players is to complete each puzzle. Additional goals include receiving badges and reaching the highest rank. The goals are defined by the rules, which are discussed shortly. However, the main goal of nucleoSLIDE is to advance scientific research on genomics and increase genome literacy in the public.

Learning about the game is introduced through the tutorial and gameplay. The tutorial tells the player the problem nucleoSLIDE addresses, how their contributions can help, and about each component of the UI. For players who are interested in the science behind nucleoSLIDE, the About Page explains MFP and citizen science games in more detail.

Solving MFP is a *novel* idea for a game. The closest known game is *Phylo*, which focuses on MSA. However, gameplay is still very much different between nucleoSLIDE and *Phylo*. For example, *Phylo* allows nucleotides to be moved individually. As a result, gaps are

permitted in sequences, where a gap is a position in a sequence without a nucleotide. Gaps negatively affect the score, as well as mismatched nucleotides. nucleoSLIDE, on the other hand, does not permit gaps and does not penalize mismatched nucleotides.

Play requires player participation. nucleoSLIDE requires the player to actively search for patterns, or subpatterns, and slide the sequences accordingly. The puzzles cannot be solved without play. Players have the freedom to choose how they solve the puzzles, in adherence to the rules.

Pretending refers to the boundary between game concepts and real-world concepts. In the real-world, motifs are found in DNA sequences to find TFBSs while, in the game-world, motifs are found to achieve high scores and possibly receive badges. In the real-world, sequences are displayed as long strings of nucleotide characters while, in the game-world, sequences are chains of color-coded blocks. However, in both the real-world and the game-world, motif finding is done to improve scientific research.

The *rules* of nucleoSLIDE are based on the rules of motif finding in the real-world. They include: (1) sequences can only slide horizontally (left/west and right/east) until the end-of-sequence block is reached, (2) patterns, or subpatterns, can be searched by using the pattern search box and found patterns are highlighted black, (3) searched patterns can be selected to automatically slide the sequences so that the pattern is left-aligned inside the pattern-matching window, (4) solutions can be evaluated by clicking the evaluate button and only the nucleotides inside the pattern-matching window will be scored, (6) workspaces can be revisited upon selection in the history display, and (5) a submitted workspace with a score within 85% of the algorithm's solution allows for clearance to the next level.

nucleoSLIDE players completely *set the pace*. There is no time limit for puzzles. New levels are open only once all previous levels are cleared.

Storytelling is done in the tutorial, when the player is asked if they would accept the job of helping scientists solve MFP. Additionally, for each level, the class of organisms in which the sequences belong to is provided to add context.

Lastly, the *UI* of nucleoSLIDE encompasses everything displayed in the front end (see section 4.1 for frontend features).

Chapter 5: Conclusions and Future Work

In this thesis, a citizen science game for MFP was developed. This chapter discusses the conclusions deduced from building nucleoSLIDE and addresses possible future extensions.

6.1 Conclusions

The research began with examining existing citizen science projects and games. Projects involve public participation in scientific research by asking participants to collect, analyze, and process data. Gamification of these projects uses common game elements to attract and sustain participation. By examining existing research and case studies on current citizen science games, a list of common game elements in citizen science games was formed. These elements include aesthetics and graphics, competition and cooperation, framing, gameplay, goal, learning, novelty, play, pretending, risks, rewards, and recognition, rules, setting the pace, socializing, storytelling, UX, and UI.

The next part of the research was studying the scientific problem to gamify: MFP. MFP was defined and a sample of current MFP algorithms was examined.

In an effort to combine citizen science and MFP, nucleoSLIDE was built. nucleoSLIDE is a web game that incorporates most of the game design elements found in the first part of the research. Based on the MFP research, the Greedy Motif Search with Pseudocounts was chosen as the background algorithm for nucleoSLIDE. The game takes advantage of the natural pattern recognition skills of humans to test players' abilities to find motifs. The goal of nucleoSLIDE is to advance the current solutions for MFP by including public participants, leading to better scientific literacy in the public as well.

6.2 Future Work

Developing citizen science projects is an iterative process that involves multiple stakeholders. The work presented in this thesis reflects that of a computer scientist with the

understanding of MFP from a bioinformatics standpoint. A suggested future extension is the inclusion of genomic researchers that can analyze the results and verify the game's capabilities. Furthermore, a genomic researcher can parse DNA sequences more appropriately if they have some prior understanding of the sequences. Another important stakeholder that would greatly benefit nucleoSLIDE, and lead to the next iteration of the project, are the participants. User testing would show which game elements and features are the most beneficial and which need improvement.

nucleoSLIDE could be advanced with the inclusion of multiple MFP algorithms. Since current algorithms have tradeoffs between speed and accuracy, it could be interesting to see which algorithms match the players' way of thinking and most appropriately fit a game setting. Moreover, player results could be used to optimize the puzzles and algorithms.

References

[1] Crowdsourcing and Citizen Science Act of 2016, 15 U.S.C. § 3724. 2016. [Online]. Available: https://www.congress.gov/bill/114th-congress/house-bill/6414/text. [Accessed 2 April 2020].

[2] "About CitizenScience.gov," *CitizenScience.gov*. [Online]. Available: https://www.citizenscience.gov. [Accessed 2 March 2020].

[3] *Phylo*. [Online]. Available: https://phylo.cs.mcgill.ca/. [Accessed 13 March 2020].

[4] *Foldit*. [Online]. Available: https://fold.it/. [Accessed 13 March 2020].

[5] *Eyewire*. [Online]. Available: https://eyewire.org/explore. [Accessed 13 March 2020].

[6] J. Garbarino and C. E. Mason, "The power of engaging citizen scientists for scientific progress," *J. of Microbiology & Biology Education*, vol. 17, no. 1, pp. 7-12. March, 2016, doi: 10.1128/jmbe.v17i1.1052. [Accessed 2 April 2020].

[7] R. Bonney, T. B. Phillips, L. H. Ballard and J. W. Enck, "Can citizen science enhance public understanding of science?," *Public Understanding of Sciene*, vol. 25, no. 1. October, 2015, doi: 10.1177/0963662515607406. [Accessed 2 April 2020].

[8] E. J. Theobald, *et al*., "Global change and local solutions: Tapping the unrealized potential of citizen science for biodiversity research," *Biological Conservation*, vol. 181, pp. 236-244. October, 2014, doi: 10.1016/j.biocon.2014.10.021. [Accessed 2 April 2020].

[9] *Zooniverse.org*. [Online]. Available: https://www.zooniverse.org/. [Accessed 17 April 2020].

[10] H. Sauermann and C. Franzoni, "Crowd science user contribution patterns and their implications," *Proc. of the National Academy of Sci.*, vol. 112, no. 3, pp. 679-684, January, 2015, doi: 10.1073/pnas.1408907112. [Accessed 2 April 2020].

[11] K. Schrier, "Designing learning with citizen science games," *The Emerging Learning Design J.*, vol. 4, no. 1, pp. 19-26, 2018. [Online]. Available: https://digitalcommons.montclair.edu/eldj/vol4/iss1/3. [Accessed 2 April 2020].

[12] R. Bonney, *et al*., "Citizen science: A developing tool for expanding science knowledge and scientific literacy," *BioScience*, vol. 59, no. 11, pp. 977-984, December 2009, doi: 10.1525/bio.2009.59.11.9. [Accessed 1 April 2020].

[13] C. B. Cooper, J. Dickinson, T. Phillips and R. Bonney, "Citizen science as a tool for conservation in residential ecosystems," *Ecology and Society*, vol. 12, no. 2, 2007. [Online]. Available: http://www.ecologyandsociety.org/vol12/iss2/art11/. [Accessed 2 April 2020].

[14] A. K. Przybylski, C. S. Rigby and R. M. Ryan, "A motivational model of video game engagement," *Review of Generaly Psychology*, vol 14, no. 2, 2010, doi: 10.1037/a0019440. [Accessed 5 April 2020].

[15] J. A. Miller, U. Narayan, M. Hantsbarger, S. Cooper and M. S. El-Nasr, "Expertise and engagement: re-designing citizen science games with players' minds in mind," in *Proc. of the 14th Int. Conf. on the Foundations of Digitial Games*, 2019, doi: 10.1145/3337722.3337735. [Accessed 5 April 2020].

[16] B. Schouten, E. van der Spek, D. Harmsen and E. Bartholomeus, "The playful scientists: Stimulating playful communities for science practice," in *The Playful Citizen*. Netherlands: Amsterdam University Press, 2019, ch. 12, pp. 235-251.

[17] A. Denisova, I. A. Nordin and P. Cairns, " The convergence of player experience questionnaires," in *Proc. of the 2016 Annual Symposium on CHI*, 2016, pp. 33-37, doi: 10.1145/2967934.2968095. [Accessed 5 April 2020].

[18] I. Iacovides, C. Jennett, C. Cornish-Trestrail and A.L. Cox, "Do games attract or sustain engagement in citizen science? A study of volunteer motivations," *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pp. 1101-1106, May, 2013, doi: 10.1145/2468356.2468553. [Accessed 2 April 2020].

[19] C. Franzoni and H. Sauermann, "Crowd science: The organization of scientific research in open collaborative projects," *Research Policy*, vol. 43, no. 1, 2013, doi: 10.1016/j.respol.2013.07.005. [Accessed 5 April 2020].

[20] J. A. Yee, "Characterizing crowd participation and productivity of foldit through web scraping," M.S. thesis, Computer Science, Naval Postgrad. School, CA, USA, 2016. [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/1027731.pdf. [Accessed 6 April 2020].

[21] F. Khatib, *et al.*, "Crystal structure of a monomeric retroviral protease solved by protein -folding game players," *Nature Structural & Molecular Biology*, vol 18, no. 10, pp. 1175-1177, September, 2011, doi: 0.1038/nsmb.2119. [Accessed 6 April 2020].

[22] "EyeWire | A Game to Crowdsource Brain Mapping," *CitizenScience.gov*, [Online]. Available: https://www.citizenscience.gov/eyewire-brain-mapping/#. [Accessed 7 April 2020].

[23] P. Compeau and P. Pevzner, "Epilogue: Multiple sequence alignment," in *Bioinformatcs Algorithms: An Active Learning Approach*. California, USA: Active Learning Publishers, 2015, vol. 1, 2nd ed., ch. 2, pp. 277-282.

[24] A. Kawrykow *et al.*, "Phylo: A citizen science approach for improving multiple sequence alignment," *PLoS One*, vol. 7, no. 3, 2012, doi: 10.1371/journal.pone.0031362. [Accessed 7 April 2020].

[25] S. Deterding, D. Dixon, R. Khaled and L. Nacke, "From game design elements to gamefulness: Defining 'gamification'," in *Proc. of the 15th Int. Academic MindTrek Conf.: Envisioning Future Media Environments*, 2011, pp. 9-15, doi: 10.1145/2181037.2181040. [Accessed 10 April 2020].

[26] K. Mitgutsch and N. Alvarado, "Purposeful by design? A serious game design assessment framework," in *Proc. of the Int. Conf. on the Foundations of Digitial Games*, 2012, pp. 121-128, doi: 10.1145/2282338.2282364. [Accessed 10 April 2020].

[27] E. Adams, "Games and Video Games," in *Fundamentals of Game Design*, ed. 3, ch. 1, New Riders, 2013.

[28] "Museum | Eyewire," *EyeWire.org*, [Online]. Available: http://museum.eyewire.org/?neurons=26065,20117,26051,17212&browse=1. [Accessed 10 April 2020].

[29] M. Sailer, J. U. Hense, S. K. Mayr and H. Mandl, "How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction," *Computers in Human Behanvior*, vol. 69, pp. 371-380, 2017, doi: 0.1016/j.chb.2016.12.033. [Accessed 10 April 2020].

[30] O. Schaffer and X. Fang, "What makes games fun? Card sort reveals 34 sources of computer game enjoyment," in *24th Americas Conf. on Info. Sys.*, New Orleans, LA, USA, 2018. [Online]. Available: https://aisel.aisnet.org/amcis2018/HCI/Presentations/2/. [Accessed 10 April 2020].

[31] B. M. Winn, "The design, play, and experience framework," pp. 1010-1024, 2009. [Online]. Available: http://ksuweb.kennesaw.edu/~rguo/2015_Spring/CGDD4303/readings/winn-dpe-chapter.pdf. [Accessed 10 April 2020].

[32] S. Fizek and A. Dippel, "Laborious playgrounds: Citizen science games as new modes of work/play in the digital age," in *The Playful Citizen*. Amsterdam, Netherlands: Amsterdam University Press, 2019, ch. 13, pp. 255-268.

[33] J. S. Kim *et al*., "Space-time wiring specificity supports direction selectivity in the retina," *Nature*, vol. 509, pp. 331-336, 2014, doi: 10.1038/nature13240. [Accessed 10 April 2020].

[34] R. Tinati, M. Luczak-Roesch, E. Simperl and W. Hall, "An investigation of player motivations in Eyewire, a gamified citizen science project," *Computers in Human Behavior*, vol. 73, pp. 527-540, August, 2017, doi: 10.1016/j.chb.2016.12.074. [Accessed 7 April 2020].

[35] P. Pevzner and P. Compeau, "Which DNA Patterns Play the Role of Molecular Clocks?," in *Bioinformatics Algorithms: An Active Learning Approach*. California, USA: Active Learning Publishers, 2015, vol. 1, 2nd ed., ch. 1, pp. 66-106.

[36] F. A. Hashim, M. S. Mabrouk and W. Al-Atabany, "Review of different sequence motif finding algorithms," *Avicenna J. of Medical Biotechnology*, vol. 11, no. 2, pp. 130, 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6490410/. [Accessed 22 March 2020].

[37] R. VijayaSatya and A. Mukherjee, "PRUNER: algorithms for finding monad patterns in DNA sequences," in *Proc. 2004 Comp. Sys. Bioinformatics Conf., 2004*, Stanford, CA, USA, 2004, pp. 662-665, doi: 10.1109/CSB.2004.1332537. [Accessed 24 March 2020].

[38] M. K. Das and H.-K. Dai, "A survey of DNA motif finding algorithms," *BMC Bioinformatics*, vol. 8, no. S7, November, 2007, doi: 10.1186/1471-2105-8-S7-S21. [Accessed 22 March 2020].

[39] W.-K. Sung, "Motif Finding," in *Algorithms in Bioinformatics: A Practical Introduction*, CRC Press, 2009, pp. 247-279.

[40] G. Pavesi, G. Mauri and G. Pesole, "An algorithm for finding signals of unknown length in DNA sequences," *Bioinformatics*, vol. 17, suppl. 1, pp. S207-S214, 2001, doi: 10.1093/bioinformatics/17.suppl_1.S207. [Accessed 24 March 2020].

[41] J. Buhler and M. Tompa, "Finding motifs using random projections," *J. of Computational Biology*, vol. 9, no. 2, pp. 225-242, 2002, doi: 10.1089/10665270252935430. [Accessed 24 March 2020].

[42] G. D. Stormo, "DNA binding sites: representation and discovery," *Bioinformatics*, vol. 16, no. 1, pp. 16-23, January, 2000, doi: 10.1093/bioinformatics/16.1.16. [Accessed 24 March 2020].

[43] C. Lei and J. Ruan, "A particle swarm optimization-based algorithm for finding gapped motifs," *BioData Mining*, vol. 13, no. 1, pp. 9, 2010, doi: 10.1186/1756-0381-3-9. [Accessed 24 March 2020].

[44] S. Aslan and D. Karaboga, "A discrete artificial bee colony algorithm for detecting transcription factor binding sites in DNA sequences," *GMR*, vol. 15, no. 2, pp. 1-11, 2016, doi: 10.4238/gmr.15028645. [Accessed 24 March 2020].

[45] A. J. Griffiths, J. H. Miller, T. D. Suzuki, R. C. Lewontin and G. M. William, “Discovery of the lac system: Negative control,” in *An Introduction to Genetic Analysis*, New York, USA: W.H. Freeman, 2005, ch. 11. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK21766/. [Accessed 24 March 2020].

[46] "GenBank," *NCBI*, [Online]. Available: https://www.ncbi.nlm.nih.gov/genbank/. [Accessed December 2019].

[47] "Available color schemes," *DNASTAR.com*, [Online]. Available: https://www.dnastar.com/manuals/MegAlignPro/15.2/en/topic/selecting-color-schemes. [Accessed December 2019].

[48] A. Sterling, "A Walk Through Eyewire," *Eyewire.org*, 2014. [Online]. Available: https://blog.eyewire.org/how-to-play-eyewire/. [Accessed 2020 10 March].

[49] "Google Play: Phylo DNA Puzzle," *Google Play*. [Online]. Available: https://play.google.com/store/apps/details?id=phylo.cs.mcgill.ca&hl=en. [Accessed 2 May 2020].

Appendix A: *Foldit*, *EyeWire*, and *Phylo* Screenshots

A.1 *Foldit*

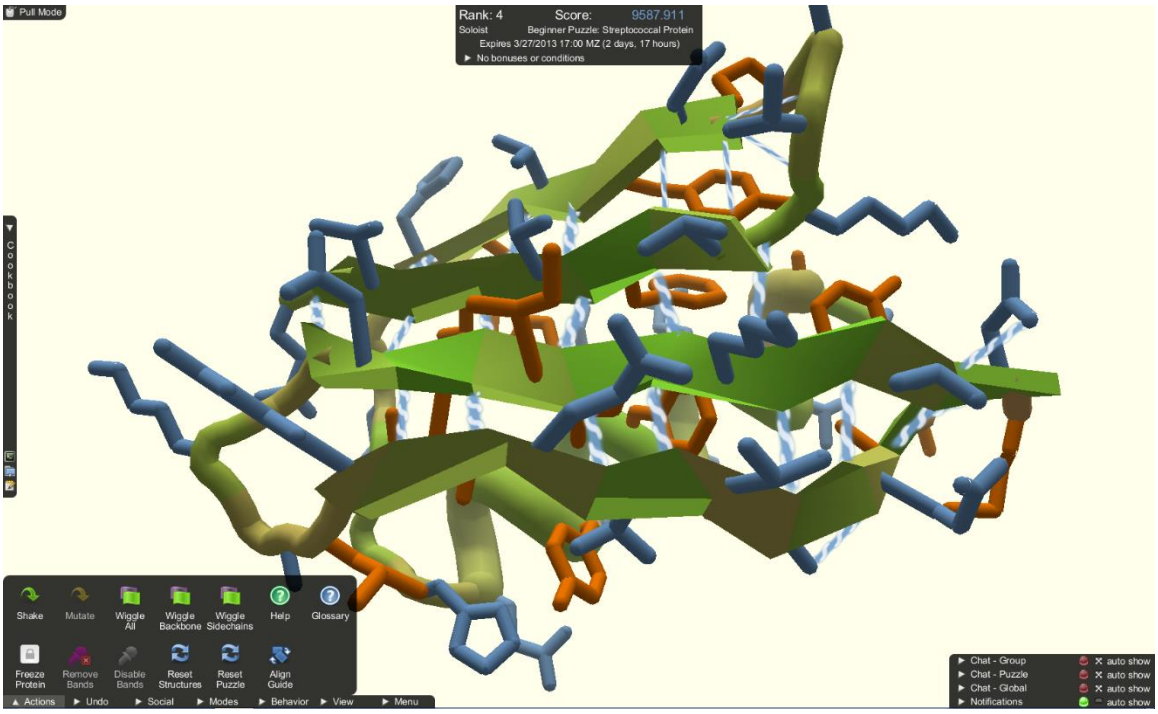

Figure 16: UI and example puzzle of *Foldit*, a 3-dimensional puzzle game for protein folding. Bottom-left corner displays gameplay actions and aids, bottom-right corner access to chatgroups and notifications, and top box the players rank and score. The player interacts with the puzzle in the 3-dimenstional space using mouse clicks and drags. The puzzle presented is of the streptococcal protein. *Foldit* can be downloaded from [4]. Image from [4].

Figure 17: Snapshots of *Foldit*'s interactive tutorial. Images taken by author of thesis.

A.2 *EyeWire*

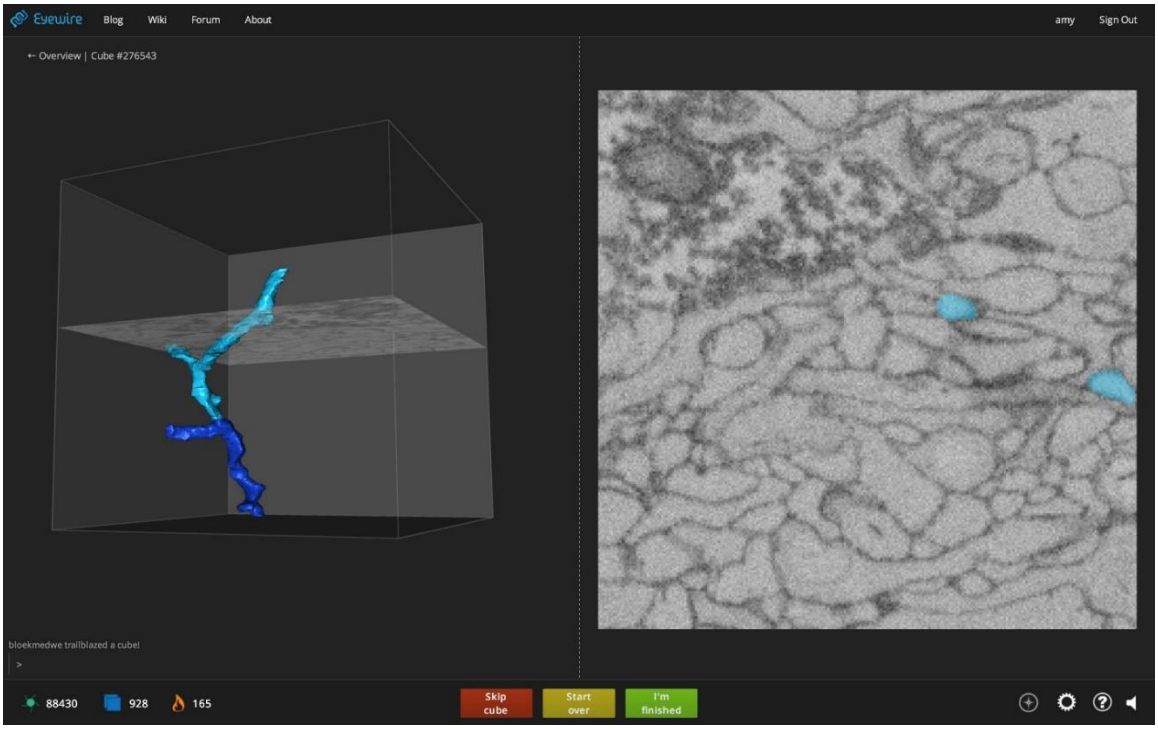

Figure 18: UI and example puzzle of *EyeWire*, a 3-dimenstional puzzle game for mapping neurons in a cube. The left view displays the neuron in the cube and the right view is a cross-section of the cube and neuron branches. Game can be played at [5]. Image from [48].

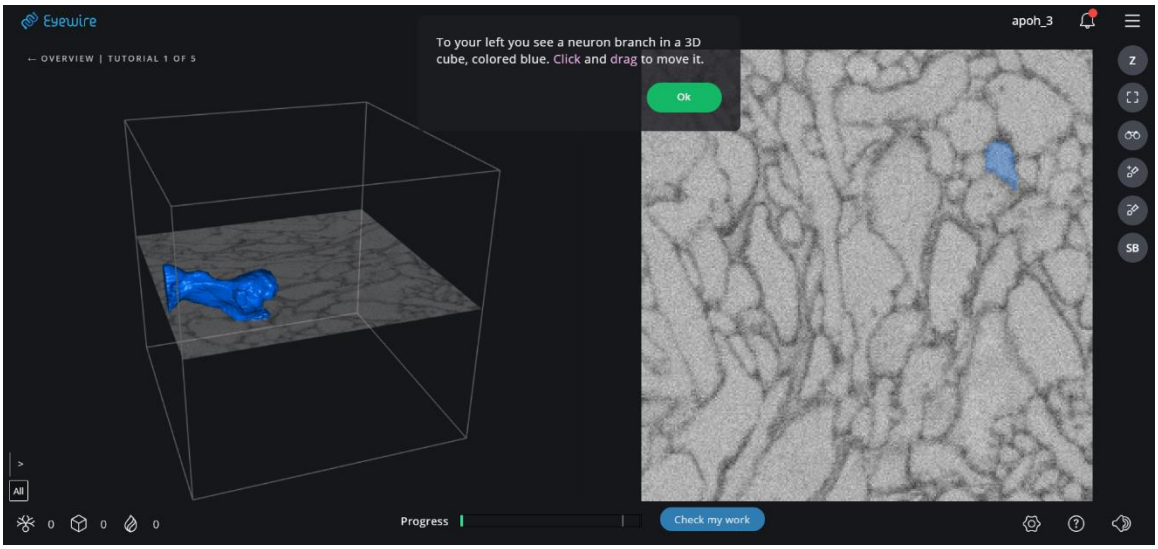

Figure 19: Snapshot of *EyeWire*'s interactive tutorial. Image taken by author of thesis.

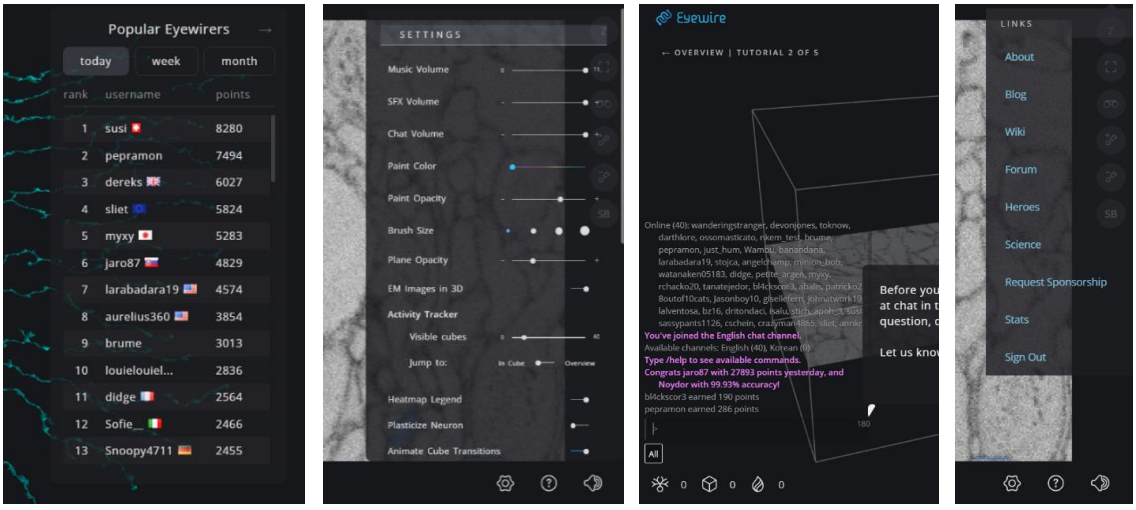

Figure 20: From left to right, leaderboard, settings, chat, and additional links provided by *EyeWire*. Images taken by author of thesis.

A.3 *Phylo*

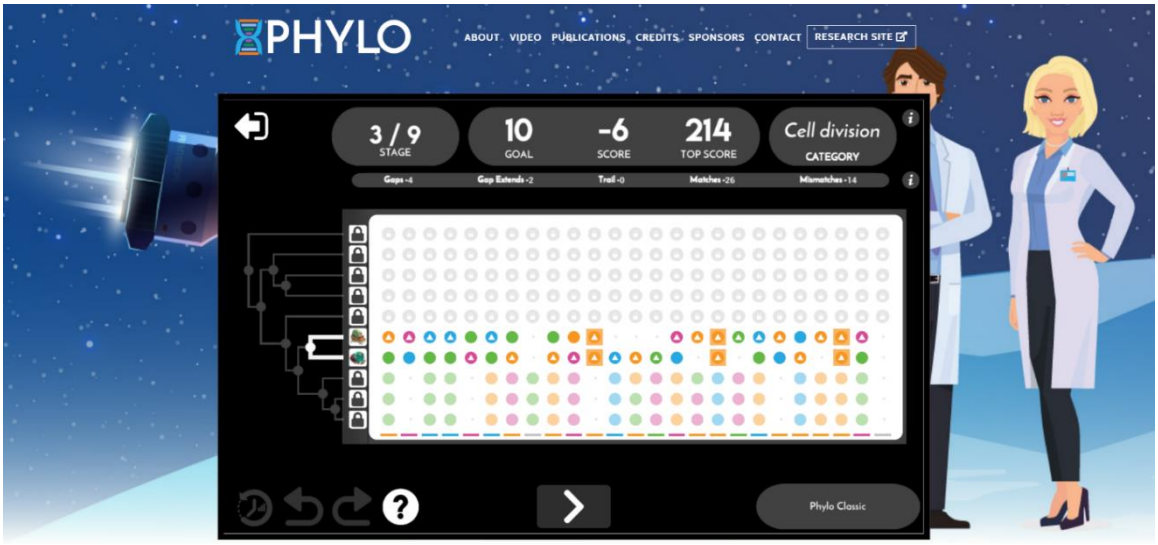

Figure 21: UI and example puzzle of *Phylo*, a 2-dimenstional puzzle game for MSA. Players slide the dots horizontally until the rows are best matched by color. The top displays stage, goal, score, top score, and puzzle reference. Game can be played at [3] or downloaded from [49]. Image taken by author of thesis.

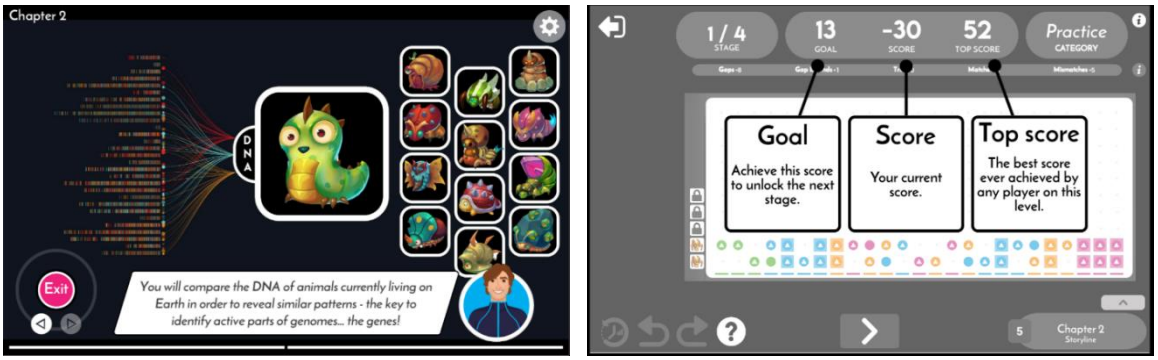

Figure 22: Snapshots of storytelling and tutorial of *Phylo*. Images taken by author of thesis.

Appendix B: Game Elements

Table 1: Common Game Design Elements found in Citizen Science Games

Element	**Description**	**Example(s)**
Aesthetics and Graphics [26], [27]	The audiovisual language and display of game elements.	contrast of bright colors on a dark background; fluid movements in 3d space
Competition and Cooperation [27], [29], [30]	Working against others towards conflicting goals (competition) or working with others towards shared goals (cooperation).	cooperation: team mode; competition: high score lists and leaderboards
Framing [26]	Framing elements to suit the target players.	minimizing science complexity by using recognizable shapes
Gameplay [25], [27], [31]	The challenges a player faces and the actions permitted to address those challenges.	the challenge of aligning dots according to color through the action of horizontal drag
Goal [30]	The object of the game.	complete puzzles; reach high score
Learning [27], [31]	Acquiring knowledge of how the game works (and its scientific purpose for CS).	design beginning levels to teach needed skills; providing about pages
Novelty [27]	New, original, or unusual gameplay or features.	rotating, “wiggling” and “shaking a 3d object
Play [27], [32]	Participation through choices and experiences.	letting the player experiment in the game space (without breaking rules)
Pretending [27], [32]	Creating a temporary magic circle, or pretended reality, in the mind.	sliding a dot (magic circle) versus aligning DNA (reality)
Risks, Rewards, and Recognition [18], [27], [29]	A risk is an action with uncertainty. Rewards are the results of successful risk-taking, and recognition is warranted for rewards.	require actions that lower the score before an action that will greatly improve the score; when an accomplishment is reached, share it in the chat
Rules (and teaching of) [27]	Definitions and instructions that the player agrees to for the duration of the game.	movement of an element only in the x-direction until it reaches a “wall” or adjacent elements reach a “wall”
Setting the Pace [27]	The player’s ability to progress through the game at their own speed.	player selects “ok” when they are ready to move on
Socializing [18], [27]	The act of communicating with players through forums, chats, or other technologies.	real-time forums and chat tools that are easily accessible during gameplay

Storytelling [26], [27], [29], [31]	Sharing background information, rules, and goals to players through narration or other expressive means.	text-narration from characters and movement of graphics to symbolize progression
UX [31]	The player's emotions and attitudes to game design decisions.	pleasure; frustration
UI [31], [34]	Everything the user sees, hears, and interacts with and how that interaction occurs.	actions panel; real-time chats; sound effects; progress bars