Resurrection of OpenPhylo

Akash Singh

Master of Science

Department of Computer Science

McGill University

Montreal,Quebec

2018-06-04

A thesis submitted to McGill University in partial
fulfillment of the requirements of the degree of
Master of Science.

DEDICATION

To my parents and my sister, even though they never came to visit me in Canada.

To the strongest man alive: me (About me: ask my ancestors because every bit of me is little bit of them.)

others: This is not for you.

ACKNOWLEDGEMENTS

I express my sincere gratitude to Prof. Jerome Waldispuhl for introducing to the applications of human computation in computational bioinformatics. I am blessed to have him as my supervisor, he has always encouraged me and has directed me throughout the master program. I also want to express my sincere gratitude to Prof. Mathieu Blanchette for providing support in bioinformatics.

I would also like to thank my friend Shabir Abdul Samadh for being there during the tough times. I would like to acknowledge the efforts of Faizy Ahsan and Zoe Hu for their contribution in feature based difficulty prediction and aggregation methodology respectively. I am also grateful to fellow bioinformatics and to McGill University and its staff members for providing an enjoyable research environment and a comfortable stay.

Finally, I thank my parents Ghanshyam and Chinta Singh and my sister Sapna Singh Sarkar for their motivational support.

ABSTRACT

In this report, we analyzed the data collected during the last 8 years (2010-2018) through Phylo. Phylo is a game with a purpose (GWAP) designed for solving a fundamental problem in bioinformatics: the multiple sequence alignment problem (MSA). Based on the analysis, we propose an improved architecture for OpenPhylo. We start by evaluating various machine learning approaches to identify promising regions in MSA, regions that have not been perfectly aligned via machine. The learning model is trained by converting segments of MSA into images and feeding it to a convolutional neural network (CNN) based model. The validation accuracy achieved by the proposed model is 80.67%. We then improve these alignments by crowdsourcing and collecting imroved solutions from the players through Phylo. While crowdsourcing these alignments/puzzles we also focus on certain aspects very crucial to any human computation system. For difficulty prediction of microtasks, we analysed both, feature based as well as deep learning models. Further, we define a novel routing algorithm which matches players' skill level against puzzle difficulty. Aggregation of solutions received via crowdsourcing is achieved using modified Needleman-Wunsch algorithm and position weight matrices. The feedback mechanism has been enhanced by introducing a user profile page which includes: achievement badges, ranking system, visual feedbacks, and a live messaging mechanism to notify when players contribute to science. Finally, we develop an enhanced teaching portal analysing the data received from the classical version with the primary purpose of helping students in understanding the problem of MSA.

ABRÉGÉ

Dans ce rapport, nous avons analysé les données collectées au cours des 8 dernières années (2010-2018) à travers Phylo. Phylo est un jeu avec un but (GWAP) conçu pour résoudre un problème fondamental en bioinformatique: le problème d'alignement de séquences multiple (MSA). Sur la base de l'analyse, nous proposons une architecture améliorée pour OpenPhylo. Nous commençons par évaluer différentes approches d'apprentissage automatique pour identifier les régions prometteuses dans le MSA, régions qui n'ont pas été parfaitement alignées par la machine. Le modèle d'apprentissage est formé en convertissant des segments de MSA en images et en les alimentant sur un modèle convolutional neural network (CNN). La précision de la validation obtenue par le modèle CNN proposé est de 80,67 %. Nous améliorons ensuite ces alignements en faisant du crowdsourcing sur ces segments de MSA et en collectant des solutions améliorées auprès des joueurs via Phylo. Tout en crowdsourcing ces alignements / puzzles, nous nous concentrons également sur certains aspects très cruciaux pour tout système de calcul humain. Pour la prédiction de la difficulté des microtâches, nous avons analysé, à la fois, les caractéristiques ainsi que des modèles d'apprentissage en profondeur. En outre, nous définissons un nouvel algorithme de routage qui assortit le niveau de compétence des joueurs contre la difficulté du puzzle. L'agrégation des solutions reçues par crowdsourcing est réalisée en utilisant l'algorithme de Needleman-Wunsch modifié et les matrices de poids de position. Le mécanisme de rétroaction a été amélioré en incluant une page de profil d'utilisateur qui comprend: des insignes de réussite, un système de classement, des

rétroactions visuelles et un mécanisme de messagerie en direct à notifier lorsqu'ils contribuent au savoir science. Enfin, nous développons un portail d'enseignement amélioré analysant les données reçues de la version classique dans le but principal d'aider les étudiants à comprendre le problème de MSA.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGEMENTS iii

ABSTRACT iv

ABRÉGÉ v

LIST OF TABLES x

LIST OF FIGURES xi

1 Introduction and related works 1

2 Intended audience 17

3 Puzzle extraction 22

3.1 DNA based puzzle extraction 22
3.2 RNA based puzzle extraction 25

4 Promising puzzles 29

4.1 Objective 30
4.2 Dataset 30
4.3 Methodology 31
4.4 Results 35

5 Difficulty prediction 39

5.1 Dataset 39
5.2 Approach 1: Difficulty prediction using feature extraction 39
5.2.1 Label creation 39
5.2.2 Methodology 40
5.2.3 Results 44

5.3 Approach 2: Difficulty prediction using convolutional neural networks . . . 48
5.3.1 Label creation . . . 48
5.3.2 Methodology . . . 49
5.3.3 Results . . . 52
5.3.4 Correlation between promising puzzles and their difficulty estimates . . . 56

6 Dynamic change in difficulty . . . 58

6.1 Dataset . . . 59
6.2 Methodology . . . 59
6.2.1 Computing Uncertainty . . . 60
6.3 Results . . . 60

7 Routing . . . 63

7.1 Routing algorithm . . . 64

8 Teaching Portal . . . 69

8.1 Classical educational platform . . . 70
8.2 Upcoming Teaching Portal . . . 73
8.2.1 Instructor . . . 74
8.2.2 Students . . . 77

9 User feedback . . . 80

9.1 Feedback to users . . . 82
9.2 Feedback to scientist . . . 84
9.3 Conclusion . . . 84

10 Aggregation . . . 86

10.1 Classical approach of aggregation . . . 87
10.2 Upcoming version of aggregation . . . 88
10.2.1 Methodology . . . 88
10.2.2 Evaluation metric . . . 91
10.2.3 Results . . . 92

11 Conclusion . . . 93

Appendix A 95
Appendix B 98
Appendix C 101
Appendix D 108
References 110

LIST OF TABLES

Table page

4–1 Number of data points in each class. 31

4–2 Train, test and validation split of dataset. 31

4–3 Hyperparameter values for the CNN model used. 35

4–4 Accuracy obtained using the proposed model. 36

5–1 Estimate of the total number of solutions that needs to be collected to obtain the highest score in the uniform (N_u), weighted (N_w) and optimal (N_{opt}) routing schemes. Puzzles are sorted in 3 categories (easy, medium, and difficult) representing the observed difficulty for players to achieve the highest score. 48

5–2 Data partition for type one difficulty prediction 49

5–3 Hyperparameter values for the CNN based difficulty prediction model. 51

5–4 Dataset split for difficulty prediction 52

5–5 Correlation between promising diffculty and promising puzzles 57

11–1 Feature values for RNA puzzles used in Ribo game. 100

11–2 Dataset of puzzles for dynamic difficulty change 101

11–3 Number of puzzles collected and rank of highest scoring alignments. . 109

LIST OF FIGURES

Figure page

1–1 Interface of Phylo (2017 release). Each row is composed of a sequence of bricks of 4 different colors representing the 4 nucleotides A, C, G, and T. These sequences have been extracted from the DNA of different species represented with a icon on the left of the grid. Participants move the tiles left or right in order to maximize the number of color matches in each column. Although, the order of the bricks cannot be changed. Color mismatches and gaps are thus unavoidable and bring penalties. The phylogenetic tree on the left indicates the priority in which the rows should be aligned. 5

1–2 Interface of RNA version in Phylo (yet to be release). Each row is composed of a sequence of tiles of 4 different colors representing the letters from the RNA alphabet: A, C, U and G. These sequences have been extracted from the RNA families stored in rfam database. Participants move the tiles left or right in order to maximize the number of color matches in each column. Also, there is an additional scoring scheme for aligning the secondary structures in RNA alignment. The order of the bricks cannot be changed, however, you can edit the secondary structure based on your requirements. Color mismatches and gaps are thus unavoidable and bring penalties. The orange colored boundries in this figure reflect possible secondary structure similarity in the RNA puzzle. 15

1–3 Architecture of OpenPhylo. Each annotation represents a module in OpenPhylo and has been discussed as a chapter in this report. . . . 16

2–1 This is a static representation of registration process in OpenPhylo. The diagram represents attributes, operations (or methods), and the relationships among the various actors/registration classes of OpenPhylo. The register package consists of all the registration related classes. User and Scientist extend an abstract class with mandatory attributes for all registration process. User account can be converted to scientist account and Guest user can also convert their account to a registered user account without losing any progress. Instructor registration has been kept discrete and independent of other actor registration. 21

3–1 Puzzle generation options available to scientists for DNA alignments. 23

3–2 Puzzle generation in game using segments of MSA 24

3–3 Puzzle generation options available to the end users for RNA alignments. 26

4–1 Sample DNA puzzle processed into an image. 32

4–2 CNN model for image classification to decide the scope of improvement in MSA. 34

4–3 Average training accuracy of the proposed CNN model. The graph shows the training curve for 10-fold cross validation. The spikes are an unavoidable consequence of mini-batch Gradient Descent in Adam (batch size=32). 36

4–4 Average training loss of the porposed CNN model. The graph shows the loss curve for 10-fold cross validation. The spikes are an unavoidable consequence of mini-batch Gradient Descent in Adam (batch size=32). 37

4–5 Average validation accuracy of the proposed model. 38

4–6 Average validation loss of the proposed CNN model. 38

5–1 Feature comparison between positive and negative class. The blue dots represent negative cases, whereas the green dots represent positive cases. 42

5–2 Pearson correlation of features selected for difficulty prediction. These correlation coefficients only measure linear correlations. 43

5–3 Histogram plot of the improvement in the score for 1556 puzzles, where improvement is measured as the difference between the best score produced by a human player and the initial computer-based alignment. 45

5–4 Measuring effects of feature combination on models performances. The features from the ordered list (based on individual feature importance in decreasing order) are added incrementally to train models. 46

5–5 Individual feature performance is measured through AUC values on testing set. Each model is trained using one feature at a time. . . 46

5–6 Lebeling of puzzles based on its success rate. X-axis represents the success rate and y-axis represents the number of puzzles for with respect to success rate. 49

5–7 Proposed CNN model for difficulty prediction. 50

5–8 Average learning accuracy in difficulty prediction. The 10 different colored lines shows ten unique learning curve generated using 10-fold cross validation. The learning accuracy achieved by the CNN model is 96%. 53

5–9 Average learning loss in difficulty prediction. The ten lines generated represents each iteration of 10-fold cross validation. 54

5–10 Average validation accuracy in difficulty prediction. 10 different colored lines represents each iteration of 10-fold cross validation. The validation accuracy received 75.15% 55

5–11 Average learning loss in difficulty prediction. The 10 lines generated represents each iteration of 10-fold cross validation. 56

6–1 Identifying the true difficulty of puzzles based on the player's success rate. Vertical bar represents the overall error rate of all the puzzles corresponding to that difficulty level. 61

8–1 Average performance of casual (no prior training) and educational users (benefiting of background knowledge) on easy, medium, and hard puzzles. The black dots above the box represent outliers more than 3/2 times of the upper quartile while the ones below represent outliers less than 3/2 times of the lower quartile. 72

8–2 Sequential flow of instructor registration. 74

8–3 Abstract flow of assignment registration. 75

8–4 Sequence flow of assignment creation. 76

8–5 Badge received after completing the assignment. 78

9–1 Average performance of rookies ($\leq$ 20 puzzles) and expert players ($\geq$ 40 puzzles) on easy, medium, and hard puzzles. The black dots above the box represent outliers with more than 3/2 times of the upper quartile. The black dots below represent outliers with less than 3/2 times of the lower quartile. 81

9–2 User profile page. Annotations are in the same order as mentioned in section 10.1. 85

10–1 Phylo crowd-sourcing system for local improvement of multiple genome alignments. 87

10–2 Change of entropy after aggregation of user aligned puzzles to the initial alignment. The x-axis represents the difference between the entropy of the machine aligned sequences and user aligned sequences. Positive values indicate an improvement of the alignment. 88

11–1 Promising puzzle prediction using LSTM. The graph was created using 480 LSTM cells, *mae* loss function and *adam* optimizer. Validation accuracy achieved: 69.4%. 95

11–2 Average confusion matrix for promising puzzle prediction using kNN. 96

11–3 Average confusion matrix for promising puzzle prediction using SVM. 97

11–4 Progression of high scores. The orange line shows the normalized high scores since the release of the puzzles. The blue curve plots the number of solutions collected. 109

CHAPTER 1
Introduction and related works

Human computation is a new and evolving research area that centers around harnessing human intelligence to solve computational problems that are beyond the scope of existing Artificial Intelligence (AI) algorithms. It has emerged as a popular approach to solve large-scale scientific problems in astronomy [68], molecular biology [15, 34], neuroscience [36], and even quantum physics [43]. Oher ways to utilize human computers is through *games-with-a-purpose* or *citizen science* resourcefulness. Such processes possess great potential for advancing biomedical science. Games like Foldit [35], Phylo [34], and Eyewire [36] have been running successfully for several years. A few other games in this genre include: EteRNA for RNA structure design [42], The Cure for breast cancer prognosis prediction [26], Dizeez for gene annotation [47], Ribo for RNA structural alignments [73] and MalariaSpot for image analysis [50]. Many of these initiatives have succeeded in independently addressing challenging technical problems through human computation, improving science education, and generally raising scientific awareness. This has raised wide interest in new applications of human computation towards science. However, the trail from a good idea to a successful citizen science game remains highly challenging. Not only the positives learned from such games should be recognized but also identifying possible pitfalls and its appropriate human-powered solutions are equally important. Therefore, such games still need to be ameliorated, need to have better problem solving

potential, need to be fun, and need to reach a large audience that remain engaged for a long-term.

With more applications of this technology underway, it is important to identify the factors that contributed to the successes of this approach, and improve the aspects that did not work as well as expected. To this end, the analysis of the data collected by the earliest systems may reveal important patterns that could benefit to the next generation of scientific games.

In 2010, we released a game-with-a-purpose named Phylo[1] , which aims to help in improving the accuracy of the comparison of DNA data [34, 39]. This problem, known as the multiple sequence alignment (MSA), is an essential piece of a vast body of biological studies [21]. A multiple sequence alignment (MSA) requires at least three homologous nucleotide or amino acid sequences. Two sequence alignments are commonly referred to as a pairwise alignment. The alignment, whether multiple or pairwise, is obtained by inserting gaps into sequences such that the resulting sequences all have the same length L. Consequently, an alignment of N sequences can be arranged in a matrix of N rows and L columns, with a motive to place in the same column characters that are homologous (i.e. derived from a same common ancestor), possibly inserting gap characters to account for the presence of insertions and deletions. MSAs are one best way to illustrate the evolutionary relations among

[1] `https://phylo.cs.mcgill.ca`

the sequences. It can also be used to reveal conserved and variable sites within different species. MSAs can provide essential information on their evolutionary and functional relationships. Therefore, it has become an essential prerequisite for genomic analysis pipelines and many downstream computational modes for homology modeling, secondary structure prediction, and phylogenetic reconstruction. They may further be used to derive profiles using hidden markov models [64] that can be used to identification of distantly related members of the family. As the enormous increase of biological sequence data has led to the requirement of large-scale sequence comparison of evolutionarily divergent sets of sequences, the performance and quality of MSA techniques is now more important than ever. Although the problem of pairwise sequence alignment can be solved optimally in quadratic time [23], calculating an optimal MSA is an $\mathcal{NP}$-hard problem [1]. A large number of fast and efficient heuristics have been developed to align genomic DNA sequences [8], but the solutions returned by these algorithms are potentially suboptimal. With the rapid expansion of genome sequencing technologies, the shear quantity of DNA sequences to be aligned (potentially hundreds of sequences of several billion characters each) makes the task of producing and maintaining highly accurate MSA intractable for small groups of experts. Because manual curation is a necessary step to guarantee the quality of biological sequence alignments, a crowdsourcing solution appears to be a perfect strategy to address this bottleneck [11].

Phylo aims to improve MSA solutions already pre-calculated by state-of-the-art algorithms. Eventually, in the case a MSA cannot be improved, it can also serve as

a certificate to validate the input data. Phylo starts from a computationally calculated MSA of multiple vertebrate genomes [59], identities portions of the alignment that are potentially sub-optimal, and transforms them into small puzzles that are dispatched to players (See Figure 1–1). Once a player has completed a puzzle, the solution found is returned to our server for evaluation. If the alignment found by the player is deemed superior to the original computer-produced alignment (based on a parsimony-based scoring scheme [75]), it replaces the segment from where it was extracted in the global alignment. Phylo thus contributes to improving a resource (multiple genome alignment) that is used every day by researchers in biology and genetics (e.g. detection of important DNA motifs associated with a biomolecular function, inference of ancestral genomes), while being a fun and educational game for non-experts.

Phylo's tasks are presented as in a casual tile matching game, where DNA alignment problems are embedded in a puzzle accessible to any player, including those without any prior training in biology or computer science. Most importantly, the game is intuitive and allows the players to play Phylo without understanding the underlying biology, and without completing any tutorial. This broadens the spectrum of participants and taps into the computing power generated by regular, non-scientist human computers. Special care is taken to present the puzzles in a fun, exciting, and accessible manner while retaining the scientific interpretability of the data. Ribo [73] on the other hand is yet another citizen science game with a difference that it aims to solve the RNA based multiple sequence alignment problem. It comes with an

additional alignment objective of aligning the sequences taking its secondary structure into consideration as well. Albeit, in the upcoming version, Phylo has Ribo integrated in it. This was done to utilize the huge count of players playing Phylo in aligning RNA sequences as well. Figure 1–2 represents the upcoming version of Ribo which has now been integrated into Phylo game as an option to align RNA sequences.

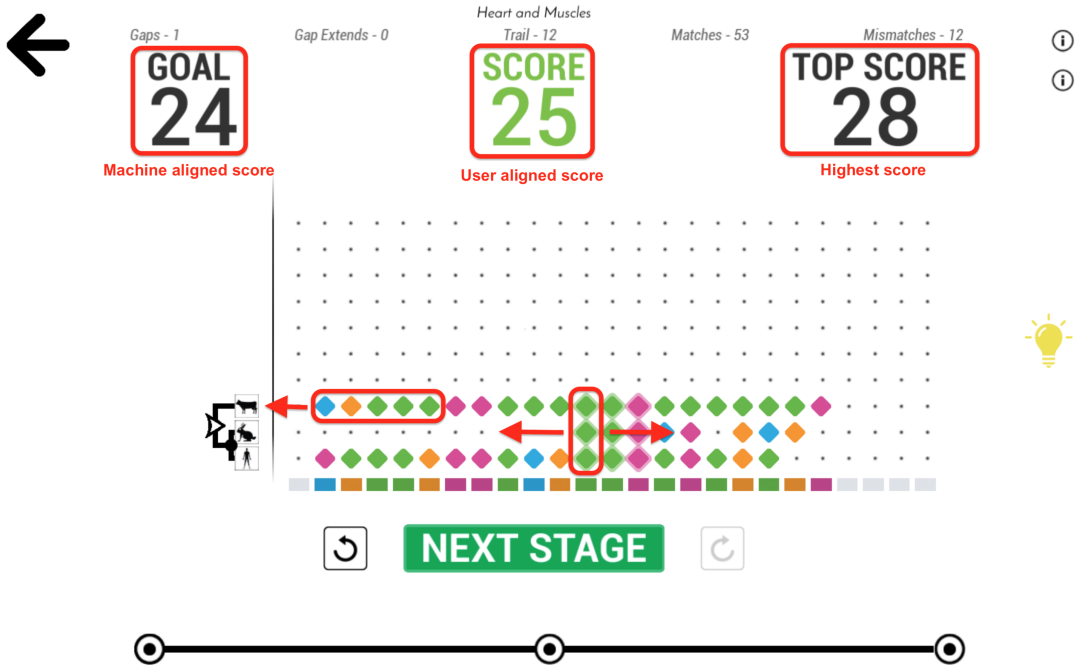

Figure 1–1: Interface of Phylo (2017 release). Each row is composed of a sequence of bricks of 4 different colors representing the 4 nucleotides A, C, G, and T. These sequences have been extracted from the DNA of different species represented with a icon on the left of the grid. Participants move the tiles left or right in order to maximize the number of color matches in each column. Although, the order of the bricks cannot be changed. Color mismatches and gaps are thus unavoidable and bring penalties. The phylogenetic tree on the left indicates the priority in which the rows should be aligned.

- **Game play:** Phylo is a single player casual game. The basic game play consists of moving the tiles horizontally- left or right– on a 10x25 game board such that the pattern becomes more similar in a column. The objective of the player is to beat the par score (machine aligned score) and move to the next level.

- **Scoring:**

  - *DNA puzzles scoring scheme:* We followed up to the scoring scheme of the classical version of the game. It has to be something that can be used to compute the score in real time at the same time being very intuitive. This ensures that the scoring can provide proper feedback to users. To score an alignment we first need the ancestral sequences based on the alignemnt's phylogenetic tree using a maximum parsimony approach [24]. The metric used for DNA puzzle/alignment score calculation: gap opening: -4, gap extension: -1, mismatch: -1, match: +1, trail error: -1. Score is the calculated as: ((match * 1) + (mismatch * (-1)) + gapOpen*(-4)+ gapExtend*(-1) + trail*(-1)); *Trail error* attribute in score calculation was added to ensure gaps across the ends of DNA segments during aggregation.

  - *RNA puzzle scoring scheme:* RNA scoring is the same as proposed in the citizen science game: Ribo [73]. The scoring scheme that Ribo [73] uses for evaluating the RNA sequences and structures is based upon the nucleotide sequence scoring scheme derived using a Markovian transition model (States et al. 1991). PAM (Point Accepted Mutation, for proteins)

matrices were derived using the same approach [18]. The match-mismatch metric used in the Ribo paper is +5 for a match, -4 for a mismatch, -5 for a gap opening and -2 for a gap extend. The overall score is the sum of all these parameters multiplied to the number of occurrences. This is a proven scoring scheme [25]. The reason for selecting a relatively low gap-open and gap-extend penalties was because indels are thought to be relatively frequent in RNA [49]. Bonuses are given for matching base-pairs such that it should match exceed the penalty for double mismatch resulting from structure-neutral variation (e.g. A·U to G·C) as well as tolerate the indels that are required to explain base-pair conservation. Therefore a bonus of +12 was selected for aligning base-pairs in Ribo.

In addition, Phylo also provides scientist's an open and freely accessible web interface that enables them to crowdsource their sequences. We call this web interface as OpenPhylo. It also has a teaching portal, which helps instructors to teach multiple-sequence alignment concepts to students. Essentially, we have worked on the following components to enhance both Phylo and OpenPhylo based on our observations from the classical version of Phylo:

1. **Promising puzzle detection:** This is the process of identifying regions in MSA which have some scope of improvement. We have used Convolutional Neural Networks (CNN) to solve this problem. CNN has greatly enhanced image recognition tasks and is the prevailing state-of-the-art for such objectives [69]. In a CNN each neuron is connected in overlapping tiles giving the network locality in a two-dimensional space. This serves nicely for input datasets that

can be used as images or grids [69]. Our source of inspiration is from image recognition tasks since the problem of predicting moves from game states can be seen as an image recognition problem. This is analogus to our problem statement where we are finding different states in segments of MSA with some scope of improvement. Other sources of inspiration include anything related to machine learning or AI in games. In *ImageNet Classification with Deep Convolutional Neural Networks* a large deep convolutional neural network is trained on millions of high-resolution images substantially improving previous state-of-the-art in the ImageNet LSVRC-2010 contest [38]. Even though the objective here is to determine what kind of object is visible in an image it is similar to our problem since the game board can be seen as an image with 3 channels as for RGB. Likewise, the object we are trying to identify consists of regions in MSA, therefore such an approach is suitable for our problem as well. We use the identified region as puzzles in Phylo. We will discuss the proposed methodology in chapter 3.

2. **Difficulty prediction:** Crowdsourcing or human computation are popular means to obtain labeled data at moderate costs, in case of Phylo it is obtaining quality solutions, which can then be used in solving the problem of multiple sequence alignment. To mitigate the problem of low-quality solutions in this context, multiple human factors must be considered to identify and deal with players who provide such solutions [71]. However, one aspect that plays a prominent role is the inherent difficulty of puzzles to be solved and how this affects the reliability of the solutions that players provide to such

puzzles. Therefore, we investigate in this preliminary study this connection using various machine learning approaches. We find that there is indeed a relationship between the user solutions and the difficulty level of puzzles. We published our findings in the fifth AAAI conference on Human Computation and Crowdsourcing [67]. In this report we have tried a different labeling scheme to further enhance the difficulty predictions. This is used to correctly define the difficulty level of the extracted puzzles which later helps in routing of tasks based on its difficulty to rightfully skilled player. The propoed methodology will be discussed in chapter 5 and 6.

3. **Routing:** Routing is the process of sending the correct difficulty level of puzzles to the rightfully skilled player. This is important to enhance the latency of the system. Routing is an integration of two unique processes. One to decide the user expertise level and the other to decide the difficulty level of microtask and number of times microtasks that have been completed. Various analytical approaches have already been evaluated to perform routing of microtask during croudsourcing. The benefit of analytical techniques to determine the performance of individuals has been examined and verified in various applications within several research communities (e.g., [2, 32, 56]). Essentially, these studies examine the user expertise level in predicting individual differences with respect to the similar jobs. Human computation based problems can vary with respect to a mixture of personal factors, behaviour presets, and the social impact modifications due to the surrounding environment. Hence, these factors may have critical effect on human functioning, and therefore play

a key role in human based systems [13]. Several studies within this area have focused on expertise assessment research, exploring expectation-maximization for expertise level prediction and using it for job matchmaking. However, this approach works best when there is a large number of workers as it relies on cross-examination of as many responses to the tasks as possible to improve reliability. The theoretical foundations of our study are based on the well-established explore-exploit dilemma [76]. The current modus operandi in the majority of crowdsourcing platforms is that the central authority of the system coordinates the assignment process. However, a number of techniques have been proposed in literature with the aim of improving task-routing and assignment in crowdsourcing. Previous work, an online crowdsourcing scenario, Ho et al. [28] explore worker assignment to many heterogeneous tasks based on a two-phase exploration-exploitation algorithm which aims to define workers based on their skill levels. Chapter 7 explains the proposed routing process to be used in the upcoming version of Phylo.

4. **Teaching Portal:** The portal is designed for instructors to educate their students the concept of multiple sequence alignments. Educators can set up assignments in less than a minutes. With teaching portal, they can manage assignments and track students' performance in one convenient place. Instructors can get instant feedback and track a student's progress. Teaching portal also helps free up educator time so they can focus on what they do best: teaching. Students just have to play anywhere, anytime, and on any device: web version or the mobile version of Phylo. This is very important for us as it

helps us evaluate the performance of casual players against those who play Phylo knowing its scientific relevance. This helps us evaluate the effectiveness of Phylo interface. We have discussed the architecture and procedure followed for teaching portal in chapter 8.

5. **User feedback:** The need for feedback in human-computer interaction has long been considered as a required norm. Many authors have published guidelines in an attempt to assist application developers in providing relevant feedback to the end-user [63, 5]. In spite of this, adequate provision of feedback proves to be a subtle art. We have all used applications which do not supply adequate feedback, which also acts as a demotivation towrds its actual objective. This makes it especially difficult for the end users to stay motivated to uphold the performance in completing the task. With the increase in human-computer interaction proportionally to the increase in its integration to different domain, the feedback mechanism has now become even more significant [4]. Human-computer interaction or crowdsorcing based systems comprise of two level of interations [30]:

    - *Crowd-related interactions:* Crowd-related interactions are interactions provided by the crowdsourcing platform between the crowd and the platform. For instance, the interaction of player with the Phylo or OpenPhylo interface.

- *Crowdsourcer-related interactions:* Crowdsourcer-related interactions are interactions provided by the crowdsourcing platform between the crowdsourcer and the platform which for us is the feedback provided by Phylo/OpenPhylo to its intended audience. User feedback falls under this category. These interactions include, but are not limited to [61, 30]:

  (a) Visibility of microtasks and its aggregation outcome to the crowdworkers.

  (b) Providing a threshold mechanism for the quantity of the obtained results to ensure a minimum and/or maximum quantity is met.

  (c) The provision of feedback about both immediate and past interaction.

  (d) The extension of feedback to provide explanations of system actions, thus promoting user understanding of the purpose of the application.

  (e) The promotion of the use of graphical rather than textual feedback mechanisms.

  In chapter 9 of the report, we discuss about the feedback mechanism used in Phylo interface.

6. **Aggregation:** Accomplishing complex tasks efficiently is often a challenge, especially when the time and resource required for the tasks is limited. Tasks such as multiple sequence alignment, RNA secondary structure prediction, etc. are hard to come up with a best possible result because they seem to require excessive amount of computational power. By using alternatives to solve the same

problem results in sub-optimal solutions. However, research shows that concrete plans with actionable steps enable people to complete their tasks better and faster [37]. Breaking a large task down into a series of smaller, microtasks with scoped context results in higher quality [12]. Various approaches to decompose tasks into smaller microtasks already exist. It ranges from manual to algorithmic [7]. It has been shown that microproductivity, or the transformation of information work into micro-work, will have a significant impact on when and how people work, enabling individuals to efficiently and easily complete large tasks that currently seem challenging [70]. The rapid developments in micro-work, micro-volunteering, and micro-learning open up new frontiers for the future of microproductivity but ends with some unanswered questions:

(a) When and where should microtasks be embedded?

(b) Can microtasks be used to build knowledge?

(c) How can we measure outcomes and contribution towards a large task?

In chapter 10 of this report, we will address all these questions in reference to Phylo.

The entire OpenPhylo framework combines the effectiveness of human computation through Phylo with some applications of machine learning. We investigate this framework for solving Multiple Sequence Alignment problem for DNA and RNA. Based on the results presented in this report, we find that actively coupling games with machine learning provides a reliable and scalable approach to solving bioinformatics problems. In addition, this report also mentions the performance analysis

of these systems: Phylo, OpenPhylo and Ribo. These results will further enable us to identify patterns that contributed to its successes, and also to diagnose possible snags. Based on these statistics we further propose new methodologies for Phylo and OpenPhylo. The proposed architecture is shown in Figure 1–3. Further, the report will first address the intended audience chapter 2 followed by the puzzle extraction procedure in chapter 3, then it will individually cover each of the scientific aspect in the same order as mentioned above.

Given below are the links to both the mentioned products:
Link to Phylo: `https://phylo.cs.mcgill.ca`
Link to OpenPhylo: `https://kovik.cs.mcgill.ca`

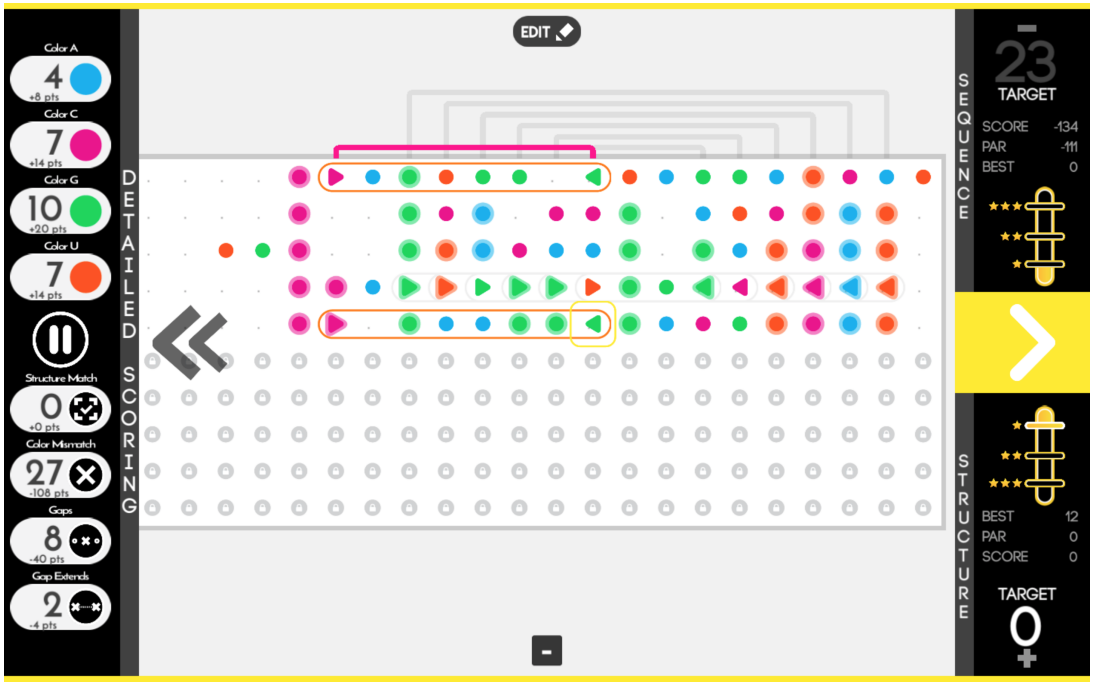

Figure 1–2: Interface of RNA version in Phylo (yet to be release). Each row is composed of a sequence of tiles of 4 different colors representing the letters from the RNA alphabet: A, C, U and G. These sequences have been extracted from the RNA families stored in rfam database. Participants move the tiles left or right in order to maximize the number of color matches in each column. Also, there is an additional scoring scheme for aligning the secondary structures in RNA alignment. The order of the bricks cannot be changed, however, you can edit the secondary structure based on your requirements. Color mismatches and gaps are thus unavoidable and bring penalties. The orange colored boundries in this figure reflect possible secondary structure similarity in the RNA puzzle.

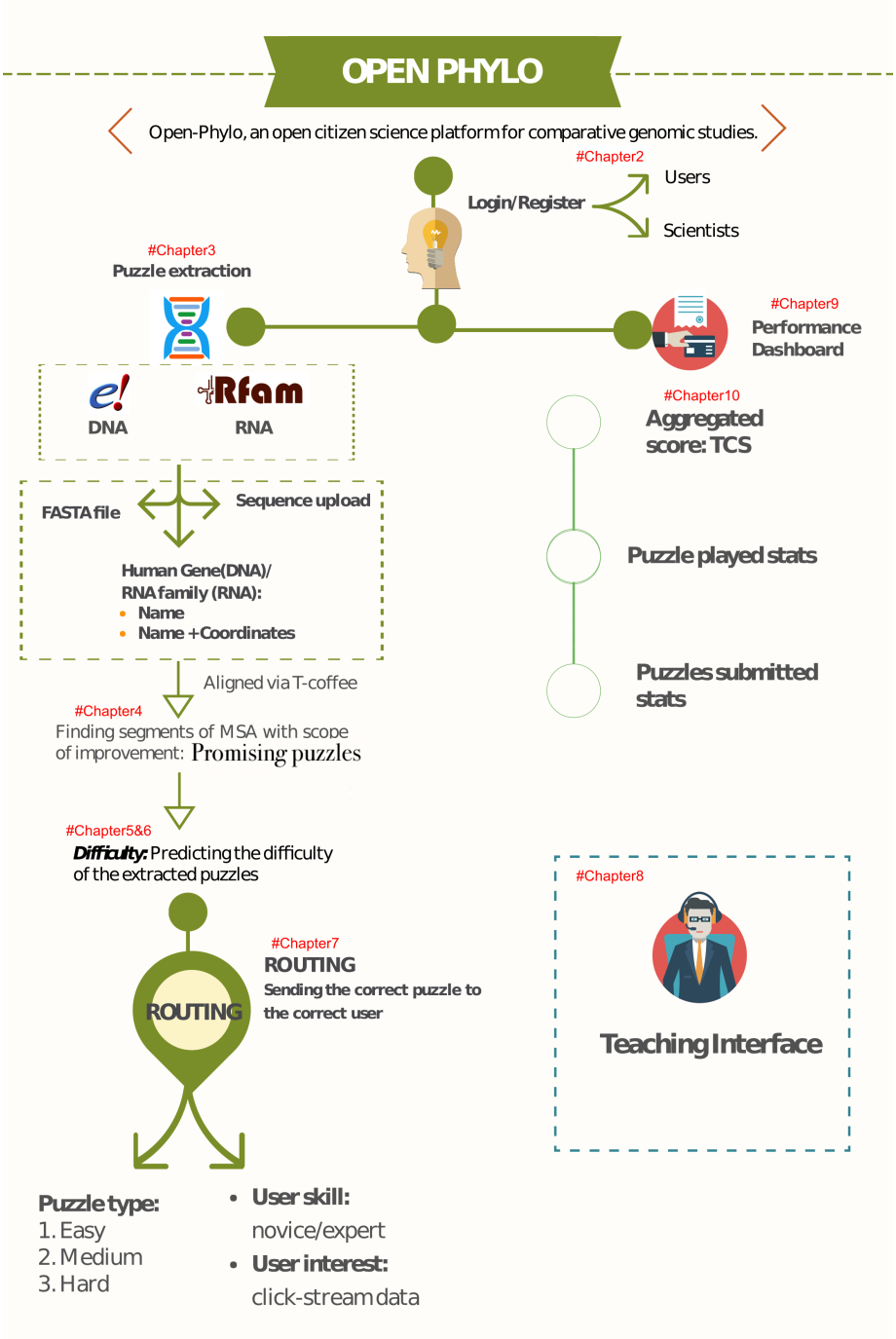

Figure 1–3: Architecture of OpenPhylo. Each annotation represents a module in OpenPhylo and has been discussed as a chapter in this report.

CHAPTER 2
Intended audience

In 2013, a new citizen science platform OpenPhylo [39] was developed to ensure easy access to all. The traditional human computation projects or games with a purpose (GWAP) are intended to benefit the researchers who designed the system, i.e. only they can select the problems that are submitted to the community, and access and analyze the solutions submitted by the players. OpenPhylo is a way to go against this paradigm and enable the access of human-computing resources to the whole scientific community. The unique feature of OpenPhylo is that it is not only computing by the people, but also for the people [74]. In this chapter, we describe types of intended audience for Phylo. Registering to OpenPhylo gives the audience's visualization of their accolades achieved by playing Phylo. We provide recent usage statistics, and illustrate the scientific impact of this technology on genomic research based on the type of registration by the user. OpenPhylo offers three different types of registrations (a refrence to its architecture is shown in Figure 2–1):

1. *Users:* They are the end users of Phylo game. These players can either register from OpenPhylo or Phylo. They can use OpenPhylo or the user profile page in Phylo to visualize their contribution to science and other statistics. They can use the same user credentials to play Phylo for fun. There are three sub-categories of users:

(a) **Registered users:** The registered users have access to the following features:

- Visualizing user statistics
- Scientific contributions made whilst playing Phylo
- Feedback on their achievements
- Account conversion from user to scientist
- Quiz to assess their knowledge in bioinformatics

(b) **Students:** are also a part of registered users except that they have an added achievement badge in their list of badges. If they complete the assigned task in Phylo, the assignment completion badge gets enabled. This architecture is covered in chapter 8 of this report.

(c) **Guests:** Players who do not register via Phylo/OpenPhylo are considered as guest players. A unique token id is assigned to them, such that all their solutions are mapped to the same token id. This token stays valid for infinite duration such that player playing in different times but using the same machine still store the solution as the same guest player. Phylo encourages guest player to ensure a smooth access to the game and later motivates them to register for Phylo and become a regular contributor to science. Such users miss out on user feedbacks that is given to registered players.

2. *Scientists:* In addition to all the features that a user enjoys, scientists also have rights to submit DNA/RNA puzzles. However, such an account needs a prior approval of their email Id. OpenPhylo provides the following features to the scientists:

    - Submit DNA puzzles using one of the following options:

        (a) Gene based DNA puzzle extraction

        (b) upload FASTA file for DNA puzzle extraction

        (c) copy paste the sequences manually for DNA puzzles extraction

    - Submit RNA puzzles using one of the following options:

        (a) RNAfamily based puzzle extraction

        (b) Stockholm file based puzzle extraction

        (c) FASTA file based RNA puzzle extraction

        (d) Copy paste the sequences manually for RNA puzzle extraction

3. *Instructors:* Instructors have discrete rights for teaching portal only. These users also need authorization of their email Ids. Instructors are actors of the teaching portal use case scenario. Instructors can use OpenPhylo to assign set of puzzles to students as a part of their course curriculum. Instructors can use OpenPhylo to teach the problem of Multiple Sequence Alignment (MSA) to students. The assignment here refers to puzzle solving in the Phylo game. For instance, instructors can create a assignment that includes n-number of

puzzles, so each student enrolled in that assignment has to play/solve those many puzzles in Phylo. Instructors can perform following operations using the teaching portal:

(a) Create new assignments

(b) Edit or view created assignments

(c) Check student progess

(d) Download the selected assignment results as a csv file

A more detailed information is available in chapter 8 where we discuss primarily about the teaching portal of OpenPhylo.

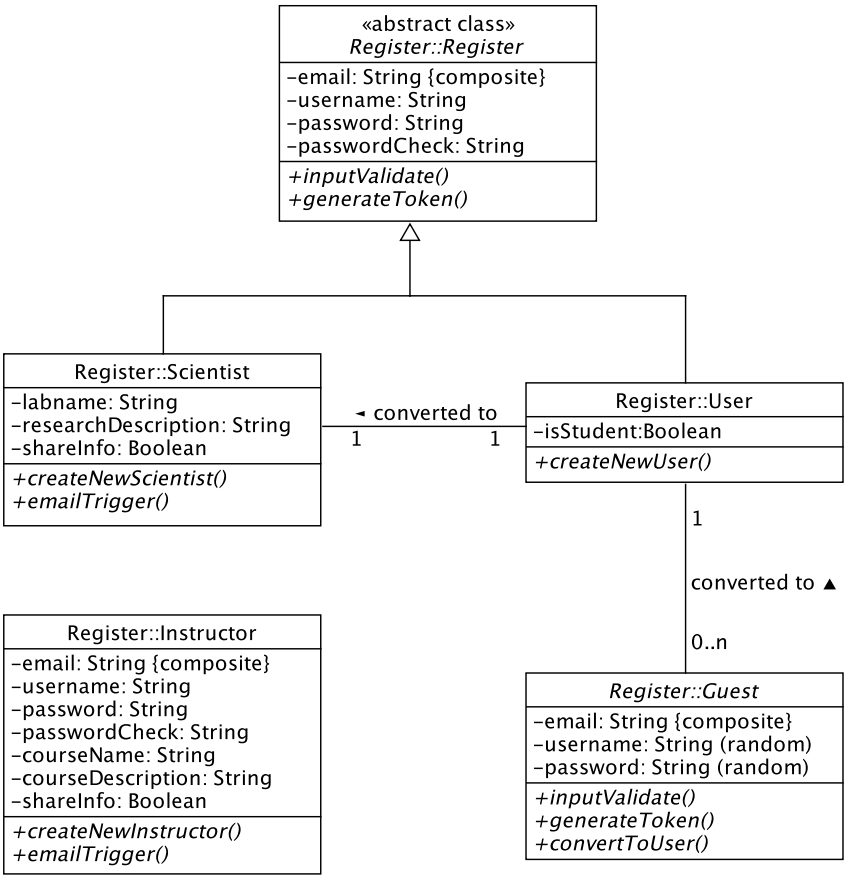

Figure 2–1: This is a static representation of registration process in OpenPhylo. The diagram represents attributes, operations (or methods), and the relationships among the various actors/registration classes of OpenPhylo. The register package consists of all the registration related classes. User and Scientist extend an abstract class with mandatory attributes for all registration process. User account can be converted to scientist account and Guest user can also convert their account to a registered user account without losing any progress. Instructor registration has been kept discrete and independent of other actor registration.

CHAPTER 3
Puzzle extraction

Puzzle extraction in OpenPhylo is the pipelined process of generating puzzles/alignments for the Phylo game. This feature is available only to scientists. OpenPhylo supports both DNA and RNA based puzzle extraction.

3.1 DNA based puzzle extraction

OpenPhylo has the following options for scientists for DNA based puzzle extraction:

1. Gene based puzzle extraction
2. FASTA file based puzzle extraction [51, 39]
3. Sequence based puzzle extraction

The same is shown in Figure 3–1. Currently, the grid of the game supports up to 10 sequences. Thus, we aim to improve MSA of similar height. Puzzle extraction is capable of producing alignments with 3 or more sequences. Gene-based puzzle extraction for DNA is achieved employing Ensembl Compara API [72] where we take the first 2000 nucleotides (promoter region in DNA sequence) for puzzle extraction via gene. FASTA file puzzle extraction takes the FASTA file and associated information like disease name, disease category and phylogenetic tree as the optional inputs. OpenPhylo first aligns these sequences using T-Coffee [54] before translating them into puzzles. If the phylogenetic tree is not available then the application uses the tree

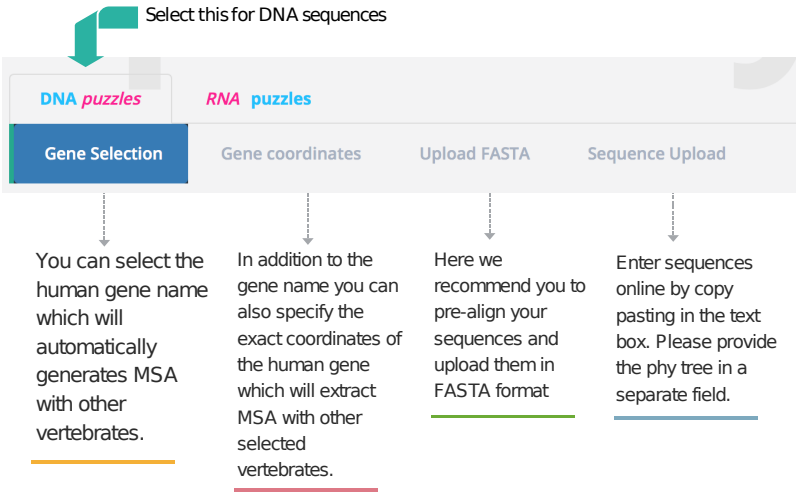

Figure 3–1: Puzzle generation options available to scientists for DNA alignments.

produced by T-Coffee. T-Coffee generates the phylogenetic tree in Newick format. We transform this tree into a binary tree because of the game's compatibility to binary phylogenetic trees only. Sequence-based puzzle extraction follows the same steps as FASTA file based puzzle extraction. Newline character is the delimiter used to differentiate between various sequences. The phylogenetic tree is compulsory for sequence-based upload since the species for the puzzles are obtained from it. We have used CNN based machine learning model for finding regions with some scope of enhancement in the Multiple Sequence Alignments (MSAs) produced via state-of-art algorithms like MUSCLE, T-Coffee, etc. Therefore, we first take an MSA section as input and transform it into an image. This image is further fed to the trained CNNmodel which predicts whether the selected puzzle is promising or not. This model for promising segment detection performs well with an accuracy of 81%. The algorithm is explained in chapter 4 of this report. If a puzzle is claimed to be

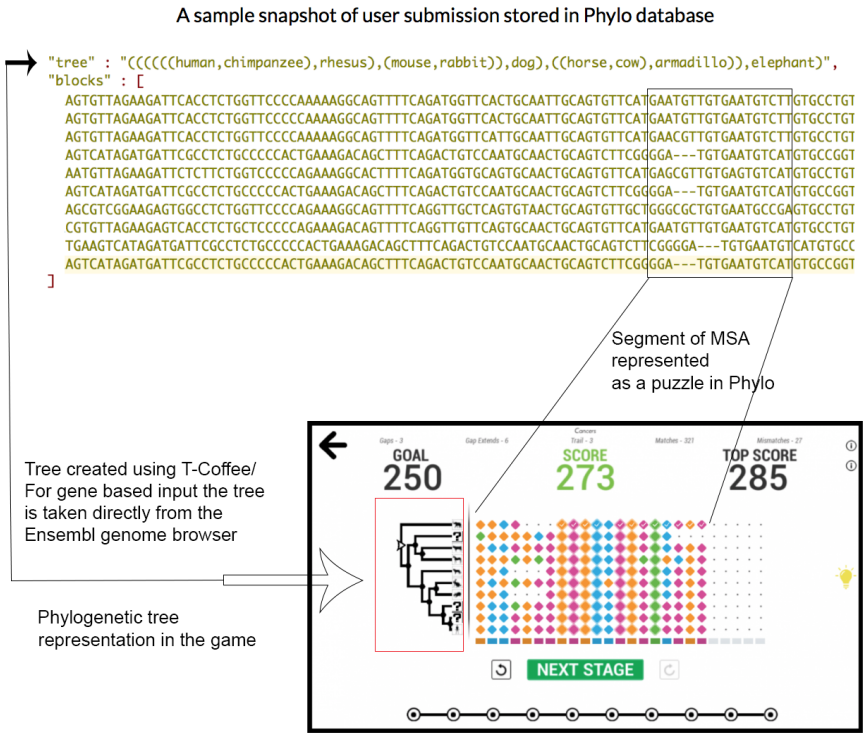

Figure 3–2: Puzzle generation in game using segments of MSA

promising, it is further passed to difficulty prediction algorithm which decides of the selected puzzle. Difficulty prediction is explained in chapter 5 of this report.

Further, each selected DNA alignment is assigned to a disease category. These categories enable us to promote the puzzles to Phylo players based on their interest in disease categories. Available categories are still the same in the upcoming version as well: "Blood and immune system diseases", "Brain and nervous system", "Cancers",

"Heart and muscles diseases", "Infectious disease", "Digestive and respiratory systems diseases", and "Metabolic disorders". If your sequences do not belong to one of these categories, or simply are not related to any disease, you can use the "other category" option. Gene to disease mapping is done based on this systematic analysis, comparison, and integration of disease based human genetic association data and mouse genetic phenotye information [77].

The puzzles extracted are stored in Phylo database and are crowdsourced to generate better results. Later, we merge the solutions received from the users to enhance the original MSA from which the puzzle was extracted. This is known as the aggregation and is explained in chapter 11.

3.2 RNA based puzzle extraction

Here, we evaluate Ribo and integrate it to the Phylo interface, a human-computing game that aims to improve the accuracy of RNA alignments already stored in Rfam. The Rfam database maintains alignments, consensus secondary structures, and corresponding annotations for RNA families. Its primary purpose is to automate and store accurate annotation of non-coding RNAs in genomic sequences [10]. However, the alignments stored in this database often have some scope of improvement. In fact, the Rfam consortium has released a open call for participation, asking its users to submit new or improved RNA alignments (`http://rfam.sanger.ac.uk/submit_alignment`). OpenPhylo provides the following RNA submission options for puzzle extraction to scientists:

1. RNA family based puzzle extration

2. FASTA file type puzzle extraction [51]

3. Stockholm type puzzle extraction [53]

4. Sequence based puzzle extraction

Figure 3–3 reflects all the above listed categories. For RNA, the grid of the game

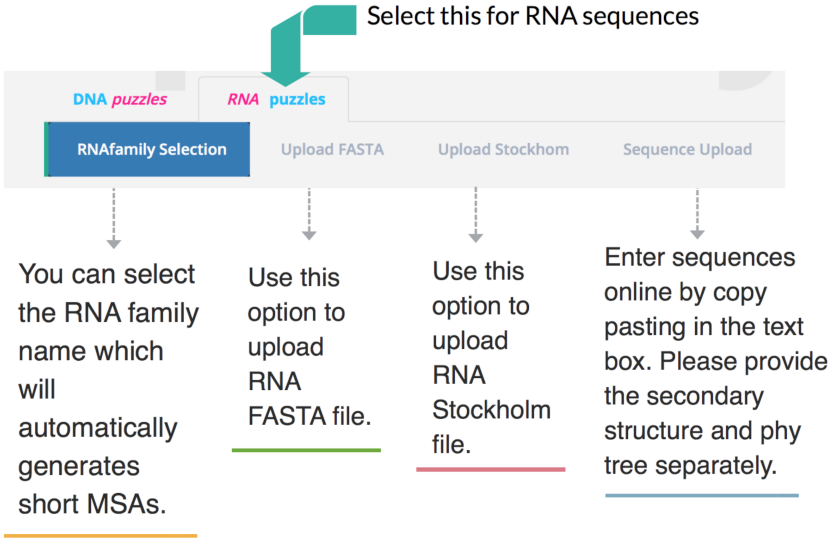

Figure 3–3: Puzzle generation options available to the end users for RNA alignments.

supports larger as well as more number of sequences. Puzzle extraction process for RNA is capable of producing alignments with 3 or more sequences. RNA family-based puzzle extraction is achieved by employing Rfam database [31, 27] which stores alignments of homologous RNAs. We selected those RNA sequences where the average similarity is less than 50% and contains at least 10% of gaps. This metric is important because sequences with low sequence similarity are hard to align as mentioned in the citizen science game for RNA alignment: Ribo [73]. Because the

experimentally determined structure may not always be available, therefore, we ignored the consensus structure available in the original Rfam database. Instead, we used RNAfold to predict the secondary structure using the maximum expected accuracy (MEA)[48, 29] for every sequence. Therefore, it is important to keep in mind that in this study, the Rfam alignments benefit of an information not used by the players. We removed all the empty columns (i.e. columns containing only gaps) from the sub-alignments, and extracted all continuous regions of 20 columns for the classic version of the game. We store the entire alignment for the expert version. We further removed from each region the base-pairs that were not included within this region. In other words, if a nucleotide has a predicted interaction with another nucleotide outside of the region of interest, this base-pair is not included in the extracted RNA puzzle. In case of the expert version of the game, if the user provides the secondary structure along with the alignment, we use the same secondary structure. If the secondary structure is not provided by the users then, we generate one using RNAfold.

In addition, each RNA puzzle extracted via *rnafamily* is also mapped to a short description available in rfam database. Rnafamily consists of grouped sequences with conserved sequence or secondary structure. This conservation is determined by a covariation models build by rfamseq. Initially we crawl through all the rnafamilies names available in the database. We further use the rfam api (documentation: `https://rfam.readthedocs.io/en/latest/api.html`)to extract the sequences from its Stockholm file stored in rfam db. This Stockholm file also contains additional information about the rnafamily. We map this information to the rnafamily so as to

provide better feedback to the users of Phylo. This however is not simulated for RNA sequences submitted via fasta files or sequences. We expect the scientists to submit such information for FASTA/Stockholm/sequence based submissions. Due to insufficient data for RNA alignments collected via Ribo, the difficulty level of RNA puzzles is decided by the length of the sequence: puzzles with length <=20 is considered as *easy* puzzles whilst those with larger and longer sequences are categorised as *hard* puzzles. In the research and build up process towards an accurate difficulty prediction of RNA alignments, we experimented with several methodologies which did not make it to our final product. The insights of which are mentioned in Appendix B of this report.

Secondary structure for RNA alignments are optional but recommended. In the absence of a secondary structure, RNAfold is used to produce one based on the input alignments. Although the phylogenetic tree for RNA sequences are unknown [40], but we allow users to submit one in OpenPhylo.

CHAPTER 4
Promising puzzles

Identifying segments of sequences having some scope of improvement is one of the prominent task to solve the problem of MSA. Here, we propose a novel way to transform identify promising puzzles in MSA. The process involves transforming the MSA into a 3-dimensional tensor which could be utilized by Convolutional Neural Networks (CNNs) for images to decide such regions in MSA.

CNNs emerged from the study of brain's visual cortex, and they can be used in images since the 1980s. In the past few years, due to an increase in amount of computational power and training data availability has resulted in superhuman performance of CNNs in certain visual tasks [66, 62]. We have used CNNs for finding regions with some scope of enhancement in the Multiple Sequence Alignments (MSA) produced via state-of-art algorithms like MUSCLE [20], T-Coffee [54], etc.

The use of CNN is the right fit for achieving the objective because human's visual perception is capable of identifying such segments of MSA, and CNN has its roots in working of human visual system. Similar problems have already been studied using multiple methods, ranging from statistical learning to machine learning methods where CNNs have proved to be very effective in image classification tasks. Deep neural networks form the most recent class of methods used for DNA sequence classification [57]. But, only a few other works aim to perform this task from sequence

and those which do the approach focuses more on the feature extraction rather than the problem itself. The main difference is the format of input which is the data obtained via crowdsourcing through Phylo and the network used to identify segments of MSA with the scope of improvement. This chapter of the report deals more with the studies that suggest there practical benefits of mapping sequences to 3D-tensors or images.

4.1 Objective

The central idea is to find segments of MSA with some scope of improvement. This is a binary classification problem where the goal is to predict whether a selected section of MSA will be interesting for users or not. Puzzles are promising to users only if there is some scope improvement i.e. if the section of alignment is already perfectly aligned by the machine then the players will find it impossible to beat the machine. Therefore, it is very important to find segments of MSA that are misaligned by the machines or state-of-art algorithms like MUSCLE [20], T-Coffee [54], etc.

4.2 Dataset

We analyzed data collected by the classical version of Phylo for over 7 years. The complete dataset consists of 147814 unique alignments or puzzles (solutions). These solutions were submitted by the players while playing puzzles using the game Phylo. The input puzzles were extracted from 575 genomic regions (alignment blocks). Each puzzle comprises a set of 3 to 12 DNA sequences from vertebrate species, including human, of length 10 to 21. The puzzles were played on by different players, and a total of 443018 puzzle solutions were received. Out of this 443018 solution we

choose 147814 unique solutions to avoid skewness in data and also to consider only the unique solutions; 54% positive class and 46% negative class. This dataset was further split into train, validation and test data. 70% of data was used for training the model, 15% as the validation split and the remaining 15% as the test split.

Table 4–1: Number of data points in each class.

Dataset	Total	+ve	-ve
Sequences	147814	80719	67095

Table 4–2: Train, test and validation split of dataset.

Dataset	Total	Train	Validation	Test
Sequences	147814	103469	22172	22172

Label creation

The difference between a puzzle's highest score and its original machine aligned score gives us the measure of improvement scope. An improvement scope of less than and equal to 9 means that the puzzle is not very promising to users and is labeled as 0. An improvement scope of more than 9 represents an interesting or promising puzzle which is labeled as 1.

4.3 Methodology

Puzzle representation

A DNA sequence is a string of letters from the DNA alphabet $\{A, C, G, T, -\}$. Each puzzle is first transformed into a 3D-tensor or images of size $3 \times 25 \times 10$. Selected dimensions of the image is due to the board size of Phylo. We prefer not to rescale the image as it reduces the original quality of the image [19]. The next step is to assign

a color to each letter in DNA alphabet. Following the same color code each puzzle is processed and transformed into an image. Uniformly separate values of colors were chosen to proper separation of each letter. Highest color value of 16777215 (the websafe version of #FFFFFF which represents white color) was given to '-' (gaps) because of its significance in puzzles (analysed in chapter 6). Likewise we placed similar values of 'G' and 'C' due to their importance in DNA structure prediction. Sample puzzle to image conversion is shown below:

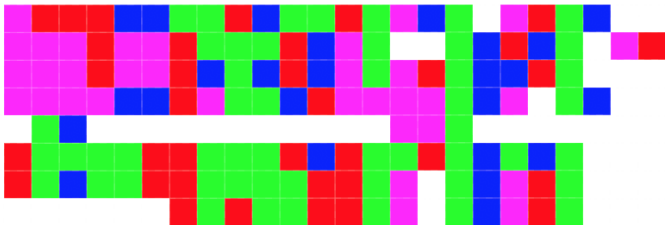

Figure 4–1: Sample DNA puzzle processed into an image.

CNN model used

We chose to use a convolutional neural network since our game board has a grid structure and CNN [41] models has had great performance on other grid based games such as Go [65] or Candycrush [22]. We tested different number of convolutional layers and found that more than two convolution layers do not improve validation accuracy instead increases overfitting. Therefore, we used only two convolution layers. We decided to use (1,1) as the subsample size to avoid any loss of information. The CNN model takes images generated from DNA based MSA sequences. The model used is

shown in Figure 4–2. This is a typical CNN architecture used in image classification. Input for the porposed model is an image of dimension 3X25X10. The entire architecture of the proposed model is shown clearly in Figure 4–2. Relu activation function [41] is to avoid any vanishing gradient problem. This is followed by a linear activation function [45] to convert the problem into a regression problem in the intial layers. Output layer of this model uses Softmax function for the classification task. 10-fold cross validation was used for the accuracy prediction.

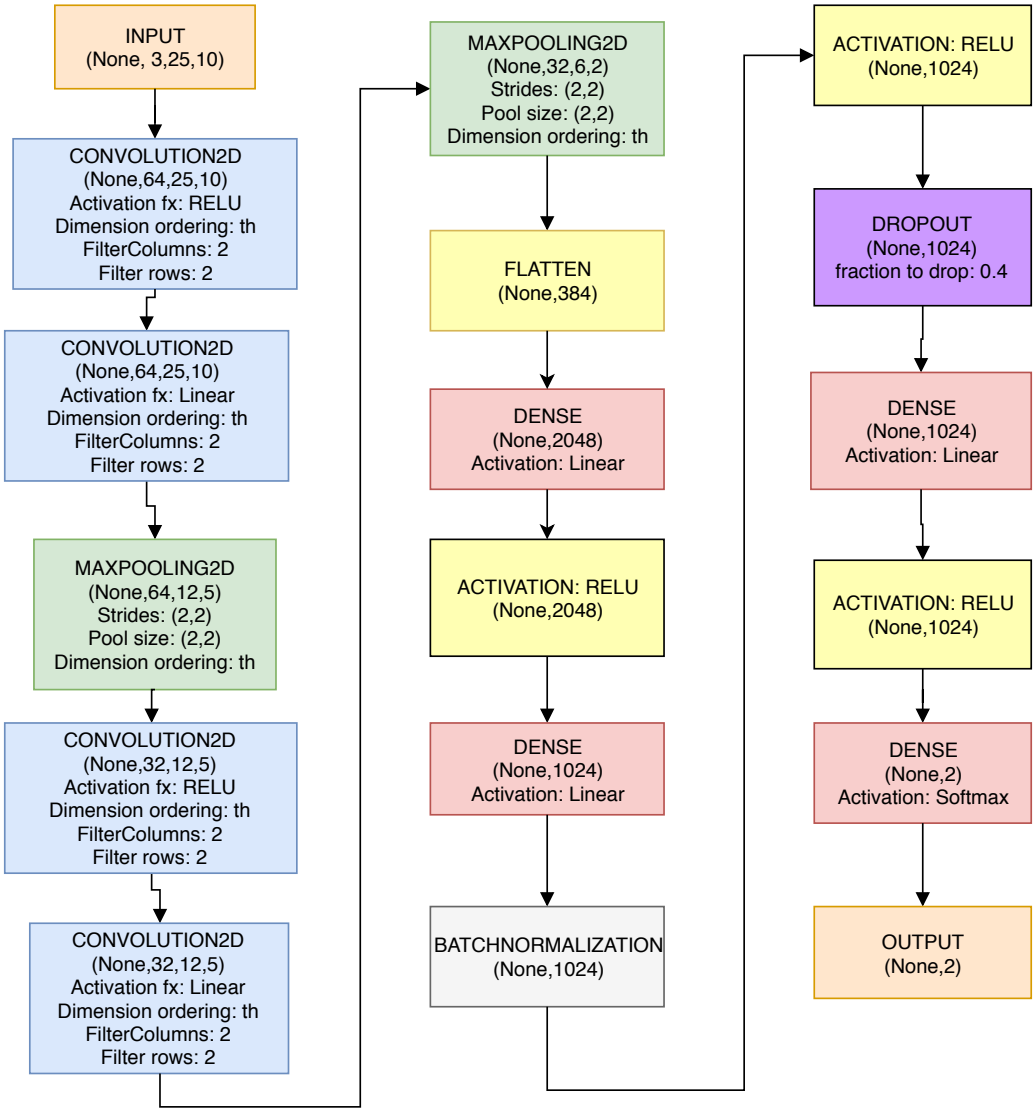

Figure 4–2: CNN model for image classification to decide the scope of improvement in MSA.

Hyperparamters used in the proposed model are listed in table 4–3. The choice of hyperparameters was made using grid search.

Table 4–3: Hyperparameter values for the CNN model used.

Hyperparameters	**Value**
# Epoch	10
Batch size	32
Loss function	Categorical-crossentropy
Optimizer	Adam
Learning rate	0.001
Epsilon	1e-08
Decay	0.5

Computational statistics

Primary computation resource used for this analysis: **Tesla GPU K80 12GB**. Computational statistics using the above GPU:

- *Total time:* 16hrs 21 min
- *Average processing speed:* 1446.77 samples per second
- *Average epoch speed:* 334.00 seconds per epoch
- *Avegare GPU load:* 80.0%
- *Free memory in GPU during processing:* 6.78GB
- *CPU utilization:* 100%

4.4 Results

In this chapter we developed a CNN model that can predict the interesting MSA segments for the Phylo puzzle creation. The approach is novel as it the first to solve this problem using filtered data via HCI. The good performance of model is because

of the application of CNN to the problem statement and this is also a typical example where we can use crowdsourcing for label creation.

Table 4–4 shows the accuracy obtained in different data splits.

Table 4–4: Accuracy obtained using the proposed model.

Data split	**Accuracy**
Training data	85.46%
Validation data	80.67%
Test data	81.01%

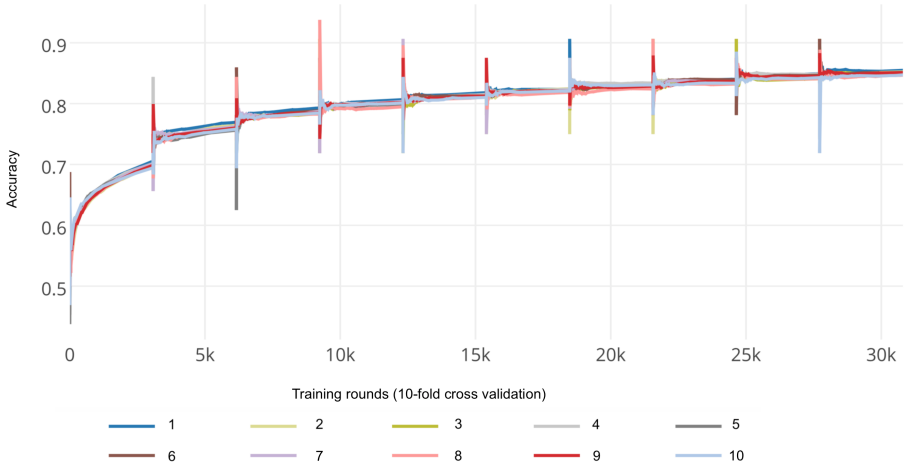

Figure 4–3: Average training accuracy of the proposed CNN model. The graph shows the training curve for 10-fold cross validation. The spikes are an unavoidable consequence of mini-batch Gradient Descent in Adam (batch size=32).

The results suggest that we can use crowdsourcing data to solve one of the biggest concerns in solving the problem of Multiple Sequence Alignment(MSA) which is to

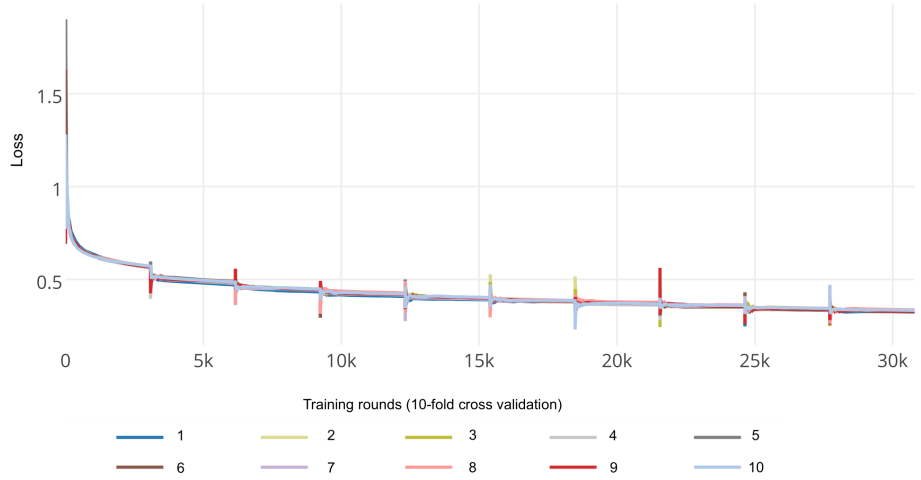

Figure 4–4: Average training loss of the porposed CNN model. The graph shows the loss curve for 10-fold cross validation. The spikes are an unavoidable consequence of mini-batch Gradient Descent in Adam (batch size=32).

indentify region with some scope of improvement. This approach is not limited to puzzle extraction for Phylo, but can be utilized for indentifying regions or segments in MSA that has some scope of improvement.

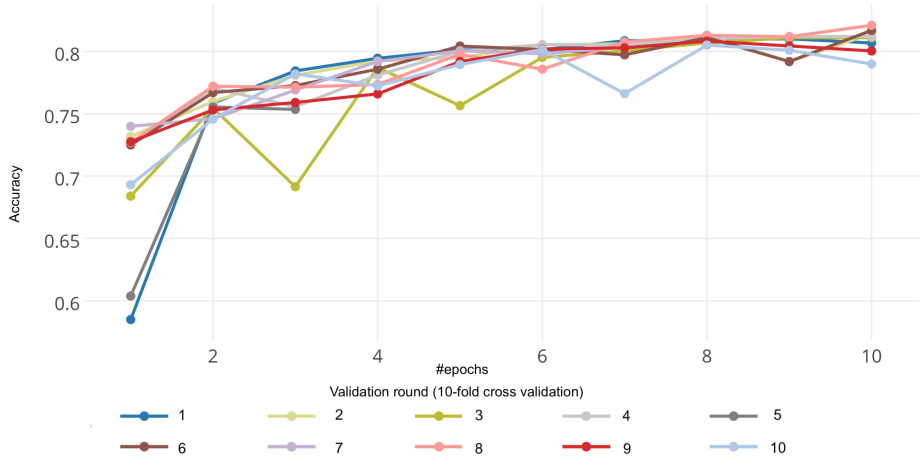

Figure 4–5: Average validation accuracy of the proposed model.

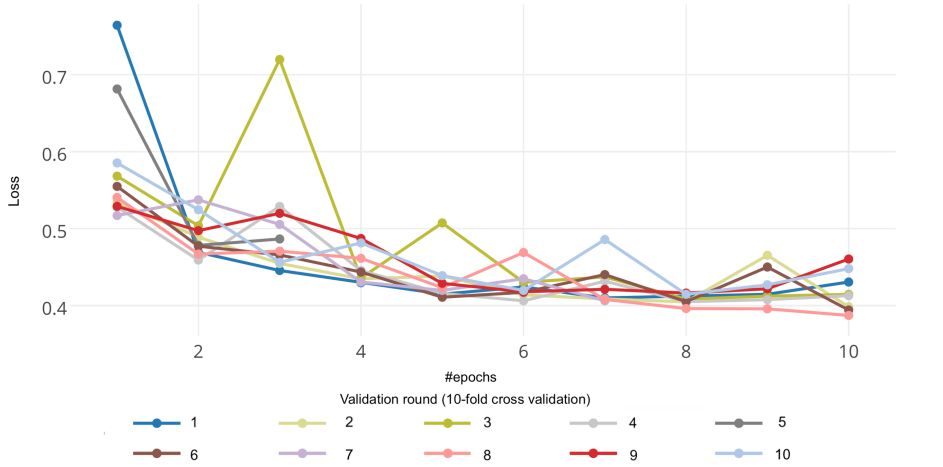

Figure 4–6: Average validation loss of the proposed CNN model.

CHAPTER 5
Difficulty prediction

A key step in Phylo is to assess the difficulty of a puzzle in order to route it to players with the appropriate skill level. Ideally, the difficulty should be estimated automatically, before any player has tried the puzzle. Here, we study how a difficulty level (assessed retrospectively based on the ability of players to improve the alignment score) can be predicted by machine learning algorithms. The puzzles which are considered as interesting are further passed to the difficulty prediction algorithm. We analysed two different machine learning appraoches: one that follows up from the promising puzzle algorithm and the other that uses feature based algorithms.

5.1 Dataset

The dataset used is the same as mentioned in section 4.2.

5.2 Approach 1: Difficulty prediction using feature extraction

5.2.1 Label creation

Each puzzle and alignment block was aligned computationally using several tools (Multiz [9], T-coffee [54], MAFFT [33]). The highest scoring of the machine-computed alignments is called the machine-computed alignment. This score is used as a "par" score that players are challenged to beat. For each puzzle, each solution submitted

by a player is evaluated and the highest scoring solution is retained. Its score is called the human-computed alignment score.

Difference between an alignments' best score and its original score is referred as the scope of improvement. We use this difference to obtain the normalized scores for each puzzle or alignment. Puzzles with the least possibility of improvement were considered as difficult puzzles, so if the average normalized scores of puzzles were less than or equal to 0.33 and greater than 0, then they were considered as difficult to align. Similarly, a normalized score value between 0.33 and 0.66 was considered as medium difficulty, whereas, a value greater than 0.66 represents easy puzzles.

5.2.2 Methodology

In order to train a machine learning predictor to recognize difficult puzzles, these puzzles need to be represented using a vector of features. We extracted and evaluated the following 11 features calculated from the machine-computed MSA: (1-4) proportions of A, C, G, and T in S, (5) proportion of gaps, (6) mean GC content (this relates to the structural properties of DNA), (7) mean entropy of alignment columns, i.e. entropy of the frequency distribution of the four nucleotides and gaps, (8) average length of sequences, (9) number of sequences, (10) tree-entropy based on depth of the leaves of phylogenetic tree of S, calculated by creating a vector of depth of all the leaf nodes in a tree and then calculating its entropy; we used tree-entropy as feature to account for phylogenetic tree information, (11) score of machine-computed alignment.

For each puzzle, we also compared the score of the best human-computed alignment to the score of the machine-computed alignment and defined the score gain as the difference between the two. Positive score gains correspond to alignments that were better aligned by (some) humans than by algorithms. We further subdivided puzzles based on the value of score gains to get two equal size classes. The puzzles with a score gain greater or equal to 17 were assigned to the positive class (i.e. the class of puzzles where humans have produced significantly improved alignments). The rest were assigned to the negative class (little or no improvement). Figure 5–1 shows the comparison of different features used for difficulty prediction. One very important observation is that the correlation is indeed very strong between number of sequences and tree depth. This is reflected as an upward trend in the plot and also the points are not very dispersed. Figure 5–2 reflects the pearson correlation for the same set of features. However, this coefficient only measures linear correlations and it may completely miss non-lienar relationships. The correlation coefficient ranges from 1 to -1. When the value is close to 1, it means that there is a strong positive correlation. When the value is close to -1, it means there is a string negative correlation [52].

Figure 5–1: Feature comparison between positive and negative class. The blue dots represent negative cases, whereas the green dots represent positive cases.

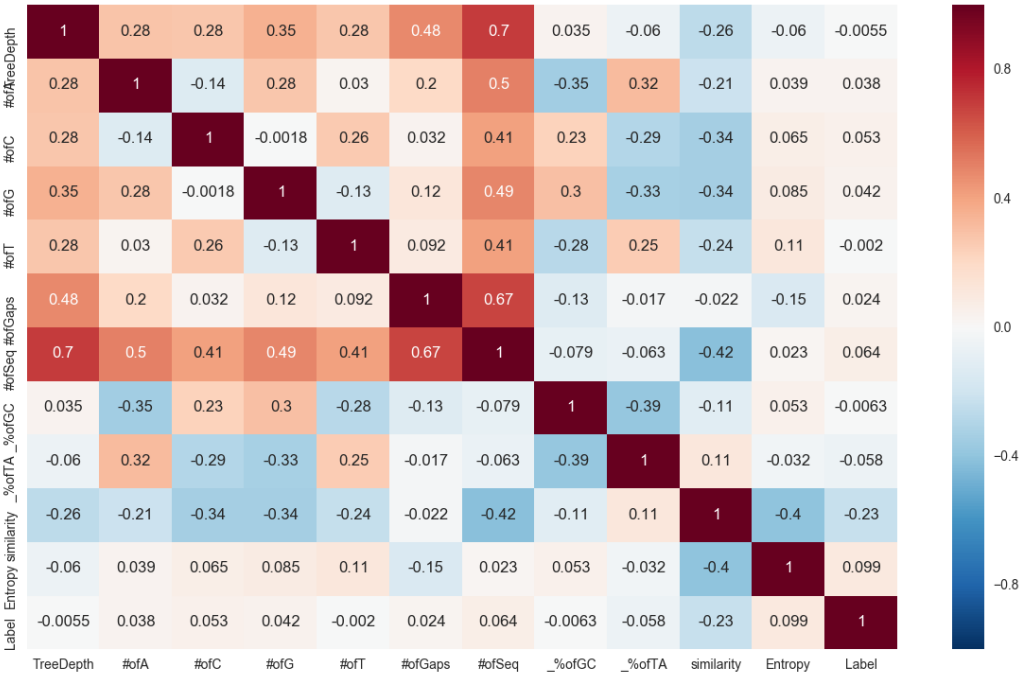

Figure 5–2: Pearson correlation of features selected for difficulty prediction. These correlation coefficients only measure linear correlations.

We trained and benchmarked multiple binary classification algorithms: logistic regression, neural network, extra tree classifier, random forest classifier, Ada boost classifier, gradient boost classifier and decision tree classifier (all implemented in scikit-learn). We partitioned the dataset into 60% training set to learn model parameters and 40% testing set to evaluate the learned models. The hyperparameters were selected using 10-fold cross-validation on the training data. Area Under the Curve of the Receiver Operating Curve (AUC ROC) on testing set was used to measure models accuracy In many cases, it can be useful to go beyond a binary classification problem and instead predict the expected value of the alignment score gain that can be expected for a given alignment. This can, for example, be used to

properly assign puzzles to users with the right level. We experimented with different machine learning regression models to attempt to predict the score gain from the set of 11 features and eventually selected neural networks because of their superior performance.

5.2.3 Results

A key step in Phylo is to assess the difficulty of a puzzle in order to route it to players with the appropriate skill level. Ideally, the difficulty should be estimated automatically, before any player has tried the puzzle. Here, we study how a difficulty level (assessed retrospectively based on the ability of players to improve the alignment score) can be predicted by machine learning algorithms.

Figure 5–3 shows the distribution of the improvement of alignment scores (as calculated in the game) produced by human players over the machine-computed alignments. Although in many cases improvements are modest, the tail of this distribution highlights a significant number of puzzles with very large improvements. It shows that puzzles are of unequal difficulty – a phenomenon we study further in this section.

We labeled the puzzles as easy or hard based on the player's success rate (See Section 5.2.2). Then, we trained different types of machine learning classifiers to predict a puzzle's label based on a variety of features (see Methods). Figure 5–5 shows the classification accuracy. For each predictor, we calculated the Area Under the ROC Curve (AUC), based on 10-fold cross-validation, using a single feature at a time. As anticipated, we observe that the most informative features are those capturing the number of sequences to be aligned, but also their dissimilarity (tree-entropy and

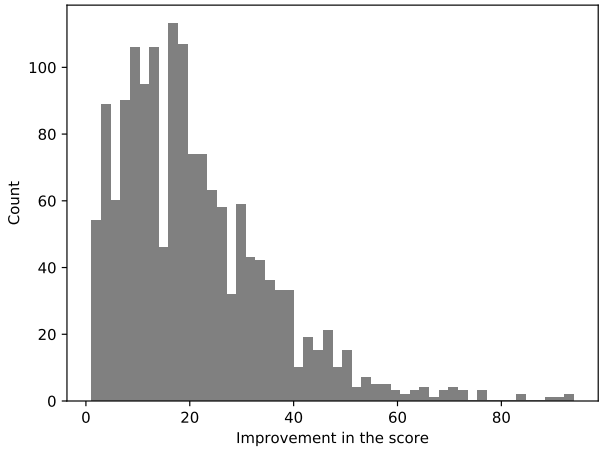

Figure 5–3: Histogram plot of the improvement in the score for 1556 puzzles, where improvement is measured as the difference between the best score produced by a human player and the initial computer-based alignment.

mean entropy of column-wise residues). Then, we trained and evaluated multi-feature models by incrementally adding features in decreasing order of their value (Figure 5–4). Remarkably, it is the inclusion of the feature corresponding to the score of the initial computer-calculated alignment that provides the largest improvement in prediction accuracy. This is not unexpected, as alignments whose initial scores are already high are difficult to improve.

We then considered the regression version of the problem, where the goal is to predict the quantitative improvement in score one can expect for a given puzzle. The best predictions, with R^2 value of 72% were obtained using a neural network regression model whose hyper parameter values with L2-norm regularization regularization (weight of 10^{-3}) and 300 nodes in each of the two hidden layers. This significantly

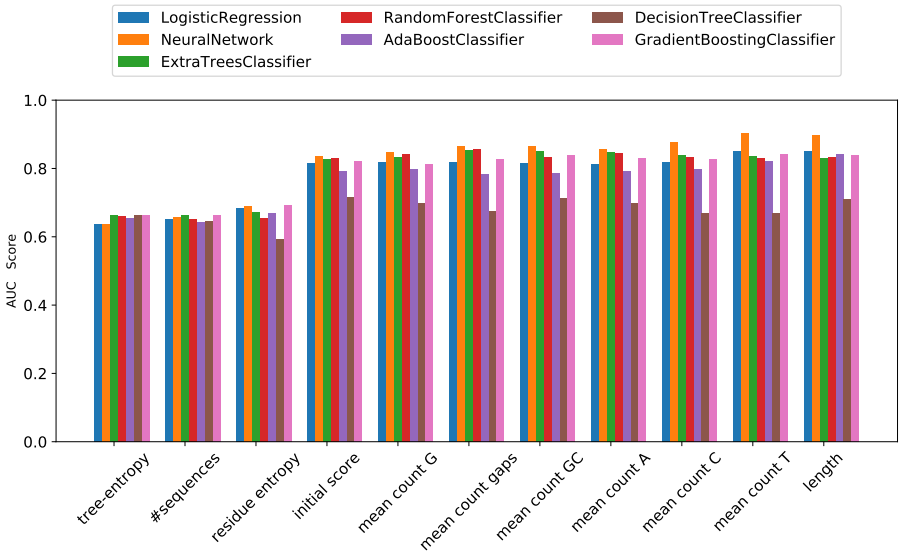

Figure 5–4: Measuring effects of feature combination on models performances. The features from the ordered list (based on individual feature importance in decreasing order) are added incrementally to train models.

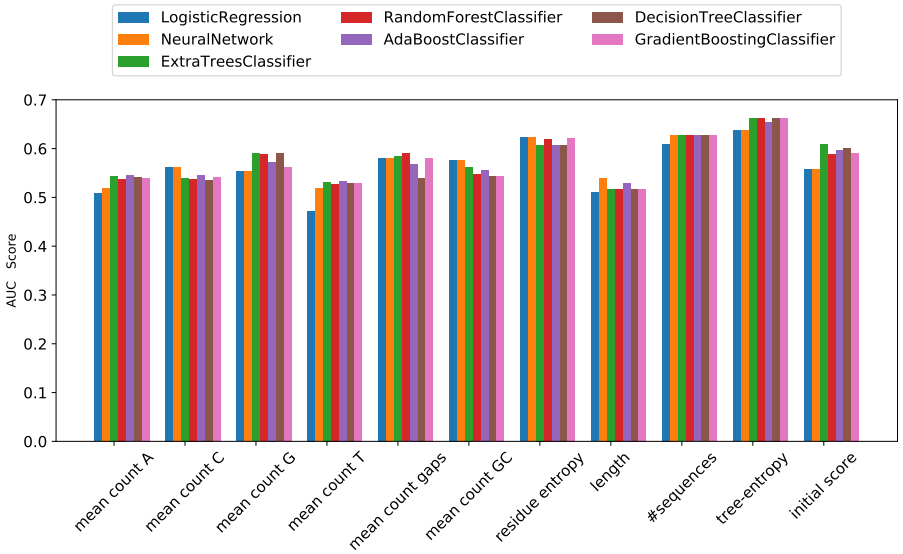

Figure 5–5: Individual feature performance is measured through AUC values on testing set. Each model is trained using one feature at a time.

outperformed other predictors such as a ridge regression model with L2-norm regularization, which obtained an R^2 value of only 56%. This is because none of the predictor variables used in the regression model are highly correlated.

The last and the most important step is to analyze the efficiency of the proposed model with respect to the current scheme that uniformly distributes the puzzles based on the number of solutions collected. The preliminary step for this comparison is to obtain the number of times a puzzle in each category should be sent to get good alignments. To obtain an accurate estimate of this, we identified the top 25% of the puzzles in each category which took maximum attempts to reach to its highest score and calculated their average. Top 25% were taken to ensure significant number of spare attempts for achieving good alignments. We obtained these values (n_0) as 150, 170 and 192 for easy, medium and difficult puzzles respectively.

Next, we calculate the efficiency as $\frac{(N_u - N_w)}{N_{opt}}$, where N_u is the total number of puzzles sent in the uniform model (all puzzles have the same weight), N_w is the total number of puzzles sent using labels produced by our classifier, and N_{opt} is the minimum number of puzzles that has to be sent to obtain all high scores. N_w is calculated as the product of the number of puzzles in a particular category, and the expected number of solutions we need to collect to obtain the highest score (n_0). Table 5–1 shows the gain in efficiency obtained for each category of puzzles (i.e. easy, medium, or hard). The data sets used is already described in the begining of this chapter.

In the research and build up process towards an efficient OpenPhylo, we experimented with several methodologies which did not make it to our final product. The results

Table 5–1: Estimate of the total number of solutions that needs to be collected to obtain the highest score in the uniform (N_u), weighted (N_w) and optimal (N_{opt}) routing schemes. Puzzles are sorted in 3 categories (easy, medium, and difficult) representing the observed difficulty for players to achieve the highest score.

Difficulty	**Number of Puzzles**	N_u	N_w	N_{opt}	**Gain of Efficiency**
Easy	14	4123	2100	1752	115.4%
Medium	80	19653	13600	12501	48.4%
Difficult	466	121531	89472	63205	50.7%

and insights gained from these experiments were highly influential towards guiding us to the chosen solution. We do not go over detail about these other experiments. However, we have provided a brief discussion about them in the Appendix A.

5.3 Approach 2: Difficulty prediction using convolutional neural networks

5.3.1 Label creation

Out of a total of 1907 puzzles, there were 1622 puzzles which were played more than 20 times. We used these input puzzles for difficulty predictions. Label creation for such puzzles is based on *play_count* and *fail_count*. Labels were created for all the input puzzles using the formula shown below:

$$success\_rate = \frac{played\_count - fail\_count}{played\_count}$$

All the puzzles were placed in one of these classes: easy, medium or hard. A success rate less than 80% represents *difficult* puzzles, whereas if the success rate is between 80% and 90% it is considered as *medium* difficulty puzzles and finally if the success rate is greater than 90% then such puzzles were considered as *easy* puzzles. Since,

most machine learning algorithm perform better with numbers, so we converted these text labels to numbers: [0,1,2] for [easy, medium, hard] respectively.

Table 5–2: Data partition for type one difficulty prediction

Difficulty level	**Success rate**	**# of puzzles**
Difficult puzzles	<= 80%	435
Medium puzzles	>80% && <= 90%	389
Easy puzzles	>90 %	798

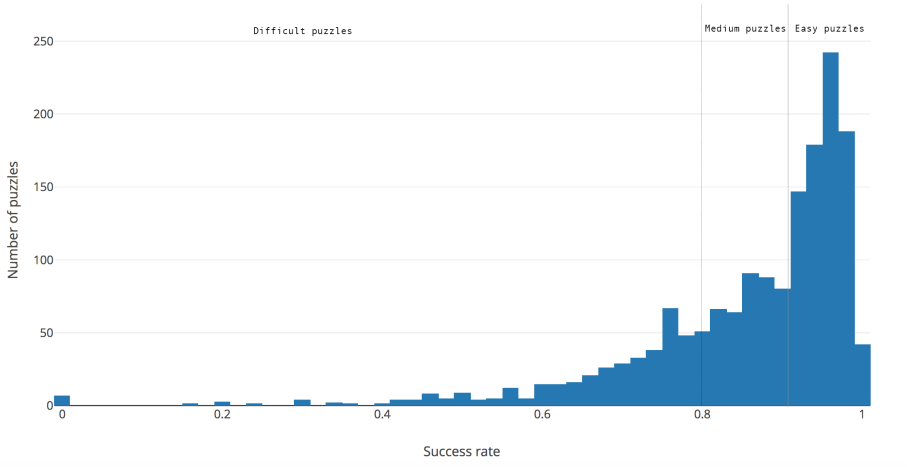

Figure 5–6: Lebeling of puzzles based on its success rate. X-axis represents the success rate and y-axis represents the number of puzzles for with respect to success rate.

5.3.2 Methodology

We restrict ourselves to Phylo game, well know game in citizen science. The model that we used is shown in Figure 5–7. We have used the same CNN model, only

the dimensions have been affected because we are using the input puzzles only for which the dimentions have been restricted to 10 rows, 20 columns and 3 channels (RGB). We did not rescale the image, since the input dimensions are already very small, rescaling them or adding unnecessary gaps at the end will either increase the computational cost or will not provide the best possible results. For now our main objective is to produce a model with best possible accuracy.

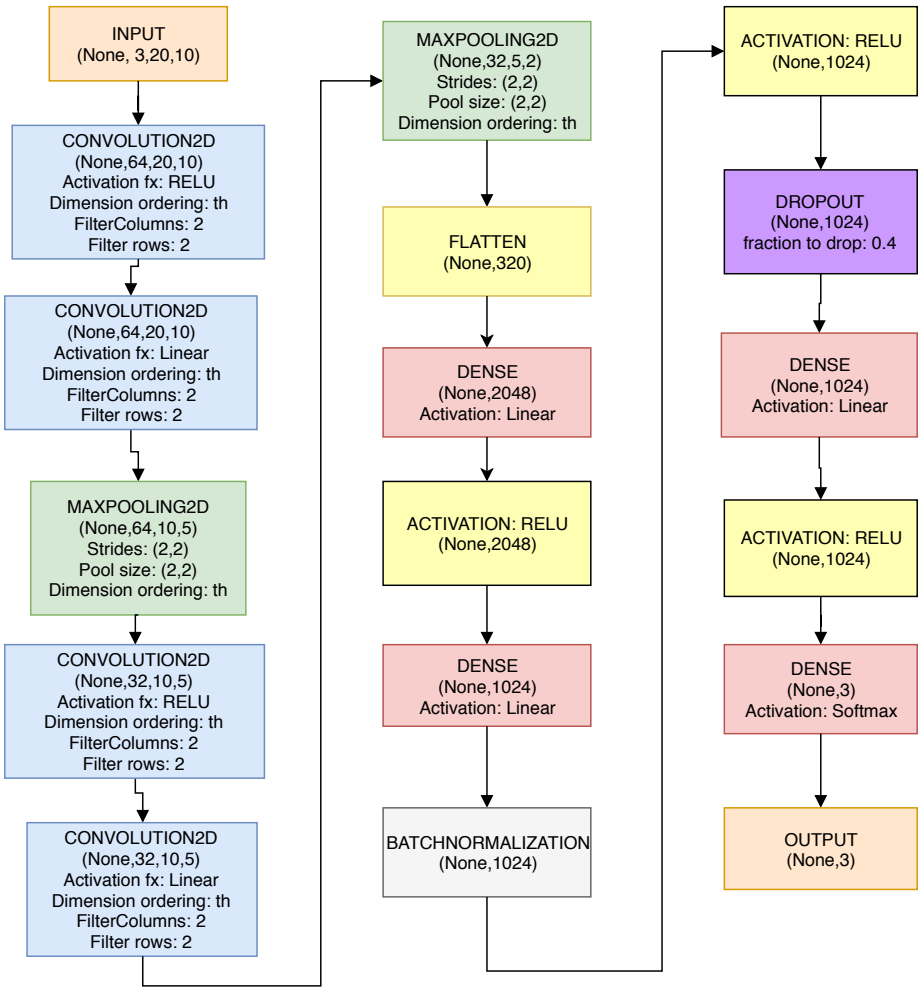

Figure 5–7: Proposed CNN model for difficulty prediction.

Hyperparameter values of the model have been selected using grid search approach and is very different from those that we are using in the proposed promising puzzle algorithm in chapter 3. This is because of the less number of input puzzles compared to all the unique output puzzles used in the promising puzzle algorithm. We received the best accuracy using the hyperparameter values listed in table5–3.

Table 5–3: Hyperparameter values for the CNN based difficulty prediction model.

Hyperparameters	**Value**
# Epoch	50
Batch size	512
Loss function	Categorical-crossentropy
Optimizer	Adam
Learning rate	0.001
Epsilon	1e-08
Decay	0.5
Rho	0.95
Regularization	None
Initial weights	0.1

We choose to use a convolutional neural network since our game board has a grid structure and CNN models has had great performance on other grid based games such as Go [65] or Candycrush [22]. We tested different number of convolutional layers and found that more than two convolution layers do not improve validation accuracy instead increases overfitting. Therefore, we used only two convolution layers. We decided to use (1,1) as the subsample size to avoid any loss of information.

Evaluation criteria

We split the dataset into training and validation sections using *stratified sampling*. Stratified 10-fold cross validation was used since the input data contains more class

Table 5–4: Dataset split for difficulty prediction

Type of dataset	**# of data points**
Training	1459
Validation	162

3 puzzles as the dataset is slightly skewed towards the easy puzzles. Therefore, we wanted to produce folds that contain a representative ratio of each class. StratifiedKfold is available to use in the sklearn library in python [55]. At each iteration, the code creates a clone of the classifier, trains that clone on the training folds, and makes predictions on the validation fold. We evaluated the proposed CNN model using the validation accuracy achieved via 10-fold cross validation.

5.3.3 Results

Figure 5–8 represents training accuracy for each fold in 10-fold cross validation. Multiple lines in Figure 5–10 shows the average performance of the CNN model during validation. The overall accuracy achieved using this model is 75.15%.

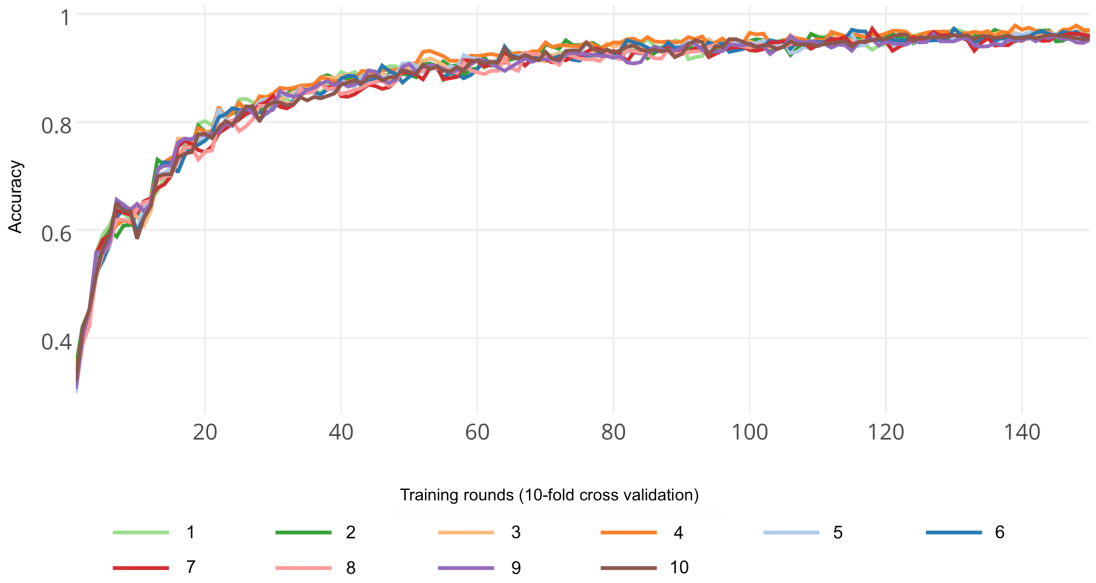

Figure 5–8: Average learning accuracy in difficulty prediction. The 10 different colored lines shows ten unique learning curve generated using 10-fold cross validation. The learning accuracy achieved by the CNN model is 96%.

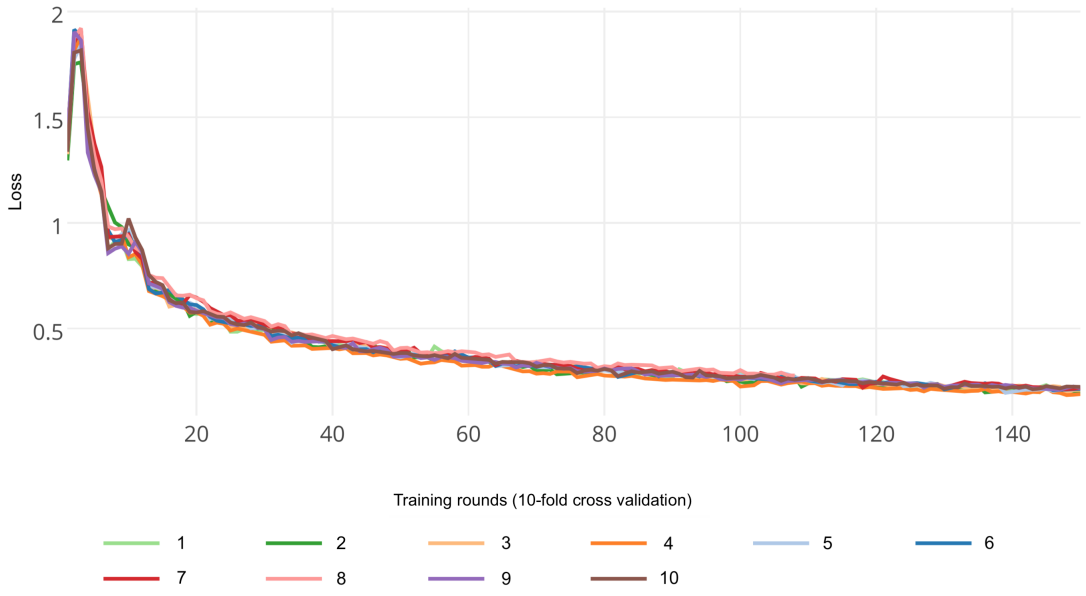

Figure 5–9: Average learning loss in difficulty prediction. The ten lines generated represents each iteration of 10-fold cross validation.

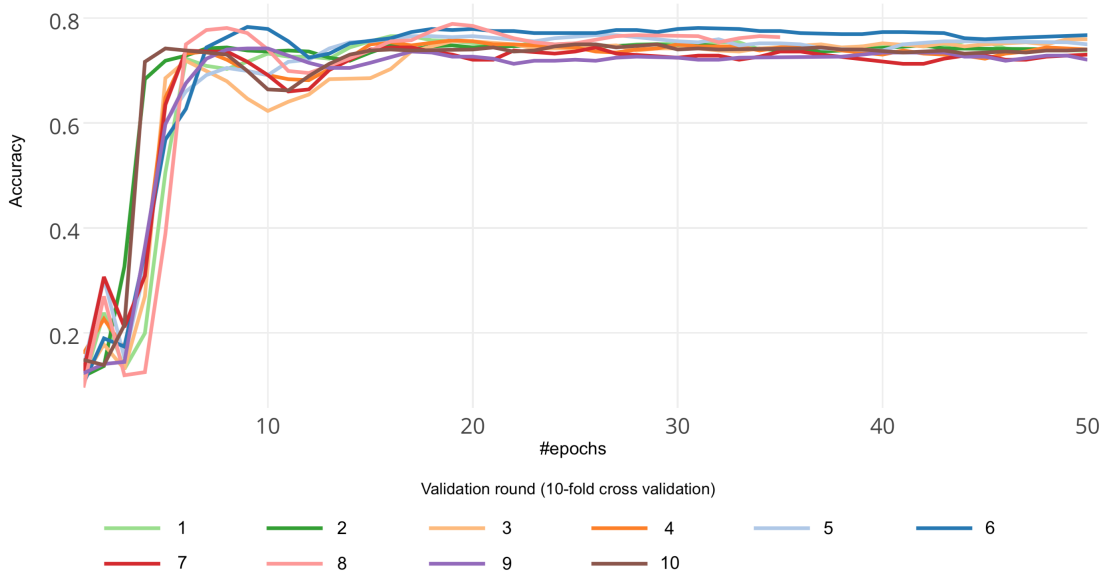

Figure 5–10: Average validation accuracy in difficulty prediction. 10 different colored lines represents each iteration of 10-fold cross validation. The validation accuracy received 75.15%

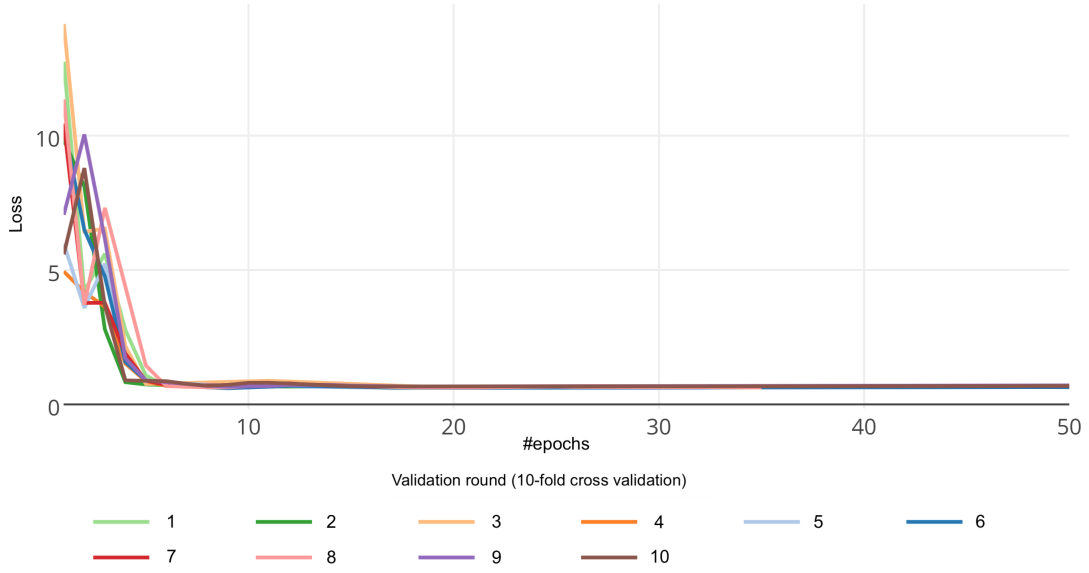

Figure 5–11: Average learning loss in difficulty prediction. The 10 lines generated represents each iteration of 10-fold cross validation.

Our results suggests that the difficulty of puzzles can be relatively well predicted. These predictions can be used to significantly improve the routing of puzzles (i.e. tasks), and thus the number of task to complete to obtain the best answer.

5.3.4 Correlation between promising puzzles and their difficulty estimates

Table 5–5 illustrates the correlation between the promising puzzles and its difficulty level.

Table 5–5: Correlation between promising diffculty and promising puzzles

	Promising Puzzles
Difficult	0.12969377
Medium	-0.00338985
Easy	-0.024694445

The correlation coefficient value between difficult and promising puzzles is 0.129 which is a weak positive correlation. We conclude that, higher difficulty puzzle is more likely to be an interesting one.

The correlation coefficient between medium difficulty level puzzles and the promising ones is -0.003. Although a negative correlation, this value suggest to have no linear relationship or a very weak linear relationship.

Likewise, in case of correlation between easy and promising puzzles we notice a negative correlation, the relationship between the variables is only weak (the nearer the value is to zero, the weaker the relationship).

CHAPTER 6
Dynamic change in difficulty

Chapter 5 deals with defining the initial difficulty level to different puzzles. However, this is not the best possible accuracy and might provide us a fit only for the initial difficulty prediction for each puzzle. To give a more accurate prediction, we need to update the difficulty based on the user success rate for different puzzles. Therefore, in this chapter we propose an online system that can update the difficulty of puzzles, if needed, whilst we receive new solutions from the game. We track number of attempts made and successful submissions to decide the success rate for a puzzle. Based on this success rate, the difficulty level of various puzzles gets validated/updated (iff, the success rate is not in line with the initial difficulty level of the puzzle predicted by the proposed difficulty prediction model). For this to work we also needed the latency (number of submissions required to ensure a quality result) of each puzzle. We have provided a brief discussion and result on this in Appendix D of this report. Based on our observations, we considered 93 as the count of solutions needed before we can decide the success rate of puzzles. A value of 93 was chosen based on our obervation of Phylo dataset [67]. Therefore, after receiving these many solutions for a puzzle, OpenPhylo validates the difficulty predicted by the algorithm proposed in chapter 5.

6.1 Dataset

The dataset used in this chapter is from puzzles generated for the new release of Phylo. The table 11–2 in the appendix C of this report contains data from puzzles of different categories. Phylo is being played actively since its deployment at the Canada Science and Technology Museum, Ottawa, Ontario on November 17, 2017. Recently, it has been a part of the successful endeavors like CitSciDay hero event, DNA day and Science Odyssey. Overall we have gathered over 5000 solutions for 200 puzzles as of May 27, 2018. We have used the top 100 puzzles for this observation to ensure significant results.

6.2 Methodology

We model this probability considering this as a Bernoulli process [46]; a binary outcome - win or lose– characterized by a single parameter p_{win} which represents the probability of winning the puzzle in single attempt. It is represented by the following equation:

$$p_{win} = \frac{\sum win}{\sum attempts}$$

Win illustrates a scenario where a user successfully completes the entire puzzle. **Fail** presents a scenario where a player starts with a puzzle but is unable to align the puzzle completely; he is unable to align all the sequences in the puzzle and therefore we do not receive a complete solution from such an attempt.

Now, let's compute the difficulty p_{win} separately for each of the 5 levels. Calculating the probability of solving a puzzle(p_{win}) of a predefined difficulty level as a Bernoulli process is a simplified version of a multidimensional problem. We know that this probability will also depend on the skill of each player and the player could learn from past attempts and play better every time. Hence, to further refine this process, we have addressed this scenario in the routing section of this report where we consider user expertise during the routing of puzzles.

6.2.1 Computing Uncertainty

Here we have a used a *StandardError* as the measure of uncertainty which is calculated as:

$$\sigma_{error} \approx \frac{\sigma_{sample}}{\sqrt{n}}$$

For a Bernoulli process, the sample standard deviation is:

$$\sigma_{sample} = \sqrt{p_{win}(1 - p_{win})}$$

Therefore, we can calculate the standard error like this:

$$\sigma_{error} \approx \sqrt{\frac{p_{win}(1 - p_{win})}{n}}$$

6.3 Results

From the Figure 6–1, we can conclude that the difficulty estimates are precise as you can see a lower success rate for difficult puzzles. However we see some discrepencies as well. There are certain puzzles in easy category for which the success rate is low and vice versa. For instance, looking at the data enclosed in appendix C the success rate

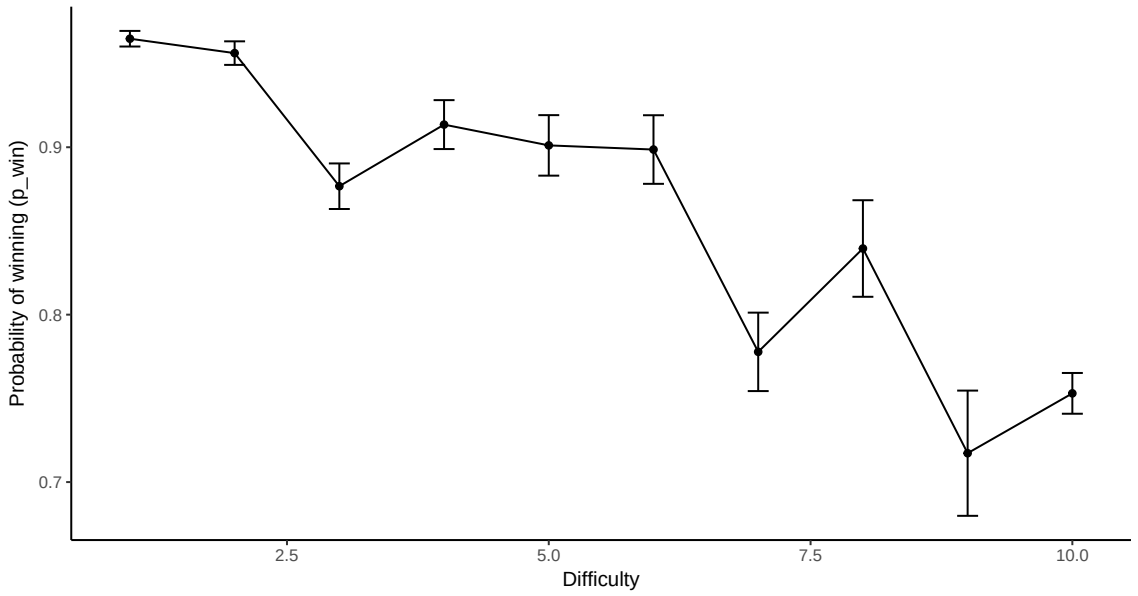

Figure 6–1: Identifying the true difficulty of puzzles based on the player's success rate. Vertical bar represents the overall error rate of all the puzzles corresponding to that difficulty level.

of Infectious disease for difficulty level 4 is lower than 80%, hence its difficulty can be increased and such puzzles can be moved to difficult category. Another example easily visible is that of category - Brain, nervous and sensory system diseases – for which the difficulty level of 10 has a very high success rate. Therefore, this puzzle can be moved to easy category. We also notice high error rates for some puzzles and this is because of less attempts made at these puzzles and these error rates will reduce after such puzzles are played more times.

We conclude that difficulty level of the puzzles can be validated/updated dynamically based on its success rate. We check for validity/update the difficulty of puzzles after 93 attempts are made for a puzzle (refer appendix D) [67]. If a puzzle has a success

rate greater than or equal to 90%. Such puzzles will be moved to easy category, whereas, for puzzles with success rate less than equal to 80%, such puzzles will be moved in difficult category. Puzzles with success rate between 80% and 90% will be placed in medium category. But, this should be done only after we have significant number of solutions which will further reduce the error rate. In the upcoming version of Phylo we choose this number to be 80 based on our observations published in paper [67].

If we consider the top 100 played puzzles in the latest version of Phylo, we notice that the difficulty levels are predicted correctly for 71 puzzles. Coincidently, this is very close to be the accuracy of our feature based difficulty prediction model which is 72%. Hence, we expect that by using this approach we can further enhance the difficulty prediction of puzzles. This will play a major role in producing quality results in the game.

CHAPTER 7
Routing

Routing is the process of assigning the right difficulty level of puzzles to the right players based on their skill level. This ensures better results and a more entertaining experience for the game players. Routing is a multidimensional problem which must address the following parameters: player's expertise level, success rate of puzzle, and number of times each puzzle is played.

In human computation, to account for user expertise, often users are given the same task and then take the majority response to get the correct answer. Luckily, Phylo has a proven scoring scheme which is capable of separating the better solutions against the good ones. This scoring scheme can be further used for collective assessment using expectation maximization. The objective of this task is to jointly estimate player expertise level and consensus answer. Each puzzle in Phylo is considered as a single problem with an unknown solution to the problem. We calculate a parameter $P_w(r|a)$ for each player p with correct response a and the incorrect response as r; where $r!=a$. Phylo ensures that each player submits his/her own solution and these solutions are independent of how other players play the same puzzle. David and Skene in their paper [17] use an iterative expectation-maximization approach to estimate which answers are correct and at the same time find the accuracy of worker expertise level. We here perform a similar operation with a minor difference that we have used

the relative score of solutions for each puzzle as a measure of solution accuracy. Any score that lies above the 50^{th} percentile with respect to its score is considered as a good solution whereas a percentile of below 50 is considered as a failed attempt. Expectation maximization then scores each player based on how many good solutions were submitted by that player. The process repeats itself every time a puzzle is sent to a player.

7.1 Routing algorithm

Goal 1: *Determine the user/ player expertise:*

Given $\{r\}$ responses from all the players for a puzzle;

while *true* **do**

  if *new submission* **then**

    Step 1: Percentile for each submission is calculated based on all the available submissions also known as *relative score*;

    Step 2: Adjust $P_w(r|a)$ by taking the average of all the relative scores;

  else

    Return the P_w value to goal 2 of this routing algorithm;

  end

end

Algorithm 1: User expertise prediction algorithm

Note: We initialize the value of $P_w(r|a)$ to 0.5 (for new players). In the original paper [17] this was set to a random value. However, we start with a value of 0.5 which ensures that the user is neither a rookie nor an expert player.

Goal 2: *Routing of puzzles based on user expertise (calculated in goal 1), difficulty of puzzles and consensus achieved in puzzles.*

Assumptions:

Puzzles with difficulty level between 1 to 5 are considered as easy puzzles whereas those with the difficulty between 6 to 10 are considered as difficult;

Initialization:

1. failure rate of a puzzle is initially set to 1 if the puzzle has been played less than 20 times;

Algorithm:

Step 1: based on the user expertise select the either all the easy puzzles or all the difficult puzzles.

Step 2: Retrieve the mean failure rate of all these puzzles from the database. Failure rate for a puzzle is calculated as:

$$failureRate_p = 1 - \frac{\sum solutionPercentileValues_p}{NumberOfSolutions}$$

Step 3: calculate the upper confidence bound for all the selected puzzles in step1 above.

$$X_{UCB-p} = \overline{X}_p + \sqrt{2\frac{lnN}{N_p}}$$

$$\overline{X}_p \, is \, the \, mean \, failure \, rate \, for \, puzzle \, 'p'$$

$$N_p \, is \, the \, total \, number \, of \, solutions \, for \, the \, puzzle \, 'p'$$

$$N \, is \, the \, total \, number \, of \, solutions \, for \, all \, the \, selected \, puzzles \, in \, step \, 1$$

If N_p is zero then initialize it to a small value of 0.1.

Step 4: Select the one with the highest upper confidence bound value. In case of a tie, we select one at random.

Step 5: Send the selected puzzle in step 4 to the user.

We start by using the user skill level (derived in Goal 1) to chose the kind of puzzles to be routed to the players. Upper Confidence Bound (UCB) is based on the principle of optimism in the face of uncertainty, which is to choose your puzzles as if all the puzzles are equally nice. However, uncertainty on overdose is not recommended; when played decent number of times we don't need to be optimistic, instead can rely on its true values. When acting optimistically, either the selected puzzle justifies it or, the optimism shown towards its selection is not justified. In the latter case a puzzle is picked based on the belief of getting a large reward when in fact it does not. If this happens sufficiently often, then the learner will learn what is the true payoff of this action and not choose that puzzle in the future. Intuitively, we know that failure rate of a puzzle calculated from a mean of 10 iterations is less accurate than failure rate calculated for that same puzzle using 1000 iterations. Therefore, if we were to draw a confidence bound around each mean, we will get a much wider bound on the 10 puzzle case and a much skinnier bound around the 1000 puzzle mean. Using this approach we are optimistic for a puzzle in the beginning when the bound is wider, we become more certain of that puzzle's failure rate. Failure rate is defined as the ratio between the number of times players achieve less than 50th percentile to the

total number of times that puzzle is played. Failure rate for a puzzle is constant 1 if the number of times that puzzle played is < 20.

This is like a greedy strategy. Key here is the ratio between two parameters: N which is the total number of times you have played, and N_p which is the total number of solutions for puzzle p. Therefore, if a puzzle is played less number of times (which means N_p is small) whereas overall other puzzles have been played many times, then this ratio will be large. This will make the upper confidence bound higher. A higher confidence bound for a player will mean an better UCB value for that puzzle. Therefore, the algorithm will pick that puzzle and route it tothe players. At the same time if N_p is larger on an average when compared to other puzzles then the ratio will be smaller than the other selected puzzles. Then the upper confidence bound will shrink for such puzzle, hence such a puzzle will not be selected over others. Therefore, chosing UCB in our routing approach will ensure a fair chance to all the input puzzles.

According to Chernoff-Hoeffding [6] bound, the confidence bound changes exponentially with the number of samples we collect.

$$P\left\{|\overline{X} - \mu| \geqslant \xi\right\} \leqslant 2exp\left\{-2\xi^2 N\right\}$$

The equation says that the absolute difference between the sample mean and the true mean is greater than or equal to epsilon(ξ) is less than two times the exponent of -2 times epsilon squared N, where N is the number of puzzles played and ξ is a

random small number. Here, we take the upper confidence bound using,

$$X_{UCB-p} = \overline{X}_p + \sqrt{2\frac{lnN}{N_p}}$$

where, N is the total number of solution for based on the type of puzzle selected for the user: easy, medium, and difficult, and N_p is the number of solutions for the puzzle 'p'. This is equivalent to, choosing an ξ equal to the square root and everything inside it. [Note: The proposed algorithm has been recently simulated for Phylo, therefore we do not have results for this.]

In layman's terms, we are using the following steps to emulate routing in Phylo:

1. Calculate the expertise level of registered players of Phylo as explained in goal 1 of the routing algorithm.
2. If user expertise is less than 0.5, we select all the puzzles with difficulty level less than 5, otherwise we select puzzles with difficulty between 6 to 10.
3. Calculate the failure rate of all the selected puzzles as mentioned in step 2 of algorithm 2.
4. Now use UCB to route one puzzle to the player ensuring the following constraints:
    - If the selected puzzle has never been played then initialize it with a small value
    - in case of a tie we pick one at random from the selected set of puzzles in point 2 above.

CHAPTER 8
Teaching Portal

Crowdsourcing that is gathering information about best practices from a variety of people is an essential way to enhance the way education is conducted by instructors and received by pupils. One of the most fascinating aspects of modern educational technology involves the ability to both teach and crowdsource information at the same time. As an illustration, the Spectral Game Platform was created using open source spectral data, and challenges students to match the spectrum data with the relevant chemical structure. Not only does this allow students to learn the material and receive real-time correction, but it also provides a crowdsourced check to the open source data. Both the students and the scientific community win with crowdsourcing. For the purpose of this project, we will first review the existing Phylo Educational Platform and comment on what modifications would need to be done. A new model would then be proposed with the aim to improve the existing platform and introduce the crowdsourcing techniques. The focus will be to design it in such a way that it improves the usefulness of Phylo in classrooms and introduces important educational concepts during the gameplay. Link to the teaching portal: `https://kovik.cs.mcgill.ca/#/loginInstructor`

8.1 Classical educational platform

The classical educational platform of Phylo is simple in design and operation. It has been divided into the following categories:

- **Home:** It begins by introducing Phylo and also contains links for instructor and student log-in.
- **Manual:** Step-by-step instructions for using Phylo as a teaching portal. It also contains instructions for instructors and students.
- **Students:** Student log-in link given. Once logged in, students can register for an event (exam) which will be active for some duration. During this time, the games played by the student is recorded and posted to the instructor as a report once the event ends.
- **Instructors:** Instructors can log-in giving their username and password. After getting logged-in, they can file their exam, specifying a unique event id and can download the report once the event ends.
- **Resources:** This page gives resources which students/teachers can benefit if they wish to know more about the concepts.
- **List of Puzzles:** Returns a list of puzzles available for the students. Classified as per diseases and displays the initial score, best score and submission count of each of the puzzle.
- **Contact:** Contact information provided.

We will now analyze the results from the classical teaching portal using the datasets generated over the past 5 years. This information allows us to study the impact of a prior knowledge of the multiple sequence alignment problem on the quality of solutions submitted. It also tells us the pros of the existing teaching portal and possible scope of improvements. We analyze the performance of casual gamers and educational players; **casual gamers**, who play just for fun, and **educational players**, who play in the context of a high-school or university biology/bioinformatics course through our educational interface. Since its launch in 2010, Phylo has now 35,913 registered users (Note: participants can also play anonymously without registration - observed ratio is 1:10), for which we recorded the number and difficulty of puzzles played. We analysed a total of 170 puzzles which were played by both types of users, which we used to assess the performance of each group. We observe a P-value of 0.47 while performing T-test on the dataset of educational players and casual players. It shows that the benefit of background knowledge toward better alignment production is difficult to assess. In Figure 8–1, we show the performance achieved by educational players playing for education against casual on puzzles with various difficulties. Although educational users seem to benefit from their background knowledge, the difference disappears when the difficulty increases. Based on the existing educational interface we conclude that a prior introduction to the principles of the game (i.e. here a multiple sequence alignment) does not seem to provide a long-term advantage. This confirms the importance of design techniques in GWAPs.

However, we do notice that a lot is expected of the students to complete the assignment. This induces the students to make mistakes not regarding academics but in

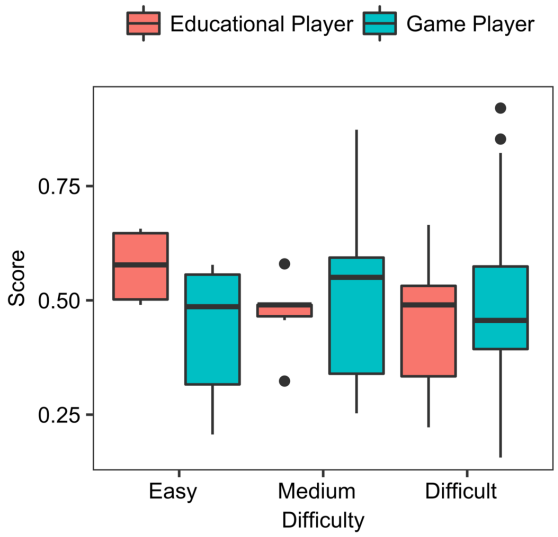

Figure 8–1: Average performance of casual (no prior training) and educational users (benefiting of background knowledge) on easy, medium, and hard puzzles. The black dots above the box represent outliers more than 3/2 times of the upper quartile while the ones below represent outliers less than 3/2 times of the lower quartile.

misreading the procedure to complete the assignment. Hence, the aim of making the assignments fun for the students is lost and it results in chaos and complexity. The instructors should be allowed to create the assignment accessing the teaching portal while the students should play the game without worrying about anything else. All they need is to do is play the game within the assignment completion deadline as set by the instructor. Also, there is no system in place which checks students progress and notifies them once they complete the assignment.

8.2 Upcoming Teaching Portal

Teaching portal will now be used only by the instructors. Students are expected to just enjoy the game. They will be notified with an assignment completion badge once they finish the assigned task. This is to make the students focus more on the knowledge and game aspect of Phylo rather than reading directions to complete the homework. We consider three actors that covers all the use cases of teaching portal:

- Instructors
- Students
- Admin

There are four primary components in the teaching portal of OpenPhylo for *instructors*:

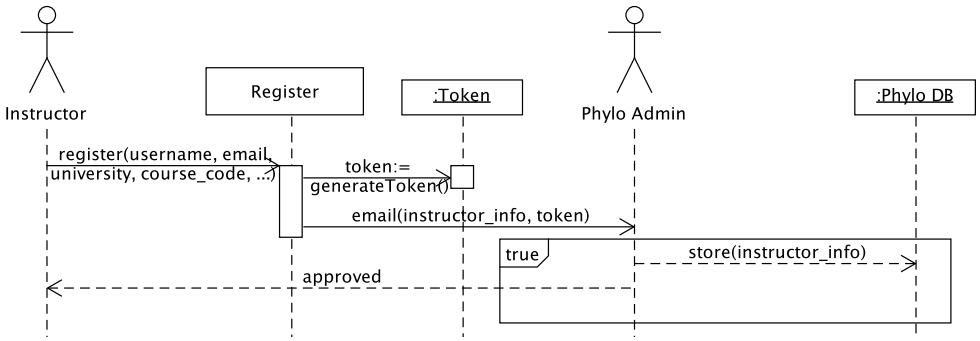

Figure 8–2: Sequential flow of instructor registration.

8.2.1 Instructor

Instructor registration/login module

The homepage of the Phylo Educational Interface will describe the principles of Phylo and the motivations for the interface. There will be a link which instructors can use to login or register into the system. Instructors will have to provide a username, password, course name and its description. Every registration request is associated with a token with a lifeline of one day/24hours. Instructor's information is further sent to the Phylo admin for approval along with the token. If the request is approved within a day then the registered data stays in Phylo database. If the request is not approved in 24 hours token loses its validity and instructors details are removed from the database. This is to ensure that only the real instructors register for the educational portal. Instructors can login to the teaching portal using their username and password.

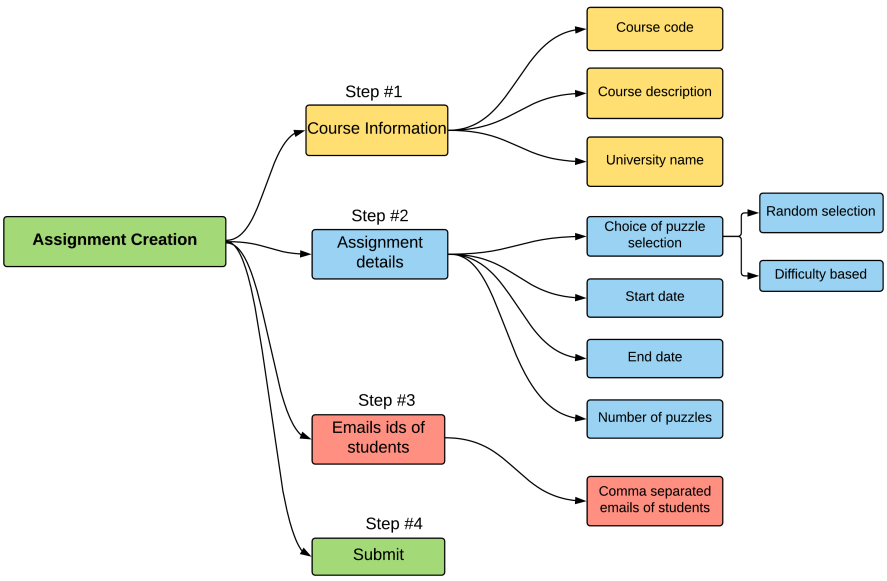

Figure 8–3: Abstract flow of assignment registration.

About

Gives an introduction to Teaching portal of OpenPhylo and roles and responsibilities for Instructor.

Assignment creation

Only approved instructors may access this component. It is simplified to an extent that instructors just have to follow the sequence of pipelined steps to finish the assignment creation. The interface for the assignment creation is very intuitive and does not require a manual. Figure 8–3 shows an abstract flow of assignment creation. After the assignment creation is complete, all the students included in the assignment receive an email with the details of the assignment and its instructor.

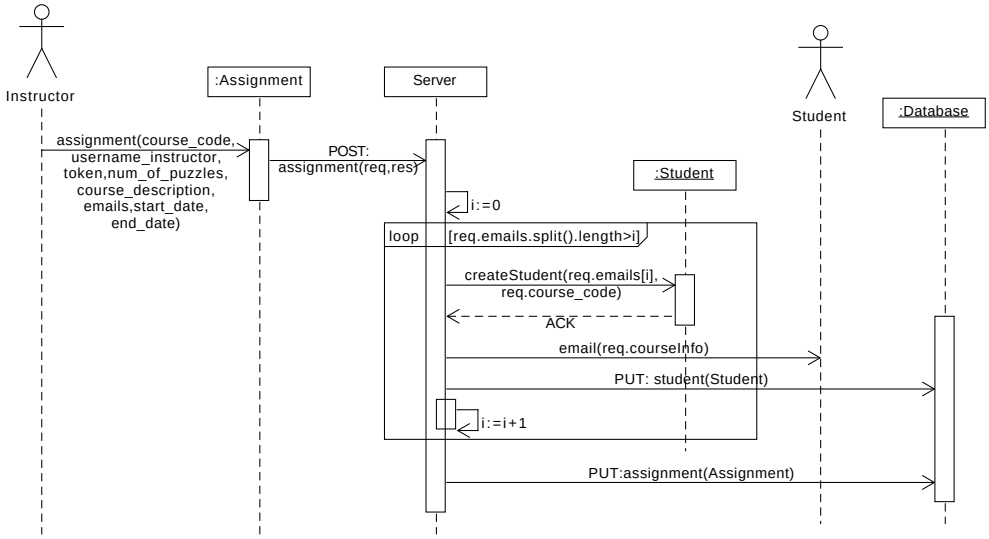

Figure 8–4: Sequence flow of assignment creation.

Viewing/Editing assignment

When an instructor selects the assignment option from the sidebar, two flashcards are displayed. One of them represent the new assignment creation option while the other represents viewing and editing of assignment. Requirements of each of these tasks are already listed in the cards. The assignment once submitted can be viewed or edited by the same instructor before the set deadline. This allows any late additions or entries to the assignment. Following options are editable in the already existing assignment:

- Start date of the assignment
- End date of the assignment
- Course description

- Add more students to the assignment

Any modifications in the assignment are notified to the students. However, if the only chnage is in terms of addition of students then only the new students will be notified.

Result generation

Results is made available based on the course code. It can also be downloaded in the form of a csv file. Each course code will show the following statistics:

- Each students email address,
- each students' assignment status,
- number of highest score achieved by each student,
- number of puzzles played by each student,
- total score achieved,
- percentage, and
- rank of each student.

8.2.2 Students

Assignment completion

Students do not have to access the teaching portal, they can directly register on Phylo game (if they haven't already) and start playing. Each student will receive the same set of puzzles based and also in the same order. This is to avoid any

Figure 8–5: Badge received after completing the assignment.

possible bias while grading the students. They will first receive the puzzles staged in the assignment for which they are enrolled. Once they have completed the homework puzzles, they can see the assignment badge being enabled in the badges section of Phylo. The badge received in Phylo is shown in Figure 8–5.

Quiz for students

Each student during his enlistment by the instructor in the assignment will receive an email with all the course and assignment information. In addition, this email will also contain a link to the quiz. Students can check their learning of bioinformatics subjects are in context with Phylo by writing a quiz. It is available for the following categories: (i) multiple-choice questions, (ii) single answer questions, and (iii) free text-based questions. Every question carries one point each, however, the free text based questions are not scored. It has multiple categories: (i) DNA, (ii) RNA, (iii) Motifs, (iv) Multiple Sequence Alignments (MSA), (v) Mutations, and (vi) Human-Computation. The players can choose one of these categories to start the quiz. The quiz interface provides you with the option of setting the time-limit, number of

questions, and also the category of quiz. Users can go through the quiz using the right and left arrow keys. Upon the quiz completion, the players receive a feedback in terms of percentage of accuracy. The user can play the same category of questions as often as they want. The questions sent to the users are randomized, so that they don't receive the same set of questions in the same order. This was done to enhance the user-experience.

CHAPTER 9
User feedback

The need of feedback in Human-Computer Interaction (HCI) or citizen science is known to everyone. However, the nature, quality and effect of feedback has no consensus. It is the best way to communicate with the end users. All strategies require feedback to observe and adjust performance. Feedback usually compares current performance with fixed goals and returns back information specifying the divergence between current and expected performance. This is true for humans as well. It influences the future behavior of the end-user and keeps them motivated. Considering the importance of feedback, we made some major enhancement in the new version of Phylo and OpenPhylo.

In addition, the efficiency of human-computing systems depends heavily on the expertise of participants. We know this from the results that we received from the classical version of the game. Figure 9–1 reflects the same.

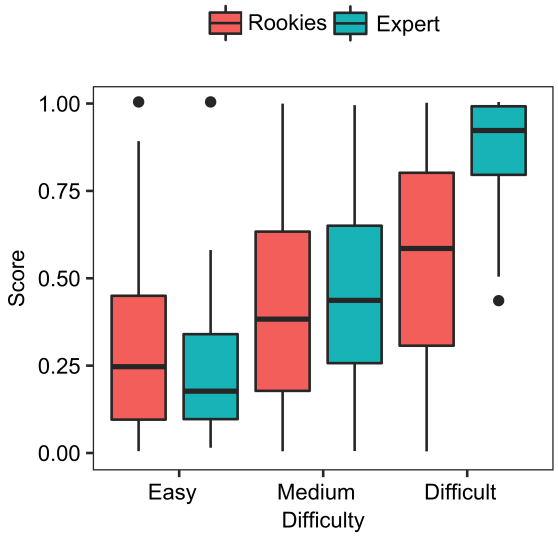

Figure 9–1: Average performance of rookies ($\leq$ 20 puzzles) and expert players ($\geq$ 40 puzzles) on easy, medium, and hard puzzles. The black dots above the box represent outliers with more than 3/2 times of the upper quartile. The black dots below represent outliers with less than 3/2 times of the lower quartile.

Characterizing the precision of answers from their level of expertise and background knowledge is thus essential to understand the capacity and behavior of the system. One best way to transform a rookie into an expert player is by proving a proper feedback that motivates the users .

A feedback should encompass the following characterstics [58]:

1. Feedback about both past as well as the immediate interaction.

2. Development of feddback that affects the future behaviour of the user.

3. Extending the feedback to provide the understanding of the computer application.

4. Use of graphic rather than textual feedback.

9.1 Feedback to users

Phylo covers all these characterstics which is covererd via the following feedbacks to the users:

1. **Achievement Badges:** A unique badge for each of the following achievements:
   (a) *Citizen Science:* A badge for creating an official account on Phylo
   (b) *Master:* Complete a puzzle with three stars
   (c) *Scientist:* Complete one puzzle per disease category
   (d) *Master:* Completed 25 puzzles
   (e) *Weekly Contributor:* Reach the top 10 of the weekly ranking
   (f) *Monthly Contributor:* Reach the top 10 of the monthly ranking
2. **Event based badges:** in additon to the above badges, Phylo also provides event based badges to help support the community:
   (a) *DNA day/ 15 for 15:* Complete 15 puzzles during the DNA day
   (b) *Assignment completion:* This badge is for students who finsih their assignment created by the instructor via the teaching portal.

(c) *Science Odyssey:* Played a puzzle duing the science odyssey celebrations.

(d) *Museum:* Played a puzzle in the Canada Science and Technology Museum, Ottawa.

3. **Live messages:** Immediate informative messages showing user's contribution to science.

4. **Line graph:** representing the comparitive analysis of their scores with respect to the other players.

5. **Doughnut graph:** that tell the users the number of 1, 2 and 3 stars received.

6. **Pie chart:** that keeps the players informed about the number of puzzles palyed in each category.

7. **Quiz:** to facilitate user assess their knowledge in bioinformatics for subjects close to Phylo.

8. **Statistics table:** that displays all the solutions submitted by the logged in user besides the scientific knowledge conforming to each puzzle, like the disease linked to that puzzle, the gene name from which the puzzle was obtained, name of the scientist who submitted that puzzle, score and also the percentile score of that solution, etc.

9. **Ranking:** weekly, monthly and overall ranking of all the users. Phylo comes with a leaderboard where users can rank against one another and compete for their rank on that list. In addition, it also provides weekly, monthly or yearly ranking.

10. **Tutorial:** to walk the user through the instruction of the game.

Figure 9–2 provides a snapshot of feedback provided to users in Phylo.

9.2 Feedback to scientist

Scientists receive all the feedback that a casual Phylo player receives. In addition, scientist can also view their puzzle submissions in Phylo and number of times those puzzles were played in Phylo. They will get the overall aggregated results and also a comparitive analysis of the T-Coffee score associated with the machine aligned sequences and enhanced sequences post replacing the user aligned sequences in the original MSA via aggregation. Aggregation is further explained in chapter 10.

9.3 Conclusion

An application or a game that can leverage volunteer human activity can escalate dramatically better because of the non-conservative nature of the reward. We believe that generally-relevant designs for data-oriented games could radically change the essence of our problem. Phylo provides gratification and reward as well as stoking competition among volunteers. Metrics like percentile score of tasks are readily visible and provide the judges a sense of reward in completion of the task. The most concrete way we reward our players is by giving them a contribution message that expresses their contribution in the best possible way. We attribute every user contribution in the database to the scientist responsible for producing that puzzle.

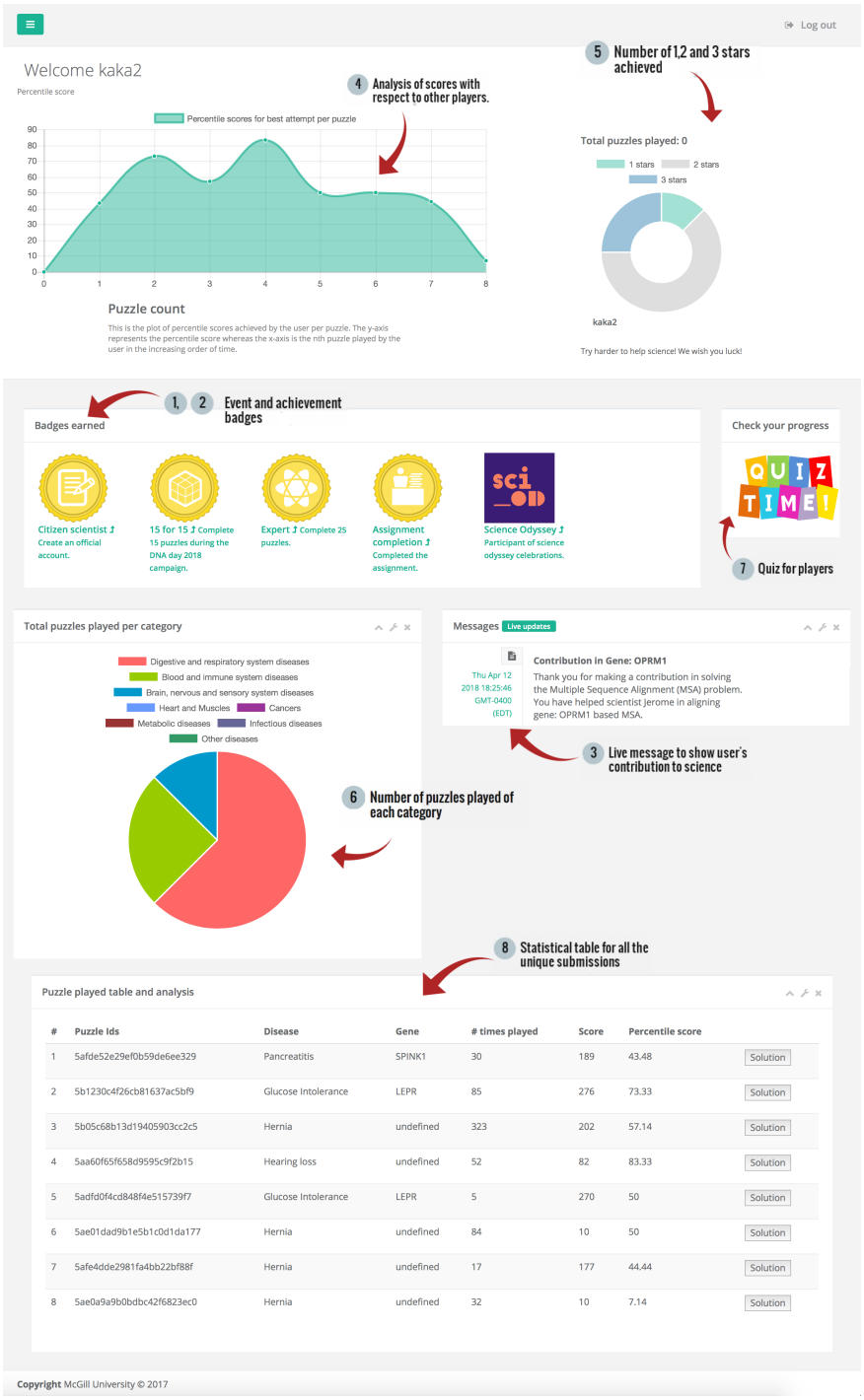

Figure 9–2: User profile page. Annotations are in the same order as mentioned in section 10.1.

CHAPTER 10
Aggregation

Due to the restraints of Phylo, i.e., its capacity of processing and aligning up to 10 sequences, meaning that large scale multiple sequence alignment problems cannot be directly aligned by Phylo. This brings us to the motivation of aggregation. Aggregation is the process of using the quality solutions received via the game and merging it back in the original larger genomic sequences in a way that it enhances the overall accuracy of the solutions received via state-of-art algorithms. It aims to help revealing conserved patterns across different species that may have a functional role. In addition, it also does the pipeline filtering and assembling the solutions from users, and quantify the magnitude of improvement. Figure 10–1 provides a very abstract overview of aggregation process followed in Phylo.

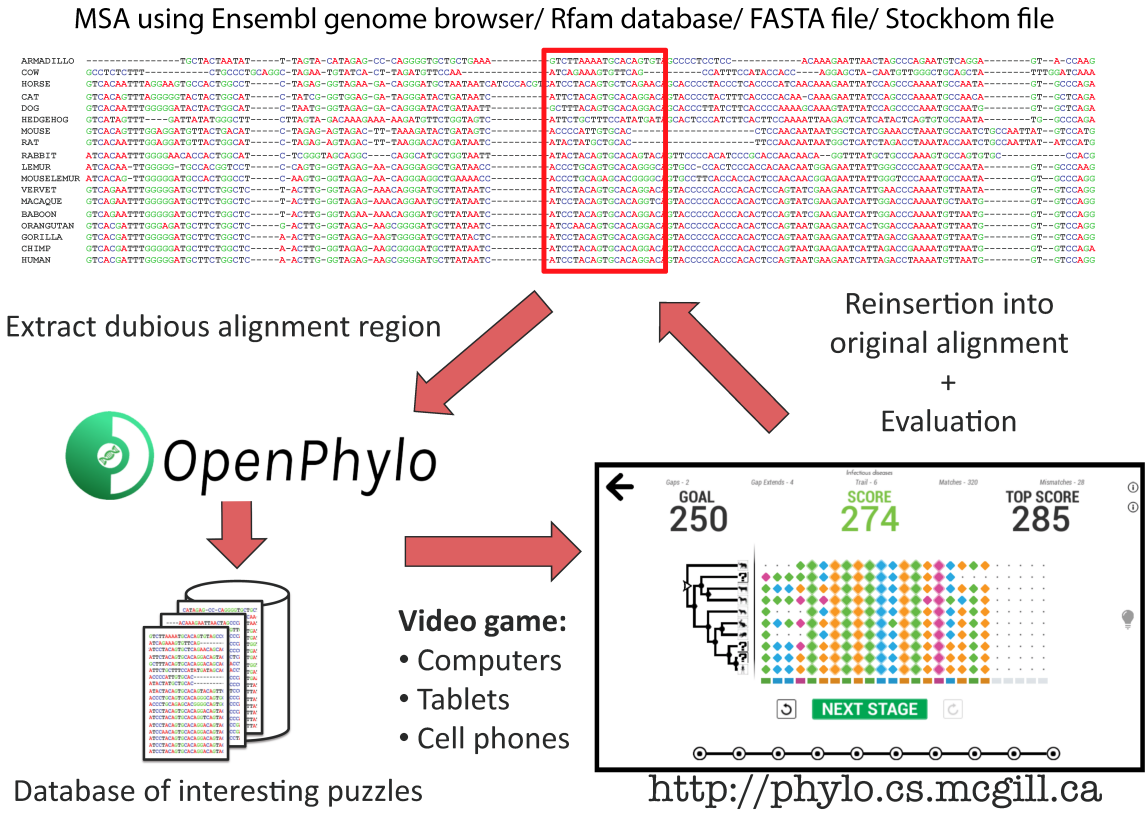

Figure 10–1: Phylo crowd-sourcing system for local improvement of multiple genome alignments.

10.1 Classical approach of aggregation

The solutions from classical Phylo was evaluated by selecting those alignment solutions from puzzles with the best score and least entropy. Then the selected alignments were reinserted and compared with the entropy of revised alignment block to the entropy of the original alignment block. Since there is no gold standard to measure the performance of aggregated solutions, we used entropy as a measure of

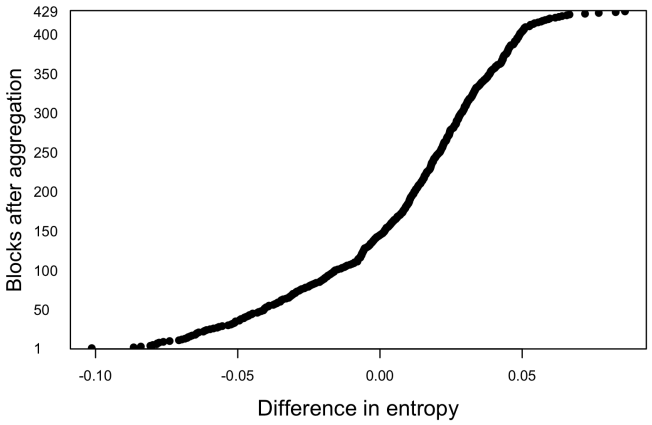

Figure 10–2: Change of entropy after aggregation of user aligned puzzles to the initial alignment. The x-axis represents the difference between the entropy of the machine aligned sequences and user aligned sequences. Positive values indicate an improvement of the alignment.

accuracy. We performed this test on 429 genome blocks that were used to generate input puzzles for the classical version of Phylo. We did observe that aggregation of solutions selected through a basic (simplified) scoring scheme available to users, is sufficient to yield significant improvements of a global problem (i.e. alignment).

10.2 Upcoming version of aggregation

10.2.1 Methodology

Pairwise sequence alignment methods seek to find the best alignment for two query sequences and are limited by the number of sequences to be aligned. This means that when aggregating multiple sequences to an alignment, only one sequence can be added at a time. On the other hand, multiple sequence alignment algorithms will align all the sequences present in a given query set. Implementing a multiple

sequence alignment could be more efficient in terms of run-time, but, multiple sequence alignment is accompanied with numerous challenges. These algorithms are computationally complex, expensive in terms of the runtime as well as space, and often lead to NP-complete combinatorial optimization problems. Furthermore, while pairwise sequence alignment is used to show similarities between two nucleotides sequences which may indicate biological relationships between the two; multiple sequence alignments seek to highlight weak signals which may represent protein families or conserved biological domains. In the upcoming version of OpenPhylo the length of the longest sequence is limited to 2000 nucleotides and a file size limitation of 5MB. Also, due to the added complexity of multiple sequence alignment algorithms and differences in primary goal, pairwise alignment algorithms were chosen to be implemented in this aggregation approach.

In an effort to perform aggregation, we have used Position Weight Matrices (PWMs) [14] and Needleman-Wunsch algorithm [44].

Position weight matrices:

are frequently used in computational biology as a key component to represent patterns in biological sequences. We started by calculating the position frequency matrix (PFM) by measuring the position-dependent frequency of each nucleotide in a DNA sequence. A PFM can be converted to a position weight matrix by calculating the log likelihoods.

$$W_{b,i} = log_2 \frac{p(b,i)}{p(b)}$$

$W_{b,i}$ represents the weight of nucleotide 'b' at index 'i',

$p(b, i)$ probabilty of 'b' at index 'i',

$p(b)$ overall probability of 'b'

Modified Needleman-Wunsch algorithm

The Needleman–Wunsch algorithm is a pairwise multiple sequence alignment algorithm used in bioinformatics to align protein or nucleotide sequences [44]. It is an aplication of dynamic programming to compare biological sequences. It aims to achieve global alignment throughout the entire input. The used algorithm for aggregation is the modified version of the Needleman-Wunsch algorithm. A matrix D(i,j) indexed by the residues of each sequence is built recursively. First we initialize the matrix: D(m+1,n+1)=0

such that

$$D(i,j) = max \begin{cases} D(i-1, j-1) + s(x_i, y_j) \\ D(i-1, j) + g \\ D(i, j-1) + g \end{cases}$$

s(i,j) is the substitution score for residues i and j

g is the gap penalty

The algorithm implementation in aggregation produces an aligned sequence from a position weight matrix of the and a to-be-aligned sequence. We use a modified version of the Needleman-Wunsch algorithm which will derive a representative sequence from

the position weight matrix which is then used in the pairwise alignment process. The alignment makes use of a dynamic programming which generates a table to keep a track of the alignment score at each position depending on if the two nucleotides are a match, mismatch, deleted or inserted nucleotide. The function then backtracks through the table to produce the aligned sequence by finding the sequence alignment with the maximal score. The scores are assigned using a similar scoring scheme as used in Phylo:

- If there is a nucleotide mismatch: -1,
- If there is a nucleotide match: +1,
- If there is a gap (insert/delete): -5

The entire aggregation process is orchestrated to takes as input a list of user-aligned sequences from Phylo and a list of unaligned sequences. It will then map the input puzzle (machine aligned) to the unique set of solutions (solutions submitted by the users via Phylo). Further, it will calculate a new position weight matrix and produce a new representative sequence every time a new sequence is aligned. This process continues until all the sequences are aligned with respect to the user aligned sequences.

10.2.2 Evaluation metric

In order to evaluate the effectiveness of multiple sequence alignment, we have used T-coffee score. The T-Coffee score (TCS) is an alignment evaluation score based on

the T-Coffee framework which uses libraries of pairwise alignments to evaluate third-party multiple sequence alignments to produce structurally accurate phylogenetic trees. The other primary reason as to why we are using T-Coffee is because all the the unaligned sequences uploaded by the users are intially scored and aligned by T-Coffee. Therefore the system will be most competitive in the case where it can challenge the one that actually produced the input alignments.

10.2.3 Results

We analysed the aggregation results using the solutions that we have received through the new version of the game. During our experimentation we found a gene: *TLR4* for which the TCS for user aggegrated solution is 982, which is comparitively better than the origin score for the T-coffee aligned solutions: 981. With time we expect the count of genes to increase, however with the limited amount of data, a single instance is good enough to showcase the validity of the proposed methodology.

CHAPTER 11
Conclusion

Multiple sequence alignments play essential role in illustrating the evolutionary relations among the sequences. It has become a prerequisite for genomic analysis pipelines and many downstream computational modes for homology modeling, secondary structure prediction, and phylogenetic reconstruction [21]. For this reason MSA have become an important research area. However, the state-of-art algorithms provide potentially suboptimal solution to this problem. Because manual curation is a necessary step to guarantee the quality of biological sequence alignments, a crowdsourcing solution appears to be a perfect strategy to address this bottleneck. Phylo provides the right interface to crowdsource this problem as microtasks to the players and thereby enhance the sequenced alignments.

Our study provides all the tools necessary to fulfill the purpose of Phylo [34, 39]. Starting with detection of regions in MSA that have some scope of improvement, we achieved a 10-fold cross validation accuracy of 81%. These promising regions were extracted from MSA and routed to the game players using a novel routing approach. We also proposed a CNN based machine learning model to predict the difficulty of routed tasks with an accuracy of 75.15%.

In addition, we also studied various human computation aspects of Phylo that are crucial for the system to perform efficiently. This includes user feedback mechanism,

teaching portal, and aggregation. Phylo provides reward as well as stoking competition among volunteers. Metrics like percentile score of tasks are readily visible and provide the judges a sense of reward in completion of the task. The most concrete way we reward our players is by giving them a contribution message that expresses their contribution in the best possible way. We attribute every user contribution in the database to the scientist responsible for producing that puzzle. For aggregation of solutions to microtasks, we have used modified Needleman-Wunsch and position weight matrix which can effectively solve the original problem of MSA.

In sum, the cognitive and motivational dimensions of human-computation which have been explored using Phylo as the exemplification can be used in broad spectrum of domains, ranging from games with a purpose (GWAP) to crowdsourcing frameworks in general. We believe that simulating the same in other applications will help establish common metrics of success, and give rise to new opportunities for research.

Appendix A

Feature based approaches to predict the promising puzzles:

The results shown below determines the promising puzzles using the feature extraction process. Labeling apprach is same as used in section 5.2.

Promising puzzle detection using LSTMs:

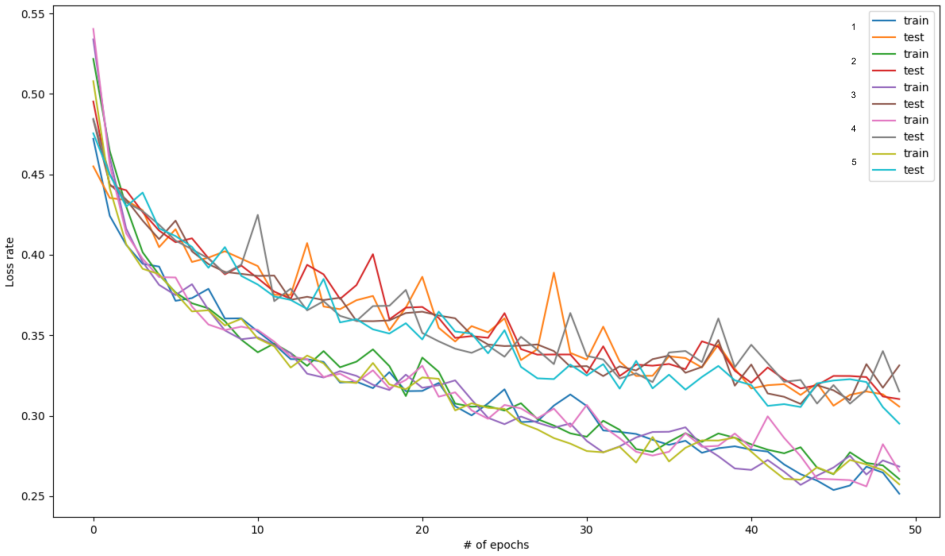

Figure 11–1: Promising puzzle prediction using LSTM. The graph was created using 480 LSTM cells, *mae* loss function and *adam* optimizer. Validation accuracy achieved: 69.4%.

Promising puzzle detection using kNN [3]:

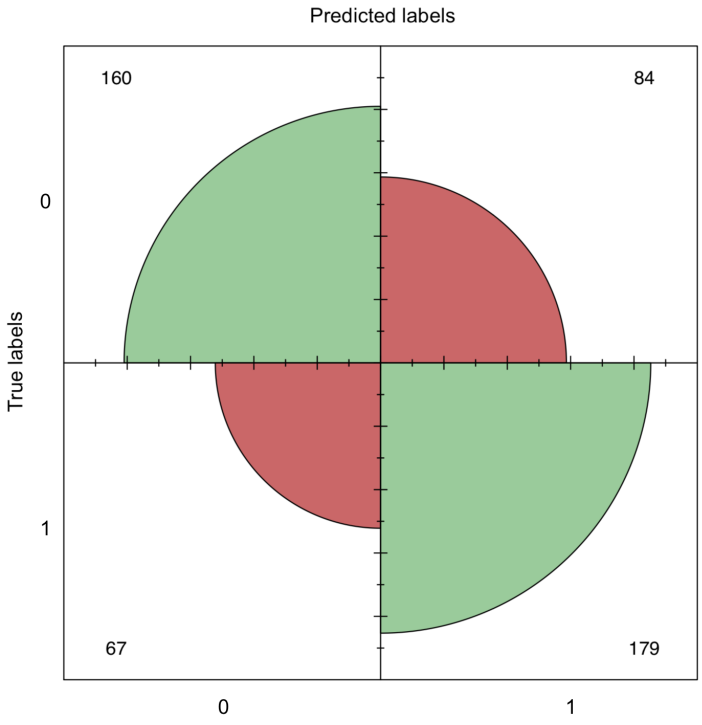

Figure 11–2: Average confusion matrix for promising puzzle prediction using kNN.

Promising puzzle detection using SVM [16]:

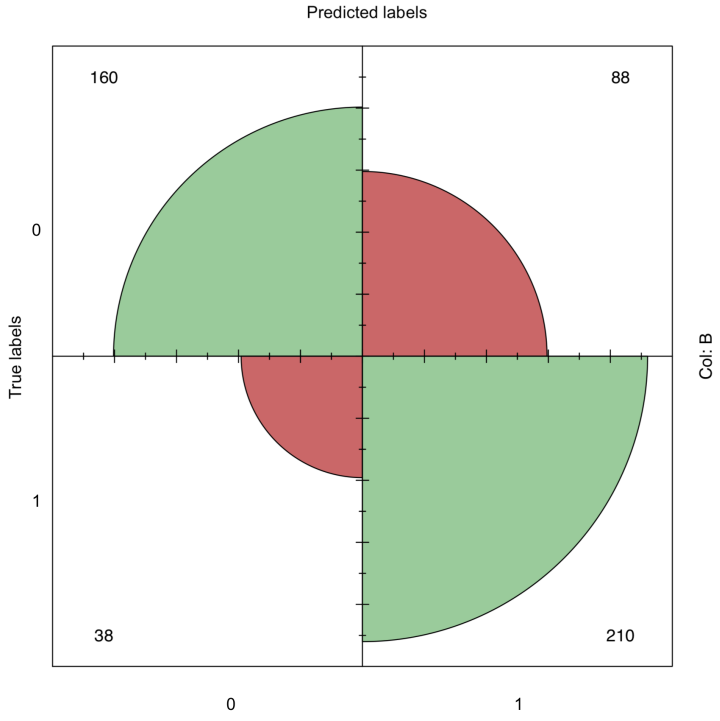

Figure 11–3: Average confusion matrix for promising puzzle prediction using SVM.

Appendix B

Difficulty prediction for RNA:

The dataset used here is the one produced via ribo game [73]. A total of 27 puzzles were used for Ribo which were played 673 times as of December 2017. Although this is insufficient data for most of the machine learning algorithms, but, we still tried to predict RNA difficulty with whatever data we had. Following features were expected for difficulty prediction of RNA:

1. Total number of base-pairs,
2. total number of gaps,
3. height of the puzzle (number of sequences in the input puzzle),
4. length of the sequence,
5. average structural entropy,
6. number of A,
7. number of C,
8. number of G,
9. number of U,
10. average structural entropy,
11. average expected energy, and

12. average free energy of the sequence.

Using backward elimination we observed that only total number of base-pairs, number of gaps, number of sequences, average structural entropy, average expected energy, and average free energy of the sequence.

Label creation: Score of each submission is the sum of the score structural and sequencial score. Difference between an alignments' best score and its original score is referred as the scope of improvement. We use this difference to obtain the normalized scores for each puzzle or alignment. Puzzles with the least possibility of improvement were considered as difficult puzzles, so if the average normalized scores of puzzles were less than or equal to 0.5 and greater than 0, then they were considered as difficult to align. Similarly, a normalized score value greater than 0.5 represents easy puzzles. Table 11–1 shows feature values along with the labels for all 27 RNA puzzles. The results and insights gained from this experiments were highly influential towards our objective mainly die to bare minimum data. However, using the huge player base of Phylo game, we can expect this appraoch to work and be a part of difficulty prediction for RNA based puzzles.

Table 11–1: Feature values for RNA puzzles used in Ribo game.

Difficulty	Base pair	Gaps	Height	Sequence length	Average structural entropy	Average expected energy	Average free energy of sequence
0	34	5	4	25	2.02	-0.74	-3.28
0	28	13	4	25	2	-0.46	-1.67
1	36	1	6	25	1.55	0.26	-0.69
1	61	4	6	25	1.01	-7.39	-8.01
0	27	5	8	25	1.46	0.28	0.62
1	61	2	8	25	1.26	-6.52	-7.3
1	31	5	10	25	1.67	-1.68	-2.7
0	40	6	10	25	1.4	-4.5	-5.3
0	37	1	4	35	2.66	-2.68	-4.31
0	55	1	4	35	2.61	-5.95	-7.56
0	39	6	6	35	3.05	-2.7	-4.58
0	44	8	6	35	1.82	-8.53	-9.66
1	30	1	8	35	2.47	-2.37	-3.89
1	38	18	8	35	2.04	-4.32	-5.58
1	35	4	10	35	2.9	-2.19	-3.99
1	47	4	10	35	2.59	-7.75	-9.35
0	32	25	4	25	1.34	-0.79	-1.62
0	40	12	4	25	1.53	-3.56	-4.5
0	46	11	6	25	2.24	-1.04	-2.42
0	33	6	6	25	1.5	-5.8	-6.73
0	32	8	4	25	2.4	-1.1	-2.58
0	37	19	4	35	2.2	-4.04	-5.4
0	38	11	6	35	1.8	-5.45	-6.13
1	32	22	8	35	1.94	-1.39	-2.88
1	40	11	8	35	3.15	-2.02	-3.96
0	34	6	10	35	1.76	-1.57	2.65
1	38	8	10	35	1.79	-6.7	-7.82

Appendix C

Table 11–2: Dataset of puzzles for dynamic difficulty change

puzzle_id	**category**	**#Solution**	**difficulty**	**#attempt**	**P_win**
1	Cancers	150	1	154	0.974
2	Infectious diseases	87	1	89	0.968
3	Infectious diseases	4	1	4	0.963
4	Heart and Muscles	180	1	186	0.966
5	Metabolic Diseases	15	1	15	0.971
6	Brain, nervous and sensory system diseases	279	1	284	0.980
7	Blood and immune system diseases	1	1	1	0.978
8	Digestive and respiratory system diseases	338	1	351	0.962
9	Other diseases	19	1	19	0.980
10	Other diseases	114	1	121	0.939
11	Cancers	151	1	164	0.920
12	Blood and immune system diseases	110	1	114	0.965

Table 11–2 continued from previous page

13	Metabolic Diseases	60	1	61	0.969
14	Cancers	111	2	115	0.958
15	Infectious diseases	43	2	44	0.969
16	Other diseases	33	2	37	0.880
17	Heart and Muscles	47	2	49	0.958
18	Metabolic Diseases	8	2	8	0.936
19	Brain, nervous and sensory system diseases	60	2	64	0.930
20	Blood and immune system diseases	76	2	77	0.980
21	Digestive and respiratory system diseases	85	2	86	0.979
22	Heart and Muscles	50	2	53	0.931
23	Other diseases	56	2	58	0.955
24	Brain, nervous and sensory system diseases	55	2	56	0.980
25	Metabolic Diseases	39	2	40	0.974
26	Blood and immune system diseases	71	2	75	0.939
27	Digestive and respiratory system diseases	74	2	83	0.884

Table 11–2 continued from previous page

28	Cancers	62	3	66	0.937
29	Infectious diseases	91	3	92	0.987
30	Other diseases	14	3	14	0.982
31	Heart and Muscles	65	3	66	0.976
32	Metabolic Diseases	24	3	31	0.768
33	Brain, nervous and sensory system diseases	87	3	106	0.818
34	Blood and immune system diseases	14	3	16	0.860
35	Digestive and respiratory system diseases	14	3	16	0.825
36	Other diseases	42	3	54	0.767
37	Blood and immune system diseases	38	3	49	0.772
38	Metabolic Diseases	33	3	39	0.831
39	Digestive and respiratory system diseases	30	3	35	0.852
40	Cancers	82	4	87	0.941
41	Infectious diseases	32	4	48	0.667
42	Other diseases	4	4	4	0.922
43	Heart and Muscles	25	4	26	0.935

Table 11–2 continued from previous page

44	Metabolic Diseases	19	4	19	0.953
45	Brain, nervous and sensory system diseases	5	4	5	0.948
46	Blood and immune system diseases	40	4	44	0.893
47	Digestive and respiratory system diseases	22	4	26	0.838
48	Digestive and respiratory system diseases	40	4	42	0.945
49	Blood and immune system diseases	37	4	37	0.976
50	Brain, nervous and sensory system diseases	32	4	32	0.973
51	Cancers	16	5	18	0.875
52	Infectious diseases	32	5	34	0.923
53	Other diseases	9	5	9	0.952
54	Heart and Muscles	31	5	33	0.925
55	Metabolic Diseases	8	5	8	0.934
56	Brain, nervous and sensory system diseases	30	5	30	0.971

Table 11–2 continued from previous page

57	Blood and immune system diseases	33	5	34	0.963
58	Digestive and respiratory system diseases	18	5	18	0.963
59	Other diseases	5	5	5	0.983
60	Other diseases	15	5	15	0.982
61	Cancers	36	5	38	0.943
62	Metabolic Diseases	13	5	13	0.935
63	Cancers	23	6	24	0.925
64	Infectious diseases	30	6	35	0.846
65	Other diseases	24	6	25	0.949
66	Heart and Muscles	29	6	33	0.872
67	Metabolic Diseases	33	6	38	0.849
68	Brain, nervous and sensory system diseases	16	6	18	0.880
69	Blood and immune system diseases	24	6	26	0.913
70	Digestive and respiratory system diseases	16	6	18	0.889
71	Cancers	35	7	38	0.921
72	Infectious diseases	26	7	74	0.348

Table 11–2 continued from previous page

73	Other diseases	13	7	13	0.989
74	Heart and Muscles	25	7	28	0.872
75	Metabolic Diseases	12	7	12	0.927
76	Brain, nervous and sensory system diseases	22	7	23	0.939
77	Blood and immune system diseases	21	7	22	0.935
78	Digestive and respiratory system diseases	91	7	105	0.861
79	Cancers	4	8	5	0.712
80	Infectious diseases	23	8	27	0.825
81	Other diseases	13	8	17	0.726
82	Heart and Muscles	2	8	2	0.850
83	Metabolic Diseases	29	8	36	0.803
84	Brain, nervous and sensory system diseases	14	8	22	0.627
85	Blood and immune system diseases	2	8	2	0.885
86	Digestive and respiratory system diseases	49	8	51	0.944
87	Cancers	9	9	11	0.773

Table 11–2 continued from previous page

88	Infectious diseases	17	9	23	0.722
89	Other diseases	20	9	27	0.718
90	Heart and Muscles	11	9	13	0.825
91	Metabolic Diseases	9	9	10	0.825
92	Brain, nervous and sensory system diseases	13	9	16	0.773
93	Blood and immune system diseases	10	9	14	0.712
94	Digestive and respiratory system diseases	15	9	21	0.708
95	Cancers	239	10	281	0.850
96	Other diseases	249	10	371	0.671
97	Metabolic Diseases	223	10	306	0.729
98	Brain, nervous and sensory system diseases	209	10	229	0.911
99	Blood and immune system diseases	215	10	252	0.850
100	Other diseases	49	10	97	0.505

Appendix D

Latency and robustness of solutions

One important aspect to guarantee the efficiency of human-computing systems at solving an optimization problem, is to make the best possible estimate of the number of solutions that needs to be collected to offer some confidence in the quality of the solution returned by the system. In particular, we need to identify the parameters that influence these estimates. Figure 11–4 shows that with an average of 320 puzzles solved per day (average over the last 6 years) most of the improvements in alignment scores are obtained relatively quickly, within the first ≈ 100 days. Past this age, the accumulation of solutions appears to be redundant and the puzzle can be retired.

Then we investigate the impact of the difficulty of puzzles on these statistics. Table 11–3 shows the number of attempts taken to get to the highest score. In other words, the number of solutions we need to collect to get one that is not further improved by players. On an average, the ranks of the highest scoring alignment are comparable, albeit a slight (expected) increase of the rank is observed on the most difficult puzzles. The length of the period during which we collected solutions enables us to offer some guarantees on the robustness of these estimates. In average, the highest scores got reached after collecting only $\approx 30\%$ of the total number of solutions. These statistics allow us to estimate the volume of data that can be treated by our system per. With an average of 320 solutions collected per day, we can currently envision to solve 2-4 puzzles per day depending of their complexity and the target level of confidence

Table 11–3: Number of puzzles collected and rank of highest scoring alignments.

Difficulty	Number of Puzzles	Number of Solutions	Rank of highscore	
			μ	Q_3
Easy	25	2075	83	150
Medium	255	21678	85	170
Difficult	1117	103635	93	192

in the results. Thus, a typical genomic bloc within a week. Of course, regular or occasional growth of the traffic will increase these numbers [60].

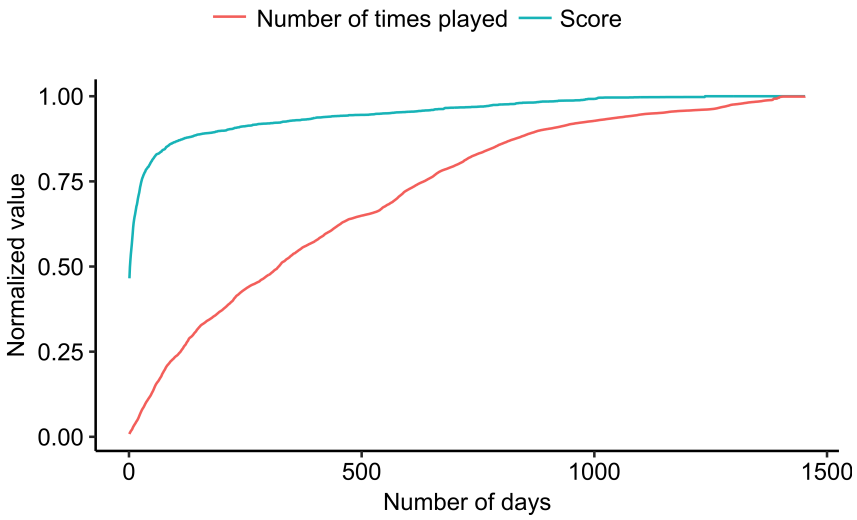

Figure 11–4: Progression of high scores. The orange line shows the normalized high scores since the release of the puzzles. The blue curve plots the number of solutions collected.

References

[1] Tatsuya Akutsu, Hiroki Arimura, and Shinichi Shimozono. On approximation algorithms for local multiple alignment. In *Proceedings of the fourth annual international conference on Computational molecular biology*, pages 1–7. ACM, 2000.

[2] Gary L Allen, Kathleen C Kirasic, Shannon H Dobson, Richard G Long, and Sharon Beck. Predicting environmental learning from spatial abilities: An indirect route. *Intelligence*, 22(3):327–355, 1996.

[3] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[4] Peter Bøgh Andersen. *A theory of computer semiotics: Semiotic approaches to construction and assessment of computer systems*, volume 3. Cambridge University Press, 1990.

[5] Inc Apple Computer. *Apple Human Interface Guidelines: The Apple Desktop Interface*. Addison Wesley Publishing Company, 1987.

[6] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[7] Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 313–322. ACM, 2010.

[8] Mathieu Blanchette. Computation and analysis of genomic multi-sequence alignments. *Annu. Rev. Genomics Hum. Genet.*, 8:193–213, 2007.

[9] Mathieu Blanchette, W James Kent, Cathy Riemer, Laura Elnitski, Arian FA Smit, Krishna M Roskin, Robert Baertsch, Kate Rosenbloom, Hiram Clawson,

Eric D Green, et al. Aligning multiple genomic sequences with the threaded blockset aligner. *Genome research*, 14(4):708–715, 2004.

[10] Sarah W Burge, Jennifer Daub, Ruth Eberhardt, John Tate, Lars Barquist, Eric P Nawrocki, Sean R Eddy, Paul P Gardner, and Alex Bateman. Rfam 11.0: 10 years of rna families. *Nucleic acids research*, 41(D1):D226–D232, 2012.

[11] Humberto Carrillo and David Lipman. The multiple sequence alignment problem in biology. *SIAM Journal on Applied Mathematics*, 48(5):1073–1082, 1988.

[12] Justin Cheng, Jaime Teevan, Shamsi T Iqbal, and Michael S Bernstein. Break it down: A comparison of macro-and microtasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 4061–4064. ACM, 2015.

[13] Chao-Min Chiu, Meng-Hsiang Hsu, and Eric TG Wang. Understanding knowledge sharing in virtual communities: An integration of social capital and social cognitive theories. *Decision support systems*, 42(3):1872–1888, 2006.

[14] Jean-Michel Claverie and Stephane Audic. The statistical significance of nucleotide position-weight matrix matches. *Bioinformatics*, 12(5):431–439, 1996.

[15] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, et al. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, 2010.

[16] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[17] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[18] MO Dayhoff and BC Orcutt. Methods for identifying proteins by using partial sequences. *Proceedings of the National Academy of Sciences*, 76(5):2170–2174, 1979.

[19] Samuel Dodge and Lina Karam. Understanding how image quality affects deep neural networks. In *Quality of Multimedia Experience (QoMEX), 2016 Eighth International Conference on*, pages 1–6. IEEE, 2016.

[20] Robert C Edgar. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792–1797, 2004.

[21] Robert C Edgar and Serafim Batzoglou. Multiple sequence alignment. *Current opinion in structural biology*, 16(3):368–373, 2006.

[22] Philipp Eisen. Simulating human game play for level difficulty estimation with convolutional neural networks, 2017.

[23] Da-Fei Feng and Russell F Doolittle. Progressive sequence alignment as a prerequisitetto correct phylogenetic trees. *Journal of molecular evolution*, 25(4):351–360, 1987.

[24] Walter M Fitch. Toward defining the course of evolution: minimum change for a specific tree topology. *Systematic Biology*, 20(4):406–416, 1971.

[25] Paul P Gardner, Jennifer Daub, John G Tate, Eric P Nawrocki, Diana L Kolbe, Stinus Lindgreen, Adam C Wilkinson, Robert D Finn, Sam Griffiths-Jones, Sean R Eddy, et al. Rfam: updates to the rna families database. *Nucleic acids research*, 37(suppl_1):D136–D140, 2008.

[26] Benjamin M Good, Salvatore Loguercio, Obi L Griffith, Max Nanis, Chunlei Wu, and Andrew I Su. The cure: design and evaluation of a crowdsourcing game for gene selection for breast cancer survival prediction. *JMIR Serious Games*, 2(2), 2014.

[27] Sam Griffiths-Jones, Alex Bateman, Mhairi Marshall, Ajay Khanna, and Sean R Eddy. Rfam: an rna family database. *Nucleic acids research*, 31(1):439–441, 2003.

[28] Chien-Ju Ho and Jennifer Wortman Vaughan. Online task assignment in crowdsourcing markets. In *AAAI*, volume 12, pages 45–51, 2012.

[29] Ivo L Hofacker, Walter Fontana, Peter F Stadler, L Sebastian Bonhoeffer, Manfred Tacker, and Peter Schuster. Fast folding and comparison of rna secondary structures. *Monatshefte für Chemie/Chemical Monthly*, 125(2):167–188, 1994.

[30] Mahmood Hosseini, Keith Phalp, Jacqui Taylor, and Raian Ali. The four pillars of crowdsourcing: A reference model. In *Research Challenges in Information Science (RCIS), 2014 IEEE Eighth International Conference on*, pages 1–12. IEEE, 2014.

[31] Ioanna Kalvari, Joanna Argasinska, Natalia Quinones-Olvera, Eric P Nawrocki, Elena Rivas, Sean R Eddy, Alex Bateman, Robert D Finn, and Anton I Petrov. Rfam 13.0: shifting to a genome-centric resource for non-coding rna families. *Nucleic acids research*, 46(D1):D335–D342, 2017.

[32] Ruth Kanfer and Phillip L Ackerman. Motivation and cognitive abilities: An integrative/aptitude-treatment interaction approach to skill acquisition. *Journal of applied psychology*, 74(4):657, 1989.

[33] Kazutaka Katoh, Kazuharu Misawa, Kei-ichi Kuma, and Takashi Miyata. Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic acids research*, 30(14):3059–3066, 2002.

[34] Alexander Kawrykow, Gary Roumanis, Alfred Kam, Daniel Kwak, Clarence Leung, Chu Wu, Eleyine Zarour, Luis Sarmenta, Mathieu Blanchette, Jérôme Waldispühl, et al. Phylo: a citizen science approach for improving multiple sequence alignment. *PloS one*, 7(3):e31362, 2012.

[35] Firas Khatib, Seth Cooper, Michael D Tyka, Kefan Xu, Ilya Makedon, Zoran Popović, and David Baker. Algorithm discovery by protein folding game players. *Proceedings of the National Academy of Sciences*, 108(47):18949–18953, 2011.

[36] Jinseop S Kim, Matthew J Greene, Aleksandar Zlateski, Kisuk Lee, Mark Richardson, Srinivas C Turaga, Michael Purcaro, Matthew Balkam, Amy Robinson, Bardia F Behabadi, et al. Space-time wiring specificity supports direction selectivity in the retina. *Nature*, 509(7500):331–336, 2014.

[37] Nicolas Kokkalis, Thomas Köhn, Johannes Huebner, Moontae Lee, Florian Schulze, and Scott R Klemmer. Taskgenies: Automatically providing action plans helps people complete tasks. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 20(5):27, 2013.

[38] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[39] Daniel Kwak, Alfred Kam, David Becerra, Qikuan Zhou, Adam Hops, Eleyine Zarour, Arthur Kam, Luis Sarmenta, Mathieu Blanchette, and Jérôme Waldispühl. Open-phylo: a customizable crowd-computing platform for multiple sequence alignment. *Genome biology*, 14(10):R116, 2013.

[40] David J Lane, Bernadette Pace, Gary J Olsen, David A Stahl, Mitchell L Sogin, and Norman R Pace. Rapid determination of 16s ribosomal rna sequences for phylogenetic analyses. *Proceedings of the National Academy of Sciences*, 82(20):6955–6959, 1985.

[41] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[42] Jeehyung Lee, Wipapat Kladwang, Minjae Lee, Daniel Cantu, Martin Azizyan, Hanjoo Kim, Alex Limpaecher, Snehal Gaikwad, Sungroh Yoon, Adrien Treuille, et al. Rna design rules from a massive open laboratory. *Proceedings of the National Academy of Sciences*, 111(6):2122–2127, 2014.

[43] Andreas Lieberoth, Mads Kock Pedersen, Andreea Catalina Marin, Tilo Planke, and Jacob Friis Sherson. Getting humans to do quantum optimization-user acquisition, engagement and early results from the citizen cyberscience game quantum moves. *arXiv preprint arXiv:1506.08761*, 2015.

[44] Vladimir Likic. The needleman-wunsch algorithm for sequence alignment. *Lecture given at the 7th Melbourne Bioinformatics Course, Bi021 Molecular Science and Biotechnology Institute, University of Melbourne*, pages 1–46, 2008.

[45] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[46] Dennis V Lindley and LD Phillips. Inference for a bernoulli process (a bayesian view). *The American Statistician*, 30(3):112–119, 1976.

[47] Salvatore Loguercio, Benjamin M Good, and Andrew I Su. Dizeez: an online game for human gene-disease annotation. *PLoS One*, 8(8):e71171, 2013.

[48] Ronny Lorenz, Stephan H Bernhart, Christian Hoener Zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. Viennarna package 2.0. *Algorithms for Molecular Biology*, 6(1):26, 2011.

[49] Ari Löytynoja and Nick Goldman. Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. *Science*, 320(5883):1632–1635, 2008.

[50] Miguel Angel Luengo-Oroz, Asier Arranz, and John Frean. Crowdsourcing malaria parasite quantification: an online game for analyzing images of infected thick blood smears. *Journal of medical Internet research*, 14(6), 2012.

[51] Bin Ma, John Tromp, and Ming Li. Patternhunter: faster and more sensitive homology search. *Bioinformatics*, 18(3):440–445, 2002.

[52] Nico JD Nagelkerke et al. A note on a general definition of the coefficient of determination. *Biometrika*, 78(3):691–692, 1991.

[53] Eric P Nawrocki, Diana L Kolbe, and Sean R Eddy. Infernal 1.0: inference of rna alignments. *Bioinformatics*, 25(10):1335–1337, 2009.

[54] Cédric Notredame, Desmond G Higgins, and Jaap Heringa. T-coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1):205–217, 2000.

[55] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[56] Thomas A Peters. Human factors in information systems: Emerging theoretical bases. *Journal of the American Society for Information Science*, 47(8):655–656, 1996.

[57] Daniel Quang and Xiaohui Xie. Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences. *Nucleic acids research*, 44(11):e107–e107, 2016.

[58] Karen Renaud and Richard Cooper. Feedback in human-computer interaction-characteristics and recommendations. *South African Computer Journal*, 2000(26):105–114, 2000.

[59] Brooke Rhead, Donna Karolchik, Robert M Kuhn, Angie S Hinrichs, Ann S Zweig, Pauline A Fujita, Mark Diekhans, Kayla E Smith, Kate R Rosenbloom, Brian J Raney, et al. The ucsc genome browser database: update 2010. *Nucleic acids research*, page gkp939, 2009.

[60] Henry Sauermann and Chiara Franzoni. Crowd science user contribution patterns and their implications. *Proc Natl Acad Sci U S A*, 112(3):679–84, Jan 2015.

[61] Pamela A Savage-Knepshield and Nicholas J Belkin. Interaction in information retrieval: Trends over time. *Journal of the American Society for Information Science*, 50(12):1067–1082, 1999.

[62] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[63] Ben Shneiderman. Designing information-abundant websites. Technical report, 1998.

[64] Christian JA Sigrist, Lorenzo Cerutti, Nicolas Hulo, Alexandre Gattiker, Laurent Falquet, Marco Pagni, Amos Bairoch, and Philipp Bucher. Prosite: a documented database using patterns and profiles as motif descriptors. *Briefings in bioinformatics*, 3(3):265–274, 2002.

[65] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[66] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[67] Singh, Ahsan, Blanchette, and Waldispühl. Lessons from an online massive genomics computer game. 2017.

[68] Ramin A Skibba, Karen L Masters, Robert C Nichol, Idit Zehavi, Ben Hoyle, Edward M Edmondson, Steven P Bamford, Carolin N Cardamone, William C Keel, Chris Lintott, et al. Galaxy zoo: the environmental dependence of bars and bulges in disc galaxies. *Monthly Notices of the Royal Astronomical Society*, 423(2):1485–1502, 2012.

[69] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[70] Jaime Teevan, Shamsi T Iqbal, Carrie J Cai, Jeffrey P Bigham, Michael S Bernstein, and Elizabeth M Gerber. Productivity decomposed: Getting big

things done with little microtasks. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 3500–3507. ACM, 2016.

[71] Petros Venetis and Hector Garcia-Molina. Quality control for comparison microtasks. In *Proceedings of the first international workshop on crowdsourcing and data mining*, pages 15–21. ACM, 2012.

[72] Albert J Vilella, Jessica Severin, Abel Ureta-Vidal, Li Heng, Richard Durbin, and Ewan Birney. Ensemblcompara genetrees: Complete, duplication-aware phylogenetic trees in vertebrates. *Genome research*, 19(2):327–335, 2009.

[73] Jérôme Waldispühl, Arthur Kam, and Paul Gardner. Crowdsourcing rna structural alignments with an online computer game. *bioRxiv*, page 009902, 2014.

[74] Waldispühl and Blanchette. Phylo & Open-Phylo: A human-computing platform for comparative genomics. Adjunct to the Proceedings of the Second Conference on Human Computation and Crowdsourcing (2 pages), November 2014.

[75] Lusheng Wang and Tao Jiang. On the complexity of multiple sequence alignment. *Journal of computational biology*, 1(4):337–348, 1994.

[76] Robert C Wilson, Andra Geana, John M White, Elliot A Ludvig, and Jonathan D Cohen. Humans use directed and random exploration to solve the explore–exploit dilemma. *Journal of Experimental Psychology: General*, 143(6):2074, 2014.

[77] Yonqing Zhang, Supriyo De, John R Garner, Kirstin Smith, S Alex Wang, and Kevin G Becker. Systematic analysis, comparison, and integration of disease based human genetic association data and mouse genetic phenotypic information. *BMC medical genomics*, 3(1):1, 2010.