COVID Sequence Alignment using Deep Reinforcement Learning

J. Syam Koteswar, J. Yotesh, G. Rajesh, P. Pavan Kumar

Vasireddy Venkatadri Institute of Technology, Guntur 522508, India

Abstract

Sequence alignment plays an important role in comparative genomic sequence analysis, being one of the most challenging problem in bioinformatics. It is the first step in solving many bioinformatics problems. Multiple of DNA, RNA or Proteins in order to maximize their regions of similarity. It is an NP-complete problem. There are many genetic-based approaches, many heuristic mentioned there are so many methods as iterative method, progressive alignment method, Dynamic programming approach which are not much accurate and has more time complexity. Now we are improving the accuracy using reinforcement learning and utilizes neural networks for estimation phase in the reinforcement learning algorithm. We perform our analysis on COVID dataset.

Keywords- : Sequence alignment, Deep reinforcement learning, Q-Learning

1. Introduction:

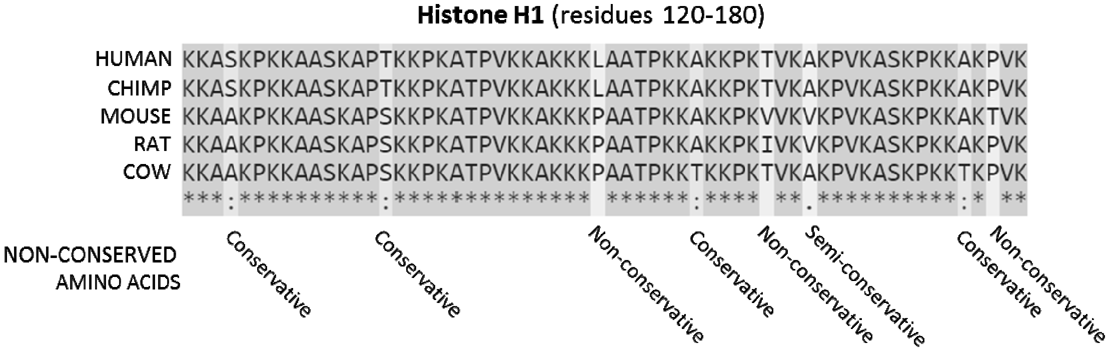
Recent advances in sequencing technology have made it possible to examine creatures with lengthy genomes[1]. Until far, we've only had cases of creatures with short sequences, which allowed us to quickly examine evolutionary distances between them using older pairwise alignment methods., such as the conventional Needleman–Wunsch (NW) algorithm[3], and the relationship bet-ween the organisms could be investigated. However, Because the traditional pairwise alignment method[4] makes determining the features of genomic sequences so simple, it's challenging to strike a balance between complexity and performance. Several attempts to align lengthy sequences have failed. Because of the complexity of the NW alignment approach, it is extremely difficult to align.

Multiple sequence alignment (MSA)[5] is a method for aligning multiple sequences at the same time that is currently being developed. Detecting similar subsequences, arranging the progressive alignment, and boosting the speed using multithreads based on GPUs[6] are just a few of the ways this technology has been improved. However, The complexity of pairwise alignment may be more severe in the case of MSA, which is an important issue to remember. In particular, there are several pairwise alignment algorithms, such as banded alignment, the BLAST[7], and the MUMMER[8] have been proposed to improve the speed of pairwise alignment. By restricting the alignment's range and increasing it after word matching or average frequent substring matching, these alignment approaches sought to overcome the complexity problem. However, there were some accuracy issues when expanding small local alignments to vast and complicated sequences in these situations.

We presented a novel alignment approach based on a deep reinforcement learning[9] agent to solve the issues. Reinforcement learning[10] is a method of teaching an agent to pick the optimal behaviours in a given system by monitoring the surroundings. Expiring and learning vast and complicated systems is challenging with traditional tabular-based reinforcement learning. The deep reinforcement learning approach was introduced to enhance this, and it overcome the restrictions by learning complicated systems approximately. The advancement of reinforcement learning has resulted in incredible results in a variety of complicated systems.As a result, we choose to apply this deep reinforcement learning approach to a sequence alignment system that seeks for the best matches between two full sequences. As a result, in this work, we will discuss the application of deep reinforcement learning to the sequence alignment system.

2. Sequence Alignment:

A sequence alignment[11] is a method of organising DNA, RNA, or protein sequences in order to find areas of similarity that may be the result of functional, structural, or evolutionary connections between the sequences. Nucleotide or amino acid residue sequences that are aligned are generally represented as rows in a matrix. To align identical or similar characters in successive columns, gaps are added between the residues. Sequence alignments may also be utilised for non-biological sequences, such as estimating the cost of distance between strings in plain language or financial data.


Histone H1 (residues 120-180)
HUMAN KKASKPKKAASKAPTKKPKATPVKKAKKKLAATPKKAKKPKTVKAKPVKASKPKKAKPVK
CHIMP KKASKPKKAASKAPTKKPKATPVKKAKKKLAATPKKAKKPKTVKAKPVKASKPKKAKPVK
MOUSE KKAAKPKKAASKAPSKKPKATPVKKAKKKPAATPKKAKKPKVVKVKPVKASKPKKAKTVK
RAT KKAAKPKKAASKAPSKKPKATPVKKAKKKPAATPKKAKKPKIVKVKPVKASKPKKAKPVK
COW KKAAKPKKAASKAPSKKPKATPVKKAKKKPAATPKKTKKPKTVKAKPVKASKPKKTKPVK
:*******:************** ******:**** **.**********:* **
NON-CONSERVED AMINO ACIDS
Conservative
Conservative
Non-conservative
Conservative
Non-conservative
Semi-conservative
Conservative
Non-conservative

The goal of a sequence alignment is generally to match the two sequences' homologous places. The homologous locations are those that originate from the same ancestral position. Because we don't know the original sequence, we can't be certain that we've succeeded. Aligning protein sections that have the same structure or function might be a secondary goal.

Aligning comparable sequences with any method typically produces proper alignments, however aligning sequences that are extremely dissimilar might be difficult. After a considerable time has elapsed after the species separated, mutations may have altered the sequences so much that any significant commonalities may have been lost, making a meaningful alignment problematic.

Evaluating the alignments:

To assess the many potential alignments, we employ a scoring system that rewards physiologically more plausible alignments with higher points. In an ideal world[12], we'd devise a scoring system that rewards alignments that align homologous locations with higher points.Counting the number of matching locations or the number of matching places along 100 residues might be a naïve scoring method. The amount of gaps is usually taken into account by scoring systems. They punish alignments based on the number and length of gaps they find.

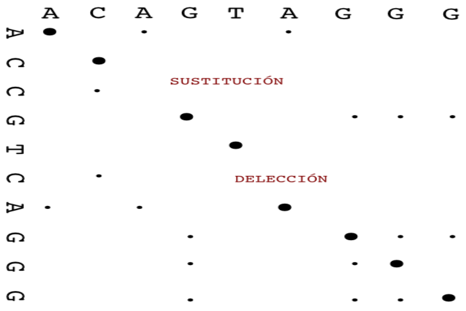
- scoring schema 1: match +1, mismatch: 0, gap creation: -1 gap extension: -1
- scoring schema 1: match +1, mismatch: -1, gap creation: -1 gap extension: -1

3. Pairwise alignment:

The best-matching piecewise (local or global) alignments of two query sequences are found using pairwise sequence alignment techniques[13]. Although pairwise alignments may only be utilised between two sequences at a time, they are quick to calculate and are frequently employed for approaches that do not require great accuracy (such as searching a database for sequences with high similarity to a query). Dot-matrix methods, dynamic programming, and word approaches are the three major methods for creating pairwise alignments; however, other sequence alignment techniques can also match pairs of sequences.Although each technique has its own set of advantages and disadvantages, all three pairwise methods struggle with extremely repeated sequences with little information richness, particularly when the number of repetitions in the two sequences to be aligned differs.

Dot-matrix method:

Dot matrix analysis[14] is essentially a method for comparing two sequences in order to check for probable character alignment. The technique is also used to predict areas in RNA that are self-complementary and so have the ability to generate secondary structure through base-pairing, as well as to detect straight or inverted repetitions in protein and DNA sequences.



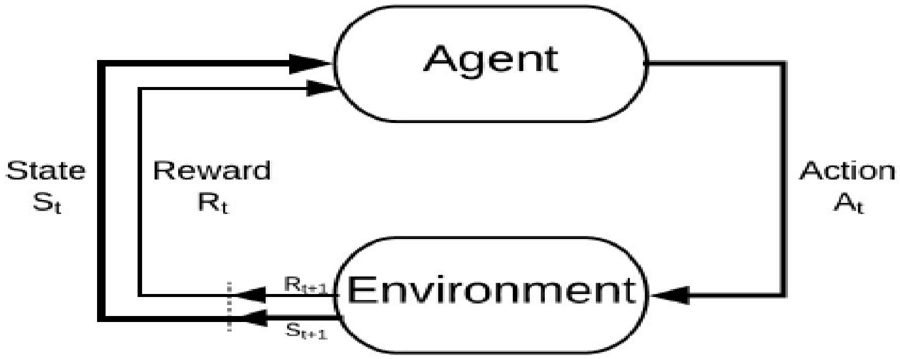
Dynamic programing:

The Needleman-Wunsch[3] method may be used to generate global alignments, and the Smith-Waterman algorithm can be used to produce local alignments. Protein alignments often employ a substitution matrix to award points to amino-acid matches and mismatches, as well as a gap penalty for matching an amino acid from one sequence to a gap in the other.

A scoring matrix may be used in DNA and RNA alignments, although in reality, a positive match score, a negative mismatch score, and a negative gap penalty are commonly used. (Because the score of each amino acid position is independent of the identity of its neighbours in traditional dynamic programming, base stacking effects are ignored.) However, by altering the method, such impacts may be accounted for.) The use of two distinct gap penalties for opening and expanding a gap is a typical addition to normal linear gap charges.The former is usually significantly more than the latter, for example, -10 for gap open and -2 for gap extension. As a result, the number of gaps in an alignment is usually minimised, and residues and gaps are often retained together, making biological sense.

4. Deep Reinforcement Learning:

Machine Learning includes the field of reinforcement learning[10]. It's all about taking the right steps to maximise your benefit in a given circumstance. It is used by a variety of software and computers to determine the best feasible action or path in a given scenario. Reinforcement learning differs from supervised learning in that the solution key is included in the training data, allowing the model to be trained with the right answer, but in reinforcement learning, there is no answer and the reinforcement agent determines what to do to complete the job. It is obliged to learn from its experience in the absence of a training dataset.

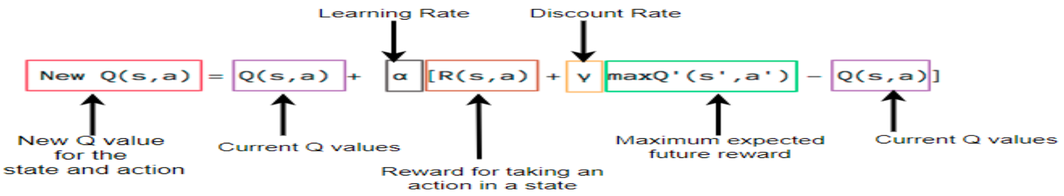


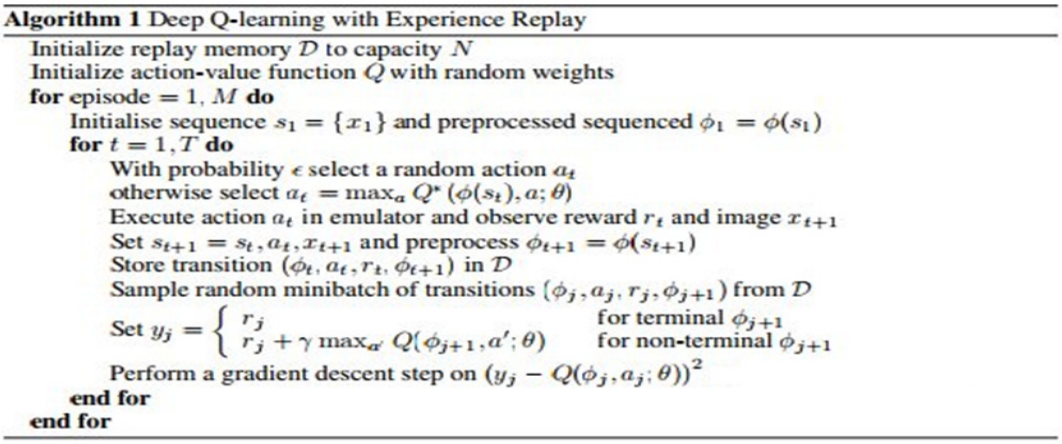
Q Learning:

Q Learning[15] is a type of learning process in which a learning agent learns to behave optimally in a given environment over time by interacting with it constantly. Throughout its learning process, the agent encounters a variety of circumstances in the environment. These are referred to as states. While in that condition, the agent can pick from a set of permissible behaviours, each of which can result in a different reward.

Over time, the learning agent learns to maximize these rewards in order to behave optimally in any given condition. Q-Learning[16] is a kind of Reinforcement Learning that use Q-values (also known as action values) to iteratively enhance the learning agent's behaviour. For states and actions, Q-values are specified. Q(S,A) is a measure of how beneficial it is to perform action A at the current condition S. The TD-Update procedure, which we shall examine in the next sections, will be used to iteratively compute this estimation of Q(S,A).Over the course of its existence, an agent begins in one state and progresses through a series of transitions from that state to the next based on its actions and the environment with which it interacts.

At each stage of the transition, the agent from one state performs an activity, sees a reward from the environment, and then moves on to the next. If the agent ends up in one of the terminating states at any point in time, it implies that no more transitions are conceivable.

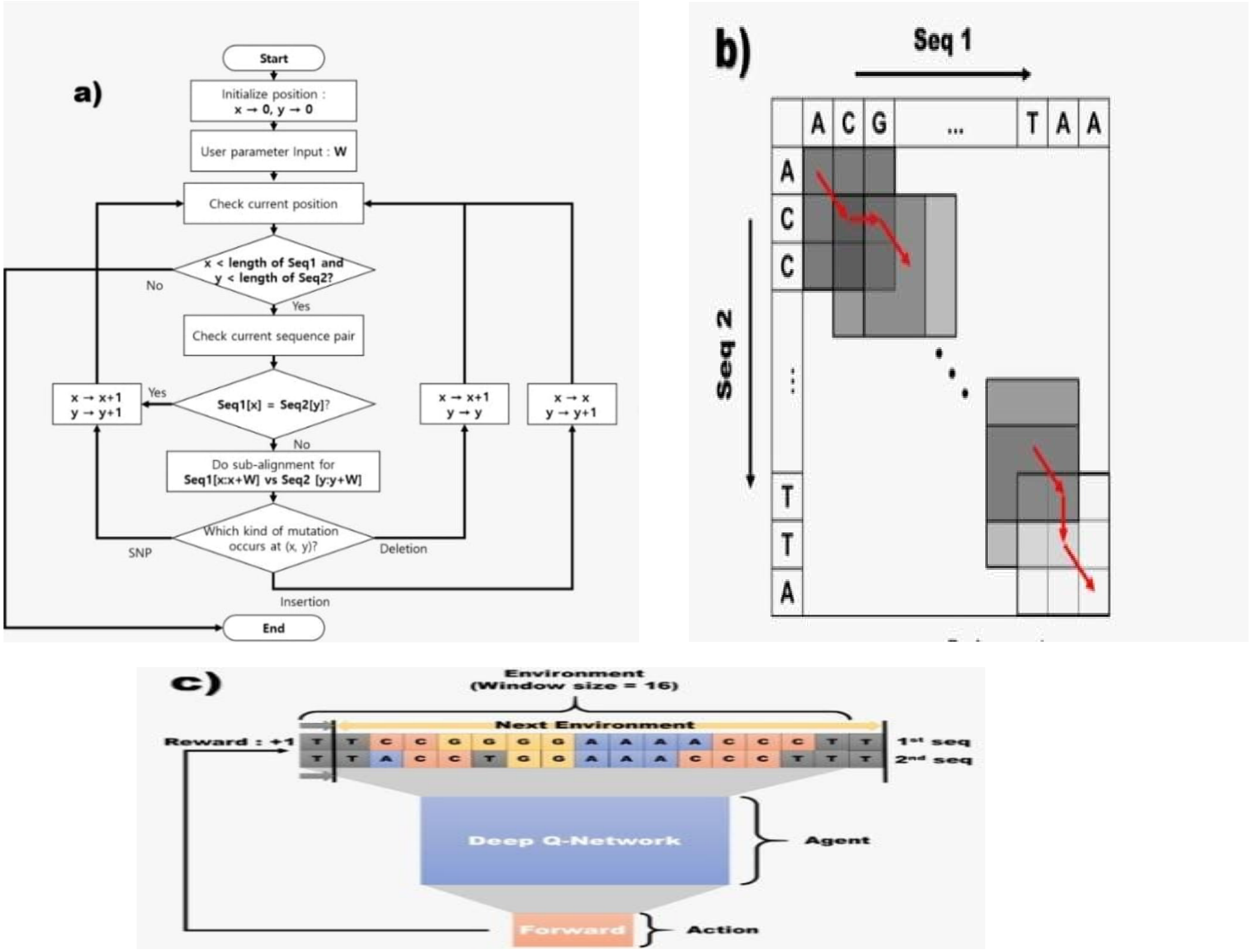

Learning Rate
Discount Rate
New Q(s,a) = Q(s,a) + α [R(s,a) + γ maxQ'(s',a') − Q(s,a)]
New Q value for the state and action
Current Q values
Reward for taking an action in a state
Maximum expected future reward
Current Q values



```
Algorithm 1 Deep Q-learning with Experience Replay
  Initialize replay memory $\mathcal{D}$ to capacity $N$
  Initialize action-value function $Q$ with random weights
  for episode $= 1, M$ do
      Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
      for $t = 1, T$ do
          With probability $\epsilon$ select a random action $a_t$
          otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
          Execute action $a_t$ in emulator and observe reward $r_t$ and image $x_{t+1}$
          Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
          Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
          Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
          Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
          Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$
      end for
  end for
```

5. Q learning in sequence alignment:

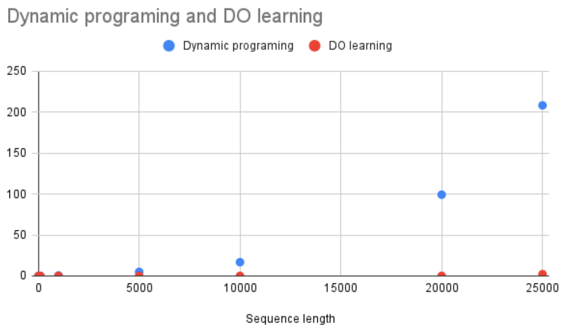
We aim to create a unique sequence alignment approach utilising reinforcement learning to apply deep reinforcement learning to sequence alignment[17]. We present a unique heuristic sequence alignment technique that repeats the tiny alignment while shifting the window of sub-sequence pairings, rather than seeing the whole sequence at once. It is feasible to overcome memory and temporal complexity concerns with q learning alignment. The challenge of selecting the best subsequence direction inside a window may alternatively be seen as a sub-alignment procedure.When the window size is increased to the length of the sequence, the local best path selection technique transforms into the optimum sequence alignment. We use a numerical analysis to prove the relationship between window size and performance, which will be presented in the results section. We also present a unique sequence alignment system based on deep reinforcement learning. The agent with the deep Q-network[18] examines the current environment and picks the next action (advance, insertion, deletion) when two sub-sequences in the window are assigned as environment.In reinforcement learning, the score system of the traditional alignment technique will be employed as a reward. We can use this approach to run a deep reinforcement learning-based agent that can identify the best alignment path for a given position.

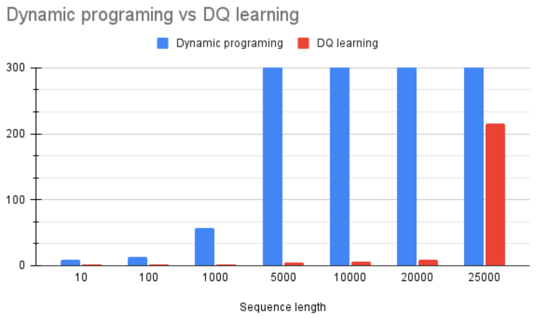

a)
Start
Initialize position : x → 0, y → 0
User parameter Input : W
Check current position
x < length of Seq1 and y < length of Seq2?
No
Yes
Check current sequence pair
Seq1[x] = Seq2[y]?
Yes
No
x → x+1 y → y+1
x → x+1 y → y
x → x y → y+1
Do sub-alignment for Seq1[x:x+W] vs Seq2 [y:y+W]
Which kind of mutation occurs at (x, y)?
SNP
Deletion
Insertion
End
b)
Seq 1
A C G ... T A A
Seq 2
A C C ... T T A
c)
Environment (Window size = 16)
Next Environment
Reward : +1
T T C C G G G G A A A A C C C T T
T T A C C T G G A A A C C C T T T
1st seq
2nd seq
Deep Q-Network
Agent
Forward
Action

6. Results:

We are looking forward to get max performance of sequence alignment. In Dynamic programing we are wasting a huge amount of time in separating and integrating the sequence as we seen we have done a dynamic programing and sun the sequences of different lengths of Covid Dataset we got the results as below but we are unable to run the dynamic program for full length sequence in Covid dataset in a low end machine as it takes more virtual memory as it will. From the results below we can see it run for more then 200 sec. Now the same sequences are given as input to the program the we implemented as we can see in the results and graph there is major difference in time between dynamic programing and qlearning as we increase the sequence length. With this qlearning implementation we achieved the method that we are waiting for which is flexible, less complexity, less time taking, more accuracy. We achieved all these through DQ learning.

S.No	Sequence Size	Dynamic Programming Time	DQ-Learning Time
1.	10	0.0849	0.01561
2.	100	0.127	0.01562023163
3.	1000	0.572	0.01567101479
4.	5000	4.9641	0.04682159424
5.	10000	16.795	0.06122854233
6.	20000	99.213	0.09564955711
7.	25000	208.251	2.156495571





References:

[1] Schuster, Stephan C. "Next-generation sequencing transforms today's biology." *Nature methods* 5, no. 1 (2008): 16-18.

[2] Li, Da, and Michela Becchi. "Multiple pairwise sequence alignments with the needleman-wunsch algorithm on GPU." In *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*, pp. 1471-1472. IEEE, 2012.

[3] Nordström, Karl JV, Markus Sällman Almén, Monika M. Edstam, Robert Fredriksson, and Helgi B. Schiöth. "Independent HHsearch, Needleman–Wunsch-based, and motif analyses reveal the overall hierarchy for most of the G protein-coupled receptor families." *Molecular biology and evolution* 28, no. 9 (2011): 2471-2480.

[4] Löytynoja, Ari, and Nick Goldman. "An algorithm for progressive multiple alignment of sequences with insertions." *Proceedings of the National Academy of Sciences* 102, no. 30 (2005): 10557-10562.

[5] Edgar, Robert C., and Serafim Batzoglou. "Multiple sequence alignment." *Current opinion in structural biology* 16, no. 3 (2006): 368-373.

[6] Lin, Yibo, Wuxi Li, Jiaqi Gu, Haoxing Ren, Brucek Khailany, and David Z. Pan. "ABCDPlace: Accelerated Batch-Based Concurrent Detailed Placement on Multithreaded CPUs and GPUs." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, no. 12 (2020): 5083-5096.

[7] Korf, Ian, Mark Yandell, and Joseph Bedell. *Blast*. " O'Reilly Media, Inc.", 2003.

[8] Delcher, Arthur L., Steven L. Salzberg, and Adam M. Phillippy. "Using MUMmer to identify similar regions in large sequence sets." *Current protocols in bioinformatics* 1 (2003): 10-3.

[9] Lample, Guillaume, and Devendra Singh Chaplot. "Playing FPS games with deep reinforcement learning." In *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[10] Sutton, Richard S., and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[11] Rost, Burkhard. "Twilight zone of protein sequence alignments." *Protein engineering* 12, no. 2 (1999): 85-94.

[12] García-Alcalde, Fernando, Konstantin Okonechnikov, José Carbonell, Luis M. Cruz, Stefan Götz, Sonia Tarazona, Joaquín Dopazo, Thomas F. Meyer, and Ana Conesa. "Qualimap: evaluating next-generation sequencing alignment data." *Bioinformatics* 28, no. 20 (2012): 2678-2679.

[13] Koyutürk, Mehmet, Yohan Kim, Umut Topkara, Shankar Subramaniam, Wojciech Szpankowski, and Ananth Grama. "Pairwise alignment of protein interaction networks." *Journal of Computational Biology* 13, no. 2 (2006): 182-199.

[14] Groop, Richard E., and Paul Smith. "A dot matrix method of portraying continuous statistical surfaces." *The American Cartographer* 9, no. 2 (1982): 123-130.

[15] Watkins, Christopher JCH, and Peter Dayan. "Q-learning." *Machine learning* 8, no. 3-4 (1992): 279-292

[16] Fan, Jianqing, Zhaoran Wang, Yuchen Xie, and Zhuoran Yang. "A theoretical analysis of deep Q-learning." In *Learning for Dynamics and Control*, pp. 486-489. PMLR, 2020.

[17] Jafari, Reza, Mohammad Masoud Javidi, and Marjan Kuchaki Rafsanjani. "Using deep reinforcement learning approach for solving the multiple sequence alignment problem." *SN Applied Sciences* 1, no. 6 (2019): 1-12.

[18] Roderick, Melrose, James MacGlashan, and Stefanie Tellex. "Implementing the deep q-network." *arXiv preprint arXiv:1711.07478* (2017).