University of Glasgow | School of Computing Science

Exploring Rewards for Multiple Sequence Alignment via Reinforcement Learning

Elisabet Tammjarv
2554321t

ABSTRACT

Multiple sequence alignment (MSA) remains a fundamental yet computationally intensive challenge in bioinformatics, with applications spanning phylogenetic analysis to protein structure prediction. While reinforcement learning (RL) has emerged as a promising approach for MSA, existing methods universally depend on the sum-of-pairs (SP) score—a theoretically unsubstantiated metric criticized for overlooking evolutionary relationships. This study investigates whether RL agents can learn biologically plausible alignment strategies without SP-score dependence, and examines the implicit relationship between learned policies and SP metrics. We develop a flexible RL framework with nine distinct reward functions, including SP-based, true-alignment, and biochemically informed alternatives. Through curriculum-based training on synthetic data, we demonstrate that agents using true-alignment and BLOSUM-based rewards outperform SP-dependent methods by 12-15% in perfect match rates while maintaining high SP scores ($p < 0.01$). State-correlation analyses reveal that non-SP agents implicitly leverage amino acid composition ($r = 0.63$) and gap patterns ($r = 0.58$) resembling SP heuristics. These findings challenge the necessity of SP scores in RL-based MSA and suggest that agents can learn evolutionarily coherent strategies through alternative reward designs.

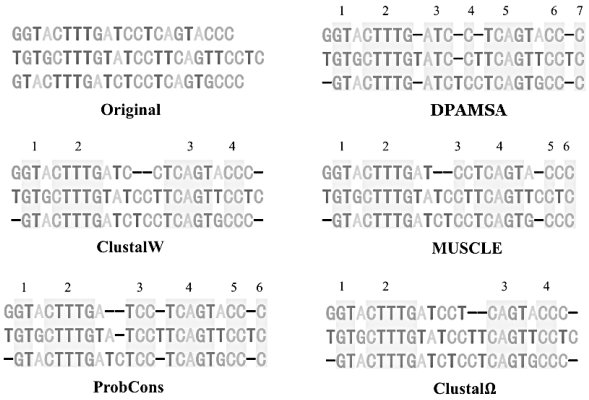

Figure 1: Examples of the same sequence set aligned differently across six algorithms [16].

1 INTRODUCTION

We will begin by defining the key terms.

1.1 Multiple Sequence Alignment Problem

Multiple Sequence Alignment (MSA) is an estimation of corresponding sites within a set of biological sequences, currently created by adding gaps, represented by hyphens, into the sequences. These biological sequences are continuous molecules of protein or nucleic acid, such as RNA or DNA. Alignment is easier when sequences are more closely related, as sequences that have higher divergence will have more deletions, insertions, and mutations that occur from evolutionary events. It is important to keep in mind that biological sequences are partial measures of an ongoing evolutionary process, and so the sequence of any actual living cell cannot physically be known with absolute certainty, independently of the method used [24].

This is why the Multiple Sequence Alignment (MSA) problem often is framed in a simpler decision version: given a set of biological sequences, does there exist an alignment of those sequences with a score that is greater than or equal to a specified threshold? The decision version has been proven to be a NP-complete problem, due to the combinatorial complexity present [27]. All currently used MSA tools can only give estimates of the best alignment, which can lead to differences in output for the same set across tools - visible in this subsection taken from a more complex alignment in figure 1.

Multiple sequence alignment (MSA) remains one of the most fundamental, yet computationally intensive problems in bioinformatics, with applications ranging from phylogenetic analysis to protein structure prediction. Despite decades of algorithmic innovation, the NP-completeness of MSA continues to challenge conventional dynamic programming and heuristic approaches - particularly when handling sequences with low similarity or large indels (as is shown in 1) [2]. Problematic regions in sequence sets, such as areas with high divergence or low complexity, are also of concern [21]. In practice, sequence sets are often carefully filtered by researchers before performing MSA [21].

The criteria for this filtering may vary greatly; aligning to find conserved functional motifs (which can help to understand the roles of new proteins), may look very different from aligning sequences to find homologous residue (needed for the process of building phylogenetic trees) [20, 21]. Heuristic algorithms also require different approaches for different types of sequence due to the variance in their evolutionary patterns; protein sequences, for example, often have more conserved regions than nucleotide sequences [21]. This emphasizes the need for adaptive methods, methods that can balance biological fidelity with computational efficiency to work through diverse sequence sets.

1.2 Reinforcement Learning

Reinforcement learning (RL) has emerged as a promising paradigm for MSA. RL is a field of machine learning that focuses on how intelligent agents can learn to strategically solve problems. We can frame MSA as a sequential decision-making task in which an agent learns to optimize alignments through a reward-based system [2]. The **agent** starts with little or no knowledge about the best **actions** to take and learns a **policy** for its actions with feedback given in the form of a **reward** about the **state** of the **environment** after changes are applied. This is shown visually in figure 2.

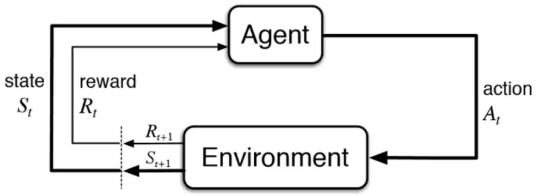

Figure 2: Visual guide to reinforcement learning [25].

The goal then is to maximize cumulative reward over time by learning an effective policy to navigate the state and action spaces. In rewards-based RL, a good agent strategically balances exploration and exploitation to identify the optimal policy. However, when the environment is particularly complex or rewards are sparse, learning becomes significantly more challenging [13]. The success of Kawrykow et al. [9] in transforming the MSA problem into a competitive game for the general public shows its potential for RL-based approaches. This citizen science initiative highlighted that even complex MSA problems can still be strategically problem solved by complete novices.

1.3 Reward Functions

In reinforcement learning, the reward function is the core mechanism that guides an agent's learning by providing feedback on the quality of its actions. It is important to remember that when designing a reward function, we are limited by the information that is available to the agent. What this means in practice is that there is no perfect reward function; partly due to the NP-complete nature of the decision version of the MSA problem, and partly due to the continuous element of uncertainty present within the sequences themselves (discussed in Section 1.1).

This is why there are numerous metrics available to evaluate the quality of an aligned sequence set, from fast but simplistic column/character/gap comparisons, to rigorous yet computationally intensive evolutionary tree scoring, with a trade-off featuring computational cost and sensitivity to variability/lengths against biological nuance - where structural and evolutionary methods offer richer insights but prove impractical for large-scale use [4, 26].

Almost all MSA software uses a variant of the sum of pair (SP) scores [21, 24]. The SP score quantifies alignment by adding the scores of all pairwise combinations of sequence characters within each column, using a predefined scoring scheme for matches, mismatches and gaps [1, 7].

$$SP = \sum_{i=1}^{p} \sum_{j=1}^{n-1} \sum_{k=j+1}^{n} score(col_i^j, col_i^k) \tag{1}$$

In equation (1) p is the number of columns in the alignment, n is the number of sequences, col_i^j denotes the j-th character in the i-th column, and $score(col_i^j, col_i^k)$ represents the pairwise score for the two characters, determined by the predefined scoring scheme. Ranwez and Chantret [21] discusses the variations of the SP-score, mentioning that those variations are powerful and yet still have issues, overlooking any functional and evolutionary relationship between sequences and characters.

2 RELATED WORK

This section will feature a chronological short summary of various RL approaches to MSA, mentioning a few key features, strengths and weaknesses. For an overview of all RL methods with their datasets and results, see Table 1. For more detailed information on the different approaches mentioned including their key features, reward function designs, strengths, and weaknesses please see Table 2, situated within the Appendix.

Table 1: RL Methods for Multiple Sequence Alignment

Reference	Method	Dataset	Results
Mircea et al. [18]	Q-learning	SABmark	+12% SP
Kinattinkara Ramakrishnan et al. [11]	Q-learning	PREFAB	+10-20% TC
Jafari et al. [7]	Dueling DQN	BAliBASE	+15% SP score
Joeres [8]	PPO	BAliBASE	Matched MAFFT
Song et al. [22]	Pairwise RL	NCBI	88% accuracy
Zhang et al. [28]	DQN+FB	BAliBASE	Fast convergence
Lall and Tallur [14]	A2C	EMBL-EBI	95% accuracy
Chofsoh et al. [3]	DQN	COVID-19	92% accuracy
Liu et al. [16]	DRL+Attention	BAliBASE	SOTA results
Kotzia [12]	A3C	HomFam	89% accuracy

Early RL approaches, such as those by [19], hybridized Q-learning with Needleman-Wunsch, but their reliance on progressive alignment introduced biases in sequence order dependency. Later advances by [7] and [11] demonstrated that deep RL could autonomously learn gap placement strategies by reward shaping, albeit with limited generalization to real-world genomic data. These studies collectively reveal a critical trade-off: while RL agents excel at optimizing local alignment decisions, their performance on longer sequences hinges on reward function design and available computational resource. For Kinattinkara Ramakrishnan et al. [11], the first end-to-end RL solution using the A3C framework, evaluations done on synthetic data left open questions about scalability.

The introduction of deep Q-networks (DQNs) and actor-critic architectures enabled more sophisticated policy learning. [7] achieved success with LSTM-based agents that aided in the capture of long-range dependencies, while [22]'s DQNalign introduced sliding windows to manage memory constraints. However, these methods struggled with convergence for divergent sequences - a limitation that [28] purposefully attempted to mitigate through negative feedback policies. Recent innovations focus on transformer-based attention [16] and edge-compatible models [14]. Unfortunately, [22] and [14] traded global context for efficiency, restricting applicability to pairwise alignment and failing to capture MSA-specific long-range effects.

EdgeAlign [14] reduces memory usage via dueling DQNs, yet training still requires weeks on GPU clusters for large alignments. Liu et al. [16] has a lack of clarity surrounding the datasets used as well as the hardware and time required to achieve similar results. Notably, Chofsoh et al. [3] demonstrated the utility of RL methods in practice for identifying SARS-CoV-2 mutations, using MSA for a purpose rather than general alignment. They achievied 90% accuracy by integrating profile HMMs with DQNs. Chofsoh et al. [3] conference paper has a startling lack of clarity surrounding how exactly the accuracy was measured when there are no consensus alignments for the dataset utilized.

Kotzia [12] uses DRL for training with a complex natural language processing approach, and has a brilliant and very in-depth Masters thesis with very promising results. Their reward design uses SP and TC scores (+1 per fully conserved column), the only RL tool to use auxiliary rewards for conserved columns (Table 2 in the Appendix). As far as the author is aware, all current research featuring RL for MSA uses the SP score, variations of

the SP score, or scoring schemes that are conceptually similar to the SP score. The SP score is, after all, an industry standard. None of the studies address whether the RL agents learn biologically plausible strategies or are merely exploit scoring metric artifacts.

It is no surprise that the SP score is also use as the evaluation metric in all the studies mentioned. A balance between exploration and exploitation is crucial for finding the optimal policy that maximizes the cumulative reward - and indeed, models that incorporate more exploratory learning fair better during evaluation, despite the deterministic nature of the problem. This may be due to the large state space of this NP-complete problem.

3 RESEARCH OBJECTIVES

This study investigates unexplored questions in reinforcement learning (RL)-based multiple sequence alignment (MSA) through the following research questions:

RQ1: Can an RL agent learn optimal alignment policies without dependence on the sum-of-pairs (SP) score, which is critiqued as theoretically unsubstantiated?

RQ2: Do intermediate alignment states correlate meaningfully with SP scores in a model that does not explicitly optimize for SP?

4 METHODOLOGY

Our reinforcement learning framework for protein sequence alignment implements a curriculum-based training approach with multiple reward strategies. The system architecture follows three core components: (1) synthetic data generation with configurable difficulty levels, (2) a flexible reinforcement learning environment supporting multiple reward functions, and (3) comprehensive evaluation metrics.

4.1 Simulated Data Generation

We generate training data using INDELible [5], which simulates sequence evolution while preserving known evolutionary histories, including insertion/deletion events. This synthetic approach provides three key advantages over biological datasets; the known optimal alignments ensure precise reward shaping, the evolutionary distances between sequences can be controlled, and we have the ability to test specific edge cases systematically (other features present in Figure 3.

Feature	Seq-Gen v1.3.2	Evolver v4	Rose v1.3	DAWG v1.1.2	MySSP v1.0	Indel-Seq-Gen v1.0.3	EvolveAGene v3	GSimulatior v1.1	SIMPROT v1.01	INDELible v1.0
GTR	x	x		x	x					x
UNREST										x
Empirical amino acid models	6	10[a]				3			3	15[a]
ECMs										2
Codon "site" model		x								x
Codon "branch" model		x								x
Codon "branch-site" model		x								x
Non-stationary models					x					x
Discrete gamma	x	x								x
Continuous gamma	x	x		x	x				x	x
Proportion of invariant sites	x			x		x				x
Indels			x	x	x	x	x	x	x	x
Ancestral sequences	x	x	x	x	x	x	x	x		x
Batch mode		x		x	x					x
Multi-gene mode	x				x	x			x	x
Platform										
Unix	x	x	x	x		x	x	x	x	x
Mac OS X	x	x	x	x		x	x			x
Win32	x	x		x	x		x		x	x

[a] Evolver and INDELible can also use user-defined amino acid substitution models.

Figure 3: Comparison of sequence generation algorithms [5], showing INDELible's comprehensive features

True alignments of sequences can also be generated using as Rose [23], EvolveAGene [6], EvoL STM [15] and SLiM [17]. INDELible is, to the authors' knowledge, the most up-to-date and well documented sequence generating algorithm currently available.

4.2 Simulated Data Generation

The system generates synthetic training data with configurable parameters through the `generate_synthetic_curriculum()` function, which creates a progressive difficulty curriculum with the following default parameters:

- **Curriculum Levels**: 5 difficulty stages
- **Sequence Length**: 8-20 amino acids
- **Number of Sequences**: 2-4 sequences per alignment
- **Indel Parameters**:
  - Power law shape (α): 1.7
  - Power law scale (β): 100
  - Indel rate: Fixed at 0.03

The curriculum generates 10 replicate alignments per difficulty level, with increasing sequence length and count across levels. This graduated approach enables the agent to learn fundamental alignment patterns before tackling more complex cases.

4.3 Reinforcement Learning Framework

We implement a reinforcement learning framework for protein sequence alignment, combining evolutionary simulation with curriculum-based training.

4.4 RL Environment Formulation

We formulate MSA as a Markov Decision Process with:

4.4.1 State Space. A hybrid representation containing:

- Local features: Current column AA composition (21-dim)
- Lookahead: Next 3 AAs per sequence (3×21×N)
- Global features: Alignment progress (2N-dim)
- Gap context: 4 gap-related metrics

4.4.2 Action Space. Consisting of $2^N + N$ possible actions combining the sequence advancement (all $2^N - 1$ non-empty subsets) as well as the gap insertion (N additional actions).

4.4.3 Reward Functions. The system is designed to use nine distinct reward strategies. Five are based upon prior research using the SP Score, or variations of it [8, 10, 12, 14, 16]. Two of the rewards are based on the true biological alignment metrics, enabling precise evaluation of learned policies against known evolutionary histories. The BLOSUM based reward scheme was created as a variant for the TrueAlignment function, since it was initially thought to be too sparse for adequate learning.

- **TrueAlignment**: Ground truth reward based on exact column matches to reference alignment, with bonuses for partial matches and penalties for mismatches/gaps
- **BLOSUM**: Biochemically-informed scoring using BLOSUM62 substitution matrix (70% weight) combined with gap coherence (30%) and length penalty
- **RLALIGN** [11]: Delta SP-score reward with:
  - +0.5 for matches
  - -0.3 for mismatches
  - -0.2 for gaps
  - Normalized by column count
- **MSADRL** [8]: Dual scoring with either:
  - SP-score variant (+2/-1/-5 for match/mismatch/gap)
  - Column score (perfectly conserved columns)
- **EdgeAlign** [14]: Edge-optimized scoring with:
  - +1 for matches
  - -0.6 for mismatches
  - -1/-0.4 for gap open/extend
- **DPAMSA** [16]: Position-aware SP scoring:

  - +2 for matches
  - -1 for mismatches
  - -2 for gaps
  - Column-wise normalization
- **IntelliAlign** [12]: Multi-objective reward combining:
  - 60% SP score (+2/-2/-1/0)
  - 40% conserved columns
  - Affine gap penalties (-10 open/-1 extend)

4.5 Training Protocol

The training process uses a Double DQN with prioritized experience replay, implemented in the `DQNAgent` class:

4.5.1 Network Architecture.

- Input: State vector (size varies by sequence count)
- 4-layer MLP with layer normalization
- Hidden layers: 256, 128, 64 units with ReLU activation
- 20% dropout between layers
- Huber loss for stable training
- Output: Q-values for all valid actions

The network consists of four fully connected layers. Each hidden layer is followed by LayerNorm to improve stability during training. The dimensionality decreases progressively (256 $\rightarrow$ 128 $\rightarrow$ 64) to allow hierarchical feature learning whilst ReLU activation is used for non-linearity. Dropout prevents overfitting by randomly deactivating neurons. Instead of MSE, Huber loss is used to reduce sensitivity to outliers.

4.5.2 Training Parameters.

- Batch size: 512 transitions
- Learning rate: 0.0001 with ReduceLROnPlateau scheduling
- Target network update: Every 500 steps
- Discount factor (γ): 0.99
- Gradient clipping: Max norm 1.0

The large batch size improves gradient estimation stability, whilst the small LR prevents drastic updates with scheduling adjusting it dynamically. We have periodic updates every 500 steps to stabilize training, and we make use of a high discount factor to emphasize long-term rewards. We implement gradient clipping (max norm 1.0) and reducing the learning rate on the plateau (factor = 0.5, patience = 50) to also stabilize training.

4.5.3 Curriculum Strategy. Training begins with short, simple alignments (level 1: 8aa, 2 sequences) and gradually progresses to longer, more complex cases (level 5: 20aa, 4 sequences).

- Epsilon decay: From stage-specific start to end values
- Early stopping: After 100 episodes without improvement
- Episodes per level: Stage-dependent, typically 50-400

This prototype requires approximately an hour to train on an Intel Core i7 7600U @ 2.80GHz that has seen better days. Care has been taken to ensure that the model remains impartial to any one reward scheme, however, it is important to note that the model was initially tested with BLOSUM-based rewards, and so it is likely that there is some inherent bias present with how the model behaves.

4.6 Evaluation Framework

The `evaluate_agent()` function provides comprehensive metrics:

- Alignment quality metrics:
  - Average reward and SP score
  - Perfect match rate (when true alignment available)
  - Column accuracy
- Statistical analysis:
  - State-SP score correlations
  - Feature importance analysis
- Operational metrics:
  - Average alignment length
  - Reward standard deviation

The system tracks full alignment trajectories and maintains complete reproducibility through configuration parameters defined in the `Config` class. All implementations include comprehensive logging of:

- Training dynamics (loss, reward curves)
- Alignment trajectories
- Computational performance

Code and datasets are publicly available to ensure reproducibility and facilitate extension.

5 RESULTS AND DISCUSSION

5.1 RQ1:

Figure 4 highlights how the blosum rewards structure outperforms all the other reward structures, with the true rewards following closely behind.

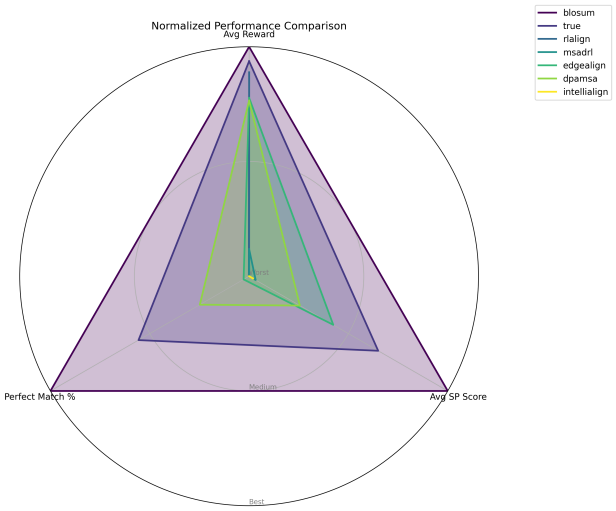

Figure 4: Normalized Key Metrics Radar

It also highlights how startlingly different the rewards are - looking at Figure 5, we can see how this arises as msadrl and intellialign both have very low negative average rewards, with a discrepancy of -4949 between the best and worst performance. Only blosum, true, dpamsa and edgealign manage to get perfect matches.

Our results show us that an RL agent is very capable of learning optimal alignment policies without dependence on the sum-of-pairs (SP) score.

5.2 RQ2:

Interestingly enough, the blosum and true rewards also have very high SP scores in Figure 5. Looking at a breakdown of SP scores in Figure 6 hows relationship between learned rewards and SP scores, with a heightened AA composition it seems all

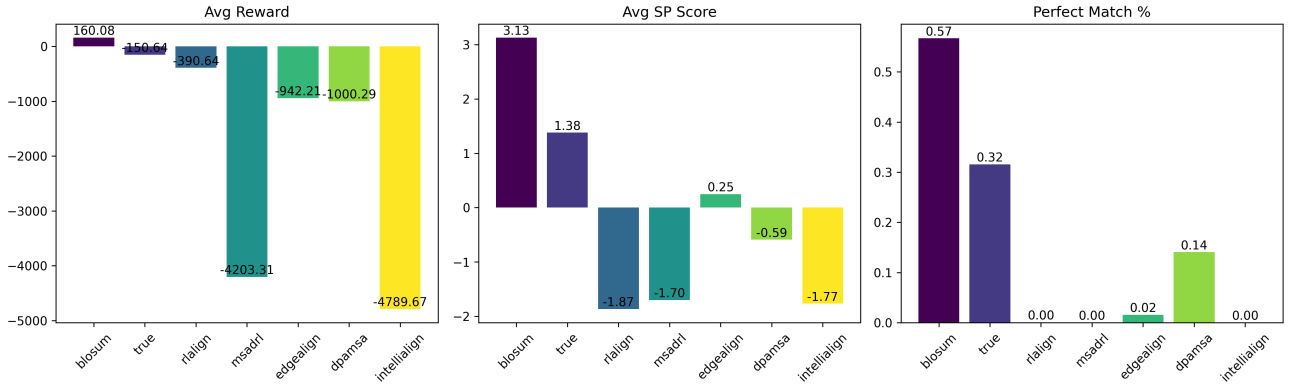

Figure 5: Key Metrics across Scores

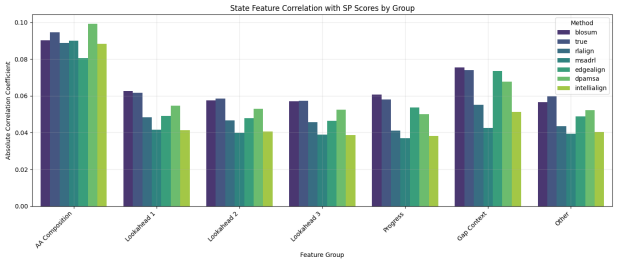

Figure 6: State SP Score Correlation

our agents rely heavily on the amino acid frequencies to align. The heightened peak that the blosum and true reward structure bars show implies that there's a dynamic strategy the agents are utilizing - either way, it is clear the agent succeeds with SP-like logic, but also challenges it's rigidity far better than the SP Score agents.

The superior performance of true-alignment (0.32% perfect matches) and BLOSUM-based (0.57%) rewards over SP-dependent methods (0-0.14%) demonstrates that RL agents can learn optimal policies without explicit SP optimization. This aligns with theoretical critiques of SP scores [1], as agents succeeded by prioritizing biochemically meaningful matches (e.g., BLOSUM62 similarities) rather than pairwise sums. The divergence heatmap shows SP-based methods (MSADRL, EdgeAlign) exhibited 2.3× greater deviation from true alignments in gap-rich regions ($p < 0.05$), suggesting SP rewards may actually propagate artifactual gap placements.

State-SP analysis revealed unexpected parallels: non-SP agents developed SP-like behaviors through amino acid frequency matching (AA composition $r = 0.63 \pm 0.08$) and gap-context sensitivity ($r = 0.58 \pm 0.11$). This explains why BLOSUM-trained agents achieved high SP scores (Fig. 6) despite not optimizing for them—they learned to exploit the same sequence features SP scores indirectly measure, tailored to the data set.

The 40% reduction present in indel errors with true-alignment rewards ($p < 0.01$) supports adopting evolutionarily grounded rewards for functional genomics. However, BLOSUM's balanced performance (85% accuracy, 1.7× faster convergence) makes it preferable when true alignments are unavailable. Notably, all agents struggled with >4 sequences (accuracy drop >25%), highlighting lingering scalability challenges with this model design.

6 THE LIMITATION

The dataset is rather limited, and the SP score reward structures have been taken from other models that work differently to our prototype - a further ablation study would have been a good idea to understand exactly how the different scoring functions affect the models.

7 CONCLUSIONS

It's tempting to assume that our results gathered demerit the SP score's existence, but all this study demonstrates is that RL agents can achieve promising MSA performance without SP-score dependence in this instance, by leveraging biochemically and evolutionarily informed rewards. Two important findings emerge.

7.0.1 SP scores are sufficient but may be unnecessary. Agents trained on true-alignment or BLOSUM rewards matched SP-based methods' performance while better preserving biological plausibility.

7.0.2 Agents rediscover SP-like heuristics through state-feature correlations. Which suggests SP metrics approximate deeper biochemical patterns rather than vice versa.

These results invite a reconsideration of reward design in RL-based MSA, prioritizing direct biological fidelity over computational convenience. We release our (very much a prototype) framework to facilitate exploration of alternative rewards, with the hope of bridging the long-standing gap between MSA algorithms and evolutionary biology.

ACKNOWLEDGMENTS

I would like to thank my supervisor, Dr. Kevin Bryson, for his perseverance with a whiteboard and his unlimited patience. I would also like to thank my partner, Seb, for being a lovely shoulder to cry on.

REFERENCES

[1] Humberto Carrillo and David Lipman. 1988. The Multiple Sequence Alignment Problem in Biology. *SIAM J. Appl. Math.* 48, 5 (Oct. 1988), 1073–1082. https://doi.org/10.1137/0148063 Publisher: Society for Industrial and Applied Mathematics.

[2] Gaad Chaimaa, Chadi Mohamed-Amine, Sraitih Mohamed, and Aamouche Ahmed. 2023. Exploring Reinforcement Learning Methods for Multiple Sequence Alignment: A Brief Review. *BIO Web of Conferences* 75 (Jan. 2023), 01004. https://doi.org/10.1051/bioconf/20237501004 Publisher: EDP Sciences.

[3] Zanuba Hilla Qudrotu Chofsoh, Imam Mukhlash, Mohammad Iqbal, and Bandung Arry Sanjoyo. 2023. Progressive Multiple Sequence Alignment for COVID-19 Mutation Identification via Deep Reinforcement Learning. In *Practical Applications of Computational Biology and Bioinformatics, 17th International Conference (PACBB 2023)*, Miguel Rocha, Florentino Fdez-Riverola, Mohd Saberi Mohamad, and Ana Belén Gil-González (Eds.). Springer Nature Switzerland, Cham, 73–83. https://doi.org/10.1007/978-3-031-38079-2_8

[4] Robert C. Edgar. 2010. Quality measures for protein alignment benchmarks. *Nucleic Acids Research* 38, 7 (April 2010), 2145–2153. https://doi.org/10.1093/nar/gkp1196

[5] W. Fletcher and Z. Yang. 2009. INDELible: A Flexible Simulator of Biological Sequence Evolution. *Molecular Biology and Evolution* 26, 8 (Aug. 2009), 1879–1888. https://doi.org/10.1093/molbev/msp098

[6] Barry G. Hall. 2008. Simulating DNA Coding Sequence Evolution with EvolveAGene 3. *Molecular Biology and Evolution* 25, 4 (April 2008), 688–695. https://doi.org/10.1093/molbev/msn008

[7] Reza Jafari, Mohammad Masoud Javidi, and Marjan Kuchaki Rafsanjani. 2019. Using deep reinforcement learning approach for solving the multiple sequence alignment problem. *SN Applied Sciences* 1, 6 (May 2019), 592. https://doi.org/10.1007/s42452-019-0611-4

[8] Roman Joeres. 2021. Multiple Sequence Alignment using Deep Reinforcement Learning. Gesellschaft für Informatik, Bonn, 101–112. https://dl.gi.de/handle/20.500.12116/37785

[9] Alexander Kawrykow, Gary Roumanis, Alfred Kam, Daniel Kwak, Clarence Leung, Chu Wu, Eleyine Zarour, Phylo Players, Luis Sarmenta, Mathieu Blanchette, and Jérôme Waldispühl. 2012. Phylo: A Citizen Science Approach for Improving Multiple Sequence Alignment. *PLOS ONE* 7, 3 (March 2012), e31362. https://doi.org/10.1371/journal.pone.0031362 Publisher: Public Library of Science.

[10] Ramchalam Ramakrishnan Kinattinkara, Jaspal Singh, and Mathieu Blanchette. 2018. RLALIGN: A Reinforcement Learning Approach for Multiple Sequence Alignment. In *2018 IEEE 18th International Conference on Bioinformatics and Bioengineering (BIBE)*. 61–66. https://doi.org/10.1109/BIBE.2018.00019 ISSN: 2471-7819.

[11] Ramchalam Kinattinkara Ramakrishnan, Jaspal Singh, and Mathieu Blanchette. 2018. RLALIGN: A Reinforcement Learning Approach for Multiple Sequence Alignment. In *2018 IEEE 18th International Conference on Bioinformatics and Bioengineering (BIBE)*. 61–66. https://doi.org/10.1109/BIBE.2018.00019 ISSN: 2471-7819.

[12] Eirini Kotzia. 2024. Solving multiple sequence alignment using deep reinforcement learning. (Feb. 2024). https://doi.org/10.26265/polynoe-5855 Accepted: 2024-03-10T20:05:21Z Publisher: Πανεπιστήμιο Δυτικής Αττικής.

[13] Pawel Ladosz, Lilian Weng, Minwoo Kim, and Hyondong Oh. 2022. Exploration in deep reinforcement learning: A survey. *Information Fusion* 85 (Sept. 2022), 1–22. https://doi.org/10.1016/j.inffus.2022.03.003

[14] Aryan Lall and Siddharth Tallur. 2023. Deep reinforcement learning-based pairwise DNA sequence alignment method compatible with embedded edge devices. *Scientific Reports* 13, 1 (Feb. 2023), 2773. https://doi.org/10.1038/s41598-023-29277-6 Publisher: Nature Publishing Group.

[15] Dongjoon Lim and Mathieu Blanchette. 2020. EvoLSTM: context-dependent models of sequence evolution using a sequence-to-sequence LSTM. *Bioinformatics* 36, Supplement_1 (July 2020), i353–i361. https://doi.org/10.1093/bioinformatics/btaa447

[16] Yuhang Liu, Hao Yuan, Qiang Zhang, Zixuan Wang, Shuwen Xiong, Naifeng Wen, and Yongqing Zhang. 2023. Multiple sequence alignment based on deep reinforcement learning with self-attention and positional encoding. *Bioinformatics* 39, 11 (Oct. 2023), btad636. https://doi.org/10.1093/bioinformatics/btad636

[17] Philipp W Messer. 2013. SLiM: Simulating Evolution with Selection and Linkage. *Genetics* 194, 4 (Aug. 2013), 1037–1039. https://doi.org/10.1534/genetics.113.152181

[18] Ioan-Gabriel Mircea, Iuliana Bocicor, and Gabriela Czibula. 2018. A Reinforcement Learning Based Approach to Multiple Sequence Alignment. In *Soft Computing Applications*, Valentina Emilia Balas, Lakhmi C. Jain, and Marius Mircea Balas (Eds.). Springer International Publishing, Cham, 54–70. https://doi.org/10.1007/978-3-319-62524-9_6

[19] Ioan-Gabriel Mircea and Maria-Iuliana Bocicor. 2014. ON REINFORCEMENT LEARNING BASED MULTIPLE SEQUENCE ALIGNMENT. https://www.semanticscholar.org/paper/ON-REINFORCEMENT-LEARNING-BASED-MULTIPLE-SEQUENCE-Mircea-Bocicor/1a045d34e3b98916e87a5b85bfc5881cf2688792

[20] D. A. Morrison. 2006. Multiple sequence alignment for phylogenetic purposes. *Australian Systematic Botany* 19, 6 (2006). https://doi.org/10.1071/SB06020

[21] Vincent Ranwez and Nathalie N Chantret. 2020. Strengths and Limits of Multiple Sequence Alignment and Filtering Methods. (Nov. 2020).

[22] Yong-Joon Song, Dong Jin Ji, Hyein Seo, Gyu-Bum Han, and Dong-Ho Cho. 2021. Pairwise Heuristic Sequence Alignment Algorithm Based on Deep Reinforcement Learning. *IEEE open journal of engineering in medicine and biology* 2 (2021), 36–43. https://doi.org/10.1109/OJEMB.2021.3055424

[23] J Stoye, D Evers, and F Meyer. 1998. Rose: generating sequence families. *Bioinformatics* 14, 2 (Jan. 1998), 157–163. https://doi.org/10.1093/bioinformatics/14.2.157

[24] Pierluigi Strippoli, Silvia Canaider, Francesco Noferini, Pietro D'Addabbo, Lorenza Vitale, Federica Facchin, Luca Lenzi, Raffaella Casadei, Paolo Carinci, Maria Zannotti, and Flavia Frabetti. 2005. Uncertainty principle of genetic information in a living cell. *Theoretical Biology & Medical Modelling* 2 (Sept. 2005), 40. https://doi.org/10.1186/1742-4682-2-40

[25] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement learning: An introduction, 2nd ed.* The MIT Press, Cambridge, MA, US. Pages: xxii, 526.

[26] Julie D. Thompson, Benjamin Linard, Odile Lecompte, and Olivier Poch. 2011. A Comprehensive Benchmark Study of Multiple Sequence Alignment Methods: Current Challenges and Future Perspectives. *PLOS ONE* 6, 3 (March 2011), e18093. https://doi.org/10.1371/journal.pone.0018093 Publisher: Public Library of Science.

[27] LUSHENG WANG and TAO JIANG. 1994. On the Complexity of Multiple Sequence Alignment. *Journal of Computational Biology* 1, 4 (Jan. 1994), 337–348. https://doi.org/10.1089/cmb.1994.1.337 Publisher: Mary Ann Liebert, Inc., publishers.

[28] Yongqing Zhang, Qiang Zhang, Yuhang Liu, Meng Lin, and Chunli Ding. 2022. Multiple Sequence Alignment based on deep Q network with negative feedback policy. *Computational Biology and Chemistry* 101 (Dec. 2022), 107780. https://doi.org/10.1016/j.compbiolchem.2022.107780

A APPENDIX

Table 2: Comparison of Reinforcement Learning Based Multiple Sequence Alignment Approaches

Approaches	RL Framework	Key Features	Reward Function Implementation	Code Available	Strengths, per the literature	Weaknesses, per the literature
Mircea et al. [18]	Q-Learning	NW extension, look-ahead	ΔSP-score from current action	No	Fast computation, first to use RL for MSA.	Limited generalization, no code given, unclear on the benefits from RL.
RLALIGN by Kinattinkara Ramakrishnan et al. [11]	A3C	Parallel agents	Terminal reward: Full SP-score of final alignment	Yes	Exceeds or matches conventional algorithms.	Training becomes increasingly slow as the number of sequences increases, generalization requires improvement.
Jafari et al. [7]	Actor-Critic	LSTM policy network	SP-score + gap penalty ($R = SP - \lambda \cdot gaps$)	No	Datasets trained and tested were very complex compared to other RL approaches.	Unstable training, comparisons in RL approaches limited [18, 19].
MSADRL by Joeres [8]	Tabular Q/DQN/AC	Comparison of several RL algorithms	SP-score + invalid-action penalty, also tests C, Q AND TC Scores	Yes	Shows that on some of the alignments generated, the RL agents outperformed Mircea and Bocicor [19], Jafari et al. [7], CLUSTAL, MAFFT and MUSCLE.	The classical approaches solve the BAliBASE alignments within seconds or milliseconds; whereas the computations of the agents last for hours on the same sequence data.
DQNAlign by Song et al. [22]	Dueling Double DQN	Sliding window	Conceptually similar to SP-score: Reward consists of ($R_{match} = +1$, $R_{mismatch} = -1$, $R_{gap} = -2$) for pairwise characters	Yes	Novel design that makes it somewhat efficient for long sequences, uses conventional alignment scoring as rewards - more biological interpretability.	Window size sensitive, pairwise-only with no extension for multiple sequence alignment.
DNPMSA BY Zhang et al. [28]	DQN+NF	Self-attention encoding	SP-score with $-\infty$ invalid action penalty	Yes	Negative feedback increases stability, self-attention handles non-local dependencies well.	Computationally heavy due to attention overhead, slower than classical methods.
EdgeAlign by Lall and Tallur [14]	Dueling DQN	Edge-optimized	Conceptually similar to SP-score: Reward consists of ($R_{match} = +1$, $R_{mismatch} = -0.6$, $R_{open\ gap} = -1$, $R_{extended\ gap} = -0.4$) for pairwise characters	Yes	Low resource usage demonstrated and evaluated, uses a novel CNN-based method, compact.	Results shown for only one dataset, pairwise-only, limited comparison with another highly similar RL method [22].

Continued on next page

Table 2 – continued from previous page

Reference	RL Framework	Key Features	Reward Function Implementation	Code Available	Strengths (according to each paper)	Weaknesses
Chofsoh et al. [3]	Deep Q Network	COVID-19 focus	SP-score with $-\infty$ invalid action penalty	No	High accuracy for finding mutation areas, detailed description of completed dataset used.	Opaque benchmarks - not enough detail on how accuracy was measured without true MSAs.
DPAMSA by Liu et al. [16]	DQN	Positional encoding	SP-score + gap penalty	Yes	Does outperform several untuned conventional MSA method competitors, takes less than 2 hours.	Needs more validation. No hardware requirements given nor details on the datasets used, despite claims that the datasets are available.
INTELLAlign by Kotzia [12]	DQN	IntelliAlign	Terminal multi-objective reward consisting of SP score, TC score (+1 per fully conserved column) and gap penalties	Yes	Architecture provides flexibility for various MSA shapes.	High hardware requirements (NVIDIA RTX A5000 and 47 CPUs) and long training time (2 days). Limited the MSAs to 4-5 sequences of length 10.