Modeling artifacts in sequence alignment

Juan J. García Mesa

Abstract

Unpublished reference genomes tend to have artifacts that, if not corrected, can impact downstream analysis including phylogenetic inference, ancestral sequence reconstruction, and gene annotation. Within coding sequences, common artifacts include abiological frameshifts and early stop codons. While these are eventually fixed for model organisms, for many species this is not the case, requiring curation efforts that discard large amounts of data. Current aligners of coding sequences depend primarily on amino acid translations, generally only support in-frame indels that occur between codons, and are not robust to artifacts. Here I discuss the development of a new statistical sequence alignment software that will be robust to artifacts in unpolished genomes, incorporate codon models, and support complex indels.

1 Background

Advancements in sequencing technology and the increasing affordability of new equipment has generated an overflow of genomic information. The abundance of processed data today is orders of magnitude larger than two decades ago. Unfortunately, the available deluge of genomic data is not free of artifacts. Uncorrected errors in genomic datasets can lead to erroneous results in functional and comparative genomic studies (Schneider *et al.* 2009). This requires costly curation practices that discard large amounts of information.

Sequence alignment is considered a fundamental task in bioinformatics and a cornerstone step in comparative and functional genomic studies (Rosenberg 2009). Modern sequence analysis began with the heuristic homology algorithms of Needleman and Wunsch in 1970 (Smith, Waterman, *et al.* 1981) and has progressed to arrive at current aligners such as BAli-Phy (Suchard & Redelings 2006), CLUSTALΩ (Sievers *et al.* 2011), MAFFT (Katoh *et al.* 2002), MACSE (Ranwez *et al.* 2011), PRANK (Löytynoja 2014). However, the alignment of molecular sequences is, in practice, often seen as a tool and the alignment inference as an ad hoc problem (Morrison 2018).

A common strategy when aligning sequences is a three step approach that translates DNA sequences to amino acids, performs alignment inference in the amino acid space, then finally back-translates the protein alignment to DNA (Bininda-Emonds, Olaf 2005; Abascal *et al.* 2010). While this approach is an improvement over DNA models, it discards information, fails in the presence of artifacts, and underperforms compared to alignment at the codon level. Although some aligners incorporate codon substitution models (e.g. BAli-Phy, PRANK), they do not support frameshifts. While indels are rarely modeled to appear within codons, it has been estimated that this is often the case (Zhu & Cartwright, personal communication, 2019). When gaps are only considered to appear between codons, the optimal alignment can be missed (Fig. 1).

While frameshifts are common in coding-sequence datasets, these are expected to be errors due to strong purifying selection. Identifying canonical coding sequences to patch this issue is the most accessible solution and yet often unsuccessful. Improving the annotation quality or re-sequencing with higher quality involves high costs with little reward. Therefore, researchers are ill-equipped to deal with uncurated heterogeneous datasets. To address this need, I propose to develop COATi, a tool that will be able to generate sequence alignments while correcting for artifacts in a feature-rich and user-friendly software package.

2 Aims

Here I describe COATi (COdon-aware Alignment Transducer), a new statistical aligner that can handle artifacts in genomic datasets and employs robust models of molecular evolution.

	Lys	Ala	Leu	Leu
H:	AAG	GGC	CTC	TTG
G:	AAG	---	GTC	TTG
	Lys	-	Val	Leu

	Lys	Ala	Leu	Leu
H:	AAG	GGC	CTC	TTG
G:	AAG	G--	-TC	TTG
	Lys	Val	-	Leu

	Pro	Pro	Lys	Leu
H:	CCC	CCC	AAG	CTG
G:	CCC	CCG	---	CTG
	Pro	Pro	-	Leu

	Pro	Pro	Lys	Leu
H:	CCC	CCC	AAG	CTG
G:	CCC	CC-	--G	CTG
	Pro	Pro	-	Leu

Figure 1: Standard algorithms produce suboptimal alignments. Rows show possible alignments of gorilla (G) against human (H) sequences. The best alignment is highlighted in blue, and nucleotide mismatches are highlighted in red. Because standard algorithms do not support within codon indels, they miss the best alignment and inflate estimates of sequence divergence.

2.1 Aim 1 - Statistical Pairwise Alignment of Protein Coding Sequences

2.1.1 Pairwise hidden Markov models.

Statistical alignment is typically performed using pairwise hidden Markov models (pair-HMMs). Pair-HMMs are computational machines with two output tapes and a set of states that emit symbols onto one or both tapes. A path through a pair-HMM represents a possible alignment between the two sequences. Conceptually, these machines generate two sequences (X and Y) from an unknown common ancestor and calculate the probability that two sequences are related, represented $P(X,Y)$ (Yoon 2009). Pair-HMMs have the ability to rigorously model molecular sequence evolution and can find an optimal alignment, among other capabilities.

2.1.2 Finite state transducers (FSTs)

A limitation of pair-HMM is the ability to only model evolution of two related sequences from an unknown ancestor, thus not being able to use the output of one pair-HMM as the input of another. Finite-state transducers (FSTs) share similar computational characteristics as pair-HMMs and have an input and output tape, instead of two output tapes. FSTs absorb symbols from an input tape and emit symbols to an output tape. Conceptually, an FST generates a descendant sequence given an ancestral one $X \Rightarrow Y$. Properly weighted, an FST can calculate the conditional probability that sequence Y evolved from sequence X, represented $P(Y|X)$.

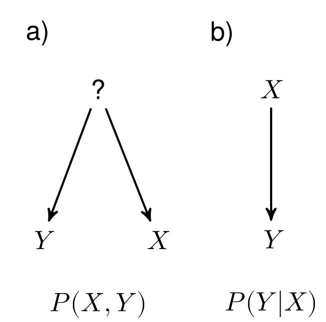

Figure 2: (a) Pair-HMMs can generate the probability that two sequences are related. (b) FSTs can generate the probability that sequence Y evolved from sequence X.

FSTs have similar benefits to pair-HMMs in addition to well established algorithms for combining them in different ways (Bradley & Holmes 2007). A powerful and versatile algorithm is composition, which consists of sending the output of one FST as the input of a second one. In the development of COATi I will design complex FSTs from smaller FSTs, each representing a specific process.

2.1.3 Evolution FST

The evolution FST is based on existing transducers (e.g. Holmes & Bruno 2001). This FST is formed by composing a substitution FST (Fig. 3-a) that models codon evolution and an indel FST (Fig. 3-b) that models insertions and deletions, including frameshifts. The power of this FST with respect to others is the combination of a codon substitution model with gaps that can occur at any position in any length.

Substitution model. Codon substitution models are uncommon in sequence aligners, despite their extensive use in phylogenetics. COATi will support an abundance of codon models by using a continuous-time Markov

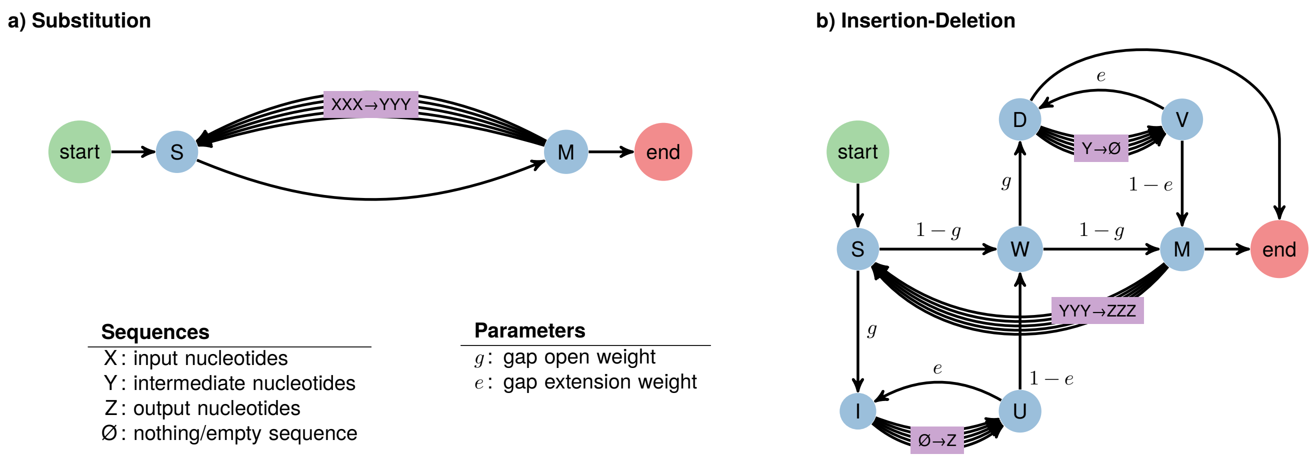

Figure 3: The evolution FST is assembled by composing a substitution FST and an indel FST. Each node represents a state in an FST while arcs display possible transitions between states (and their weights). Absorption and emission of symbols occurs between states. (a) The substitution FST encodes a 64x64 codon substitution model with 64 arcs from M to S. (b) The indel FST allows for insertions (I to U) and deletions (D to V). Contiguous insertions and deletions are always arranged for insertions to precede deletions to limit equivalent alignments.

model, with instantaneous substitution rate matrix Q:

$$Q_{ij} = \begin{cases} \mu_{ij} & \text{if } i \text{ and } j \text{ are synonymous} \\ \omega \cdot \mu_{ij} & \text{if } i \text{ and } j \text{ are nonsynonymous} \end{cases}$$

$$Q_{ii} = -\sum_{j:j\neq i} Q_{ij}$$

where each position in Q defines the rate that codon i changes to codon j and total rate for each row is zero. Model parameter μ_{ij} is the mutation rate of codon i to j and ω represents the strength of selection for amino acid changes. The substitution probability after time t is calculated via matrix exponentiation $P(j|i;t) = e^{Qt}$.

COATi will offer different ways to specify mutation rates (μ), including built-in models such as Muse and Gaut (MG94) (Muse & Gaut 1994), empirical codon model (ECM) (Kosiol *et al.* 2007), and the ability to read user-provided models via an input file.

2.1.4 Dynamic programming

Composition is one of the most powerful operations on FSTs, as it allows complex FSTs to be build from smaller and simpler parts. However, composing many large FSTs is expensive and can be prohibitive. Despite the existence of efficient C++ FST libraries, runtime is still limiting when dealing with sequences longer than a few thousand nucleotides.

To solve this issue, the search for an optimal path (alignment) through the evolution FST can be refor-

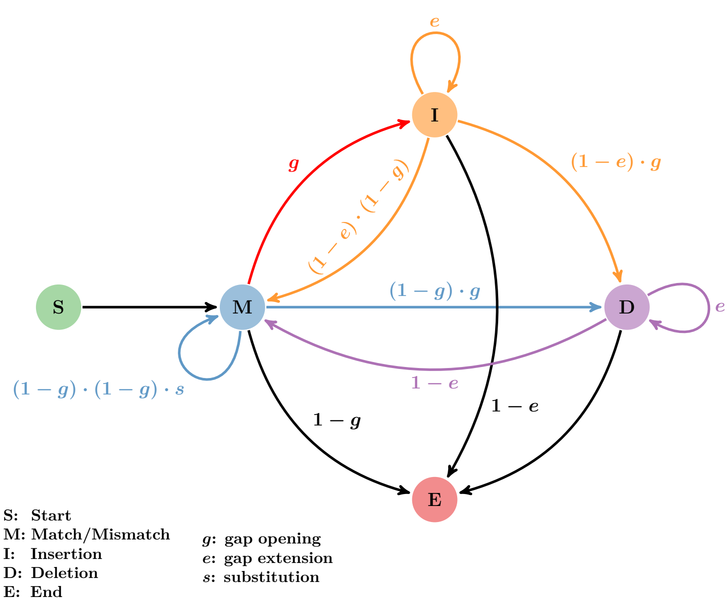

Figure 4: Simplified evolution FST, maintaining the exact transition weights.

mulated as a dynamic programming problem. Maintaining the statistical framework, COATi will implement a Gotoh-inspired algorithm thus reducing the problem to a manageable $\mathcal{O}(nm)$ runtime, where n and m are the length of the sequences. This can be further improved via Myers and Miller (Myers & Miller 1988), which implements a divide and conquer approach.

2.2 Aim 2 - Artifacts in Genomic Datasets

Errors and artifacts are a common problem in genomic datasets, notably frameshifts and early stop codons. In order to prevent errors from leading to inaccurate downstream analyses, current practices involve time and resource-consuming curation efforts that discard large amounts of data, consequently losing information.

Genomes for model organisms are often of high-quality after being refined over many iterations and having their coding sequences meticulously curated. On the contrary, non-model organisms typically have lower-quality genomes that have been only partially curated. Low-quality genomes often lack the amount of sequencing data needed to fix artifacts, including missing exons, erroneous mutations, and indels (Jackman *et al.* 2018).

FSTs and their well established methods provide an efficient framework to statistically align a sequence from a non-model organism against a sequence from a model organism. Therefore, I plan to equip COATi to correctly handle artifacts present in heterogeneous genomic datasets. COATi-alignpair will be able to model the pairwise alignment of a low-quality sequence against a high-quality sequence as a path through the evolution FST.

2.2.1 Marginal substitution model

Codon substitution models define the rate of change between nucleotide triplets, with the implicit assumption that codons from both sequences are accurately sequenced and mapped. To lessen this assumption and leverage the alignment on the high-quality sequence, over the low-quality sequence, the default codon model implemented on the substitution FST will be a marginalized codon model. Given the 64x64 codon substitution matrix P defined in aim 1, the marginalized version is defined

$$P'_{ijp} = \sum_{cod} \begin{cases} P(i|cod) & \text{if } cod_p = j \\ 0 & \text{otherwise} \end{cases}$$

Conceptually, P'_{ijp} represents the probability that codon i from the ancestor sequence changes to nucleotide j of the descendant sequence at position $p \in \{0,1,2\}$ of the reading frame. This results in a 64x4x3 matrix.

Codon models can be affected in the presence of erroneous bases, especially when these are found in first and second positions. This model emphasizes the position where the substitution in a codon occurs. Focusing on one nucleotide substitution at a time helps restrict the effects of low-quality data. Using the high-quality sequence as the ancestor, this marginal substitution matrix is more robust to artifacts in the descendant sequence.

2.2.2 Artifacts and ambiguous data

In the DNA alignment problem, the alphabet of nucleotides is ideally composed of four residues {A,C,G,T} plus gap {-}. Unfortunately, errors in sequencing and assembly introduce uncertainty that is represented by ambiguous residues. To represent all possibilities, the alphabet can be extended to include up to sixteen symbols, according to standardized IUPAC notation (Cornish-Bowden 1985).

Given that sequences from model organisms have been polished and refined over time, it is reasonable to assume that the high-quality sequence in our model is free of ambiguous nucleotides. In addition, adding support for all IUPAC nucleotide symbols for the reference sequence can add complexity to the marginal substitution model without a clear payoff. However, I plan on exploring the possibility of adding this feature.

In contrast, low-quality sequences are expected to contain ambiguous nucleotides and COATi will be equipped to handle them. A common strategy to handle ambiguous nucleotides, when not directly removing the containing codon, is to average over all possibilities. However, an ambiguous residue represents a single nucleotide that was inaccurately interpreted instead of an average of possibilities. To my knowledge, no alternative approaches have been explored for handling uncertain nucleotides in alignment. Therefore, I plan on evaluating other strategies to treat ambiguous nucleotides.

2.2.3 Model frameshifts

Indel FST (3-b) models the insertions and deletions when aligning a pair of sequences, including frameshift causing indels, by allowing gaps of any length to occur at any position. To distinguish between frameshift indels and non-frameshift indels, a more parameter-rich transducer can be designed. When setting the indel FST to only allow gaps of length multiple of three (one or more codons), this can be composed with a similar transducer that only allows gaps of length one or two. With this approach, longer frameshifts can be modeled by combining an indel (length multiple of three) with a frameshift (length one or two). I will compare the performance of the initial indel FST with the frameshift-specific model.

Assuming frameshifts are false positives, COATi will provide the option to correct frameshifts by adding ambiguous nucleotides that restore the original reading frame. This will ensure the alignment produced by our tool is properly formatted for use by any software in comparative genomic pipelines.

2.2.4 Biological frameshifts

While most frameshifts found in the alignment of protein coding sequences are expected to be errors due to strong purifying selection, in some cases frameshifts are believed to be biological (Hu & Ng 2012). To my knowledge, this particular case is not addressed by any current aligners, therefore, I plan on developing an approach that can model biological frameshifts.

2.3 Aim 3 - Estimate parameter values for COATi's model

The development of new models and tools that help understand natural phenomena moves science forward. COATi will help alleviate the expensive data curation steps that cause large amounts of information to be discarded, thus improving sequence alignment and the vast array of downstream analyses that follow. In addition, the model is designed to properly handle a wide variety of molecular data including pseudo-genes, with an emphasis on protein coding sequences.

While COATi will have a positive impact in the field, developing feature-rich models can present users with a challenge if left alone to tune its parameters. Thus, COATi will be capable of inferring biologically meaningful parameter estimates from sequence data.

2.3.1 Expectation-Maximization algorithm

The expectation-maximization algorithm (EM) (Dempster *et al.* 1977) is a classic method for deriving maximum likelihood estimates (MLE) of parameters in statistical models with latent variables. This iterative algorithm alternates between an expectation step and a maximization step until a convergence threshold is achieved. During the expectation step, information about the hidden data is inferred, which is then used to improve parameter estimates in the maximization step. The efficacy of the EM algorithm has been proven in the context of molecular evolution (e.g.Holmes & Rubin 2002; Holmes 2005). Therefore, I will use an EM approach to infer parameter values estimates from sequence data for COATi.

2.3.2 Model parameters: substitution parameters

COATi offers the possibility to use a custom substitution model by providing a substitution matrix. In addition, the built-in options are MG94 and ECM models. While both models characterize the codon to codon interactions, ECM specifies the codon frequencies while MG94 does not. For the latter, an underlying DNA substitution model is required. COATi will feature the popular general time reversible model (GTR) (Tavaré *et al.* 1986) when using

MG94. A characteristic of GTR is its ability to encode other well-known DNA models that can be seen as sub-cases such as JC69 (Jukes *et al.* 1969), HKY (Hasegawa *et al.* 1985), or TN93 (Tamura & Nei 1993). GTR is composed of ten total parameters, four nucleotide frequencies π_i and six transition rate parameters σ_j. In addition, one parameter, coefficient of selection ω, is required for constructing MG94.

2.3.3 Model parameters: indel parameters

The indel model, as described in figure 3-b, distinguishes between insertion and deletions, with two governing parameters, gap opening g and gap extension e. The probability that a gap occurs follows a geometric distribution with parameter g. The model can be extended by splitting both parameters to be event specific, i.e. insertion opening i_o, insertion extension i_e, deletion opening d_o, and deletion extension d_e.

2.3.4 Validation

A common approach for validation is to generate data with a wide set of known parameters values and assert that the estimates are correct. I will use DAWG (Cartwright 2005), an open-source C++ sequence evolution simulator able to generate sequence alignments. DAWG is well suited to generate a dataset for testing given its ability to specify both a substitution model (e.g. MG94) and an indel model. As in COATi, DAWG allows gaps to happen anywhere in the sequence, including within codons, and to span any number of bases, thus allowing frameshifts.

3 Preliminary Results

The most updated version of COATi can be found as an open source project on GitHub. Written in C++ 17, COATi can be built using the open source software Meson. Once compiled, it can be run from the command line using the syntax `coati command arguments [options]`.

The software development cycle follows best practices including continuous integration, unit testing with doctest, linting and formatting according to the Google C++ stylesheet with clang. Results from continuous integration together with test coverage are displayed on the GitHub repository.

Currently, COATi includes a functional version of `coati alignpair`, a pairwise aligner that offers different substitution models and can find an optimal alignment given a low and a high-quality sequence. The software package also includes a utility command `coati format` that is able to convert between fasta and phylip formatted files as well as extract specific sequences from a multi-sequence input. In addition, `coati msa`, under development, produces an initial multiple sequence alignment given a phylogenetic tree in newick format.

To illustrate the obstacles with current aligners and to showcase the performance of COATi, I have simulated pairwise alignments with empirical gaps patterns and evaluated the accuracy of popular cutting edge aligners ClustalΩ (Sievers *et al.* 2011), MACSE (Ranwez *et al.* 2011), MAFFT (Katoh *et al.* 2002), and PRANK (Löytynoja 2014) together with COATi.

I downloaded 4000 human genes and their gorilla homologous pairs from the ENSEMBL database (Hubbard *et al.* 2002) and aligned them using all five aligners. From those, 1660 alignments contained gaps identified by at least one method. Gap patterns extracted from all five methods were randomly introduced into the other 2340 initially ungapped sequence pairs to generate the 'true alignments' Alignment accuracy was measured using the distance metric d_{seq} (Blackburne & Whelan 2011) between simulated and inferred alignments. In addition, accuracy of positive and negative selection was calculated

Results. COATi was significantly more accurate (lower d_{seq}) than other aligners; all p-values were equal (MAFFT) or less than 1.714x10^{-8} according to the one-tailed Wilcoxon signed rank test. In addition, COATi produced more perfect alignments, less imperfect alignments, and had a higher positive and negative selection accuracy (Table 1).

MACSE was the only software to model frameshifts and out-of-phase gaps. Despite claiming a hybrid method that combines information from both DNA and amino acid levels, the implementation of MACSE is based solely

	COATi	PRANK	MAFFT	CLUSTALΩ	MACSE
Avg alignment error (d_{seq})	0.00060	0.01086	0.00671	0.01300	0.00611
Perfect alignments	1300	86	1282	634	1059
Best alignments	1756	188	1463	666	1129
Imperfect alignments	437	1651	455	1109	678
Accuracy of positive selection	97.3%	87.3%	85.8%	69.1%	81.5%
Accuracy of negative selection	99.8%	98.9%	98.7%	97.3%	98.5%

Table 1: Accuracy of COATi, PRANK, MAFFT, CLUSTALΩ, and MACSE, on 2340 simulated sequence pairs. Perfect alignments have ($d_{seq} = 0$), best alignments have lowest d_{seq}, and imperfect alignments have $d_{seq} > 0$ when at least one aligner found a perfect alignment.

on amino acid translation and scored using the popular BLOSUM62 (S. Henikoff & J. G. Henikoff 1992) matrix, for simplicity and speed reasons, as reported in Ranwez *et al.* 2011.

Among the remaining aligners, MAFFT was run with a DNA model, CLUSTALΩ performed a common strategy of aligning via amino acid translation, while PRANK was the only aligner with a codon model available. However, when using the codon model, PRANK replaces any unknown codons with 'NNN', modifying the original sequences and losing information.

To showcase the limitations of pairwise alignment using FSTs I benchmarked the FST version and the dynamic programming counterpart. Despite the existence of efficient C++ FST libraries and the usage of known optimization techniques, the runtime and memory requirements are impractical for sequences longer than a few hundred bases. Fortunately, the dynamic programming adaptation of COATi's model reduces costs significantly to levels similar to current aligners (Fig. 5).

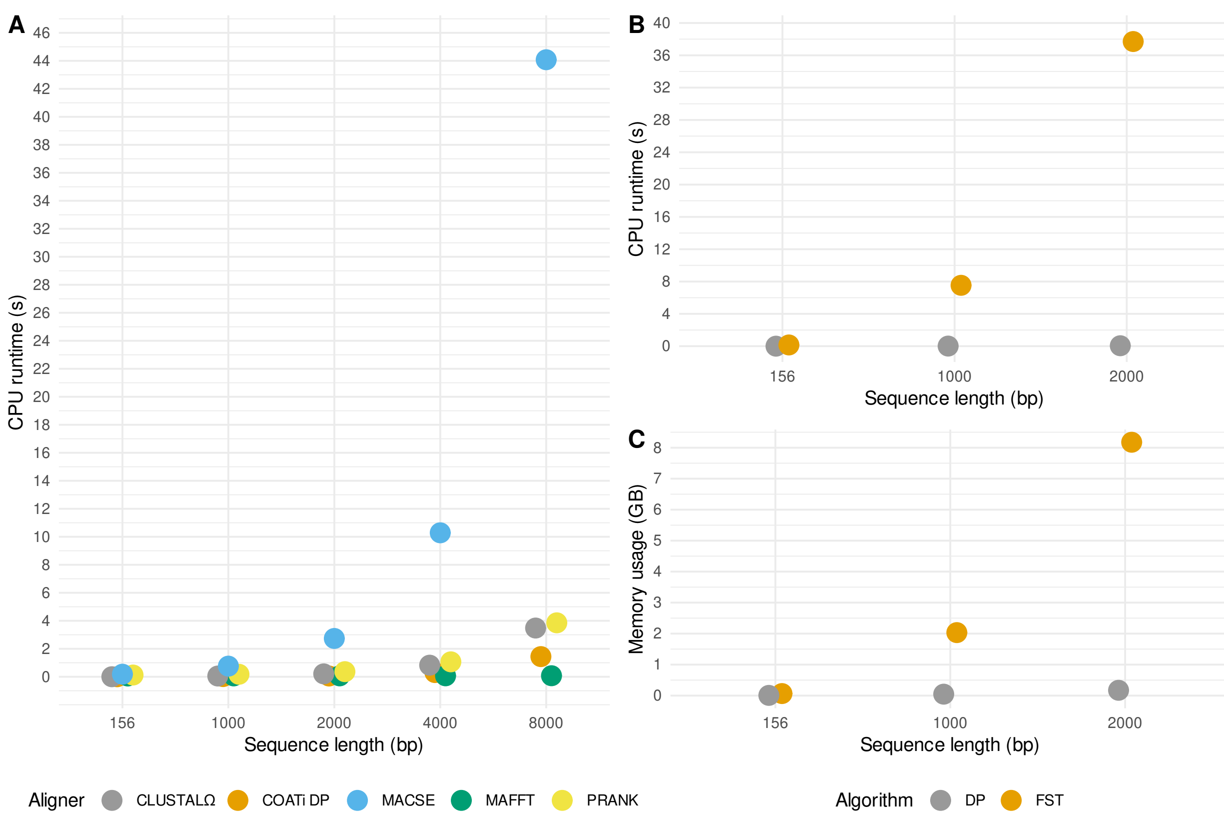

Figure 5: Runtime benchmark in seconds of CLUSTALΩ, COATi, MACSE, MAFFT, and PRANK aligning pairwise sequences of different lengths (A). Runtime (B) and memory usage (C) of COATi when aligning pairwise sequences of different lengths when using FSTs and a dynamic programming approach (DP).

4 Future Work

Looking forward, the logical and most beneficial next step for COATi should be extending the current model into a multiple sequence aligner (MSA). The first addition would be an algorithm that can assemble an initial alignment both given a phylogenetic tree and build a guide tree when not available. An iterative refinement step would follow by sampling alignment space in search of better alternatives. This would transform COATi into a complete and widely used tool.

References

1. Abascal, F., Zardoya, R. & Telford, M. J. TranslatorX: multiple alignment of nucleotide sequences guided by amino acid translations. *Nucleic acids research* **38,** W7–W13 (2010).
2. Bininda-Emonds, Olaf. transAlign: using amino acids to facilitate the multiple alignment of protein-coding DNA sequences. *BMC bioinformatics* **6,** 1–6 (2005).
3. Blackburne, B. P. & Whelan, S. Measuring the distance between multiple sequence alignments. *Bioinformatics* **28,** 495–502. ISSN: 1367-4803. eprint: https://academic.oup.com/bioinformatics/article-pdf/28/4/495/563214/btr701.pdf. https://doi.org/10.1093/bioinformatics/btr701 (Dec. 2011).
4. Bradley, R. K. & Holmes, I. Transducers: an emerging probabilistic framework for modeling indels on trees. *Bioinformatics* **23** (2007).
5. Cartwright, R. A. DNA assembly with gaps (Dawg): simulating sequence evolution. *Bioinformatics* **21,** iii31–iii38 (2005).
6. Cornish-Bowden, A. Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations 1984. *Nucleic acids research* **13,** 3021 (1985).
7. Dempster, A. P., Laird, N. M. & Rubin, D. B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* **39,** 1–38. ISSN: 00359246. %7Bhttp://www.jstor.org/stable/2984875%7D (2022) (1977).
8. Hasegawa, M., Kishino, H. & Yano, T.-a. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of molecular evolution* **22,** 160–174 (1985).
9. Henikoff, S. & Henikoff, J. G. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences* **89,** 10915–10919 (1992).
10. Holmes, I. & Rubin, G. An expectation maximization algorithm for training hidden substitution models. *Journal of molecular biology* **317,** 753–764 (2002).
11. Holmes, I. Using evolutionary expectation maximization to estimate indel rates. *Bioinformatics* **21,** 2294–2300 (2005).
12. Holmes, I. & Bruno, W. J. Evolutionary HMMs: a Bayesian approach to multiple alignment. *Bioinformatics* **17,** 803–820 (2001).
13. Hu, J. & Ng, P. C. Predicting the effects of frameshifting indels. *Genome biology* **13,** 1–11 (2012).
14. Hubbard, T. *et al.* The Ensembl genome database project. *Nucleic acids research* **30,** 38–41 (2002).
15. Jackman, S. D. *et al.* Tigmint: correcting assembly errors using linked reads from large molecules. *BMC bioinformatics* **19,** 1–10 (2018).
16. Jukes, T. H., Cantor, C. R., Munro, H., *et al.* Mammalian protein metabolism (1969).
17. Katoh, K., Misawa, K., Kuma, K.-i. & Miyata, T. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic acids research* **30,** 3059–3066 (2002).

18. Kosiol, C., Holmes, I. & Goldman, N. An empirical codon model for protein sequence evolution. *Molecular biology and evolution* **24,** 1464–1479 (2007).

19. Löytynoja, A. in *Multiple sequence alignment methods* 155–170 (Springer, 2014).

20. Morrison, D. A. *Multiple Sequence Alignment is not a Solved Problem* 2018. arXiv: 1808.07717 [q-bio.PE].

21. Muse, S. V. & Gaut, B. S. A likelihood approach for comparing synonymous and nonsynonymous nucleotide substitution rates, with application to the chloroplast genome. *Molecular biology and evolution* **11,** 715–724 (1994).

22. Myers, E. W. & Miller, W. Optimal alignments in linear space. *Bioinformatics* **4,** 11–17 (1988).

23. Ranwez, V., Harispe, S., Delsuc, F. & Douzery, E. J. MACSE: Multiple Alignment of Coding SEquences accounting for frameshifts and stop codons. *PloS one* **6,** e22594 (2011).

24. Rosenberg, M. S. *Sequence alignment: methods, models, concepts, and strategies* (Univ of California Press, 2009).

25. Schneider, A. *et al.* Estimates of positive Darwinian selection are inflated by errors in sequencing, annotation, and alignment. *Genome biology and evolution* **1,** 114–118 (2009).

26. Sievers, F. *et al.* Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Molecular systems biology* **7,** 539 (2011).

27. Smith, T. F., Waterman, M. S., *et al.* Identification of common molecular subsequences. *Journal of molecular biology* **147,** 195–197 (1981).

28. Suchard, M. A. & Redelings, B. D. BAli-Phy: simultaneous Bayesian inference of alignment and phylogeny. *Bioinformatics* **22,** 2047–2048 (2006).

29. Tamura, K. & Nei, M. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Molecular biology and evolution* **10,** 512–526 (1993).

30. Tavaré, S. *et al.* Some probabilistic and statistical problems in the analysis of DNA sequences. *Lectures on mathematics in the life sciences* **17,** 57–86 (1986).

31. Yoon, B.-J. Hidden Markov models and their applications in biological sequence analysis. *Current genomics* **10,** 402–415 (2009).