COATi: statistical pairwise alignment of protein-coding sequences

Juan J. García Mesa[1,2], Ziqi Zhu[1,3], Reed A. Cartwright[1,3,*]

1 The Biodesign Institute, Arizona State University, Tempe, Arizona, USA
2 Ira A. Fulton Schools of Engineering, Arizona State University, Tempe, Arizona, USA
3 School of Life Sciences, Arizona State University, Tempe, Arizona, USA

* cartwright@asu.edu

Abstract

Sequence alignment is an essential method in bioinformatics and the basis of many analyses, including phylogenetic inference, ancestral sequence reconstruction, and gene annotation. Sequence artifacts and errors made in alignment reconstruction can impact downstream analyses leading to erroneous conclusions in comparative and functional genomic studies. For example, abiological frameshifts and early stop codons are common artifacts found in protein coding sequences that have been annotated in reference genomes. While such errors are eventually fixed in the reference genomes of model organisms, many genomes used by researchers contain these artifacts, and researchers often discard large amounts of data in comparative genomic studies to prevent artifacts from impacting results. To address this need, we present COATi, a statistical, codon-aware pairwise aligner that supports complex insertion-deletion models and can handle artifacts present in genomic data. COATi allows users to reduce the amount of discarded data while generating more accurate sequence alignments.

Introduction

Sequence alignment is a fundamental task in bioinformatics and a cornerstone step in comparative and functional genomic analysis (Rosenberg, 2009). While sophisticated advancements have been made, the challenge of alignment inference has not been fully solved (Morrison, 2015). The alignment of protein-coding DNA sequences is one such challenge, and a common approach to this problem is to perform alignment inference in amino-acid space (e.g. Bininda-Emonds, 2005; Abascal et al., 2010). While this approach is an improvement over DNA models, it discards information, underperforms compared to alignment at the codon level, and fails in the presence of artifacts, such as frameshifts and early stop codons. While some aligners can utilize codon substitution models, they are often not robust against coding-sequence artifacts.

Within protein-coding sequences, indels may occur in between any pair of adjacent nucleotides, and therefore, gaps in alignments of natural sequences may occur both between and within codons (Fig. 1). Gaps that occur after the first position or second position of a codon are known as phase-1 and phase-2 gaps, respectively. Gaps that occur between codons are known as either phase-3 gaps (this study) or phase-0 gaps (e.g. Taylor et al., 2004). While all three phases of gaps occur in natural sequences, alignments performed in amino-acid or codon space force all gaps to be phase-3 gaps. Because only about 42% of indels are phase-3 (Taylor et al., 2004; Zhu, 2022), this mismatch between aligner

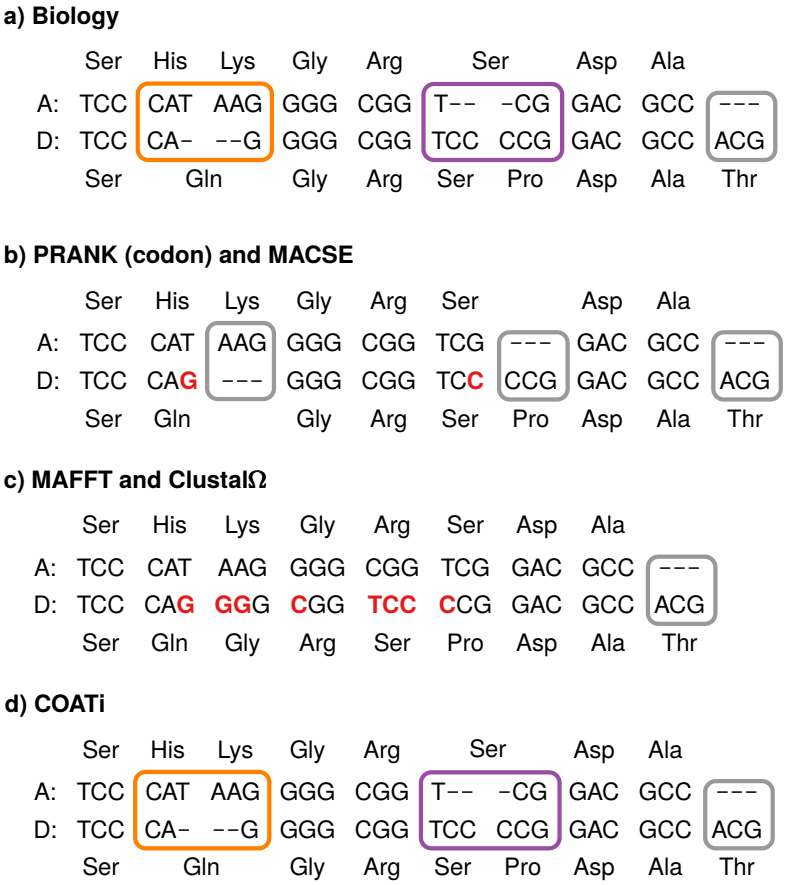

Figure 1: Standard algorithms produce suboptimal alignments. (a) shows the true alignment of an ancestor sequence (A) and a descendant sequence (D). (b–d) are the results of different aligners. Nucleotide mismatches are highlighted in red.—Notably, COATi is the only aligner able to retrieve the biological alignment in this example.—Indels in protein-coding sequences can be classified as having one of three difference phases and being one of two different types. Phases refer the the location of the gap with respect to the reading frame, while types refer to the consequence of the indel. Phase-1, phase-2, and phase-3 indels are shown in purple, orange, and gray respectively. Additionally, the orange indel is type-II (an amino-acid indel plus an amino-acid change) while the purple indel is type-I (an amino-acid indel only). The difference between in-frame and frameshift indels is not displayed.

assumptions and biology can produce sub-optimal alignments and inflated estimates of sequence divergence (Fig. 1).

Bioinformatic pipelines need to be robust to variation in quality across genomic datasets because uncorrected errors in the alignment stage can lead to erroneous results in comparative and functional genomic studies (Schneider et al., 2009; Fletcher and Yang, 2010; Hubisz et al., 2011). While genomes for model organisms often get refined over many iterations and contain meticulously curated protein-coding sequences, genomes for non-model organisms might only receive partial curation and typically have lower quality sequences and annotations. These genomes often lack the amount of sequencing data needed to

fix artifacts, including missing exons, erroneous mutations, and indels (Jackman et al., 2018). When comparative and functional genomics studies include data from non-model organisms, care must be taken to identify and manage such artifacts; however, current alignment methods are ill-equipped to handle common artifacts in genomic data, requiring costly curation practices that discard significant amounts of information.

To address current limitations of alignment software to accurately align protein-coding sequences, we present COATi, short for COdon-aware Alignment Transducer, a pairwise statistical aligner that incorporates evolutionary models for protein-coding sequences and is robust to artifacts present in modern genomic data sets.

Materials and Methods

Statistical Alignment via Finite State Tranducers

In statistical alignment, sequence alignments are scored based on a stochastic model, typically derived from molecular evolutionary processes (Lunter et al., 2005). An advantage of statistical alignment is that its parameters are derived from biological processes, allowing them to be estimated directly from data or extracted from previous studies. While approaches vary, a statistical aligner for a pair of sequences, X and Y, typically finds an alignment, Aln, that maximizes the joint probability $P(Aln, X, Y)$ or samples alignments from the posterior $P(Aln|X, Y)$. This is typically performed using pairwise hidden Markov models (pair-HMMs; Bradley and Holmes, 2007). Pair-HMMs are computational machines with two output tapes. Each tape represents one sequence, and a path through the pair-HMM represents an alignment of the two sequences. Conceptually, pair-HMMs generate two sequences from an unknown ancestor and can calculate the joint probability $P(Aln, X, Y)$ (Yoon, 2009).

While the use of pair-HMMs is ubiquitous in bioinformatics, they are limited to modeling the evolution of two related sequences from an unknown ancestor. As an alternative, finite-state transducers (FSTs, Fig. 2) allow researchers to model the evolution of a descendant sequence from an ancestral sequence. FSTs are computational machines with one input tape and one output tape and provide similar benefits to pair-HMMs, while being more suitable for evolutionary models (Bradley and Holmes, 2007). FSTs consume symbols from an input tape and emit symbols to an output tape based on the symbols consumed and the structure of the FST. Conceptually, FSTs generate a descendant sequence, D, from a known ancestor, A, and can calculate the conditional probability $P(Aln, D|A)$.

There are well-established algorithms for combining FSTs in different ways allowing the design of complex models by combining simpler FSTs, including concatenation, composition, intersection, union, and reversal (Bradley and Holmes, 2007; Silvestre-Ryan et al., 2021). Specifically, composition is an algorithm to combine two FSTs by sending the output of one FST into the input of another, creating a new, more complex transducer (Mohri et al., 2005). Figure 2 illustrates how FSTs modeling sequencing errors (Fig. 2a) and ambiguity (Fig. 2b) can be combined via composition to produce an FST that does both (Fig. 2c). Conceptually, composition creates an FST that generates a descendant sequence from a known ancestor via an unknown intermediate, J, and can calculate the conditional probability $P(D|A) = \sum_J P(D|J)P(J|A)$.

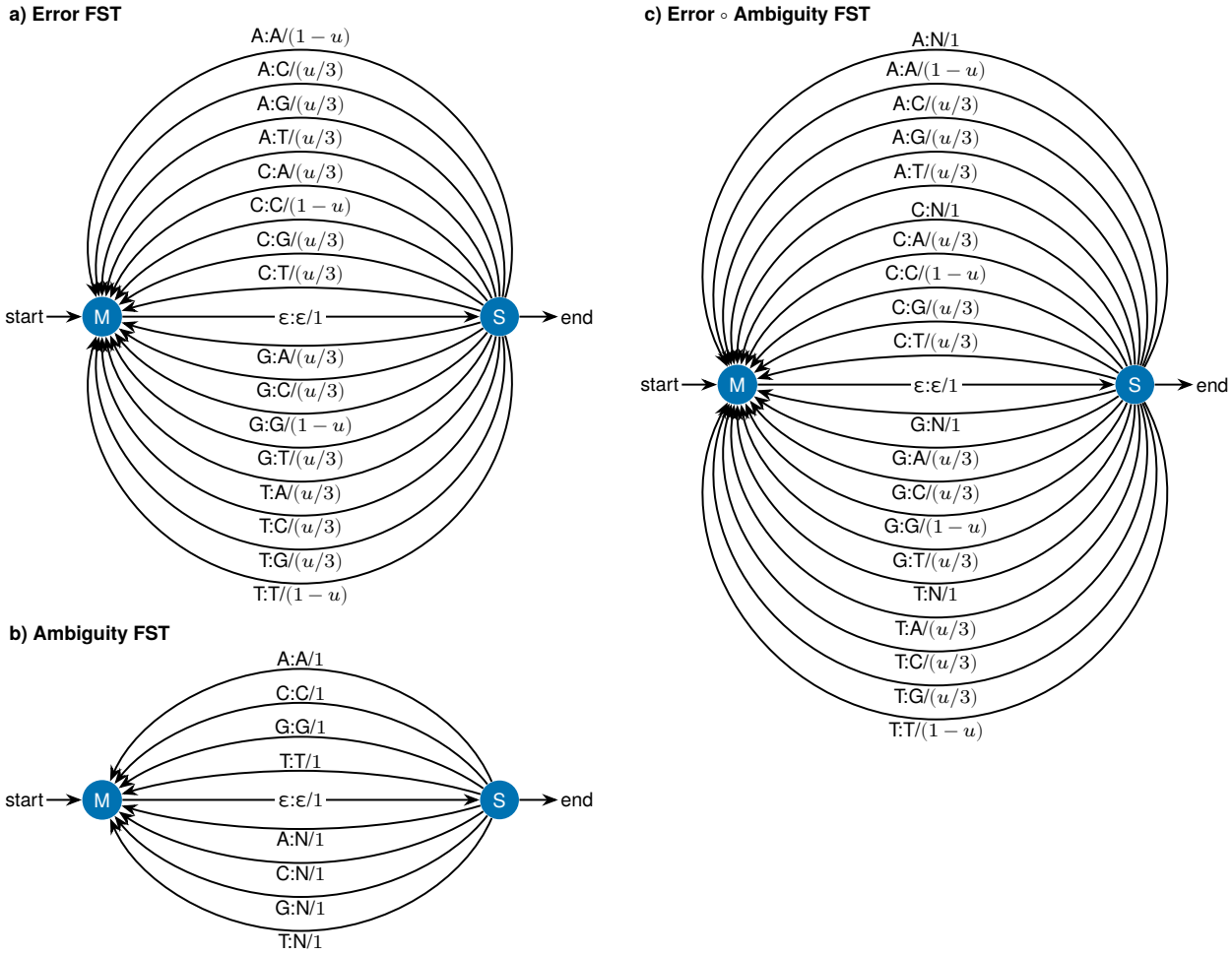

Figure 2: Finite state transducers (FSTs) model the generation of an output sequences based on an input sequence. (a) A graph of a probabilistic FST (Cotterell et al., 2014) for base-calling errors using a Mealy-machine architecture, where parameter u is the error rate. This graph contains two states (S and M) connected by arcs, with labels "input symbols : output symbols / weight". Arcs consume symbols from the input sequence and emit symbols to the output sequence. Weights describe the probability that an arc is taken given the input symbols. Epsilon (ε) is a special symbol denoting that no symbols were either consumed or emitted. (b) An FST for matching sequences against ambiguous nucleotides (N). This FST is not a true probabilistic FST and cannot be used to simulate output sequences since it is missing a parameter to control how often Ns are added to the output sequence. (c) An FST that results from the composition (∘ operation) of the Error FST with the Ambiguity FST. As with (b), this composed FST cannot be used to simulate output sequences; however, it does properly weight the ambiguous nucleotide N as representing any other symbol.

The COATi FST

COATi aligns pairs of sequences using a statistical alignment model, which is implemented as a finite-state transducer derived from the composition of multiple FSTs, each representing a specific biological or technical process: (1) the codon substitution FST, (2) the indel FST, (3) the error FST, and (4) the ambiguity FST (Figs. 2–3; c.f. Holmes and Bruno, 2001). We call this transducer the COATi FST. Codon substitution models are uncommon in

sequence aligners, despite their extensive use in phylogenetics. COATi implements the Muse and Gaut (1994) codon model (codon-triplet-mg) and the Empirical Codon Model (codon-triplet-ecm; Kosiol et al., 2007). It also lets the user provide a codon substitution matrix. A key innovation of COATi is that it combines a codon substitution model with a nucleotide-based indel model, allowing gaps to occur both between and within codons (c.f. Ranwez et al., 2011, 2018). This also allows the aligner to be robust against sequencing artifacts that produce sequences with disrupted reading frames.

Since codon substitution is the first process in COATi's model, the input sequence to the COATi FST must be compatible with a codon-substitution model, i.e. be a multiple of three nucleotides and not contain any ambiguous symbols or stop codons. The phases, reading frames, and amino-acid contexts of alignment columns is determined by the input sequence, and better alignments will be generated if the input is of high quality and free of artifacts. Depending on context, we may refer to the input sequence as the "ancestral" or "reference" sequence. In contrast, the output sequence must be compatible with the ambiguity FST, can be of any length, and can contain any nucleotides or "N". This allows COATi to align lower-quality sequences that may contain artifacts against a high-quality reference sequence. We may refer to the output sequence as the "descendant", "non-reference", or "query" sequence. The choice of which sequence is the input sequence and which sequence is the output sequence is left up to the user.

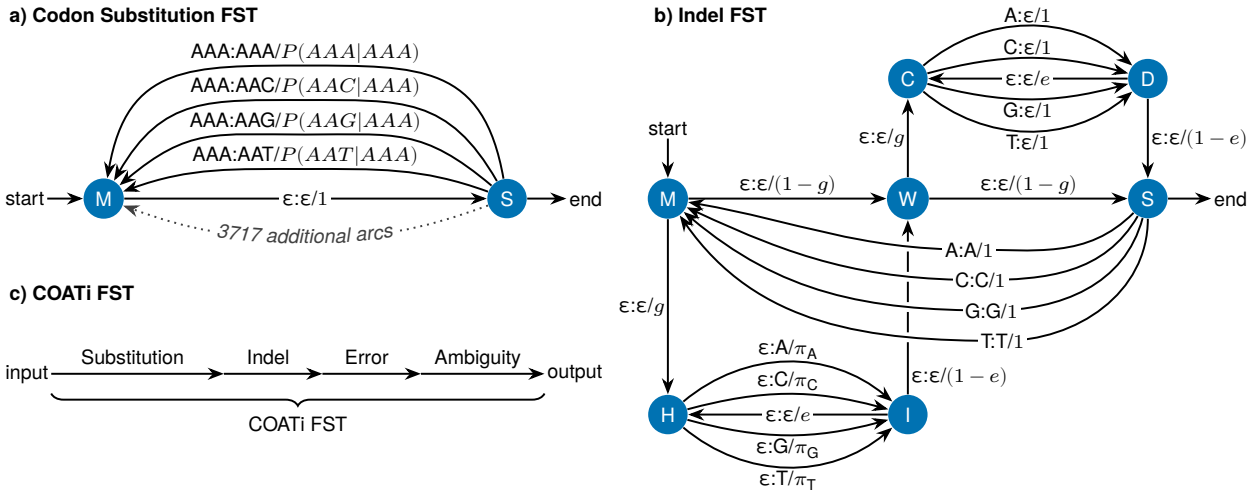

Figure 3: The COATi FST is built from simpler FSTs via composition. (a) The substitution FST encodes a 61×61 codon substitution model with 3721 arcs from S to M. These arcs consume three nucleotides from the input tape and emit three nucleotides to the output tape. The weight of each arc is a conditional probability derived from a codon substitution model. See Fig. 2 for more details about reading this graph. (b) The indel FST allows for insertions (H to I) and deletions (C to D). Here g is the "gap opening parameter" and e is the "gap extension" parameter. Insertion arcs are weighted according to the codon model's stationary distribution of nucleotides, and deletion arcs have a weight of 1. This FST is structured such that if insertions and deletions are contiguous, insertions will precede deletions. (c) The COATi FST is derived via composition from the codon substitution, indel, error, and ambiguity FSTs.

In order to use the COATi FST to align an output sequence against an input sequence,

we first convert each sequence into an acceptor, represented as a linear transducer where the input and output symbols of each transition are identical and each transition represents one nucleotide of a sequence (Allauzen et al., 2007). By composing the input and output acceptors with the COATi FST, we generate a transducer of all possible alignments of the two sequences. Any path through this FST represents a pairwise alignment, while the shortest path (by weight) corresponds to the best alignment. If more that one optimal alignment exists, ties are broken according to the implementation of the shortest-path algorithm. All FST operations in COATi, including model development, composition, search for the shortest path, and other optimization algorithms, are performed using the C++ openFST library (Allauzen et al., 2007). An example of an FST-based alignment can be found in Supplementary Materials Figure 1.

The Marginal Model

The COATi FST has a large state space to keep track of codon substitution rates when codons can be interspersed with indel events. This additional state space increases the computational complexity of the alignment algorithm. To reduce the runtime complexity of COATi, we have also developed an approximation of the COATi FST that can be implemented with standard dynamic programming techniques. This approximation uses a marginal substitution model where the output nucleotides are independent of one another and only depend on the input codon and position. This produces a $(61 \times 3) \times 4$ substitution model and eliminates the need to track dependencies between output nucleotides.

A marginal substitution model is calculated from a standard substitution model by calculating the marginal probabilities that each ancestral codon produces specific descendant nucleotides at each reading frame position. Specifically, let

$$P_{\text{cod}}\left(Y_1 = y_1, Y_2 = y_2, Y_3 = y_3 | X_1 = x_1, X_2 = x_2, X_3 = x_3\right)$$

represent transition probabilities from a codon model, and

$$P_{\text{mar}}\left(Y_p = y | X_1 = x_1, X_2 = x_2, X_3 = x_3\right) = \sum_{y_1, y_2, y_3} I(y_p = y) P_{\text{cod}}\left(y_1, y_2, y_3 | x_1, x_2, x_3\right)$$

represent the marginal transition probabilities, where $p \in \{1, 2, 3\}$ is the position of the descendant nucleotide relative to the ancestral reading frame and $I()$ is an indicator function. COATi contains marginal models for both Muse and Gaut (1994) or the Empirical Codon Model, resulting in the marginal models codon-marginal-mg and codon-marginal-ecm. These models emphasize the position in a codon where the substitution occurs, help restrict the effects of low-quality data in the descendant sequence, and allow more than one substitution per codon. In combination with the indel model, alignment using the marginal model is implemented using dynamic programming.

Empirical Dataset

Humans and gorillas are two closely related species with very different levels of genome curation. The human reference genome has been revised dozens of times and is currently on version GRCh38.p14, while the gorilla reference genome has only been revised a handful of times and is currently on version gorGor4 (cf. ENSEMBL database v110; Hubbard et al.,

2002). Additionally significant levels of investment have been made to correctly identify and annotate human genes, while gorilla annotations have received limited support in comparison. Together, these reference genomes provide a good opportunity to compare COATi against other aligner as they offer one genome that is high-quality (human) and sister genome that is lower-quality (gorilla).

We used ENSEMBL database v110 (Hubbard et al., 2002) to create an empirical dataset of protein-coding sequences for both human genes and their gorilla orthologs. We first selected human protein-coding genes that belonged to the Consensus Coding Sequence Set, that were located on an autosomal chromosome, and that had a one-to-one gorilla ortholog. We selected the canonical isoform for both species and removed any pair in which the total nucleotide length was larger larger than 6,000 nucleotides. This resulted in 14,127 sequence pairs and corresponding FASTA files containing CDS sequences. Due to the way that canonical isoforms are identified, there is no guarantee that the isoforms are orthologous even though the genes are. Therefore, a subset of the sequence pairs in this dataset contain human and gorilla sequences with different exon compositions. We have made no attempt to correct these sequence pairs because in our experience genome-wide studies rarely control for such artifacts.

Alignment Strategies

In order to compare COATi against other aligners, we evaluated five different alignment models: (1) COATi's FST model (i.e. codon-triplet-mg), (2) the amino-acid model of ClustalΩ v1.2.4 via translation and reverse translation, (3) the amino-acid aware nucleotide model of MACSE v2.06, (4) the DNA model of MAFFT v7.505, and (5) the codon model of PRANK v.150803 (Sievers et al., 2011; Ranwez et al., 2018; Katoh and Standley, 2013; Löytynoja, 2014). Together, these five methods allow us to evaluate both different alignment strategies and different software implementations.

Empirical Simulation Algorithm

In order to compare COATi against other aligners on realistic datasets with known alignments, we developed a procedure to introduce realistic gap patterns into human-gorilla orthologous gene pairs that did not previous contain indels. Briefly, we downloaded 16000 human genes and their gorilla orthologs from the ENSEMBL database (Hubbard et al., 2002). After downloading, we removed 2232 sequence-pairs longer than 6000 nucleotides and aligned the remaining pairs with the five alignment methods. Next we separated sequence pairs into two sets: (1) 6048 sequence pairs for which at least one aligner added gaps and (2) 7761 sequence pairs for which all five aligners did not introduce any gaps. We identified gap patterns from the sequence pairs that were aligned with gaps and randomly introduced them into the ungapped sequence pairs. We used an equal number of randomly sampled gap patterns from each aligner to produce a benchmark set of known alignments. The phases of gaps were preserved as were the spacings of any clusters of gaps. Segments of matches that were 99 nucleotides or longer were allowed to change length to accommodate the length of the ungapped sequence pair. (This criteria was recursively lowered if gap pattern did not fit the ungapped sequence pair.)

For example, consider the gapped alignment represented by the CIGAR string "170M 3D 10M 6I 102M" applied to an ungapped alignment of human and gorilla sequences that

are 300 nucleotides long. First, we modify the CIGAR string to insert flexible lengths for any match segment that is 99 or more nucleotides long while preserving phase. This results in a new CIGAR string of "101M *M 3D 10M 6I 99M *M", where * represents the locations that have flexible length. Considering only matches and deletions, this CIGAR string has 213 fixed nucleotides, leaving 87 nucleotides in the human sequence to be allocated to the flexible locations. Since there are two locations of flexible length, we draw one random break point uniformly while maintaining phase, producing the final CIGAR string of "101M 9M 3D 10M 6I 99M 78M", which is used to add gaps into the target, ungapped sequence pair. To apply deletions, we remove the corresponding nucleotides from the gorilla sequence, and to apply insertions, we add random nucleotides according to their stationary frequency to the gorilla sequence at the corresponding locations. The human sequence is left unchanged.

Alignment Accuracy Measures

We used multiple statistics to to quantify the similarity between each alignment in the benchmark and the corresponding output obtained by the different tools.

Alignment Error. To quantify the error between estimated alignments and the benchmark alignments, we used the alignment error metric d_{seq} (Blackburne and Whelan, 2011). Intuitively, d_{seq} ranges between zero and one and can be interpreted as the fraction of nucleotides in the sequence pair that are aligned differently between estimated and benchmark alignments. This metric summarizes each alignment by building homology sets for each nucleotide in the sequence pair. Briefly, if an alignment column contains position i from the first sequence and position j from the second sequence, then $\{j\}$ is the homology set for position i in the first sequence and $\{i\}$ is the homology set for position j in the second sequence. If a position is aligned against a gap, then its homology set is simply $\{\text{gap}\}$. This treats all gaps equally as the location of the gap is not recorded. Finally, d_{seq} is calculated between an estimated and a benchmark alignment as the average, normalized Hamming distance between the homology sets for each nucleotide in the sequence pair. See Blackburne and Whelan (2011) for further details.

Perfect, Imperfect, and Best Alignments. We quantified the number of perfect, imperfect and best alignments each aligner produced. We define perfect alignments as alignments with a distance of zero to their benchmark alignment ($d_{seq} = 0$) or any alignment that is evolutionary equivalent to an alignment with a distance of zero. Equivalency was included in our definition of perfect alignments to prevent the manner by which aligners break ties between multiple optimal alignments from impacting our results. Evolutionary equivalence was determined by scoring alignments using COATi's marginal model. Any alignment that had a score which matched the score of its benchmark alignment was considered perfect even if its distance to the benchmark was greater than zero. Imperfect alignments are defined as alignments that are not perfect when another method successfully produces a perfect alignment for the same pair of sequences. Best alignments are the alignments that the lowest distance d_{seq} to the true alignment, including ties. Taken together, these three statistics not only allow a direct comparison of aligners but also expose instances where all aligners fall short of achieving a perfect result.

Selection. Alignment error negatively impacts the identification of genes that have experienced positive and negative selection. To evaluate the impact of aligners on selection identification, we estimated K_S and K_A statistics (also known as d_s and d_n) for each estimated alignment and compared them to the benchmark alignment. Briefly, K_S is the number of substitutions per synonymous site and K_A is the number of substitutions per non-synonymous site between two protein-coding sequences. We used the method developed by Li (1993) and independently derived by Pamilo and Bianchi (1993) to estimate these statistics as implemented in the R package seqinr (Charif and Lobry, 2007).

Briefly, this method takes two aligned sequences and classifies the sites in each sequence as non-degenerate, two-fold degenerate, or four-fold degenerate based on the standard genetic code. A site is non-degenerate if 0/3 possible nucleotide changes to that site are synonymous, two-fold degenerate if 1/3 of the possible changes are synonymous, and four-fold degenerate if 3/3 possible changes are synonymous. (The rare three-fold degenerate sites are treated as two-fold degenerate in this method.) First the method calculates the average numbers of sites of each type in the sequence pair (L_0, L_2, and L_4). Next, following Li et al. (1985), it uses Kimura's two-parameter model (Kimura, 1980) and the alignment to estimate the numbers of transitions (A_i) and transversions (B_i) that occurred per each i-th type. Using these statistics, Li (1993) estimated K_S and K_A as

$$K_S = \frac{L_2 A_2 + L_4 A_4}{L_2 + L_4} + B_4 \qquad\qquad K_A = A_0 + \frac{L_0 B_0 + L_2 B_2}{L_0 + L_2}$$

We considered any alignment to be showing evidence of positive selection if $K_A/K_S > 1$ and negative selection if $K_A/K_S < 1$. We estimated F_1 scores for both positive and negative selection by comparing estimated alignments to benchmark alignments. F_1 is the harmonic mean of precision and recall:

$$F_1 = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

where TP is the number of alignments that correctly predicted positive (or negative) selection, FP is the number of alignments that incorrectly predicted positive (or negative) selection, and FN is the number of alignments that incorrectly predicted the absence of positive (or negative) selection. F_1 allows us to measure how well aligners produce alignments that correctly identify the presence and absence of positive selection or negative selection.

Evolutionary Distance. Alignment error also negatively impacts the estimation of evolutionary distances and phylogenetic trees. To quantify the impact of aligners on the estimation of evolutionary distances, we compared Kimura's 2-parameter distance (K2P; Kimura, 1980) calculated for the estimated and benchmark alignments. This distance corrects for multiple mutations at a site and takes into account differences in transition and transversion rates. It also assumes equal nucleotide frequencies and no variations in the rate of substitution across sites. While Kimura's 2-parameter distance is more suitable for non-coding sequences, it is straight forward to calculate and provides a quantitative measure of the evolutionary divergence between sequences. We used the R software package ape (Paradis and Schliep, 2019) to calculate K2P distances.

Results and Discussion

COATi obtained better results compared to a wide variety of alignment strategies. It was significantly more accurate (lower d_{seq}) at inferring the empirically simulated alignments compared to other methods; all p-values were less than $1.3 \cdot 10^{-76}$ according to the one-tailed, paired Wilcoxon signed-rank tests (Supplementary Materials Figure 1). Notably, the average alignment error of the second best protocol was six times larger than COATi's. In addition, COATi produced more perfect alignments, less imperfect alignments, and more accurately inferred events of positive and negative selection (Table 1). Furthermore, the estimated evolutionary divergence from the alignments retrieved by COATi was substantially less overestimated than other methods (Table 1, Supplemental Materials Figure 8).

Aligning sequences using amino-acid translation via ClustalΩ obtained the highest average alignment error and also had difficulties correctly identifying positive selection and estimating evolutionary distances. These results are not surprising because alignments in amino-acid space only permit phase-3 gaps and are sensitive to frameshift artifacts, which inflate divergence estimates and impact selection detection. Conversely, we obtained better results when aligning in nucleotide space using both amino-acid aware MACSE and amino-acid agnostic MAFFT. These results indicate that aligning protein-coding sequences in nucleotide space works reasonably well if the sequences come from closely related species, such as humans and gorillas, as it supports all three gap phases and is robust to frameshift artifacts. If the sequences were further apart, we predict that amino-acid agnostic approaches would begin to produce unreasonable alignments. And finally, our PRANK protocol used a codon substitution model which should have been an improvement over both amino-acid and nucleotide models; however, PRANK's codon model only permits phase-3 gaps and is very sensitive to frameshift artifacts. In fact, PRANK refuses to align any sequence using a codon model if its length is not a multiple of three.

	COATi	**MAFFT**	**PRANK***	**MACSE**	**ClustalΩ**
Method	Trip-MG	DNA	Codon	DNA+AA	AA
Avg alignment error (d_{seq})	0.00221	0.01471	0.01828	0.01399	0.02929
Best alignments	5139	4692	4774	3737	2615
Perfect alignments	5793	5292	4725	2861	2893
Imperfect alignments	1048	1549	2116	3980	3948
F_1 score for positive selection	98.1%	84.3%	86.7%	79.5%	68.7%
F_1 score for negative selection	99.8%	98.4%	98.7%	98.2%	96.9%
Overestimated K2P distances	10.9%	26.6%	33.8%	48.7%	61.8%

*PRANK produced 42 empty alignments, calculations are based on 7719 alignments.

Table 1: COATi generates better alignments than other alignment algorithms. Results of COATi, PRANK, MAFFT, ClustalΩ, and MACSE aligning 7761 empirically simulated sequence pairs. Best alignments have the lowest d_{seq} (including ties), perfect alignments have the same score as the true alignment, and imperfect alignments have a different score than the true alignment when at least one method found a perfect alignment.

COATi is not a symmetric pairwise aligner, as reference sequences are more constrained than non-reference sequences. To test how well COATi performs when the roles of se-

quences are reversed, we realigned the 7761 benchmark sequence alignments using gorilla as the reference. Notably, COATi was only able to align 4003 sequence pairs due to the presence of early stop codons or other artifacts in 3758 simulated gorilla sequences. While the simulation algorithm prevents disrupting the reading frame and introducing frameshifts, it does not prevent early stop codons from being formed in the descendant sequence. Despite this limitation, we analyzed the 4003 alignments and compared the results across methods, including COATi using the human sequence as the reference. The results show a decrease, albeit small, in accuracy across all metrics when the low-quality sequence is used as the reference in comparison to the reverse (Supplementary Materials Tab. 6). However, the results for COATi continue to be a significant improvement over other aligners.

COATi uses finite-state transducers to align pairs of sequences using a codon-aware statistical model. While COATi has multiple modes and models, here we have focused on evaluating the utility of COATi FST for estimating pairwise alignments. We have shown that COATi offers a biologically significant improvement over other methods when aligning pairs of sequences separated by short evolutionary distances in the presence of genomic artifacts. In other work, we have begun to explore application of COATi to more divergent sequences and the accuracy of the marginal approximation of the COATi FST (García Mesa, 2023).

COATi is under active development. We plan on extending it to support more complex gap models, e.g. mixtures of single-nucleotide and triple-nucleotide indel models and weighing gap openings to reflect known selection on indel phases (Zhu, 2022). We plan on improving its multiple-sequence alignment and alignment sampling capabilities, as well as implement new models for aligning long-read sequences of genes against reference genomes. Our goal is to develop COATi into a user-friendly suite of tools that will allow researchers to analyze more data with higher accuracy and facilitate the study of important biological processes that shape genomic data.

Availability

The source code for COATi, along with documentation, is freely available on GitHub: https://github.com/CartwrightLab/coati and is implemented in C++. Additional information, code, and workflows to replicate the analysis can be found on GitHub: https://github.com/jgarciamesa/coati-testing.

Acknowledgments

This research was funded by NSF award DBI-1929850. The authors would like to thank Profs. Marco Mangone, Ted Pavlic, Banu Oskan, Jay Taylor, and Jeremy Wideman for their helpful support on two separate PhD dissertation projects. The authors would also like to thank the associate editor and two reviewers for their helpful comments on earlier versions of this manuscript.

Conflict of interest: none declared.

References

Abascal F, Zardoya R, and Telford MJ. 2010. TranslatorX: multiple alignment of nucleotide sequences guided by amino acid translations. *Nucleic Acids Research* 38:W7–W13.

Allauzen C, Riley M, Schalkwyk J, Skut W, and Mohri M. 2007. Openfst: A general and

efficient weighted finite-state transducer library. In *International Conference on Implementation and Application of Automata*, pages 11–23. Springer.

Bininda-Emonds O. 2005. transAlign: using amino acids to facilitate the multiple alignment of protein-coding DNA sequences. *BMC Bioinformatics* 6:1–6.

Blackburne BP and Whelan S. 2011. Measuring the distance between multiple sequence alignments. *Bioinformatics* 28:495–502. ISSN 1367-4803.

Bradley RK and Holmes I. 2007. Transducers: an emerging probabilistic framework for modeling indels on trees. *Bioinformatics* 23.

Charif D and Lobry J. 2007. SeqinR 1.0-2: a contributed package to the R project for statistical computing devoted to biological sequences retrieval and analysis. In U Bastolla, M Porto, H Roman, and M Vendruscolo, editors, *Structural approaches to sequence evolution: Molecules, networks, populations*, Biological and Medical Physics, Biomedical Engineering, pages 207–232. New York: Springer Verlag.

Cotterell R, Peng N, and Eisner J. 2014. Stochastic contextual edit distance and probabilistic FSTs. In K Toutanova and H Wu, editors, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (*Volume 2: Short Papers*), pages 625–630. Baltimore, Maryland: Association for Computational Linguistics.

Fletcher W and Yang Z. 2010. The effect of insertions, deletions, and alignment errors on the branch-site test of positive selection. *Molecular Biology and Evolution* 27:2257–2267.

García Mesa JJ. 2023. Statistical sequence alignment of protein coding regions. Ph.D. thesis, Arizona State University.

Holmes I and Bruno WJ. 2001. Evolutionary HMMs: a bayesian approach to multiple alignment. *Bioinformatics* 17:803–820.

Hubbard T, Barker D, Birney E, Cameron G, Chen Y, Clark L, Cox T, Cuff J, Curwen V, Down T, et al. 2002. The ensembl genome database project. *Nucleic Acids Research* 30:38–41.

Hubisz MJ, Lin MF, Kellis M, and Siepel A. 2011. Error and error mitigation in low-coverage genome assemblies. *PLoS One* 6:e17,034.

Jackman SD, Coombe L, Chu J, Warren RL, Vandervalk BP, Yeo S, Xue Z, Mohamadi H, Bohlmann J, Jones SJ, et al. 2018. Tigmint: correcting assembly errors using linked reads from large molecules. *BMC Bioinformatics* 19:1–10.

Katoh K and Standley DM. 2013. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Molecular Biology and Evolution* 30:772–780.

Kimura M. 1980. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution* 16:111–120.

Kosiol C, Holmes I, and Goldman N. 2007. An empirical codon model for protein sequence evolution. *Molecular Biology and Evolution* 24:1464–1479.

Li WH. 1993. Unbiased estimation of the rates of synonymous and nonsynonymous substitution. *Journal of Molecular Evolution* 36:96–99.

Li WH, Wu CI, and Luo CC. 1985. A new method for estimating synonymous and nonsynonymous rates of nucleotide substitution considering the relative likelihood of nucleotide and codon changes. *Molecular Biology and Evolution* 2:150–174. ISSN 0737-4038.

Löytynoja A. 2014. Phylogeny-aware alignment with PRANK. In *Multiple Sequence Alignment Methods*, pages 155–170. Springer.

Lunter G, Drummond AJ, Miklós I, and Hein J. 2005. Statistical alignment: Recent progress, new applications, and challenges. In *Statistical Methods in Molecular Evolution*, pages 375–405. New York: Springer-Verlag.

Mohri M, Pereira F, and Riley M. 2005. Weighted automata in text and speech processing.

Morrison DA. 2015. Is sequence alignment an art or a science? *Systematic Botany* 40:14–26.

Muse SV and Gaut BS. 1994. A likelihood approach for comparing synonymous and nonsynonymous nucleotide substitution rates, with application to the chloroplast genome. *Molecular Biology and Evolution* 11:715–724.

Pamilo P and Bianchi NO. 1993. Evolution of the Zfx and Zfy genes: rates and interdependence between the genes. *Molecular Biology and Evolution* 10:271–281. ISSN 0737-4038.

Paradis E and Schliep K. 2019. ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics* 35:526–528.

Ranwez V, Douzery EJP, Cambon C, Chantret N, and Delsuc F. 2018. MACSE v2: toolkit for the alignment of coding sequences accounting for frameshifts and stop codons. *Molecular Biology and Evolution* 35:2582–2584. ISSN 0737-4038.

Ranwez V, Harispe S, Delsuc F, and Douzery EJ. 2011. MACSE: Multiple Alignment of Coding SEquences accounting for frameshifts and stop codons. *PLoS One* 6:e22,594.

Rosenberg MS. 2009. *Sequence Alignment: Methods, Models, Concepts, and Strategies*. Univ of California Press.

Schneider A, Souvorov A, Sabath N, Landan G, Gonnet GH, and Graur D. 2009. Estimates of positive Darwinian selection are inflated by errors in sequencing, annotation, and alignment. *Genome Biology and Evolution* 1:114–118.

Sievers F, Wilm A, Dineen D, Gibson TJ, Karplus K, Li W, Lopez R, McWilliam H, Remmert M, Söding J, et al. 2011. Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Molecular Systems Biology* 7:539.

Silvestre-Ryan J, Wang Y, Sharma M, Lin S, Shen Y, Dider S, and Holmes I. 2021. Machine Boss: rapid prototyping of bioinformatic automata. *Bioinformatics* 37:29–35.

Taylor MS, Ponting CP, and Copley RR. 2004. Occurrence and consequences of coding sequence insertions and deletions in mammalian genomes. *Genome Research* 14:555–566.

Yoon BJ. 2009. Hidden Markov models and their applications in biological sequence analysis. *Current Genomics* 10:402–415.

Zhu Z. 2022. Profiling of indel phases in coding regions. Ph.D. thesis, Arizona State University.