COATi: statistical pairwise alignment of protein-coding sequences

Juan J. Garcia 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. Additionally these aligners force gaps to occur between codons, whereas in natural sequences, only about 42% of indels occur between codons (Taylor et al., 2004; Zhu, 2022). This mismatch between aligner 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

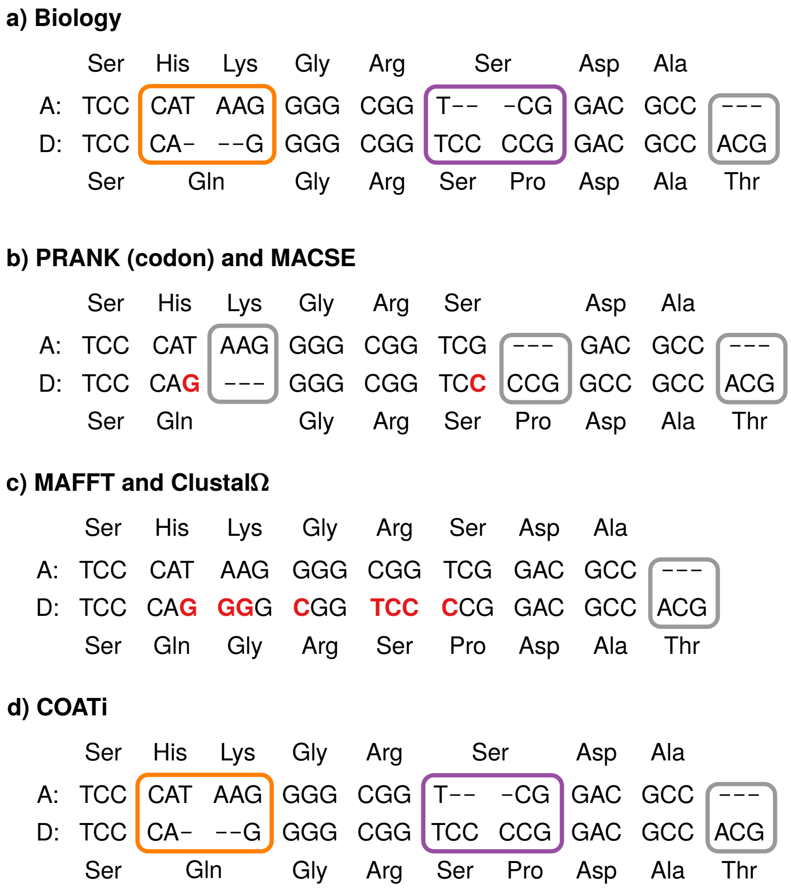
a) Biology
Ser His Lys Gly Arg Ser Asp Ala
A: TCC CAT AAG GGG CGG T-- -CG GAC GCC ---
D: TCC CA- --G GGG CGG TCC CCG GAC GCC ACG
Ser Gln Gly Arg Ser Pro Asp Ala Thr
b) PRANK (codon) and MACSE
Ser His Lys Gly Arg Ser Asp Ala
A: TCC CAT AAG GGG CGG TCG --- GAC GCC ---
D: TCC CAG --- GGG CGG TCC CCG GCC GCC ACG
Ser Gln Gly Arg Ser Pro Asp Ala Thr
c) MAFFT and ClustalΩ
Ser His Lys Gly Arg Ser Asp Ala
A: TCC CAT AAG GGG CGG TCG GAC GCC ---
D: TCC CAG GGG CGG TCC CCG GAC GCC ACG
Ser Gln Gly Arg Ser Pro Asp Ala Thr
d) COATi
Ser His Lys Gly Arg Ser Asp Ala
A: TCC CAT AAG GGG CGG T-- -CG GAC GCC ---
D: TCC CA- --G GGG CGG TCC CCG GAC GCC ACG
Ser Gln Gly Arg Ser Pro Asp Ala Thr

Figure 1: Standard algorithms produce suboptimal alignments. (a) shows the true alignment of an ancestor sequence (A) and a descendant sequence (D). (b), (c), and (d) are the results of different aligners. Nucleotide mismatches are highlighted in red. 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). COATi is the only aligner able to retrieve the biological alignment in this example.

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

In statistical alignment, sequence alignments are scored based on a stochastic model, typically derived from molecular evolutionary processes (Lunter et al., 2005). While approaches vary, a statistical aligner for a pair of sequences X and Y typically finds an alignment A that maximizes the joint probability $P(X, Y, A)$ or samples alignments from the posterior $P(A|X, Y)$. 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 a sequence, and a path through a pair-HMM is a possible pairwise alignment. Conceptually, these machines generate two sequences from an unknown ancestor and can calculate the probability that two sequences are related, i.e. $P(X, Y)$ (Yoon, 2009).

While the use of pair-HMMs is ubiquitous in bioinformatics, their

A limitation of pair-HMMs is that they only model the evolution of two related sequences from an unknown ancestor. Finite-state transducers (FSTs) have similar benefits to pair-HMMs with the additional feature that they can model the generation of a descendant sequence given an ancestral one. FSTs consume symbols from an input tape and emit symbols to an output tape. Properly weighted, an FST can calculate the probability that a descendant sequence, Y, evolved from an ancestral sequence, X, represented by $P(\mathrm{Y}|\mathrm{X})$.

Furthermore, there are well-established algorithms for combining FSTs in different ways, including concatenation, composition, intersection, union, or reversal, allowing the design of complex models by combining simpler FSTs (Bradley and Holmes, 2007; Silvestre-Ryan et al., 2021). Specifically, composition is a powerful and versatile algorithm for comparative sequence analysis, consisting of sending the output of one FST into the input of a second FST. Composition allows combining two or more FSTs to create a new, more complex transducer. Figure **??** illustrates how DNA transcription for a codon can be achieved by composing a complementing FST with a transducer that replaces thymines with uracil, where the three nucleotides are read and complemented in **??**-a, which are then used as the input of **??**-b, resulting in the transcription of the codon (Fig. **??**-c). COATi uses composition to derive a statistical alignment model from the combination of smaller FSTs, each representing a specific process.

COATi implements the pairwise alignment of a potentially lower-quality sequence against a high-quality sequence as a path through the Evolution FST (Fig. 3) (c.f. Holmes and Bruno, 2001). Here, COATi treats the high-quality (reference) sequence as the "ancestor" and the potentially lower-quality sequence as the "descendant". The assumption is that the reference sequence is in-phase, which is used to help preserve the reading frame and safeguard against possible frameshifts in the "descendant". Therefore, the high-quality sequence must not contain incomplete codons (the number of nucleotides is multiple of three) and be free of any ambiguous nucleotides or early stop codons. In contrast, the potentially lower-quality sequence has no such restrictions and can be of any length, contain early stop codons—treated as artifacts—, and include ambiguous codons.

The Evolution FST is the result of composing a substitution FST that encodes a codon model (Fig. 3-a) and an indel FST that models insertions and deletions, including frameshifts (Fig. 3-b). A key innovation of this FST, with respect to others, is the combination of a

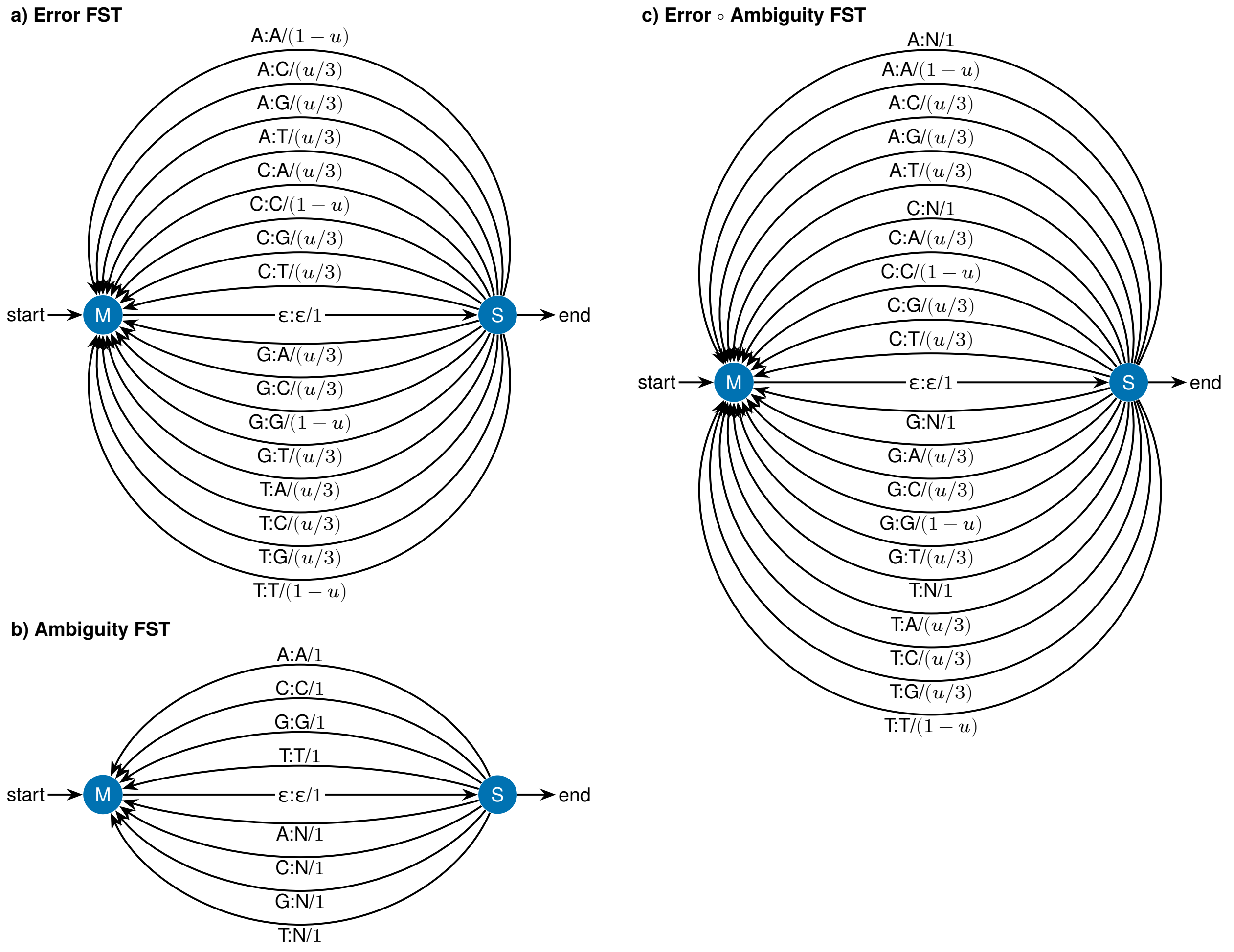

Figure 2: Finite state transducers (FSTs) model the generation of an output sequences based on an input sequence. (a) A graph of a probabilistic FST (Cotterell et al., 2014) for base-calling errors using a Mealy-machine architecture, where parameter u is the error rate. This graph contains two states (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.

codon substitution model with a nucleotide-based geometric indel model that allows gaps to occur at any position.

Composing both sequences with the Evolution FST results in the transducer of all possible alignments. Any path through this FST represents a pairwise alignment, while the shortest path (by weight) corresponds to the best alignment. When equally optimal alignments exist, ties are broken according to the FST shortest-path algorithm. An example

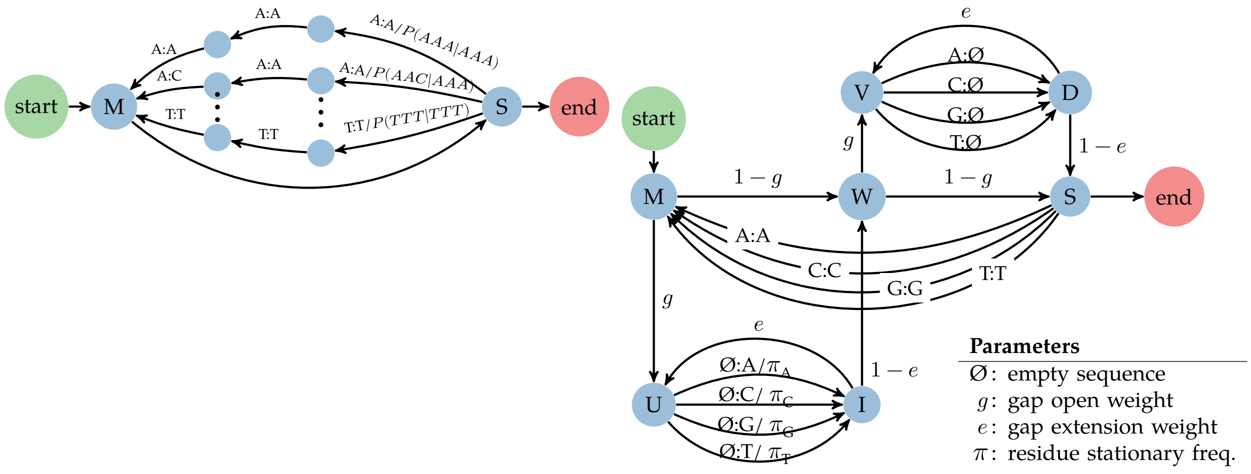

Figure 3: The Evolution FST is assembled by composing a substitution FST and an indel FST. Each node represents a state in an FST while arcs display possible transitions between states (and their weights). Unlabeled arcs have weights of 1. (a) The substitution FST encodes a 61 × 61 codon substitution 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. (b) The indel FST allows for insertions (U to I) and deletions (V to D). Insertion arcs are weighted according to the codon model's stationary distribution of nucleotides, and deletion arcs have a weight of 1. On top of the indel FST, we add a base-calling-error FST (Supplemental Materials Figure 1) to model sequencing errors. Contiguous insertions and deletions are always arranged for insertions to precede deletions to limit equivalent alignments.

of an FST-based alignment can be found in Supplementary Materials Figure 2. 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). However, the Evolution FST has a large state space to keep track of codon substitution rates when codons might be interspersed with indel events. This additional state space increases the computational complexity of the alignment algorithm.

Codon Substitution Models

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 (Kosiol et al., 2007) (codon-triplet-ecm). It also lets the user provide a codon substitution matrix. The default FST model (codon-triplet-mg) does not allow early stop codons in the ancestor sequence; although, it does support mutations to (early) stop codons under the assumption that these are artifacts common in low-quality data.

To reduce the runtime complexity of COATi, we have also developed an approximation of the Evolution 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 standard codon model, and

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

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 (default model) 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 Simulation Algorithm

I downloaded 16000 human genes and their gorilla orthologs from the ENSEMBL database (Hubbard et al., 2002). After downloading, I removed 2232 sequence-pairs longer than 6000 nucleotides and aligned the remaining pairs with all five methods. At least one aligner added gaps to 6048 sequence pairs, and no aligner added gaps to 7761 sequence pairs. Then, I randomly introduced gap patterns extracted from all five methods into the ungapped sequence pairs to generate the benchmark alignments.

The simulation algorithm can introduce a pairwise alignment pattern to any two nucleotide sequences of equal length. The alignment pattern is given as a CIGAR string (Compact Idiosyncratic Gapped Alignment Report), a format commonly used to summarize aligned reads to a reference genome. Assigning one of the sequences as the reference, to distinguish between insertions and deletions, CIGAR strings can also summarize pairwise alignments by grouping the number of contiguous matches or mismatches 'M', deletions 'D', and insertions 'I'. The resulting pattern combines these letters preceded by the number of characters for each section as they appear in the alignment. This pattern is introduced by replacing nucleotides with gaps as indicated by deletions on one sequence and randomly introducing residues where the CIGAR strings indicated insertions.

I created the benchmark of alignments by using an equal number of randomly sampled gap patterns from each aligner. I used the dataset to evaluate the accuracy of COATi and a suite of popular aligners spanning various alignment methods: ClustalΩ v1.2.4 (Sievers et al., 2011), MACSE v2.06 (Ranwez et al., 2011), MAFFT v7.505 (Katoh and Standley, 2013), and PRANK v.150803 (Löytynoja, 2014).

Metrics

To quantify the similarity between each alignment in the benchmark and the corresponding output obtained by the different tools, I used the alignment error metric d_{seq} (Blackburne and Whelan, 2011). This metric accounts for indels and is more informative than conventional distance scores like sum-of-pairs or total columns. Intuitively, d_{seq} ranges between zero and one and can be interpreted as the probability that a randomly selected residue will be aligned to a different location against a sequence that does not contain such residue.

The computation of d_{seq} involves characterizing the gaps present in the alignment. Then, a site-wise homology set $H(A)^i_j$ is calculated for each alignment A, sequence i, and character j. The distance between two alignments A and B is the average across all characters of the symmetric difference (or Hamming distance), represented as '$\triangle$' between homology sets over the length of such sets:

$$d(A,B) = \frac{1}{c}\sum_i \sum_j \frac{|H(A)^i_j \triangle H(B)^i_j|}{|H(A)^i_j| + |H(B)^i_j|} \tag{1}$$

where c is the sum of the sequence lengths.

In addition, I compared the number of perfectly and imperfectly retrieved alignments for each aligner. Perfect alignments are defined as those with a distance of zero to the reference alignment ($d_{seq} = 0$), indicating 100% similarity. Notably, a set of sequences can have more than one optimal alignment under the same evolutionary model (same score), despite algorithms typically producing a single result. Consequently, to account for evolutionary equivalent alignments, I scored all alignments using the marginal model and also considered perfect those with scores identical to the benchmark. Furthermore, I counted the number of alignments with the lowest distance d_{seq} to the true alignment, including ties, reported as best alignments. Moreover, I computed the count of imperfect alignments, where an alignment is considered imperfect when its distance to the reference alignment is greater than zero ($d_{seq} > 0$) and another method successfully produced an alignment with 100% similarity. This analysis exposes instances where all aligners fall short of achieving a perfect result in addition to a direct comparison.

To evaluate how well the aligners were able to identify positive and negative selection, I estimated k_s and k_a statistics. k_s and k_a are, respectively, the number of substitutions per synonymous site (no changes at the amino acid level) and per non-synonymous site (introduces changes at the amino acid level) between two protein-coding genes. They are also denoted as d_s and d_n in the literature. I used the R package seqinr (Charif and Lobry, 2007) to estimate these metrics, which follows the popular method put forth by Li (Li, 1993). First, this method takes two aligned homologous protein-coding sequences and classifies the nucleotide sites in a sequence as nondegenerate, twofold degenerate, and fourfold degenerate. A site is nondegenerate if all possible changes at that site are nonsynonymous, twofold degenerate if one of the three possible changes is synonymous, and fourfold degenerate if all possible changes are synonymous. Second, the nucleotide changes between the two sequences are counted and divided as transitional (A↔G, C↔T) and transversional ({A, G}↔{C, T}). Third, the Kimura two-parameter distance (Kimura, 1980) is used to estimate the number of transitions and transversions per site type (nondegenerate, twofold degenerate, and fourfold degenerate), which is used as a correction factor for multiple hits. Finally, k_s is the estimate of the average transitional rate at twofold and fourfold degenerate sites, and k_a is the estimate of the average transversional rate at nondegenerate and twofold sites. In the results, these metrics are reported as the F_1 score, which is the harmonic mean of precision (true positives over total positives) and recall (true positives over true positives and false negatives). This score ranges between 0 and 1, with a score of 1 representing a perfect result.

In addition, we compared the estimated evolutionary distance between the reference and inferred alignments using the Kimura 2-parameter model (Kimura, 1980). The calculation of the Kimura 2-parameter model involves considering the rates of nucleotide transitions and transversions. The formula used for distance calculation is $D = -0.5 \cdot \log((1 - 2P - Q) \cdot \sqrt{1 - 2Q})$, where P represents the proportion of transitional substitutions and Q represents the proportion of transversional substitutions. The resulting distance provides a quantitative measure of the evolutionary divergence between the sequences. These calculations were performed using the R software package ape (Paradis and Schliep, 2019).

Results and Discussion

COATi, using the codon-triplet-mg model, 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). 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 is substantially less overestimated than other methods (Table 1, Supplemental Materials Figure 8).

ClustalΩ generated alignments via amino acid translations and obtained the highest average alignment error while having difficulties retrieving positive selection. MACSE used a DNA-AA hybrid model, allowing frameshifts, and obtained similar results to MAFFT using a DNA model. PRANK, using a codon model, had an average alignment error between MACSE/MAFFT and ClustalΩ but was unable to generate alignments for some sequence pairs.

	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
F1 score for positive selection	98.1%	84.3%	86.7%	79.5%	68.7%
F1 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.

To test how well COATi performs when the roles of reference and low-quality sequence are reverted, I aligned the 7761 simulated alignments using gorilla as the reference. Notably, COATi was only able to align 4003 sequence pairs due to the presence of early stop codons in the gorilla sequence on the remaining alignments. 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, I analyzed the 4003 alignments and compared the results with all 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 ancestor in comparison to the reverse (Supplementary Materials Tab. 6). However, the results for COATi continue to be a significant improvement over other aligners.

Despite human and gorilla sequences having a relatively short evolutionary distance, COATi showed a biologically significant improvement over other methods, with an average alignment error nine-fold smaller than the next best method. COATi is an FST-based application that can calculate the optimal alignment between a pair of sequences in the presence of artifacts using a statistical model. Using COATi will allow researchers to analyze more data with higher accuracy and facilitate the study of important biological processes that shape genomic data.

Future work include extending the indel FST to combine a 3-mer gap model with a frameshift parameter and weighing each indel phase differently to reflect known selection on indel phases (Zhu, 2022). We also plan on comparing the marginal and triplet models to evaluate the implications of the marginalization.

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.

TODO: Reviewers, Editor, and COmmittee members

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.

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.

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.

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.

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, 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.