An alignment method for nucleic acid sequences against annotated genomes

Koen Deforche[1]

[1]Emweb bvba, 3020 Herent, Belgium

October 10, 2017

Abstract

Motivation: Biological sequence alignment is fundamental to their further interpretation. Current alignment algorithms typically align either nucleic acid or amino acid sequences. Using only nucleic acid sequence similarity, divergent sequences cannot be aligned reliably because of the limited alphabet and genetic saturation. To align divergent coding nucleic acid sequences, one can align using the translated amino acid sequences. This requires the detection of the correct open reading frame, is prone to eventual frame shift errors, and typically requires the treatment of genes separately. It was our motivation to design a nucleic acid sequence alignment algorithm to align a nucleic acid sequence against a (reference) genome sequence, that works equally well for similar and divergent sequences, and produces an optimal alignment considering simultaneously the alignment of all annotated coding sequences.

Results: We define a *genome alignment score* for evaluating the quality of an alignment of a nucleic acid query sequence against a reference genome sequence, for which coding sequence features have been annotated (for example in a GenBank record). The genome alignment score combines the affine gap score for the nucleic acid sequence with an affine gap score for all amino acid alignments resulting from coding sequences in open reading frames contained within the query sequence. We present a Dynamic Programming algorithm to compute the optimal global or local alignment using this genomic alignment score and provide a formal proof of correctness. This algorithm allows the alignment of nucleic acid sequences from closely related and highly divergent sequences within the same software and using the same parameters, automatically correcting any eventual frame shift errors and produces at the same time the aligned translated amino acid sequences of all relevant coding sequence features.

Availability: The software is available as a web application at `http://www.genomedetective.com/app/aga` and as command-line application at `https://github.com/emweb/aga`

1 Introduction

Obtaining nucleic acid sequences that span partial or full genomes are becoming cost-effective with recent advances in sequencing technology. This enables new applications that use use this genomic sequence data from known and unknown species in clinical or environmental samples [7] [15].

Alignment, as part of a database similarity search, or against a specific reference genome, is typically a first step in the interpretation of these sequences. Current alignment methods however work either on the nucleic acid or amino acid sequence, but typically do not consider both simultaneously. The benefit of aligning amino acid sequences is that they are usable for more diverged sequences, but they require the correct detection of open reading frames and a priori correction of eventual frame shift errors that interfere with proper translation. By considering only amino acid sequence similarity, the more sensitive evolutionary information within synonymous substitutions is however lost.

We describe here an algorithm (AGA) which computes the optimal pairwise alignment of an (unknown) nucleic acid sequence against a reference sequence, using a score that combines nucleic acid similarity and amino acid similarity. Amino acid similarity is based on translated Coding DNA Sequence (CDS) annotations of the reference sequence. In a typical application, the reference sequence would be a complete genome sequence annotated with the location of coding sequences of contained proteins or polyproteins (see for example Figure 2). The proposed method can deal with multiple coding sequences in different open reading frames, which use either the forward or the reverse complemented strand, and which may be composed of different regions that are spliced together. Open reading frames may also verlap, which is not uncommon for compact viral genomes.

The alignment of coding nucleic acid sequences using amino acid sequence similarity has been considered before [5] [1] [18] [18] [2] [19] [11] [13] [8] [17]. The method outlined in this paper differs with this previous work in the sense that it considers specifically the problem of how to optimally align a nucleic acid sequence against an annotated (reference) genome, considering simultaneously nucleotide simularity and amino acid sequence simularity in the annotated coding sequences. The method results in alignments with a minimum number of frame shifts in coding sequences and with gaps preferably at codon boundaries, by penalizing both such events in the scoring function that it optimizes. By combining similarity of the nucleic acid sequences and similarity of coded amino acid sequences, the method can optimally align sequences to a reference genome, regardless of whether they are highly similar or distant to the reference genome.

Applications such as phylogenic tree reconstruction, sequence similarity evaluation, read or contig mapping towards a reference sequence, or determination of nucleic acid and amino acid substitutions for genotypic/phenotypic assocations, all depend on a high quality alignment. We believe that AGA is useful to most of these applications.

2 Approach

We define a *Genome Alignment Score* which scores the quality of a local or global alignment of a nucleic acid *Query* sequence against a *Reference Genome*. In this context, a reference genome is a nucleic acid sequence annotated with *Coding DNA Sequence* information. Each coding sequence indicates the location of one or regions in the genome that jointly translate to a protein or polyprotein. The coding sequence may be based on a single region or may span multiple regions that are spliced together. Each region is part of one of three forward or reverse complemented open reading frames within the genome sequence. Coding Sequences may also overlap, sharing the same or different open reading frames, which is not unusual for compact viral genomes.

The genome alignment score is based on (a) the alignment score of the nucleic acid sequence using a traditional nucleic acid substitition matrix score, with affine gap open and gap extension penalties [16], (b) the alignment score of each of the covered coding sequences using a traditional amino acid substition matrix score, with affine gap open an gap extension penalties, and (c) additional penalties for frameshift insertions and deletions, and for insertions and deletions that do not align with codon boundaries.

By defining the genome alignment score in this way, it can be used to compare the similarity of a nucleic acid sequence to multiple reference genomes. By including the alignment score of the nucleic acid sequence itself, the genome alignment score is applicable to sequences that are highly similar to the reference, having for example only meaningful differences in their nucleic acid sequences, but virtually no changes in their amino acid sequences. By including the alignment score of all covered coding sequences as well, the alignment score is in particular also applicable to highly divergent sequences, which may have lost much of their similarity at the nucleic acid sequence, but which still share some similarity in their protein sequences, especially when considering all coding sequences (covered by the query sequence) simultaneously. Finally, by considering the possibility of (likely erroneous) insertions or deletions that cause frame shifts, the alignment score is suitable to align sequences obtained from sequencing techniques that are prone to such sequencing errors without misusing frameshift mutations artificially as a means to optimize amino acid sequence similarity.

We show how a Dynamic Programming algorithm can be designed which computes the optimal local or global alignment subject to maximizing the genome alignment score. This work thus builds further on the optimal alignment algorithms first proposed by Needleman-Wunsch [16], Smith-Waterman [16], and Gotoh [4], by expanding the induction state with additional state parameters.

3 Methods

3.1 Notation

Let the two nucleic acid sequences be a reference genome $\mathrm{G} = \mathrm{g}_1\mathrm{g}_2 \ldots \mathrm{g}_M$ and a query sequence $\mathrm{B} = \mathrm{b}_1\mathrm{b}_2 \ldots \mathrm{b}_N$. For the reference genome G, information on coding sequences are available, which are for example the CDS annotations from a GenBank record. This information can be represented as a list of codons in which the nucleotide at position m participates: $C_m = \{[c_{m,1}, r_{m,1}], [c_{m,2}, r_{m,2}], ..., [c_{m,t}, r_{m,t}]\}$ where $1 \leq c_{m,i} \leq 3$, indicating the position of the nucleotide in a codon, and $r_{m,i}$ a boolean indicating whether the codon is in the forward strand or in the reverse complementary strand. A splice site does not necessarily occur at a codon boundary, and in that case the nucleic acids that are translated to the codon may be scattered through the genome. To allow the algorithm below to process the sequences sequentially from start to end, we exclude such codons from the scoring models. We define $t_m = |C_m|$, the number of codons in which the nucleotide takes part. A value of $t_m > 1$ indicates that there are multiple overlapping coding sequences at the given nucleotide position m, possibly in different open reading frames. We denote as $T_i(\mathrm{a}, r)$ the translation of the codon $\mathrm{a}_i\mathrm{a}_{i+1}\mathrm{a}_{i+2}$ to an amino acid in the forward or reverse complementary strand depending on the value of r.

3.2 Genome alignment score

For a nucleic acid sequence alignment $A_{\mathrm{na}}(\mathrm{G}, \mathrm{B})$, we define a *genome alignment score* $S_{\mathrm{ga}}\{A_{\mathrm{na}}(\mathrm{G}, \mathrm{B})\}$ that is based on a nucleic acid sequence alignment score $S_{\mathrm{na}}\{A_{\mathrm{na}}(\mathrm{G}, \mathrm{B})\}$ for the nucleic acid sequence alignment $A_{\mathrm{na}}(\mathrm{G}, \mathrm{B})$ itself, and the amino acid sequence alignment score $S_{\mathrm{aa}}\{A_{\mathrm{aa}}(\mathrm{X}, \mathrm{Y})\}$ for each amino acid sequence alignments $A_{\mathrm{aa}}(\mathrm{X}, \mathrm{Y})$ that results from translation of the aligned sequences G and B according to the coding sequence annotations of G.

The alignment scores S_{na} and S_{aa} are of the same form but use a distinct set of parameters. They score a match in the alignment according to a substitution weight matrix W, and a gap of length k in the query or reference sequence using an affine gap model: $w_k = p_u(k-1) + p_v$ with $p_v \leq 0$ the penalty for opening a gap, and $p_u \leq 0$ is the cost for extending a gap. The incremental cost for a gap is then

$$\Delta w_k = \begin{cases} p_v & \text{if } k = 1 \\ p_u & \text{if } k > 1 \end{cases}$$

Because the genome alignment score, and the alignment algorithm described below, consider the insertion of gaps anywhere in the nucleic acid sequence, and gaps of any length, we need to define how these gaps within a coding sequence translate to gaps in the amino acid sequence. We make the choice to consider a gap of length k with $3(n-1) < k \leq 3n$ as a gap of length n in the amino acid sequence, regardless of whether the gap aligns with a codon boundary. For practical reasons, one typically prefers gaps to align with codon boundaries because otherwise they render the interrupted start and end codons as non-translatable, which lack a proper notation and scoring at the amino acid sequence level (usually denoted as an 'X' in the resulting sequence alignment). Therefore a misalignment penalty is added whenever a gap starts within a codon, disrupting proper translation of that codon. More formally, a penalty $p_m \leq 0$ is added whenever a gap opens after position g_m, either in the query or reference sequence, when C_m contains a codon position $c_{m,i} \neq 3$.

The amino acid sequence alignment score S_{AA} by itself does does not consider insertions and deletions in the underlying nucleic acid sequence that do not occur in a multiple of three. These however cause frameshift mutations which change the translation profoundly and thus has a large impact on the amino acid sequence alignment. Frame shifts are unexpected in a viable coding sequence, but they are not uncommon as a consequence of sequencing errors and thus their possibility needs to be considered (with low probability) to obtain a better alignment. The introduction of a frame shift needs thus to be weighted against the quality of the amino acid sequence alignment. Thereofre, in the genome alignment score, a frame shift penalty $p_f \leq 0$ is added for a gap of length k with $(k \bmod 3) \neq 0$.

Finally, the genome alignment score S_{ga} is defined as a weighted sum of the nucleic acid sequence alignment score and the amino acid sequence alignment scores, using a weight w_{aa} for the amino acid alignment score contribution.

3.3 Algorithm

We now describe an algorithm that calculates the optimal alignment, maximizing the above genome alignment score, using Dynamic Programming. The algorithm extends the idea developed in [4] to expand the induction state with additional state matrices to properly accomodate the affine gap cost in the induction step. In particular, in [4] the induction state was defined as [$D_{m,n}$, $P_{m,n}$, $Q_{m,n}$] in which $D_{m,n}$ is the best score for an alignment of $\mathrm{g}_1\mathrm{g}_2...\mathrm{g}_m$ versus $\mathrm{b}_1\mathrm{b}_2...\mathrm{b}_n$, $P_{m,n}$ the best score for an alignment of the same sequences but ending with a gap after g_m, and $Q_{m,n}$ the best score for an alignment of the same sequences but ending with a gap after b_m. These additional two matrices $P_{m,n}$ and $Q_{m,n}$ allow the algorithm to properly evaluate the score for a gap taking into account the different cost for a gap open and gap extension. With the genome alignment score as defined above, the cost for opening a gap will be different from a gap extension, not only because of the affine gap penalty present in the nucleic acid sequence and amino acid sequence alignment scores, but also because opening a gap may break one or more codons and cause a codon misalignment cost p_m. But also the cost of a gap extension differs from one to another since the length of the gap influences the occurence of a frame shift penalty p_f, and depending on the position relative to a codon, a gap penalty may be added to the amino acid alignment.

We define as induction state [$D_{m,n}$, $M_{m,n}$, $P^{(g)}_{m,n}$, $Q^{(h)}_{m,n}$] with $1 \leq g \leq 3$ and $1 \leq h \leq 3$ where for an alignment of $\mathrm{g}_1\mathrm{g}_2...\mathrm{g}_m$ versus $\mathrm{b}_1\mathrm{b}_2...\mathrm{b}_n$:

- $D_{m,n}$ is the best score overall;
- $M_{m,n}$ is the best score for an alignment ending with a match;
- $P^{(g)}_{m,n}$ is the best score for an alignment ending with a gap of length $k = 3i + g$ after g_m with $i \geq 0$ in the reference sequence;
- $Q^{(h)}_{m,n}$ is the best score for an alignment ending with a gap of length $k = 3j + h$ after b_n with $j \geq 0$ in the query sequence.

At each induction step m, n the induction state is updated using an incremental computation of the genomic alignment score. Let $\Delta d_{m,n}$ be the incremental score for extending the alignment of $\mathrm{g}_1\mathrm{g}_2...\mathrm{g}_{m-1}$ with $\mathrm{b}_1\mathrm{b}_2...\mathrm{b}_{n-1}$ with a match. $\Delta p_{m,n}(k)$ is the incremental score for extending an alignment of $\mathrm{g}_1\mathrm{g}_2...\mathrm{g}_m$ with $\mathrm{b}_1\mathrm{b}_2...\mathrm{b}_{n-1}$ by opening a gap $(k = 1)$ or extending a gap in the reference sequence to length k. Likewise let $\Delta q_{m,n}(k)$ be the incremental score for extending an alignment of $\mathrm{g}_1\mathrm{g}_2...\mathrm{g}_{m-1}$ with $\mathrm{b}_1\mathrm{b}_2...\mathrm{b}_n$) by opening a gap $(k = 1)$ or extending the gap in the query to length k.

$$\Delta d_{m,n} = W_{\mathrm{na}}(\mathrm{g}_m, \mathrm{b}_n) + w_{\mathrm{aa}} \sum_{1 \leq i \leq t_m} \chi_{m,n}(c_{m,i}, r_{m,i})$$

The sum component in the above equation adds the amino acid substitution score for each codon which starts at position m, and thus for which $c_{m,i} = 1$.

$$\chi_{m,n}(c, r) = \begin{cases} W_{\mathrm{aa}}(T_n(\mathrm{g}, r), T_n(\mathrm{b}, r)) & \text{if } c = 1 \\ 0 & \text{if c} \neq 1 \end{cases}$$

$$\Delta p_{m,n}(k) = \Delta w_{\mathrm{na},k} + w_{\mathrm{aa}} \sum_{1 \leq i \leq t_{m+1}} \eta_{m+1,n}(c_{m+1,i}, r_{m+1,i}, k)$$

The sum component adds a score component η for each amino acid sequence which may be affected by the gap.

$$\eta_{m,n}(c, r, k) = \nu_{m,n}(c, r, k) + \omega(k) + \phi(k)$$

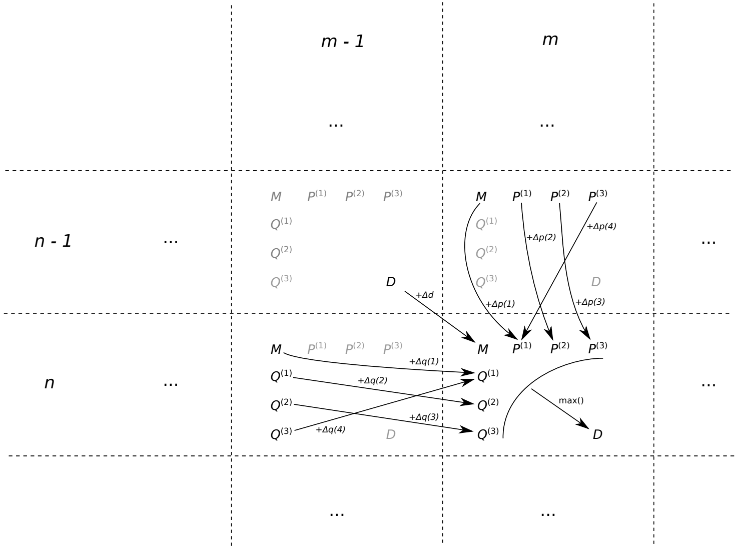
The codon breakage cost ν undoes an amino acid substitution weight previously added at the beginning of the codon, plus adds a misalignment penalty :

$$\nu_{m,n}(c, r, k) = \begin{cases} -W_{\mathrm{aa}}(T_{m-c+1}(\mathrm{g}, r), T_{n-c+1}(\mathrm{b}, r)) + p_m \\ \qquad\qquad\qquad\qquad \text{if } c \neq 1 \text{ and } k = 1 \\ 0 \qquad\qquad\qquad\qquad\quad \text{if } c = 1 \text{ or } k \neq 1 \end{cases}$$

An incremental amino acid open gap cost is added when opening the gap, and an amino acid gap extension cost is added for every additional 3 nucleic acid gaps, according to the affine gap model:

$$\omega(k) = \begin{cases} \Delta w_{\mathrm{aa},(k-1)/3+1} & \text{if } (k \bmod 3) = 1 \\ 0 & \text{if } (k \bmod 3) \neq 1 \end{cases}$$

Figure 1: Data dependencies in induction rule



A frame shift penalty is incrementally updated based on the length of the gap k:

$$\phi(k) = \begin{cases} p_f & \text{if } (k \bmod 3) = 1 \\ 0 & \text{if } (k \bmod 3) = 2 \\ -p_f & \text{if } (k \bmod 3) = 0 \end{cases}$$

Likewise we can define the incremental score for opening or extending a gap in the query sequence:

$$\Delta q_{m,n}(k) = \Delta w_{\text{na},k} + w_{\text{aa}} \sum_{1 \leq i \leq t_m} \eta_{m,n+1}(c_{m,i}, r_{m,i}, k)$$

At each induction step, we use the above incremental score update functions to update the induction state (Figure 1):

$$M_{m,n} = D_{m-1,n-1} + \Delta d_{m,n}$$

$$P^{(1)}_{m,n} = \max[M_{m,n-1} + \Delta p_{m,n}(1), P^{(3)}_{m,n-1} + \Delta p_{m,n}(4)]$$

$$P^{(i)}_{m,n} = P^{(i-1)}_{m,n-1} + \Delta p_{m,n}(i) \text{ for } 2 \leq i \leq 3$$

$$Q^{(1)}_{m,n} = \max[M_{m-1,n} + \Delta q_{m,n}(1), Q^{(3)}_{m-1,n} + \Delta q_{m,n}(4)]$$

$$Q^{(i)}_{m,n} = Q^{(i-1)}_{m-1,n} + \Delta q_{m,n}(i) \text{ for } 2 \leq i \leq 3$$

$$D_{m,n} = \max[M_{m,n}\ , P^{(i)}_{m,n}\ , Q^{(i)}_{m,n}] \text{ for } 1 \leq i \leq 3$$

The optimal alignment is retrieved by tracking back the path through the induction state matrix from $D_{m,n}$ to $D_{0,0}$ following back the path that led to the optimal solution.

3.4 Proof

At each induction step m, n we need to prove that the definitions of the induction state parameters are satisfied.

The definition of $M_{m,n}$ is satisfied since $\Delta d_{m,n}$ only depends on m and n and thus the optimal alignment ending in a match is the alignment that extends the optimal alignment of $\mathrm{g}_1\mathrm{g}_2...\mathrm{g}_{m-1}$ versus $\mathrm{b}_1\mathrm{b}_2...\mathrm{b}_{n-1}$ with score $D_{m-1,n-1}$.

The defintions of $P^{(g)}_{m,n}$ and $Q^{(g)}_{m,n}$ for $g = 2$ or $g = 3$ are satisfied because the incremental gap cost terms $\Delta p_{m,n}(k)$ and $\Delta q_{m,n}(k)$ depend only on m, n, and gap length k. When extending a gap ($k \neq 1$), it can easily be verified that the following equalities exist:

$$\Delta p_{m,n}(k) = \Delta p_{m,n}(k + 3i)$$

$$\Delta q_{m,n}(k) = \Delta q_{m,n}(k + 3i)$$

This means that the cost for extending a gap to length $k = 3i + g$ is the same for any value of i, and thus the highest score for an alignment ending with a gap of length $k = 3i+g$ is the highest score for an alignment ending with a gap of length $k = 3i + (g - 1)$, incremented with the gap extension cost for k.

The definitions of $P^{(1)}_{m,n}$ and $Q^{(1)}_{m,n}$ are satisfied because the algorithm uses the maximum score considering either the opening of a gap or extending of a gap of length $k = 3i$. The cost for opening a gap $\Delta p_{m,n}(1)$ and $\Delta q_{m,n}(1)$ depend only on m and n, and thus the highest score for ending the alignment with a gap of length $k = 3i + 1$ is either the cost for opening a gap added to the highest score for ending with a match, or for extending a gap of length $k = 3i$.

Finally, the optimal alignment score $D_{m,n}$ is defined as the maximum alignment score considering the different options of ending the alignment in a match, or ending with a gap in the reference of all possible lengths, or ending with a gap in the query of all possible lengths.

4 Implementation

The above algorithm was implemented in C++11 as a standalone command-line tool (AGA). The inputs are a reference genome (GenBank record) and a query nucleic acid sequence (FASTA file). The implementation does not keep the entire induction state matrix in memory by encoding the backtrace information within the induction state variables, and is thus in practice suitable for genome lengths up to 10^6 base pairs (most viral genomes), provided sufficient computation time.

Through command-line parameters, the following parameters of the algorithm may be configured: the choice of nucleic acid and amino acid substitution weight matrices W_{na} and W_{aa}, the affine gap parameters $p_{u,\mathrm{na}}$, $p_{v,\mathrm{na}}$, $p_{u,\mathrm{aa}}$, $p_{v,\mathrm{aa}}$, parameters to weight the nucleic acid versus amino acid score w_{aa}, and frame shift penalty p_f and misalignment penalty p_m. For the nucleic acid substition weight matrix, a score for a match and a score

Table 1: Algorithms whose performance was compared to AGA.

Name	Version	Score	Input	Description
needle	6.6.0	NT	pair-wise	no special support for CDS
MACSE	0.9b1	AA	multiple	one CDS, minimizes frame shifts
codaln	1.0	NT + AA	multiple	multiple CDS

Table 2: Sequences used in the evaluation of AGA.

Name	Accession	Length	Description
HXB2	NC_001802	9181	HIV-1 reference strain (Subtype B)
hiv1b	AY835761	9824	HIV-1 Subtype B complete genome
hiv1c	U46016.1	9031	HIV-1 Subtype C complete genome
siv	KF304708.1	9449	SIV complete genome
hiv2	KP890355.1	9480	HIV-2 complete genome

for a mismatch can be configured. As amino acid substitution weight matrices the tool offers the choice between BLOSUM30 and BLOSUM62 [6]. AGA can compute an optimal local or global alignment.

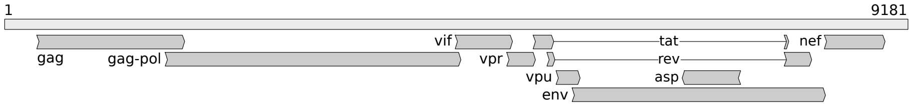
The tool outputs the nucleic acid sequence alignment and all amino acid sequence alignments of coding sequences (after eventual frame shift corrections). For each alignment, it outputs the corresponding score and provides various statistics (coverage length, number of matches, identities, indels, frame shifts, and codon misalignments).

5 Evaluation

To assess the benefit of the algorithm, we compared it to an implementation of a an optimal nucleic acid sequence alignment which doesn't take into account CDS features (EMBOSS needle [14]), and other alignment algorithms designed to take into account CDS features (Table 1). For the comparison, we evaluated how these algorithms perform to align different HIV and other primate lentivirus genome sequences against the HIV-1 reference sequence (Table 2).

HIV-1 was chosen for this evaluation since it has a complex genome organization (Figure 2), which is not uncommon for RNA viruses, with multiple overlapping reading frames, some of which are joined by different

Figure 2: HIV-1 Genome Organization



Coding DNA Sequence (CDS) annotations of HIV-1, as derived from the GenBank record NC_001802 (HIV-1 reference sequence)

Table 3: Parameter values used for AGA in its evaluation.

Parameter	**Value**
W_{aa}	BLOSUM62
W_{na}	+2 for match, -2 for mismatch
$p_{u,\text{na}}$	-1
$p_{v,\text{na}}$	-10
$p_{u,\text{aa}}$	-2
$p_{v,\text{aa}}$	-6
w_{aa}	1
p_f	-100
p_m	-20

Table 4: Command-line arguments used for other algorithms used in the evaluation.

Name	**Configuration**
needle	-gapopen 10 -gapextend 1 -datafile EDNASIMPLE2
MACSE	-g -10 -x -1 -f -100 -d 1 -s -10
codaln	-go 10 -gx 1 -m0 2 -m1 2

genomic segments, and including one that uses the complementary strand (for the *asp* gene). Although one routinely will align HIV-1 sequences to the reference genome, also the alignment of HIV-2 and SIV strains against HIV-1 may be useful to unravel genetic causes for the different phenotypic properties of these viruses [12] [3].

To evaluate the quality of each alignment, we used as meaningful statistics number of frame shifts introduced within coding sequences, the number of amino acids not aligned at a codon boundary, and the affine gap model score for the nucleic acid sequences and all amino acid sequences (excluding frame shift and misalignment penalties). The run time performance was also compared on a Dell XPS laptop.

Tables 3 and 4 detail value parameters used for running the different algorithms in the comparison, which we tried to make comparable despite differences in scoring models.

We also compared the different alignments using an alignment viewer that shows simultaneously the nucleic acid sequence alignment and all coding sequence alignments (this alignment viewer is also part of the web version othe tool).

6 Results

AGA produced alignments with high nucleic acid and amino acid alignment scores while introducing only a minimum amount of frame shifts and codon misalignments within coding sequences (Table 5). Because MACSE makes the assumption of a single CDS, it introduces a frame shift near the beginning or the end of a coding sequence region, or within

Table 5: Alignment quality of AGA compared to other algorithms

Sequence	**Algorithm**	**# FS**	**# MA**	**NT Score**	**AA Score**
hiv1b	AGA	0	10	16182	18130
	needle	3	20	16219	18054
	MACSE	13	29	15521	18013
	codaln	3	16	15615	18035
hiv1c	AGA	1	19	12478	16127
	needle	22	54	12525	16097
	MACSE	13	42	11772	15813
	codaln	4	21	11930	15907
siv	AGA	2	23	1497	8536
	needle	146	298	1980	8159
	MACSE	13	66	62	8301
	codaln	5	38	34	7552
hiv2	AGA	1	30	1164	8080
	needle	161	338	1650	7472
	MACSE	18	78	213	7799
	codaln	1	27	-85	6983

Performance of other algorithms compared to AGA for aligning four genome sequences against the HIV-1 reference strain (HXB2): *# FS* is the number of frame shifts caused by indels within Coding DNA Sequences (CDS) in either the reference or query sequence; *# MA* is the number of indels not aligned to a codon boundary; *NT Score* is the nucleic acid affine gap model score of the alignment; *AA Score* is the sum of all amino acid affine gap model scores of the translated CDS alignments, after inserting additional gaps to correct eventual frame shift errors.

an overlapping region, to jump to the new gene open reading frame, even for the alignment of the highly similar HIV-1 subtype B sequence, but otherwise also effectively minimizes the number of frame shifts. Codaln on the other hand, like AGA, uses its knowledge of open reading frames to minimize frame shifts within Coding Sequences, but the alignments are of a lower quality (at both the nucleic acid and amino acid level) because of an inaccurate estimation of open reading frames in the query sequence.

EMBOSS needle produces alignments with a maximum nucleic acid alignment score, as can be expected, but it will introduce frame shifts and codon misaligned gaps in order to optimize its nucleic acid sequence alignment.

AGA as well as the other algorithms tested, implement a Dynamic Programming solutions with a time complexity $O(MN)$, and with the exception of MACSE, all had a similar run time performance (Table 6). Taking into account the possibility of multiple overlapping open reading frames, AGA has an $O(MNT)$ time complexity, where T is the maximum amount of overlapping coding sequences within the genome: $T = \max_i t_i$.

Table 6: Runtime performance of AGA compared to other algorithms

Algorithm	**Run time (s)**
AGA	6.3
needle	4.5
MACSE	185.2
codaln	5.5

Run time performance of other algorithms compared to AGA for aligning an HIV-1 Subtype B full genome sequence (hiv1b, 9824 bps) against the HIV-1 reference strain (HXB2, 9181bps) on a Dell XPS (Intel Core i7-7500U CPU @ 2.70GHz).

7 Discussion

The proposed genomic alignment score combines the affine gap model score of the nucleic acid sequence alignment with all affine gap model scores of amino acid alignments. These scores are however not independent: a gap in an amino acid sequence alignment will also correspond to a gap in the nucleic acid sequence alignment. A matching amino acid in the amino acid sequence alignment may correspond to matching codons in the nucleic acid sequence alignment. Nevertheless the approach to allow gaps to be penalized at both levels was chosen since the cost for the introduction of the gaps is also weighted against the score for character matches in the alignments at each level. The parameter w_{aa} can be used to give more or less weight to the amino acid sequence alignments compared to the nucleic acid sequence alignment and a suitable value will depend also on the nature of both scoring models since these are dimensionless numbers that are not necessarily of the same order of magnitude and thus comparable.

The original motivation for affine gap costs in nucleic acid sequence alignments was in part to avoid gaps that introduce frame shifts [16]. Since in the genomic score the amino acid sequence alignment score is included, it may have become redundant. AGA can also implement a linear gap model by setting $p_{u,\text{na}} = p_{v,\text{na}}$.

From the results (Table 5), it can be seen that AGA can still generate indels that are not aligned with a codon boundary in one of the coding sequences. Provided a sufficiently high penalty for this, this will only happen in regions of overlapping reading frames where the gap can only be codon-aligned with one of the amino acid sequences.

A number of algorithms have been proposed which align coding nucleic acid sequences by back-translation of the corresponding amino acid sequence alignment [1] [18] [19] [18] [2]. Such methods however are limited to coding sequences for a single protein or polyprotein (and cannot deal with overlapping open reading frames), cannot easily deal with frame shift errors that prevent the translation in the first place, and disregard the contribution of nucleotide similarity entirely.

To explicitly consider the possibility of frame shifts, another class of algorithms more similar to AGA have been proposed, which therefore modify nucleic acid sequence alignment algorithms to take into account the translation, but still allow for frame shifts [13], [5], [17]. Like AGA, these algorithms use a Dynamic Programming induction matrix to compute an optimal alignment subject to a scoring function, and different results are caused by different assumptions embedded in their scoring functions.

Codaln [17] implements an algorithm which, like AGA, can read the annotations of coding sequences from a GenBank record. To align an unknown query sequence against an annotated genome, it will first search for open reading frames in the query sequence, which are used in a second step in the scoring function of a Dynamic Programming alignment algorithm. In our results (Table 5), we found that the lower alignment scores, and erroneous frame shifts, of its alignments were caused by errors in these estimated open reading frames.

MACSE [13] penalizes frame shifts but assumes that the entire sequence is a single Coding Sequence which needs to be translated with a minimum of frame shifts and stop codons into a (pseudo-)protein, while optimizing the resulting amino acid alignments. Although we included MACSE in our evaluation, it is thus not well suited to align entire genome with multiple, partially overlapping, open reading frames, some of which may be using the complementary strand.

Future work could be to extend the scoring model to multiple sequence alignment, considering then that one of the included sequences is a reference genome with CDS annotations. This could use the progressive combination of pairwise alignments as originally implemented in CLUSTALW [10], or other heuristic approaches such as implemented in MAFFT [9] or MUSCLE [9].

8 Conclusion

We presented an optimal solution to a fundamental problem in biological sequence analysis, namely how to best align an unknown nucleic acid sequence against a (reference) genome, considering simultaneously similarity at the nucleic acid and amino acid sequence level, and condidering possible frame shifts and gaps causing codon misalignment, but scoring such events with a user-defined penalty.

The proposed method is generally applicable to align any nucleic acid sequence against any genome for which Coding Sequence feature annotations are available. In practice, it is especially well suited to RNA virus sequences, since they are generally rapidly evolving and also typcally have compact viral genomes. By considering amino acid sequence similarity across all Coding Sequences, the method can overcome the large diversity caused by the high rate of evolution, and deals properly with overlapping reading frames common to these viruses.

The method has been implemented in a software package AGA which is available as a command-line package or can be used through a simple web page.

Funding

The VIROGENESIS project receives funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 634650.

References

[1] F. Abascal, R. Zardoya, and M. J. Telford. TranslatorX: multiple alignment of nucleotide sequences guided by amino acid translations. *Nucleic Acids Res.*, 38(Web Server issue):7–13, Jul 2010.

[2] O. R. Bininda-Emonds. transAlign: using amino acids to facilitate the multiple alignment of protein-coding DNA sequences. *BMC Bioinformatics*, 6:156, Jun 2005.

[3] B. Foley, T. Leitner, C. Apetrei, B. Hahn, I. Mizrachi, J. Mullins, A. Rambaut, S. Wolinsky, and B. Korber. HIV Sequence Compendium 2017. *Published by Theoretical Biology and Biophysics Group, Los Alamos National Laboratory, NM, LA-UR 17-25240*, 2017.

[4] O. Gotoh. An improved algorithm for matching biological sequences. *J. Mol. Biol.*, 162(3):705–708, Dec 1982.

[5] J. Hein. An algorithm combining DNA and protein alignment. *J. Theor. Biol.*, 167(2):169–174, Mar 1994.

[6] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U.S.A.*, 89(22):10915–10919, Nov 1992.

[7] C. J. Houldcroft, M. A. Beale, and J. Breuer. Clinical and biological insights from viral genome sequencing. *Nat. Rev. Microbiol.*, 15(3):183–192, Mar 2017.

[8] S. Jammali, E. Kuitche, A. Rachati, F. Belanger, M. Scott, and A. Ouangraoua. Aligning coding sequences with frameshift extension penalties. *Algorithms Mol Biol*, 12:10, 2017.

[9] K. Katoh, K. Misawa, K. Kuma, and T. Miyata. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res.*, 30(14):3059–3066, Jul 2002.

[10] M. A. Larkin, G. Blackshields, N. P. Brown, R. Chenna, P. A. McGettigan, H. McWilliam, F. Valentin, I. M. Wallace, A. Wilm, R. Lopez, J. D. Thompson, T. J. Gibson, and D. G. Higgins. Clustal W and Clustal X version 2.0. *Bioinformatics*, 23(21):2947–2948, Nov 2007.

[11] B. Morgenstern, O. Rinner, S. Abdeddaim, D. Haase, K. F. Mayer, A. W. Dress, and H. W. Mewes. Exon discovery by genomic sequence alignment. *Bioinformatics*, 18(6):777–787, Jun 2002.

[12] E. Pollom, K. K. Dang, E. L. Potter, R. J. Gorelick, C. L. Burch, K. M. Weeks, and R. Swanstrom. Comparison of SIV and HIV-1 genomic RNA structures reveals impact of sequence evolution on conserved and non-conserved structural motifs. *PLoS Pathog.*, 9(4):e1003294, 2013.

[13] V. Ranwez, S. Harispe, F. Delsuc, and E. J. Douzery. MACSE: Multiple Alignment of Coding SEquences accounting for frameshifts and stop codons. *PLoS ONE*, 6(9):e22594, 2011.

[14] P. Rice, I. Longden, and A. Bleasby. EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet.*, 16(6):276–277, Jun 2000.

[15] M. Shi, X. D. Lin, J. H. Tian, L. J. Chen, X. Chen, C. X. Li, X. C. Qin, J. Li, J. P. Cao, J. S. Eden, J. Buchmann, W. Wang, J. Xu, E. C. Holmes, and Y. Z. Zhang. Redefining the invertebrate RNA virosphere. *Nature*, Nov 2016.

[16] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147(1):195–197, Mar 1981.

[17] R. R. Stocsits, I. L. Hofacker, C. Fried, and P. F. Stadler. Multiple sequence alignments of partially coding nucleic acid sequences. *BMC Bioinformatics*, 6:160, Jun 2005.

[18] M. Suyama, D. Torrents, and P. Bork. PAL2NAL: robust conversion of protein sequence alignments into the corresponding codon alignments. *Nucleic Acids Res.*, 34(Web Server issue):W609–612, Jul 2006.

[19] R. Wernersson and A. G. Pedersen. RevTrans: Multiple alignment of coding DNA from aligned amino acid sequences. *Nucleic Acids Res.*, 31(13):3537–3539, Jul 2003.