

Back-translation for discovering distant protein homologies

Marta Gârdea, Laurent Noé, Gregory Kucherov

► **To cite this version:**

Marta Gârdea, Laurent Noé, Gregory Kucherov. Back-translation for discovering distant protein homologies. the 9th International Workshop in Algorithms in Bioinformatics (WABI), Sep 2009, Philadelphia, United States. 5724, pp.108-120, 2009, Lecture Notes in Computer Science. <<http://www.springerlink.com/content/3236004m84465n7j/>>. <10.1007/978-3-642-04241-6_10>. <inria-00448741>

HAL Id: inria-00448741

<https://hal.inria.fr/inria-00448741>

Submitted on 25 Jan 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Back-translation for discovering distant protein homologies

Marta Gîrdea, Laurent Noé, and Gregory Kucherov*

INRIA Lille - Nord Europe, LIFL/CNRS, Université Lille 1, 59655 Villeneuve d'Ascq, France

Abstract. Frameshift mutations in protein-coding DNA sequences produce a drastic change in the resulting protein sequence, which prevents classic protein alignment methods from revealing the proteins' common origin. Moreover, when a large number of substitutions are additionally involved in the divergence, the homology detection becomes difficult even at the DNA level. To cope with this situation, we propose a novel method to infer distant homology relations of two proteins, that accounts for frameshift and point mutations that may have affected the coding sequences. We design a dynamic programming alignment algorithm over memory-efficient graph representations of the complete set of putative DNA sequences of each protein, with the goal of determining the two putative DNA sequences which have the best scoring alignment under a powerful scoring system designed to reflect the most probable evolutionary process. This allows us to uncover evolutionary information that is not captured by traditional alignment methods, which is confirmed by biologically significant examples.

1 Introduction

In protein-coding DNA sequences, frameshift mutations (insertions or deletions of one or more bases) can alter the translation reading frame, affecting all the amino acids encoded from that point forward. Thus, frameshifts produce a drastic change in the resulting protein sequence, preventing any similarity to be visible at the amino acid level.

When the coding DNA sequence is relatively well conserved, the similarity remains detectable at the DNA level, by DNA sequence alignment, as reported in several papers, including [1,2,3,4].

However, the divergence often involves additional base substitutions. It has been shown [5,6,7] that, in coding DNA, there is a base compositional bias among codon positions, that does not apply when the translation reading frame is changed. Hence, after a reading frame change, a coding sequence is likely to undergo base substitutions leading to a composition that complies with this bias. Amongst these substitutions, synonymous mutations (usually occurring on the third position of the codon) are more likely to be accepted by natural selection,

* On leave in J.-V.Poncelet Lab, Moscow, Russia

since they are silent with respect to the gene’s product. If, in a long evolutionary time, a large number of codons in one or both sequences are affected by these changes, the sequence may be altered to such an extent that the common origin becomes difficult to observe by direct DNA comparison.

In this paper, we address the problem of finding distant protein homologies, in particular when the primary cause of the divergence is a frameshift. We achieve this by computing the best alignment of DNA sequences that encode the target proteins. This approach relies on the idea that synonymous mutations cause mismatches in the DNA alignments that can be avoided when all the sequences with the same translation are explored, instead of just the known coding DNA sequences. This allows the algorithm to search for an alignment by dealing only with non-synonymous mutations and gaps.

We designed and implemented an efficient method for aligning putative coding DNA sequences, which builds expressive alignments between hypothetical nucleotide sequences that can provide some information about the common ancestral sequence, if such a sequence exists. We perform the analysis on memory-efficient graph representations of the complete set of putative DNA sequences for each protein, described in Section 3.1. The proposed method, presented in Section 3.2, consists of a dynamic programming alignment algorithm that computes the two putative DNA sequences that have the best scoring alignment under an appropriate scoring system (Section 3.3) designed to reflect the actual evolution process from a codon-oriented perspective.

While the idea of finding protein relations by frameshifted DNA alignments is not entirely new, as we will show in Section 2 in a brief related work overview, Section 4 – presenting tests performed on artificial data – demonstrates the efficiency of our scoring system for distant sequences. Furthermore, we validate our method on several pairs of sequences known to be encoded by overlapping genes, and on some published examples of frameshifts resulting in functional proteins. We briefly present these experiments in Section 5, along with a study of a protein family whose members present high dissimilarity on a certain interval. The paper is concluded in Section 6.

2 Related Work

The idea of using knowledge about coding DNA when aligning amino acid sequences has been explored in several papers.

A *non-statistical approach* for analyzing the homology and the “genetic semi-homology” in protein sequences was presented in [8,9]. Instead of using a statistically computed scoring matrix, amino acid similarities are scored according to the complexity of the substitution process at the DNA level, depending on the number and type (transition/transversion) of nucleotide changes that are necessary for replacing one amino acid by the other. This ensures a differentiated treatment of amino acid substitutions at different positions of the protein sequence, thus avoiding possible rough approximations resulting from scoring

them equally, based on a classic scoring matrix. The main drawback of this approach is that it was not designed to cope with frameshift mutations..

Regarding *frameshift mutation discovery*, many studies [1,2,3,4] preferred the plain BLAST [10,11] alignment approach: BLASTN on DNA and mRNA, or BLASTX on mRNA and proteins, applicable only when the DNA sequences are sufficiently similar. BLASTX programs, although capable of insightful results thanks to the six frame translations, have the limitation of not being able to transparently manage frameshifts that occur inside the sequence, for example by reconstructing an alignment from pieces obtained on different reading frames.

An interesting approach for *handling frameshifts at the protein level* was developed in [12]. Several substitution matrices were designed for aligning amino acids encoded on different reading frames, based on nucleotide pair matches between respective codons. This idea has the advantage of being easy to use with any classic protein alignment tool. However, it lacks flexibility in gap positioning.

On the subject of *aligning coding DNA in presence of frameshift errors*, some related ideas were presented in [13,14]. The author proposed to search for protein homologies by aligning their *sequence graphs* (data structures similar to the ones we describe in Section 3.1). The algorithm tries to align pairs of codons, possibly incomplete since gaps of size 1 or 2 can be inserted at arbitrary positions. The score for aligning two such codons is computed as the maximum substitution score of two amino acids that can be obtained by translating them. This results in a complex, time costly dynamic programming method that basically explores all the possible translations. In Section 3.2, we present an algorithm addressing the same problem, more efficient since it aligns symbols, not codons, and more flexible with respect to scoring functions. Additionally, we propose to use a scoring system relying on codon evolution rather than amino acid translations, since we believe that, in frameshift mutation scenarios, the information provided by DNA sequence dynamics is more relevant than amino acid similarities.

3 Our approach to distant protein relation discovery

The problem of inferring homologies between distantly related proteins, whose divergence is the result of frameshifts and point mutations, is approached in this paper by determining the best pairwise alignment between two DNA sequences that encode the proteins.

Given two proteins P_A and P_B , the objective is to find a pair of DNA sequences, D_A and D_B , such that $translation(D_A) = P_A$ and $translation(D_B) = P_B$, which produce the best pairwise alignment under a given scoring system.

The alignment algorithm (described in Section 3.2) incorporates a gap penalty that limits the number of frameshifts allowed in an alignment, to comply with the observed frequency of frameshifts in a coding sequence's evolution. The scoring system (Section 3.3) is based on possible mutational patterns of the sequences. This leads to reducing the false positive rate and focusing on alignments that are more likely to be biologically significant.

3.1 Data structures

An explicit enumeration and pairwise alignment of all the putative DNA sequences is not an option, since their number increases exponentially with the protein’s length¹. Therefore, we represent the protein’s “back-translation” (set of possible source DNAs) as a directed acyclic graph, whose size depends linearly on the length of the protein, and where a path represents one putative sequence.

As illustrated in Figure 1(a), the graph is organized as a sequence of length $3n$ where n is the length of the protein sequence. At each position i in the graph, there is a group of nodes, each representing a possible nucleotide that can appear at position i in at least one of the putative coding sequences. Two nodes at consecutive positions are linked by arcs if and only if they are either consecutive nucleotides of the same codon, or they are respectively the third and the first base of two consecutive codons. No other arcs exist in the graph.

Note that in the implementation, the number of nodes is reduced by using the IUPAC nucleotide codes. If the amino acids composing a protein sequence are non-ambiguous, only 4 extra nucleotide symbols – R , Y , H and N – are necessary for their back-translation. In this condensed representation, the number of ramifications in the graph is substantially reduced, as illustrated by Figure 1. More precisely, the only amino acids with ramifications in their back-translation are amino acids R , L and S , each encoded by 6 codons with different prefixes.

3.2 Alignment algorithm

We use a dynamic programming method, similar to the Smith-Waterman algorithm, extended to data structures described in Section 3.1 and equipped with gap related restrictions.

Given the input graphs G_A and G_B obtained by back-translating proteins P_A and P_B , the algorithm finds the best scoring local alignment between two DNA sequences comprised in the back-translation graphs (illustrated in Figure 2). The alignment is built by filling each entry $M[i, j, (\alpha_A, \alpha_B)]$ of a dynamic programming matrix M , where i and j are positions of the first and second graph respectively, and (α_A, α_B) is a pair of nodes that can be found in G_A at position i , and in G_B at position j , respectively. An example is given in Figure 3.

The dynamic programming algorithm begins with a classic local alignment initialization (0 at the top and left borders), followed by the recursion step described in equation (1). The partial alignment score from each cell $M[i, j, (\alpha_A, \alpha_B)]$ is computed as the maximum of 6 types of values:

- (a) 0 (similarly to the classic Smith-Waterman algorithm, only non-negative scores are considered for local alignments).
- (b) the substitution score of symbols (α_A, α_B) , denoted $score(\alpha_A, \alpha_B)$, added to the score of the best partial alignment ending in $M[i - 1, j - 1]$, provided that the partially aligned paths contain α_A on position i and α_B on position

¹ With the exception of M and W , which have a single corresponding codon, all amino acids are encoded by 2, 3, 4 or 6 codons.

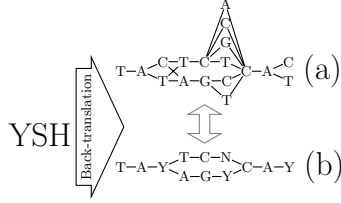


Fig. 1. Example of fully represented (a) and condensed (b) back-translation graph for the amino acid sequence YSH.

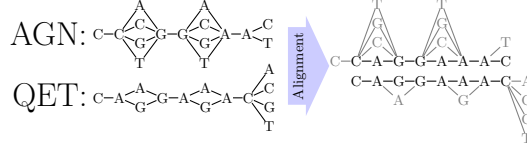


Fig. 2. Alignment example. A path (corresponding to a putative DNA sequence) was chosen from each graph so that the match/mismatch ratio is maximized.

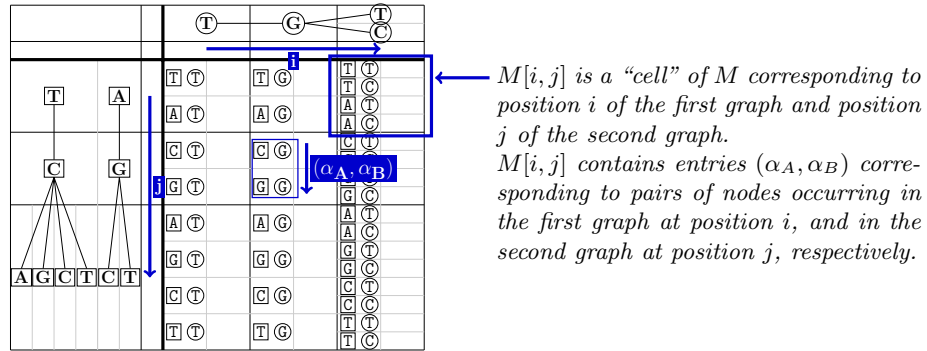


Fig. 3. Example of dynamic programming matrix M .

- j respectively; this condition is ensured by restricting the entries of $M[i - 1, j - 1]$ to those labeled with symbols that precede α_A and α_B in the graphs.
- (c) the cost *singleGapPenalty* of a frameshift (gap of size 1 or extension of a gap of size 1) in the first sequence, added to the score of the best partial alignment that ends in a cell $M[i, j - 1, (\alpha_A, \beta_B)]$, provided that β_B precedes α_B in the second graph; this case is considered only if the number of allowed frameshifts on the current path is not exceeded, or a gap of size 1 is extended.
 - (d) the cost of a frameshift in the second sequence, added to a partial alignment score defined as above.
 - (e) the cost *tripleGapPenalty* of removing an entire codon from the first sequence, added to the score of the best partial alignment ending in a cell $M[i, j - 3, (\alpha_A, \beta_B)]$.
 - (f) the cost of removing an entire codon from the second sequence, added to the score of the best partial alignment ending in a cell $M[i - 3, j, (\beta_A, \alpha_B)]$.

We adopted a non-monotonic gap penalty function, which favors insertions and deletions of full codons, and does not allow a large number of frameshifts – very rare events, usually eliminated by natural selection. As can be seen in equation (1), two particular kinds of gaps are considered: **i) frameshifts** – gaps

of size 1 or 2, with high penalty, whose number in a local alignment can be limited, and **ii) codon skips** – gaps of size 3 which correspond to the insertion or deletion of a whole codon.

$$M[i, j, (\alpha_A, \alpha_B)] = \max \begin{cases} 0 & \text{(a)} \\ M[i-1, j-1, (\beta_A, \beta_B)] + \text{score}(\alpha_A, \alpha_B), \quad \beta_k \in \text{pred}(\alpha_k); & \text{(b)} \\ (M[i, j-1, (\alpha_A, \beta_B)] + \text{singleGapPenalty}), \quad \beta_B \in \text{pred}(\alpha_B); & \text{(c)} \\ (M[i-1, j, (\beta_A, \alpha_B)] + \text{singleGapPenalty}), \quad \beta_A \in \text{pred}(\alpha_A); & \text{(d)} \\ (M[i, j-3, (\alpha_A, \beta_B)] + \text{tripleGapPenalty}), \quad j \geq 3 & \text{(e)} \\ (M[i-3, j, (\beta_A, \alpha_B)] + \text{tripleGapPenalty}), \quad i \geq 3 & \text{(f)} \end{cases} \quad (1)$$

3.3 Translation-dependent scoring function

In this section, we present a new translation-dependent scoring system suitable for our alignment algorithm. The scoring scheme we designed incorporates information about possible mutational patterns for coding sequences, based on a codon substitution model, with the aim of filtering out alignments between sequences that are unlikely to have common origins.

Mutation rates have been shown to vary within genomes, under the influence of several factors, including neighbor bases [15]. Consequently, a model where all base mismatches are equally penalized is oversimplified, and ignores possibly precious information about the context of the substitution.

With the aim of retracing the sequence’s evolution and revealing which base substitutions are more likely to occur within a given codon, our scoring system targets pairs of triplets (α, p, a) , where α is a nucleotide, p is its position in the codon, and a is the amino acid encoded by that codon, thus differentiating various contexts of a substitution. There are 99 valid triplets out of the total of 240 hypothetical combinations.

Pairwise alignment scores are computed for all possible pairs of valid triplets $(t_1, t_2) = ((\alpha_1, p_1, a_1), (\alpha_2, p_2, a_2))$ as a classic log-odds ratio:

$$\text{score}(t_1, t_2) = \lambda \log \frac{f_{t_1 t_2}}{b_{t_1 t_2}} \quad (2)$$

where $f_{t_1 t_2}$ is the frequency of the $t_1 \leftrightarrow t_2$ substitution in related sequences, and $b_{t_1 t_2} = p(t_1)p(t_2)$ is the background probability.

In order to obtain the foreground probabilities $f_{t_i t_j}$, we will consider the following scenario: two proteins are encoded on the same DNA sequence, on different reading frames; at some point, the sequence was duplicated and the two copies diverged independently; we assume that the two coding sequences undergo, in their independent evolution, synonymous and non-synonymous point mutations, or full codon insertions and removals.

The insignificant amount of available real data that fits our hypothesis does not allow classical, statistical computation of the foreground and background probabilities. Therefore, instead of doing statistics on real data directly, we will rely on codon frequency tables and codon substitution models.

We assume that codon substitutions in our scenarios can be modeled by a Markov model presented in [16]² which specifies the relative instantaneous substitution rate from codon i to codon j as:

$$Q_{ij} = \begin{cases} 0 & \text{if } i \text{ or } j \text{ is a stop codon, or} \\ & \text{if } i \rightarrow j \text{ requires more than 1 nucleotide substitution,} \\ \pi_j & \text{if } i \rightarrow j \text{ is a synonymous transversion,} \\ \pi_j \kappa & \text{if } i \rightarrow j \text{ is a synonymous transition,} \\ \pi_j \omega & \text{if } i \rightarrow j \text{ is a nonsynonymous transversion,} \\ \pi_j \kappa \omega & \text{if } i \rightarrow j \text{ is a nonsynonymous transition.} \end{cases} \quad (3)$$

for all $i \neq j$. Here, the parameter ω represents the nonsynonymous-synonymous rate ratio, κ the transition-transversion rate ratio, and π_j the equilibrium frequency of codon j . As in all Markov models of sequence evolution, absolute rates are found by normalizing the relative rates to a mean rate of 1 at equilibrium, that is, by enforcing $\sum_i \sum_{j \neq i} \pi_i Q_{ij} = 1$ and completing the instantaneous rate matrix Q by defining $Q_{ii} = -\sum_{j \neq i} Q_{ij}$ to give a form in which the transition probability matrix is calculated as $P(\theta) = e^{\theta Q}$ [18]. Evolutionary times θ are measured in expected number of nucleotide substitutions per codon.

With this codon substitution model, $f_{t_i t_j}$ can be deduced in several steps. Basically, we first need to identify all pairs of codons with a common subsequence, that have a perfect semi-global alignment (for instance, codons CAT and ATG satisfy this condition, having the common subsequence AT ; this example is further explained below). We then assume that the codons from each pair undergo independent evolution, according to the codon substitution model. For the resulting codons, we compute, based on all possible original codon pairs, $p((\alpha_i, p_i, c_i), (\alpha_j, p_j, c_j))$ – the probability that nucleotide α_i , situated on position p_i of codon c_i , and nucleotide α_j , situated on position p_j of codon c_j have a common origin (equation (5)). From these, we can immediately compute, as shown by equation (6), $p((\alpha_i, p_i, a_i), (\alpha_j, p_j, a_j))$, corresponding in fact to the foreground probabilities $f_{t_i t_j}$, where $t_i = (\alpha_i, p_i, a_i)$ and $t_j = (\alpha_j, p_j, a_j)$.

In the following, $\mathbf{p}(\mathbf{c}_1 \xrightarrow{\theta} \mathbf{c}_2)$ stands for the probability of the event *codon* c_1 mutates into codon c_2 in the evolutionary time θ , and is given by $P_{c_1, c_2}(\theta)$.

$\mathbf{c}_1[\mathbf{interval}_1] \equiv \mathbf{c}_2[\mathbf{interval}_2]$ states that codon c_1 restricted to the positions given by *interval*₁ is a sequence identical to c_2 restricted to *interval*₂. This is equivalent to having a word w obtained by “merging” the two codons. For instance, if $c_1 = CAT$ and $c_2 = ATG$, with their common substring being placed in *interval*₁ = [2..3] and *interval*₂ = [1..2] respectively, w is $CATG$.

Finally, $\mathbf{p}(\mathbf{c}_1[\mathbf{interval}_1] \equiv \mathbf{c}_2[\mathbf{interval}_2])$ is the probability to have c_1 and c_2 , in the relation described above, which we compute as the probability of the word w obtained by “merging” the two codons. This function should be symmetric, it

² Another, more advanced codon substitution model, targeting sequences with overlapping reading frames, is proposed and discussed in [17]. It does not fit our scenario, because it is designed for overlapping reading frames, where a mutation affects both translated sequences, while in our case the sequences become at one point independent and undergo mutations independently.

should depend on the codon distribution, and the probabilities of all the words w of a given length should sum to 1. However, since we consider the case where the same DNA sequence is translated on two different reading frames, one of the two translated sequences would have an atypical composition. Consequently, the probability of a word w is computed as if the sequence had the known codon composition when translated on the reading frame imposed by the first codon, or on the one imposed by the second. This hypothesis can be formalized as:

$$p(w) = p(w \text{ on } rf_1 \text{ OR } w \text{ on } rf_2) = p^{rf_1}(w) + p^{rf_2}(w) - p^{rf_1}(w) \cdot p^{rf_2}(w) \quad (4)$$

where $p^{rf_1}(w)$ and $p^{rf_2}(w)$ are the probabilities of the word w in the reading frame imposed by the position of the first and second codon, respectively. This is computed as the products of the probabilities of the codons and codon pieces that compose the word w in the established reading frame. In the previous example, the probabilities of $w = CATG$ in the first and second reading frame are:

$$\begin{aligned} p^{rf_1}(CATG) &= p(CAT) \cdot p(G**) = p(CAT) \cdot \sum_{c:c \text{ starts with } G} p(c) \\ p^{rf_2}(CATG) &= p(**C) \cdot p(ATG) = \sum_{c:c \text{ ends with } C} p(c) \cdot p(ATG) \end{aligned}$$

The values of $p((\alpha_i, p_i, c_i), (\alpha_j, p_j, c_j))$ are computed as:

$$\sum_{\substack{c'_i, c'_j: c'_i[interval_i] \equiv c'_j[interval_j] \\ p_i \in interval_i, p_j \in interval_j}} p(c'_i[interval_i] \equiv c'_j[interval_j]) \cdot p(c'_i \xrightarrow{\theta} c_i) \cdot p(c'_j \xrightarrow{\theta} c_j) \quad (5)$$

from which obtaining the **foreground probabilities** is straightforward:

$$f_{t_i t_j} = p((\alpha_i, p_i, a_i), (\alpha_j, p_j, a_j)) = \sum_{\substack{c_i \text{ encodes } a_i, \\ c_j \text{ encodes } a_j}} p((\alpha_i, p_i, c_i), (\alpha_j, p_j, c_j)) \quad (6)$$

The **background probabilities** of (t_i, t_j) , $b_{t_i t_j}$, can be simply expressed as the probability of the two symbols appearing independently in the sequences:

$$b_{t_i t_j} = b_{(\alpha_i, p_i, a_i), (\alpha_j, p_j, a_j)} = \sum_{\substack{c_i \text{ encodes } a_i, \\ c_j \text{ encodes } a_j}} \pi_{c_i} \pi_{c_j} \quad (7)$$

Substitution matrix for ambiguous symbols From matrices built as explained above, the versions that use IUPAC ambiguity codes for nucleotides (as proposed in the final paragraph of 3.1) can be computed: the score of pairing two ambiguous symbols is the maximum over all substitution scores for all pairs of nucleotides from the respective sets.

Score evaluation The score significance is estimated according to the Gumbel distribution, where the parameters λ and K are computed with the method described in [19,20]. Since the forward alignment and the reverse complementary alignment are two independent cases with different score distributions, two parameter pairs, λ_{fw}, K_{fw} and λ_{rc}, K_{rc} are computed and used in practice.

4 Validation

To validate the translation-dependent scoring system we designed in the previous section, we tested it on an artificial data set consisting in 96 pairs of protein sequences of average length 300. Each pair was obtained by translating a randomly generated DNA sequence on two different reading frames. Both sequences in each pair were then mutated independently, according to codon mutation probability matrices corresponding to each of the evolutionary times 0.01, 0.1, 0.3, 0.5, 0.7, 1.0, 1.5, 2.00 (measured in average number of mutations per codon).

To this data set we applied four variants of alignment algorithms: i) classic alignment of DNA sequences using classic base substitution scores and affine gap penalties; ii) classic alignment of DNA sequences using a translation-dependent scoring scheme designed in Section 3.3; iii) alignment of back-translation graphs (Section 3.2) using classic base substitution scores and affine gap penalties; iv) alignment of back-translation graphs using a translation-dependent scoring scheme. For the tests involving translation-dependent scores, we used scoring functions corresponding to evolutionary times from 0.30 to 1.00.

Table 1 briefly shows the e-values of the scores obtained with each setup when aligning sequence pairs with various evolutionary distances. While all variants perform well for highly similar sequences, we can clearly deduce the ability of the translation-dependent scores to help the algorithm build significant alignments between sequences that underwent important changes.

		Evolutionary distance between the aligned inputs							
Scores ^(*)	Input type	0.01	0.10	0.30	0.50	0.70	1.00	1.50	2.00
TDS 0.30	graphs	10^{-179}	10^{-171}	10^{-149}	10^{-121}	10^{-109}	10^{-83}	10^{-61}	10^{-37}
	known DNAs	10^{-152}	10^{-136}	10^{-110}	10^{-76}	10^{-54}	10^{-21}	10^{-6}	1.00
TDS 0.50	graphs	10^{-166}	10^{-156}	10^{-140}	10^{-118}	10^{-107}	10^{-85}	10^{-55}	10^{-34}
	known DNAs	10^{-140}	10^{-128}	10^{-105}	10^{-75}	10^{-61}	10^{-34}	10^{-6}	10^{-1}
TDS 0.70	graphs	10^{-153}	10^{-145}	10^{-130}	10^{-113}	10^{-102}	10^{-83}	10^{-57}	10^{-51}
	known DNAs	10^{-130}	10^{-120}	10^{-101}	10^{-76}	10^{-64}	10^{-42}	10^{-13}	10^{-7}
TDS 1.00	graphs	10^{-137}	10^{-131}	10^{-118}	10^{-104}	10^{-97}	10^{-80}	10^{-59}	10^{-54}
	known DNAs	10^{-117}	10^{-110}	10^{-93}	10^{-70}	10^{-65}	10^{-46}	10^{-21}	10^{-8}
classic scores	graphs	10^{-127}	10^{-24}	10^{-12}	10^{-11}	10^{-7}	10^{-5}	10^{-3}	10^{-2}
	known DNAs	10^{-86}	10^{-20}	10^{-9}	10^{-7}	10^{-4}	10^{-1}	1.00	1.00

Table 1. Order of the e-values of the scores obtained by aligning artificially diverged pairs of proteins resulted from the translation of the same ancestral sequence on two reading frames. ^(*) $TDS < evolutionary\ distance > =$ translation-dependent scores; classic substitution scores: match = 3, transversion = -4, transition = -2.

The resulting alignments reveal that, even after many mutations, the translation-dependent scores manage to recover large parts of the original shared sequence, by correctly aligning most positions. On the other hand, with classic match/mismatch scores, the algorithm usually fails to find these common zones.

Moreover, due to the large number of mismatches, the alignment has a low score, comparable to scores that can be obtained for randomly chosen sequences. This makes it difficult to establish whether the alignment is biologically meaningful or it was obtained by chance. This issue is solved by the translation-dependent scores by uneven substitution penalties, according to the codon mutation models.

We conclude that the usage of translation-dependent scores makes the algorithm more robust, able to detect the common origins even after the sequences underwent many modifications, and also able to filter out alignments where the nucleotide pairs match by pure chance and not due to evolutionary relations.

5 Experimental results

5.1 Tests on known overlapping and frameshifted genes

We tested the method on pairs of proteins known to be encoded by overlapping genes in viral genomes (phage X174 and Influenza A) and in *E.coli* plasmids, as well as on the newly identified overlapping genes *yaaW* and *htgA* from *E.coli* K12 [21]. In all cases, we obtained perfect identification of gene overlaps with simple substitution scores and with translation-dependent scoring matrices corresponding to low evolutionary distances (at most 1 mutation per codon). Translation-dependent scoring matrices of higher evolutionary distances favor, in some (rare) cases, substitutions instead of matches within the alignment. This is a natural consequence of increasing the codon's chance to mutate, and it illustrates the importance of choosing a score matrix corresponding to the real evolutionary distance. Our method was also able to detect, directly on the protein sequences, the frameshifts resulting in functional proteins reported in [1,2,3,4].

5.2 New divergence scenarios for orthologous proteins

In this section we discuss the application of our method to FMR1NB (Fragile X mental retardation 1 neighbor protein) family. The Ensembl database [22] provides 23 members of this family, from mammalian species, including human, mouse, dog and cow. Their multiple alignment, provided by Ensembl, shows high dissimilarity on the first part (100 amino acids approximately), and good conservation on the rest of the sequence. We apply our alignment algorithm on proteins from several organisms, where the complete sequence is available.

We performed our experiments with translation-dependent scoring matrices corresponding to 0.3, 0.5 and 0.7 mutations per codon. Given that, in our scenario (presented in section 3.3), the divergence is applied on two reading frames, this implies an overall mutation rate of 0.6, 1.0 and 1.4 mutations per codon respectively. Thus, the mutation rate per base reflected by our scores is less than 0.5, which is approximately the nucleotide substitution rate for mouse relative to human [23]. The number of allowed frameshifts was limited to 3. The gap penalties were set in all cases to -20 for codon indels, -20 for size 1 gaps and -5 for the extension of size 1 gaps (size 1 and size 2 gaps correspond to frameshifts). These choices were made so that the penalty for codon indels is higher than the average penalty for 3 substitutions.

An implementation of our method is available at <http://bioinfo.lifl.fr/path/>.

References

1. Raes, J., Van de Peer, Y.: Functional divergence of proteins through frameshift mutations. *Trends in Genetics* **21**(8) (2005) 428–431
2. Okamura, K. *et al.*: Frequent appearance of novel protein-coding sequences by frameshift translation. *Genomics* **88**(6) (2006) 690–697
3. Harrison, P., Yu, Z.: Frame disruptions in human mRNA transcripts, and their relationship with splicing and protein structures. *BMC Genomics* **8** (2007) 371
4. Hahn, Y., Lee, B.: Identification of nine human-specific frameshift mutations by comparative analysis of the human and the chimpanzee genome sequences. *Bioinformatics* **21**(Suppl 1) (2005) i186–i194
5. Grantham, R., Gautier, C., Gouy, M., Mercier, R., Pave, A.: Codon catalog usage and the genome hypothesis. *Nucleic Acids Research* (8) (1980) 49–62
6. Shepherd, J.C.: Method to determine the reading frame of a protein from the purine/pyrimidine genome sequence and its possible evolutionary justification. *Proceedings National Academy Sciences USA* (78) (1981) 1596–1600
7. Guigo, R.: DNA composition, codon usage and exon prediction. *Nucleic protein databases* (1999) 53–80
8. Leluk, J.: A new algorithm for analysis of the homology in protein primary structure. *Computers and Chemistry* **22**(1) (1998) 123–131
9. Leluk, J.: A non-statistical approach to protein mutational variability. *BioSystems* **56**(2-3) (2000) 83–93
10. Altschul, S. *et al.*: Basic local alignment search tool. *JMB* **215**(3) (1990) 403–410
11. Altschul, S. *et al.*: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* **25**(17) (1997) 3389–3402
12. Pellegrini, M., Yeates, T.: Searching for Frameshift Evolutionary Relationships Between Protein Sequence Families. *Proteins* **37** (1999) 278–283
13. Arvestad, L.: Aligning coding DNA in the presence of frame-shift errors. *Proceedings of the 8th Annual CPM Symposium* **1264** (1997) 180–190
14. Arvestad, L.: Algorithms for biological sequence alignment. PhD thesis, Royal Institute of Technology, Stockholm, Numerical Analysis and Computer Science (2000)
15. Blake, R., Hess, S., Nicholson-Tuell, J.: The influence of nearest neighbors on the rate and pattern of spontaneous point mutations. *JME* **34**(3) (1992) 189–200
16. Kosiol, C., Holmes, I., Goldman, N.: An Empirical Codon Model for Protein Sequence Evolution. *Molecular Biology and Evolution* **24**(7) (2007) 1464
17. Pedersen, A., Jensen, J.: A dependent-rates model and an MCMC-based methodology for the maximum-likelihood analysis of sequences with overlapping reading frames. *Molecular Biology and Evolution* **18** (2001) 763–776
18. Lio, P., Goldman, N.: Models of Molecular Evolution and Phylogeny. *Genome Research* **8**(12) (1998) 1233–1244
19. Altschul, S. *et al.*: The estimation of statistical parameters for local alignment score distributions. *Nucleic Acids Research* **29**(2) (2001) 351–361
20. Olsen, R., Bundschuh, R., Hwa, T.: Rapid assessment of extremal statistics for gapped local alignment. *ISMB* (1999) 211–222
21. Delaye, L., DeLuna, A., Lazcano, A., Becerra, A.: The origin of a novel gene through overprinting in *Escherichia coli*. *BMC Evolutionary Biology* **8** (2008) 31

22. Hubbard, T. *et al.*: Ensembl 2007. *Nucleic Acids Res.* **35** (2007)
23. Clamp, M. *et al.*: Distinguishing protein-coding and noncoding genes in the human genome. *Proc Natl Acad Sci* **104**(49) (2007) 19428–19433
24. Oostra, B., Chiurazzi, P.: The fragile X gene and its function. *Clinical genetics* **60**(6) (2001) 399