A simple theory for finding related sequences by adding probabilities of alternative alignments

Martin C. Frith[1,2,3]

[1]Artificial Intelligence Research Center, AIST, Tokyo, Japan
[2]Department of Computational Biology and Medical Sciences, University of Tokyo, Chiba, Japan
[3]Computational Bio Big Data Open Innovation Laboratory, AIST, Tokyo, Japan

Summary

The main way of analyzing genetic sequences is by finding sequence regions that are related to each other. There are many methods to do that, usually based on this idea: find an alignment of two sequence regions, which would be unlikely to exist between unrelated sequences. Unfortunately, it is hard to tell if an alignment is likely to exist by chance. Also, the precise alignment of related regions is uncertain. One alignment does not hold all evidence that they are related. We should consider alternative alignments too. This is rarely done, because we lack a simple and fast method that fits easily into practical sequence-search software.

Here is described a simplest-possible change to standard sequence alignment, which sums probabilities of alternative alignments. Remarkably, this makes it easier to tell if a similarity is likely to occur by chance. This approach is better than standard alignment at finding distant relationships, at least in a few tests. It can be used in practical sequence-search software, with minimal increase in implementation difficulty or run time. It generalizes to different kinds of alignment, e.g. DNA-versus-protein with frameshifts. Thus, it can widely contribute to finding subtle relationships between sequences.

Introduction

Many methods have been developed for finding related parts of nucleotide or protein sequences. They usually follow a standard approach: define positive and negative scores for monomer matches, mismatches, and gaps, then seek high-scoring alignments. Why is this bizarre, intricate approach a good way to find related sequences? Its result depends utterly on the score parameters, so how should we choose them?

The answer to both questions is that the approach is equivalent to probability-based alignment [1, 2]. This uses probabilities of matches, mismatches, and gaps, e.g. their average rates in the kinds of related sequences we wish to find, and seeks high-probability alignments. An alignment's score is a log probability ratio [3]:

$$\text{score} = \log \frac{\text{prob(alignment)}}{\text{prob(null alignment)}} \,. \tag{1}$$

A "null alignment" is a length-zero alignment between the two sequences. A precise definition of prob(null alignment) was given previously [3], and is repeated in this paper's appendix. A high alignment score indicates high probability with these match/mismatch/gap rates. The base of the log (e.g. $\log_2$, $\log_{10}$) is unspecified, because any base is equally correct. A change of base merely rescales the scores. (In the precursor to this paper [3], $\log x$ was equivalently written as $t \ln x$, where t is an arbitrary positive constant.) This approach is termed "pairwise local alignment", where "local" means that the alignment is not necessarily of the whole sequences.

A more direct way to judge whether two sequence regions are related is to calculate their probability

without fixing an alignment. In other words, use a score like this:

$$\begin{aligned}\text{score} &= \log \frac{\sum_{\text{alignments}} \text{prob(alignment)}}{\text{prob(null alignment)}} \\ &= \log \sum_{\text{alignments}} \frac{\text{prob(alignment)}}{\text{prob(null alignment)}} \,.\end{aligned} \tag{2}$$

If the sum includes all possible ways of aligning the sequence regions, it gives their total probability with the assumed rates of matches, mismatches, and gaps. Thus, it includes all evidence that the sequence regions are related, while one alignment includes just some evidence.

A similar sum-over-alignments method has been used with great success in the HMMER software [4, 5]. So it is curious that many sequence-search tools developed since HMMER do not use this approach. No-one has described a sum-over-alignments method that fits easily into typical sequence-search software, with minimal increase in implementation difficulty or computational cost. HMMER is too complicated and specialized, e.g. it has an enormous "implicit probabilistic model", and sequence-length–dependent parameters [4]. HMMER is designed to search a smallish "query" sequence against a large "database", so it can't find related parts of two large sequences, e.g. two genomes.

It is important to know whether a similarity score (eqs. 1, 2) is likely to occur by chance between unrelated sequences. The simplest definition is: the probability of an equal or higher score between two sequences of random i.i.d. (independent and identically distributed) monomers. Even this simplest definition, however, is hard to calculate. There is a solution for gapless alignment, in the limit where the sequences are long [6]. Gapped alignment relies on conjecture and simulation. BLAST can calculate it only for a few sets of score parameters, not including realistic DNA scores that distinguish transition (`a:g`, `c:t`) from transversion mismatches [7]. Some widely-used aligners do not calculate it at all [8]. The ALP library can calculate it for arbitrary parameters, by complex simulations [9]. Remarkably, HMMER calculates it in an easier way, based on other conjectures. The scope of those conjectures is unclear, e.g. they seem to hold only after tweaking HMMER to use a "uniform entry/exit distribution" [4].

Here is described a minimal modification of standard alignment, which sums probabilities of alternative alignments (eq. 2). It can replace the alignment component of practical sequence-search software, with minimal increase in implementation difficulty or run time. This also makes it easier to calculate probabilities of similarity scores between random sequences, based on a clear conjecture.

Limitations of previous methods

Many previous studies have summed probabilities of alternative alignments. This section surveys some representatives, to understand why this approach is rarely used to find related parts of sequences.

One line of research considered substitutions, insertions, and deletions occuring over time, so that match/mismatch/gap probabilities depend on the length of time [10–14]. These methods achieve several aims. Given two related sequences, they can: infer substitution/insertion/deletion rates per unit time, find the most likely alignment, and report the probability that any pair of monomers descend from a common ancestor. Furthermore, Hein et al. used a log likelihood ratio to compare two hypotheses, that two whole sequences are related or not:

$$\log \frac{\text{prob(sequences} \mid \text{related)}}{\text{prob(sequences} \mid \text{unrelated)}} \,. \tag{3}$$

The numerator is a sum of probabilities of all possible alignments between the whole sequences. These studies achieved exact mathematical solutions only for simple models of sequence evolution, which do not fit real sequences well. In particular, it seems hard to allow insertion and deletion events of length > 1. To my knowledge, this approach has never been used to find related parts of sequences.

Another line of research (including the present study) simplifies things by ignoring time-dependence, and directly using probabilities of matches, mismatches, and gaps. Allison et al. [15] estimated such probabilities from a pair of related sequences, allowing complex gap-length distributions. They also used a

log likelihood ratio to compare hypotheses that two whole sequences are related or not.

In a pioneering study, Bucher & Hofmann used a log likelihood ratio to compare two hypotheses, that two sequences have a related segment or not [16]:

$$\text{score} = \log \frac{\text{prob(sequences} \mid \text{have related part)}}{\text{prob(sequences} \mid \text{lack related part)}} \quad (4)$$

Each hypothesis has probabilities for the lengths of the two sequences. Their two hypotheses are identical in this regard. Thus, the sequence lengths never favor either hypothesis. This is arguably unnatural, biologically and mathematically. Biologically, if related regions are equally likely to occur per unit sequence length, longer sequences are more likely to have related regions. For example, a longer chromosome is more likely to contain a mobile DNA element. Mathematically, they need two summation algorithms. The likelihood ratio becomes a fraction where the numerator is a sum of exponentiated alignment scores, and the denominator is another sum. This denominator increases with increasing sequence lengths [17, fig. 2]. So scores of related regions decrease with increasing lengths of sequences containing them.

Lunter et al. compared DNA sequences accurately, using probability methods similar to the present study [18]. In their scenario, they are given a pair of sequences that are assumed to have one related region. So they didn't use any score to find related regions.

HMMER uses a log likelihood ratio similar to eq. 4 [4]. HMMER is more general, however. It finds related parts between a sequence and a sequence family: the family has position-specific monomer and gap probabilities. The null hypothesis (no related parts) is compared to either a "unihit" hypothesis (one related part), or a "multihit" hypothesis (one or more related parts). Each hypothesis has probabilities for the length of the sequence. The hypotheses are roughly equal in this regard. HMMER achieves that by adjusting each hypothesis to fit the length of each sequence that is provided for analysis [4].

FEAST is practical heuristic software for finding related sequence parts, and it sums probabilities of alternative alignments [19]. It doesn't determine whether similarities would be unlikely to exist between random sequences. Compared to standard alignment, it is slower, and implementation difficulty was not made clear. It seems to be an implementation, rather than a simple method that can be widely implemented.

Several of these previous methods use a likelihood ratio, which is the optimal way to judge between two hypotheses with those likelihoods [20]. It is however not clear that comparing two hypotheses is the best way to find related sequence parts. For example, if we compare the human and mouse X chromosomes, it is not very useful just to know whether or not they have 0 related parts. There is a tradeoff between judging whether sequences are related, and how they are related. A single alignment specifies a relationship in detail, but has the lowest power to detect subtle relationships. Summing over all alignments has the highest power, but least detail. We seek a compromise between these extremes.

The final previous idea that we shall examine is "hybrid alignment" [21]. When hybrid alignment compares a sequence of length m to a sequence of length n, it considers $m \times n$ hypotheses for how they are related. Each hypothesis concerns the the length-i prefix of one sequence and length-j prefix of the other. The hypothesis is that the sequences have a related region that ends exactly at the ends of these prefixes. The probability associated with this hypothesis is the sum of probabilities of all alignments that start anywhere and end at the ends of the prefixes. This sum of alignment probabilities is converted to a score (eq. 2). Remarkably, it is easy to calculate the probability of an equal or higher score between random i.i.d. sequences, provided that the alignment parameters satisfy a "conservation condition". This calculation was not proven correct, but was supported by theoretical arguments and tests on random sequences.

The present study builds on hybrid alignment. Hybrid alignment has been neglected, probably because its description was mathematically fearsome, and it was presented as a strange "hybrid" method. Here, instead of alignments ending at (i, j), we shall sum over alignments that pass through (i, j). This treats uncertainty about starts and ends of related regions symmetrically. It is a good fit for practical heuristic aligners, which extend alignment forwards and backwards from a candidate starting point.

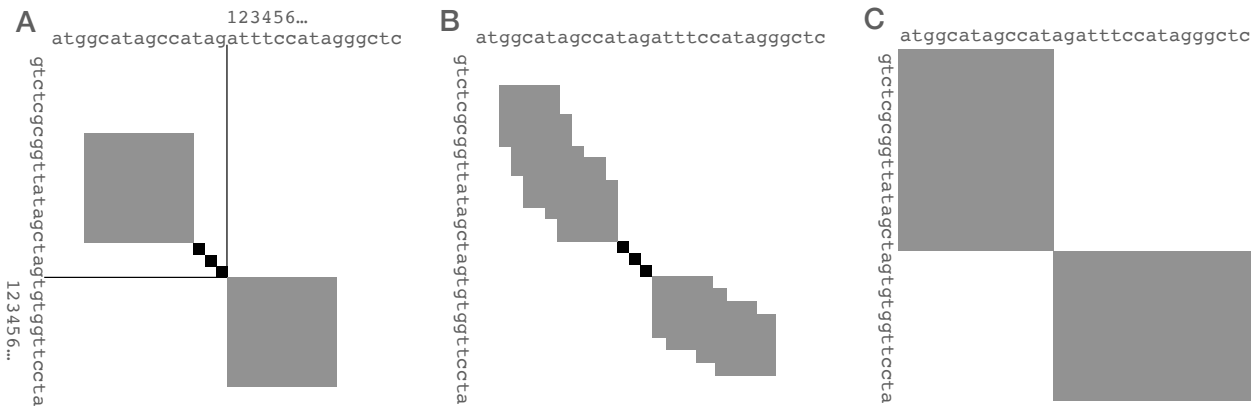

Figure 1: Some ways of extending an alignment in both directions from a small "core" alignment (black). In each direction, an optimum alignment extension is sought from among all possible ones within the gray area. **A** Simple toy method: the gray areas are blocks of predefined size $n \times n$. **B** Realistic example: the gray areas are adjusted as the calculation proceeds, attempting to follow good alignments. **C** Unlimited extension from zero-size core.

Methods

Review of standard alignment

Practical methods for finding related parts of sequences often have two steps. To cope with big data, the first step uses fast heuristics to find potentially related regions, then the second step defines the regions precisely by alignment. The heuristics are various: short exact matches, spaced matches, reduced-alphabet matches, groups of nearby matches, etc. [7, 22–24]. The second step is often done by extending alignments to the left and right of a small "core" alignment (fig. 1B). This finds the highest-scoring alignment extension, within a limited range shown in gray in fig. 1B. The range is adjusted as the algorithm proceeds, attempting to follow the highest score. The best way to do this is unclear: several ways have been suggested [7, 25–28].

To avoid irrelevant complexity, here is the extension algorithm in a block of predefined size $n \times n$ (fig. 1A). This algorithm extends forwards (towards the lower-right in fig. 1A): a similar algorithm extends backwards. The two sequences are R (e.g. "reference") and Q (e.g. "query"), with positions numbered as in fig. 1A. The score for aligning monomer x (`a`, `c`, `g`, or `t` for DNA) in R to monomer y in Q is S_{xy}. Let us use different scores for starting versus extending gaps, because gaps are somewhat rare (low start probability) but often long (high extension probability). So the score for a length-k deletion is $a_D + b_D \times (k-1)$, for a length-k insertion $a_I + b_I \times (k-1)$.

Algorithm 1 Optimum alignment extension in an $n \times n$ block (lower-right gray block in fig. 1A)

$$X_{00} \leftarrow 0 \qquad X_{i0} \leftarrow -\infty \qquad X_{0i} \leftarrow -\infty \qquad (0 < i \le n)$$
$$v \leftarrow -\infty \qquad Z_{i0} \leftarrow -\infty \qquad Y_{0i} \leftarrow -\infty \qquad (0 \le i \le n)$$

$$\left.\begin{aligned} w &\leftarrow \max(X_{ij}, Y_{ij}, Z_{ij}) \\ v &\leftarrow \max(v, w) \\ X_{i+1\ j+1} &\leftarrow w + S_{R_{i+1} Q_{j+1}} \\ Y_{i+1\ j} &\leftarrow \max(w + a_D,\ Y_{ij} + b_D) \\ Z_{i\ j+1} &\leftarrow \max(w + a_I,\ Z_{ij} + b_I) \end{aligned}\right\} \begin{aligned} 0 \le i \le n \\ 0 \le j \le n \end{aligned}$$

Alg. 1 finds the highest possible score for an alignment ending at R_i and Q_j, with R_i aligned to Q_j (X_{ij}), or R_i aligned to a gap (Y_{ij}), or Q_j aligned to a gap (Z_{ij}). v tracks the best alignment score seen so far, and is the final output. Several variants of this algorithm are possible. This variant has simple

boundary conditions, while minimizing add, max, and array access operations [29]. The alignment score after left and right extension is:

$$\text{score} = v_{\text{left}} + (\text{core score}) + v_{\text{right}}\,. \tag{5}$$

New extension algorithm

This section shows the new algorithm, and the next section explains how it sums probabilities. There are two changes from alg. 1. The first, which makes no difference to the result, is to work with exponentiated values: probability ratios instead of log probability ratios. This is so we can easily add them. This change means that $-\infty \Rightarrow 0$, $0 \Rightarrow 1$, and addition $\Rightarrow$ multiplication. The other change is: maximization $\Rightarrow$ addition. This calculates the sum of probability ratios

Algorithm 2 Sum over alignment extensions in an $n \times n$ block (lower-right gray block in fig. 1A)

$$\begin{array}{llll} X'_{00} \leftarrow 1 & X'_{i0} \leftarrow 0 & X'_{0i} \leftarrow 0 & (0 < i \leq n) \\ v' \leftarrow 0 & Z'_{i0} \leftarrow 0 & Y'_{0i} \leftarrow 0 & (0 \leq i \leq n) \end{array}$$

$$\left.\begin{aligned} w' &\leftarrow X'_{ij} + Y'_{ij} + Z'_{ij} \\ v' &\leftarrow v' + w' \\ X'_{i+1\;j+1} &\leftarrow w' \cdot S'_{R_{i+1}Q_{j+1}} \\ Y'_{i+1\;j} &\leftarrow w' \cdot a'_D + Y'_{ij} \cdot b'_D \\ Z'_{i\;j+1} &\leftarrow w' \cdot a'_I + Z'_{ij} \cdot b'_I \end{aligned}\right\} \begin{array}{l} 0 \leq i \leq n \\ 0 \leq j \leq n \end{array}$$

for all alignment extensions in the gray area (eq. 2), instead of the maximum probability ratio. The score with left and right extension is:

$$\text{score} = \log[v'_{\text{left}}] + (\text{core score}) + \log[v'_{\text{right}}]\,. \tag{6}$$

Note that the number of computational operations is unchanged: the only changes are replacing addition with multiplication, and maximization with addition. So the computational complexity is unchanged. The practical run time depends on implementer expertise (e.g. using SIMD), but plausibly it need not increase too much.

Unfortunately, the values in alg. 2 may become too large for the computer to handle. This can be fixed by occasionally rescaling them (see the Supplement). Because the rescaling is only done occasionally, it does not increase the run time noticeably.

Parameters from probabilities

To use alg. 1 or 2, we must choose values for the parameters (e.g. a'_D, b'_I). They are related to probabilities of matches, mismatches, and gaps. We need a precise definition of these probabilities. We shall use the definition in fig. 2. The main alignment probabilities are shown in the middle: α_D is the deletion start probability, β_D the deletion extension probability, α_I the insertion start probability, and β_I the insertion extension probability. Apart from starting an insertion or deletion, the other possibilities are to have a pair of aligned monomers (probability γ), or end the alignment (probability $1 - \gamma - \alpha_D - \alpha_I$). A pair of aligned monomers has probability π_{xy} of being monomer type x in sequence R and y in sequence Q. There are also probabilities for monomers left and right of the alignment: ω_D and ω_I (for sequence R and Q respectively). An unaligned monomer is of type z with probability ϕ_z in R, and ψ_z in Q.

A "null alignment" is produced by a path that never traverses the α_D, α_I, or γ arrows. This corresponds to a length-0 alignment between the sequences. The score of a null alignment is traditionally zero, which is achieved by eq. 1. This means that an alignment score is determined solely by the aligned regions, and not by flanking sequences or their lengths.

Having defined these probabilities, we can state the parameters for alg. 2, as shown previously [3]:

$$S'_{xy} = \frac{\gamma}{\omega_D \omega_I} \cdot \frac{\pi_{xy}}{\phi_x \psi_y} \tag{7}$$

$$a'_D = \alpha_D(1 - \beta_D)/\omega_D \tag{8}$$

$$b'_D = \beta_D/\omega_D \tag{9}$$

$$a'_I = \alpha_I(1 - \beta_I)/\omega_I \tag{10}$$

$$b'_I = \beta_I/\omega_I \tag{11}$$

With these parameters, alg. 2 calculates the sum of probability ratios in eq. 2. The parameters for alg. 1 are defined similarly (see the Supplement).

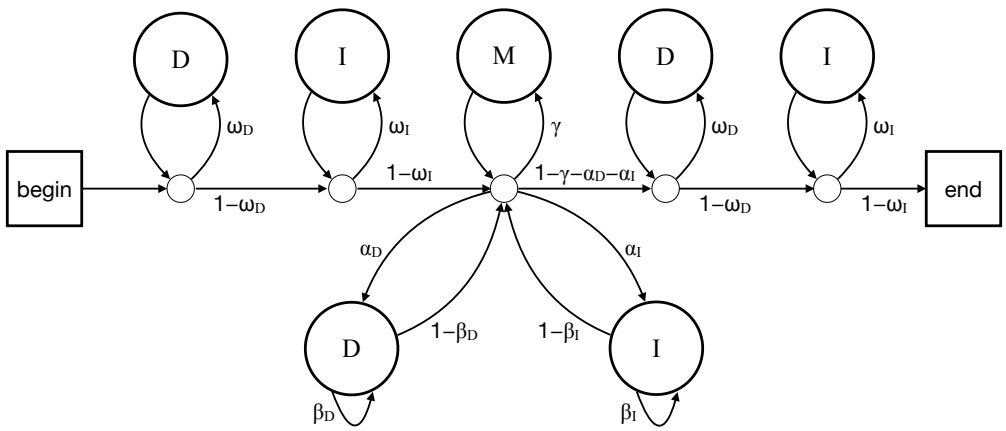

Figure 2: A probability model for two sequences with a related region. The model produces different alignments with different probabilities. Starting at "begin", the arrows are followed according to their probabilities (e.g. ω_D versus $1-\omega_D$). Each pass through a **D** state produces an unaligned letter x in one sequence (R), with probabilities ϕ_x. Each pass through an **I** state produces an unaligned letter y in the other sequence (Q), with probabilities ψ_y. Each pass through **M** generates two aligned letters, x in R and y in Q, with probabilities π_{xy}.

Fig. 2 is not the only way to define the probabilities [3]. There are other ways that cause no change to alg. 1, but cause slight changes to alg. 2. The definition in fig. 2 seems to produce the simplest equations, and minimizes the number of computational operations in alg. 2. This is a minor improvement over hybrid alignment, which defined probabilities with an unnatural asymmetry between insertions and deletions [3, fig. S2].

A non-heuristic similarity score

What would we do if we did not need fast heuristics? As mentioned in the Introduction, we seek a compromise between using just one alignment, and summing over all alignments. We shall use a compromise with two useful features: it is approximated by the heuristic approach (fig. 1B), and we can tell if a similarity is unlikely to exist between random sequences.

When we compare a length-m sequence (R_1, R_2, …, R_m) to a length-n sequence (Q_1, Q_2, …, Q_n), we shall consider $(m+1)\times(n+1)$ hypotheses for how they are related. Imagine cutting a length-i prefix of R and a length-j prefix of Q. These cuts define a point, illustrated in fig. 1C by the touching corners of the gray rectangles. The hypothesis is that the sequences are related by any alignment that passes through this point (including alignments that start or end there). The score of this hypothesis is given by eq. 2, with the sum taken over these alignments.

We can calculate this score for each (i,j). We first calculate W^F_{ij}, the sum of probability ratios of all alignments that end immediately after i letters of R and j letters of Q:

$$0 \le i \le m,\ \ 0 \le j \le n$$

$$\begin{aligned} W^F_{ij} &= S'_{R_i Q_j} W^F_{i-1\ j-1} + Y^F_{i-1\ j} + Z^F_{i\ j-1} + 1 \\ Y^F_{ij} &= a'_D W^F_{ij} + b'_D Y^F_{i-1\ j} \\ Z^F_{ij} &= a'_I W^F_{ij} + b'_I Z^F_{i\ j-1} \end{aligned}$$

The boundary condition is: when $i<0$ or $j<0$, $W^F_{ij} = Y^F_{ij} = Z^F_{ij} = 0$.

We then calculate W^B_{ij}, the sum of probability ratios of all alignments that start immediately after i letters

of R and j letters of Q:

$$m \geq i \geq 0, \;\; n \geq j \geq 0$$
$$W^B_{ij} = S_{R_{i+1}Q_{j+1}} W^B_{i+1\;j+1} + Y^B_{i+1\;j} + Z^B_{i\;j+1} + 1$$
$$Y^B_{ij} = a'_D W^B_{ij} + b'_D Y^B_{i+1\;j}$$
$$Z^B_{ij} = a'_I W^B_{ij} + b'_I Z^B_{i\;j+1}$$

The boundary condition is: when $i > m$ or $j > n$, $W^B_{ij} = Y^B_{ij} = Z^B_{ij} = 0$. Finally, the sum of probability ratios of all alignments passing through (i, j) is $W^F_{ij} W^B_{ij}$. So the score is $\log[W^F_{ij} W^B_{ij}]$.

A minor point is that this calculation excludes alignments that pass through (i, j) while traversing the β_D or β_I arrow (fig. 2). It's not clear whether it's better to include them, but it seems unlikely to make much difference.

Similarity scores occurring by chance

We wish to know the probability of getting a similarity score, $s_{\max} = \log[\max_{ij}(W^F_{ij} W^B_{ij})]$, between random sequences. Because our score is similar to that of hybrid alignment [21], we shall assume (and test) that we can use the same approach. This approach requires that the alignment parameters satisfy a "conservation condition".

The conservation condition is related to alignment length bias. The probabilities in fig. 2 may be biased towards short or long alignments between two given sequences. For example, if α_D, α_I, and γ are all low (almost 0), but ω_D and ω_I are high (almost 1), shorter alignments will have higher probability. In the converse situation, longer alignments are favored. It was shown previously [3] that the probabilities are unbiased when:

$$\frac{\gamma}{\omega_D \omega_I} + \frac{\alpha_D(1-\beta_D)}{\omega_D - \beta_D} + \frac{\alpha_I(1-\beta_I)}{\omega_I - \beta_I} = 1\,. \qquad (12)$$

Eq. 12 is a special case of the conservation condition. The general case is shown in the Appendix, but we shall only use the special case.

We can calculate the probability of a similarity score between two random i.i.d. sequences: one with length m and monomer probabilities ϕ_x, the other with length n and probabilities ψ_y. These monomer probabilities must match those in eq. 7. The prediction is that

$$\text{prob}(s_{\max} \geq s) = 1 - e^{-Kmn/\exp(s)}\,, \qquad (13)$$

in the limit where m and n are large. exp is defined to mean inverse of log.

Eq. 13 has one unknown parameter, K. We can estimate K by generating some pairs of random i.i.d. sequences, calculating $s_{\max}$ for each pair, and fitting K. The maximum likelihood fit is:

$$K = \text{harmonic mean}\,[\exp(s_{\max})] \,/\, mn\,. \qquad (14)$$

The random chance of a similarity score s is often reported as an E-value. The E-value is the expected number of similarities with score $\geq s$ between two random sequences of length m and n. Chance similarities occur at a constant rate between any part of one sequence and any part of the other. So they follow a Poisson distribution. This means the E-value is:

$$E = Kmn/\exp(s)\,. \qquad (15)$$

Implementation and availability

This approach to finding related sequence parts was added to LAST (`https://gitlab.com/mcfrith/last`). This implementation is just a proof of principle: the heuristics are not necessarily the best. LAST can use either alg. 1 with E-values from the ALP library, or alg. 2. In both cases, it gets a representative alignment by running alg. 1 and tracing back a way to get the maximum v. LAST runs these algorithms not in an $n \times n$ block (fig. 1A), but in a range that is adjusted as the algorithm proceeds (fig. 1B, see the Supplement).

LAST estimates K from random sequences. It is not clear how long these sequences need to be. Based on some tests (see the Supplement), LAST uses 50 pairs of length-500 sequences by default.

The match/mismatch/gap probabilities (π_{xy}, α_D, etc.) were determined using `last-train`, which finds related regions of given sequences and infers these probabilities [30]. `last-train` was modified to make the probabilities satisfy eq. 12. It sets ω_D and ω_I by assuming that they are equal, and finding

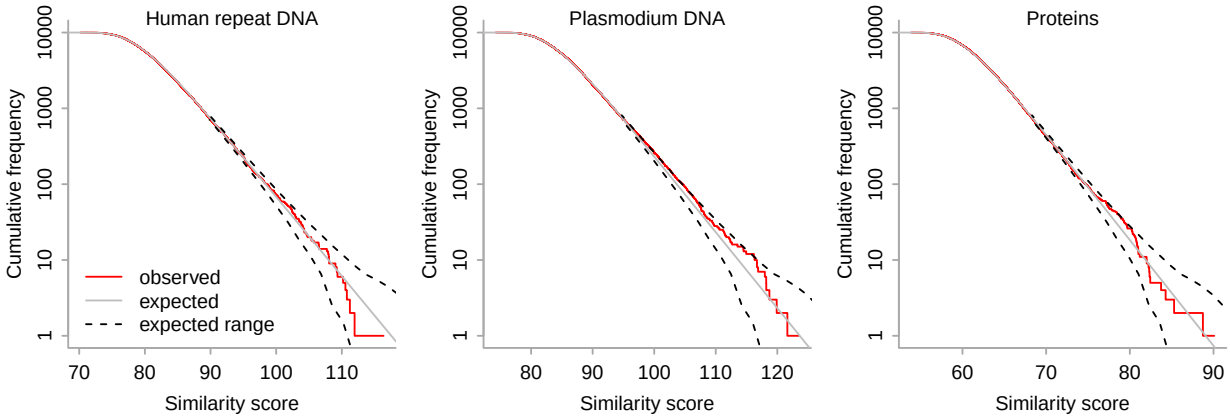

Figure 3: Non-heuristic similarity score ($s_{\max}$) between 10000 pairs of random i.i.d. sequences (length 10000), for three sets of alignment parameters. The frequency has a 5% chance of being outside the dashed lines (2.5% each).

the unique value that satisfies eq. 12 (see the Appendix). `last-train` is not necessarily the best way to get the probabilities. Another option is to use BLOSUM or PAM substitution probabilities (π_{xy}), which are in the NCBI C++ toolkit (version 25.2, in `matrix_frequency_data.c`).

The `last-train` output files from this study are at https://gitlab.com/mcfrith/sum-align. There too is a simple Python script to help use these ideas. It takes as input α_D, α_I, β_D, β_I, and γ, finds $\omega_D = \omega_I$ that satisfies eq. 12, and outputs a'_D, a'_I, b'_D, b'_I, and $\frac{\gamma}{\omega_D \omega_I}$. LAST includes code for calculating the non-heuristic similarity score (used for estimating K), in a self-contained file (`mcf_alignment_path_adder.cc`).

Results

Similarity scores occurring by chance

Let us test whether we can calculate probabilities of similarity scores between random sequences. Three sequence-search scenarios are considered, with typical DNA, `at`-rich DNA, and proteins.

The first scenario is finding ancient (shared with reptiles) repeats in the human genome. This can be done by comparing the genome to repeat consensus sequences from Dfam [31]. Some are relics of transposable elements, others are source unknown. The match/mismatch/gap probabilities were found by comparing the genome to the consensus sequences with `last-train`. Then, pairs of random i.i.d. sequences were generated, with monomer probabilities ϕ_x and ψ_y equal to those found by `last-train`. For each pair of sequences, the highest score $s_{\max}$ was found by the non-heuristic algorithm. The observed scores are consistent with their probabilities predicted by eq. 13 (fig. 3 left).

The next scenario was to find related parts of the *Plasmodium falciparum* and *Plasmodium yoelii* genomes, which are both about 80% `a`+`t`. Alignment parameters were found by `last-train`, then similarity scores were found between random `at`-rich sequences (fig. 3 middle). Again, the observed scores are consistent with eq. 13.

The third scenario was to find related parts of proteins from *Aquifex aeolicus* (a heat-loving bacterium) and *Pyrolobus fumarii* (a heat-loving archaeon). Alignment parameters were found by `last-train`, then similarity scores were found between random sequences (fig. 3 right). Yet again, the observed scores are consistent with eq. 13.

Related versus biased sequences

A high similarity score could occur because the sequence regions have a common ancestor, and/or simi-

```
A tccttcctctcttttcttcccttccctcttctcttcccttcacttccctttctcttctctccctc       C acttgatcttagccaaaaggccgagaagcgat
  |  || | | ||| || | ||||  |||||||||||| ||  |||  ||||||||| | |||| ||         ||||||||||||||||||||||||||||||||
  tnttttcccccttctcctaccttntctcttctcttccattggcttttcttttctctgccctccttc         acttgatcttagccaaaaggccgagaagcgat

B cctgtttggctcctggagtcttagcagagaccaccagcaggggagctaattactgccag-------------tctgtaatgtggacctccgtcc
  ||  ||  |  | |||  |||  || |||||  ||| || ||| || ||||| |||||              | ||| ||||||||  | ||||
  cccattcagtccttggcctctctgctgagactgccaacaagggtgccaattagtgccaattgtgatggtggttttgtgatgtggacacctgtcc
```

Figure 4: Alignment between parts of: **A** reversed human chromosome Y (upper), and UCON1 (lower). This is the highest-scoring alignment in the top-left panel of fig. 5. **B** Human chromosome 6 (upper) and LFSINE_Vert (lower). **C** Human chromosome 21 (upper) and human U2 (lower).

lar composition bias (fig. 4A). We can investigate this by comparing two biological sequences after reversing (but not complementing) one of them (but see [32]). Then there are no segments related by evolution, but there may be similarities due to composition bias (fig. 4A).

Thus, the reversed human genome was searched against the repeat consensus sequences, the reversed *P. yoelii* genome against *P. falciparum*, and reversed *P. fumarii* proteins against *A. aeolicus* proteins. In each case, alg. 2 found more high similarity scores than expected between random sequences (fig. 5 upper row), due to biased regions (fig. 4A).

For standard alignment, these similarities can be avoided by detecting biased regions with `tantan` [33], and "masking" them. An effective masking scheme is to change $S_{xy} \leftarrow \min(S_{xy}, 0)$ when x or y is in a biased region [34]. For alg. 2, this masking scheme was tried: $S'_{xy} \leftarrow \min(S'_{xy}, 1)$.[1] With this masking scheme, the similarity scores between reversed and non-reversed sequences are close to what is expected between random sequences (fig. 5 lower row).

The new method can be more sensitive

Related sequence parts were sought by standard alignment (alg. 1) and by the new method (alg. 2), with biased-sequence masking. The two methods often find identical alignments, but with different scores and E-values. The new method tended to assign lower E-values, for alignments of the human genome to repeat consensus sequences (fig. 6A), and *P. fumarii* proteins to *A. aeolicus* proteins (fig. 6B). For example, both methods found the same alignment between human chromosome 6 and the LFSINE_Vert consensus (fig. 4B): the alg. 1 E-value is 22, and the alg. 2 E-value is 0.000031. This means the new method can find related sequence parts more confidently.

In other words, at a fixed E-value cutoff, the new method predicts more related regions. The preceding results indicate that the E-values are not overconfident. This implies that the new method finds more true positives for a given false positive rate.

The new method can be less sensitive

Another sequence search scenario was considered: finding U2 fragments in the human genome. U2 is one of the small nuclear RNAs that splice introns. The genome has many fragmentary copies of it, which are thus considered repeats.

The human genome was searched against a U2 DNA sequence, using the match/mismatch/gap probabilities found earlier for the human genome and ancient repeat consensus sequences. This time, alg. 2 tended to assign higher E-values than alg. 1 (fig. 6C). For example, both methods found the alignment shown in fig. 4C: the alg. 1 E-value is 0.00021, and the alg. 2 E-value is 0.0011.

The likely explanation is as follows. The genome has many short, near-exact copies of the first 30–40 bases of U2. The match/mismatch/gap probabilities, however, reflect greater sequence divergence (e.g. fig. 4B). These mistuned probabilities pessimize both algorithms, but they pessimize alg. 2 more. That is because probabilities with greater divergence attach more weight to alternative alignments.

This suggests that alg. 2 should work better than alg. 1 if the probabilities are a good fit to the se-

[1] Another untested idea is $S'_{xy} \leftarrow \min(S'_{xy}, \frac{\gamma}{\omega_D \omega_I})$.

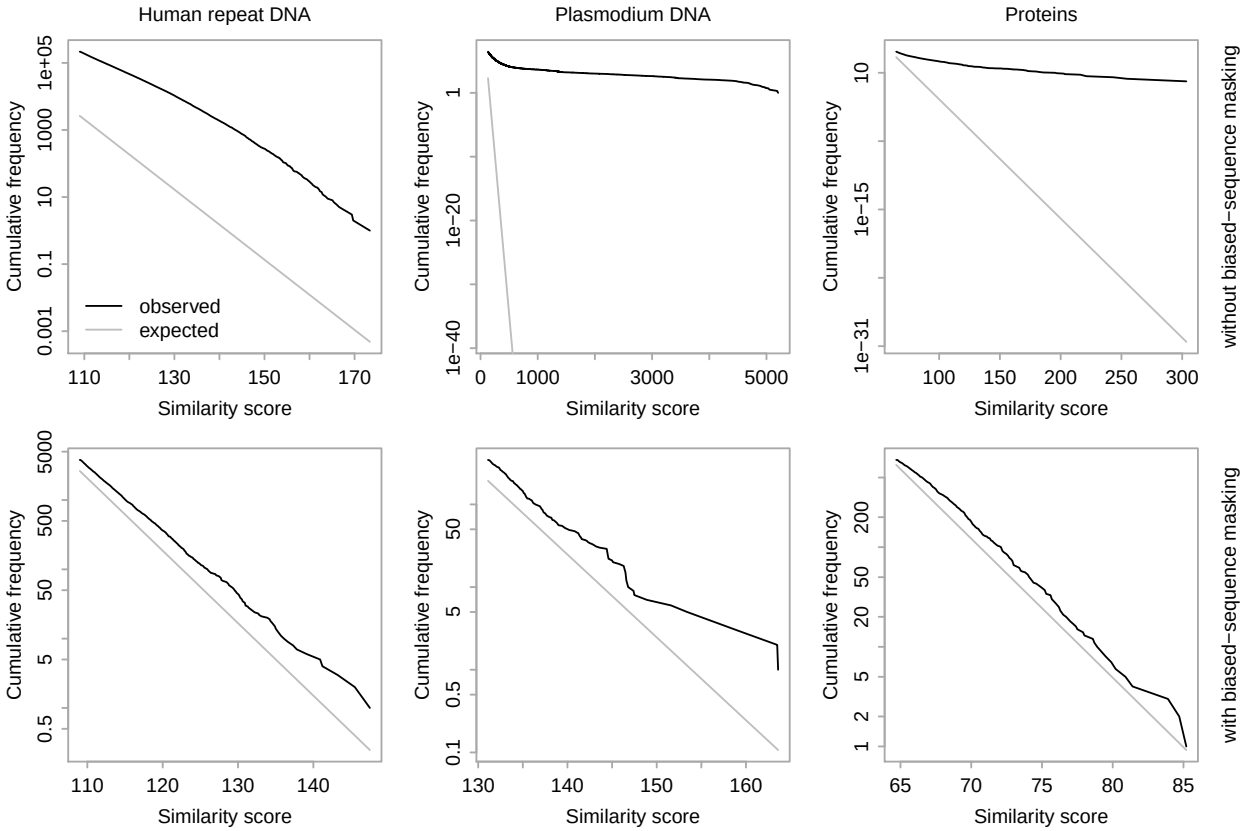

Figure 5: Heuristic similarity scores, between reversed and non-reversed sequences, with or without biased-sequence masking.

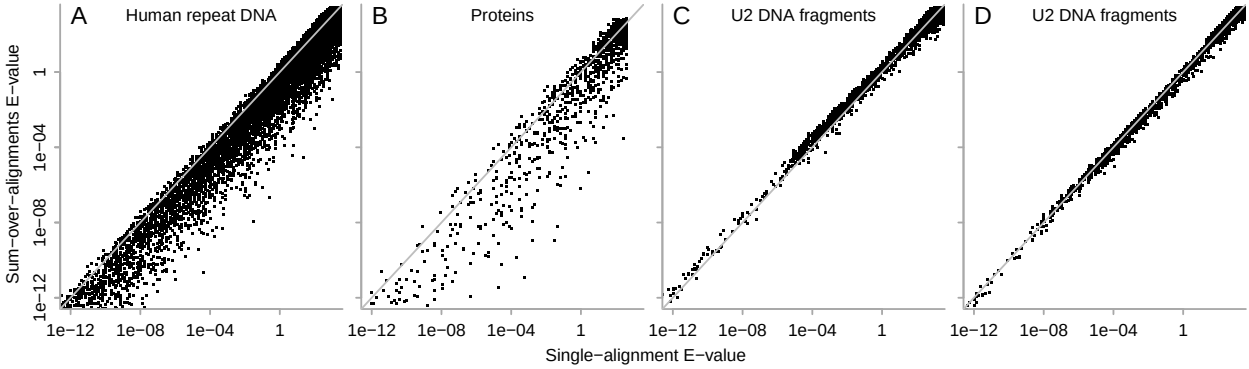

Figure 6: E-values for identical alignments found by alg. 1 (horizontal axis), and alg. 2 (vertical axis). Each point is one alignment. The diagonal gray lines indicate equal E-values. These E-values were calculated with m = total length of all reference sequences (e.g. all *A. aeolicus* proteins), and n = total length of all query sequences (e.g. all *P. fumarii* proteins). For DNA, one query strand was searched against both reference strands, so m was multiplied by 2.

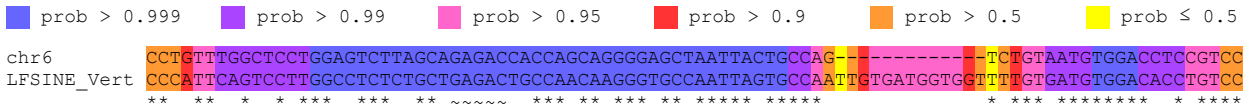

Figure 7: An alignment (the same as fig. 4B) with colors indicating the probability that each column is correct. ~ indicates the "core" (fig. 1), whose column probabilities are assumed to be 1. ⋆ indicates non-core exact matches.

quences. To test this, match/mismatch/gap probabilities were found by comparing the genome to the U2 sequence with `last-train`. Then, the genome was searched against U2 using these probabilities. This time, alg. 2 tends to assign slightly lower E-values than alg. 1 (fig. 6D). Even with well-tuned probabilities, the advantage of alg. 2 is expected to decrease for more similar sequences, because the evidence for relatedness is more concentrated in one alignment.

Discussion

We have seen how to find related parts of sequences, by summing probabilities of alternative alignments between them. This is better than standard alignment at detecting subtle relationships, at least in a few tests. It can be worse than standard alignment, when the probabilities are tuned for high divergence but the sequences have low divergence. The new method is more beneficial for higher divergence, where alternative alignments contribute more. In a spectacular example, it found gene-regulating DNA conserved in diverse animals since the Precambrian [35].

A major further benefit is that the new method simplifies E-value calculation. This makes it easier to experiment with different alignment parameters, beyond the BLAST parameter sets. This can be useful for proteins [36], but especially for DNA, because there has been less effort to optimize DNA parameters.

Moreover, the new method can be generalized to other kinds of alignment. For example, we used the same approach for DNA-versus-protein alignment [37]. We defined probabilities similarly to fig. 2, but with extra probabilities for frameshifts. We summed probabilities of alternative alignments, and accurately predicted similarity scores between random sequences. This method was highly sensitive: it found ancient and degraded relics of Paleozoic mobile elements in vertebrate genomes. Some previous methods have used two kinds of alignment gap: short-and-frequent and long-and-rare [15, 18]. This can be done by elaborating fig. 2. The approach of this paper is predicted to work in that case too. This makes it easier to try such different kinds of alignment, in particular because it provides E-values. Another alignment elaboration is position-specific match, mismatch, and gap rates [2, 4, 38, 39]: this was incorporated into hybrid alignment [40].

By combining alg. 2 with another similar algorithm (see the Supplement), we can calculate the probability that each alignment column is correct (fig. 7), based on the assumed rates of matches, mismatches, and gaps [2, 15]. It is also possible to make alignments based on these probabilities: for example, align monomers whose alignment probability is > 0.5 [41].

For short sequences, the E-values are too high. Eq. 15 assumes the number of chance similarities is proportional to mn. That becomes too high for short sequences, because a similarity must have non-zero length and fit in the sequences. There are corrections for this effect, including in hybrid alignment, but it is somewhat complex [21].

We should consider some basic aims and assumptions. Suppose we seek related regions between a genome of length m, and q "query" sequences with lengths $n_1, n_2, \ldots, n_q$. We could assume that related regions are equally likely to occur per query sequence, or per unit sequence length. We could aim to find relationships for as many query sequences as possible, or as many related regions as possible. If part of query i is similar to part of the genome, we can calculate an E-value from eq. 15 with sequence lengths m and n_i. This increasingly disfavors related regions in increasingly long queries. As mentioned in the Introduction, some previous methods similarly penalize

related regions in longer sequences. The alternative is to use eq. 15 with sequence lengths m and $\sum_{j=1}^{q} n_j$. This treats related regions equally, regardless of the containing sequence's length. The aim is important when testing sequence search methods, because the test may count the number of sequences or related regions found.

Acknowledgments

I am grateful to John Spouge for encouragement that the E-values would be simple, Yi-Kuo Yu for advice about hybrid alignment, and Travis Wheeler for advice about HMMER.

References

1. Dayhoff, M, Schwartz, R & Orcutt, B. A model of evolutionary change in proteins. *Atlas of protein sequence and structure* **5,** 345–352 (1978).
2. Durbin, R, Eddy, SR, Krogh, A & Mitchison, G. *Biological sequence analysis: probabilistic models of proteins and nucleic acids* (Cambridge University Press, 1998).
3. Frith, MC. How sequence alignment scores correspond to probability models. *Bioinformatics* **36,** 408–415 (2020).
4. Eddy, SR. A probabilistic model of local sequence alignment that simplifies statistical significance estimation. *PLoS Computational Biology* **4,** e1000069 (2008).
5. Wheeler, TJ & Eddy, SR. nhmmer: DNA homology search with profile HMMs. *Bioinformatics* **29,** 2487–2489 (2013).
6. Karlin, S & Altschul, SF. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences* **87,** 2264–2268 (1990).
7. Altschul, SF *et al.* Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research* **25,** 3389–3402 (1997).
8. Harris, RS. *Improved pairwise alignment of genomic DNA*. PhD thesis (The Pennsylvania State University, 2007).
9. Sheetlin, S, Park, Y, Frith, MC & Spouge, JL. ALP & FALP: C++ libraries for pairwise local alignment E-values. *Bioinformatics* **32,** 304–305 (2016).
10. Bishop, M & Thompson, EA. Maximum likelihood alignment of DNA sequences. *Journal of molecular biology* **190,** 159–165 (1986).
11. Thorne, JL, Kishino, H & Felsenstein, J. An evolutionary model for maximum likelihood alignment of DNA sequences. *Journal of Molecular Evolution* **33,** 114–124 (1991).
12. Thorne, JL & Churchill, GA. Estimation and reliability of molecular sequence alignments. *Biometrics,* 100–113 (1995).
13. Hein, J, Wiuf, C, Knudsen, B, Møller, M & Wibling, G. Statistical alignment: computational properties, homology testing and goodness-of-fit. *Journal of Molecular Biology* **302,** 265–279 (2000).
14. Knudsen, B & Miyamoto, MM. Sequence alignments and pair hidden Markov models using evolutionary history. *Journal of molecular biology* **333,** 453–460 (2003).
15. Allison, L, Wallace, C & Yee, C. Finite-state models in the alignment of macromolecules. *Journal of Molecular Evolution* **35,** 77–89 (1992).
16. Bucher, P & Hofmann, K. *A sequence similarity search algorithm based on a probabilistic interpretation of an alignment scoring system* in *Proc Int Conf Intell Syst Mol Biol.* (1996), 44–51.
17. Webb, BJM, Liu, JS & Lawrence, CE. BALSA: Bayesian algorithm for local sequence alignment. *Nucleic Acids Research* **30,** 1268–1277 (2002).
18. Lunter, G *et al.* Uncertainty in homology inferences: assessing and improving genomic sequence alignment. *Genome research* **18,** 298–309 (2008).
19. Hudek, AK & Brown, DG. FEAST: sensitive local alignment with multiple rates of evolution. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **8,** 698–709 (2010).
20. Neyman, J & Pearson, ES. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* **231,** 289–337 (1933).

21. Yu, YK & Hwa, T. Statistical significance of probabilistic sequence alignment and related local hidden Markov models. *Journal of Computational Biology* **8,** 249–282 (2001).

22. Ma, B, Tromp, J & Li, M. PatternHunter: faster and more sensitive homology search. *Bioinformatics* **18,** 440–445 (2002).

23. Noé, L & Kucherov, G. Improved hit criteria for DNA local alignment. *BMC Bioinformatics* **5,** 1–9 (2004).

24. Roytberg, M *et al.* On subset seeds for protein alignment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **6,** 483–494 (2009).

25. Zhang, Z, Berman, P & Miller, W. Alignments Without Low-Scoring Regions. *Journal of Computational Biology* **5,** 197–210 (1998).

26. Suzuki, H & Kasahara, M. Acceleration of nucleotide semi-global alignment with adaptive banded dynamic programming. *bioRxiv* (2017).

27. Guidi, G, Ellis, M, Rokhsar, D, Yelick, K & Buluç, A. *BELLA: Berkeley efficient long-read to long-read aligner and overlapper* in *SIAM Conference on Applied and Computational Discrete Algorithms (ACDA21)* (2021), 123–134.

28. Liu, D & Steinegger, M. Block Aligner: an adaptive SIMD-accelerated aligner for sequences and position-specific scoring matrices. *Bioinformatics,* btad487 (2023).

29. Cameron, M, Williams, HE & Cannane, A. Improved gapped alignment in BLAST. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **1,** 116–129 (2004).

30. Hamada, M, Ono, Y, Asai, K & Frith, MC. Training alignment parameters for arbitrary sequencers with LAST-TRAIN. *Bioinformatics* **33,** 926–928 (2017).

31. Storer, J, Hubley, R, Rosen, J, Wheeler, TJ & Smit, AF. The Dfam community resource of transposable element families, sequence models, and genome annotations. *Mobile DNA* **12,** 1–14 (2021).

32. Glidden-Handgis, GE & Wheeler, TJ. WAS IT A MATch I SAW? Approximate palindromes lead to overstated false match rates in benchmarks using reversed sequences. *bioRxiv* (2023).

33. Frith, MC. A new repeat-masking method enables specific detection of homologous sequences. *Nucleic Acids Research* **39,** e23–e23 (2011).

34. Frith, MC. Gentle masking of low-complexity sequences improves homology search. *PLoS One* **6,** e28819 (2011).

35. Frith, MC & Ni, S. DNA conserved in diverse animals since the Precambrian controls genes for embryonic development. *Molecular Biology and Evolution,* msad275 (2023).

36. Ng, PC, Henikoff, JG & Henikoff, S. PHAT: a transmembrane-specific substitution matrix. *Bioinformatics* **16,** 760–766 (2000).

37. Yao, Y & Frith, MC. Improved DNA-Versus-Protein Homology Search for Protein Fossils. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **20,** 1691–1699 (2023).

38. Steinegger, M *et al.* HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinformatics* **20,** 1–15 (2019).

39. Roddy, JW, Rich, DH & Wheeler, TJ. nail: software for high-speed, high-sensitivity protein sequence annotation. *bioRxiv* (2024).

40. Yu, YK, Bundschuh, R & Hwa, T. Hybrid alignment: high-performance with universal statistics. *Bioinformatics* **18,** 864–872 (2002).

41. Miyazawa, S. A reliable sequence alignment method based on probabilities of residue correspondences. *Protein Engineering, Design and Selection* **8,** 999–1009 (1995).

Appendix

Conservation condition

Suppose we compare two random i.i.d. sequences, with monomer probabilities Φ_x and Ψ_y. The conservation condition is

$$\left(\sum_{x,y} \Phi_x \Psi_y S'_{xy}\right) + \frac{a'_D}{1-b'_D} + \frac{a'_I}{1-b'_I} = 1\,. \qquad (16)$$

In the special case where $\Phi_x = \phi_x$ and $\Psi_y = \psi_y$, eq. 16 is the same as eq. 12.

The conservation condition depends on the precise definition of gap probabilities (fig. 2), as explained previously [3]. The conservation condition for hybrid alignment was different [21, eq. 28], because they defined gap probabilities differently [3, fig. S2].

The conservation condition generalizes an equation in the proven E-value solution for gapless alignment [6, eq. 4].

Finding $\omega_D = \omega_I$ that satisfies eq. 12

First, assume that $\omega_D > \beta_D$ and $\omega_I > \beta_I$. Otherwise, alignment gaps would have non-negative scores, which is not useful for sequence comparison.

As $\omega_D = \omega_I$ increases from $\max(\beta_D, \beta_I)$ to 1, the LHS of eqn. 12 decreases, from a value > 1 to a value < 1. So it equals 1 at a unique point, which can be found by e.g. repeated bisection.

prob(null alignment)

If we define gap probabilities as in fig. 2, a null alignment is produced by a path that never traverses the α_D, α_I, or γ arrows. Suppose we compare a length-m sequence R to a length-n sequence Q.

$$\text{prob(null alignment)} = \left(\prod_{k=1}^{m} \omega_D \phi_{R_k}\right)\left(\prod_{k=1}^{n} \omega_I \psi_{Q_k}\right) (1-\omega_D)^2(1-\omega_I)^2(1-\gamma-\alpha_D-\alpha_I)\,. \quad (17)$$