Computing Posterior Probabilities for Score-based Alignments Using ppALIGN

Stefan Wolfsheimer[1], Alexander K. Hartmann[2], Ralf Rabus[3], and Gregory Nuel[1,4]

1: Dpt. of Applied Mathematics (MAP5), University of Paris Descartes, France;

2: Institut für Physik, Universität Oldenburg, Germany;

3: Institut für Chemie und Biologie des Meeres, Universität Oldenburg, Germany;

4: Institute for Mathematical Sciences (INSMI), CNRS, Paris, France.

February 17, 2012

Abstract

Score-based pairwise alignments are widely used in bioinformatics in particular with molecular database search tools, such as the BLAST family. Due to sophisticated heuristics, such algorithms are usually fast but the underlying scoring model unfortunately lacks a statistical description of the reliability of the reported alignments. In particular, close to gaps, in low-score or low-complexity regions, a huge number of alternative alignments arise which results in a decrease of the certainty of the alignment.

`ppALIGN` is a software package that uses hidden Markov Model techniques to compute position-wise reliability of score-based pairwise alignments of DNA or protein sequences. The design of the model allows for a direct connection between the scoring function and the parameters of the probabilistic model. For this reason it is suitable to analyze the outcomes of popular score based aligners and search tools without having to choose a complicated set of parameters. By contrast, our program only requires the classical score parameters (the scoring function and gap costs). The package comes along with a library written in C++, a standalone program for user defined alignments (`ppALIGN`) and another program (`ppBLAST`) which can process a complete result set of BLAST. The main algorithms essentially exhibit a linear time complexity (in the alignment lengths), and they are hence suitable for on-line computations. We have also included alternative decoding algorithms to provide alternative alignments.

ppALIGN is a fast program/library that helps detect and quantify questionable regions in pairwise alignments. Due to its structure, the input/output interface it can to be connected to other post-processing tools. Empirically, we illustrate its usefulness in terms of correctly predicted reliable regions for sequences generated using the ROSE model for sequence evolution, and identify sensor-specific regions in the denitrifying betaproteobacterium *Aromatoleum aromaticum* EbN1.

1 Introduction

Many search tools for large molecular databases, such as the BLAST family [Altschul, Gish, Miller, Myers, and Lipman (1990)], are fast search heuristics which rely on score-based alignments. Usually, for a given query, a list of alignments are reported in decreasing order of significance, essentially in linear time.

Since score-based aligners produce the alignments that maximize the score, they do not compute alternative optimum or suboptimum alignments and hence fail to provide a statistical analysis of the accuracy of the produced alignment. To address this problem, various probabilistic alignment methods, such as pair hidden Markov Models (HMMs) or the finite-temperature alignment (FTA), were developed in the last decade [Durbin, Eddy, Krogh, and Mitchison (1998)]. They provide a statistical description of the set of all the alignments for a given pair of sequences including alternative meaningful alignments that may be hidden behind the optimum. For instance, in the framework of the probabilistic alignment, we can assess numerically the confidence for each aligned pair of letters and gaps. This allows us to identify questionable regions in the alignment.

FTA was introduced in 1995, [Miyazawa (1995), Zhang and Marr (1995), Kschischo and Lässig (2000)] and attributes weights of an exponential (or Boltzmann) distribution to given alignments. It can directly be applied to any classical scoring function with one additional parameter, the temperature T (contrast parameter). Using the canonical value $T = 1$, and a normalized scoring function, FTA it approximates more complex probabilistic models.

On the other hand, standard pair HMMs [Durbin et al. (1998)] provide a stronger probabilistic description because each parameter can be explained as a transition, or an emission probability. However, the typical model layout is usually much larger than score-based alignments and it is hard to find a one-to-one relationship between both approaches. Usually, it is possible to derive a scoring function from pair HMMs, but the reverse is not always possible [Durbin et al. (1998), Arribas-Gil, Gassiat, and Matias (2006)].

With standard pair HMMs (ex: [Durbin et al. (1998)]), the length of the sequences is not explicitly modeled. But it is possible to derive pair HMMs with a reduced set of parameters thanks to a conditioning on the sequence lengths. Yu and Hwa [Yu and Hwa (2001)] is an example of this type of HMM.

Lunter et. al. [Lunter, Rocco, Mimouni, Heger, Caldeira, and Hein (2008)] illustrated the usefulness of probabilistic alignments by detecting regions of low confidence. Many competitive alignments decrease the accuracy of the maximum score alignment, especially close to gaps (i.e., insertions or deletions). These biases have been identified as "gap wander" [Holmes and Durbin (1998)], "gap attraction", and "gap annihilation" [Lunter (2007b)]. Gap wander describes the effect of an inferred gap position which differs from the "true alignment" by a few pairs. Gap attraction occurs when two close gaps merge into a single gap in the inferred alignment, and the third effect is a cancellation of an insertion and a deletion.

In this work, we introduce `ppALIGN`, which uses either the FTA or a pair HMM to compute posterior probabilities. To the best of our knowledge, `ppALIGN` is the first approach and the only practical tool which can directly compute posterior probabilities and alternative alignments from alignments obtained with standard score-based methods such as BLAST. This is done *consistently*, hence the posterior probabilities are based on the same scoring scheme as the original BLAST calculations. For the FTA, the temperature can be tuned as an additional parameter, but it is always possible to use the canonical value $T = 1$ in order to be close to a full probabilistic model. Our pair HMM is close to the model of Yu and Hwa [Yu and Hwa (2001)], in particular it has the same number of free parameters as the score-based alignment. Nevertheless, Yu and Hwa did not calculate posterior probabilities, neither did they provide a numerical tool to analyze actual alignments.

Note that there are indeed some tools which are related to `ppALIGN` and that compute posterior probabilities as well, such as `StatAlign` [Novák, Miklós, Lynsgoe, and Hein (2008)], `BAli-Phy` [Suchard and Redelings (2006)] or `HMMoC` [Lunter (2007a)]. However our approach goes further than previous ones in many ways. For example, the `StatAlign` tool does not take alignments as input but isolated sequences, hence it is not suitable for the post-processing of BLAST and similar outputs. Furthermore, its insertion/deletion model is based on the TFK92 model [Thorne, Kishino, and Felsenstein (1992)], which can only handle the standard affine gap costs approximately. `BAli-Phys` on the other hand allows for alignments as input, but does not directly support score-based alignment analysis making it inconsistent when applied to standard alignment output. Finally `HMMoC`, offers sampling and calculation of posterior probabilities for arbitrarily defined HMMs, but is not specifically adapted to generate or analyze alignments and requires a complete specification/programming with the XML language.

Hence, in contrast to other software packages, `ppALIGN` offers a direct and effortless analysis of alignment outputs using standard score-based methods. Of course, our program can be run locally, but in contrast with all the programs mentioned above, it can also be used online via a user-friendly web access, which makes it particularly convenient for practitioners. The software presented here can process either a single alignment or the entire output of BLAST. Furthermore it includes an interface to integrate new features such as alternative posterior decoding algorithms as proposed by [Miyazawa (1995), Holmes and Durbin (1998), Durbin et al. (1998), Lunter et al. (2008)].

We start Section 2 with some reminders on both score-based and probabilistic alignments, and we introduce our notations. In Section 3, we introduce the score-based equivalent pair HMM which we have developed. In Section 4 we discuss the practical implementation of `ppALIGN`. Then finally in Section 5 we present and discuss three illustrations of the usefulness of our software. Two of these illustrations are simulation studies, while the last one is done on real data. Relevant mathematical details can be found in the appendix.

2 Reminders on pairwise alignment

2.1 Score-based alignment

Let $a_1^\ell = a_1 \dots a_\ell \in \Sigma^\ell$ and $b_1^m = b_1 \dots b_m \in \Sigma^m$ denote a pair of sequences over the finite alphabet Σ (either nucleotides or amino acids). An alignment $\pi_1^t = \pi_1 \dots \pi_t$ of a_1^ℓ and b_1^m is a sequence of edit operations with $\pi_k \in \{\mathtt{P}, \mathtt{A}, \mathtt{B}\}$ such that, with $F(\mathtt{P}) = (1,1), F(\mathtt{A}) = (1,0), F(\mathtt{B}) = (0,1)$, $\sum_{k=1}^t F(\pi_k) = (\ell, m)$. The operation `P` is referred to as pair, the operation `A` as insertion in sequence a_1^ℓ and the operation $\texttt{B}$ as insertion[1] in sequence b_1^m. If we define $(i(k), j(k)) = \sum_{k'=1}^{k} F(\pi_{k'})$, we note that $a_{i(k)}$ and $b_{j(k)}$ give the last pair of letters in the alignment position k. If $\pi_k = \texttt{P}$ those letters are paired. A more formal definition is given in Appendix A.1.

Score-based methods determine the optimal global alignment $\hat{\pi}$ by maximizing an objective function s, $\hat{\pi} = \operatorname{argmax}_\pi s(\pi; a_1^\ell, b_1^m)$. The local version Local [Smith and Waterman (1981)] being $\hat{\pi} = \operatorname{argmax}_{i_1 \le i_2, j_1 \le j_2} \; s(\pi; a_{i_1}^{i_2}, b_{j_1}^{j_2})$.

Let $\mathcal{S}(a, b)$ denote the classical scoring function which assigns an integer number, $\mathcal{S} : \Sigma \times \Sigma \to \mathbb{Z}$ to each pair of letters. Given the penalty d for a gap open, and a penalty e for a gap extension[2], the objective function for Needleman-Wunsch global alignments [Needleman and Wunsch (1970)] is classically defined by

$$s(\pi_1^t; a_1^\ell, b_1^m) = \sum_{k, \pi_k = \texttt{P}} \mathcal{S}(a_{i(k)}, b_{j(k)}) + \sum_{k=1}^{t} \tilde{s}_{\pi_{k-1}\pi_k}, \tag{1}$$

where $\tilde{s}_{\texttt{PP}} = \tilde{s}_{\texttt{AP}} = \tilde{s}_{\texttt{BP}} = 0$, $\tilde{s}_{\texttt{PA}} = \tilde{s}_{\texttt{PB}} = \tilde{s}_{\texttt{BA}} = -d - e$, $\tilde{s}_{\texttt{AB}} = -\infty$, and $\tilde{s}_{\texttt{AA}} = -e = \tilde{s}_{\texttt{BB}} = -e$. By convention, note that we set $\pi_0 = \texttt{P}$ without loss of generality.

2.2 Probabilistic alignmment

Probabilistic alignment methods go beyond the optimum and consider the set of possible alignments weighted with the so-called posterior distribution

$$\mathbb{P}\left(\Pi = \pi \left| a_1^\ell, b_1^m \right.\right). \tag{2}$$

In cases where the optimal alignment agrees undoubtedly with the true (unknown) alignment, virtually all weight is put on the optimal alignment. When less similar sequences are compared to each other there might be regions of low confidence where letters might be aligned incorrectly or gaps misplaced. The posterior distribution Eq. 2 is appropriate to quantify the degree of confidence for a given alignment.

`ppALIGN` uses pair-HMM techniques to marginalize the posterior distribution of Eq. 2 and determine *column-wise posterior probabilities* [Durbin et al.

[1] or conversely, $\texttt{A}$ as deletion in sequence b_1^m, and $\texttt{B}$ as deletion in sequence a_1^ℓ, for this reason we call such events *indel*.

[2] A gap of length γ is penalized with $-d - e\gamma$.

(1998)]. Let us assume that the optimal alignment relates position a_i in the first sequence with position b_j in the second sequence, or, according to our alignment definition, $\pi_k = \mathtt{P}$ and $(i(k), j(k)) = (i, j)$. The confidence that this pair is aligned correctly can be assessed by the marginal posterior probability for this event $P^{\mathtt{P}}_{i,j} = \mathbb{P}\left(\pi : a_i \text{ and } b_j \text{ aligned } \left| a_1^\ell, b_1^m \right.\right)$. Concerning the gaps we follow the definition of Lunter et al. (2008) and define the probability $P^{\mathtt{A}}_i = \mathbb{P}\left(\pi : a_i \text{ gapped } \left| a_1^\ell, b_1^m \right.\right)$ that the position i in a the sequence a is related to a gap in the sequence b and the same applies to gaps in the other sequence with: $P^{\mathtt{B}}_j = \mathbb{P}\left(\pi : b_j \text{ gapped } \left| a_1^\ell, b_1^m \right.\right)$.

In the case of the local alignment, two points can be questioned. First, how sure we can we be that the starting and ending of the aligned part are correct? Secondly, how accurate is the alignment itself? To address the second question we simply turn to the global alignment problem assuming that the starting and ending are correct. Concerning the confidence of the boundaries of the local alignment, `ppALIGN` computes the marginal probabilities $P^{\text{start}}_{i,j} = \mathbb{P}\left(\pi \text{ starts in } a_i \text{ and } b_j \left| a_1^\ell, b_1^m \right.\right)$ and $P^{\text{end}}_{i,j} = \mathbb{P}\left(\pi \text{ ends in } a_i \text{ and } b_j \left| a_1^\ell, b_1^m \right.\right)$.

The probabilities $P^{\mathtt{P}}_{i,j}$, $P^{\mathtt{A}}_i$, $P^{\mathtt{B}}_j$, $P^{\text{start}}_{i,j}$, and $P^{\text{end}}_{i,j}$ can be efficiently computed by forward and backward algorithms [Rabiner (1989), Durbin et al. (1998)] for a pair HMM. See Appendix A.2 and A.3 for more details.

3 The score-based equivalent pair HMM

In the context of the alignment, the pair HMM is a generative model assuming: 1) that the unobserved alignment π is generated according to a Markov chain with transition τ; 2) that State $\mathtt{P}$ emits a pair (a, b) of symbols with the probability $p(a, b)$, State $\mathtt{A}$ a pair $(a, -)$ with the probability $q(a)$, and State $\mathtt{B}$ a pair $(-, b)$ with the probability $q(b)$.

The layout of the pair HMM is adjusted such that a direct connection to the corresponding score-based alignment method can be made. This is possible, if the gap costs are not too small. When $\mathcal{S}(a, b)$ is derived from a likelihood ratio $\mathcal{S}(a, b) = \lambda \log \frac{p(a,b)}{q(a)q(b)}$ (like the BLOSUM and PAM families) with the pair probabilities p, the background probabilities q, and the scale λ the pair emission probabilities of the HMM is simply set to $p(a, b) = \exp\left(\mathcal{S}(a, b)/\lambda\right) q(a)q(b)$.

If we denote by $\alpha = d + e$ the total gap-opening cost and by $\beta = e$ the gap-extension cost we can express the transition matrix τ over $\{\mathtt{P}, \mathtt{A}, \mathtt{B}\}$ (in this order) as:

$$\tau = \begin{pmatrix} 1-2\nu & \nu_{\mathtt{A}} & 2\nu - \nu_A \\ (1-2\nu)\eta_{\mathtt{A}} & \varepsilon & 0 \\ (1-2\nu)\eta_{\mathtt{B}} & 1-\varepsilon-(1-2\nu)\eta_{\mathtt{B}} & \varepsilon \end{pmatrix}$$

where $(\nu, \nu_{\mathtt{A}}, \eta_{\mathtt{A}}, \eta_{\mathtt{B}}, \varepsilon)$ is the unique solution of the following system of five equations:$\nu_{\mathtt{A}}\eta_{\mathtt{A}} = (2\nu - \nu_A)\nu_{\mathtt{B}} = \exp(-\alpha/\lambda)$, $\varepsilon = \exp(-\beta/\lambda)$, $(1-2\nu)\eta_{\mathtt{A}} + \varepsilon = 0$, and $1 - \varepsilon = (1 - 2\nu + \varepsilon)\eta_{\mathtt{B}}$. See Appendix A.4 for more details.

However, for simple scoring functions we often only know $\mathcal{S}(a, b)$. In this case `ppALIGN` estimates $q(a)$ from the input sequences and determine λ by the unique root of the equation

$$\sum_{a,b} \exp\left(\mathcal{S}(a,b)/\lambda\right) q(a)q(b) = 1.$$

Beside the pair HMM, we implemented the FTA model [Miyazawa (1995), Zhang and Marr (1995), Kschischo and Lässig (2000)] for which alignments are weighted with an exponential distribution

$$\mathbb{P}\left(\pi \left| a_1^\ell, b_1^m \right.\right) \propto \exp\left[\frac{s(\pi; a_1^\ell, b_1^m)}{\lambda T}\right].$$

The free parameter T is termed as the temperature (a contrast parameter). For $T = 1$ the FTA model approximates the pair HMM. In the supplementary material we confirm that the differences between our pair HMM, the FTA and the HMM by Yu and Hwa [Yu and Hwa (2001)] are only marginal. In the limit $T \to 0$ essentially only optimal alignments have a positive probability, whereas for $T \to \infty$ all alignments have equal weight. The FTA model allows us to explore the alignment space more generally. For example, if a certain region in an alignment persists even for a larger temperature, we have a more conservative evidence.

4 Implementation

The layout of the software and the data flow are illustrated in Figure 1. The user provides a set of alignments either in FASTA format, or in the form of an entire BLAST output (in XML format) which is processed by the core library. The default output is again XML. Alternatively, one may choose a plain text

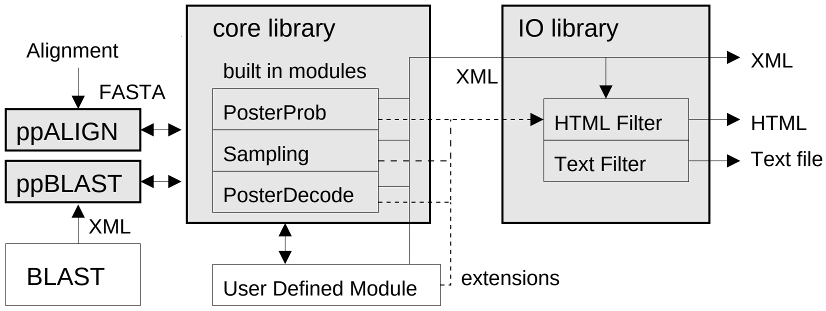

Figure 1: Layout of the software and data flow. The user may provide an alignment (local or global) in FASTA format or a structured BLAST output (XML format). The ppALIGN computes the posterior probabilities of the alignment and generates an XML stream as default output. Alternatively the user may choose a human-readable HTML or text output. Built-in and dynamically loadable modules perform further computations such as decoding of alternative alignments. They extend the XML stream of the core library and may specify certain formating rules for the output filters.

or an HTML output filter which format the output into a human-readable form.

In a UNIX environment, `ppALIGN` is typically called:

```
> ppalign -i alignment.fasta -o alignment.html -f html
```

The program reads an alignment in FASTA format (the option `-i`) and produces an HTML page (the option `-o`). Different output formats may be chosen with the option `-f`.

The main algorithm determines for each aligned column the marginal posterior probabilities as described above. In order to provide possible alternative alignments we implemented two additional modules in the core library of `ppALIGN`,

- a sampler that draws alignment from the posterior distribution [Durbin et al. (1998), Mückstein, Hofacker, and Stadler (2002)];

- a decoder which maximizes the posterior probability [Durbin et al. (1998), Holmes and Durbin (1998)].

The resulting alignments of both decoding methods are compared with the user supplied alignment and the regions that do not agree are pointed out. The alignment is partitioned in different segments where the alternative alignments are consistent with the user supplied alignment (referred to as *non-ambiguous* in what follows) and the segments where at least one alternative alignment is not consistent with the reference (referred to as *ambiguous segments*).[3] In the ambiguous segments, the user may switch between different alternative alignments and may therefore obtain further information about the structure of the weighted alignment space. In particular, the locations of the ambiguous segments are particularly suitable to detect regions of uncertainty, as we show in our "result" section.

Knowing the optimal alignment (or a nearly optimal alignment) reduce the quadratic time complexity of the algorithms can be reduced to a computation time which is essentially proportional to the length of the optimal alignment. This becomes possible because the further away from the optimal alignment the pir probabilities $P^{\mathtt{P}}_{i,j}$ are, the more negligible they are. By default, `ppALIGN` uses a heuristic where the forward and backward sums are only computed on a strip around the optimum (see Figure 2). The size of the strip is determined by successively increasing the offset between the alignment and the boundary of the computed area. The size is assumed to be sufficient when the relative change of the forward sum $\mathbb{P}(a_1^\ell, b_1^m)$ between the last two iterations are sufficiently small (say $\sim 10^{-8}$). Note that the strip method might not work when the algorithmic parameters are chosen in the so-called linear regime [Arratia and Waterman (1994), Kschischo and Lässig (2000), Wolfsheimer, Melchert, and Hartmann (2009)] which is easily signaled by a weak convergence in the procedure to estimate the strip size. This leads to a strip in the order of the quadratic search space. However, we advise not to choose the parameter in this regime because suboptimal alignments are usually given too much weight. It should also be noted that in general, the strip approach can fail in the presence of large duplicated segments of the sequence. However, since we start from the result of a local alignment algorithm that precisely takes care of such situations, the problem vanishes. In all cases, if the needed workspace exceeds the provided memory, the algorithms rely on Newberg's checkpoint method [Newberg (2008)].

During the computation, built-in or plug-in modules can handle intermediate results of the computation to provide additional information or alternative alignments. The module concept is designed such that further decoding

[3] In principle, if the posterior probability is not exactly $P = 1$, sampling a huge number of sequences will generate only ambiguous segments.

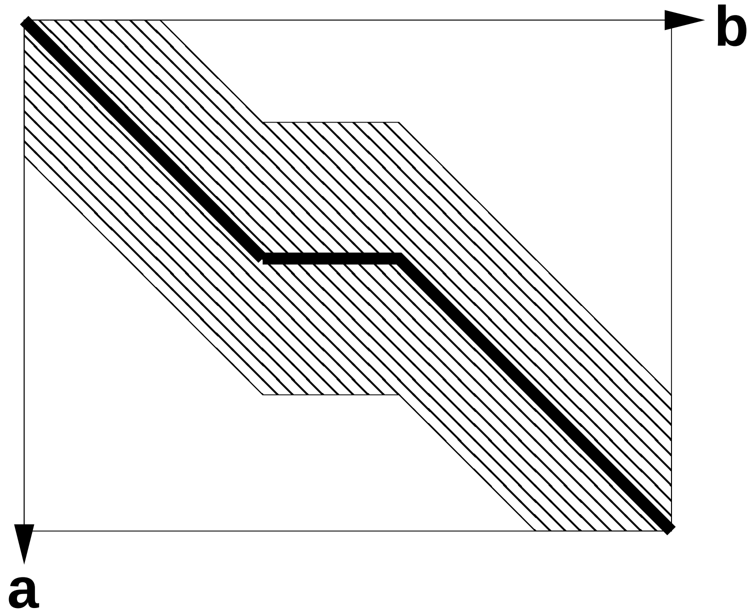

Figure 2: Restriction of the search space around the optimal alignment leads to linear complexity.

algorithms or other computations based on intermediate dynamic programming results can be added without changing the core library. Developers need not consider the model details (pair HMM or FTA) or think about the estimation of the strip size.

5 Results and discussion

5.1 Simulation: global alignment

To evaluate the ability of `ppALIGN` to identify uncertain regions of global alignments, we have performed computer simulations [Hartmann (2009)]. Using the `ROSE` model of random sequence evolution [Stoye, Evers, and Meyer (1998)], we have generated protein sequences according to a evolutionary tree given by a complete binary tree with three levels, hence exhibiting a total of $2^3 = 8$ leaves. At the root we always started with a sequence of 86 residue-long human insulin. Each branch of the tree, corresponding to a descendant was PAM D in length, which means that we performed D times mutations according the 1-PAM matrix [Dayhoff, Schwartz, and Orcutt (1979)] (ROSE default), and D times insertions and deletions with the probability $p_{\text{ins}} = p_{\text{del}} = 0.003$. These rates are about 10 times larger than the default value of ROSE, hence resulting in a larger number of gaps. The alignment problem is more difficult in this case and therefore more interesting for the purpose of quantifying its reliability. We performed simulations for $D = 10$, 20, and 40. As for the length distributions of insertions and deletions, we also took the ROSE default: lengths between one and six appear with probability 0.1, while lengths between 7 and 14 appear with probability 0.05. The ROSE output is, in our case, the generated sequences located at the leaves of the tree together with a full multiple alignment showing the true evolutionary history of the eight sequences.

In Figure 3, the alignment between two sample sequences is examined. We can observe that the optimum alignment (Figure 3b) is consistent with the true evolutionary alignment (Figure 3a) in most places. Note that most regions where all the alignments are the same have been omitted in order to improve readability. Typically, the posterior probabilities, obtained using the HMM in this case, are small where the optimum alignment is not consistent with the evolutionary true alignment, i.e., it is not *correct*.[4] Since the optimum alignment is the alignment with the maximum probability, the posterior probabilities are typically somewhat larger compared to the true alignment. Beside the optimal alignment, ppALIGN also displays alternative alignments, the maximum posterior alignment and 2 out of 10 sampled alignments, are shown in Figure 3c-e. As described above, we may regard the ambiguous segments as less reliable. For sampled alignments the user may choose the number of samples. One may expect longer and more ambiguous segments with an increasing number of samples. To our observation this is virtually only the case up to a value of about 10. Above this value it becomes more and more unlikely that a new sample should explore alternatives in the non-ambiguous regions (because of the large posterior probability there). For example, if we consider the true alignment and the first sample we will detect two small ambiguous segments. Including the second sample, these two segments merge into a larger one which is already the one displayed in Figure 3 apart from three positions. Without considering the maximum

[4] A pair of letters is correct, if two letter at the same positions are paired in the true alignment. A gapped letter is considered correct, if in the true alignment it is gapped as well, irrespective of the position of the gap [Lunter et al. (2008)].

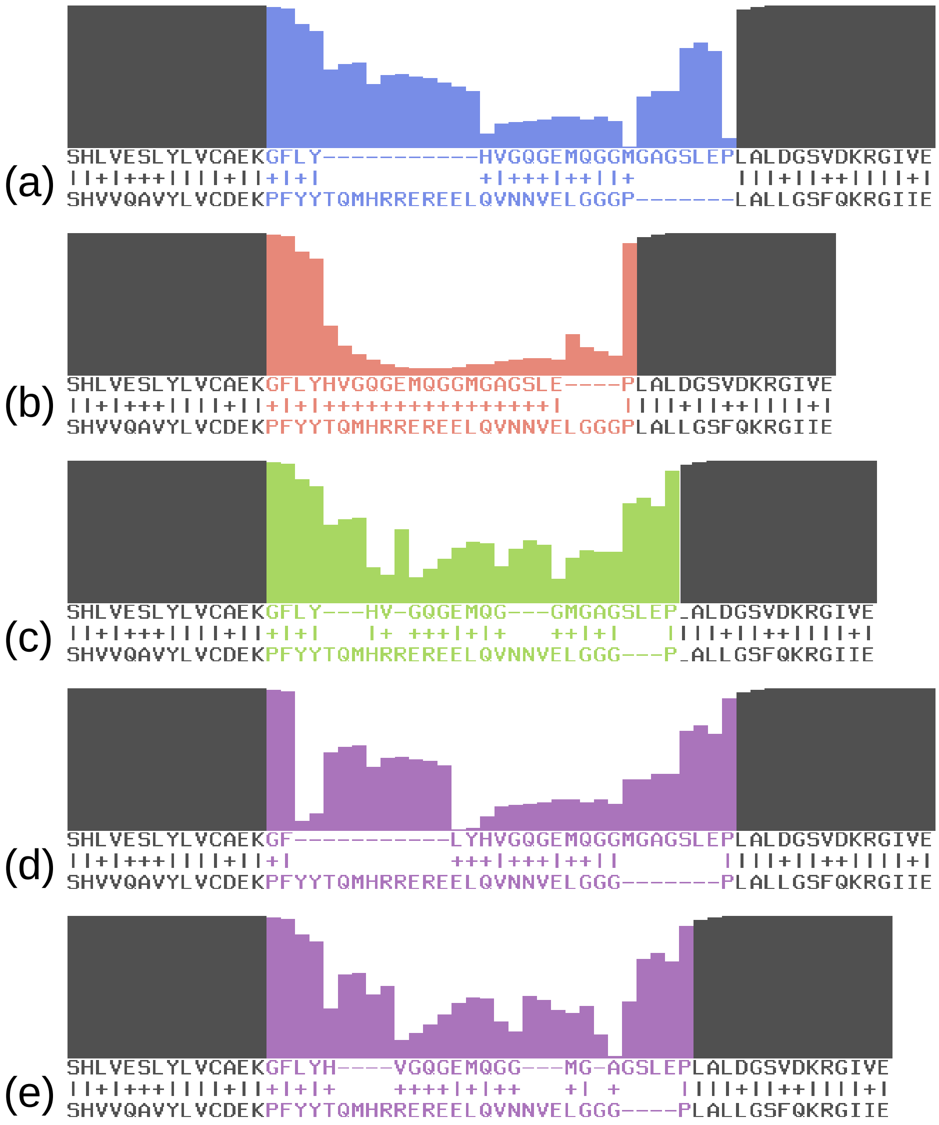

Figure 3: Part of the alignment of a pair of sequences generated with ROSE (see text for details). The posterior probabilities are based on BLOSUM62 with gap open penalty of $d = 11$ and extension penalty of $e = 1$. (a) The correct alignment according to the evolutionary tree simulated with ROSE (b) Optimal alignment (c) Maximum posterior alignment. (d)-(e) sampled alignments (2 out of 10 samples).

posterior alignment the actual size of the ambiguous segment in Figure 3 is achieved within the first 10 samples. When we draw 100 samples the critical segment is increased by four positions and for 1000 samples by three further columns. In other words, ambiguous segments only grow very slowly with the number of samples and in our experience about 10 samples are sufficient to obtain meaningful results.

Thus, in principle, the posterior probabilities can be used to identify the regions of low and high confidence. To assess this quantitatively, we simulated 1000 independent evolutions sequences for values of $D = 10, 20, 40$. Then we ran `ppALIGN` each time for all $7 \times 8/2$ pairs of leave sequences and compared the results to the true alignments. In the inset of Figure 4, we show the average fraction of correctly aligned positions as a function of the (binned) posterior probabilities $P \in \{P^{\mathtt{P}}_{i,j}, P^{\mathtt{A}}_{i}, P^{\mathtt{B}}_{j}\}$, the different probabilities not being distinguished here. The relationship is nearly linear. This shows that the posterior probabilities computed by `ppALIGN` (without knowing the evolutionary true alignment) are well correlated with the probability that the optimum alignment is correct.

On average, the optimum alignment was correct for 95.8% of all alignment positions for $D = 10$ (86.5% for $D = 20$ and 54% for $D = 40$). In the non-ambiguous regions, which are considered reliable, the alignment for $D = 10$ was on average correct for 99.8% of all positions (99.1% for $D = 20$ and 95% for $D = 40$). In the ambiguous regions, which are considered unreliable, on average 84.2% were correct (70.8% for $D = 20$ and 45% for $D = 40$).

Furthermore, we performed a Receiver Operating Characteristic (ROC) curve: If one accepts all alignment positions where the posterior probability P is larger than some threshold p_{thres} how large will be the *true positive rate* (the fraction of correct alignment positions, where $p > p_{\text{thres}}$), and the *false negative rate* (the fraction of incorrect alignment positions where $p > p_{\text{thres}}$). Clearly, for $p_{\text{thres}} \to 0$ both rates converge to 1, while for $p_{\text{thres}} \to 1$, both rates will approach zero. The behavior for intermediate values, for different values of D, is shown in the main plot of Figure 4. The curves run close to $(0, 1)$ if D is not too large, which means that using `ppALIGN` and this simple threshold-based criterion, correctly aligned regions can be identified in a reliable fashion. It must be noted that for closely related sequences (a small D) the alignment problem is easy and the distance between the curves and the point $(0, 1)$ in the ROC space becomes smaller. In this case, the optimal threshold value is also larger.

This experiment shows that the choice of p_{thres} is critical and strongly

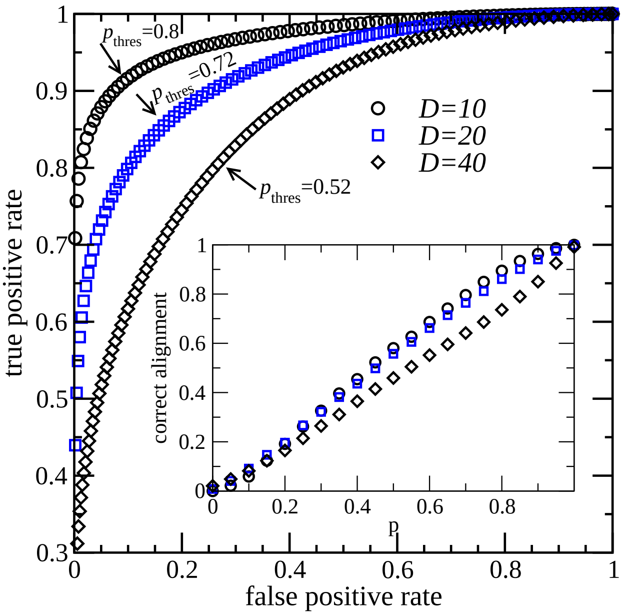

Figure 4: Main figure: ROC for sets of sequences obtained from the ROSE simulations (see text). Alignment positions of the optimum alignments where the posterior probability P calculated by ppALIGN is larger than a threshold p_{thres} are considered as being correct. In comparison with the evolutionary true alignments, given by ROSE, we infer the true positive rate as a function of false positive rate, while varying p_{thres} for three different evolutionary distances D. Sample threshold values leading to data points close to the upper left corner $(0, 1)$ are indicated together with little arrows. Inset: fraction of the correct alignment positions as a function of the binned posterior probabilities calculated by ppALIGN for these positions.

depends on the evolutionary distance between the considered sequences. As a rule-of-thumb, we suggest using a conservative threshold (ex: $p_{\text{thres}} = 0.9$) for closely related sequences, and a more relaxed threshold (ex: $p_{\text{thres}} = 0.5$) for more distant sequences.

Note that due to the layout of the evolutionary tree we have used, the PAM distances between the leaves varies from $2D$ to $6D$, the average distance being $37D/7$. We have verified (not shown here) that the results remain in principle the same, when we use simpler trees, where all pairs of sequences have the same distance ($D = 30, 60, 120$ in this case).

5.2 Simulation: local alignment

Next, we turn to the question of the uncertainty of the correct starting and ending of local alignments. From Jaroszewski et. al. [Jaroszewski, Li, and Godzik (2002)] we know that the optimal local sequence alignment can strongly differ from the structural alignment (for example obtained with the combinatorial expansion method [Shindyalov and Bourne (1998)]). For example, when we compare dihydrodipicolinate reductase (DIH) from *Escherichia coli* with malate dehydrogenase (BDM) from *Thermus thermophilus* (PDB: 1DIH:A and 1BDM:A), the optimal sequence alignment (with the standard set of parameters as above) starts at $i_1 = 4$ and $j_1 = 3$ and ends at $i_2 = 23$ and $j_2 = 22$. In contrast, the structural alignment ranges from $i_1 = 4$ and $j_1 = 3$ to $i_2 = 156$ and $j_2 = 210$. If we consider the structural alignment as the golden standard, we can infer that the optimal sequence alignment produces the correct starting point, but largely failes to find the correct ending of the alignment.

To illustrate how `ppALIGN` may detect such inconsistencies we consider the starting and ending probabilities for local alignments, $P_{i,j}^{\text{start}}$ and $P_{i,j}^{\text{end}}$, computed by our software. These two-dimensional distributions close to the points of the optimum are shown in Figures 5 (c) and (f). As expected, the maximum of the starting point in Figure 5 (c) is much sharper than the one for the ending point of the alignment in Figure 5(f).

However, such two-dimensional plots are interesting when we want to find the correct pair of starting or ending positions but they are harder to interpret than an one-dimensional representations. Therefore `ppALIGN` can additionally provide marginalized representations displaying the probabilities that the alignment starts / ends at certain positions i or j in the input sequences. For example, the probability that an alignment starts at position i in the first sequence a_1^ℓ is given by $P_i^{\text{start},\mathtt{A}} = \sum_j P_{i,j}^{\text{start}}$. To illustrate the reliability of the correct starting / ending position one can determine a confidence interval in the sequences around the position of the optimum with more than $x\%$ probability. This is simply done from the starting position with the highest possible posterior probability by extending to the adjacent position (left or right) and systematically adding the highest posterior probability. The resulting confidence interval is hence not necessary symmetrical.

This is illustrated in Figures 5 (a) and (b) for the starting point and in Figures 5 (d) and (e) for the ending point. We learn that the 90% confidence intervals for the starting point are even smaller than the 50% intervals for

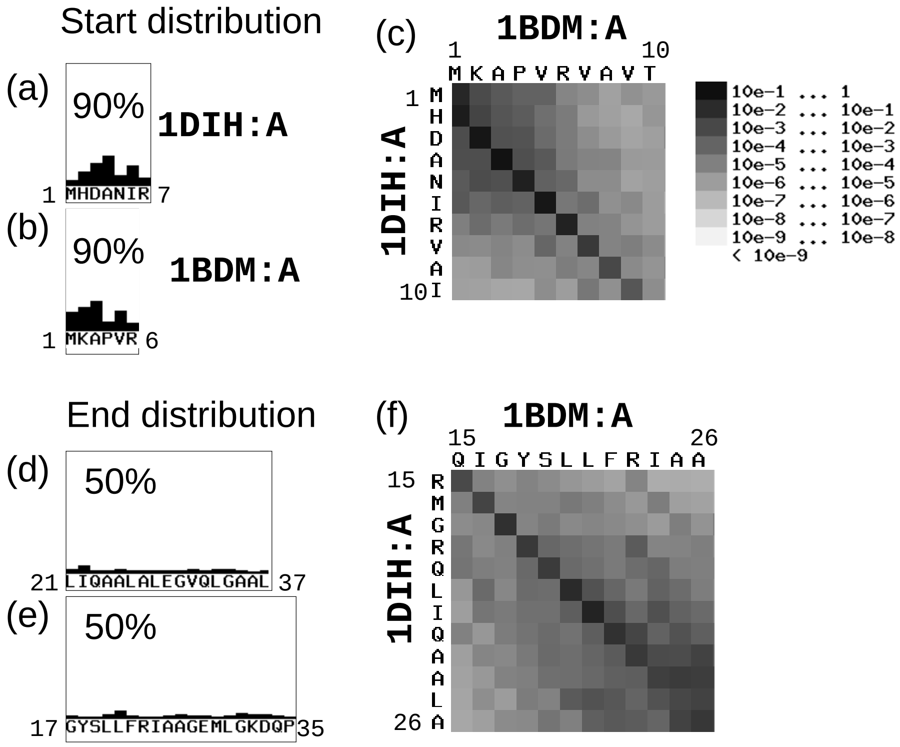

Figure 5: The posterior probabilities for starts (a)-(c) and ends (d)-(f) of local alignments. (a) and (b) show the 90% confidence interval for a start in the query protein 1DIH and the subject protein 1BDM. These are the subsequences around the optimal start point, whose posterior start/end probabilities sum up to at least 90%. The two-dimensional start and end distributions are shown in (c) and (f). (d) and (e) are the 50% confidence intervals for ends of local alignments.

the ending point. The 90% intervals which are not shown here range from $i = 1$ to $i = 122$ and from $j = 1$ to $j = 121$. This can be interpreted as strong evidence that the end of the alignment is incorrectly predicted. This observation which has been made on the sequence alignment alone is consistent with the structural alignment.

5.3 Application: genes for aromate sensing

Finally, we use ppALIGN to find regions in predicted sensor proteins which are suspected to be responsible for sensing aromatic compounds. For that purpose, we study the denitrifying betaproteobacterium *Aromatoleum aromaticum* EbN1 which has the unique catabolic feature of degrading toluene and ethylbenzene under anoxic conditions [Rabus and Widdel (1995)]. Despite the chemical similarity of these alkylbenzenes, strain EbN1 uses two fundamentally different reaction sequences for their conversion to the common intermediate benzoyl-CoA. For example, while toluene is activated by radical addition to fumarate yielding benzylsuccinate as the first intermediate, ethylbenzene is hydroxylated (H_2O-dependent) at the methylene carbon forming (*S*)-1-phenylethanol [Rabus and Heider (1998)]. Coding genes for degradation enzymes and substrate-sensing two-component regulatory systems are subsequently identified via proteomic-directed whole-genome-sequencing of strain EbN1, see Figure 6. The determined ethylbenzene-related gene cluster contains two operon-like structures proposed to be sequentially regulated by the ethylbenzene-responsive Tcs2/Tcr2 and the acetophenone-responsive Tcs1/Tcr1 systems [Rabus, Kube, Beck, Widdel, and Reinhardt (2002)]. The toluene-related gene cluster is also composed of two operon-like structures, which are however suggested to be coordinately regulated by the toluene-responsive TdiSR system [Kube, Heider, Hufnagel, Kühner, Beck, Reinhardt, and Rabus (2004)]. A related two-component regulatory system has previously been proved to control gene expression of aerobic toluene degradation in *Pseudomonas putida* [Lau, Wang, Patel, Labbé, Bergeron, Brousseau, Konishi, and Rawlings (1997)]. The sensor domains of the sensory systems contain so-called PAS-domains, generally assumed to sense environmental stimuli [Taylor and Zhulin (1999)]. Notably, each of the two proposed alkylbenzene-responsive sensors (Tcs2 and TdiS) contains two PAS-domains displaying different degrees of similarity between the two sensors: 42% identity for the first and 16% identity for the second PAS domain. These identity differences could be key to the sensory distinction between structurally similar ethylbenzene and toluene [Kube et al. (2004)].

To test whether a bioinformatic approach can help in seeking candidates regions, which are responsible for this sensory distinction, we perform a ClustalW [Higgins and Sharp (1988), Larkin et al. (2007)] alignment of the genes which code the corresponding substrate-sensing system. The resulting alignment is contained in the supplementary material in the FASTA format.

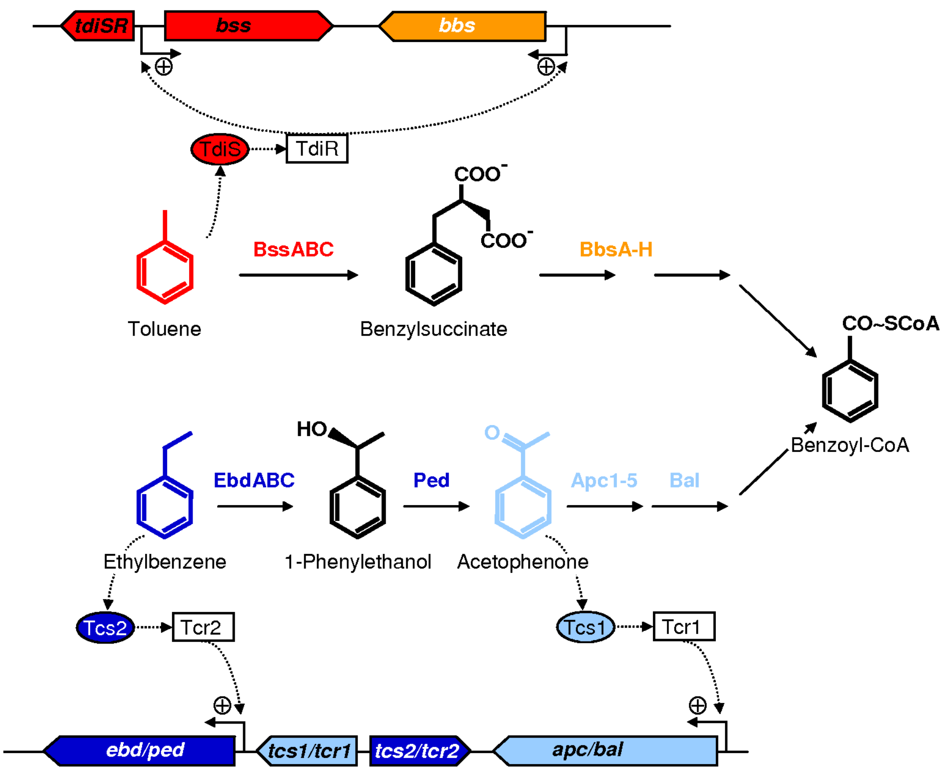

Figure 6: Sensory discrimination between toluene and ethylbenzene, and regulatory circuits for alkylbenzene-specific gene expression in *Aromatoleum aromaticum* EbN1. Designations of two-component regulatory systems: TdiSR, toluene-responsive; Tcs2/Tcr2, ethylbenzene-responsive; Tcs1/Tcr1, acetophenone-responsive. Gene designations: bssABC, benzylsuccinate synthase; bbsA-H, β-oxidation of benzylsuccinate to benzoyl-CoA; ebdABC, ethylbenzene dehydrogenase; ped, 1-phenylethanol dehydrogenase; apc1-5, acetophenone carboxylase; bal, benzoylacetate CoA-ligase. Modified from [Kühner et al. (2005)].

From the alignment alone, which contains only a few very similar regions, it would be very difficult to infer the regions of interest. We calculated the posterior probabilities using ppALIGN. The result is shown in Figure 7. Some regions of the alignment exhibit a particular low posterior probability, indicating a high variability of high-scoring alignment. Since the two different organisms sense different alkylbenzenes, it is very likely that the genes exhibit a strong variability at the positions where the sensing of different alkylbenzes is coded.

The second PAS domain is located precisely in one of the lowest confidence region which is consistent with its low conservation (identity 16%) among the different species. The first PAS domain is not located in a low confidence region but close to the second lowest confidence one which is consistent with its higher conservation level (identity 42%). In both cases, the low confidence regions of the alignment suggest a high variability among the species and could therefore prove to be a useful targeted region for further biological investigations such as knock out experiments.

This shows again that regions which displays low posterior probabilities are good candidates when seeking for organism-specific subsequences. This makes it possible to reduce the effort invested while using biochemical methods to locate such regions, since the search space is greatly reduced.

6 Conclusion and future prospects

The package `ppALIGN` (including stand-alone command-line programs and a C++ library) provides efficient algorithms that compute the posterior probabilities for score-based alignment. The stand-alone program `ppALIGN` allows the user to provide a single alignment and the set of parameters, while `ppBLAST` directly uses the structured output of BLAST (XML-format). Both programs compute the posterior probabilities for each alignment, and siplay the results either in a structured XML format, a plain text, or a more visual HTML page. The flexible library can be extended to new decoding algorithms and other ways of marginalization of the posterior distribution.

From a mathematical point of view, we provide several advances with this paper: 1) a new pair HMM parametrization which is consistent with arbitrary score-based parameters; 2) a comparison of this pair HMM to close models such as the FTA model [Miyazawa (1995), Zhang and Marr (1995), Kschischo and Lässig (2000)] and the model of Yu and Hwa [Yu and Hwa (2001)]; 3) detailed formulas and algorithms to compute classical posterior distributions ($P^{\mathtt{P}}_{i,j}$, $P^{\mathtt{A}}_{i}$, and $P^{\mathtt{B}}_{j}$), as well as more original ones ($P^{\text{start}}_{i,j}$ and $P^{\text{end}}_{i,j}$).

Using two practical examples, analyzing sequences generated with the

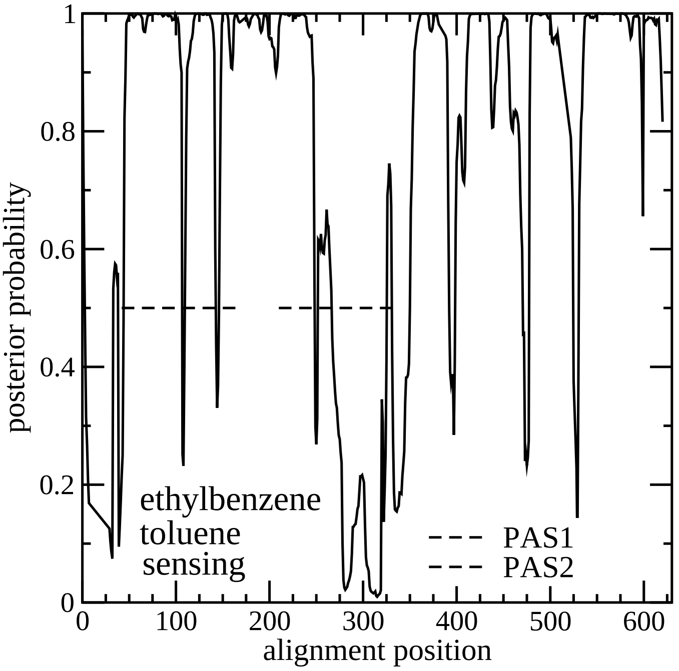

Figure 7: Posterior probabilities for the ClustalW alignment of EbN1:Tcs2 (ethylbenzene-responsive) and EbN1:Tdis (toluene-responsive). The horizontal lines PAS1/PAS2 indicate the regions where the sensing of the respective alkylbenze is most likely coded.

ROSE package and analyzing the ClustalW alignment of two sequences of the denitrifying betaproteobacterium *Aromatoleum aromaticum* EbN1, we have shown that regions of interest can definitely be located by our approach.

We are currently working on a module that performs a more flexible marginalization on the basis of user supplied pattern in the spirit of the work of Aston and Martin [Aston and Martin (2007)]. We are also interested in several natural extensions of the method in order to deal with profile related alignments (profile-sequence, profile-profile), multiple alignments, or position specific scoring functions.

Appendix

In the first supplementary document we describe in detail the mathematical background of ppALIGN. We describe the pair HMM, the finite-temperature approach and their connection to score based alignments.

Furthermore a ClustalW output (FASTA format) for the alignment of two sequences of the denitrifying bacterium, strain EbN1 is given. The alignment can directly be pasted into the ppALIGN web page to calculate the posterior probabilities, as shown in Fig. 7.

A Theoretical background of ppALIGN

In this appendix we go into the mathematical details of the ppALIGN software. We first explain the pair-HMM we are using in Section A.1 and the corresponding algorithms in Section A.2. In Section A.4 we compare the classical score based alignment with the pair HMM.

A.1 Pair hidden Markov Model

Score-based methods determine the optimal global alignment $\hat{\pi}$ by maximizing an objective function s, $\hat{\pi} = \operatorname{argmax}_{\pi} s(\pi; a_1^\ell, b_1^m)$. The local version Local [Smith and Waterman (1981)] being $\hat{\pi} = \operatorname{argmax}_{i_1 \leq i_2, j_1 \leq j_2} \; s(\pi; a_{i_1}^{i_2}, b_{j_1}^{j_2})$.

We define global and local alignment on sequences over finite alphabets as follows:

Definition 1. Let $a_1^\ell = a_1 \dots a_\ell \in \Sigma^\ell$ and $b_1^m = b_1 \dots b_m \in \Sigma^m$ denote a pair of sequences.

(i) A *global alignment* π of a_1^l and b_1^m is a sequence of edit operations $\pi_1^t = \pi_1 \dots \pi_t$ with $\pi_k \in \{\mathtt{P}, \mathtt{A}, \mathtt{B}\}$ $(k = 1, 2 \dots t)$ such that, with $F(\mathtt{P}) = (1, 1), F(\mathtt{A}) = (1, 0), F(\mathtt{B}) = (0, 1)$,

$$\sum_{k=1}^{t} F(\pi_k) = (\ell, m). \tag{3}$$

where the operation P is referred to as pair, the operation A as insertion in sequence a_1^ℓ and the operation B as insertion[5] in sequence b_1^m.

[5] or conversely, A as deletion in sequence b_1^m, and B as deletion in sequence a_1^ℓ, for this reason we call such events *indel*.

(ii) Let $1 \le i_1 \le i_2 \le \ell$ and $1 \le j_1 \le j_2 \le m$. A *local alignment* $(\pi_1^t, i_1, i_2, j_1, j_2)$ of a_1^ℓ and b_1^m is a global alignment of the pair of subsequences $a_{i_1}^{i_2}$ and $b_{j_1}^{j_2}$ with $\pi_1 = \mathtt{P}$ and $\pi_t = \mathtt{P}$.

We shall use the notation $i(t) = (1,0) \cdot \sum_{k=1}^t F(\pi_k)$ and $j(t) = (0,1) \cdot \sum_{k=1}^t F(\pi_k)$, where "$\cdot$" denotes the inner product $(r,s)\cdot(t,u) = rt + su$. These quantities give the position in the sequences after t positions in the alignment.

Examples. Let $a_1^7 = \mathtt{GGTACCG}$, $b_1^6 = \mathtt{GCCTGG}$ and consider the global alignment $\pi_1^8 = \mathtt{PPAAPBPP}$ of a_1^7 and b_1^6. It is represented as

$$\begin{array}{cccccccc}
\mathtt{G} & \mathtt{G} & \mathtt{T} & \mathtt{A} & \mathtt{C} & - & \mathtt{C} & \mathtt{G} \\
\mathtt{G} & \mathtt{C} & - & - & \mathtt{C} & \mathtt{T} & \mathtt{G} & \mathtt{G} \\
\mathtt{1} & \mathtt{2} & \mathtt{3} & \mathtt{4} & \mathtt{5} & \mathtt{6} & \mathtt{7} & \mathtt{8}
\end{array}$$

The consistency relation (3) ensures that $(i(8), j(8)) = (1,1)+(1,1)+(1,0)+(1,0)+(1,1)+(0,1)+(1,1)+(1,1) = (7,6)$. After, for instance, 6 positions in the alignment, we have seen the subsequences $a_1^{i(6)} = a_1^5 = \mathtt{GGTAC}$ and $b_1^{j(6)} = b_1^4 = \mathtt{GCCT}$.

On the same sequences, the local alignment $(\pi_1^4 = \mathtt{PAPP}, i_1 = 2, i_2 = 5, j_1 = 1, j_2 = 3)$ is represented as

$$\begin{array}{cccccc}
\mathtt{2} & \mathtt{G} & \mathtt{T} & \mathtt{A} & \mathtt{C} & \mathtt{5} \\
\mathtt{1} & \mathtt{G} & - & \mathtt{C} & \mathtt{C} & \mathtt{3} \\
 & \mathtt{1} & \mathtt{2} & \mathtt{3} & \mathtt{4} &
\end{array}$$

For local alignment, the consistency relation (3) must be fulfilled on the aligned part, i.e. $\sum_{k=1}^t F(\pi_k) = (i_2 - i_1 + 1, j_2 - j_1 + 1) = (1,1) + (1,0) + (1,1) + (1,1) = (4,3)$.

Model layout. In general, a HMM consists of a Markovian process describing a sequences of *unobserved states* and a sequence of *outcomes* conditioned on the states (Rabiner, 1989). When modeling alignments as a pair-HMM, the sequences a_1^ℓ and b_1^m refer to the outcomes and the alignment π_1^t to the sequence of unobserved states. Both objects are considered to be random. In the posterior analysis of alignments we are particularly interested in events of the unobserved states conditioned by the outcomes, i.e. events of the type $\Pi_1^t = \pi_1^t | A_1^\ell = a_1^\ell, B_1^m = b_1^m$.

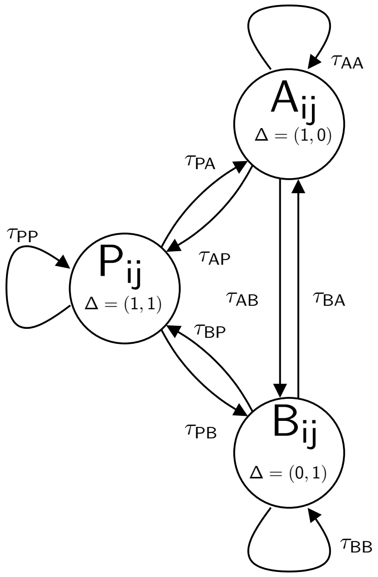

Figure 8: Setup of the pair HMM for global alignment. A pair of correlated sequences and an alignment is generated by a Markov process on the graph. For each transition the counter of the alignment t is advanced by 1, whereas the sequence position counters (i, j) are advanced by Δ.

In the classical methodology (Durbin et al., 1998) the lengths of the sequences are modeled by a geometric distribution. This requires an additional parameter and therefore it is not easy to uniquely determine the HMM parameters from a given classical scoring function. It is however possible to formulate a generative process such that the alignment length t is fixed and the sequence lengths ℓ and m remain random but restricted according to the condition in Equation (3). However, posterior probabilities are conditioned to fixed sequences a_1^ℓ and b_1^m. Therefore we need to fix the lengths ℓ and m rather than t. Yu and Hwa (2001) solved this problem by a probabilistic model which gives alignments normalized weights constrained by a probability conservation condition. This leads to an approximative mapping from a classical scoring function to the probabilistic model. However, the interpretation of the model parameters as HMM transition and emission probabilities is quite difficult in this framework. For this reason we implemented a slightly different model which is described below. We have numerically confirmed that the results for both models are compatible and differences are only marginal (not shown here).

A random alignment Π_1^t of the length t is described by a Markov process with the transition matrix

$$\tau_{x,y} \doteq \mathbb{P}\left(\Pi_{t+1} = y | \Pi_t = x\right) \tag{4}$$

where $\sum_y \tau_{x,y} = 1$ for all $x \in \{\mathtt{P}, \mathtt{A}, \mathtt{B}\}$ and the starting distribution

$$\mu(x) = \mathbb{P}(\Pi_1 = x)$$

with $\sum_x \mu(x) = 1$. In order to reduce the number of free parameters we use $\mu(x) = \tau_{\mathtt{P}x}$ for $x \in \{\mathtt{P}, \mathtt{A}, \mathtt{B}\}$. The *emission probabilities* which connect the outcomes to the hidden variables are defined as conditional probabilities

$$\begin{aligned} p(a,b) &\doteq \mathbb{P}\left(A_{i(t)} = a, B_{j(t)} = b \mid \Pi_t = \mathtt{P}\right) && (5)\\ q(a) &\doteq \mathbb{P}\left(A_{i(t)} = a \,|\Pi_t = \mathtt{A}\right) \\ &= \mathbb{P}\left(B_{j(t)} = a \,|\Pi_t = \mathtt{B}\right), && (6) \end{aligned}$$

where

- $p : \Sigma \times \Sigma \to \mathbb{R}$ denotes the probability of building pairs with the normalization condition $\sum_{a,b\in\Sigma} p(a,b) = 1$, and
- $q : \Sigma \to \mathbb{R}$ is referred as background frequencies, $\sum_{a\in\Sigma} q(a) = 1$.

To describe the algorithms in the next section in a way which is closely related to HMMs of a single sequence we unify the emission probabilities in Equations (5) and (6) as

$$p_x(a,b) \doteq \begin{cases} p(a,b) & \text{if } x = \mathtt{P} \\ q(a) & \text{if } x = \mathtt{A} \\ q(b) & \text{if } x = \mathtt{B}. \end{cases} \tag{7}$$

Note that $p_{\mathtt{A}}(a,b)$ does not depend on b, and, likewise, $p_{\mathtt{B}}(a,b)$ is independent of a.

Having specified the parameter set $\Theta = \{\mu, \tau, p, q\}$, the generative process of the pair-HMM is described by the following algorithm (see Figure 8).

1. Choose x with probability $\mu(x)$ and set $(i,j) \leftarrow F(x)$, $\pi_1 \leftarrow x$, and, $t \leftarrow 1$. If $x = \mathtt{P}$, set $a_1 = a$ and $b_1 = b$ with the probability $p(a,b)$. If $x = \mathtt{A}$, set $a_1 = a$ with the probability $q(a)$. If $x = \mathtt{B}$, set $b_1 = b$ with the probability $q(b)$.

2. Repeat the following step until $i(t) = \ell$ or $j(t) = m$:

    - Advance the time step $t \leftarrow t + 1$. Set $\pi_t \leftarrow y$ chosen with the probability τ_{xy}.
    - Set $(i, j) \leftarrow (i, j) + F(\pi_t)$
    - If $\pi_t = \mathtt{P}$ set $(a_i, b_j) \leftarrow (a, b) \in \Sigma^2$ chosen with the probability $p(a, b)$.
    - If $\pi_t = \mathtt{A}$ set $a_i \leftarrow a \in \Sigma$ chosen with the probability $q(a)$.
    - If $\pi_t = \mathtt{B}$ set $b_j \leftarrow b \in \Sigma$ chosen with the probability $q(b)$.
    - Set $x \leftarrow y$.

3. If $i(t) = \ell$ *and* $j(t) = m$, return the tuple $(a_1^\ell, b_1^m, \pi_1^t)$. Otherwise reset a_1^ℓ, b_1^m, and, π_1^t and start again with step (1).

Note that in Step 1 of the algorithm, we systematically have $\pi_1 \leftarrow \mathtt{P}$ with a local alignment.

In the following we shall denote the probabilities of events without the restart condition (3) as $\mathbb{P}_{\text{free}}$. Without the subscript "free" we implicitly refer to the model with the restart condition.

A.2 Forward and backward recursions

The probability that the model without the restart condition (3) should generate after t time steps the tuple $a_1^{i(t)}$, $b_1^{j(t)}$ and p_1^t is given by

$$\begin{aligned} & \mathbb{P}_{\text{free}}\left(A_1^{i(t)} = a_1^{i(t)}, B_1^{j(t)} = b_1^{j(t)}, \Pi_1^t = \pi_1^t\right) && (8) \\ = \; & \mu(\pi_1)\, p_{\pi_1}\left(a_{i(1)}, b_{j(1)}\right) \prod_{k=2}^{t} \tau_{\pi_{k-1},\pi_k}\, p_{\pi_k}\left(a_{i(k)}, b_{j(k)}\right), && (9) \end{aligned}$$

where we have used the compact notation defined in Equation (7). The (joint) probability that the pair HMM including the restart condition generates the tuple a_1^ℓ, b_1^m, π_1^t can be written as the following fraction

$$\begin{aligned} & \mathbb{P}_{\text{global}}\left(A_1^\ell = a_1^\ell, B_1^m = b_1^m, \Pi_1^t = \pi_1^t\right) && (10) \\ = \; & \frac{\mathbb{P}_{\text{free}}\left(A_1^\ell = a_1^\ell, B_1^m = b_1^m, \Pi_1^t = \pi_1^t\right)}{Z_{(\ell,m)}} \\ = \; & \frac{\mu(\pi_1)\, p_{\pi_1}\left(a_{i(1)}, b_{j(1)}\right) \prod_{k=2}^{t} \tau_{\pi_{k-1},\pi_k}\, p_{\pi_k}\left(a_{i(k)}, b_{j(k)}\right)}{Z_{(\ell,m)}}. && (11) \end{aligned}$$

The normalization factor

$$Z_{(\ell,m)} \doteq \mathbb{P}_{\text{free}}\left(\exists t : i(t) = \ell, j(t) = m\right) \tag{12}$$

in Equation (10) can be interpreted as the probability that the generative process ends at $i(t) = \ell$ and $j(t) = m$ in the first trial. This is a consistency condition: whatever the length t of the alignment, it uses exactly ℓ letters from sequence A and m letters from sequence B. This ensures that

$$\sum_{a_1^\ell}\sum_{b_1^m}\sum_{\pi_1^t : i(t)=\ell, j(t)=m} \mathbb{P}_{\text{global}}\left(A_1^\ell = a_1^\ell, B_1^m = b_1^m, \Pi_1^t = \pi_1^t\right) = 1.$$

In the framework of local alignment, parts of the sequences which are not aligned are generated independently from each other according to the background model of i.i.d. sequence described by the probabilities q. The joint probability that $(a_1^\ell, b_1^m, \pi, i_1, i_2, j_1, j_2)$ is generated by the local version of the pair HMM is given by

$$\begin{aligned}
&\mathbb{P}_{\text{local}}\left(A_1^\ell = a_1^\ell, B_1^m = b_1^m,\ \Pi_1^t = \pi_1^t\right) \\
= &\prod_{i=1}^{i_1-1} q(a_i) \prod_{j=1}^{j_1-1} q(b_j) \times \mathbb{P}_{\text{global}}^{\text{bound}}\left(A_{i_1}^{i_2} = a_{i_1}^{i_2}, B_{j_1}^{j_2} = b_{j_2}^{j_2},\ \Pi_1^t = \pi_1^t\right) \times \\
&\prod_{i=i_2+1}^{l} q(a_i) \prod_{j=j_2+1}^{m} q(b_j),
\end{aligned}$$

From now on, $\delta^{\text{local}} = 1$ will be used in formulas related to local alignment, and $\delta^{\text{local}} = 0$ in formulas related to global alignment.

A.3 Computation of the marginal probabilities

In the following we describe the methods to compute the marginal probabilities $P_{i,j}^{\mathsf{P}}$, P_i^{A}, P_j^{B}, and, $P_{i,j}^{\text{start/end}}$ defined in the main text. In Theorem 2 we introduce the related forward and backward quantities and the marginalization is shown in Theorem 4. Since general results have been known for some time (Baum, Petrie, Soules, and Weiss, 1970, Rabiner, 1989, Durbin et al., 1998), we skip the corresponding proofs.

Theorem 2 (Baum et al. (1970)). Let $C_{(i_1,j_1)}^{(i_2,j_2)}$ denote the event $A_{i_1}^{i_2} = a_{i_1}^{i_2}, B_{j_1}^{j_2} = b_{j_1}^{j_2}$. Furthermore, let $\mathbb{P}_{\text{free}}(A)$ denote the probability of the event

A under the pair-HMM without restart (without step (4) in the generative process) and $\mathbb{P}(A)$ denote the probability of A according to the model with restart.

(i) The *forward probabilities* defined as

$$\phi_y(i,j) \doteq \mathbb{P}\left(C_{(1,1)}^{(i,j)},\ \exists k : i(k) = i,\ j(k) = j,\ \Pi_k = y\right) \tag{13}$$

can be computed by the recurrence relation

$$\begin{aligned}\phi_y(i,j) \quad = \quad & p_y(a_i,b_j) \sum_x \tau_{xy} \cdot \phi_x\left((i,j) - F(y)\right) + \\ & p_y(a_i,b_j)\ \delta^{\text{local}}\ \mathbb{I}_{y=\mathtt{P}}\ q(a_1^{i-1})\ q(b_1^{j-1}) \end{aligned} \tag{14}$$

(ii) The *backward probabilities* defined as

$$\beta_x(i,j) \doteq \mathbb{P}\left(C_{(i+1,j+1)}^{(l,m)}\ \mid \exists k : i(k) = i,\ j(k) = j,\ \Pi_k = x\right) \tag{15}$$

can be computed by the recurrence relation

$$\beta_x(i,j) \quad = \quad \sum_y \tau_{xy} \cdot p_y(a_{i+1},b_{j+1}) \cdot \beta_y\left((i,j) + F(y)\right) + \tag{16}$$

$$\delta^{\text{local}}\ \mathbb{I}_{x=\mathtt{P}}\ q(a_{i+1}^{\ell})\ q(b_{j+1}^{m}) \tag{17}$$

Decomposing the events $C_{(i_1,j_1)}^{(i_2,j_2)}$ and applying Bayes' theorem yields the following lemma.

Lemma 3. In the case of global alignment and for a given pair of sequences a_1^ℓ and b_1^m we have

(i)

$$\mathbb{P}\left(C_{(1,1)}^{(\ell,m)},\ \exists k : (i(k),j(k)) = (i,j), \Pi_k = x\right) = \frac{\phi_x(i,j)\beta_x(i,j)}{Z_{(\ell,m)}} \tag{18}$$

(ii)

$$\begin{aligned} & \mathbb{P}\left(C_{(1,1)}^{(\ell,m)},\ \exists k : (i(k),j(k)) = (i,j), \Pi_{k+1} = y, \Pi_k = x\right) \\ = \quad & \frac{\phi_x(i,j)\tau_{xy}\beta_y\left((i,j)+F(y)\right)p_y(a_{i+1},b_{j+1})}{Z_{(\ell,m)}}, \end{aligned} \tag{19}$$

where $Z_{(\ell,m)}$ is defined in Equation (12).

Note that we have $Z_{(\ell,m)} = \sum_x \beta_x(0,0) = \sum_x \phi_x(\ell,m)$.

Theorem 4 (Marginal probabilities)**.** For global alignment the following marginal probabilities are given in terms of ϕ and β.

$$\begin{aligned}
P_x(i,j) &\doteq \mathbb{P}\left(\exists k : (i(k), j(k)) = (i,j),\ \Pi_k = x \;\middle|\; C^{(\ell,m)}_{(1,1)}\right) \\
&= \frac{\phi_x(i,j)\ \beta_x(i,j)}{\sum_y \phi_y(\ell,m)} \qquad (20) \\
T_{x,y}(i,j) &\doteq \mathbb{P}\left(\exists k : (i(k), j(k) = (i,j))\,,\ \Pi_{k+1} = y \;\middle|\; \Pi_k = x,\ C^{(l,m)}_{(1,1)}\right) \\
&= p_y(a_{i+1}, b_{j+1})\ \tau_{x,y}\ \frac{\beta_y((i,j) + F(y))}{\beta_x(i,j)} \qquad (21)
\end{aligned}$$

Proof. If we note that $\mathbb{P}(C^{\ell,m}_{(1,1)}) = \sum_y \phi_y(\ell,m)\ \delta^{\mathrm{local}} = 0$, Equations (20) and (21) follow directly Equations (18) and (19). □

Theorem 4 provides the theoretical basis for `ppALIGN`. Eq. 20 is used to compute the posterior probability for paired letters and gaps for a given alignment. The probability that the positions i in a_1^ℓ and j in b_1^m are paired is given by $P_{\mathtt{P}}(i,j)$. Since we are not interested in which particular position j is in the second sequence a is gapped with letter at position i in the first sequence appears, we define the gap probability by marginalization over all possible positions j,

$$P_i^{\mathtt{A}} = \sum_j P_{\mathtt{A}}(i,j) \ \text{ and } \ P_j^{\mathtt{B}} = \sum_i P_{\mathtt{B}}(i,j)$$

For the model of local alignment `ppALIGN` determines posterior probabilities for the start (i_1, j_1) and end (i_2, j_2) positions, $P_{i,j}^{\mathrm{start/end}}$. Those probabilities are determined from the forward and backward probabilities and the q describing the padding states before and after the alignment,

$$\begin{aligned}
P^{\mathrm{start}}_{i_1,j_1} &= \frac{1}{Z^{\mathrm{local}}_{(\ell,m)}} \prod_{i=1}^{i_1-1} q(a_i) \prod_{j=1}^{j_1-1} q(b_j)\ \beta_{\mathtt{P}}(i,j)\ p(a_i, b_j) \\
P^{\mathrm{end}}_{i_2,j_2} &= \frac{1}{Z^{\mathrm{local}}_{(\ell,m)}} \prod_{i=i_2+1}^{\ell} q(a_i) \prod_{j=j_2+1}^{m} q(b_j)\ \phi_{\mathtt{P}}(i,j),
\end{aligned}$$

with

$$Z^{\text{local}}_{(\ell,m)} \doteq \mathbb{P}\left(C^{(\ell,m)}_{(1,1)}\right) = \sum_{i_2,j_2} \phi_{\mathtt{P}}(i_2,j_2) \prod_{i=i_2+1}^{\ell} q(a_i) \prod_{j=j_2+1}^{m} q(b_j)$$

Eq. 21 only depends on a ratio between backward probabilities. The matrix $\{x,y : T_{x,y}(i,j)\}$ describes the transition probabilities of a heterogeneous Markov chain conditioned on the input sequence a_1^l and b_1^m. On this basis we can easily implement a forward sampling algorithm that samples alignments from the posterior distribution. The algorithm is similar to the generative process of the pair-HMM. We assume that we have obtained the transition matrix $T_{x,y}$ for every (i,j) and a corresponding starting probability $\tilde{\mu}(x) \doteq \mathbb{P}\left(\Pi_1 = x \middle| C^{(\ell,m)}_{(1,1)}\right)$ via the backward recursion Eq. 17 To sample alignments from the posterior distribution $\mathbb{P}\left(\Pi_1^t = \pi_1^t \middle| C^{(\ell,m)}_{(i,j)}\right)$, we proceed as follows:

1. Choose x with probability $\hat{\mu}(x)$ and set $(i,j) \leftarrow F(x)$, $\pi_1 \leftarrow x$, and, $t \leftarrow 1$.
2. Repeat the following step until $i = \ell$ and $j = m$:
    - Advance the time step $t \leftarrow t+1$. Set $\pi_t \leftarrow y$ chosen with the probability $T_{xy}(i,j)$.
    - Set $(i,j) \leftarrow (i,j) + F(y)$
    - Set $x \leftarrow y$.
3. Return π_1^t where π_1^t is an alignment of a_1^ℓ and b_1^m.

The resulting algorithm is obviously a dramatic improvement on the rejection algorithm presented in Section A.1.

A.4 The connection between score based alignment and pair-HMM

So far, we discussed the algorithms to compute posterior probabilities and to provide alternative alignments. One essential feature of `ppALIGN` is the fact that the user does not need to provide the full parameter set Θ of the pair-HMM. In this section we discuss the relationship between the score-based alignment and the pair-HMM.

Classical scoring functions usually involve rescaled matrices $\mathcal{S}(a,b) = \lambda \log \frac{p(a,b)}{q(a)q(b)}$ where $\lambda > 0$ defines the scale of the matrix. When λ is known `ppALIGN` determines the pair probabilities $p(a,b)$ from a classical scoring function with

$$p(a,b) = \exp(\mathcal{S}(a,b)/\lambda)\ q(a)\ q(b)$$

Fortunately protein score matrices, such as the `PAM` or `BLOSUM` family (Heinkoff and Heinkoff, 1992, Schwartz and Dayhoff, 1978) are published together with

- the pair probability matrices $p(a,b)$,
- the background frequencies $q(a)$, and,
- the scale λ.

Hence, we could easily include those values in the software.

For more simple scoring matrices $\mathcal{S}(a,b)$, where the background model is unknown, `ppALIGN` estimates the background frequencies from the input sequences for each alignment. Then the pair emission probability matrix is given by $p(a,b) = q(a)q(b)e^{\mathcal{S}(a,b)/\lambda}$ where λ is a scale factor that is determined numerically by the normalization condition $\sum_{a,b} q(a)q(b)e^{\mathcal{S}(a,b)/\lambda} = 1$.

Estimating the transition matrix τ_{xy} from classical gap costs requires some approximations which are explained next. Due to the normalization condition there are $3 \times 3 - 3 = 6$ free parameters. The aim is to reduce the number of free parameters to the gap open and gap extension penalties. We parametrize the set of 9 remaining transition probabilities (i.e. 6 free parameters) by

$$\begin{array}{lll} \tau_{\texttt{PP}} = 1 - 2\nu & \tau_{\texttt{PA}} = \nu_A & \tau_{\texttt{PB}} = 2\nu - \nu_A \\ \tau_{\texttt{AP}} = (1-2\nu)\eta_A & \tau_{\texttt{AA}} = \varepsilon & \tau_{\texttt{AB}} = \eta_A \psi_A \\ \tau_{\texttt{BP}} = (1-2\nu)\eta_B & \tau_{\texttt{BA}} = \eta_B \psi_B & \tau_{\texttt{BB}} = \varepsilon \end{array}$$

where $(1-2\nu)$ is the probability of entering the pair state `P`, η_A and η_B the probability of leaving the gap state. $\psi_A \doteq \frac{1-\varepsilon}{\eta_A} - (1-2\nu)$ and $\psi_B \doteq \frac{1-\varepsilon}{\eta_B} - (1-2\nu)$ are probabilities of entering the state `B` after having left the state `A` and vice versa. The gap extension probability ε is already chosen to be equal for both types of gaps.

The score of an alignment π_1^t of a_1^ℓ and b_1^m is defined as the log-likelihood ratio

$$v(\pi) = \log \frac{\mathbb{P}\left(C_{(1,1)}^{(\ell,m)},\ \exists k : (i(k), j(k)) = (\ell, m)\right)}{\prod_{i=1}^{\ell} q(a_i)\ \prod_{j=1}^{m} q(b_j)}.$$

The most likely alignment $\hat{\pi} = \operatorname{argmax}_\pi v(\pi)$ is referred to as the Viterbi alignment and the corresponding score $\hat{v} = \max_\pi v(\pi)$ as Viterbi score.

The Viterbi score $v(\pi_1^t)$ can be computed by the log-scale version of Equation (10),

$$\begin{aligned} v(\pi) &= \log \mu(\pi_1) + \sum_{k=2}^{|\pi|} \log \tau_{\pi_{k-1},\pi_k} + \sum_{k=1}^{|\pi|} \log p_{\pi_k}\left(a_{i(k)}, b_{j(k)}\right) - \log Z_{(\ell,m)} \\ &\quad - \sum_{i=1}^{\ell} \log q(a_i) - \sum_{j=1}^{m} \log q(b_j). \qquad (22) \end{aligned}$$

Due to the choice $\mu(x) = \tau_{\mathtt{P}x}$ as start distribution, Equation (22) can be rearranged using Equation (7) into:

$$\begin{aligned} v(\pi) &= \sum_{k:\pi_k=\mathtt{P}} \left[\log \frac{p(a_{i(k)}, b_{j(k)})}{q(a_{i(k)})q(b_{j(k)})} + \log(1-2\nu)\right] + \\ &\quad \sum_{k:\pi_k\neq\mathtt{P}} \tilde{s}_{\pi_{k-1}\pi_k} - \log \eta_{\pi_{|\pi|}} - \log Z_{(\ell,m)}, \qquad (23) \end{aligned}$$

with $\pi_0 = \mathtt{P}$, and, $\tilde{s}_{\mathtt{PP}} = \tilde{s}_{\mathtt{AP}} = \tilde{s}_{\mathtt{BP}} = 0$, $\tilde{s}_{\mathtt{PA}} = \log \nu_{\mathtt{A}}\eta_{\mathtt{A}}$, $\tilde{s}_{\mathtt{PB}} = \log(2\nu - \nu_{\mathtt{A}})\eta_{\mathtt{B}}$, $\tilde{s}_{\mathtt{AA}} = \tilde{s}_{\mathtt{BB}} = \log \epsilon$ $\tilde{s}_{\mathtt{AB}} = \log \psi_{\mathtt{A}}$ and $\tilde{s}_{\mathtt{BA}} = \log \psi_{\mathtt{B}}$. The term $-\log \eta_{\pi_{|\pi|}}$ in Equation (23) penalizes alignments which end in gap states by $-\log \eta_{\mathtt{A}}$ or $-\log \eta_{\mathtt{B}}$. The pair states P at the end of the alignment are not penalized ($\eta_{\mathtt{P}} = 1$). These contributions can be safely ignored for long alignments. We may then relate the scoring function derived from the pair HMM (23) with the classical scoring function of the Needleman-Wunsch global alignment

$$\begin{aligned} s(\pi) &= \sum_{\text{paired}(a,b)} \mathcal{S}(a_i, b_j) + \mathcal{S}(\text{gaps}) \\ &= \sum_{k:\pi_k=\mathtt{P}} \mathcal{S}(a_{i(k)}, b_{j(k)}) + \sum_{k:\pi_k\neq\mathtt{P}}^{t} \tilde{s}_{\pi_{k-1}\pi_k}, \end{aligned}$$

When $\nu \ll 1$ the additive term $\log(1-2\nu)$ in Equation (23) is negligible and we may identify the classical scoring function s with the corresponding scoring function of the HMM v as:

$$v(\pi) \approx s(\pi)/\lambda + \log Z_{(\ell,m)},$$

Event	pair HMM $v(\pi)$	score-based	parametrization
pair	$\log \frac{p(a,b)}{q(a)q(b)}$ $+(1-2\nu)$	$\mathcal{S}(a,b)$	$\mathcal{S}(a,b)/\lambda$
start a pair	$\tilde{s}_{\texttt{PP}}$, $\tilde{s}_{\texttt{AP}}$, $\tilde{s}_{\texttt{BP}}$	0	0
gap open	$\tilde{s}_{\texttt{PA}}$	$-\alpha = -d - e$	$\log \nu_{\texttt{A}} \eta_{\texttt{A}} = -\alpha/\lambda$
	$\tilde{s}_{\texttt{PB}}$	$-\alpha = -d - e$	$\log(2\nu - \nu_{\texttt{A}})\eta_{\texttt{B}} = -\alpha/\lambda$
gap extension	$\tilde{s}_{\texttt{AA}}$, $\tilde{s}_{\texttt{BB}}$	$-\beta = -e$	$\log \varepsilon = -\beta/\lambda$
gap followed by gap	$\tilde{s}_{\texttt{AB}}$	$-\infty$ †	$\log \psi_{\texttt{A}} = -\infty$
	$\tilde{s}_{\texttt{BA}}$	$-\alpha = -d - e$	$\log \psi_{\texttt{B}} = -\alpha/\lambda$

Table 1: Relationship between the scores of the pair HMM and those of score-based alignments. † This is forbidden by convention.

where λ is the scale of the scoring function defined above. Note that this global rescaling of the score and the term $\log Z_{(\ell,m)}$ does not change the optimal alignment and is hence arbitrary. The resulting parametrization are summarized in Table A.4. The solution of the set of equations in the last column uniquely determine $\nu, \nu_{\texttt{A}}, \eta_{\texttt{A}}, \varepsilon, \psi_{\texttt{A}}, \psi_{\texttt{B}}$.

B Software Availability

The software is published under following conditions:

- **Project name:** ppALIGN - Posterior probabilities for score-based alignments
- **Project home page:** http://ppalign.sourceforge.net
- **Demo server:** http://www.mi.parisdescartes.fr/ppblast
- **Operating system(s):** Platform independent, tested with linux and OS X
- **Programming Language:** C++, tested with gcc 4.4
- **Other requirements:** expat, GD library (not for the core library), cmake or GNU make

- **License:** GPL

References

Altschul, S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman (1990): "Basic local alignment search tool," *J. Mol. Biol.*, 215, 403–410.

Arratia, R. and M. S. Waterman (1994): "A phase transition for the score in matching random sequences allowing deletions," *Ann.Appl. Prob.*, 4, 200–225.

Arribas-Gil, A., E. Gassiat, and C. Matias (2006): "Parameter estimation in pair-hidden markov models," *Scandinavian Journal of Statistics*, 33, 651–671, URL http://dx.doi.org/10.1111/j.1467-9469.2006.00513.x.

Aston, J. A. D. and D. E. K. Martin (2007): "Distributions associated with general runs and patterns in hidden markov models," *Ann. Appl. Stat.*, 1, 585–61.

Baum, L. E., T. Petrie, G. Soules, and N. Weiss (1970): "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *Ann. Math. Statist.*, 41, 164–171.

Dayhoff, M. O., R. M. Schwartz, and B. C. Orcutt (1979): "A model of evolutionary change in proteins," in M. O. Dayhoff, ed., *Atlas of Protein Sequence and Structure*, volume 5 Suppl. 3, National Biomedical Research Foundation, 345–352.

Durbin, R., S. Eddy, A. Krogh, and G. Mitchison (1998): *Biological Sequence Analysis*, Cambridge University Press.

Hartmann, A. K. (2009): *Practical Guide to Computer Simulations*, Singapore: World Scientific.

Heinkoff, S. and J. G. Heinkoff (1992): "Amino acid substitution matrices from protein blocks," *Proc.Natl.Acad.Sci.U.S.A.*, 89, 10915–10919.

Higgins, D. G. and P. M. Sharp (1988): "CLUSTAL: a package for performing multiple sequence alignment on a microcomputer." *Gene*, 73, 237–244.

Holmes, I. and R. Durbin (1998): "Dynamic programming alignment accuracy," *Journal of Computational Biology*, 5, 493–, URL http://dx.doi.org/10.1089/cmb.1998.5.493.

Jaroszewski, L., W. Li, and A. Godzik (2002): "In search for more accurate alignments in the twilight zone," *Protein Sci*, 11, 1702–1713, URL http://www.proteinscience.org/cgi/content/abstract/11/7/1702.

Kschischo, M. and M. Lässig (2000): "Finite-temperature sequence alignment," in *Pacific Symposium on Biocomputing 5*, 624–635.

Kube, M., J. Heider, P. Hufnagel, S. Kühner, A. Beck, R. Reinhardt, and R. Rabus (2004): "Genes involved in the anaerobic degradation of toluene in a denitrifying bacterium, strain EbN1," *Arch. Microbiol.*, 181, 182–184.

Kühner, S., L. Wöhlbrandt, P. Hufnagel, I. Fritz, C. Hultschig, M. Kube, P. Reinhardt, and R. Rabus (2005): "Substrate-dependent regulation of anaerobic ethylbenzene and toluene metabolism in a denitrifying bacterium, strain EbN1." *J. Bacteriol.*, 187, 1493–1503.

Larkin, M. A. et al. (2007): "Clustal w and clustal x version 2.0," *Bioinformatics*, 23, 2947–2948.

Lau, P. C., Y. Wang, A. Patel, D. Labbé, H. Bergeron, R. Brousseau, Y. Konishi, and M. Rawlings (1997): "A bacterial basic region leucine zipper histidine kinase regulating toluene degradation." *Proc. Natl. Acad. Sci. USA*, 94, 1453–1458.

Lunter, G. (2007a): "Hmmoc – a compiler for hidden markov models," *Bioinformatics*, 23, 2485–2487.

Lunter, G. (2007b): "Probabilistic whole-genome alignments reveal high indel rates in the human and mouse genomes," *Bioinformatics*, 23, i289–296, URL http://bioinformatics.oxfordjournals.org/cgi/content/abstract/23/13/i289.

Lunter, G., A. Rocco, N. Mimouni, A. Heger, A. Caldeira, and J. Hein (2008): "Uncertainty in homology inferences: Assessing and improving genomic sequence alignment," *Genome Research*, 18, 298–309, URL http://genome.cshlp.org/content/18/2/298.abstract.

Miyazawa, S. (1995): “A reliable sequence alignment method based on probabilities of residue correspondences,” *Protein Eng.*, 8, 999–1009, URL http://peds.oxfordjournals.org/cgi/content/abstract/8/10/999.

Mückstein, U., I. Hofacker, and P. Stadler (2002): “Stochastic pairwise alignments,” *Bioinformatics*, 18, 153–160, URL http://bioinformatics.oxfordjournals.org/cgi/content/abstract/18/suppl_2/S153.

Needleman, S. B. and C. D. Wunsch (1970): “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *J. Mol. Biol.*, 48, 443–453.

Newberg, L. A. (2008): “Memory-efficient dynamic programming backtrace and pairwise local sequence alignment,” *Bioinformatics*, 24, 1772–1778.

Novák, A., I. Miklós, R. Lynsgoe, and J. Hein (2008): “Statalign,” URL http://phylogeny-cafe.elte.hu/StatAlign/.

Rabiner, L. R. (1989): “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, 77, 257–286.

Rabus, R. and J. Heider (1998): “Initial reactions of anaerobic metabolism of alkylbenzenes in denitrifying and sulfate-reducing bacteria,” *Arch. Microbiol.*, 170, 377 – 384.

Rabus, R., M. Kube, A. Beck, F. Widdel, and R. Reinhardt (2002): “Genes involved in the anaerobic degradation of ethylbenzene in a denitrifying bacterium, strain ebn1.” *Arch. Microbiol.*, 178, 506–516.

Rabus, R. and F. Widdel (1995): “Anaerobic degradation of ethylbenzene and other aromatic hydrocarbons by new denitrifying bacteria,” *Arch. Microbiol.*, 163, 96–103.

Schwartz, R. M. and M. O. Dayhoff (1978): “Matrices for detecting distant relationships,” in M. Dayhoff, ed., *Atlas of Protein Sequence and Structure*, volume 5,Suppl.3, Washington,D.C.: National Biomedical Research Foundation, 353–358.

Shindyalov, I. N. and P. E. Bourne (1998): “Protein structure alignment by incremental combinatorial extension (CE) of the optimal path,” *Protein Eng.*, 11, 739–747, URL http://peds.oxfordjournals.org/cgi/content/abstract/11/9/739.

Smith, T. F. and M. S. Waterman (1981): “Identification of Common Molecular Subsequences,” *J. Mol. Biol.*, 147, 195–197.

Stoye, J., D. Evers, and F. Meyer (1998): “Rose: generating sequence families,” *Bioinformatics*, 14, 157–163, URL http://bibiserv.techfak.uni-bielefeld.de/rose/.

Suchard, M. A. and B. D. Redelings (2006): “Bali-phy: simultaneous bayesian inference of alignment and phylogeny,” *Bioinformatics*, 22, 2047–2048.

Taylor, B. L. and I. B. Zhulin (1999): “Pas domains: internal sensors of oxygen, redox potential, and light.” *Microbiol. Mol. Biol. Rev.*, 63, 479–506.

Thorne, J. L., H. Kishino, and J. Felsenstein (1992): “Inching toward reality: an improved likelihood model of sequence evolution,” *J. Mol. Evol.*, 34, 3–16.

Wolfsheimer, S., O. Melchert, and A. K. Hartmann (2009): “Finite-temperature local protein sequence alignment: Percolation and free-energy distribution,” *Phys. Rev. E*, 80, 061913.

Yu, Y. and T. Hwa (2001): “Statistical significance of probabilistic sequence alignment and related local hidden markov models,” *J. Comp. Biol.*, 8, 249–282.

Zhang, M. and T. Marr (1995): “Alignment of molecular sequences seen as random path analysis,” *J. Theor.Biol.*, 174, 119–129.