

ppALIGN: posterior probabilities for score based alignments

Stefan Wolfsheimer^{*1} , Alexander K Hartmann² , Gregory Nuel^{*1}

¹ MAP5 (UMR CNRS 8145), Department of Applied Mathematics, Paris Descartes University, France

² Institut für Physik, Universität Oldenburg, Germany

Email: Stefan Wolfsheimer* - stefan.wolfsheimer@parisdescartes.fr; Alexander K Hartmann* - alexander.hartmann@uni-oldenburg.de; Gregory Nuel* - gregory.nuel@parisdescartes.fr;

*Corresponding author

Abstract

Background: Score-based pairwise alignments are widely used in bioinformatics in particular in molecular database search tools, like the BLAST family. Such algorithms are usually fast due to sophisticated heuristics, but unfortunately the underlying scoring model lacks in a statistical description of the reliability of the reported alignments. In particular close to gaps, in low-score regions or in low-complexity regions (exhibiting repetitive pattern) of the alignment a huge number alternative alignments arise which leads the certainty about the optimal alignment to decrease.

Result: 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 parameters of the probabilistic model. For this reason it is suitable to directly analyze the outcome of popular score based aligners and search tools without the need to choose a complicated set of parameters. It only requires the classical score parameters (the score matrix and gap costs). The package comes along with a library written in C++, a standalone program for a single user defined alignment (ppALIGN) and another one (ppBLAST) that treats 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 computation. We have included alternative decoding algorithms to provide alternative alignments.

Conclusions: ppALIGN is a fast program/library that helps to detect and quantify questionable regions in pairwise alignments. Due to its structure, the input and output interface it is suitable to connect with 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.

Background

Many search tools for large molecular database, like the BLAST family [1], rely on score-based alignments due to the existence of fast search heuristics. Usually, for a given query, a list of alignments are

reported in descending order of the significance essentially in linear time.

Since score based aligners produce unique alignments that maximize the score, they lack in a statistical analysis of the accuracy of the produced align-

ment. To address this problem, various probabilistic alignment methods, such as pair hidden Markov Models (HMMs), have been developed in the last decade [2]. They provide a statistical description of the set of all alignments for given pair of sequences including alternative meaningful alignments that may be hidden behind the optimum. For instance, in the framework of probabilistic alignment, we are able to assess numerically the confidence for each aligned pair of letters and gaps. This allows us to identify questionable regions in the alignment.

A classical model, termed “finite-temperature alignment”, introduced in 1995, [3–5] gives alignments the weight of an exponential (or Boltzmann) distribution. It can directly applied to any classical scoring function with one additional parameter, the temperature T , or contrast parameter. For the canonical value $T = 1$, and using a normalized score matrix, it approximates more complex probabilistic models.

Standard pair HMMs [2] have a stronger probabilistic description because each parameter can be explained as a transition or emission probability. However, the typical model layout is usually much larger than score based alignment 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. Conditioning on fixed sequence lengths is a way to reduce the parameters of probabilistic alignments because one does not explicitly model the length of the sequences like in the standard pair HMM described for example in Durbin et.al. [2]. Yu and Hwa [6] is an example of this type of HMM.

Lunter et. al. [7] illustrated the usefulness of probabilistic alignment in detecting regions of low confidence. Especially close to gaps (i. e. insertions or deletions) often many competitive alignments decrease the accuracy of the maximum score alignment. These biases have been identified as “gap wander” [8], “gap attraction” and “gap annihilation” [9]. Gap wander describes the effect that an inferred gap position is shifted by a few pairs with respect to the “true alignment”. Gap attraction occurs when two closely distant gaps merge into a single gap in the inferred alignment and the third effect is an cancellation of an insertion and a deletion.

In this work, we introduce **ppALIGN**, which uses

¹or 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*.

²A gap of length γ is penalized with $-d - e(\gamma - 1)$

either the finite-temperature model or a pair HMM to compute the posterior probabilities. 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. The pair HMM is close to (but not exactly) the model of Yu and Hwa [6], because it virtually has the same number of free parameters than score based alignments.

The software can process either a single alignment or the entire output of BLAST. Furthermore it features an interface to integrate new features such as alternative posterior decoding algorithms as proposed by [2, 3, 7, 8].

In the following we briefly describe the model and the software architecture, followed by illustrations of the results of sample applications of **ppALIGN**. Mathematical details can be found in the supplementary material.

Implementation

Posterior probabilities

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 sequence of edit operations with $\pi_k \in \{\text{P}, \text{A}, \text{B}\}$ such that, with $\Delta_{\text{P}} = (1, 1)$, $\Delta_{\text{A}} = (1, 0)$, $\Delta_{\text{B}} = (0, 1)$, $\sum_{k=1}^t \Delta_{\pi_k} = (\ell, m)$. The operation P is referred as pair, the operation A as insertion in sequence a_1^ℓ and the operation B as insertion¹ in sequence b_1^m . If we define $(i(k), j(k)) = \sum_{k'=1}^k \Delta_{\pi_{k'}}$, we note that $a_{i(k)}$ and $b_{j(k)}$ gives the last pair of letters in the alignment position k . If $\pi_k = \text{P}$ those letters are paired.

Score based methods determine the optimal alignment $\hat{\pi}$ by maximizing an objective function s ,

$$\hat{\pi} = \underset{\pi}{\operatorname{argmax}} s(\pi; a_1^\ell, b_1^m).$$

Let $\mathcal{S}(a, b)$ denote the classical score matrix which assigns each pair of letters an integer number, $\mathcal{S} : \Sigma \times \Sigma \rightarrow \mathbb{Z}$. Given the gap open d and gap extension penalty² e the objective function for Needleman-Wunsch global alignments [10] is classically defined by

$$s(\pi_1^t; a_1^\ell, b_1^m) = \sum_{\text{paired } (a, b)} \mathcal{S}(a, b) + \mathcal{S}(\text{gaps}) \quad (1)$$

which can be rewritten here as

$$s(\pi_1^t; a_1^\ell, b_1^m) = \sum_{k, \pi_k = \text{P}} \mathcal{S}(a_{i(k)}, b_{j(k)}) + \sum_{k=1}^t \tilde{\tau}_{\pi_{k-1} \pi_k}, \quad (2)$$

where π_0 is set to P and $\tilde{\tau}_{\text{PP}} = \tilde{\tau}_{\text{AP}} = \tilde{\tau}_{\text{AP}} = 0$, $\tilde{\tau}_{\text{PA}} = \tilde{\tau}_{\text{PB}} = \tilde{\tau}_{\text{AB}} = \tilde{\tau}_{\text{BA}} = -d - e$, and, $\tilde{\tau}_{\text{AA}} = -e = \tilde{\tau}_{\text{BB}} = -e$. The symbol $\mathbb{I}_{\pi_k = \text{P}}$ denotes the indicator function which is 1 if $\pi_k = \text{P}$ and 0 otherwise.

Local Smith-Waterman alignments [11] are can be obtained by

$$\hat{\pi} = \operatorname{argmax}_{i_1 \leq i_2, j_1 \leq j_2} s(\pi; a_{i_1}^{i_2}, b_{j_1}^{j_2})$$

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

$$\mathbb{P}(\Pi = \pi | a_1^\ell, b_1^m). \quad (3)$$

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 are misplaced. The posterior distribution Eq. 3 is appropriate to quantify the degree of confidence for a given alignment.

ppALIGN uses pair-HMM techniques to marginalize the posterior distribution of Eq. 3 and determine *column-wise posterior probabilities* [2]. Let us assume that the optimal alignment relates position a_i in the first sequence is aligned with position b_j in the second sequence, or, according to our alignment definition, $\pi_k = \text{P}$ 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

$$\begin{aligned} P_{i,j}^{\text{P}} &= \mathbb{P}(\pi : a_i \text{ and } b_j \text{ aligned} | a_1^\ell, b_1^m) \\ &\equiv \mathbb{P} \begin{bmatrix} \dots & a_i & \dots \\ \dots & b_j & \dots \end{bmatrix} \end{aligned} \quad (4)$$

For the gaps we follow the definition of Lunter et. al. [7] and define the probability that the position i is gapped irrespectively from the position j and $j + 1$ between which this gap appears

$$\begin{aligned} P_i^{\text{A}} &= \mathbb{P}(\pi : a_i \text{ gapped} | a_1^\ell, b_1^m) \\ &\equiv \sum_j \mathbb{P} \begin{bmatrix} \dots & \dots & \dots & a_i & \dots & \dots & \dots \\ \dots & b_j & - & - & - & b_{j+1} & \dots \end{bmatrix} \end{aligned}$$

and likewise for gaps in the other sequence

$$\begin{aligned} P_j^{\text{B}} &= \mathbb{P}(\pi : b_j \text{ gapped} | a_1^\ell, b_1^m) \\ &\equiv \sum_i \mathbb{P} \begin{bmatrix} \dots & a_i & - & - & - & a_{i+1} & \dots \\ \dots & \dots & \dots & b_j & \dots & \dots & \dots \end{bmatrix}. \end{aligned}$$

In the case of local alignment one may ask two questions. Firstly, how sure we are that the start and end of the aligned part is correct? Secondly, how accurately is the alignment itself? To address the second question we simply turn to the global alignment problem assuming that the start and end is correct. For the confidence of the boundaries of the local alignment **ppALIGN** computes the marginal probabilities

$$P_{i,j}^{\text{start}} = \mathbb{P}(\pi \text{ starts at positions } i \text{ and } j | a_1^\ell, b_1^m)$$

and

$$P_{i,j}^{\text{end}} = \mathbb{P}(\pi \text{ ends at positions } i \text{ and } j | a_1^\ell, b_1^m)$$

The probabilities $P_{i,j}^{\text{P}}$, P_i^{A} , P_j^{B} , $P_{i,j}^{\text{start}}$, and $P_{i,j}^{\text{end}}$ can be efficiently computed by a generalization of the forward and backward algorithm [2, 12] for a pair HMM.

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. In the case that $\mathcal{S}(a, b)$ is derived from a likelihood ratio $\mathcal{S}(a, b) = \lambda \frac{p(a,b)}{q(a)q(b)}$ (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 $p(a, b) = \exp(\mathcal{S}(a, b)/\lambda) q(a)q(b)$. However, for simple scoring functions we often only know a score for matches and mismatches. In this case **ppALIGN** estimates $q(a)$ from the input sequences and determine λ by the unique root of the equation $\sum_{a,b} \exp(\mathcal{S}(a, b)/\lambda) q(a)q(b) = 1$.

Beside the pair HMM, we implemented the finite temperature model [3–5] where alignments are weighted with an exponential distribution

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

The free parameter T is termed as temperature or contrast parameter. For $T = 1$ the temperature finite temperature model approximates the

pair HMM. In the supplementary material we confirm that the differences between the pair-HMM, the finite-temperature and the HMM by Yu and Hwa [6] are only marginal. In the limit $T \rightarrow 0$ essentially only optimal alignments have a finite probability, whereas for $T \rightarrow \infty$ all alignments have equal weight. The finite temperature model allows us to explore the alignment space more generally. For example, if a certain region in an alignment persists reliable even for a larger temperature, we have a more conservative evidence.

The layout of the software

The layout of the software and data flow is illustrated in Figure 1. The user provides alignments either in FASTA format, or an entire BLAST output (in XML format) which is passed to the core library. The default output again XML. Optionally one may choose a plain text or an HTML output filter which format the output into an human readable form.

In an UNIX environment, `ppALIGN` is typically called like

```
> 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 [2, 13] and
- a decoder which maximizes the posterior probability [2, 8].

The resulting alignments of both decoding method are compared with the user supplied alignment and those regions that do not agree are pointed out. The alignment is partitioned in different segments where the alternative alignments agree with the user supplied alignment (referred as *overlapping segments* in the following) and those segments where at least one alternative alignment disagrees with the reference (referred as *non-overlapping segments*). In the non-overlapping segments, the user may cycle

between different alignment alternatives and may therefore obtain further information about the structure of the weighted alignment space. In particular, the locations of the non-overlapping segments are suitable to detect regions of uncertainty, as we show in the result section.

Knowing the optimal alignment (or a nearly optimal alignment) one may reduce the quadratic time complexity of the algorithms to a computation time which is essentially proportional to the length of the optimal alignment. This becomes possible because in the quadratic search space the pair probabilities Eq. 4 become negligible far away from the optimal alignment. By default `ppALIGN` uses an heuristic where the forward and backward sums are only computed around 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 two last 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 [5, 14, 15] which is easily signaled by a weak convergence in the procedure of estimating the strip size. This leads to a strip in the order of the quadratic search space. However, we advise to not choosing the parameter in this regime because suboptimal alignments are usually given too much weight. In all cases, if the needed workspace exceeds the provided memory, the algorithms rely on the checkpoint method by Newberg [16].

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 algorithms or other computations based on intermediate dynamic programming results can added without changing the core library. Developers need not take care about the model details (pair HMM or finite temperature) nor think about the estimation of the strip size.

Results

Global alignment

To evaluate the ability of `ppALIGN` to identify uncertain regions of global alignments, we have performed computer simulations [17] which generated

families of random sequences using the ROSE model of random sequence evolution [18]. We have generated protein sequences according a evolutionary tree given by a complete binary tree of height 3, hence exhibiting 8 leaves. At the root we started always with a sequence of human insulin of length of 86 residues. Each branch of the tree, corresponding to a descendant had length PAM D , which means that we performed D times mutations according the 1 PAM matrix [19] (ROSE default), and D times insertions and deletions with probability $p_{\text{ins}} = p_{\text{del}} = 0.003$. This rates are about 10 times greater than the default value of ROSE, hence we observe a larger number gaps. The alignment problem is more difficult and therefore more interesting for the purpose of quantifying the reliability. For the length distributions of insertions and deletions, we also took the ROSE default, i.e., lengths between one and six appear with probability 0.1, respectively, while lengths between 7 and 14 appear with probability 0.05, respectively. 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, two sample sequences are examined. One observes that the optimum alignment (Figure 3b) agrees well with the true evolutionary alignment (Figure 3a), but not everywhere. Note that most regions where we have an agreement of all alignment have been omitted in order to improve readability. Typically, the posterior probabilities, obtained using the HMM model in this case, are small where the optimum alignment does not agree with the evolutionary true alignment, i.e., it is not *correct*.³ Since the optimum alignment is the alignment with the maximum probability, the posterior probabilities are typically a bit 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 non-overlapping segments as less reliable. For sampled alignments the user may choose the number of samples. One may expect longer and more non-overlapping segments with 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 may explore al-

ternatives in the so far overlapping region (because of the large posterior probability there). For example, if we consider the true alignment and the first sample we would detect two small non-overlapping segments. Including the second sample, these two segments merge to a larger one which is already the one displayed in Figure 5 apart from 3 positions. Without considering the maximum posterior alignment the actual size of the non-overlapping segment in Figure 5 is achieved within the 10 first samples. When we draw 100 samples the critical segment is increased by 4 positions and for 1000 samples by 3 further columns. In other words, non-overlapping segments only grow very slowly with the number of samples and to our experience about 10 samples leads to meaningful results.

Thus, in principle, one may use the posterior probabilities to identify the regions of low and high confidence. To assess this quantitatively, we have simulated for values of $D = 10, 20, 40$ each time 1000 independent evolutions of the sequences, run **ppALIGN** each time for all $7 \times 8/2$ pairs of leave sequences and compared to the true alignments. In the inset of Figure 4, we show the average fraction of correctly aligned positions as function of the (binned) posterior probabilities $P \in \{P_{i,j}^P, P_i^A, P_j^B\}$, 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 correlated well with the probability that the optimum alignment is correct.

One 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 overlapping regions, which are considered as being reliable, the alignment was for $D = 10$ on average correct for 99.8% of all positions (99.1% for $D = 20$ and 95% for $D = 40$), while in the non-overlapping regions, which are considered as being unreliable, one 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*, i.e., the fraction of correct alignment positions where $p > p_{\text{thres}}$, and the *false negative rate*, i.e., the fraction of incorrect alignment

³A pair of letters is correct, if two letter at the same positions are paired in the true alignment. A gapped letter is correct, if in the true alignment it is gapped as well, irrespective of the position of the gap.

positions where $p > p_{\text{thres}}$. Clearly, for $p_{\text{thres}} \rightarrow 0$ both rates converge to 1, while for $p_{\text{thres}} \rightarrow 1$, both rates will approach zero. The behavior for intermediate values, for different values of D , is shown in the main plot of Figure 4. One observes that the curves run close to $(0, 1)$ if D is not too large, which means that using ppALIGN and this simple threshold-based criterion, one can reliably identify correctly aligned regions. Note 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, also the optimal threshold value is larger.

Note that due to the layout of the evolutionary tree we have used, the leaves have varying PAM distances $2D$ to $6D$ among each other, the average distance is $37/7D$. 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).

Confidence of local alignment

Next, we turn to the question of uncertainty of the correct start and end of local alignments. From ref [20] we know that optimal local sequence alignment might differ strongly from the structural alignment (for example obtained by the combinatorial expansion method [21]). For example, when we compare the proteins Escherichia coli dihydrodipicolinate reductase against the Thermus thermophilus malate dehydrogenase (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 regard the structural alignment as golden standard, we would infer that optimal sequence alignment produces the correct starting point but largely failed to find the correct end of the alignment.

To illustrate how ppALIGN may detect such disagreements we consider the start and end 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 Figure 5 (c) and (f). As expected, the maximum of the start point in Figure 5 (c) is much sharper than the one for the end point of the alignment Figure 5 (f).

However, such 2 dimensional plots might be in-

teresting 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 provides additionally marginalized representation displaying the probabilities that the alignment starts / ends at certain positions i or j in the input sequences irrespective of the position in the other sequence. For example, the probability that an alignment starts at position i in the first sequence a_1^ℓ is given by $P_i^{\text{start},A} = \sum_j P_{i,j}^{\text{start}}$. To illustrate the reliability of the correct start / end position one may determine a kind of confidence interval in the sequences around the position of the optimum with more than $x\%$ probability. This is illustrated in Figure 5 (a) and (b) for the start point and in Figure 5 (d) and (e) for the end point. We learn that the 90% confidence intervals for the start point are even smaller than the 50% intervals for the end 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 a strong evidence that the end of the alignment is predicted incorrectly. This observation which has been made on sequence alignment alone is consistent with the structural alignment.

Conclusion and outlook

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. One stand-alone program, ppALIGN, allows the user to provide a single alignment and the set of parameters. The other one, ppBLAST, directly uses the structured output of BLAST (XML-format). It computes the posterior probabilities for each alignment. Both programs allow for a structured output in the XML format, plain text, and a more visual HTML page. The flexible library can be extended towards new decoding algorithms and other ways of marginalization of the posterior distribution.

Currently we are 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 [22]. 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.

Availability

- **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

Authors contributions

GN designed the project, SW developed and documented the software, AKH critically tested the software and performed the ROC analysis using the ROSE simulations to illustrate the practical usefulness. All authors contributed to the manuscript.

Acknowledgements

SW obtained a PostDoc grand from the University Paris Descartes.

References

1. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: **Basic Local Alignment Search Tool.** *J. Mol. Biol.* 1990, **215**:403–410.
2. Durbin R, Eddy S, Krogh A, Mitchison G: *Biological Sequence Analysis.* Cambridge University Press 1998.
3. Miyazawa S: **A reliable sequence alignment method based on probabilities of residue correspondences.** *Protein Eng.* 1995, **8**(10):999–1009, [<http://peds.oxfordjournals.org/cgi/content/abstract/8/10/999>].
4. Zhang M, Marr T: **Alignment of Molecular Sequences Seen as Random Path Analysis.** *J. Theor. Biol.* 1995, **174**:119–129.
5. Kschischo M, Lässig M: **Finite-temperature sequence alignment.** In *Pacific Symposium on Biocomputing 5* 2000.
6. Yu Y, Hwa T: **Statistical Significance of Probabilistic Sequence Alignment and Related Local Hidden Markov Models.** *J. Comp. Biol.* 2001, **8**:249–282.
7. Lunter G, Rocco A, Mimouni N, Heger A, Caldeira A, Hein J: **Uncertainty in homology inferences: Assessing and improving genomic sequence alignment.** *Genome Research* 2008, **18**(2):298–309, [<http://genome.cshlp.org/content/18/2/298.abstract>].
8. Holmes I, Durbin R: **Dynamic Programming Alignment Accuracy.** *Journal of Computational Biology* 1998, **5**(3):493–, [<http://dx.doi.org/10.1089/cmb.1998.5.493>].
9. Lunter G: **Probabilistic whole-genome alignments reveal high indel rates in the human and mouse genomes.** *Bioinformatics* 2007, **23**(13):i289–296, [<http://bioinformatics.oxfordjournals.org/cgi/content/abstract/23/13/i289>].
10. Needleman SB, Wunsch CD: **A general method applicable to the search for similarities in the amino acid sequence of two proteins.** *J. Mol. Biol.* 1970, **48**:443–453.
11. Smith TF, Waterman MS: **Identification of Common Molecular Subsequences.** *J. Mol. Biol.* 1981, **147**:195–197.
12. Rabiner LR: **A tutorial on hidden Markov models and selected applications in speech recognition.** *Proceedings of the IEEE* 1989, **77**(2):257–286.
13. Mückstein U, Hofacker I, Stadler P: **Stochastic pairwise alignments.** *Bioinformatics* 2002, **18**(2):153–160, [http://bioinformatics.oxfordjournals.org/cgi/content/abstract/18/suppl_2/S153].
14. Arratia R, Waterman MS: **A Phase Transition for the Score in Matching Random Sequences Allowing Deletions.** *Ann. Appl. Prob.* 1994, **4**:200–225.
15. Wolfsheimer S, Melchert O, Hartmann AK: **Finite-temperature local protein sequence alignment: Percolation and free-energy distribution.** *Phys. Rev. E* 2009, **80**(6):061913.
16. Newberg LA: **Memory-efficient dynamic programming backtrack and pairwise local sequence alignment.** *Bioinformatics* 2008, **24**(16):1772–1778.
17. Hartmann AK: *Practical Guide to Computer Simulations.* Singapore: World Scientific 2009.
18. Stoye J, Evers D, Meyer F: **Rose: generating sequence families.** *Bioinformatics* 1998, **14**:157–163, [<http://bibiserv.techfak.uni-bielefeld.de/rose/>].
19. Dayhoff MO, Schwartz RM, Orcutt BC: **A model of evolutionary change in proteins.** In *Atlas of Protein Sequence and Structure, Volume 5 Suppl. 3.* Edited by Dayhoff MO, National Biomedical Research Foundation 1979:345–352.
20. Jaroszewski L, Li W, Godzik A: **In search for more accurate alignments in the twilight zone.** *Protein Sci* 2002, **11**(7):1702–1713, [<http://www.proteinscience.org/cgi/content/abstract/11/7/1702>].
21. Shindyalov IN, Bourne PE: **Protein structure alignment by incremental combinatorial extension (CE) of the optimal path.** *Protein Eng.* 1998, **11**(9):739–747, [<http://peds.oxfordjournals.org/cgi/content/abstract/11/9/739>].
22. Aston JAD, Martin DEK: **Distributions associated with general runs and patterns in hidden Markov models.** *Ann. Appl. Stat.* 2007, **1**:585–61.

Figures

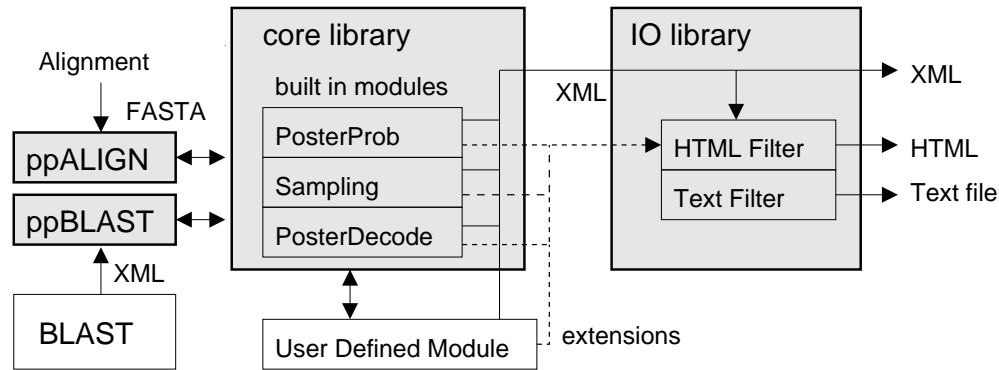


Figure 1 - Software layout and data flow

Layout of the software and data flow. The user may provide an alignment (local or global) in FASTA format or an 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 an 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 formatting rules for the output filters.

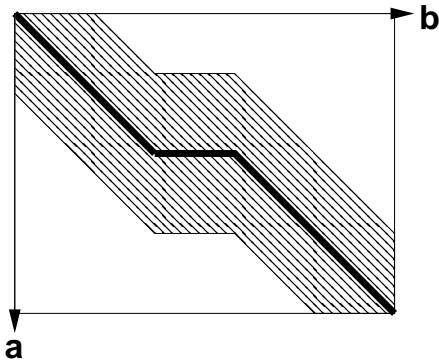


Figure 2 - Strip layout

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

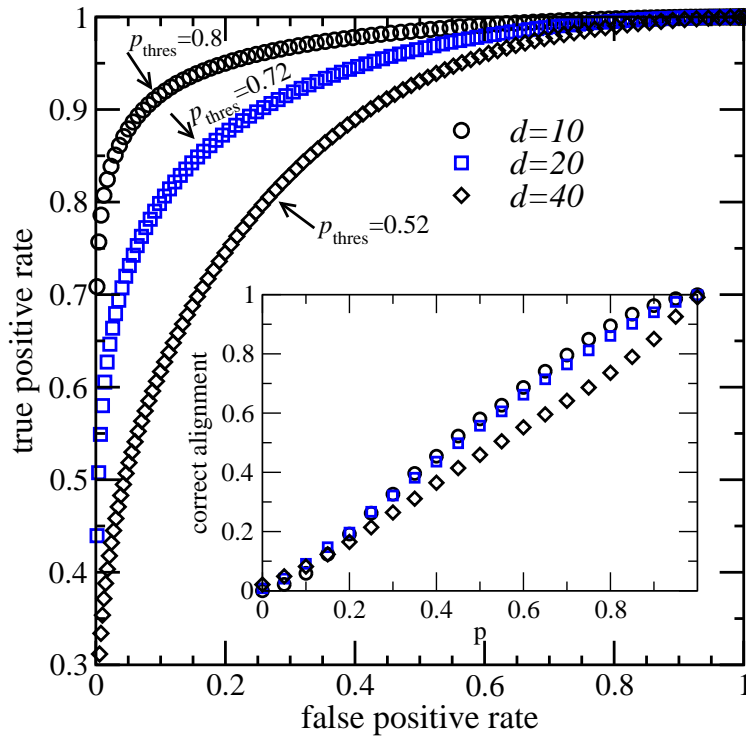


Figure 4 - ROC analysis

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 correct alignment positions as a function of the binned posterior probabilities calculated by **ppALIGN** for these positions.

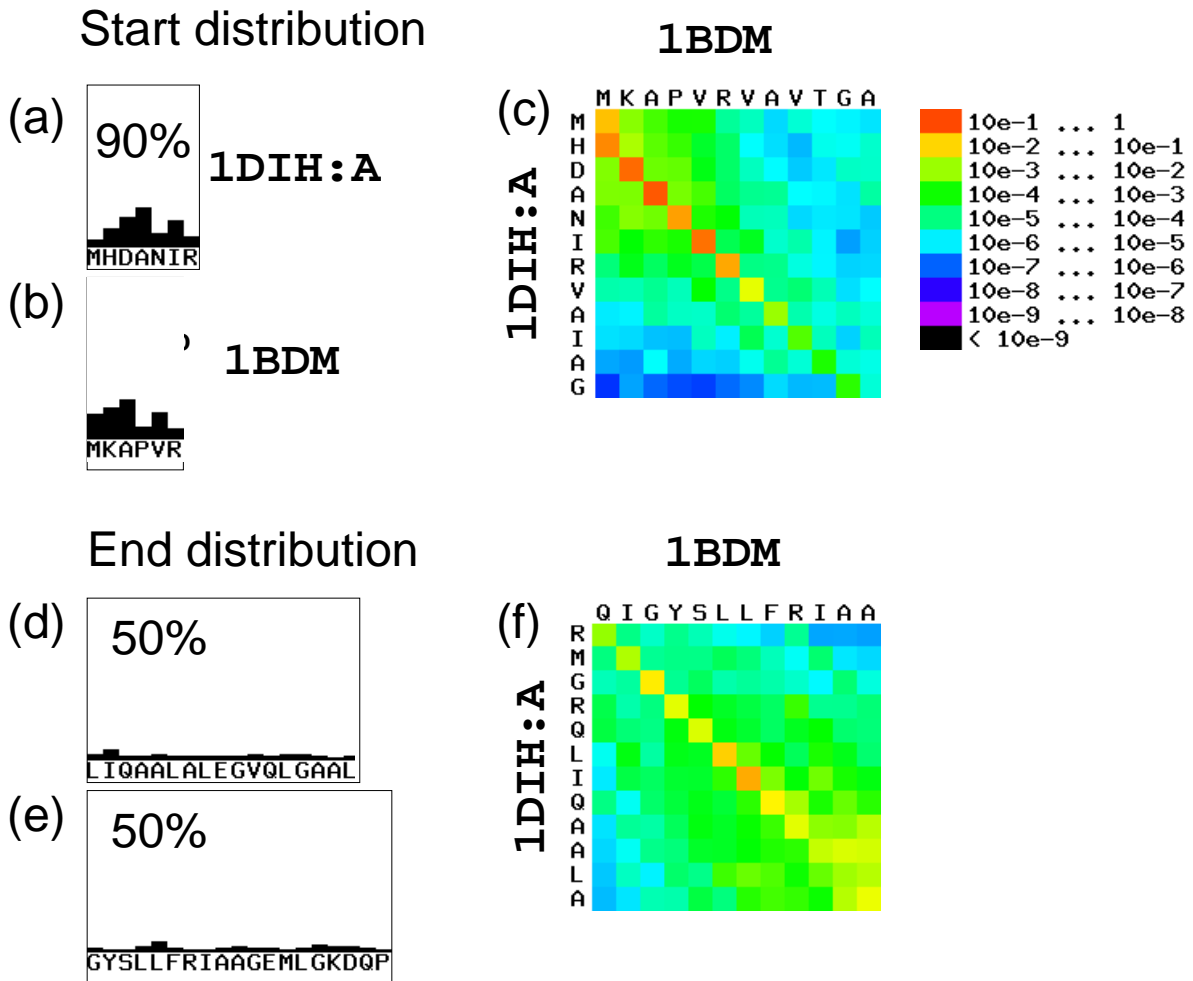


Figure 5 - Posterior probabilities for boundaries of local alignments.

The posterior probabilities for starts (a)-(c) and ends (d)-(f) of local alignments. (a) and (b) shows the 90% confidence interval for a start in the query 1DIH and the subject 1BDM. This are the subsequences around the optimal start point whose posterior start/end probabilities sum up to at least 90%. The 2 dimensional start and end distributions are shown in (c) and (f). Figure (d) and (e) are the 50% confidence intervals for ends of local alignments.

Additional Files

Additional file 1 - Theoretical background of ppALIGN

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