Aligning Sequences With Repetitive Motifs

Peter Kováč, Broňa Brejová, and Tomáš Vinař

Faculty of Mathematics, Physics, and Informatics, Comenius University in Bratislava,
Mlynská dolina, 842 48 Bratislava, Slovakia
kovac.peter@fotopriestor.sk, {vinar,brejova}@fmph.uniba.sk

Abstract. *Pairwise sequence alignment is among the most intensively studied problems in computational biology. We present a method for alignment of two sequences containing repetitive motifs. This is motivated by biological studies of proteins with zinc finger domain, an important group of regulatory proteins. Due to their evolutionary history, sequences of these proteins contain a variable number of different zinc fingers (short subsequences with specific symbols at each position).*
Our algorithm uses two types of hidden Markov models (HMM): pair HMMs and profile HMMs. Profile HMMs describe the structure of sequence motifs. Pair HMMs assign a probability to alignment of two motifs. Combination of the these two types of models yields an algorithm that uses different score when aligning conserved vs. variable motif residues. The dynamic programming algorithm that computes the motif alignments is based on the well known Viterbi algorithm. We evaluated our model on sequences of zinc finger proteins and compared it with existing alternatives.

1 Introduction

Pairwise sequence alignment is one of the most studied problems in bioinformatics. We will concentrate on alignment of protein sequences, where a protein can be represented as a string over the alphabet of 20 different amino acids. During the evolution, particular amino acids in a protein can be substituted by another amino acid, or even get inserted or deleted. The goal of sequence alignment is to compare two proteins, quantify their sequence similarity, and to identify pairs of amino acids that have likely evolved from the same amino acid in the common ancestor. Over the years, multitude of variations of this problem have been introduced and many practical software tools were developed.

Our work is motivated by the study of zinc finger proteins. These proteins contain a variable number of up to 40 zinc finger domains [18]. Zinc finger domain is a stretch of approximately 28 amino acids, the purpose of which is to bind DNA at specific places. Comparison of zinc fingers form different proteins reveals that some positions are very conserved due to their importance in assuming desired function, while other positions are highly variable, since they distinguish specific DNA sequences where individual zinc fingers bind (Figure 1).

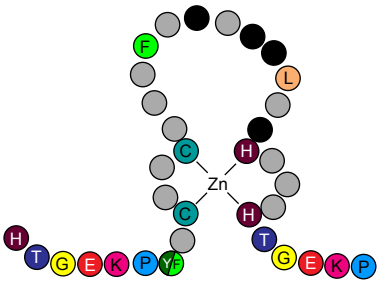

Fig. 1. The structure of a zinc-finger. Highly variable sites are marked with black color. The most conserved amino acids are the four involved in binding the zinc ion [14].

We will focus our attention on the KRAB-ZNF proteins that have a region encoding one or more Krüppel-associated box domains (KRAB, [2]) followed by a zinc finger region (Fig. 2). The human genome encodes more that 600 of proteins from this family, and a lot of effort is dedicated to building and maintaining their catalogues [9], [11], [3]. Complicated repetitive structure of these genes is a result of a dynamic evolutionary history, full of sequence duplications [7, 12], and many mutations which help to gain new functions for duplicated copies.

The repetitive nature of zinc finger protein sequences complicates their sequence alignment. Traditional alignment methods based purely on sequence similarity frequently misalign individual zinc fingers, or even align a single zinc finger in one sequence to parts of several different zinc fingers in the other sequence. Consequently, many studies of these proteins limit their analyses and infer conclusions based only on the the KRAB domains or sequences before the zinc finger region (e.g. [14], [7]), or dispute the relevance of standard methods applied to genes with high variance in the number of fingers [16].

In this work, we develop a new method for aligning sequences with repetitive motifs, such as zinc finger proteins. To overcome the problems outlined above, we combine the strength of profile hidden Markov models which are used to characterize the properties of these repetitive motifs, and pair hidden Markov models as a model of sequence alignments. We compare our work to MotifAligner that was previously used to align zinc

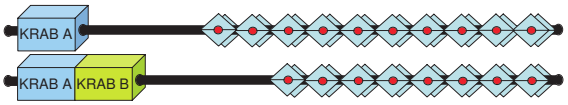

Fig. 2. Domain structure of typical KRAB-ZNF genes. The protein contains one or more KRAB domains and an array of 3 to cca. 40 zinc fingers. [18]

finger proteins [11], and we find our new method to produce more accurate alignments on our testing set.

In the rest of this section, we introduce necessary background and notation and describe the MotifAligner approach to repetitive sequence alignment in more detail. In Section 2, we describe our new profile-profile-pair alignment method (PPP). We present the results of experimental comparison of PPP and MotifAligner in Section 3.

1.1 Background and notation

In this paper, we rely on several standard tools from computational biology, namely alignments, pair hidden Markov models, and profile hidden Markov models, which we briefly explain in this section.

We start by defining hidden Markov models (HMMs). An HMM is a probabilistic finite state automaton. We can use it to generate a random sequence over some alphabet as follows. We start in a designated start state B. In each step, we sample a character of the sequence from the emission probability distribution associated with the current state and then randomly change the state according to the transition probability distribution. The process ends when we reach the designated final state E.

The sequence of states visited in the individual steps is called a state path. We will denote the probability of emitting x in state v as $e_v(x)$ and the probability of transition from state v to w as $t_{v,w}$. The joint probability of emitting a sequence $x = x_1 \dots x_n$ along the state path $s = s_1 \dots s_n$ in a given HMM is

$$P(x, s) = e_{s_1}(x_1) \prod_{i=2}^{n} t_{s_{i-1} s_i} e_{s_i}(x_i).$$

A typical task solved with HMMs is to find the most probable state path that could generate a given sequence, i.e. to find $s^* = \arg\max_s P(x, s)$. This task is solved by the Viterbi algorithm based on dynamic programming [19].

The second important notion is sequence alignment. Given a set of related protein sequences, we can align them by inserting dashes to individual sequences so that they all have the same length and when we arrange them in a table, as in Figure 3, many columns contain the same or similar amino acids. Several consecutive dashes form a gap in the alignment, indicating that a part of the sequence was deleted or inserted during the evolution. The sequence alignment problem can be formulated as an optimization problem and solved by existing algorithms. For two sequences, the problem can be solved easily by Needleman-Wunsch dynamic programming algorithm [10], for multiple sequences it is NP-hard [6]. The scoring function for pairwise alignment is typically based on a substitution matrix scoring all pairs of aligned amino acids and on parameters for scoring gaps: gap opening penalty g for the first dash in a gap and gap extension penalty e for each additional gap.

```
ZNF626_4799/12  YKC--EECGKAF-NQSSILTTHERIILERN-
ZNF727_4861/2   YKC--EECGKDC--RLSDFTIQKRIHTADRS
ZXDB_644/5      YQCAFSGCKKTF-ITVSALFSHNRAHFREQE
LLNL1236_4814/2 SMC--PECSKTSATDSSCLLMHQRSHTGKRP
ZNF23_141/15    FQC--KECGKAF-HVNAHLIRHQRSHTGEKP
```

Fig. 3. Alignment of five sequences of zinc finger motifs from human proteins.

One way of systematically deriving a scoring function for pairwise alignments is to use pair HMMs [4]. These models emit two sequences simultaneously. In one step, the HMM can emit a single character in one of the sequences or in both. The later case corresponds to two symbols aligned to each other, the former to a symbol aligned to a dash. Figure 4 shows the pair HMM used in our work. The match state M emits pairs of aligned characters, state X emits characters only in the first sequence, and state Y emits characters only in the second sequence. Given two sequences, we can find the most probable state path that could generate them and this will give us an alignment of these two sequences.

To represent a typical sequence of a motif, we will use another kind of HMMs, called profile HMMs [4]. A profile HMM is typically constructed based on an alignment of several motif instances, such as the one in Figure 3. Each position of the motif is represented by one state with emission probabilities set to the observed frequencies of amino acids in the corresponding alignment column (possibly with some pseudocounts added to avoid zero probabilities). These so called match states are arranged in a chain (see Figure 5). These states used alone would generate sequences of the same length. However, real sequences may have various insertions and deletions compared to the consensus motif; these are modeled by additional insert and delete states. Given a profile HMM and a sequence, we can again find the most probable state

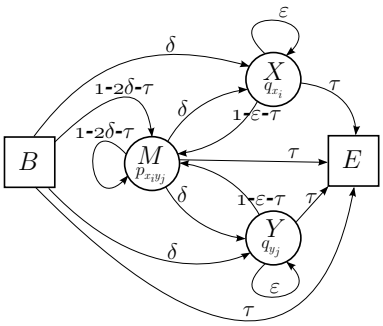

Fig. 4. A pair HMM for global alignment. Transition probabilities are defined by three parameters $\delta, \varepsilon, \tau$, emission probabilities by matrices p and q.

path, which in this case gives us an alignment of the sequence to the motif represented by the profile HMM. Note, however, that the profile HMM emits only a single sequence; the motif itself is represented directly in the structure and parameters of the model.

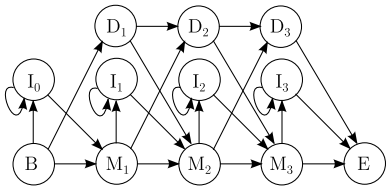

Fig. 5. Example of a profile HMM. States M_k are match states, I_k are insert states and D_k are delete states. States B, E, and $D_1 \dots D_3$ are silent, which means that they do not generate any characters.

1.2 MotifAligner Approach

To obtain high quality alignments even on sequences with highly variable number of zinc finger motifs, Nowick et al. developed a pairwise alignment tool called MotifAligner [11]. To our knowledge, it is the only sequence alignment method designed specifically to align sequences with variable number of repetitive motifs. Part of our work was inspired by this algorithm.

MotifAligner first uses a profile HMM tool HMMER [5] and finds all canonical motif occurrences with statistically significant scores in both input sequences. Let $T = (t_1, \dots, t_a)$ and $U = (u_1, \dots, u_b)$ be the sequences of all motif occurrences found by HMMER in the original input sequences x and y, respectively. In the second step, MotifAligner computes scores of all gapless pairwise alignments of motifs t_k, u_ℓ, for all $1 \le k \le a$, $1 \le \ell \le b$:

$$s[t_k, u_\ell] = \sum_{i=1}^{L} S[t_{k_i}, u_{\ell_i}], \tag{1}$$

where $S[x_i, y_j]$ is the score of aligning amino acids x_i and y_j (they use a standard BLOSUM85 substitution matrix [8]; motif occurrences are padded to have the same length).

In this way we obtain a similarity score between each pair of motif occurrences. Next MotifAligner applies the Needleman-Wunsch algorithm [10] to T and U, treating motifs as sequence symbols and using matrix s as the substitution matrix. In this way we obtain pairs of aligned zinc fingers between the two proteins.

2 Profile-Profile-Pair Alignment

In this section, we present a new approach to alignment of sequences with repetitive motifs. We adopt an approach similar to MotifAligner, however, we change the alignment algorithm and the scoring scheme to take into account the structure of the repeated motif.

For example, the zinc-finger motif (Fig. 2) contains several highly conserved positions, among them the four amino-acids binding the zinc ion (positions 3, 7, 20, 24). These four amino acids are crucial to the function of the motif and as such should be used to anchor the whole alignment. However, the fact that these positions match in the two aligned sequences should not be very surprising and should not by itself contribute much to the resulting score. On the other hand, there are several variable positions, and the differences at these positions will be very informative of the evolutionary distance.

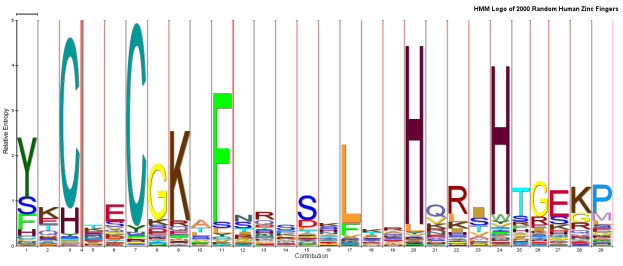

Fig. 6. The profile HMM of random 2000 human zinc fingers from the complete dataset, viewed as a HMM logo [15].

To take these issues into account, we have developed a new *profile-profile-pair* method (PPP) for alignment of individual motifs. The method uses a combination of two profile HMMs and a pair HMM for se-

quence alignment and aligns the two sequences by finding the best possible path through all three models simultaneously. To align the complete protein sequences containing these repeating motifs, we first align each possible pair of motifs through PPP, compute their similarity score, and use a modification of a traditional global alignment algorithm, now operating on individual motif occurrences as a unit. We describe the details of the method in the remainder of this section.

2.1 Pairwise Alignment of Individual Motifs

The input to PPP consists of two instances of the repeating motif $x = x_1 \dots x_{L_x}$ and $y = y_1 \dots y_{L_y}$, a profile HMM encoding the same motif, and a pair HMM characterizing the properties of a typical alignment. Our goal is to align both x and y to a separate copy of profile HMM and at the same time, use the pair HMM as a glue.

In particular, we are simultaneously seeking the three paths through the three HMMs that satisfy the following constraints:

Constraint 1 (Profile match states constraint) *If x_i and y_j are emitted by the same match state M_k in their profile models then the pair model has emit x_i and y_j together in the match state M.*

Constraint 2 (Pair match state constraint) *If the pair model emits x_i and y_j together in the match state M then both profile models emit x_i and y_j in the same match state M_k or in the same insert state I_k.*

In other words, if the pair model is in the state X or Y (which is interpreted as a gap in one of the sequences), the two profile models should not be in the same match state: symbols belonging to the same consensus column should be aligned. However, if both profile models are in the same insert state they can either be evolutionarily related, in which case they should be aligned using M state of the pair model, or they could have been inserted in the sequence independently, which would correspond to using X and Y states of the pair model. Constraint 2 also implies, that if the profile models are neither in the same match state M_k nor in the same insert state I_k (i.e. either are in completely different columns or in the same column k, but different states M_k and I_k), which means that the symbols being emitted are unrelated, then the pair model should not be in the match state. These constraints thus ensure that the sequence and the profile alignment can be interpreted in a consistent manner.

Thus our goal is to compute three paths s_p^*, s_x^*, s_y^* through the pair HMM and the two profile HMMs that would satisfy our constraints and the product of joint probabilities implied by all three models would be maximized:

$$(s_p^*, s_x^*, s_y^*) = \arg \max_{\substack{\text{valid} \\ s_p, s_x, s_y}} \text{score}(x, y, s_p, s_x, s_y), \quad (2)$$

where $\text{score}(x, y, s_p, s_x, s_y) = P_{\text{pair}}(x, y, s_p) \cdot P_{\text{profile}}(x, s_x) \cdot P_{\text{profile}}(y, s_y)$, where $P_{\text{pair}}(x, y, s)$ is the joint probability of the state path s aligning sequences x and y in the pair HMM and $P_{\text{profile}}(x, s)$ is the joint probability of the state path s and sequence x in the profile HMM.

We obtain an optimal solution using dynamic programming similar to the Viterbi algorithm used to compute the most probable state paths in individual HMMs. Let $S = (S_p, S_x, S_y)$ be the triplet of states of pair and x-profile and y-profile models satisfying our conditions. We denote $V[S_p, S_x, S_y, i, j]$ the score of the highest scoring state path combination ending with the triplet S and covering the prefixes $x_1 \dots x_i$, $y_1 \dots y_j$ of the two sequences.

The computation of $V[S_p, S_x, S_y, i, j]$ depends on the types of states S_p, S_x, S_y. For example, if S_p is the match state M of the pair HMM and S_x is the match state M_k of the profile HMM, then according to our constraints S_y must be the same match state M_k and we have the following recurrence:

$$V[M, M_k, M_k, i, j] = e_M(x_i, y_j) e_{M_k}(x_i) e_{M_k}(y_j) \cdot \max \begin{cases} t_{MM} t_{M_\ell M_k} t_{M_\ell M_k} \cdot V[M, M_\ell, M_\ell, i-1, j-1] \\ \quad \text{for } 0 \le \ell < k \\ t_{MM} t_{I_{k-1} M_k} t_{I_{k-1} M_k} \cdot V[M, I_{k-1}, I_{k-1}, i-1, j-1] \\ t_{XM} t_{M_\ell M_k} t_{M_n M_k} \cdot V[X, M_\ell, M_n, i-1, j-1] \\ \quad \text{for } 0 \le \ell, n < k, n \ne \ell \\ t_{XM} t_{M_\ell M_k} t_{I_{k-1} M_k} \cdot V[X, M_\ell, I_{k-1}, i-1, j-1] \\ \quad \text{for } 0 \le \ell < k \\ t_{XM} t_{I_{k-1} M_k} t_{M_\ell M_k} \cdot V[X, I_{k-1}, M_\ell, i-1, j-1] \\ \quad \text{for } 0 \le \ell < k \\ t_{XM} t_{I_{k-1} M_k} t_{I_{k-1} M_k} \cdot V[X, I_{k-1}, I_{k-1}, i-1, j-1] \\ t_{YM} t_{M_\ell M_k} t_{M_n M_k} \cdot V[Y, M_\ell, M_n, i-1, j-1] \\ \quad \text{for } 0 \le \ell, n < k, n \ne \ell \\ t_{YM} t_{M_\ell M_k} t_{I_{k-1} M_k} \cdot V[Y, M_\ell, I_{k-1}, i-1, j-1] \\ \quad \text{for } 0 \le \ell < k \\ t_{YM} t_{I_{k-1} M_k} t_{M_\ell M_k} \cdot V[Y, I_{k-1}, M_\ell, i-1, j-1] \\ \quad \text{for } 0 \le \ell < k \\ t_{YM} t_{I_{k-1} M_k} t_{I_{k-1} M_k} \cdot V[Y, I_{k-1}, I_{k-1}, i-1, j-1] \end{cases}$$

The value at $V[M, M_k, M_k, i, j]$ has to include the emission probabilities of x_i and y_j in all three models. Then we take a maximum over all choices of previous cells from which the current cell value could be computed. Every value considered in the maximum is the product of the value of the predecessor cell and transition probabilities in all three models. All the other

cases can be derived analogously; we omit the derivations due to the space constraints.

Every time we compute a value for any cell, we keep a pointer to the cell from which the value was derived. We use those pointers later to trace back the resulting state paths. Of particular importance is the path in the pair model, since it defines the alignment of x and y. For each of the resulting state path, we also compute its joint probability its respective model, obtaining values $P_{\text{pair}}(x, y, s_p^*)$, $P_{\text{profile}}(x, s_x^*)$, and $P_{\text{profile}}(y, s_y^*)$.

2.2 Alignment of Complete Motif Arrays

We use the same procedure as MotifAligner for alignment of complete motif arrays. We compute all pairwise alignments of individual motifs, where the score of a pairwise motif alignment is based on joint probabilities of motif sequences and state paths in all three models as described below. Since we perform the motif alignment for all pairs of motifs and assign a score to each such alignment, we get a scoring system similar to a scoring matrix. Treating motifs as symbols and using this scoring matrix, we obtain the full alignment of input motif arrays using Needleman-Wunsch algorithm.

More formally, for motif arrays $A_x = (x_1, \ldots, x_n)$ and $A_y = (y_1, \ldots y_m)$, we calculate $n \times m$ matrix S, where

$$S(x_i, y_j) = \ln \frac{P_{\text{pair}}(x_i, y_j, s^*_{p,i,j})}{P_{\text{profile}}(x_i, s^*_{x,i,j}) P_{\text{profile}}(y_j, s^*_{y,i,j})}, \quad (3)$$

where $s^*_{p,i,j}$, $s^*_{x,i,j}$ and $s^*_{y,i,j}$ are the three state paths computed when aligning motifs x_i, y_j by PPP. This score compares the hypothesis that the two motif sequences are related (given by probability from the pair HMM) to the hypothesis that these are simply two independent sequences following the same profile (as determined by scores from the two profile HMMs).

2.3 Algorithm Complexity

The time complexity of the PPP algorithm on two motif arrays with $O(n)$ motifs, each of length $O(m)$ is $O(n^2 m^6)$. There are $O(n^2)$ individual motif alignments. The time needed to compute one such alignment is $O(L_x L_y L^4)$, where L_x, L_y are the lengths of motifs and L is the number of columns in the profile HMM. This follows from the observation that in the recurrent step of individual motif alignment we fill $3 \times L \times L \times L_x \times L_y$ matrix, and time required to compute each cell is at most $O(L^2)$, the upper bound on the number of values considered in the recurrence. Typically, the number of columns in the profile HMM and the length of motifs is almost the same, so we can say that $L_x, L_y, L = O(m)$ and hence the time required to compute the alignment of one motif pair is $O(m^6)$. From the same observation, one can easily see that the space complexity is $O(n^2 + m^4)$. The running time and memory is practical, since values of n and m tend to be small in real proteins (for zinc-finger arrays, both n and m are less than 30).

Table 1. The complete dataset, based on genes from the whole human genome. One gene can have multiple variants that differ in organization of zinc fingers.

	Number of		Finger Motifs		
Genome	Genes	Variants	Total	Average	Median
hg19	612	1071	13363	12.48	12
mm9	302	513	5226	10.19	10
canFam2	477	828	9259	11.18	11
rheMac2	578	1010	12143	12.02	12

3 Experiments and Evaluation

Gold standard data set. We evaluated our approach on human zinc-finger genes and their counterparts in related species macaque, mouse, and dog. We downloaded the set of annotations of KRAB zinc finger genes from the Human KZNF Catalog [9] and remapped the annotation to the current human genome assembly hg19 using *liftOver* tool. To obtain the sequences of these zinc finger genes in other species, we used the whole genome alignments from the UCSC genome browser [17] as a mapping between the human (hg19) and the macaque (rheMac2), mouse (mm9), and dog (canFam2) genomes.

The resulting genomic sequences were translated into amino acid sequences and cleaned for apparent artifacts. In particular, we removed genes that contained fingers shorter than 10 amino acids. Summary statistics of the resulting dataset is shown in Table 1. Because of relatively high time complexity of the PPP algorithm, alignment of genes with high number of fingers takes a lot of time. For that reason, we prepared a subset of the complete dataset, omitting genes from human chromosome 19 and their putative orthologs in other genomes. These genes contain the highest numbers of repeating motifs (30 or more). Summary statistics for this *restricted dataset* are shown in the Table 2.

Model parameters. The emission probabilities of the pair HMM used in our experiments were based on the BLOSUM85 substitution matrix. This particular matrix was chosen in order to compare our results to MotifAligner [11]. In particular, we used the probability distributions p and q from which the matrix was derived, as supplied in EMBOSS software package [13]. The transition probability parameters (see Figure 4)

Table 2. The restricted dataset, omitting genes from human chromosome 19 and their orthologs.

	Number of		Finger Motifs		
Genome	Genes	Variants	Total	Average	Median
hg19	323	510	5249	10.29	9
mm9	201	314	2818	8.97	8
canFam2	257	406	3710	9.14	8
rheMac2	305	484	4766	9.85	9

were set as follows: $\tau = 0.0345$, so that the expected length of an alignment is 28, which is the length of a typical human C_2H_2 zinc finger motif; $\delta = 0.05185$ so that the expected length of a match region is 13.45, because the most variable region of a zinc finger motif spans positions 12-15; $\varepsilon = 0.4769$ so that the expected length of a gap is 1.1.

The complete parameter set of the profile model was acquired from the Pfam database entry for the ZNF C_2H_2 family [1]. The length of the profile is 23, which is shorter than a typical human zinc finger motif. The reason is that the model is based on a more diverse set of sequences from various species.

3.1 The PPP Score Distribution

To compare the scoring function of the PPP model with the scores used by MotifAligner, we created two sets of zinc-finger motif pairs. The *related* set contained 1000 fingers from the human genome, each paired with the corresponding finger from macaque, mouse or dog. The *random* set contained 1000 random pairs of fingers; we assume that these fingers are on average more distantly related to each other than paired fingers in the first set.

We have computed a PPP alignment of each sequence pair in both samples. The score distributions are shown in Figure 7. Both distributions resemble the normal distribution, with mean of the related set close to 20 and mean of the random set at around 5.

For comparison purposes, we have reimplemented MotifAligner algorithm as described in [11]. Figure 7 shows the score distributions of the MotifAligner approach to alignment of individual motifs, based on the BLOSUM85 substitution matrix. These score distributions do not resemble the normal distribution. In particular, the distribution for the related set has a heavy tail, which is clearly not desirable.

The important property of the scoring scheme is how well it is able to distinguish positive examples from negative ones. Figure 9 shows a ROC curve, where the related set was treated as positives and the random set as negative examples. The classification performance of the PPP is clearly better, demonstrating

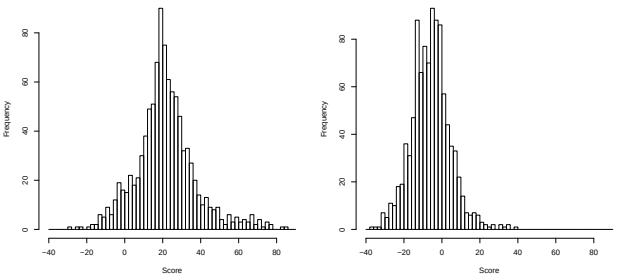

Fig. 7. Score distributions in related (left) and random (right) datasets om PPP model.

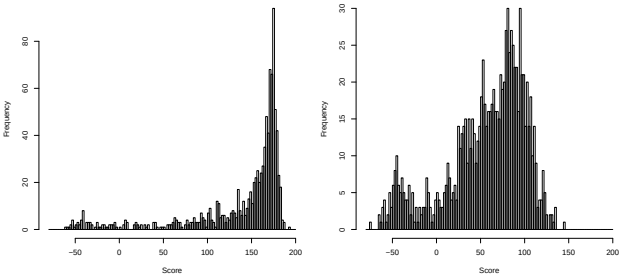

Fig. 8. The score distributions in related (left) and random (right) datasets, MotifAligner approach based on the BLOSUM85 matrix.

that our scheme is more suitable as a score for classification of paired motifs from random pairs.

3.2 Alignment Accuracy

Next we use the PPP and the simpler MotifAligner method for scoring pairs of zinc fingers as building blocks in the whole motif array alignment. The Needleman-Wunsch algorithm for the whole motif array alignment has three parameters: the gap opening penalty g, the gap extension penalty e, and the substitution matrix s that scores individual motif alignments. For MotifAligner, we have used the original parameters [11], in particular the BLOSUM85 substitution matrix and the gap penalties set to $g = 84$ and $e = 75.6$. In the PPP model, the matrix s is determined by the equation 3, and we have tested several different settings of the parameters g and e.

We carried out three tests. In the first one, we aligned all zinc finger arrays of orthologous proteins in the complete dataset. The second and the third experiments simulated a loss of fingers during the evolution—we created two artificial datasets with 1/5 and 1/3 of the total number of fingers removed in each zinc finger array in all four genomes, and we aligned the original human dataset with the four reduced sets.

In our tests, we achieved the best results when the gap opening penalty g was set to 30 and the gap extension penalty e to 20. The results of all tests are shown

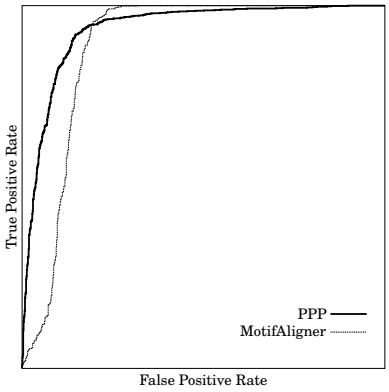

Fig. 9. ROC curve for related (positive) and random (negative) datasets.

Table 3. The comparison of MotifAligner and PPP model. PPP^1 refers to Needleman-Wunsch gap penalty parameters set to $g = 30$, $e = 20$ and PPP^2 to $g = 20$, $e = 10$. The third column lists the number of different zinc finger array pairs aligned; fourth column lists the number of wrongly aligned motifs.

Dataset	Program	Aligned arrays	Misaligned motifs
Complete, Unchanged	MotifAligner	2161	234
Complete, Unchanged	PPP^1	2161	178
Complete, Unchanged	PPP^2	2161	331
Restricted, 1/5 Loss	MotifAligner	1609	149
Restricted, 1/5 Loss	PPP^1	1609	139
Restricted, 1/5 Loss	PPP^2	1609	142
Restricted, 1/3 Loss	MotifAligner	1651	169
Restricted, 1/3 Loss	PPP^1	1651	254
Restricted, 1/3 Loss	PPP^2	1651	252

in the Table 3. PPP^1 was able to outperform the MotifAligner on the *Unchanged* and *1/5 Loss* datasets. On the other hand, our model performed slightly worse as the number of lost fingers was increased.

4 Conclusion

We have designed and implemented an algorithm for alignment of sequences with repetitive motifs. The algorithm is built on top of two types of hidden Markov models. It utilizes positional information from two copies of a profile HMM and uses a pair HMM to align the motif sequences. We were able to apply our model on real world data, and obtained better results than the only existing program specifically designed to align sequences with repetitive motifs.

There is still a room for improvement of our work. Apart from obvious upgrades, like a more efficient implementation, the underlying model can be enhanced in several ways. For example, an interesting question is whether some other scoring function of individual motif alignments would perform better. Such a function might be based on different properties of the underlying models, e.g. the full probability of a sequence, instead of the probability of the Viterbi path.

To alleviate problems caused by the computational complexity of the algorithm, various heuristics could be applied, especially methods avoiding exhausting computations of the whole dynamic programming matrix. In order to apply our model to other protein families with repeating motifs, a more robust procedure for parameter estimation should be established. In addition, a method for assessment of statistical significance of alignments may be helpful when computing alignments of large datasets where random similarities are more likely to occur.

The model we have implemented is not the only way of doing sequence alignment with repetitive motifs. It is very appealing to use a monolithic probabilistic model instead of multiplying probabilities of three separate models. We have tried to develop such a model, but we were not able to overcome some of their intrinsic difficulties.

From the practical point of view, the most serious problem we have encountered is the lack of reliable benchmark for assessing the accuracy of alignments with repetitive motifs. A high quality reference is very valuable, because it allows exact evaluation of algorithms and can give a clue where are the weak and the strong parts of a particular method, or how to set the method parameters to ensure optimal performance. We hope that our work will at least partially serve as a catalyst towards the creation of such a resource.

Acknowledgements. This work was supported by the European Community FP7 Marie Curie grants IRG-224885 to TV and IRG-231025 to BB, and by a grant from VEGA 1/1085/12.

Bibliography

[1] Bateman, A., Boehm, S., Sonnhammer, E. L. L., and Gago, F. (2011). Mulitple Sequence Alignment of Zinc Finger C2H2 Type Family. Pfam Family: zf-C2H2 (PF00096). Online. http://pfam.sanger.ac.uk/family/PF00096.

[2] Bellefroid, E. J., Poncelet, D. A., Lecocq, P. J., Revelant, O., and Martial, J. A. (1991). The evolutionarily conserved Krppel-associated box domain defines a subfamily of eukaryotic multifingered proteins. *Proc. Natl. Acad. Sci. U.S.A.*, 88(9):3608–3612.

[3] Ding, G., Lorenz, P., Kreutzer, M., Li, Y., and Thiesen, H. J. (2009). SysZNF: the C2H2 zinc finger gene database. *Nucleic Acids Res.*, 37(Database issue):D267–273.

[4] Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological sequence analysis*. 1st edition. Cambridge University Press. 356 p., ISBN: 978-0521629713.

[5] Eddy, S. R. (2011). Accelerated Profile HMM Searches. *PLoS Comput. Biol.*, 7(10):e1002195.

[6] Elias, I. (2006). Settling the intractability of multiple alignment. *Journal of Computational Biology*, 13(7):1323–1339.

[7] Hamilton, A. T., Huntley, S., Tran-Gyamfi, M., Baggott, D. M., Gordon, L., and Stubbs, L. (2006). Evolutionary expansion and divergence in the ZNF91 subfamily of primate-specific zinc finger genes. *Genome Res.*, 16(5):584–594.

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

[9] Huntley, S., Baggott, D. M., Hamilton, A. T., Tran-Gyamfi, M., Yang, S., Kim, J., Gordon, L., Branscomb, E., and Stubbs, L. (2006). A comprehensive catalog of human KRAB-associated zinc finger genes: insights into the evolutionary history of a large family of transcriptional repressors. *Genome Res.*, 16(5):669–677.

[10] Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48(3):443–453.

[11] Nowick, K., Fields, C., Gernat, T., Caetano-Anolles, D., Kholina, N., and Stubbs, L. (2011). Gain, loss and divergence in primate zinc-finger genes: a rich resource for evolution of gene regulatory differences between species. *PLoS ONE*, 6(6):e21553.

[12] Nowick, K., Hamilton, A. T., Zhang, H., and Stubbs, L. (2010). Rapid sequence and expression divergence suggest selection for novel function in primate-specific KRAB-ZNF genes. *Molecular Biology and Evolution*, 27(11):2606–2617.

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

[14] Schmidt, D. and Durrett, R. (2004). Adaptive evolution drives the diversification of zinc-finger binding domains. *Mol. Biol. Evol.*, 21(12):2326–2339.

[15] Schuster-Bockler, B., Schultz, J., and Rahmann, S. (2004). HMM Logos for visualization of protein families. *BMC Bioinformatics*, 5:7.

[16] Thomas, J. H. and Emerson, R. O. (2009). Evolution of C2H2-zinc finger genes revisited. *BMC Evol. Biol.*, 9:51.

[17] UCSC (2009). Human Genome Feb. 2009 (hg19, GRCh37) Pairwise Alignments. Online. http://hgdownload.cse.ucsc.edu/downloads.html.

[18] Urrutia, R. (2003). KRAB-containing zinc-finger repressor proteins. *Genome Biology*, 4(10):231.

[19] Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymtotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, IT-13:260–267.