

# Phylogenetic Footprinting and Consistent Sets of Local Alignments

Wolfgang Otto<sup>1,2</sup>, Peter F. Stadler<sup>4,1,2,5-8</sup>, Sonja J. Prohaska<sup>3,2</sup>

<sup>1</sup>Max Planck Institute for Mathematics in the Sciences, Inselstraße 22, D-04103 Leipzig, Germany; <sup>2</sup>Interdisciplinary Center for Bioinformatics, University of Leipzig, Härtelstraße 16-18, D-04107 Leipzig, Germany; <sup>3</sup>Computational EvoDevo Group, Department of Computer Science, University of Leipzig, Härtelstraße 16-18, D-04107 Leipzig, Germany; <sup>4</sup>Bioinformatics Group, Department of Computer Science, University of Leipzig, Härtelstraße 16-18, D-04107 Leipzig, Germany; <sup>5</sup>Fraunhofer Institut für Zelltherapie und Immunologie, Perlickstraße 1, D-04103 Leipzig, Germany; <sup>6</sup>Center for noncoding RNA in Technology and Health, University of Copenhagen, Grønnegårdsvej 3, DK-1870 Frederiksberg C, Denmark; <sup>7</sup>Department of Theoretical Chemistry, University of Vienna, Währingerstraße 17, A-1090 Wien, Austria; <sup>8</sup>Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM 87501, USA

**Abstract.** The problem of constructing alternative local multiple sequence alignments from a collection of local pairwise alignments arises naturally in phylogenetic footprinting. Based on a theoretical analysis of the problem we devise an efficient heuristic and introduce the software tool **tracker2** for this task. Tests on both biological and random data demonstrated the heuristic yields excellent results at very short runtimes.

**Key words:** alignment consistency, phylogenetic footprinting, combinatorial optimization, **tracker2**

## 1 Introduction

The discovery of functional sequence elements in genomic DNA data is an important research topic in bioinformatics [1]. Most individual binding motifs, in particular transcription factor binding sites (TFBS) are short and gapless. Their overrepresentation in the surrounding of co-regulated genes makes them detectable by motif discovery approaches such as **meme** [2] and **footprinter** [3]. Regulatory sequence elements are often (but not always) subject to stabilizing selection and hence evolve much more slowly than adjacent non-functional DNA. Such *phylogenetic footprints* are therefore detectable by comparative sequence analysis. A large class of tools thus combines pattern search with the explicit analysis of conservation, see e.g. [4–7]. However, pattern discovery approaches usually fail when not only small promoter-proximal regions but large intergenic regions are under investigation. Techniques based on global or local sequence alignments are successfully employed in such cases [8, 9].

As an alternative to the analysis of a single global alignment it has been suggested to start from local alignments between all pairs of sequences of interest

[10]. This is appealing in particular when large stretches of orthologous sequences need to be analyzed. This approach leads, however, to contradictory signals of sequence similarity that require a sophisticated post-processing. Since regulatory modules need not be co-linear, the construction of alternative clusters of pairwise alignments has been suggested. Here we revisit the approach of [10]. We discuss its theoretical foundation, starting with ideas from [11] on the consistency of alignments, see also [12]. An efficient heuristic for the assembly of maximal local multiple sequence alignments from local pairwise alignments is implemented in the software tool `tracker2`.

## 2 Theory

*Definitions and Basic Properties.* Following [13, 11] we consider sequence alignments as vertex-labeled graphs. Each position of an input sequences corresponds to a vertex. The so-called alignment edges, that is, matches or mismatches between sequence positions, form the edges of the graph.

We formalize this picture as follows: Consider a set  $X$  of strings  $x_a$ ,  $1 \leq a \leq M$ , with non necessarily equal lengths  $n_a = |x_a|$  over a common alphabet  $\mathcal{A}$ . The symbol  $x_{ai}$ ,  $1 \leq i \leq n_a$  refers to the  $i$ -th letter of the  $a$ -th string.

**Definition 1.** *A multiple alignment of  $X$  is a graph  $\Gamma(X)$  with vertex set*

$$\mathbb{V} = \{(a, i) | 1 \leq i \leq |n_a|, 1 \leq a \leq M\}, \quad (1)$$

*vertex labels  $x : \mathbb{V} \rightarrow \mathcal{A}$ ,  $(a, i) \mapsto x_{ai}$ , and an edge set  $A$  satisfying the following two conditions:*

1. *The connected components of  $(\mathbb{V}, A)$  are complete graphs. These complete graphs correspond to the alignment columns.*
2. *If  $(a, i)$  and  $(a, j)$  are contained in the same connected component, then  $i = j$ . Thus, every alignment column contains at most one position from each sequence.*
3. *There is a partial order  $\prec$  on the set of connected components so that for any two components  $P$  and  $Q$  containing vertices  $(a, i) \in P$  and  $(a, j) \in Q$  the ordering  $i < j$  along the sequence implies  $P \prec Q$ . Alignment columns therefore never cross each other.*

Given a set  $X$  of  $M$  sequences, whose length is bounded by  $N$ , and a set of edges  $E$  over  $\mathbb{V}$ , it can be decided in  $O(N^2 M^3)$  time whether  $G = (\mathbb{V}, E)$  is an alignment, by first finding the connected components of  $G$  [14], checking whether each component is a complete graph, checking time whether  $\prec$  is well defined for all pairs of components (the time-limiting step), computing the transitive closure of  $\prec$ , and using topological sorting [15] to verify that  $\prec$  is a partial order.

In the following we write  $x_a[i, j] := x_{ai}x_{a,i+1} \dots x_{a,j-1}x_{aj}$  for any  $1 \leq i \leq j \leq n_a$  for an infix of  $x_a$ . We write  $y_a \subseteq x_a$  to denote an arbitrary order-preserving subset of  $x_a$ . We choose the notation so that  $y_a$  inherits the positional indices from  $x_a$  and set  $y_{ai} = \emptyset$  for those positions of  $x_a$  that are not contained in  $y_a$ .

**Lemma 1.** *Let  $\Gamma(X)$  be an alignment of a set of strings  $X$ , and let  $X' = \{y_a | a \in U\}$  be a set of strings such that  $U \subseteq \{1, \dots, M\}$  and  $y_a \subseteq x_a$ . Then the subgraph of  $\Gamma(X)$  induced by  $X'$  is an alignment.*

*Proof.* It suffices to observe that every string in  $X'$  is of the form  $y_a = x_a[l, r]$  with  $1 \leq l \leq r \leq |y_a|$  and hence  $X'$  corresponds to the vertex set  $\mathbb{V}' = \{(a, i) | a \in U, i : y_{ai} \neq \emptyset\}$ . By construction,  $\mathbb{V}' \subseteq \mathbb{V}$ . Furthermore,  $y : \mathbb{V}' \rightarrow \mathcal{A}$  satisfies  $(a, i) \mapsto y_{ai} = x_{ai}$ . The subgraph of  $\Gamma(X)$  defined by  $X'$  thus equals the induced subgraph  $\Gamma(X)[\mathbb{V}']$ . Every induced subgraph of a complete graph is again complete, thus property (i) is satisfied. The other two properties are satisfied for all subgraphs.

For simplicity we write  $\Gamma[X']$  or, if necessary,  $\Gamma(X)[X']$  to denote the restriction of an alignment to a subset of subsequences.

In particular, therefore, every pair of sequences  $x_a, x_b$  gives rise to a pairwise alignment as an induced subgraph  $\Gamma(X)[\{x_a, x_b\}]$  of  $\Gamma(X)$ . Similarly, every individual alignment edge of  $\Gamma(X)$  can also be interpreted as an alignment.

**Definition 2.** *Let  $X$  be a set of sequences and let  $\mathcal{C} = \{\Gamma_k(Y_k)\}$  be a collection of alignments of subsequences  $y_a \subseteq x_a \in X$ . We say that  $\mathcal{C}$  is consistent if there is an alignment  $\Psi(X)$  of  $X$  so that  $\Psi(X)[Y_k] = \Gamma(Y_k)$ , i.e., the given alignments  $\Gamma_k$  are the restrictions of  $\Psi(X)$  to the subset of subsequences  $Y_k$ .*

Given a set  $X$  of sequences and a collection  $\mathcal{C} = \{\Gamma_k(Y_k)\}$  of alignments let  $\bigcup\{\mathcal{C}\}$  denote the union of set of all alignment edges in each of the  $\Gamma_k(Y_k)$ .

**Lemma 2.** *Let  $X$  be a collection of sequences and let  $\mathbb{V}$  be defined as in equ.(1). A collection  $\mathcal{C}$  of alignments on  $X$  is consistent if and only if the transitive closure of the graph  $(\mathbb{V}, \bigcup\{\mathcal{C}\})$  is an alignment.*

*Proof.* An alignment graph is transitive since, by definition, it is a disjoint union of complete graph. Consistency of  $\mathcal{C}$ , on the other hand, implies that  $(\mathbb{V}, \bigcup\{\mathcal{C}\})$  is a subgraph of an alignment  $\Psi$ . In particular, therefore, the connected components of  $\Psi$  contain at most one vertex from each sequence, and hence the connected components of  $(\mathbb{V}, \bigcup\{\mathcal{C}\})$  also contain at most one vertex from each sequence. Now observe that the transitive closure of a graph equals the disjoint union of the transitive closures of its connected components. Thus, the transitive closure of  $(\mathbb{V}, \bigcup\{\mathcal{C}\})$  is a transitive subgraph of  $\Psi$ , and thus itself an alignment.

Finally,  $\beta$  denotes a weighting function defined on the alignments. Note that it not necessary that  $\beta$  is additive. In fact, the weight of each input alignment  $\Gamma_k(Y_k)$  is arbitrary in our setting.

*Combinatorial Optimization Problem.* *Given a set of strings  $X$  and a collection  $\mathcal{C} = \{\Gamma_k(Y_k)\}$  of alignments of subsequences of the elements of  $X$ , find a maximum sub-collection  $\mathcal{C}' \subseteq \mathcal{C}$  that is consistent.*

Here, maximality can be defined either in terms of cardinality or in terms of the weights  $\beta(\Gamma_k(Y_k))$ . Note that as an alternative one might want to optimize

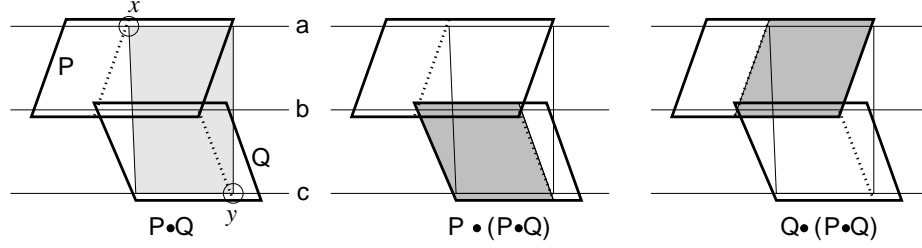


Fig. 1. Concatenation of pairwise alignments.

the sum-of-pair score of the multiple alignment  $M$  formed by combining the alignment edges of the members of  $\mathcal{C}'$ .

In practise, this is of particular interest in two settings:

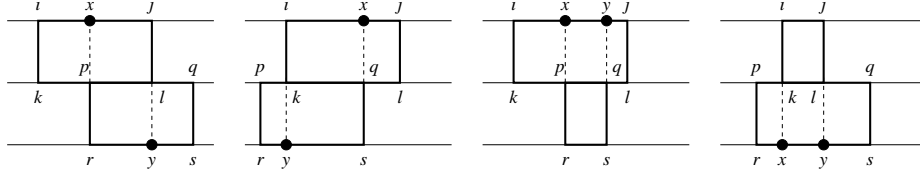
1. All  $\Gamma_k(Y_k)$  are individual alignment edges. This version of our problem is the problem faced by consistency-based alignment procedures. For example **T-coffee** [16] takes a “library” of alignment edges and then employs a heuristic approach to extract a collection of alignment edges consistent with a multiple alignment of maximal score. In practise, the pairwise alignment edges are often computed from pairwise alignments.
2. All  $\Gamma_k(Y_k)$  are local pairwise sequence alignments. This has been a starting point for footprinting tool **tracker** [10].

Instead of the maximum consistent subset we are interested in the collection of all maximal consistent subsets of  $\mathcal{C}$  in particular in the context of phylogenetic footprinting.

Clearly, consistency is hereditary, i.e., the consistent subsets of  $\mathcal{C}$  form an independence system [17]. It is not a matroid or greedoid, however. In general, therefore, distinct maximal consistent subsets may have different cardinalities and the canonical greedy algorithm will in general fail to find maximum consistent subsets [18]. In fact, the problem is NP-complete in general because the multiple alignment problem is the special case with  $\mathcal{C}$  being the collection of all possible alignment edges on  $X$ . Hardness results for multiple sequence alignment are proved in [19].

*Alignment Splitting.* The construction of the transitive closure above has an alternative interpretation as a concatenation or transfer operation between pairwise alignments  $P$  and  $Q$ . This operation, denoted by  $\bullet$ , acts as follows:

1. If  $P$  and  $Q$  are *disjoint*, i.e, there is no vertex  $(a, i)$  incident to both an edge in  $P$  and  $Q$ , we set  $P \bullet Q = \emptyset$ .
2. If  $P$  and  $Q$  are two pairwise alignments of the same two sequences, we define  $P \bullet Q := P \cap Q$  as the set of vertices and edges that are common to both alignments.
3. If  $P$  and  $Q$  have exactly one sequence in common, say  $b$ , we define at  $\{(a, i), (c, k)\}$  is an edge of  $P \bullet Q$  if and only if there is a vertex  $(b, j)$  such



**Fig. 2.** Construction of additional boundaries when alignments are concatenated.

that  $\{(a, i), (b, j)\}$  is an alignment edge in  $P$  and  $\{(b, j), (c, k)\}$  is an alignment edge in  $Q$ . The vertex set belonging to  $P \bullet Q$  is the minimal interval of  $x_a$  and  $x_c$  so that all edges of  $P \bullet Q$  are supported.

Note that the edge set  $E(P \bullet Q)$  is the relational composition  $E(P) \circ E(Q)$ , see also [11]. In fact,  $P \bullet Q$  is the pairwise alignment of sequences  $a$  and  $c$  that is *implied* by the alignment edges of  $P$  and  $Q$ . By construction, furthermore,  $\{P, Q, P \bullet Q\}$  is consistent provided  $\{P, Q\}$  is consistent.

We observe that the  $\bullet$  operation is commutative by definition. It is not associative, however. In case (2), repeated application of the operation will not produce any additional alignments. In case (3), however, we observe that  $P \bullet (P \bullet Q)$  is a non-trivial subset of alignment edges of  $Q$ : To see this just note that  $\{(i, a), (j, b)\} \in P$  concatenated with  $\{(i, a), (k, c)\} \in P \bullet Q$  yields the edge  $\{(j, b), (k, c)\}$ , which by construction of  $P \bullet Q$  is contained already in  $Q$ . Analogously,  $Q \bullet (P \bullet Q) \subseteq P$ . Further concatenations do not lead to additional distinct alignments, see Figure 1. For instance we have  $(P \bullet Q) \bullet (P \bullet (P \bullet Q)) = Q \bullet (P \bullet Q)$ .

For a collection of pairwise alignments  $\mathcal{C}$  the transitive closure w.r.t. the  $\bullet$  operation,  $\mathfrak{T}(\mathcal{C})$  is well defined as the collection of all pairwise alignments that can be generated from  $\mathcal{C}$  by repeated application of the  $\bullet$  operator. By construction, the union of the alignment edges in all pairwise alignments of  $\mathfrak{T}(\mathcal{C})$  equals the transitive closure of  $(\mathbb{V}, \bigcup \{\mathcal{C}\})$ . A set of pairwise alignments is therefore consistent if and only if the set of pairwise alignments generated by  $\bullet$  is consistent.

Differences of alignments are also well-defined in terms of their graphs. For example,  $P \setminus (Q \bullet (P \bullet Q))$  specifies the part of  $P$  in Fig. 1 which cannot be extended to an alignment of all three sequences. The transitive closure of any consistent collection of pairwise can be decomposed in this way into alignment blocks.

*Alignments as paired intervals.* In [10] local alignments were treated as matches (or pairings) between two sequence intervals, disregarding the exact position of the individual alignment edges within the intervals. In the present formalism developed above this can be implemented by representing only the *delimiting* edges of every pairwise alignment.

Clearly, this is only an approximation. In order to construct  $P \bullet Q$ , two additional interval boundaries, denoted by  $x$  and  $y$  in the example of Fig. 1, must be computed. There are only four cases, listed in Fig. 2. In principle, these

boundaries could be derived from edges in the original alignment. Instead, [10] uses a linear interpolation scheme. For instance, in the first case in Fig. 2 with  $k \leq p \leq l \leq q$ , we compute

$$x = i + (p - k) \frac{j - i}{l - k} \quad \text{and} \quad y = r + (l - p) \frac{s - r}{q - p} \quad (2)$$

Analogous equations are easily derived for the other three cases.

In this approximate model it makes sense to relax the consistency conditions for alignments: For example, we may want to require that two vertices of the form  $(a, i)$  and  $(a, j)$  in the same connected component satisfy  $|i - j| \leq \varepsilon$ . Analogously, for two edges with  $\{(a, i), (b, j)\}$  and  $\{(a, k), (b, l)\}$  with  $i < k - \varepsilon$  we require  $j < l + \varepsilon$ . Consistency as defined above is recovered as the special case  $\varepsilon = 0$ .

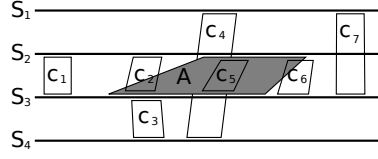
It will be convenient to specify an interval on sequence  $x$  as a triple  $[x, b, e]$  where  $b$  and  $e$  are the begin and end coordinates. A pairwise alignment is then a unordered pair of the form  $\{[x, b_x, e_x], [y, b_y, e_y]\}$ .

### 3 Heuristic Algorithm

The structure of the collection  $\mathcal{C}$  of alignments is an independence system. This suggests to explore greedy-like heuristics. We therefore construct a multiple alignment  $\mathbf{M}$  iteratively by adding one pairwise alignment  $P \in \mathcal{C}$  after the other so that the sum of the scores  $\beta(P)$  of the incorporated pairwise alignments is maximized at the end. The crucial issue, therefore, is the choice of a good order in which the input alignments are added to the growing multiple alignment  $\mathbf{M}$ . Intuitively, the optimal collection  $\mathcal{C}'$  will contain in particular all those alignments that are “biologically correct”. These are unknown in real life, of course. However, partial alignments that are supported by many other alignments are at least good candidates. We therefore adopt the idea, which proved successful in **T-coffee** [16], to increase the scores of those alignments that are well-supported by other alignments.

*Extended Scores.* A pairwise alignment  $A \in \mathcal{C}$  is supported by  $B \in \mathcal{C}$  if  $A$  and  $B$  align the same regions, i.e., if  $A \bullet B \neq \emptyset$ . Similarly,  $B$  and  $C$  together support  $A$  if  $A \bullet (B \bullet C) \neq \emptyset$ . In each case, the score of  $A$  is increased proportional to the relative size of overlapping region. For example in the latter case the bonus for  $A$  is  $|A \bullet (B \bullet C)| / |A| \times \text{score}(A)$ . Bonus scores are computed from all pairs and triples of alignments.

*Greedy Heuristic.* We order  $\mathcal{C}$  by the extended alignment scores and treat it as a queue. In the beginning  $\mathcal{C}' = \emptyset$  and the multiple alignment  $\mathbf{M}$  is the graph with vertex  $\mathbb{V}$  without any edges. In the induction step, we take the highest-scoring alignment  $A$  off  $\mathcal{C}$  and check whether  $\mathcal{C}' \cup \{A\}$  is consistent. If so, we add  $A$  to  $\mathcal{C}$ , insert all alignment edges of  $A$  into the graph  $\mathbf{M}$ , and compute its transitive closure.



**Fig. 3.** Possible locations of columns relative to the entries in alignment  $A$ . Column  $c_1$  is a prefix in both species  $S_2$  and  $S_3$ , while  $c_7$  is a suffix,  $c_4$  is independent, and  $c_5$  overlaps in both species. The remaining columns  $c_2$  (prefix in  $S_2$ ),  $c_3$  (overlap in  $S_3$ ) and  $c_6$  (suffix in  $S_3$ ) touch  $A$  in only one species.

In practise, insertion of  $A$  and the consistency test is performed columnwisely. We insert the alignment edge  $(a, i)(b, j)$  of  $A$  into  $\mathbf{M}$ , compute the transitive closure by connecting  $(a, i)$  to all neighbors of  $(b, j)$  in  $\mathbf{M}$ , and *vice versa*. Then we check whether the expanded column still satisfies the partial order condition.

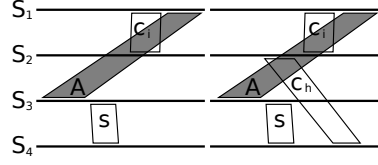
*Interval pairs.* Insertion of  $A$  into the growing alignment  $\mathbf{M}$  and the subsequent consistency checks can be performed with particular efficiency when pairwise alignments are represented as interval pairs. In the following paragraphs we describe the fast exact insertion algorithm for this case in detail. It is implemented in **tracker2**.

Each “column” (connected component)  $c_i$ ,  $i = 1, \dots, n$ , of  $\mathbf{M}$  is represented as a set of intervals containing at most one member from each input sequence. The candidate  $A$  has the form  $A = \{[x, b_x^A, e_x^A], [y, b_y^A, e_y^A]\}$ . We first determine the position of  $A$  relative to the columns  $c_i$ , Fig. 3. For both intervals of  $A$  we distinguish four cases:

- independence:**  $c_i$  contains no entry  $[z, b_z, e_z]$  with  $z \in \{x, y\}$ . In this case,  $c_i$  contains no information about the location of the  $z$ -entry of  $A$  in  $\mathbf{M}$ .
- overlap:**  $c_i$  contains an entry  $[z, b_z, e_z]$  with  $z \in \{x, y\}$  and  $[z, b_z^A, e_z^A] \cap [z, b_z, e_z] \neq \emptyset$ . Thus  $c_i$  can be extended by the information of  $A$ .
- prefix:**  $c_i$  contains an entry  $[z, b_z, e_z]$  with  $z \in \{x, y\}$  and  $e_z < b_z^A$ . Here,  $c_i$  is in front of  $A$  in species  $z$  and we check whether  $c_i$  is the closest prefix  $p_z$  of  $A$  in species  $z$  detected so far.
- suffix:**  $c_i$  contains an entry  $[z, b_z, e_z]$  with  $z \in \{x, y\}$  and we have  $b_z > e_z^A$ . Here,  $c_i$  is behind  $A$  in species  $z$  and we check whether  $c_i$  is the closest suffix  $s_z$  of  $A$  in species  $z$  detected so far.

If  $c_i$  is independent of  $A$  for both  $x$  and  $y$ , or independent in one species and a prefix or a suffix in the other, there is nothing to do. In the prefix case, if we already found a closest suffix  $s$ ,  $A$  contains additional information on the order of  $s$  and  $c_i$  that allows us to move  $c_i$  and all smaller columns between  $s$  and  $c_i$  to the front of  $s$ . If  $s$  is smaller then  $c_i$  we have detected a crossing, see Fig. 4, and therefore reject  $A$ . In the suffix case we do not have to check for changes in ordering since we work through the columns of  $\mathbf{M}$  in their current order. Of course, if  $A$  is a prefix in one species and a suffix in the other one,  $A$  and  $c_i$  cross, and hence  $A$  is rejected.

Now we consider the cases where  $A$  and  $c_i$  overlap in at least one species, say  $x$ . We denote the overlapping interval by  $[x, \bar{b}_x, \bar{e}_x]$ . Depending on the location of  $c_i$  and  $A$  in the other species  $y$  we distinguish the five cases shown in Figure 5:



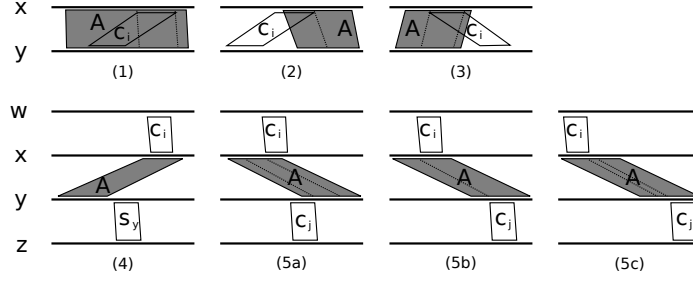
**Fig. 4.** Switching columns. In the first case we have  $c_i \prec A$  and  $A \prec s$ . Thus we move all columns  $c_j \preceq c_i$  in front of  $s$ . In the second case we have  $s \prec c_i$  so that insertion of  $A$  leads to a contradiction.

1.  $c_i$  also overlaps  $A$  in  $[y, \bar{b}_y, \bar{e}_y]$ . We need to check whether  $[y, \bar{b}_y, \bar{e}_y]$  agrees with the projection of  $[x, \bar{b}_x, \bar{e}_x]$  mediated by the column  $c_i$  and the alignment  $A$ , resp., using the linear transformation as outlined in Fig. 2, see also Equ.(2). If original and projected intervals differ by more than a tolerance  $\epsilon \geq 0$ ,  $A$  is rejected. Otherwise,  $A$  and/or  $c_i$  are split into the overlapping and the non-overlapping part.
2.  $c_i$  is a prefix of  $A$  in  $y$ . This is a contradiction and  $A$  is rejected if original and projected intervals differ by more than  $\epsilon$ . Otherwise,  $A$  is shortened to the non-overlapping part.
3. The case that  $c_i$  is a suffix of  $A$  in  $y$  is treated analogously.
4.  $c_i$  is independent for species  $y$  and we already have determined a closest suffix  $s_y$  for  $A$  in  $y$  we can move  $c_i$  and all smaller columns between  $s_y$  and  $c_i$  to the front of  $s_y$ . If this causes a crossing,  $A$  is rejected, otherwise  $A$  is split.
5.  $c_i$  is independent for species  $y$  but we have no suffix in  $y$  so far. We search the columns  $c_j$ ,  $j > i$  for the first column that overlaps  $A$  in  $y$ . If no such column exists,  $c_i$  is updated and  $A$  is split. Otherwise, we have three subcases depending on the  $c_j$ -projection of  $\bar{b}_y$  onto  $x$ , denoted by  $b'_x$ . (1) If  $b'_x < \bar{b}_x$ , we move  $c_j$  and all smaller columns to the front of  $c_i$  and check for a crossing. (2) If  $b'_x = \bar{b}_x$ , then  $A$  connects the columns  $c_i$  with  $c_j$ , which are merged. Furthermore the overlap with  $A$  is updated and  $A$  is split according to the overlap with the merged column. (3) If  $b'_x > \bar{b}_x$ , the end of the  $c_i$  overlap is updated for species  $x$  and  $c_i$  and  $A$  are split accordingly.

The splitting of the alignment and the columns is based on the overlap data. If we have an overlap in only one sequence, say  $[x, \bar{b}_x, \bar{e}_x]$  we determine the overlap with  $y$  by projection. Otherwise, the overlap is already known in both sequences. We apply the appropriate case depicted in Fig. 2 to determine the additional interval boundaries. If  $A$  has a part before and/or after the overlap with  $c_j$ , it is added as new alignment in front or behind  $c_i$ , respectively. If a new boundary arises within  $c_i$ , the alignment column is split at this position  $q_x$  using the  $c_i$  projection of  $q_x$  to all sequence represented in  $c_j$ , furthermore closest prefix and closest suffix information is updated for both parts of  $c_i$ . After all columns are checked a part of  $A$  may still remain. It is inserted as a new column into  $\mathbf{M}$  behind the closest prefix column  $p$ , in front of the closest suffix column  $s$ , or – if neither exists – at the end of  $\mathbf{M}$ .

The effort for inserting a single (local) alignment  $A$  of length at most  $\ell$  is bounded by  $M$  times the number of columns of  $\mathbf{M}$  that it intersects, i.e., by  $O(\ell M^2)$ . The greedy heuristic thus produces a solution in  $O(|\mathcal{C}| \ell M^2)$  time. In





**Fig. 5.** Types of overlaps of pairwise alignment  $A$  with a multiple alignment column  $c_i$ . In the first three cases, column  $c_i$  contains entries for both species  $x$  and  $y$  of alignment  $A$ . In the other cases  $c_i$  contains only an entry for species  $x$ . In case 4, a column behind  $A$  has already been determined. In the remaining three cases  $A$  connects two alignment columns.

practise, however, the number of columns of  $\mathbf{M}$  intersected by  $A$  is much smaller than the theoretical upper bound of  $\ell M$ .

*Alternative Solutions.* The greedy procedure above can be repeated on the set  $\mathcal{C} \setminus \mathcal{C}'$  of pairwise alignment that are inconsistent with the approximately optimal solution found in the first pass. After having extracted a maximal consistent set from  $\mathcal{C} \setminus \mathcal{C}'$  we try to add additional alignments from  $\mathcal{C}'$ . This yields another maximal consistent subset of  $\mathcal{C}$ . The procedure is iterated by initially removing all alignments from  $\mathcal{C}$  that have already been incorporated in a previous solution. We stop when every pairwise alignment is included in at least one consistent subset.

In the worst case, we obtain  $O(|\mathcal{C}|)$  solutions each comprising  $O(|\mathcal{C}|)$  consistent alignments. For datasets of practical interest we observe that the effort required to compute a set of maximal consistent alignments that cover all input alignments at least once is dominated by computation of the extended alignment scores.

*Quality of the solutions.* We used about 30 multiple alignments of different classes of ncRNAs comprising of 5-11 sequence with low pairwise similarity provided in BRaliBase [20] as source of homologous sequence sets. For each set we computed pairwise local **blast**-alignments. These instances are small enough to compute the maximum compatible set by exhaustive enumeration. In all cases, our heuristic returned the correct optimal solution. Interestingly, this solution also agrees very well with the manually curated reference alignment. In order to test the efficiency of our approach, we constructed 5000 random alignments with an average length of 50nt in 100 sequences with an average length of 200nt. On a 2.66GHz Quad Core CPU **tracker2** determined all maximal consistent subsets in the entire test set in only 27 seconds.

*Availability.* The source code of **tracker2** is available at <http://www.bioinf.uni-leipzig.de/Software/tracker2/>.

## References

1. Elnitski, L., Jin, V.X., Farnham, P.J., Jones, S.J.M.: Locating mammalian transcription factor binding sites: A survey of computational and experimental techniques. *Genome Res.* **16** (2006) 1455–1464
2. Bailey, T.L., Williams, N., Misleh, C., Li, W.W.: MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Res.* **34** (2006) W369–W373
3. Blanchette, M., Tompa, M.: Discovery of regulatory elements by a computational method for phylogenetic footprinting. *Genome Res.* **12** (2002) 739–748
4. Liu, Y., Liu, X., Wei, L., Altman, R., Batzoglou, S.: Eukaryotic regulatory element conservation analysis and identification using comparative genomics. *Genome Res.* **14** (2004) 451458
5. Siddharthan, R., Siggia, E., van Nimwegen, E.: **PhyloGibbs**: A gibbs sampling motif finder that incorporates phylogeny. *PLoS Comput. Biol.* **1** (2005) e67
6. van Nimwegen, E.: Finding regulatory elements and regulatory motifs: a general probabilistic framework. *BMC Bioinformatics* **8 Suppl 6** (2007) S4
7. Gordân, R., Narlikar, L., Hartemink, A.J.: Finding regulatory DNA motifs using alignment-free evolutionary conservation information. *Nucleic Acids Res.* **38** (2010) e90
8. Margulies, E.H., Blanchette, M., Haussler, D., Green, E.D.: Identification and characterization of multi-species conserved sequences. *Genome Res.* **13** (2003) 2507–2518
9. Zhang, Z., Gerstein, M.: Of mice and men: phylogenetic footprinting aids the discovery of regulatory elements. *J. Biol.* **2** (2003) 11
10. Prohaska, S., Fried, C., Flamm, C., Wagner, G., Stadler, P.F.: Surveying phylogenetic footprints in large gene clusters: Applications to Hox cluster duplications. *Mol. Phyl. Evol.* **31** (2004) 581–604
11. Morgenstern, B., Stoye, J., Dress, A.W.M.: Consistent equivalence relations: a set-theoretical framework for multiple sequence alignments. Technical report, University of Bielefeld, FSPM (1999)
12. Corel, E., Pitschi, F., Morgenstern, B.: A *min-cut* algorithm for the consistency problem in multiple sequence alignment. *Bioinformatics* **26** (2010) 1015–1021
13. Morgenstern, B., Frech, K., Dress, A., Werner, T.: DIALIGN: finding local similarities by multiple sequence alignment. *Bioinformatics* **14**(3) (1998) 290–294
14. Tarjan, R.E.: Depth first search and linear graph algorithms. *SIAM J. Computing* **1** (1972) 146–160
15. Kahn, A.B.: Topological sorting of large networks. *Comm. ACM* **5** (1962) 558–562
16. Notredame, C., Higgins, D.G., Heringa, J.: T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.* **302** (2000) 205–217
17. Euler, R.: On a classification of independence systems. *Math. Methods Operations Res.* **27** (1983) 123–136
18. Helman, P., Moret, B.M.E., Shapiro, H.D.: An exact characterization of greedy structures. *SIAM J. Discrete Math.* **6** (1993)
19. Elias, I.: Settling the intractability of multiple alignment. *J. Comp. Biol.* **13** (2006) 1323–1339
20. Wilm, A., Mainz, I., Steger, G.: An enhanced RNA alignment benchmark for sequence alignment programs. *Algorithms Mol. Biol.* **1** (2006) 19