Local Protein Threading by Mixed Integer Programming

Guillaume Collet, Rumen Andonov, Nicola Yanev, Jean-François Gibrat

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Local Protein Threading by Mixed Integer Programming

Guillaume Collet — Rumen Andonov — Nicola Yanev — Jean-François Gibrat

N° 7122

Octobre 2009

Thème BIO

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7122--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
RENNES - BRETAGNE ATLANTIQUE

Local Protein Threading by Mixed Integer Programming

Guillaume Collet [*], Rumen Andonov [*] , Nicola Yanev [†‡] ,
Jean-Franois Gibrat [§]

Thème BIO — Systèmes biologiques
Équipes-Projets Symbiose

Rapport de recherche n° 7122 — Octobre 2009 — 17 pages

Abstract: During the last decade, significant progresses have been made in solving the Protein Threading Problem (PTP). However, all previous approaches to PTP only perform global sequence–structure alignment. This obvious limitation is in clear contrast with the "world of sequences", where local sequence-sequence alignments are widely used to find functionally important regions in families of proteins. This paper presents a novel approach to PTP which allows to align a part of a protein structure onto a protein sequence in order to detect local similarities. We show experimentally that such local sequence–structure alignments improve the quality of the prediction. Our approach is based on Mixed Integer Programming (MIP) which has been shown to be very successful in this domain. We describe five MIP models for local sequence–structure alignments, compare and analyze their performances by using ILOG CPLEX 10 solver on a benchmark of proteins.

Key-words: Mixed integer programming, combinatorial optimization, protein threading problem, protein structure alignment

[*] IRISA-Symbiose team, Campus de Beaulieu, 35042 Rennes Cedex, France

[†] Faculty of Mathematics and Informatics, Sofia University, blvd. James Bourchier 5,1164, Sofia, Bulgaria

[‡] Institute of Mathematics and Informatics, Bulgarian Academy of Sciences

[§] MIG - INRA, Domaine de Vilvert, 78350 Jouy en Josas Cedex, France

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu, 35042 Rennes Cedex
Téléphone : +33 2 99 84 71 00 — Télécopie : +33 2 99 84 71 71

Local Protein Threading by Mixed Integer Programming

Résumé : Au cours des dernières décennies, le problème de la reconnaissance de repliements des protéines ou *Protein Threading Problem* (PTP) a connu de grandes avancées. Néanmoins, toutes les méthodes développées ne proposent que des alignements séquence–structure globaux. Or, les alignements séquence–séquence locaux ont montrés qu'ils permettaient de détecter des régions fonctionnelles dans des familles de protéines. Ce rapport présente une nouvelle approche de la reconnaissance des repliements qui permet d'aligner une partie d'une structure de protéine avec une partie d'une séquence de protéine afin de détecter des similarités locales. Nous montrons que cette approche, basée sur la programmation mixte en nombre entiers (MIP), améliore la qualité de la reconnaissance. Nous avons modélisé les alignements séquence–structure locaux par cinq modèles MIP que nous comparons et analysons grâce au logiciel CPLEX 10.0 sur un jeu de test de protéines.

Mots-clés : Programmation mixte en nombre entiers, Optimisation combinatoire, Reconnaissance des repliements, Alignements séquence–structure de protéines

1 Introduction

To exploit the amount of new genomic data, the most important *in silico* methods are based on the concept of homology. The principle of homology-based analysis is to identify a homology relationship between a new protein and a protein whose function is known. Detecting homology relationships when the sequences are sufficiently similar is relatively straightforward. There exist efficient $O(N^2)$ algorithms based on dynamic programming techniques to align protein sequences [1, 2]. Suitable modifications of the fundamental algorithm enable the user to perform global, semi-global and local alignments (see Fig.1).

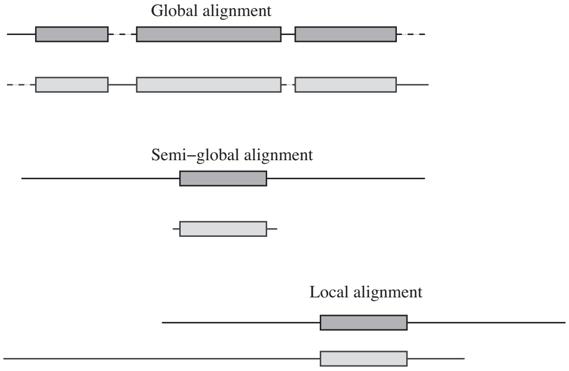


Figure 1: **Types of alignments.** When aligning protein sequences, it is important to be able to carry out different types of alignments according to the situation encountered. **Top:** When aligning proteins belonging to the same family, global alignments are used, i.e. gaps (illustrated here by dashed lines) at both ends of the alignments are penalized. **Middle:** On the other hand, when one looks for a protein domain in a longer sequence, semi-global alignments that allow the shorter sequence to be aligned (entirely) along the longer one are required. Gaps at both ends are not penalized in semi-global alignments. **Bottom:** The most general type of alignment is the local alignment, where only substrings of both sequences can be aligned (this might correspond to a common domain found in two proteins that are otherwise different).

For remote homologs, i.e. homologous proteins for which the sequences are no longer similar, sequence alignment methods fail. One, then, must resort to the good conservation of the 3-dimensional (3D) structures. In such a case, to figure out a homology relationship between 2 proteins, one aligns the sequence of the new protein with the 3D structures of known proteins. Such methods are called fold recognition methods or threading methods since this can be thought as "threading" the sequence through the 3D structure.

For a long time, threading methods using *non local parameters* (see section 2.1 for the definition of this term) suffered from the lack of a rigorous method capable of determining the sequence–structure alignment with the optimal score. They relied on heuristic techniques, e.g. stochastic techniques such as a Gibbs Monte Carlo [3]. R. Lathrop showed that, in the most general case, when variable length alignment gaps are allowed and pairwise amino acid interactions are considered in the score function (*i.e. non local parameters*), the problem of aligning a sequence onto a 3D structure is NP-hard [4]. The algorithm proposed by Lathrop & Smith was the first for finding the global alignment with the optimal score [5]. Since then, other methods have been developed that improved the efficiency of the global sequence–structure alignment algorithms [6, 7, 8, 9].

This study focusses on the local sequence–structure alignment problem. We propose an approach based on integer programming which expands upon our previous results [7, 9]. To the best of our knowledge, no attempt has been done previously for solving this problem. We show experimentally that such local sequence–structure alignments improve the quality of the prediction. This allows threading methods to cover the whole spectrum of alignment types needed to analyze homologous proteins.

2 Outline of the Protein Threading Problem

In this section we use classical notations and definitions concerning PTP as given in [5].

2.1 Definition of alignments

Query Sequence A query sequence is a string of length N over the 20-letter amino acid alphabet. This is the amino acid sequence of an unknown protein which must be aligned to structure templates from a given database.

Structure Template All current threading methods replace the 3D coordinates of the known structure by an abstract template description in terms of blocks, neighbor relationships, distances, environments, etc (see Fig 2). Blocks correspond to the most conserved parts of the structure, usually the secondary structure elements (SSEs : α-helices and β-strands). We consider that a structure template is an ordered set M of m segments or blocks. Block k has a fixed length of L_k amino acids.

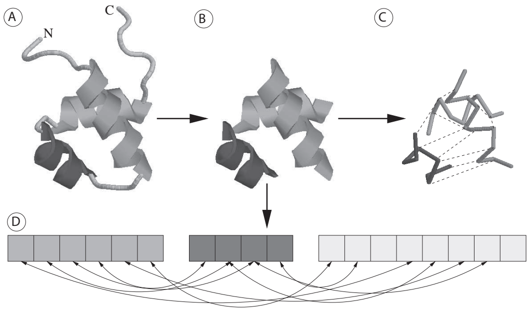


Figure 2: **Proteins as contact maps.** A) “ribbon” representation of a protein with 3 α helices, B) only the secondary structure elements, here the 3 helices, are taken into consideration, C) dotted lines represent interactions between positions in the helices, i.e., residues in contact, D) contact map representation of the protein, double headed arrows correspond to the dotted line in C. In the blocks respectively 6, 4 and 8 amino acids can be aligned.

Let $I \subseteq \{(k,l) \mid 1 \leq k < l \leq m\}$ be the set of blocks interactions. The graph $G = (M, I)$ with a set of vertices M ($|M| = m$) and a set of edges I, is called the generalized contact map graph (see Fig 3).

Alignments The alignment of a sequence with a 3D structure can be described as positioning the blocks along the sequence. Such an alignment is called feasible if blocks preserve their original order and do not overlap. An alignment is completely determined by the starting positions of all blocks along the sequence. In fact, in the classical PTP, it is more convenient to use relative positions instead of absolute positions. If block k starts at the jth query character, its relative position is $r_k = j - \sum_{i=1}^{k-1} L_i$. In this way the possible (relative) positions of each segment are between 1 and $n = N - \sum_{k=1}^{m} L_k + 1$. The set of possible alignments (feasible threadings) is $\mathcal{T} = \{(r_1, \ldots, r_m) \mid 1 \leq r_1 \leq \cdots \leq r_m \leq n\}$.

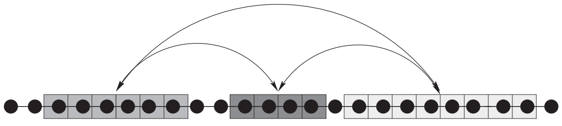


Figure 3: **Example of alignment.** Alignment of query sequence of length $N = 24$ and template containing $m = 3$ blocks of lengths $L_1 = 6$, $L_2 = 4$ and $L_3 = 8$. An arrow is drawn between 2 blocks if there is at least one position in each block which are in interaction in the 3D structure. The corresponding generalized contact map graph is given by $G = (M, I)$, where $M = 1, 2, 3$ and $I = \{(1,2), (1,3), (2,3)\}$

The cardinality of this set (the search space size of PTP) is given by $|\mathcal{T}| = \binom{m+n-1}{m}$, which is a huge number even for small instances (for example, if $m = 20$ and $n = 100$ then $|\mathcal{T}| \approx 2.5 \times 10^{22}$).

With this definition of the PTP, gaps are not allowed within blocks. They are confined to loops joining the blocks.

2.2 Network flow formulation

In order to develop an appropriate mathematical model, PTP has been reformulated in [7, 10] as a network optimization problem. Let $G = (V, E)$ be a m-partite graph with vertex set V and edge set E. The vertex set V is organized in $n \times m$ grid as shown in Fig 4. The vertex (i, k) (where i stands for row, while k stands for column) corresponds to the relative position of the block k along the sequence. The graph G is called an *alignment* graph. To preserve blocks order, each edge $((i,k),(j,l)) \in E$ satisfies the condition $i < j$ and $k < l$ (i.e. all edges are southwest-northeast oriented). For this reason, the alignment graph can be also thought as a directed graph, and since we search for a southwest-northeast oriented path in this graph, there is no ambiguity when we sometimes indicate the edges of a given vertex as input or output edges.

A *feasible path* F in the alignment graph G is a set of **exactly** m vertices $F = \{v_{i_1,1}, v_{i_2,2}, \ldots, v_{i_m,m}\}$ such that $i_k \leq i_{k+1}, \forall k,$ and $1 \leq k < m$ (i.e. a non-decreasing set of vertices). It is easy to see the one-to-one correspondence between the set of feasible threadings (or matchings) and the set of feasible

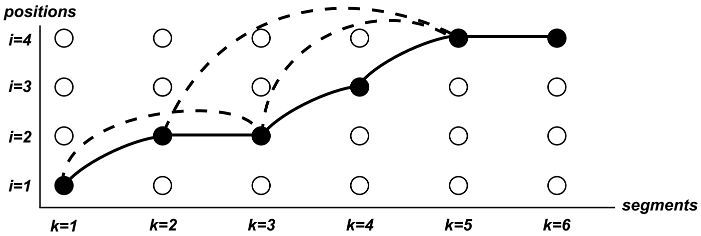


Figure 4: **Example of alignment graph.** The thick line path corresponds to the threading in which the relative positions of blocks are 1,2,2,3,4,4. Dashed line edges represent non local interactions where the set of block interactions is $I = \{(1,3),(2,5),(3,5)\}$.

paths in G.

A cost C_{ik} is associated to each vertex (i,k) as defined by the scoring function. C_{ik} is a local score that depends only on the location of the block k at position i along the sequence. To each edge $((i,k),(j,l))$ we associate a cost C_{ikjl} as defined by the scoring function. C_{ikjl} is a non local score that depends on the location of blocks k and l on positions i and j along the sequence. We say that an edge is *activated* by a feasible path if both ends are on the path. The subgraph induced by the activated edges of a feasible path is called an *augmented path*. The main result from [7] claims that solving PTP is equivalent to finding the optimal (in our case the longest) augmented path in the alignment graph G.

2.3 Mixed Integer Programming Formulation

Let y_{ik} be binary variables associated with vertices in the previous network. Then y_{ik} is 1 if block k is at position i and 0 otherwise (vertex (i,k) is activated or not). To take into account the interaction costs, we introduce a second set of variables $0 \leq z_{ikjl} \leq 1$, with $(k,l) \in I$ and $1 \leq i \leq j \leq n$. The variable z_{ikjl} is set to 1 if the corresponding edge is activated. Finding the optimal augmented path in graph G (i.e. solving PTP) is then equivalent to maximizing the following function :

$$\sum_{k=1}^{m}\sum_{i=1}^{n} C_{ik} y_{ik} + \sum_{((i,k),(j,l))\in E} C_{ikjl} z_{ikjl} \qquad (1)$$

This objective function is subject to the following constraints:

$$\sum_{i=1}^{n} y_{ik} = 1 \qquad k \in [1,m] \tag{2}$$

$$\sum_{j=1}^{i} y_{j(k+1)} - \sum_{j=1}^{i} y_{jk} \leq 0 \qquad k \in [1,m-1],\ i \in [1,n-1] \tag{3}$$

$$y_{ik} \in \{0,1\} \qquad k \in [1,m],\ i \in [1,n] \tag{4}$$

$$y_{jl} = \sum_{i=1}^{j} z_{ikjl} \qquad (k,l) \in I,\ j \in [1,n] \tag{5}$$

$$y_{ik} = \sum_{j=i}^{n} z_{ikjl} \qquad (k,l) \in I,\ i \in [1,n] \tag{6}$$

$$0 \leq z_{ikjl} \leq 1 \qquad ((i,k),(j,l)) \in E \tag{7}$$

This model, known as MYZ and first introduced in [7], has been shown to outperform the MIP model used in the RAPTOR package [6] for large PTP instances. Both models (MYZ and RAPTOR) are solved by first relaxing constraints (4), i.e., letting y_{ik} be real variables such that $0 \leq y_{ik} \leq 1$. This allows the use of a Linear Programming (LP) technique. The solution of the LP technique is then used as a lower bound in a subsequent branch & bound algorithm that finds the integer solution.

3 Local alignments : towards better PTP models

Constraints (2) from the above model force each block (i.e SSE) to be aligned with the query sequence. However, proteins belonging to the same family do not always have the same number of SSEs. Forcing all blocks to be aligned with the sequence results in spurious alignments with bad scores. Such alignments prevent the method to detect remote homologs. To tackle this issue, we develop new models that allow a block to be omitted (not aligned) if its score with all other interacting blocks is negative. Such an approach realizes *local* sequence–structure alignments.

Towards this goal, we slightly modify the definition of a feasible path by accepting that **any** non-decreasing set of vertices is a feasible path of the alignment graph G (i.e. the cardinality of a feasible path could be now less than m). Obviously, there is a one-to-one correspondence between the set of local alignments and the set of feasible paths in G.

In a local alignment, each block can be aligned with the whole sequence because the number of omitted blocks is not known in advance. It follows that relative positions are no longer meaningful. Thus, each block k takes $n_k = N - L_k + 1$ absolute positions along the sequence. This results in columns having different heights in the network flow formulation and leads to the need of using an "offset" to move from one column to the next (this offset is illustrated in Fig 5).

In order to implement the mechanism for omitting blocks during an alignment, we propose two schemes : (i) we modify constraints (2) to allow the sum

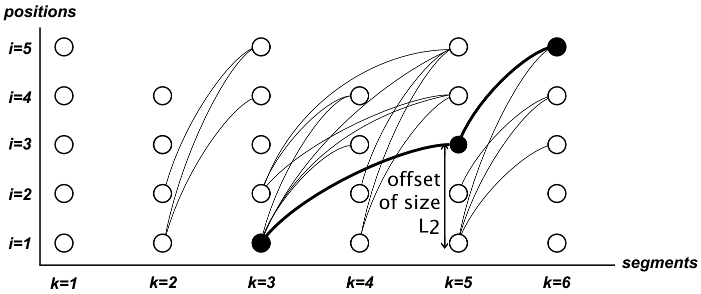


Figure 5: **Example of flexible alignment graph.** The path in thick lines corresponds to the threading in which: blocks 1, 2, and 4 are omitted, and blocks 3, 5, and 6 are on positions 1, 3 and 5. All edges in E are represented. Because of absolute positions, notice that a offset of size L_k is needed to go from column k to the followings (blocks sizes are : 2,3,2,3,2,2).

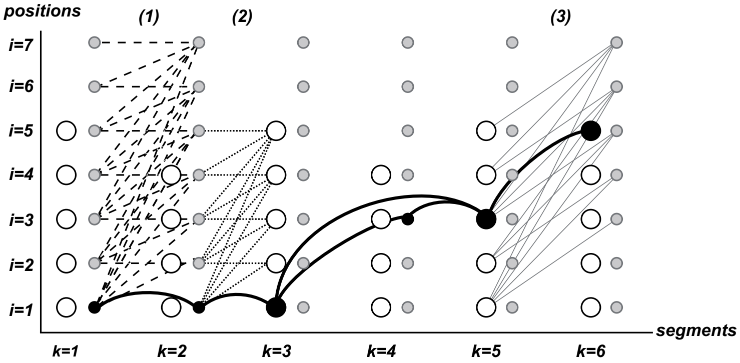


Figure 6: **Example of a flexible alignment graph with dummy vertices.** Dummy vertices are the small gray circles. This illustrates the same alignment as in figure 5 but not all edges are represented for clarity. (1) dashed lines represent D to D paths, (2) dotted lines represent D to R paths, and (3) gray lines represent R to D paths. Notice that an offset of size L_k is needed to move from a real vertex ik to the next (be it dummy or real). No offset is needed when moving from a dummy vertex.

of a column to be zero, (ii) $N + 1$ dummy vertices are added in every column. If a dummy vertex is activated in a column, the corresponding block is omitted. In the second scheme, two types of vertices exist : real (R) and dummy (D). This generates 4 types of subpaths: R to R , R to D, D to R, and D to D (as shown in fig 6).

According to the above two schemes, we implemented 5 models : Compact model (CM) for the first one and Extended models (EM1, EM2, EM3, EM4) that make use of the dummy nodes. These models are described below.

3.1 Mathematical models

In the following models, we use N, m, L_k, I, C_{ik}, C_{ikjl}, y_{ik}, z_{ikjl} notations from section 2.3 and the following :

- $n_k = N - L_k + 1$ is the number of possible positions of block k along the query sequence.
- $E = \{((i,k),(j,l)) | (k,l) \in I,\ 1 \le i \le n_k,\ i + L_k \le j \le n_l\}$ is the set of edges.

All models share the same objective function:

$$\max \sum_{k=1}^{m} \sum_{i=1}^{n_k} C_{ik} y_{ik} + \sum_{((i,k),(j,l)) \in E} C_{ikjl} z_{ikjl} \tag{8}$$

3.2 Compact model (CM)

The objective function (8) is subject to the following constraints:

$$y_{ik} \in \{0,1\}, \qquad k \in M,\ i \in [1, n_k] \tag{9}$$

$$0 \le z_{ikjl} \le 1, \qquad ((i,k),(j,l)) \in E \tag{10}$$

$$\sum_{i=1}^{n_k} y_{ik} \le 1, \qquad k \in M \tag{11}$$

$$\sum_{j=i+L_k}^{n_l} z_{ikjl} - y_{ik} \le 0, \qquad (k,l) \in I,\ i \in [1, n_k] \tag{12}$$

$$\sum_{i=1}^{j-L_k} z_{ikjl} - y_{jl} \le 0, \qquad (k,l) \in I,\ j \in [1, n_l] \tag{13}$$

$$y_{ik} + \sum_{j=1}^{min(n_l, i+L_k-1)} y_{jl} \le 1, \qquad 1 \le k \le l \le m,\ i \in [1, n_k] \tag{14}$$

$$\sum_{i=1}^{n_k} y_{ik} + \sum_{i=1}^{n_l} y_{il} - \sum_{j=L_k+1}^{n_l} \sum_{i=1}^{j-L_k} z_{ikjl} \le 1, \qquad (k,l) \in I \tag{15}$$

Constraints (11) correspond to constraints (2) in MYZ model and they impose a block to have one position or zero (i.e to be omitted). Constraints (12) and (13) force node activation if an outpout (resp. input) edge is activated. Constraints (14) keep blocks order and impose that they do not overlap. Finaly, constraints (15) impose the activation of an edge if its ends are activated.

3.3 Extended model 1 (EM1)

Based on the compact model, each extended model is an attempt to find a more similar model with MYZ [7] which was shown to be efficient for global alignments. EM1 extends the compact model by adding dummy positions for each column. An active dummy position in a column means that the corresponding block is deleted.

- $d_{ik} \in \{0,1\}$ are dummy variables added for all extended models.

The objective function (8) is subject to constraints (9), (10), (12), (13) and (15) from CM. We add the following constraints:

$$d_{ik} \in \{0,1\}, \quad k \in M,\ i \in [1, N+1] \tag{16}$$

$$\sum_{i=1}^{n_k} y_{ik} + \sum_{i=1}^{N+1} d_{ik} = 1, \quad k \in M \tag{17}$$

$$\sum_{i=1}^{j} d_{ik} + \sum_{i=1}^{min(j,n_k)} y_{ik} - \sum_{i=1}^{j} d_{ik-1} - \sum_{i=1}^{j-L_{k-1}} y_{ik-1} \leq 0, \quad k \in [2,m],\ j \in N+1 \tag{18}$$

Constraints (17) replace constraints (11) with the use of dummy nodes. Our goal was to transform these inequalities in equalities. The use of dummy nodes also change constraints (14) into constraints (18). One can notice that, $m-1$, the number of constraints (18) is smaller than, $\frac{m*(m-1)}{2}$, the number of constraints (14).

3.4 Extended model 2 (EM2)

This model extends $EM1$ by adding edges between dummy vertices and original vertices (or real vertices). The aim of this model is to transform constraints (12) and (13) in equalities.

- $rd_{ikjl} \in \mathbb{R},\ (k,l) \in I,\ (i,j) \in RD$ are dummy variables representing an edge from a dummy vertex to a real vertex(with $RD = \{(i,j)\ |(k,l) \in I\ 1 \leq i \leq n_k,\ i + L_k \leq j \leq N+1\}$).
- $dr_{ikjl} \in \mathbb{R},\ (k,l) \in I,\ (i,j) \in DR$ are dummy variables representing an edge from a real vertex to a dummy vertex(with $DR = \{(i,j)\ |(k,l) \in I,\ 1 \leq i \leq j \leq n_l\}$).

The objective function (8) is subject to constraints (9), (10), (15), (16), (17), (18) and to the following constraints:

$$0 \leq rr_{ikjl} \leq 1, \qquad (k,l) \in I,\ (i,j) \in RD \tag{19}$$

$$0 \leq dr_{ikjl} \leq 1, \qquad (k,l) \in I,\ (i,j) \in DR \tag{20}$$

$$\sum_{j=i+L_k}^{n_l} z_{ikjl} + \sum_{j=i+L_k}^{N+1} rd_{ikjl} - y_{ik} = 0, \qquad (k,l) \in I,\ i \in [1, n_k] \tag{21}$$

$$\sum_{i=1}^{j-L_k} z_{ikjl} + \sum_{i=1}^{j} dr_{ikjl} - y_{jl} = 0, \qquad (k,l) \in I,\ j \in [1, n_l] \tag{22}$$

Constraints (12) and (13) are replaced by constraints (21) and (22) which are equalities.

3.5 Extended model 3 (EM3)

This model extends $EM2$ by adding edges between dummy vertices.

- $dd_{ikjl} \in \mathbb{R},\ (k,l) \in I,\ (i,j) \in DD$ are dummy variables representing an edge from a dummy vertex to a dummy vertex(with $DD = \{(i,j) \mid 1 \leq i \leq j \leq N+1\}$).

The goal of this model is to have equality constraints like (21), (22) applied to dummy vertices. The objective function (8) is subject to constraints (9), (10), (16), (18), (19), (20), (21), (22) and to the following:

$$0 \leq dd_{ikjl} \leq 1, \qquad (k,l) \in I,\ (i,j) \in DD \tag{23}$$

$$\sum_{j=i}^{N+1} dd_{ijkl} + \sum_{j=i}^{n_l} dr_{ijkl} - d_{ik} = 0, \qquad (k,l) \in I,\ i \in [1, N+1] \tag{24}$$

$$\sum_{i=1}^{j} dd_{ijkl} + \sum_{i=1}^{j-L_k} rd_{ijkl} - d_{jl} = 0, \qquad (k,l) \in I,\ j \in [1, N+1] \tag{25}$$

Constraints (24) (resp (25)) impose the activation of one and only one edge going out (resp. in) a dummy node. These constraints replace activation constraints (15).

3.6 Extended model 4 (EM4)

This model is based on $EM2$. In this model, we try to delete constraints (15). The objective function (8) is subject to constraints (9), (10), (16), (17), (18), (19) , (20) , (21), (22) and to the following :

$$\sum_{j=i}^{n_l} dr_{ijkl} - d_{ik} \leq 0, \qquad (k,l) \in I,\ i \in [1, N+1] \tag{26}$$

$$\sum_{i=1}^{j-L_k} rd_{ijkl} - d_{jl} \leq 0, \qquad (k,l) \in I,\ j \in [1, N+1] \tag{27}$$

In this model we replace constraints (15) by constraints (26) and (27).

4 Results

4.1 MIP models comparison

4.1.1 Benchmark

These five MIP models have been implemented with Ilog CPLEX 10.0 Library in our protein threading package FROST [11]. We also randomly created a benchmark of 100 alignments to compare their performances. Each alignment can be solved in less than 1000 seconds with CPLEX MIP solver for all models. Alignments have been processed on a cluster of computers with 2.5GHz AMD Opteron biprocessors and 4MB of memory. Comparisons of models have been carried out using four measures : the number of variables, the number of constraints, the relative gap, and the computation times.

4.1.2 Number of variables and constraints

CM obviously has the smallest number of variables because there is only real nodes and real edges. EM1 is larger than CM because dummy nodes are added. Then EM2 and EM3 follow due to the addition of edges. EM2 and EM4 have the same number of variables. In summary, based on the number of variables, the ranking of the models is: CM < EM1 < EM2 = EM4 < EM3.

EM1 is the model with the smallest number of constraints. Although EM1 has more variables, dummy nodes create less constraints (18) compared to constraints (14) in CM for the same effect. EM1 and EM2 have similar structure, but EM1 has also less constraints than EM2. Finally, EM3 and EM4 have the largest number of constraints, about twice the number of constraints in EM1.

4.1.3 Relative Gap

The relative gap (RG) is the relative difference between the solution of the relaxed problem (LP) and the optimal solution (OPT): $RG = \frac{LP-OPT}{OPT}$. RG is a good indicator of the efficiency of the model since the smaller RG, the easier for the branch & bound algorithm to find the solution.

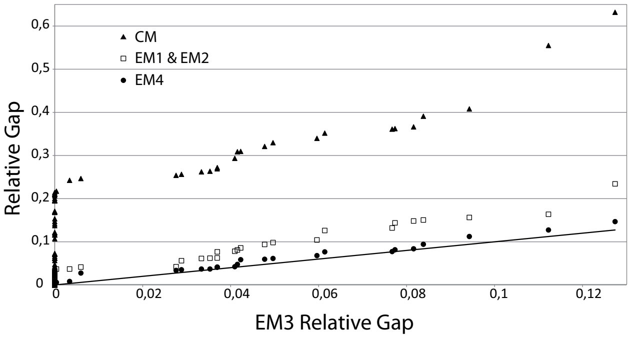


Figure 7: **Relative gap of CM, EM1, EM2 and EM4 compared to relative gap of EM3 for 100 alignments**. EM3 relative gap values are on the thick line. One can notice that EM3 and EM4 are the tightest models, and that EM1 always give an equal relative gaps with EM2. Overall, the tightest model is EM3 because it gives a tighter (or equal) relative gap than EM4 for all instances.

Figure 7 shows that EM3 and EM4 give tighter relative gaps compared to other models. In fact, EM3 always gives a tighter relative gap than any other model. Moreover, EM3 finds the optimal solution by solving the relaxed problem for 77% of alignments. This rate if greater than for CM (37%), EM1 and EM2(64%), and EM4(75%).

4.1.4 Computation Times

Figure 8 presents statistics on computation times for the five models. EM1 and EM2 seem to be the fastest models compared to CM, EM3 and EM4.

Actually, EM1 computation times are always smaller than the others.

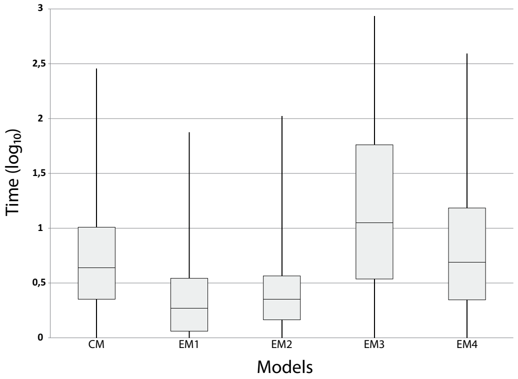


Figure 8: **Statistics on computation times of each model for 100 alignments.** Times are in seconds on a logarithmic scale. A box represents 50% of instances, the thick line inside is the median. The ends of lines represent maximum and minimum values. EM1 and EM2 are the fastest models.

4.2 Biological considerations

In order to measure the quality of local alignments, we need accurate alignments between proteins. These alignments are provided by structural alignments of proteins. We used the TopMatch server [12, 13] for alignments of two proteins and the Mammoth server [14] for multiple structure alignments.

4.2.1 Local similarities

A first improvement of a local alignment approach compared to a global alignment approach is to align a long template with a small sequence. Such alignments are not allowed by global alignments because all blocks must be aligned. For example, the TopMatch alignment of proteins 9gaaA (length 180) and 1apyB (length 141) show that 1apyB is a sub domain of 9gaaA (85% structure similarity and 40% sequence identity). Thus, a global alignment between template 9gaaA and sequence 1apyB is impossible. However, with a local alignment, a part of 9gaaA can be deleted and the subdomain can be aligned with 1apyB as illustrated in figure (9).

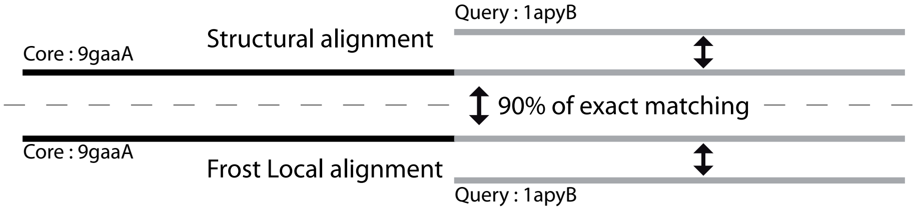
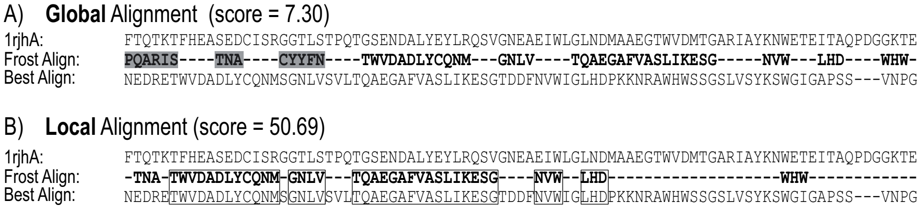


Figure 9: **Alignment of proteins 9gaaA and 1apyB by TopMatch and FROST.** Although 9gaaA is larger than 1apyB, a local alignment is possible. Moreover, FROST local approach produces a coherent alignment with the TopMatch structural alignment (90% of exact matching).

4.2.2 Better alignments

In section 3 we presented the problem of having proteins with similar structures but not the same number of SSEs. Such a case is illustrated in figure (10.A). Template 1qddA is aligned with sequence 1rjhA. These two proteins share a common domain (76% structure similarity and 21% sequence identity) but the first three blocks of 1qddA (in gray) are not in this domain. Because the global alignment approach imposes these three blocks to be aligned with the sequence, it results in a decrease of the overall score and in a spurious alignment.



```
A)  Global Alignment  (score = 7.30)
1rjhA:         FTQTKTFHEASEDCISRGGTLSTPQTGSENDALYEYLRQSVGNEAEIWLGLNDMAAEGTWVDMTGARIAYKNWETEITAQPDGGKTE
Frost Align:   PQARIS----TNA----CYYFN----TWVDADLYCQNM---GNLV----TQAEGAFVASLIKESG-----NVW---LHD----WHW-
Best Align:    NEDRETWVDADLYCQNMSGNLVSVLTQAEGAFVASLIKESGTDDFNVWIGLHDPKKNRAWHWSSGSLVSYKSWGIGAPSS---VNPG

B)  Local Alignment (score = 50.69)
1rjhA:         FTQTKTFHEASEDCISRGGTLSTPQTGSENDALYEYLRQSVGNEAEIWLGLNDMAAEGTWVDMTGARIAYKNWETEITAQPDGGKTE
Frost Align:   -TNA-TWVDADLYCQNM-GNLV---TQAEGAFVASLIKESG----NVW--LHD-------------------WHW-----------
Best Align:    NEDRETWVDADLYCQNMSGNLVSVLTQAEGAFVASLIKESGTDDFNVWIGLHDPKKNRAWHWSSGSLVSYKSWGIGAPSS---VNPG
```

Figure 10: **Frost Global and Local Alignments of core 1qddA with query 1rjhA.** "Best Align" is the alignment obtained by Mammoth structural alignment tool [14]. Frost alignment is in bold. **A)** Gray blocks are aligned although they are not in the structural domain shared by 1qddA and 1rjhA. This results in an alignment which is totaly different from the best alignment. **B)** Because two gray blocks have been omitted, local alignment approach gives a better score and an alignment more coherent with the best alignment (frames correspond to exact matchings between Frost and Mammoth alignments).

A local alignment permits to omit blocks when needed (i.e. based on score function) as illustrated in figure (10.B). The deletion of two gray blocks results in a better score and an alignment which is more consistent with the best alignment found by Mammoth.

5 Discussion

This paper deals with the description and the comparison of five MIP models for local alignments. We do not explore the sensitivity and specitivity of the approach from a biological point of view. Such statistics about recognition rates and quality of alignments need a huge amount of alignments and are beyond the scope of this paper.

Based on the compact model, we develop four extended models. The aim of these extended models is to find a similar model to MYZ. MYZ has been proven to be very efficient and to give an optimal solution for 95% of cases by solving the relaxed problem. Our assumption is that a close model to it would be as efficient for local alignments as MYZ is for global alignments. EM3 is the closest model to MYZ because it uses very similar equality constraints. By always giving the tighter relative gap, EM3 confirms our hypothesis.

Aligning difficult instances (i.e. long sequences with long templates) with the CPLEX solver is too slow for a massive usage. For this reason, we need to develop a fast dedicated algorithm based on one of our models. Usually, we use the model with the tightest relative gap but in this case, the tightest model (EM3) is also the slowest (section 4.1.4) and the largest (section 4.1.2). Thus, choosing the best model for the development of a dedicated solver is still an open research problem on which we are currently working.

Although the local alignment approach seems to behave as we expected, it completely relies on the score function. Indeed, a block is omitted if its score decreases the overall alignment score, i.e. its score is negative. But we do not know if 0 is the right boundary to decide to omit a block. We are currently looking for an automatic learning method to evaluate the correct boundary for a given score function.

6 Conclusion

This paper describes a method allowing local alignments of protein sequences and structures. As far as we know, this is the first attempt to propose alignments that give the possibility of omitting parts of the 3D structure that might not be conserved in remote homolog proteins. Results show that our algorithm is indeed capable of omitting blocks when required. The accuracy of the resulting alignment strongly depends on the quality of the score function. It is likely that score functions adapted to the local alignments will contribute for the success of the approach.

Local alignments have been modelled and tested with the CPLEX 10 solver. This is a general purpose solver for integer programming problems that is very convenient for rapidly testing new ideas. However, our computation experiments show that it is not enough rapid for our particular problem. We show that EM1 and EM3 are respectively the fastest and the tightest model. The next step is to choose one of these models to create a dedicated algorithm in order to solve difficult instances and to evaluate the recognition rate of the local alignment approach.

7 Acknowledgements

This work is supported by ANR project PROTEUS “ANR-06-CIS6-008”. N. Yanev is supported by DVU/01/197 project from SWU “Neofit Rilski” and by DO 02359/2008 grant. All computations were done on the Biogenouest bioinformatics platform[1].

References

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

[2] T.F. Smith and M.S. Waterman. Identification of Common Molecular subsequences. *J. Mol. Bio.*, 147:195-197 (1981).

[1] http://genouest.org

[3] C.E Lawrence, S.F Altschul, M.S Boguski, J.S. Liu,A.N. Neuwald and J. Wootton. Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science*, 262:208-14 (1993).

[4] R.H. Lathrop.The protein threading problem with sequence amino acid interaction preferences is NP-complete *Protein Engineering*, 255:1059-1068 (1994).

[5] R.H. Lathrop and T.F. Smith. Global optimum protein threading with gapped alignment and empirical pair potentials. *Journal of Molecular Biology*, 255:641-665 (1996).

[6] J. Xu, et al. RAPTOR: optimal protein threading by linear programming. *Journal of Bioinformatics and Computational Biology*, 1(1):95-118 (2003).

[7] R. Andonov, S. Balev et N. Yanev. Protein Threading Problem: From Mathematical Models to Parallel Implementations. *INFORMS Journal on Computing*, 16(4):393:405 (2004).

[8] N. Yanev, P. Veber, R. Andonov and S. Balev. Lagrangian approaches for a class of matching problems in computational biology. *Comput. Math. Appl.*, 55:1054-1067 (2008).

[9] R. Andonov, et al. Recent Advances in Solving the Protein Threading Problem. *Grids for Bioinformatics and Computational Biology*, E-G. Talbi and A. Zomaya, editors. Chapter 14, pages 325-356. Wiley-Interscience (2007).

[10] N. Yanev and R. Andonov. Parallel Divide&Conquer Approach for the Protein Threading Problem. *Concurrency and Computation: Practice and Experience*, 16:961-974 (2004).

[11] A. Marin and J.Pothier and K. Zimmermann and J-F. Gibrat. FROST: A Filter Based Fold Recognition Method. *Proteins*, 49(4):493-509 (2002).

[12] M.J. Sippl and M. Wiederstein. A note on difficult structure alignment problems. *Bioinformatics*, 24:426-427 (2008).

[13] M. Sippl. On distance and similarity in fold space. *Bioinformatics*, 24:872-873 (2008).

[14] D. Lupyan, A. Leo-Macias and A.R. Ortiz. A new progressive-iterative algorithm for multiple structure alignment. *Bioinformatics*, 21(15):3255-3263 (2005).

Contents

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399