

Complexity analysis of finding optimal assignments' problem in local descriptor-based approach to protein structure prediction.

Michał Drabikowski* Jerzy Tiuryn*

8th September 2004

1 Introduction

It is a common knowledge that protein function depends on its structural conformation. Moreover, the structure of a protein is determined only by its sequence of amino acids. Among thousands of proteins that are sequenced each day unfortunately only a small fraction of their structures can be experimentally solved because of technology and time limitations. For this reason, predicting the three-dimensional structure of a protein from its amino acid sequence is one of the most important (and still open) problems of computational biology.

Among various approaches to the problem of protein structure prediction the so-called fragment-based methods have probably the most impressive record of successes (one of the precursors is David Baker ([1, 2, 3, 4, 5])). The main common idea of these methods is to gather information about properties of structurally similar backbone segments from various proteins, and use this information to assign a probable 3D shape to parts of protein. The approach, whose part we are going to focus on in this paper, is a kind of an extension of the aforementioned methods. The idea of this approach (called *local descriptor-based approach to protein structure prediction*) was developed by Krzysztof Fidelis ([6]). It can be divided into five steps presented below:

- (1) Constructing the database of *descriptors* defining local space environment for each amino acid in each protein with a known structure. For any residue in a protein a set of segments is defined: each segment is a 3D fragment of the protein's backbone (containing a minimum of five residues) situated close to considered residue in space. This set of segments (by a segment we understand both a shape and a piece of sequence) is named a descriptor. Distances between

*Institute of Informatics, Warsaw University

residues within each descriptor (even between residues from different segments) are known but it isn't relevant how far on the sequence different segments appear.

- (2) Clustering descriptors into *groups* according to the function of descriptors' structure similarity. Descriptors within each group must be aligned. For each group of structurally similar descriptors a representative descriptor which represents the shape of the whole group is defined. It is possible to think about groups as extended descriptors: each segment of such an extended descriptor is related to a matrix of amino acids.
- (3) *Signal extracting* may be regarded as finding regularities in sequences of descriptors within each group explaining why these descriptors have a similar shape. This leads to defining a signal function that is specified for any pair of extended descriptors containing the same number of segments such that the corresponding segments have the same length. The function returns a real number measuring sequence similarity between those two descriptors.
- (4) For a given a query sequence s with an unknown structure (or a set of aligned sequences, if we can find proteins of high homology to the query sequence) and a group g , we consider the set of all possible associations of all segments belonging to this group with unoverlapping fragments of s (all distances between fragments are possible). Each such association is called an *assignment* of g to s . For each possible assignment of g to s , the signal function gives a score which expresses how well the extended descriptor related to group g suits to extended descriptor related to fragments s . For each group the best (in sense of signal function) assignments to the query sequence s are chosen.
- (5) A set of the best groups' assignments to the query sequence is the input for this step. In the process of *assembling*, as large as possible subset of structurally non-conflicting assignments (two groups' assignments are non-conflicting if they are situated in different places in the query sequence, or are assigned to the same place and representative descriptors have a very similar shape) is defined. It gives a set of local structures which after additional evaluations produces a global protein structure prediction.

The approach presented briefly above is currently being investigated. Leaving the first three steps to the rest of the group working on this approach, we focus on the last two steps. In this paper we are going to present results corresponding to the fourth step only. An input for the problem we consider, is a group g (groups are defined in step 2), signal function ξ (signal is defined in step 3) and a query sequence s . The idea is to find a set of the best possible assignments of g to s . The problem is computationally complex so in order to solve it in biological applications, we propose a certain reasonable restriction of signals being additive. Two main results are presented. The first result is a proof of NP-completeness of this restricted

problem. The second result is an algorithm based on dynamic programming solving this problem very efficiently for biological data.

The structure of the paper is organized as follows. In the next part basic definitions are introduced. We define the general problem of finding the best assignments and propose its restricted variant. In the third part we prove that this restricted problem is NP-complete. In the fourth part we propose an algorithm solving this problem. The last part contains a short discussion of obtained results.

2 Basic definitions and problem statement

As we mentioned before, each group of descriptor sequences may be treated as an extended descriptor. Each element of each segment of such a descriptor is a sequence of amino acids. Analogously, a query sequence (with all aligned sequences of a high homology proteins) is a segment containing a sequence of amino acids in each position. This observation shows, that a segment of a group and a query sequence are objects of the same type.

Let \mathcal{W} be a set of all sequences shorter than a certain constant c over a set of 20 amino acids. Each string over set \mathcal{W} is called a **segment**. Each string over a set of all segments is called a **descriptor**.

Let $d = s_1s_2\dots s_k$ be a descriptor. A **type** of a descriptor d (denoted by $type(d)$) is a sequence u_1, u_2, \dots, u_k of integers such that $u_i = |s_i|$ for $i = 1, 2, \dots, k$. A **k-marking** for a segment s is a sequence of different positive integers l_1, l_2, \dots, l_k such that $l_i \leq |s|$ for $i = 1, 2, \dots, k$.

Any substring of a segment may be defined by two integers: a starting position in a segment and a length of the substring. Analogously, any set of not overlapping substrings may be defined by a k-marking (representing a vector of starting positions) and a type (representing vector of associated lengths). From the other hand not any pair of a k-marking and a tape produces set of not overlapping substrings. This observation is a starting-point to the following definition.

Definition 1 A type $\underline{u} = \langle u_1, u_2, \dots, u_k \rangle$ is said to be **admissible** for a k-marking $\underline{l} = \langle l_1, l_2, \dots, l_k \rangle$ for segment s , if the following conditions are satisfied:

- $\forall 1 \leq i \leq k (1 \leq l_i \leq |s| - u_i + 1)$,
- $\forall 1 \leq i < j \leq k (l_i + u_i \leq l_j \vee l_j + u_j \leq l_i)$.

If \underline{l} is a k-marking for $s = w_1w_2\dots w_n$ and type \underline{u} is admissible for \underline{l} and s , then by $s_{\underline{l}, \underline{u}}$ we denote a descriptor $s_1s_2\dots s_k$ of type \underline{u} such that $s_i = w_{l_i}w_{l_i+1}\dots w_{l_i+u_i-1}$ for $i = 1, 2, \dots, k$. In other words $s_{\underline{l}, \underline{u}}$ denotes a descriptor cut out of a segment s by a k-marking \underline{l} and a type \underline{u} .

Definition 2 Let d be a descriptor, let s be a segment. Let $\mathcal{P}_{d,s}$ be a set of k-markings \underline{l} for segment s such that $type(d)$ is admissible for \underline{l} and s . Each element of set $\mathcal{P}_{d,s}$ is called an **assignment** of d to s .

Let $\overline{\mathcal{D}^2} = \{\langle d_1, d_2 \rangle : d_1, d_2 \in (\mathcal{W}^*)^* \wedge \text{type}(d_1) = \text{type}(d_2)\}$. **Signal** is a function $\xi: \overline{\mathcal{D}^2} \rightarrow R$ that defines the similarity of two descriptors. For any assignment of a given descriptor to a given segment signal function defines a score of this assignment. The problem of finding the best assignments (in sense of this score) is defined below.

Definition 3 Let $r \in R$ be an arbitrary constant. For a given descriptor d of type \underline{u} , a segment s , a signal $\xi \in R^{\overline{\mathcal{D}^2}}$ and a positive integer m , the problem of **optimal assignments** is defined as follows: find a maximal set $P \subseteq \mathcal{P}_{d,s}$ that fulfills the following conditions:

- (1) $\forall \underline{l} \in P \ \xi(d, s_{\underline{l}, \underline{u}}) \geq r$,
- (2) $\forall \underline{l} \in P \ \forall \underline{l}' \in (\mathcal{P}_{d,s} - P) \ \xi(d, s_{\underline{l}, \underline{u}}) > \xi(d, s_{\underline{l}', \underline{u}})$,
- (3) $\{\xi(d, d') : \exists \underline{l} \in \mathcal{P}_{d,s} \ d' = s_{\underline{l}, \underline{u}}\}$ has at most m elements.

Decision problem of maximal assignment is defined as follows: is there an assignment $\underline{l} \in \mathcal{P}_{d,s}$ such that $\xi(d, s_{\underline{l}, \underline{u}}) \geq r$?

Obviously the problem of maximal assignment may be easily reduced to the problem of optimal assignments: the answer for the maximal assignment's problem is "yes" if and only if $|P| > 0$.

Let's suppose that we have both an access to the "black box" B returning a value of signal function ξ for a given pair of descriptors and no additional knowledge about the properties of ξ . Then in the worst case, each algorithm solving the problem of maximal assignment must use B to evaluate the value of ξ for each pair of descriptors from the set $\{\langle d, d' \rangle : \exists \underline{l} \in \mathcal{P}_{d,s} \ d' = s_{\underline{l}, \underline{u}}\}$.

Let $d = s_1 s_2 \dots s_k$ be a descriptor, let s be a segment of length n . If all segments s_i (for $i = 1, 2, \dots, k$) have length 1, then there are exactly $\binom{n}{k}$ different assignments of descriptor d to segment s . If lengths are different but less than a certain constant and we treat k as constant, then there is still $O(\binom{n}{k}) = O(n^k)$ different assignments. This implies, that if B evaluates the value of ξ in constant time, then the complexity of the optimal algorithm solving the maximal assignment's problem is $O(n^k)$ (such a number of operations is necessary if all possible assignments correspond to different pairs of descriptors).

Definition 4 Additive signal is a signal $\xi: \overline{\mathcal{D}^2} \rightarrow R$ satisfying for each $\langle d_1, d_2 \rangle \in \overline{\mathcal{D}^2}$ the equation

$$\xi(d_1, d_2) = \sum_{i=1}^k \xi(s_i, t_i),$$

where $d_1 = s_1 s_2 \dots s_k$ and $d_2 = t_1 t_2 \dots t_k$.

One may expect that considering only additive signals may lead to polynomial algorithm for finding the maximal assignment. Unfortunately it is not like that.

Definition 5 *The problem of optimal assignments restricted to additive signals is called the problem of **optimal additive assignments (OAA)**. The related maximal assignment's problem restricted in the same way is called the **maximal additive assignment's problem** and is denoted by **MAA**.*

3 NP-completeness of MAA problem

Theorem 1 *MAA problem is NP-complete.*

Proof: It is easy to see that $MAA \in NP$, since a nondeterministic algorithm needs only to guess an assignment \underline{l} of descriptor d of type \underline{u} to segment s and check in polynomial time that $\xi(d, s_{\underline{l}, \underline{u}}) \geq r$.

We will transform the known NP-complete problem of satisfiability of clauses' collection in which each variable appears at most 3 times, limited to clauses containing at most 3 literals ([7]).

Let boolean expression $\phi = C_1 \wedge C_2 \wedge \dots \wedge C_k$ be an arbitrary instance of such a satisfiability problem. Let $X = \{x_1, x_2, \dots, x_m\}$ be the set of all variables in ϕ . Obviously each clause C_i (for $i = 1, 2, \dots, k$) contains at most 3 literals. Additionally, each variable $x_i \in X$ appears at most in 3 different clauses. Without loss of generality we can assume, that each variable appears in a negative way at most as many times as in a positive way. For each positive integer $i \leq m$ let S_i^+ be a set of clauses in which variable x_i appears in a positive way, and S_i^- be a set of clauses in which variable x_i appears in a negative way. Obviously $S_i^+ \cap S_i^- = \emptyset$, $0 \leq |S_i^-| \leq |S_i^+| \leq 3$ and $1 \leq |S_i^-| + |S_i^+| \leq 3$. From the above it follows that $|S_i^-| \leq 1$.

Having ϕ we will describe construction of the instance $\langle d, s, \xi \rangle$ of MAA problem ($d \in (\mathcal{W}^*)^*$, $s \in \mathcal{W}^*$, $\xi \in R^{\overline{D^2}}$). In this construction we will use different elements $a_0, a_1, a_2, a_3, a_4, a_5$ from set \mathcal{W} .

Let $s = t_1 t_2 \dots t_m$ where (for $i = 1, 2, \dots, m$)

$$t_i = a_4 t_i^1 a_5 t_i^2 a_4 t_i^3 \text{ and } t_i^j = a_0^{2i-2} a_j a_0^{2m-2i} \text{ for } j = 1, 2, 3.$$

One can observe that that $|s| = 6m^2$.

Let $d = s_1 s_2 \dots s_k$ where $s_i = w_i^1 w_i^2 \dots w_i^{2m}$ (for $i = 1, 2, \dots, k$) and for each positive integer $j \leq m$:

$$w_i^{2j} = \begin{cases} a_{|\{t \in N: C_t \in S_j^+ \wedge t \leq i\}|} & \text{if } C_i \in S_j^+, \\ a_0 & \text{if } C_i \notin S_j^+, \end{cases}$$

and

$$w_i^{2j-1} = \begin{cases} a_1 & \text{if } C_i \in S_j^-, \\ a_0 & \text{if } C_i \notin S_j^-. \end{cases}$$

The length of each segment s_i (for $i = 1, 2, \dots, k$) is equal to $2m$, so the type of d is $\underline{u} = \langle 2m, 2m, \dots, 2m \rangle$. From the definition we can easily see that the i -th segment of d contains element a_1 in position $2j - 1$ (for a certain positive integer j) if and only

if variable x_j appears in clause C_i in a negative way. Analogously the i -th segment of d contains element a_1 or a_2 or a_3 in position $2j$ (for a certain positive integer j) if and only if variable x_j appears in clause C_i in a positive way.

Because the function $\xi: \overline{\mathcal{D}^2} \rightarrow R$ we want to construct is an additive signal, it is enough to define ξ on set

$$P = \{\langle d, d' \rangle : \exists \underline{l} \in \mathcal{P}_{d,s} \ d' = s_{\underline{l}, \underline{u}}\}.$$

Let then

$$\xi(w_1 w_2 \dots w_{2m}, v_1 v_2 \dots v_{2m}) = \begin{cases} \frac{r}{k} & \text{if } (v_1 \in \{a_4, a_5\} \vee v_{2m} = a_5) \wedge \exists j \leq 2m \ v_j = w_j \neq a_0, \\ 0 & \text{in other case.} \end{cases}$$

The construction described above is obviously polynomial because $|s| = 6m^2$, $\sum_{i=1}^k |s_i| = 2mk \leq 6m^2$ and $|P| < 6km^2 \leq 18m^3$. To prove that this construction is indeed a reduction, we must show that ϕ is satisfiable if and only if $\langle d, s, \xi \rangle \in MAA$.

Let's suppose first that there exists an assignment $\langle l_1, l_2, \dots, l_k \rangle \in \mathcal{P}_{d,s}$ such that $\xi(d, s_{\langle l_1 l_2 \dots l_k \rangle, \underline{u}}) \geq r$. Obviously $\xi(s_i, s_{l_i, 2m}) = \frac{r}{k}$ for $i = 1, 2, \dots, k$. If l_i (for $i = 1, 2, \dots, k$) is even then $l_i \in \{2, 6m+2, 12m+2, \dots, 6m(m-1)+2\}$ because a_5 occurs in s in positions $2m+1, 8m+1, 14m+1, \dots, 6m(m-1)+2m+1$ only. According to the definition of ξ , a_1 must occur in s_i in position $2\lceil l_i/6m \rceil - 1$, so $x_{\lceil l_i/6m \rceil}$ appears in C_i in a negative way. If l_i (for $i = 1, 2, \dots, k$) is odd then $l_i \in \{1, 2m+1, 4m+1, \dots, 6m(m-1)+4m+1\}$ because a_4 and a_5 occurs in s in positions $1, 2m+1, 4m+1, \dots, 6m(m-1)+4m+1$ only. According to the definition of ξ , a_1 or a_2 or a_3 must occur in s_i in position $2\lceil l_i/6m \rceil$, so $x_{\lceil l_i/6m \rceil}$ appears in C_i in a positive way.

Summing up: if l_i is even, then $\neg x_{\lceil l_i/6m \rceil} \in C_i$, and if l_i is odd, then $x_{\lceil l_i/6m \rceil} \in C_i$ (for $i = 1, 2, \dots, k$). Moreover, for any integers $1 \leq i_1 < i_2 \leq k$, if $\lceil l_{i_1}/6m \rceil = \lceil l_{i_2}/6m \rceil$, then $x_{\lceil l_{i_1}/6m \rceil} \in C_{i_1}$ and $x_{\lceil l_{i_2}/6m \rceil} \in C_{i_2}$, because in ϕ each variable appears in a negative way at most once and \underline{u} is admissible for $\langle l_1, l_2, \dots, l_k \rangle$.

Let $X' = \{x \in X : \exists 1 \leq i \leq k \ x = x_{\lceil l_i/6m \rceil}\}$. Let ψ be a truth assignment defined for each element of X' as follows:

$$\psi(x) = \begin{cases} F & \text{if } \exists 1 \leq i \leq k \ (x = x_{\lceil l_i/6m \rceil} \wedge \neg x \in C_i), \\ T & \text{in other case.} \end{cases}$$

In this truth assignment each clause from ϕ is "true", what shows that ϕ is satisfiable.

Now we will prove, that if a boolean expression ϕ is satisfiable, then such an assignment $\underline{l} \in \mathcal{P}_{d,s}$ that $\xi(d, s_{\underline{l}, \underline{u}}) \geq r$ exists. It is enough to show that there exists a k -marking $\underline{l} = \langle l_1, l_2, \dots, l_k \rangle$ such that:

- $\forall 1 \leq i \leq k \ \xi(s_i, s_{l_i, 2m}) = \frac{r}{k}$,
- \underline{u} is admissible for \underline{l} .

Let's suppose that it is untrue. This implies that for each k -marking $\underline{l} = \langle l_1, l_2, \dots, l_k \rangle$ such that $\forall 1 \leq i \leq k \ \xi(s_i, s_{l_i, 2m}) = \frac{r}{k}$ there exists integers $1 \leq i_1 < i_2 \leq k$ such that $|l_{i_1} - l_{i_2}| < 2m$.

The boolean expression ϕ is satisfiable if and only if there exists a truth assignment in which each clause is "true". It is equivalent to say that there exists a sequence of literals y_1, y_2, \dots, y_k that satisfies the following two conditions:

- $\forall 1 \leq i \leq k \ y_i \in C_i,$
- $\forall 1 \leq i_1 < i_2 \leq k \ y_{i_1} \neq \neg y_{i_2}.$

Let x_{j_1} be a variable from literal y_{i_1} and x_{j_2} be a variable from literal y_{i_2} . If $j_1 \neq j_2$, then $6mj_1 - 6m + 1 \leq l_{i_1} \leq 6mj_1 - 2m + 1$ and $6mj_2 - 6m + 1 \leq l_{i_2} \leq 6mj_2 - 2m + 1$, so $|l_{i_1} - l_{i_2}| \geq 2m$. If $j_1 = j_2$ then $y_{i_1} = y_{i_2}$ what implies that $|l_{i_1} - l_{i_2}| \geq 2m$ (from ξ definition). This contradiction proves that there exists a k -marking $\underline{l} = l_1, l_2, \dots, l_k$ that satisfies the demanded conditions.

NP-completeness of the MAA problem means that we can't expect an algorithm solving the problem of maximal additive assignment in time $O(p(k))$, where p is a polynomial and k is a number of segments in the considered descriptor. The same refers to the problem of optimal additive assignments.

However, in certain special cases a polynomial algorithm exists. If for example, all segments in a descriptor d have the same length equal to one, the maximal additive assignment problem is equivalent to the solvable in polynomial time problem of finding the perfect matching of maximal weight in bipartite graph with edge weights. For a given descriptor $d = s_1 s_2 \dots s_k$, a segment s and an additive signal ξ , let $G = (V_1 \cup V_2, E)$ be a bipartite graph such that:

- $|V_1| = k$ and $|V_2| = |s|,$
- for each assignment l of segment s_i to segment s there is an edge of weight $\xi(s_i, s_{l,|s_i|})$ between the i -th vertex of V_1 and the l -th vertex of V_2 .

It is easy to see that the above construction is a polynomial reduction.

Unfortunately, for biological data the length of segments in descriptor is equal to at least five, so the above reduction doesn't apply. However, there is an algorithm of a complexity lower than $\Omega(n^k)$ solving the MAA problem. After a little modification this algorithm also solves the OAA problem in the same time, if only there are certain restrictions for the size of an output.

4 Algorithm for finding optimal additive assignments

Let X be a finite set of real numbers, let m be a positive integer. $MAX_m(X)$ is the set of the largest m elements of X , if X has at least m elements; otherwise we set $MAX_m(X) = X$.

Let s be a segment of length n , let $d = s_1s_2\dots s_k$ be a descriptor. For a given positive integer $i \leq n$ by $s^{\leq i}$ we denote a prefix of length i of segment s . Let Y be a subset of $\{1, 2, \dots, k\}$. By Y_j we denote such an integer $y \in Y$ that $|\{y' \in Y : y' \leq y\}| = j$ (Y_j is the j -th smallest element of Y). By d_Y we denote descriptor $s_{Y_1}s_{Y_2}\dots s_{Y_{|Y|}}$. In other words d^Y is a subsequence of d defined by increasing elements of Y .

Let $\xi \in R^{\overline{D^2}}$ be an additive signal. Let Z be a two-dimensional matrix having 2^k rows corresponding to all subsets of set $\{1, 2, \dots, k\}$ and n columns. For each $Y \subseteq \{1, 2, \dots, k\}$ and $1 \leq i \leq n$, $Z[Y, i]$ is a set of real numbers defined as follows:

$$Z[Y, i] = \text{MAX}_m \{ \xi(d_Y, s_{\underline{l}, \underline{u}}^{\leq i}) : \underline{l} \in \mathcal{P}_{d_Y, s^{\leq i}} \wedge \underline{u} = \text{type}(d_Y) \}.$$

$Z[\{1, 2, \dots, k\}, n]$ is a set of the best m values of assignments from the set $\mathcal{P}_{d, s}$. The remaining fields of matrix Z correspond to assignments of a certain fragment of descriptor d to a certain prefix of a segment s .

Theorem 2 *Let segment s , descriptor d and matrix Z be defined as above. For each $Y \subseteq \{1, 2, \dots, k\}$ and $1 \leq i \leq n$ the following equation takes place:*

$$Z[Y, i] = \begin{cases} \emptyset & \text{if } Y = \emptyset \vee i < \sum_{y \in Y} |s_y|, \\ \text{MAX}_m(V_{Y, i}) & \text{in other case,} \end{cases}$$

where ¹

$$V_{Y, i} = \bigcup_{y \in Y} (Z[Y - \{y\}, i - |s_y|] + \xi(s_y, s_{i+1-|s_y|, |s_y|})) \cup Z[Y, i - 1].$$

Proof: If $Y = \emptyset$ or $i < \sum_{y \in Y} |s_y|$, then obviously the equation is true.

Let $|Y| = p \geq 1$. Let $d_Y = t_1t_2\dots t_p$, let $\langle l_1, l_2, \dots, l_p \rangle$ be an assignment of descriptor d_Y of type $\underline{u} = \langle u_1, u_2, \dots, u_p \rangle$ to segment $s^{\leq i}$ such that $\xi(d_Y, s_{\langle l_1, l_2, \dots, l_p \rangle, \underline{u}}^{\leq i}) \in Z[Y, i]$. There are two possible cases:

(1) $\forall 1 \leq j \leq p \ l_j + u_j \leq i$.

In this case obviously we have $\xi(d_Y, s_{\langle l_1, l_2, \dots, l_p \rangle, \underline{u}}^{\leq i}) = \xi(d_Y, s_{\langle l_1, l_2, \dots, l_p \rangle, \underline{u}}^{\leq i-1})$.

(2) $\exists 1 \leq j \leq p$ such that $l_j + u_j = i + 1$.

Let $d' = t_1t_2\dots t_{j-1}t_{j+1}\dots t_p$, let \underline{u}' be a type of d' . Let:

- $x = \xi(d', s_{\langle l_1, l_2, \dots, l_{j-1}, l_{j+1}, \dots, l_p \rangle, \underline{u}'}^{\leq i-u_j})$,
- $y = \xi(t_j, s_{i+1-u_j, u_j})$.

¹The notation $Z + r$ means $\{x + r : x \in Z\}$

Let $A = Z[Y - \{Y_j\}, i - u_j]$. For each $z \in A$ there exists an assignment $\langle l'_1, l'_2, \dots, l'_{p-1} \rangle$ of descriptor d' to segment $s^{\leq i - u_j}$ such that $\xi(d', s^{\leq i - u_j}_{\langle l'_1, l'_2, \dots, l'_{p-1} \rangle, \underline{u}'}) = z$. Obviously $\langle l'_1, l'_2, \dots, l'_{j-1}, (i - u_j + 1), l'_j, \dots, l'_{p-1} \rangle$ is an assignment of d_Y to $s^{\leq i}$ and

$$\xi(d_Y, s^{\leq i}_{\langle l'_1, l'_2, \dots, l'_{j-1}, (i - u_j + 1), l'_j, \dots, l'_{p-1} \rangle, \underline{u}}) = z + y.$$

If $|A| < m$, then obviously $x \in A$ because $\langle l_1, l_2, \dots, l_{j-1}, l_{j+1}, \dots, l_p \rangle$ is an assignment of d' to $s^{\leq i - u_j}$. If $|A| = m$, then also $x \in A$. In order to show this, suppose for a while that $x \notin A$. Then $Z[Y, i]$ contains m different elements of shape $z + y$ where $z \in A$ and element $x + y$ which is smaller than all the above elements. This implies that $|Z[Y, i]| = m + 1$ what contradicts the definition. So

$$\xi(d_Y, s^{\leq i}_{\langle l_1, l_2, \dots, l_p \rangle, \underline{u}}) = x + \xi(t_j, s_{i+1-u_j, u_j}),$$

where $x \in Z[Y - \{Y_j\}, i - u_j]$.

The above argument proves that each element of set $Z[Y, i]$ must belong to the set $V_{Y, i}$. This implies that $Z[Y, i] = MAX_m(V_{Y, i})$ because for each $v \in V_{Y, i}$ there exists an assignment \underline{l}' of d_Y to $s^{\leq i}$ such that $\xi(d_Y, s^{\leq i}_{\underline{l}', \underline{u}}) = v$.

Theorem 2 gives rise to a dynamic programming algorithm which we present below. The input for this algorithm is: a descriptor $d = s_1 s_2 \dots s_k$, a segment s of length n , an additive signal ξ and a positive integer m ; the output is a set of the best m values of assignments.

OPTIMAL_ASSIGNMENTS

```

1  for j := 0 to n do
2    for i := 1 to 2k do
3      if (j = 0) then
4        S := ∅
5      else S := Z[i, j - 1]
6      tmp := i
7      for l := 1 to k do
8        r := tmp mod 2
9        tmp := (tmp - r)/2
10     if (r = 1) and (j ≥ |sl|) then
11       S := S ∪ Z[i - 2l-1, j - |sl|] + ξ(sl, sj+1-|sl|, |sl|)
12     Z[i, j] := MAXm(S)
13 return Z[2k, n]
```

The above algorithm uses an auxiliary matrix Z containing 2^k rows and $n + 1$ columns. The first column plays the role of a "guard" and is filled with empty sets (line 4). The remaining columns are filled (the way of running through the matrix is defined in lines 1 and 2) according to the equation from Theorem 2 (lines 5 to 12).

The technical operation is to code subsets of a set of all segments using integers. In the binary representation of each such integer bits one correspond to these segments which belong to the subset. The result appears in the bottom right field of matrix Z (line 13).

It is worth mentioning, that algorithm *OPTIMAL_ASSIGNMENTS* is presented in a very schematic way. The undefined structure S we use in the algorithm explains well the idea of algorithm, but doesn't reveal its complexity. The best results are achieved when in each field of matrix Z the elements are sorted decreasingly. In order to evaluate $Z[i, j]$ for certain integers i and j it is necessary in the worst case to choose m times the greatest number from the set of $k + 1$ numbers. It can be done in $O(km)$ time. This implies that time complexity of the whole algorithm is $O(kmn2^k)$ (space complexity is $O(mn2^k)$). For biological data parameter k may be treated as a constant (for all constructed groups the number of segments does not exceed 10, while a length of a protein n is usually a few hundreds) what causes the presented algorithm to be very efficient.

The algorithm *OPTIMAL_ASSIGNMENTS* solves the problem of maximal assignment in $O(kn2^k)$ time: if $Z[2^k, n]$ contains an element x such that $x \geq r$, then the answer for maximal assignment's problem is "yes", otherwise "no". In order to find not only the values of the best assignments but also these assignments, it is necessary to introduce a standard modification for dynamic programming: with each element x in matrix Z three integers describing coordinates (in Z) of element y depending on which x was evaluated should be associated. If x may be evaluated depending on a few different elements, than for each such an element a triple of integers must be associated. When the whole matrix Z is calculated using *OPTIMAL_ASSIGNMENTS* algorithm, the set P of optimal additive assignments may be reconstructed. Obviously the discussed modification doesn't influence the aforementioned complexity of the algorithm, if only $|P| = O(mn2^k)$.

5 Discussion

As said in the introduction a local descriptor-based approach to protein structure prediction is being researched. The prototype set of a few thousand groups has been already constructed, a simple additive signal function has been defined. Presented *OPTIMAL_ASSIGNMENTS* algorithm makes it possible to efficiently evaluate the best assignments of these groups to different query sequences. The first trials to protein structure prediction are being undertaken, based on these assignments.

Considering additive signals is an essential simplification. Such a signal doesn't take into account the correlation between sequences of different segments from the same group. For this reason, a more sophisticated approach (based on Dirichlet mixtures) to signal definition is being developed. For solving optimal assignments' problem for such a not additive function we plan to use certain heuristics based on the aforementioned results.

References

- [1] K. F. Han, D. Baker (1996) Global properties of mapping between local amino acid sequence and local structure in proteins. *Proc. Natl. Acad. Sci.* 93, 5814–5818
- [2] K. T. Simons, C. Kooperberg, E. Huang, D. Baker (1997). Assembly of Protein Tertiary Structures from Fragments with Similar Local Sequences using Simulated Annealing and Bayesian Scoring Functions. *J. Mol. Biol.* 268, 209–225
- [3] R. Bonneau, J. Tsai, I. Ruczinski, D. Chivian, C. Rohl, C. E. Strauss, D. Baker (2001). ROSETTA in CASP4: Progress in ab initio protein structure prediction. *Proteins Suppl.* 5, 119–126
- [4] C. Bystroff, D. Baker (1998). Prediction of local structure in proteins using a library of sequence-structure motifs. *J. Mol. Biol.* 281, 565–577
- [5] K. Simons, R. Bonneau, D. Baker (2001) Prospects for ab initio protein structural genomics. *J. Mol. Biol.* 306, 1191–1199
- [6] A. Kryshchak, K. Fidelis Local descriptors of protein structure. Part I. General approach and classification of local 3D regions in proteins. *In preparation*
- [7] M. R. Garey, D. S. Johnson (2000) Computers and intractability. A guide to the theory of NP-completeness. *W. H. Freeman and Company*