BioSystems, 30 (1993) 173–182
Elsevier Scientific Publishers Ireland, Ltd.

Prediction of the exon-intron structure by a dynamic programming approach

M. S. Gelfand[a] and M. A. Roytberg[b]

[a] *Institute of Protein Research, Russia Academy of Sciences, Pushchino, Moscow region, 142292, Russia and*
[b] *Institute of Mathematical Problems in Biology, Russia Academy of Sciences, Pushchino, Moscow region, 142292, Russia*

Introduction

Prediction of protein-coding regions is one of the most actual problems in computer genetics. These predictions can be performed with reasonable accuracy for prokaryotic DNA, in which proteins are encoded by long uninterrupted open reading frames. However, the situation in eukaryotic case is much more difficult. Non-coding introns, interrupting reading frames, are generally longer than coding fragments (exons) (Hawkins, 1988). Predictions in this case can employ two kinds of information.

First, it is possible to devise a procedure for prediction of exon-intron boundaries (splicing sites; see (Gelfand, 1989) and references therein, reviewed in (Gelfand, 1992a, 1992c). The second approach is to measure statistical constraints imposed by the protein-coding function onto DNA sequence (reviewed in (Gelfand, 1990b, 1992c). Unfortunately, existing methods in both fields are not sufficiently reliable. Site-prediction algorithms either produce a lot of false positives, or miss true sites, while global functions do not map exon-intron boundaries with the required precision and tend to miss exons that are shorter then the width of the scanning window (the latter cannot be decreased due to the ever-present statistical noise).

Some recently suggested algorithms combining both approaches produced rather hopeful results (Gelfand, 1990a, 1992d; Fields and Soderlund, 1990; Uberbacher and Mural, 1991; Legouis et al., 1991; Guigo et al., 1992; Rogozin, 1992). These algorithms predict the exon-intron structure as a whole, employing variants of the following technique. First a site-prediction function is used with a very relaxed conditions on a putative site. The purpose of it is not to miss any true site, while false positives are allowed and will be dealt with on the next stage. Then all possible combinations of donor and acceptor sites defining variants of spliced mRNA (exon-intron structures) are considered and for each of them its coding potential (in the global sense described above) is evaluated. This value is summed with mean predicted strengths of donor and acceptor sites defining this structure (for compatibility all parameters can be measured in SD units) and the sum R serves as an estimate of the structure quality. Several mRNAs with highest R are translated and the resulting proteins are considered to be possibly encoded in the DNA region under consideration.

From the user point of view the last feature is extremely important, since it allows prediction results to be immediately used for further analysis (protein homology searches, search for functional patterns, analysis of physical properties, secondary structure prediction, and, finally, prediction of function). Experiments, in which Berg and von Hippel discrimination energy (Berg and von Hippel, 1987) was used for prediction of sites and Fickett TestCode (Fickett, 1982) served as a coding potential, produced reasonable results (Gelfand, 1990a, 1992d). In most cases the true spliced mRNA was among the top 1–30 of mRNAs with high R, while the best variant (that with the highest R) always included more than half of the encoded protein. Moreover, the true structure either was the best among structures with the correct number of exons, or differed from the best one only slightly.

However, this approach meets rather serious computational difficulties. The problem is that the number

0303-2647/93/$ 06.00

of structures on the average exponentially depends on the number of putative sites. Thus either we have to consider only relatively short DNA sequences (Gelfandn 1990a, 1992d), or to impose too strict conditions on putative sites, thus risking to lose some true sites (Gelfand, 1990a, 1992d; Rogozin, 1992) or to employ various empirical techniques in order to avoid the full search (Guigo et al., 1992; Rogozin, 1992).

Another possibility is to employ a dynamic programming technique and thus obtain a faster algorithm without loss of possible sites and good structures. In this paper we present such an algorithm in several modifications. First, the problem is stated formally and some main definitions are given. Then the algorithm searching for the best structure in each class (defined by the number of exons) is presented. Then its generalizations are described (search for k best structures and search for structures in ε-neighborhood of the best one). Finally, we discuss relation of the present problem to the classical problem of dynamic programming.

Statement of the problem

Consider a nucleotide sequence in which putative donor and acceptor sites, as well as start and stop codons, are marked. Each site is characterized by a positive number estimating its strength. It is assumed that all true sites have strengths exceeding some threshold and that the strength of a true site usually exceeds strengths of most false sites (but exclusions from this rule are allowed). In particular, the latter property does not allow the use of binary Yes/No site-prediction functions.

Definition. *Left bracket* is a translation start codon or an acceptor site. *Right bracket* is a translation stop codon or a donor site.

Remark. We assume that brackets are situated between nucleotides (in half-integer positions $p = 1/2, 3/2, \dots$).

Definition. *Reading frame* f of a sequence fragment is determined by the codon position of the first nucleotide; $f = 1, 2, 3$.

Definition. *Exon* is a sequence fragment bounded by a left bracket and a right bracket to which a reading frame f is associated. Exon cannot contain in-frame stop codons. Each exon e is characterized by a positive number $c(e)$ called *coding potential* and strengths of its acceptor and donor sites $a(e)$ and $d(e)$ respectively.

Definition. *Initial exon* is an exon whose left bracket is a start codon. *Terminal exon* is an exon whose right bracket is a stop codon. *Inner exon* is an exon bounded by a donor site and an acceptor site.

Remark. A pair of donor and acceptor sites can define 0 though 3 inner codons (in the first case all reading frames between the donor and acceptor sites are interrupted by stop codons, while in the last case none are).

Definition. *Intron* is a sequence fragment betweed a donor site and an acceptor site.

Definition. *Structure* is a sequence of non-overlapping exons ordered from left to right along a sequence with consistent reading frames, such that all exons, except maybe the first one and the last one, are inner exons. Each structure $\mathbf{s} = e_1, \dots, e_N$ is characterized by the following parameters:

$N(\mathbf{s})$ — number of exons;

$A(\mathbf{s})$ — total strength of acceptor sites, $A(\mathbf{s}) = \sum_{n=1}^{N} a(e_n)$;

$D(\mathbf{s})$ — total strength of donor sites, $D(\mathbf{s}) = \sum_{n=1}^{N} d(e_n)$;
$L(\mathbf{s})$ — total exon length;
$C(\mathbf{s})$ — total coding potential of exons, $C(\mathbf{s}) = \sum_{n=1}^{N} c(e_n)$;
$E(\mathbf{s})$ — position of the right bracket of the last exon.

Definition. A structure is *initial* if its first exon is initial. A structure is *terminal* if its last exon is terminal.

Unless specified otherwise we consider only initial structures.

Definition. *Target set* $\mathcal{T}$ is a set of initial terminal structures such that for any monotonically increasing function of mean site strengths and weighted mean coding potential of exons $R(A/N, D/N, C/L)$ (*structure quality*) and any fixed number of exons N^* the set $\mathcal{T}$ contains the structure $\mathbf{t}$ with the maximum R among structures with the given number of exons N^*: $N(\mathbf{t}) = N^*$ and

$$R\big(A(\mathbf{t})/N(\mathbf{t}), D(\mathbf{t})/N(\mathbf{t}), C(\mathbf{t})/L(\mathbf{t})\big) \quad = \quad \max_{\{\,\mathbf{s} | N(\mathbf{s}) = N^*\,\}} R\big(A(\mathbf{s})/N(\mathbf{s}), D(\mathbf{s})/N(\mathbf{s}), C(\mathbf{s})/L(\mathbf{s})\big).$$

Construction of a target set is the objective of the algorithm.

The basic algorithm

Consider the following *domination conditions* (p is the current left bracket position):

$$\begin{aligned}
\mathbf{T}(\mathbf{r},\mathbf{s}) &= C(\mathbf{r}) \geq C(\mathbf{s}) \,\&\, L(\mathbf{r}) \leq L(\mathbf{s}) \\
&\quad \&\, A(\mathbf{r}) \geq A(\mathbf{s}) \,\&\, D(\mathbf{r}) \geq D(\mathbf{s}) \,\&\, N(\mathbf{r}) \leq N(\mathbf{s}), \\
\mathbf{NT}(\mathbf{r},\mathbf{s}) &= \mathbf{T}(\mathbf{r},\mathbf{s}) \\
&\quad \&\, L(\mathbf{r}) = L(\mathbf{s}) \mod 3 \\
&\quad \&\, \big(E(\mathbf{s}) < p \Rightarrow E(\mathbf{r}) < p\big) \\
&\quad \&\, \big(E(\mathbf{s}) \geq p \Rightarrow E(\mathbf{r}) \leq E(\mathbf{s})\big).
\end{aligned}$$

Definition. Structure $\mathbf{r}$ *dominates* over structure $\mathbf{s}$ (denoted $\mathbf{r} \succ \mathbf{s}$) if either $\mathbf{r}$ and $\mathbf{s}$ are terminal and $\mathbf{T}(\mathbf{r},\mathbf{s})$ holds, or $\mathbf{r}$ and $\mathbf{s}$ are non-terminal and $\mathbf{NT}(\mathbf{r},\mathbf{s})$ holds.

Lemma 1. If $\mathbf{r} \succ \mathbf{s}$, then $R(\mathbf{r}) \geq R(\mathbf{s})$ for any structure quality R.

Proof. Obvious.

Lemma 2. The relation of domination is transitive, i.e. if $\mathbf{r} \succ \mathbf{s}$ and $\mathbf{s} \succ \mathbf{t}$, then $\mathbf{r} \succ \mathbf{t}$.

Proof. Transitivity of condition $(\mathbf{T})$ is trivial. If $E(\mathbf{t}) < p$, then (since $\mathbf{s} \succ \mathbf{t}$) $E(\mathbf{s}) < p$, and thus (since $\mathbf{r} \succ \mathbf{s}$) $E(\mathbf{r}) < p$. If $E(\mathbf{t}) \geq p$ and $E(\mathbf{s}) \geq p$, then $E(\mathbf{r}) \leq E(\mathbf{s}) \leq E(\mathbf{t})$. Finally, if $E(\mathbf{t}) \geq p$, while $E(\mathbf{s}) < p$, then (since $\mathbf{s} \succ \mathbf{r}$) $E(\mathbf{r}) < p$ also, and thus $E(\mathbf{r}) < E(\mathbf{t})$.

Three structure sets are supported: the *back set* $\mathcal{B}$ and the *front set* $\mathcal{F}$ consist of non-terminal structures (the union $\mathcal{M} = \mathcal{B} \cup \mathcal{F}$ is called *base set*), while $\mathcal{T}$ consists of terminal structures. At each step defined by the current position p these sets satisfy the following conditions:

(i) if $\mathbf{r}, \mathbf{s} \in \mathcal{B} \cup \mathcal{F}$, then neither $\mathbf{r} \succ \mathbf{s}$, nor $\mathbf{s} \succ \mathbf{r}$;

(ii) if $\mathbf{s} \notin \mathcal{B}$ is non-terminal and $E(\mathbf{s}) < p$, then there exists $\mathbf{r} \in \mathcal{B}$ such that $\mathbf{r} \succ \mathbf{s}$;

(iii) if $\mathbf{s} \notin \mathcal{F}$ is non-terminal, the last exon of $\mathbf{s}$ starts to the left of p and $E(\mathbf{s}) < p$ (thus p is inside the last exon of $\mathbf{s}$), then there exists $\mathbf{r} \in \mathcal{F}$ such that $\mathbf{r} \succ \mathbf{s}$;

(iv) if $\mathbf{r}, \mathbf{s} \in \mathcal{T}$, then neither $\mathbf{r} \succ \mathbf{s}$, nor $\mathbf{s} \succ \mathbf{r}$;

(v) if $\mathbf{s} \notin \mathcal{T}$ is terminal and the last exon of $\mathbf{s}$ starts to the left of p, then there exists $\mathbf{r} \in \mathcal{T}$ such that $\mathbf{r} \succ \mathbf{s}$.

All sets are initialized as empty ones. The basic procedure of the algorithm is *modification* of a set $\mathcal{S}$ by a structure $\mathbf{r}$. Denote the modified set by $\mathcal{S}_{\mathbf{r}}$. If for some $\mathbf{s} \in \mathcal{S}$ $\mathbf{s} \succ \mathbf{r}$, then $\mathcal{S}_{\mathbf{r}} = \mathcal{S}$. Otherwise $\mathbf{r}$ in included into the modified set, while all structures dominated by it are excluded. Formally,

$$\mathcal{S}_{\mathbf{r}} = \begin{cases} \mathcal{S}, & \text{if } \exists \mathbf{s} \in \mathcal{S} \text{ such that } \mathbf{s} \succ \mathbf{r}, \\ \{\mathbf{s} \in \mathcal{S} \mid \mathbf{r} \not\succ \mathbf{s}\} \cup \{\mathbf{r}\}, & \text{otherwise.} \end{cases}$$

Current position p is moved along the sequence from left to right. When a start codon is encountered, the algorithm generates all initial exons corresponding to it, and each of them initializes a structure. When an acceptor site is encountered, all structures $\mathbf{s}$ with $E(\mathbf{s}) < p$ are excluded from the front subset $\mathcal{F}$ and each of them modifies the back subset $\mathcal{B}$. Then each exon corresponding to the current acceptor site is added to all structures from the back subset $\mathcal{B}$ that can be extended by this exon (recall that the reading frame is incuded into the definition of an exon, and thus each acceptor site generally defines three classes of exons).

In both cases (i.e. if the left bracket is a start codon or an acceptor site) exons are generated in the order defined by their right brackets from left to right. If the current exon is terminal, and thus all newly generated structures also are terminal, then these structures modify the target set $\mathcal{T}$. If the current exon is non-terminal, then the generated structures modify the front subset $\mathcal{F}$.

The process terminates when the current position reaches the end of the sequence.

Lemma 3. The set $\mathcal{T}$ constructed by the above procedure satisfies conditions (4) and (5) and is a target set. is a target set.

Remark 1. Condition (4) is not necessary, it just frees us from unnecessary elements.

Remark 2. Actually, the presented algorithm constructs a target set for any function $R^*(A, D, C, L, N)$ monotonically increasing on A, D, and C and decreasing on L and N.

Proof. Simply follows from Lemmas 1 and 2.

Embellishments

In this section we briefly describe some modifications of the main algorithm.

Search for k best structures

If one desires to obtain not one, but k best structures in each exon class, it is sufficient to require existence of k dominating structures in the set modification procedure.

Partially sequenced genes

In order to account for the possibility of a partially sequenced gene, it is sufficient to introduce dummy left and right brackets at both ends of a DNA sequence in all three reading frames.

Restrictions on intron length

can be of two kinds. Restriction on the minimum intron length $l_{\min}$ is satisfied if the boundary between the back subset $\mathcal{B}$ and the front subset $\mathcal{F}$ is set at the position $p - l_{\min}$ (where p is the current position). Restriction on the maximum intron length $l_{\max}$ is satisfied if we forcefully exclude from the back subset $\mathcal{B}$ such structures $\mathbf{r}$ that $E(\mathbf{r}) + l_{\max} < p$.

Additional parameters and restrictions

can be introduced into the general scheme in a similar manner. Thus it is possible to account for preferences between intron types (codon positions of the exon-intron boundary), exon types (exon length mod 3) and correlations between these characteristics for neighboring exons and/or introns (Gelfand, 1992b). Dependence of the splicing site signals from the intron type (Gelfand, 1992b) can be accounted for by independent prediction of all three types of splicing sites (both acceptor and donor) with the requirement that types of sites defining an exon, its reading frame, and its type are consistent.

The use of *a priori* parameters also is possible. In particular, rough estimates of protein length or molecular weight can be used, first, for excluding structures coding for too short (small) proteins from the target set, and second, for exclusion from the front subset $\mathcal{F}$ structures corresponding to too long or heavy proteins. Similar technique applies to *a priori* estimates on the number of exons, but in this situation one has to be vary careful, since our formal definition of an exon differs from the biological one (we consider only protein-coding exons, while experimental methods largely do not discriminate between coding and noncoding exons).

Search for structures in the ε-neighborhood of the best one

We fix now the structure quality function $R(A/N, D/N, C/L)$. Denote $R(\mathbf{s}) = R(A(\mathbf{s})/N(\mathbf{s}), D(\mathbf{s})/N(\mathbf{s}), C(\mathbf{s})/L(\mathbf{s}))$ and let $R_{\max} = \max_{\mathbf{s}} R(\mathbf{s})$. In this section we present an algorithm that constructs the set $\mathcal{T}_\varepsilon = \{\, \mathbf{t} \mid R(\mathbf{t}) > R_{\max} - \varepsilon \,\}$. As before, only the case of fully sequenced genes is considered, although the algorithm can be easily modified in order to account for a possibility of partial sequencing.

The procedure consists of two stages. At the first stage we construct the optimal structure as described above. However, for all acceptor sites we retain parameters A, D, N, C, L of all structures from the back subset $\mathcal{B}$ corresponding to the position p of this site (i.e. of such structures $\mathbf{r} \in \mathcal{M}$ that $E(\mathbf{r}) < p$). Let $R_{\max}$ be the obtained maximum structure quality.

Now re-initialize the target set $\mathcal{T}$. At the second stage we consider right brackets in the right to left order. We redefine the front set $\mathcal{F}$. It is now a set of non-initial terminal structures situated entirely to the right of the current right bracket position. For a given right bracket consider in the right to left order left brackets corresponding to exons (thus the reading frame has to be taken into account and no in-frame terminal codons should occur). The base set $\mathcal{M}$ consists now of the front subset $\mathcal{F}$ and the remainder $\mathcal{R}$ in which other terminal non-initial structures are placed; structures from $\mathcal{R}$ modify the subset $\mathcal{F}$ when the current right bracket moves.

Assume for clarity that both brackets are sites (the case of an initial and/or a terminal codons is simpler) and let e be the exon defined by the current acceptor and donor sites. Consider all structures $\mathbf{s} \in \mathcal{F}$ and compute qualities of all correct structures that are concatenations of a structure from the back subset $\mathcal{B}$ corresponding to the current acceptor site, the current exon e, and $\mathbf{s}$ (it is not necessary to know the structures from $\mathcal{B}$ themselves, since their parameters, which have been retained at the first stage, are sufficient). Let $R^*(\mathbf{s})$ be the maximum of structure qualities associated with $\mathbf{s}$. If $R^*(\mathbf{s}) \geq R_{\max} - \varepsilon$, then

add the current exon e to the structure $\mathbf{s}$ and place the obtained structure into the remainder subset $\mathcal{R}$. Otherwise consider the next structure from $\mathcal{F}$. Processing of the current exon completes after consideration of all structures from $\mathcal{F}$, while processing of the current right bracket completes after consideration of all exons corresponding to this right bracket.

If the current right bracket is a stop codon, it is not associated with a front subset and we consider terminal structures that are concatenations of a structure from the back subset $\mathcal{B}$ and the current (terminal) exon. Similarly, if the current left bracket is a start codon, there is no back subset associated with it, we consider initial terminal structures that are concatenations of the current (initial) exon and a structure from the front subset $\mathcal{F}$ and the resulting structures are placed not into the remainder subset $\mathcal{R}$, but into the target set $\mathcal{T}$.

The desired set $\mathcal{T}_\varepsilon$ of structures situated in the ε-neighborhood of the optimal one coincides with the target set $\mathcal{T}$ after processing of all right brackets.

The general statement of the problem in the graph theory language

Recall some definitions (for details see monograph (Aho et al.,1976) and papers (Lengauer and Theune, 1991; Finkelstein and Roytberg, this volume)).

Let $G = (V, E)$ be an acyclic directed oriented graph (below only such graphs will be considered). A vertex $v \in V$ is called *source* if no arc enters it, and *sink* if no arc exits it. A path starting in a source is called *initial*, and an initial path ending in a sink is called *full*.

Cost system on G is a quadruplet $S = (K, \lambda, \otimes, \prec)$, where

K is some set (*set of weights*);

λ is a function $E \to K$ ascribing to each arc its weight;

$\otimes$ is an associative operation defining the weight of a path $\mathbf{p} = (e_1, \ldots, e_n)$ given weights of the constituent arcs:

$$\lambda(\mathbf{p}) = ((\lambda(e_1) \otimes \lambda(e_2)) \ldots) \otimes \lambda(e_n);$$

$\prec$ is a relation of full ordering on K.

In the simplest case K is the set $\mathbb{R}$ of real numbers, $\otimes$ is addition, $\prec$ is the usual ordering. More complicated exaples of weigth systems are presented in (Lengauer and Theune, 1991; Finkelstein and Roytberg, this volume). For instance, it can be assumed that K is an arbitrary set, $a \prec b$ if $\mu(a) \prec \mu(b)$, where $\mu : K \to \mathbb{R}$ is some encoding function.

Example. Search for the opimal path. For a graph $G = (V, E)$ and a weight system $S = (K, \lambda, \otimes, \prec)$ find in G a full path $\mathbf{p}_0$ of the minimal (with respect to the relation $\prec$) weight, that is such that

$$\lambda(\mathbf{p}_0) = \min\{\, \lambda(\mathbf{p}) \mid \mathbf{p} \text{ is a full path in } G \,\}.$$

If the operation $\otimes$ is distibutive relative to minimum, i.e. if

$$a \prec b \Rightarrow a \otimes c \prec b \otimes c, \qquad (*)$$

then the problem is solved by the usual Bellman algorithm. This algorithm (more exactly, one of its numerous versions) consists of moving from the source to the sink, determining for each vertex v a minimal initial path coming to v. The condition $(*)$ allows to retain for each vertex only one (minimal) path.

In the exon-intron structure prediction problem one can consider as vertices of a graph $G_X = (V_X, E_X)$ the set $V_X = \{s\} \cup X\{t\}$, where X is the set of exons, s and t are special vertices (resp. source and sink).

Let us be interested only in terminal structures and let no restrictions on intron length be considered. Then arcs from s come to all initial exons, arcs from all terminal exons come to t, and an arc comes from

$x_1 = (a_1, b_1)$ to $x_2 = (a_2, b_2)$ if and only if $b_1 < a_2$ (for simplicity we do not account for reading frames; the generalization for this case, as well as for other problems considered above, is fairly straighforward). The weigth of an arc e is completely defined by the exon which is entered by this arc. If an arc e enters an exon x, then let

$$\lambda(e) = \big(A(x), D(x), C(x), L(x), 1\big).$$

The composition operation $\otimes$ is the component-wise addition. In particular, if $\mathbf{p}$ is a path, then the last component of the weight $\lambda(\mathbf{p})$ is the length $N(\mathbf{p})$ of the path $\mathbf{p}$. The relation $\prec$ is determined by the quality function $R(A/N, D/N, C/L)$. The problem of search for the minimal path for the described graph G_X with the weight system S_X is the search for the best structure among structures with arbitrary number of exons. Other considered problems also reduce to known problems of paths analysis in the graph G_X with the weight system S_X.

Unfortunately there is no distributivity between the introduced operations $\otimes$ and $\prec$, since R is monotonically increasing in the components A, D, C and monotonically decreasing in L and N. Thus it is not possible to consider a single path for each vertex and we use the approach suggested in (Roytberg, 1992) (see also (Hirschberg, 1975; Avdoshin et al., 1984). This appoach employs the notion of the *base set* of path, that is, such a set $\mathcal{M}$ of initial paths, that there exists a minimal full path extending one of the paths from $\mathcal{M}$. The algorithm constructs a sequence of base sets $\mathcal{M}_1, \ldots, \mathcal{M}_m$ until we obtain a base set in which it is simple to find the desired path. The set $\mathcal{M}_{i+1}$ is obtained from $\mathcal{M}_i$ by adding new elements and deleting of unnecessary paths.

Here is an example of an "unnecessity" condition for a path $\mathbf{p}$: there exists $\mathbf{p}' \in \mathcal{M}$ such that for any extension $\mathbf{r}$ of the path $\mathbf{p}$ there exists an extension $\mathbf{r}'$ of the path $\mathbf{p}'$ such that $\lambda(\mathbf{r}') \leq \lambda(\mathbf{r})$.

Conditions of this sort are called conditions of domination of $\mathbf{p}'$ over $\mathbf{r}$. Other conditions of unnecessity of a path $\mathbf{p}$ in the base set $\mathcal{M}$ are possible. For instance, in (Roytberg, 1992) one of the conditions of deleting a path $\mathbf{p}$ from the base set $\mathcal{M}$ is based on the connection between $\mathbf{p}$ and the history of $\mathcal{M}$. In our algorithm, however, only the domination conditions (**T**) and (**NT**) are employed;

The problem of path analysis in a non-distributive case was considered in (Lengauer and Theune, 1991), where a notion of a *grouping function* was introduced. Roughly speaking, the grouping function defines such subsets of the set K, that possess distributivity, and thus can be substituted for by a single "typical representative". In our case these are structures with coinciding number of exons $N(\mathbf{s})$ and of similar length $L(\mathbf{s})$.

The above approach can be employed not only to the problem of the search for an optimal path, but to the more general problem of the statistical sum computation, (Finkestein and Roytberg, this volume). However, when applied to the optimal path problem, it does not allow to connect by the domination relation paths ending in different vertices (compare it with condition (**NT**)).

Finally, we point out three properties of the graph G_X and the weight system S_X that are used in our algorithms. We need

Definition. An oriented graph $G = (V, E)$ is called *segmented* if

(i) V is a set of segments of a straight line;

(ii) if there is an arc from $[a, b]$ to $[c, d]$, then $b \leq c$.

The properties employed by our algorithm are:

(i) a weight $\lambda(e)$ is a vector, and the quality function is monotonic on each component;

(ii) a weight $\lambda(e)$ depends only on the end vertex of the arc e;

(iii) G_X is a segmented graph;

Property (1) allows to formulate the domination conditions. Property (2) allows to compare (in terms of (**NT**)) paths ending in different vertices. Property (3) is used in the notion of the left boundary $E(\mathbf{s})$

and allows to simplify the checking of the domination condition (**NT**).

On the other hand, *mutatis mutandis* our algorithm can be employed to non-segmented graphs. For an arbitrary graph the procedure of considering left brackets corresponds to looking over vertices in order of level (maximum length of a path from the source to this vertex) increase. When a vertex v is considered, the base set $\mathcal{M}$ is supplemented by paths of the type $\mathbf{p} \cdot (\mathrm{End}(\mathbf{p}), v)$, where $\mathbf{p} \in \mathcal{M}$.

Condition $E(\mathbf{r}) = E(\mathbf{s})$ in (**NT**) corresponds to the condition that the sets of inheritors Next(**r**) and Next(**s**) of the terminal vertices End(**r**) and End(**s**) coincide. Condition $(E(\mathbf{r}) < p \,\&\, E(\mathbf{s}) < p)$ corresponds to the situation when Next(**r**) and Next(**s**) may differ, but their intersections with the set of not considered vertices coincide.

Finally we formulate the condition on an arbitrary graph in order to be isomorphic to a segmented graph. For a arbitrary vertex $v \in V$ denote by More(v) the following set of vertices:

$$\mathrm{More}(v) = \{\, w \in V \mid \text{there exists a path from } v \text{ to } w \,\}.$$

Lemma 4. If the set $\{\, \mathrm{More}(v) \mid v \in V \,\}$ is fully ordered by inclusion, then the graph $G = (V, E)$ is isomorphic to a segmented graph.

Proof. Let $\{\, \mathrm{More}(v) \mid v \in V \,\}$ consist of N elements

$$M_1 \supset M_2 \supset \ldots \supset M_N.$$

Let $M_0 = V$ and for an arbitrary $v \in V$ let b_v equal the maximum i such that $v \in V$, and let e_v be defined by $\mathrm{More}(v) = M_{e_v}$. Obviously a path from v to w exists if and only if $e_v \leq b_w$, which proves the lemma.

Conclusion

The theoretical analysis of the computational effectivity of the suggested algorithm is difficult. Indeed, the number of generated structures is strongly dependent on the mutual arrangement of left and right brackets and their nature (consider trivial examples of sequences in which predicted donor and acceptor sites are arranged as follows: $(dd \ldots daa \ldots a)$, $(dada \ldots da)$ and $(aa \ldots add \ldots d)$). However, the preliminary computer experiments with the draft version of the program implementing the algorithm demonstrate that it decreases the search by at least the order of magnitude.

Moreover, as it has been already mentioned above, the constructed base set of structures contains an optimal structure for an arbitrary quality function satisfying the natural monotonicity conditions. Thus it is possible to to construct base sets for a large number of sequences and to use them as an input for a pattern recognition procedure, thus deriving the best structure quality function combining the diverse considered parameters. Although our situation is not the classical case for the pattern recognition theory (we do not require the universal threshold distinguishing correct and incorrect structures which would be impossible to derive, while for our purposes it would be sufficient if the correct structure would have the highest quality among structures generated *by the same sequence*), some modifications of the classical methods would work. In particular, in the class of linear quality functions it is possible to employ a simple modification of the generalized portrait approach (Alexandrov and Mironov, 1990).

Acknowledgements

We are grateful to Drs. P. A. Pevzner and M. Vingron for a very useful discussion, and anonimous referees for valuable suggestions. This work was partially supported by a grant from the Human Genome project of the Russia Academy of Sciences.

References

Aho, A., Hopcroft, J. and Ullman, J., 1976, The Design and Analysis of Computer Algorithms (Addison-Wesley, Reading, MA).

Alexandrov, N.N. and Mironov, A.A., 1990, Application of a new method of pattern recognition. Nucl. Acids Res. 18, 1847–1852.

Avdoshin, S.M., Belov, B.B. and Maslov, V.P., 1984, Mathematical aspects of software synthesis (VINITI, Moscow) (in Russian).

Berg, O. and von Hippel, P., 1987, Selection of DNA binding sites by regulatory proteins. Statistical-mechanical theory and application to operators and promoters. J. Mol. Biol. 193, 723–750

Fickett, J.W., 1982, Recognition of protein coding reg.ions in DNA sequences. Nucleic Acids Res. 10, 5303–5318.

Fields, C.A. and Soderlund, C.A., 1990, gm: a practial tool for automating DNA sequence analysis. Comput. Appl. Biosci. 6, 263–270.

Finkelshtein, A.V. and Roytberg, M.A., Computation of biopolymers: a general approach to different problems (this volume)

Gelfand, M.S., 1989, Statistical analysis of mammalian pre-mRNA splicing sites. Nucleic Acids Res. 17, 6369–6382.

Gelfand, M.S., 1990a, Computer prediction of the exon-intron structure of mammalian pre-mRNAs. Nucleic Acids Res. 18, 5865–5869.

Gelfand, M.S., 1990b, Global methods for the computer prediction of protein-coding regions in nucleotide sequences. Biotechnology Software 7(4), 4–11.

Gelfand, M.S., 1992a, Computer functional analysis of nucleotide sequences. II. Prediction of regulatory sites. Biofizika (in press) (in Russian).

Gelfand, M.S., 1992b, Statistical analysis and prediction of the exonic structure of human genes. J. Mol. Evol. 35, 239–252.

Gelfand, M.S., 1992c, Computer functional analysis of nucleotide sequences: problems and approaches, in: Mathematical Methods of the Analysis of Biopolymer Sequences (DIMACS, vol. 8), S.G. Gindikin (ed.) (AMS, Providence, RI), pp. 19–62.

Gelfand, M.S., 1992c, Prediction of protein-coding regions in DNA of higher eukaryotes, in: Mathematical Methods of the Analysis of Biopolymer Sequences (DIMACS, vol. 8), S.G. Gindikin (ed.) (AMS, Providence, RI), pp. 87–98.

Guigo, R., Knudsen, S., Drake, N. and Smith, T., 1992, Prediction of gene structure. J. Mol. Biol. 226, 141–157.

Hawkins, J.D., 1988, A survey of intron and exon lengths. Nucleic Acids Res. 16, 9893–9908.

Hirschberg, D.S., 1975, A linear space algorithm for computing maximum common subsequences. Commun. ACM 18, 341–343.

Legouis, R., Hardelin, J.-P., Levilliers, J., Claverie, J.-M., Compain, S., Wunderle, V., Millasseau, P., LePaslier, D., Cohen, D., Caterina, D., Bougueleret, L., Lutfalla, G., Weissenbach, J. and Petit C., 1991, The candidate gene for the X-linked Kallmann syndrome encodes a protein related to adhesion molecules. Cell 67, 423-435.

Lengauer, T. and Theune, D., 1991, Unstructured path problems and the making of semirings, in: Proceedings of the WADS'91.

Roytberg, M.A., 1992, Fast algorithm for optimal aligning of symbol sequences, in: Mathematical Methods of the Analysis of Biopolymer Sequences (DIMACS, vol. 8), S.G. Gindikin (ed.) (AMS, Providence, RI), pp. 113–126.

Rogozin, I.B., 1992, Prediction of exon-intron structure in nucleotide sequences of the human genome, report at workshop "Problems and Methods of Recognition of functionally important regions in nucleotide sequences of the human genome" (Novosibirsk, 1992).

Uberbacher, E. and Mural R., 1991, Locating protein coding segments in human DnA sequences by a multiple sensor–neural network approach. Proc. Natl. Acad. Sci. 88, 11261–11265.