# An Algorithm for Highly Specific Recognition of Protein-coding Regions

M. S. Gelfand [1] misha@imb.imb.ac.ru

T. V. Astakhova [2]

M. A. Roytberg [2] roytberg@impb.serpukhov.su

[1] Institute of Protein Research, Russian Academy of Sciences, Pushchino, 142292, Russia

[2] Institute of Mathematical Problems of Biology, Russian Academy of Sciences, Pushchino, 142292, Russia

**Abstract**

*Since absolutely reliable recognition of protein-coding regions in eukaryote genomic DNA sequences by computational methods is unattainable, most existing algorithms try to keep some balance between underprediction and overprediction. However, in experimental practice it is often sufficient to have just a few protein-coding segments, but predicted with high specificity, that is, with (almost) no overprediction. Such predictions are then used for construction of oligonucleotide probes and PCR primers for analysis of cDNA libraries or total cellular RNA.*

*Here we present a combinatorial algorithm solving this problem. Unlike other prediction schemes, the algorithm uses only the simplest statistical parameters (codon usage and positional nucleotide sequences in splicing sites) and thus can be used for analysis of obscure genomes, when large learning sets are unavailable. The algorithm's structure allows one to simply tune it for various experimental settings.*

## 1 Introduction

Recognition of protein-coding regions is one of traditional problems of computational molecular biology. Recently it gained additional importance caused by generation of large amount of unannotated DNA sequences by numerous sequencing projects, search for disease genes by positional cloning etc. The traditional approach to gene recognition is based on measuring statistical differences between protein-coding and non-coding sequences and analysis of statistical properties of exon-intron boundaries (splicing sites), reviewed in [1]. Currently there

exist more than a dozen packages and electronic servers for prediction of individual exons or complete genes. The recognition quality, defined as the average correlation between predicted and actual genes, usually does not exceed 70% [2], and unless a major breakthrough is made in understanding the mechanisms of splicing, there are no reasons to hope that it can be increased.

Such predictions can be useful, but in many cases there is no necessity to predict a complete gene, since it will be found experimentally. At the same time, prediction of relatively short protein-coding segments can be done with almost 100% reliability. In particular, such predictions can be used for synthesis of oligonucleotide probes or PCR primers with subsequent screening of cDNA libraries or total cellular RNA. The existing methods are not suitable for this task, since they do not allow to reliably determine a desired segment, if it is guaranteed only that overprediction on the average is approximately 20%, and in some cases it can be much higher.

Another problem is the fact that in many cases the methods based on application of neural networks or pattern recognition algorithms use a large number of complicated statistical parameters, and thus require a large learning set consisting of well-characterized sequences. Such samples are available if one works with traditional genomes (mammals, nematode *Caernorabditis elegans*, *Drosophila*), but they are absent for many important genomes (many invertebrates, plants, fungi, protists). Finally, most algorithms use linear scoring functions, although nonlinear functions provide better recognition [3].

These problems are addressed by an algorithm based on vector dynamic programming [4] and computation of partition function of path weights on a graph [5]. At the first step the algorithm constructs a set of exon chains (sub)optimal for some simple scoring function. The second step is based on the following observation: segments occurring in the majority of suboptimal genes are truly coding. Thus segment weights are recomputed using the partition function, and a small number of highest scoring segments is produced as the output, with the guarantee that some fixed number (usually one or two) of segments is truly coding. It should be noted that this formulation of the protein-coding recognition problem is stated here for the first time, although some existing algorithms can be re-shaped for such predictions.

This algorithm was implemented as a module in the GREAT package and tested on 50 long (10–30 thousand nucleotides) human DNA sequences. The first predicted segment was coding in 96% cases, the first two segments contained a coding one in all cases. To have two coding segments at a distance not less than a given one, it was sufficient to retain three segments in 86% cases and five segments in all cases but one. These results are comparable with the results by GRAIL [6], a neural network using many complicated statistical parameters.

# 2 Algorithm

Let $S$ be a set of candidate genes (exon chains), and let $S_b \subseteq S$ be a subset of genes containing nucleotide $b$. Each gene $p \in S$ is assigned a statistics-based weight $R(p)$ (see below). Score of nucleotide $b$ is computed as

$$U(b) = \sum_{p \in S_b} \exp(cR(p)),$$

where $c$ is a normalizing constant. Score of a segment $B = b_1 \dots b_k$ is defined as the average weight of the constituting nucleotides

$$W(B) = \frac{1}{k} \sum_{i=1}^{k} U(b_i).$$

Segment is called *locally optimal* if its weight is greater than the weights of all segments closer to it than by some fixed distance. The set of locally optimal segments is output as the prediction.

Note that these definitions do not depend on the choice of the gene weights $R$. If $R$ is additive, $S$ can be the set of all genes on the given sequence, and then nucleotide weights can be computed by the dynamic programming algorithm for computation of the partition function [5]. If $R$ is not additive, $U$ cannot be computed effectively, and the set $S$ should be reduced. Here we use the *Pareto set* $P$ of genes, guaranteed to contain the optimal gene for any weight function satisfying some natural conditions.

More exactly, let each candidate gene be described by a *set* of additive parameters $W_1, \dots, W_m$. We say that a gene $p$ *dominates* over a gene $r$ (denoted $p \succ r$) if $W_j(p) \geq W_j(r)$ for all $j = 1, \dots, m$ and at least one inequality is strict. The *Pareto-optimal* set $P$ contains all genes such that

- for any gene $r \notin P$ there exists a dominating gene $p \in P$: $p \succ r$;
- any two genes $p_1, p_2 \in P$ are incomparable: neither $p_1 \succ p_2$, nor $p_2 \succ p_1$.

The Pareto set can be constructed by the *vector dynamic programming* algorithm described in [4], [3]. It is simple to demonstrate [4] that for any function $R(W_1, \dots, W_m)$ monotonically increasing on its variables the Pareto-optimal set $P$ contains an optimal gene, and that it does not contain "unnecessary" genes, that is, for any gene $p \in P$ there exists a function $R$, for which it is optimal.

Thus, to compute nucleotide weights $U$ we use only genes from the Pareto-optimal set. We use the following gene parameters: length $L$ (in codons), number of exons $N$, coding potential $C$, total weights of donor and acceptor sites $D$ and $A$ respectively.

The coding potential was defined as follows. Let $f(abc)$ be the frequency of the codon $abc$ in the learning set. Codon weight is defined as

$$w(abc) = 100 \frac{\log f(abc) - \log f_{\min}}{\log f_{\max} - \log f_{\min}},$$

where $f_{\max}$ and $f_{\min}$ are the frequencies of the most frequent (resp. most rare) codons in the learning set. The coding potential of a gene $a_1 b_1 c_1 \dots a_L b_L c_L$ consisting of $L$ codons is defined as

$$C(a_1 b_1 c_1 \dots a_L b_L c_L) = \sum_{i=1}^{L} w(a_i b_i c_i).$$

We will need also the average weight of codons in the learning set $\mu_C$ and the standard deviation $\sigma_C$.

To define site weights, consider learning sets of splicing sites aligned by the exon-intron boundaries (donor and acceptor sites are considered separately). Let $n(b, i)$ be the frequency

of codon $b$ in alignment position $i$, and let $n^*(i)$ be the frequency of the consensus nucleotide, so that $n^*(i) = \max_b n(b,i)$. Weight of a site $b_1 \dots b_K$ is defined as

$$s(b_1 \dots b_K) = \sum_{i=1}^{K} \frac{n(b_i,i) + 0.5}{n^*(i) + 0.5}.$$

For a gene consisting of $N$ exons $D$ is the total weight of its $N-1$ donor sites, $A$ is the total weight of $N-1$ acceptor sites. The average weight of donor (acceptor) sites in the learning set is denoted by $\mu_D$ (resp. $\mu_A$), the standard deviations are denoted by $\sigma_D$ and $\sigma_A$ respectively.

Finally, the gene weight is computed as

$$R = \frac{D-(N-1)\mu_D}{(N-1)\sigma_D} + \frac{A-(N-1)\mu_A}{(N-1)\sigma_A} + \frac{C-L\mu_C}{L^{1/2}\sigma_C}.$$

# 3 Testing

## 3.1 Implementation

The algorithm has been implemented as a module in the GREAT package and is available from the authors by e-mail.

## 3.2 Test set

The test set consisted of 50 human sequences of length 10–30 thousands of nucleotides, each containing not more than one gene. All sequences from GenBank (as of Spring 1996) satisfying these conditions were considered. Alternatively spliced, incomplete,single exon genes and genes with abnormal splicing sites were not excluded.

## 3.3 Parameters and procedures

The parameters (codon frequencies and positional nucleotide frequencies in splicing sites) were taken from [3]. Each sequence was divided into overlapping fragments of length 4 thousand nucleotides. After prediction of coding segments (independently for each sequence fragment), the fragments were ordered by decrease of $\max R$ and fragments with $\max R$ lower that some fixed threshold were deleted. Segments of length 30 nucleotides at the minimum distance 70 nucleotides were predicted. We retained the best and second best segments for each fragment in the obtained order. Partially coding segments were considered as false predictions.

## 3.4 Benchmarking

The algorithm was compared with the GRAIL e-mail server [6] in the following way. The same sequences were submitted to GRAIL II. The predicted exons were ordered by decrease of scores, and the best segments were taken from the best (highest scoring) exon, the second best exon, an so on. Since GRAIL produces many ties, results were summarized in two ways. The optimistic estimate resolved all ties in favor of GRAIL (that is, if a true exon and a falsely predicted exon had the same scores, the true exon was assigned higher rank), the pessimistic estimate resolved all ties against GRAIL.

(a) GREAT

| | 1 | 2 | 3 | 4 | 5 | >5 | impossible |
|---|---|---|---|---|---|---|---|
| 1 | 47 | 3 | 0 | 0 | 0 | 0 | 0 |
| 2 | -- | 34 | 9 | 4 | 2 | 0 | 1 |

(b) GRAIL - optimistic resolution of ties

| | 1 | 2 | 3 | 4 | 5 | >5 | impossible |
|---|---|---|---|---|---|---|---|
| 1 | 47 | 3 | 0 | 0 | 0 | 0 | 0 |
| 2 | -- | 45 | 5 | 0 | 0 | 0 | 0 |

(c) GRAIL - pessimistic resolution of ties

| | 1 | 2 | 3 | 4 | 5 | >5 | impossible |
|---|---|---|---|---|---|---|---|
| 1 | 36 | 8 | 5 | 0 | 0 | 1 | 0 |
| 2 | -- | 32 | 11 | 4 | 2 | 1 | 0 |

Figure 1: Prediction results. The values in the cells show the number of candidate segments that should be considered in order to have the given number of coding segments (1 or 2).

### 3.5 Results

The overall results of testing are given in the table. If only one coding segment was needed, in 47 cases out of 50 the highest scoring segment was sufficient; the two best segments were sufficient in all cases. To have two coding segments, one had to consider 3 candidate segments in 43 cases, and 5 segments in 49 cases.

GRAIL results were slightly better than ours for the optimistic resolution of ties, and much worse for the pessimistic resolution. Thus the GREAT performance was at least comparable to that of GRAIL.

## 4 Discussion

The results of demonstrate that the algorithm reliably (with high specificity) finds coding segments in human DNA. With the probability close to 100% the output contains a given number of coding segments among very few candidates. It should be noted that the testing was deliberately performed in hard conditions, since we considered long sequences and did not exclude numerous anomalies.

Simplicity of the statistical base of the algorithm allows one to use it for analysis of less studied genomes when large learning sets are unavailable. Preliminary results demonstrate that the quality of recognition of *Drosophila* genes is the same as for human genes. Currently we are performing testing on plant and protist genes.

At the same time, combinatorial flexibility of the algorithm makes it possible to specifically tune it for various experimental designs. In particular, similar ideas can be applied for high sensitivity recognition, when loss of exons is less acceptable than overprediction (work in preparation).

## Acknowledgements

We are grateful to Dr. O. E. Evgrafov for discussion of experiments that initialized this work. Some procedures in GREAT were written by L. I. Podolsky, M. N. Semionenkov and D. O. Fomin.

This work was supported by grant 95/70 from Russian State Scientific Program "Human Genome", grant 94-04-12330 from Russian Fund of Fundamental Research, and in part by grant DE-FG-94ER61919 from DOE (USA).

## References

[1] M. S. Gelfand, "Prediction of function in DNA sequence analysis," *Journal of Computational Biology*, Vol. 2, pp. 87–115, 1995.

[2] M. Burset, R. Guigo, "Evaluation of gene structure prediction programs," *Genomics*, Vol. 31, 1996 (in press).

[3] M. S. Gelfand, L. I. Podolsky, T. V. Astakhova, M. A. Roytberg, "Recognition of genes in human DNA sequences," *Journal of Computational Biology*, Vol. 3, 1996 (in press).

[4] M. S. Gelfand, M. A. Roytberg, "Prediction of the exon-intron structure by a dynamic programming approach," *BioSystems*, Vol. 30, pp. 173–182, 1993.

[5] A. V. Finkelstein, M. A. Roytberg, "Computation of biopolyers: A general approach to different problems," *BioSystems*, Vol. 30, pp. 173–182, 1993.

[6] Y. Xu, R. J. Mural, M. Shah, E. C. Uberbacher, "Recognizing exons in genomic sequence using GRAIL II," *Genetic Engineering: Principles and Methods*, Vol. 16, pp. 241–253, New York, Plenum Press, 1987.